

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
2004



# WEB SERVER PERFORMANCE TUNING

ΟΝΟΜΑ:

ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ (ΑΕΜ 519)

e-mail: pitpap@otenet.gr

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΚΑΤΣΑΡΟΣ ΠΑΝΑΓΙΩΤΗΣ

Περιλαμβάνεται CD-ROM

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΠΕΡΙΕΧΟΜΕΝΑ

# ΠΕΡΙΕΧΟΜΕΝΑ

<i>ΠΡΟΛΟΓΟΣ</i> .....	7
-----------------------	---

## **ΜΕΡΟΣ ΠΡΩΤΟ**

### **ΚΕΦΑΛΑΙΟ 1<sup>ο</sup> - ΕΙΣΑΓΩΓΗ ΣΤΟ PERFORMANCE TUNING & OPTIMIZATION (ΡΥΘΜΙΣΗ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΤΗΣ ΑΠΟΔΟΣΗΣ)**

Σημασία του ΡΤΟ.....	13
Το ΡΤΟ ως τέχνη.....	15
Τι είναι το Bottleneck; .....	16
Tests.....	17
Performance, Load ή Stress Testing; .....	18
Χρήστες, συναλλαγές, σελίδες, hits .....	20
Γλωσσάρι.....	20

### **ΚΕΦΑΛΑΙΟ 2<sup>ο</sup> – ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΛΕΙΤΟΥΡΓΙΑΣ ΕΝΟΣ SERVER**

Βασική Λειτουργία.....	23
Στο Παρασκήνιο.....	23
Το Internet.....	24
Clients και Servers.....	25
Διευθύνσεις IP.....	26
Ονόματα Πεδίου (Domain names).....	26
Εξυπηρετές Ονομάτων (Name Servers).....	27
Θύρες (Ports).....	27
Ασφάλεια.....	28
Δυναμικές σελίδες (Dynamic pages) .....	29
Cookies .....	29
Firewalls .....	31
CGI Scripts .....	32
ASP (Active Server Pages).....	33
COM (Component Object Model).....	35

### **ΚΕΦΑΛΑΙΟ 3<sup>ο</sup> – Ο IIS 5.0 ΚΑΙ Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ**

Τι είναι ο IIS;.....	38
Πρωτόκολλο TCP/IP .....	39
Βασικές Υπηρεσίες IIS .....	40
Διαχείριση του IIS.....	42

Ικανότητα προγραμματισμού (programmability) .....	45
Νήματα στον IIS .....	47

## **ΚΕΦΑΛΑΙΟ 4<sup>ο</sup> – ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΑΠΟΔΟΣΗΣ (PERFORMANCE MONITORING)**

Κονσόλα MMC (Microsoft Management Console) .....	50
Objects, Counters και Instances .....	51
Περιγραφή Σημαντικών Performance Objects .....	53
Περιγραφή Σημαντικών Performance Counters .....	55
System Monitor Toolbar .....	58
Chart View .....	60
Report View .....	62
Performance Logs and Alert Views .....	63
Εξάγοντας τα δεδομένα απόδοσης .....	69

## **ΚΕΦΑΛΑΙΟ 5<sup>ο</sup> – ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ**

Τεχνικά χαρακτηριστικά υπολογιστών .....	71
Δημιουργία LAN .....	71
Εγκατάσταση μιας ASP εφαρμογής .....	76

## **ΚΕΦΑΛΑΙΟ 6<sup>ο</sup> – WEBSERVER STRESS TOOL 6.0**

Είδη test .....	80
Ρυθμίσεις τεστ .....	81
Επιλέγοντας τα URLs .....	83
Ρυθμίσεις Browser .....	89
Επιλογές (Options) .....	92
Τρέχοντας το τεστ .....	94
Επιθεώρηση των αποτελεσμάτων των Logfiles .....	95
Ανάλυση Γραφικών Αποτελεσμάτων .....	98
Δημιουργία Αναφορών (Reports) .....	104

# **ΜΕΡΟΣ ΔΕΥΤΕΡΟ**

## **ΚΕΦΑΛΑΙΟ 7<sup>ο</sup> – BOTTLENECKS ΕΠΕΞΕΡΓΑΣΤΗ**

Τι είναι το bottleneck επεξεργαστή; .....	109
Ενδείξεις για ανίχνευση ενός bottleneck επεξεργαστή .....	109
Εφαρμογές process-bound .....	111
Ανάλυση δραστηριότητας επεξεργαστή .....	112
Test #1 .....	114
Test #2 .....	118
Συνδέσεις (Connections) .....	120

Test #3.....	124
Νήματα.....	128
Test #4.....	130
Εφαρμογές ASP.....	133
Συμβουλές για την επίλυση των bottlenecks επεξεργαστή .....	134

## **ΚΕΦΑΛΑΙΟ 8<sup>ο</sup> – BOTTLENECKS ΜΝΗΜΗΣ**

To Working Set του IIS.....	138
ASP Caching .....	139
IIS Object Cache .....	139
Test #5.....	141
File System Cache.....	145
Test #6.....	147
Συμβουλές για τη βελτιστοποίηση της χρήσης μνήμης .....	148

## **ΚΕΦΑΛΑΙΟ 9<sup>ο</sup> – BOTTLENECKS ΔΙΣΚΟΥ**

Τι είναι το bottleneck δίσκου;.....	152
Ανιχνεύοντας τα bottlenecks δίσκου.....	152
Test #7.....	157
Τεχνολογίες disk-formatting .....	159
Συμβουλές.....	160

<b><i>BIBΛΙΟΓΡΑΦΙΑ</i></b> .....	163
----------------------------------	-----

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΠΡΟΛΟΓΟΣ

**Ό**που κι αν κοιτάξετε σήμερα στον εμπορικό κόσμο, όλοι μιλάνε για θέματα απόδοσης. Δεν έχει σημασία αν πρόκειται για τοπικά δίκτυα (LANs), δίκτυα ευρείας περιοχής (WANs), μητροπολιτικά δίκτυα (MANs), το Internet, έναν ατομικό server ή ακόμη κι ένα απλό εξάρτημα μέσα σε έναν server. Ο καθένας θέλει να έχει την απόδοση για την οποία έχει πληρώσει, ή αν είναι δυνατόν, ακόμη καλύτερη. Γι' αυτόν το λόγο η ρύθμιση και η βελτιστοποίηση της απόδοσης (PTO) είναι τόσο σημαντική. Ο χρόνος είναι χρήμα και ο καθένας θέλει να μειώσει το κόστος, μειώνοντας το χρόνο που απαιτείται για την εκτέλεση των ενεργειών του στο δίκτυό του.

Ο κλάδος του management παρακολουθεί με δέος το γεγονός ότι σήμερα χρειάζεται ο μισός χρόνος για να εκδοθεί μια αναφορά απ' ό,τι χρειαζόταν χθες. Αν ένας διαχειριστής μπορεί να ρυθμίσει ένα server ώστε να παρέχει ένα μικρό ποσοστό βελτίωσης της απόδοσης, το management θα το καταλάβει λόγω της οικονομίας στον εξοπλισμό και της αποδοτικότητας των χρηστών. Επιπλέον, οι χρήστες αντιδρούν θετικά όταν βλέπουν ότι οι εργασίες δικτύου ολοκληρώνονται σε χρόνο ρεκόρ. Με λίγα λόγια, οι γνώσεις και οι τεχνικές που πρόκειται να μάθετε από τη μελέτη αυτήν, είναι μια καλή αρχή για να μειώσετε το φόρτο εργασίας του δικτύου σας, να κερδίσετε την εύνοια του management (αν εργάζεστε σε εταιρία) και να γίνετε γνωστοί για τις ικανότητές σας στους ανθρώπους του δικτύου.

---

## Τι μπορείτε να μάθετε

Η εργασία αυτή έχει σαν στόχο να σας δώσει να καταλάβετε το τι είναι η ρύθμιση και η βελτιστοποίηση της απόδοσης ενός server και ποια είναι τα οφέλη από τη διαδικασία αυτήν. Το λογισμικό server που χρησιμοποιείται για την μελέτη αυτήν είναι ο IIS 5.0 της Microsoft, οπότε και παρουσιάζουμε την εσωτερική αρχιτεκτονική της λειτουργίας του (κεφάλαιο 3<sup>ο</sup>), έτσι ώστε ο αναγνώστης να έχει σαφή γνώση για το τι είναι στην πραγματικότητα αυτό που πρέπει να ρυθμίσουμε. Βασική προϋπόθεση για να επιτύχουμε την βελτιστοποίηση της απόδοσης του server είναι το τεστάρισμά του. Έτσι χρησιμοποιήσαμε το Webserver Stress Tool προκειμένου να διενεργήσουμε μια σειρά από τεστ, τα οποία μας βοηθάνε να καθορίσουμε τα όρια του server μας και να βρούμε τρόπους αύξησης της απόδοσης. Στην εργασία περιλαμβάνεται ένα αναλυτικό εγχειρίδιο του λογισμικού αυτού (κεφάλαιο 6<sup>ο</sup>). Όμως, για να δούμε πραγματικά τα αποτελέσματα των ρυθμίσεων που κάναμε στο server, θα πρέπει να παρακολουθήσουμε το σύστημά μας. Αυτό μπορεί να γίνει και μέσα από το λογισμικό Webserver Stress tool, αλλά κυρίως από το System Monitor των Windows, το οποίο μελετάται και περιγράφεται στο 4<sup>ο</sup> κεφάλαιο. Τέλος, αφού περιγράψαμε τον τρόπο με τον οποίο μπορούμε να στήσουμε ένα LAN, καθώς και τα συστήματα που χρησιμοποιήσαμε (κεφάλαιο 5<sup>ο</sup>), εξηγούμε τα τρία σημαντικότερα bottlenecks (επεξεργαστή, μνήμης, δίσκου) έτσι ώστε να δείξουμε,

στην πράξη πλέον, ποιες είναι αυτές οι ενέργειες που πρέπει να ακολουθήσουμε προκειμένου να φτάσουμε στην πολυπόθητη βελτιστοποίηση της απόδοσης του server (Μέρος Δεύτερο – Πειραματικό Μέρος). Τα θέματα με τα οποία καταπιάνεται η μελέτη μας είναι αρκετά επεξηγηματικά, καθιστώντας την έτσι προσιτή και στους αρχάριους αναγνώστες (για παράδειγμα στο 2<sup>ο</sup> κεφάλαιο εξηγούνται κάποιες βασικές έννοιες που θα βοηθήσουν τους πιο αρχάριους, σε περίπτωση που αυτοί αντιμετωπίσουν δυσκολία στην κατανόησή τους, όταν αυτές αναφέρονται κάπου, σε κάποιο άλλο κεφάλαιο). Τελικά η μελέτη αυτή προσπαθεί να σας εφοδιάσει με όλες τις κατάλληλες γνώσεις ώστε να μπορέσετε να βελτιώσετε την απόδοση του δικτύου σας, είτε αν έχετε μια απλη σελίδα στο Internet είτε αν διατηρείτε μια επιχείρηση σε οποιοδήποτε δίκτυο κι επομένως θέλετε να εξοικονομήσετε χρόνο και χρήμα.

---

## Σε ποιους απευθύνεται

Σε όσους επιθυμούν να μάθουν τους τρόπους με τους οποίους μπορεί να επιτευχθεί η βελτιστοποίηση της απόδοσης ενός server. Έτσι, είτε είστε ιδιοκτήτης κάποιας σελίδας στο Internet και θέλετε απλά να την υποστηρίξετε με το δικό σας server είτε διαθέτετε μια επιχείρηση στο δίκτυο, μπορείτε να μάθετε τις τεχνικές που θα σας οδηγήσουν στο επιθυμητό αποτέλεσμα, δηλαδή την αύξηση της απόδοσης των υπηρεσιών που θέλετε να παρέχετε.

Η εργασία απευθύνεται σε γνώστες χρήσης των Windows και των βασικών αρχών λειτουργίας των δικτύων γενικά, αλλά και σε πιο αρχάριους χρήστες, οι οποίοι απλά θα πρέπει να έχουν κάποιες βασικές γνώσεις χρήσης των Windows.

---

## Δομή της εργασίας

Τα κεφάλαια οργανώνονται ως εξής:

### **ΜΕΡΟΣ ΠΡΩΤΟ**

Το πρώτο μέρος της εργασίας είναι καθαρά θεωρητικό και στόχος του είναι να σας εφοδιάσει με όλες τις γνώσεις, ώστε να μπορέσουμε να περάσουμε πλέον στο πειραματικό μέρος, όπου θα προσπαθήσουμε να περιγράψουμε τους τρόπους λύσης των πιο σημαντικών bottlenecks που μπορεί να αντιμετωπίσετε.



## **Κεφάλαιο 1 – Εισαγωγή στο Performance Tuning και Optimization (Ρύθμιση και Βελτιστοποίηση της απόδοσης)**

Αυτό το κεφάλαιο περιέχει την πληροφορία που χρειάζεστε για να καταλάβετε τη σημασία του PTO (Performance Tuning & Optimization) και πώς αυτό επηρεάζει την απόδοση του server. Ο κύριος σκοπός του κεφαλαίου είναι να σας παρέχει έναν καλό λόγο για να βελτιστοποιήσετε το σύστημά σας.

## **Κεφάλαιο 2 – Βασικές Αρχές Λειτουργίας ενός Server**

Στο κεφάλαιο αυτό περιγράφονται τα βασικά χαρακτηριστικά της λειτουργίας ενός server, καθώς και κάποιοι βασικοί όροι και στοιχεία του Internet, η κατανόηση των οποίων είναι σημαντική για τη συνέχεια. Το κεφάλαιο απευθύνεται σε αρχάριους, οπότε οι πιο έμπειροι μπορούν να παραβλέψουν την ανάγνωσή του.

## **Κεφάλαιο 3 – Ο IIS 5.0 και η αρχιτεκτονική του**

Στο κεφάλαιο αυτό θα μελετήσουμε την αρχιτεκτονική του IIS, του server δηλαδή που θα χρησιμοποιήσουμε σε παρακάτω κεφάλαια προκειμένου να περιγράψουμε προβλήματα του δικτύου και να πραγματοποιήσουμε πειράματα που θα βοηθήσουν στην επίλυσή τους.

## **Κεφάλαιο 4 – Παρακολούθηση Απόδοσης (Performance Monitoring)**

Οποιοδήποτε σύστημα κι αν μελετήσουμε, δεν μπορούμε να βελτιώσουμε την απόδοσή του, αν πρώτα δεν το μετρήσουμε. Σ' αυτό το κεφάλαιο θα δείτε με ποιον τρόπο μπορείτε να συλλέξετε τα δεδομένα που θα σας βοηθήσουν στη συνέχεια, στη δύσκολη δουλειά της βελτίωσης της απόδοσης του συστήματός σας.

## **Κεφάλαιο 5 – Περιγραφή Συστήματος**

Στο κεφάλαιο αυτό παρουσιάζουμε τα τεχνικά χαρακτηριστικά του συστήματος που χρησιμοποιήσαμε στα πειράματα που ακολουθούν στο δεύτερο μέρος της εργασίας αυτής, ώστε να βελτιώσουμε την απόδοση του server μας.

## **Κεφάλαιο 6 – Webserver Stress Tool**

Το Webserver Stress Tool 6.0 (της εταιρίας Paessler) είναι ένα πολύ χρήσιμο εργαλείο, το οποίο μπορεί να προσομοιώσει σχεδόν οποιονδήποτε αριθμό από χρήστες που προσβαίνουν σε ένα website μέσω HTTP/HTTPS. Στο κεφάλαιο αυτό παρουσιάζουμε ένα tutorial του εργαλείου, για να περάσουμε έπειτα στο πειραματικό μέρος.

## **ΜΕΡΟΣ ΔΕΥΤΕΡΟ**

Στο δεύτερο μέρος της εργασίας (πειραματικό μέρος), παρουσιάζουμε τα τρία σημαντικότερα bottlenecks που είναι δυνατόν να συναντήσετε, προτείνουμε τρόπους λύσης και διενεργούμε κάποια πειράματα ώστε να γίνουν κατανοητά στον αναγνώστη τα θέματα που περιγράφουμε.

### **Κεφάλαιο 7 –Bottlenecks Επεξεργαστή**

Θα μπορούσαμε να πούμε ότι τα bottlenecks επεξεργαστή είναι ο μεγαλύτερος φόβος για έναν διαχειριστή δικτύου, καθώς επηρεάζουν ολόκληρο το σύστημα. Με τις γνώσεις που σας παρέχονται στο κεφάλαιο αυτό, μπορείτε να κατανοήσετε τους λόγους για τους οποίους προκύπτει ένα bottleneck επεξεργαστή και τους τρόπους με τους οποίους μπορείτε να το αντιμετωπίσετε.

### **Κεφάλαιο 8 –Bottlenecks Μνήμης**

Στο κεφάλαιο αυτό δίνονται συμβουλές σχετικά με την αντιμετώπιση αυτού του είδους των bottlenecks. Εφόσον ο IIS χρειάζεται μνήμη για να τρέξει εφαρμογές που εξυπηρετούν μεγάλο αριθμό χρηστών και, αφού η μνήμη κοστίζει, οι τρόποι αντιμετώπισης των bottlenecks μνήμης που σας προτείνουμε μπορούν να σας βοηθήσουν να επιλύσετε τα προβλήματά σας, εξοικονομώντας χρήματα από πιθανή αγορά μεγαλύτερης μνήμης.

### **Κεφάλαιο 9 –Bottlenecks Δίσκου**

Αφού πρώτα έχετε ήδη αποκλείσει την περίπτωση ύπαρξης των δύο προηγούμενων bottlenecks και ενώ το πρόβλημα απόδοσης εξακολουθεί να υπάρχει, τότε είναι πολύ πιθανό να αντιμετωπίζετε ένα bottleneck δίσκου. Στο κεφάλαιο αυτό, όπως και στα προηγούμενα, παρατίθενται συμβουλές επίλυσης προβλημάτων απόδοσης που οφείλονται στο σκληρό σας δίσκο.

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΜΕΡΟΣ ΠΡΩΤΟ

**ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 1

**ΕΙΣΑΓΩΓΗ ΣΤΟ**  
**PERFORMANCE TUNING & OPTIMIZATION**  
**(ΡΥΘΜΙΣΗ ΚΑΙ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΑΠΟΔΟΣΗΣ)**

Αυτό το κεφάλαιο περιέχει την πληροφορία που χρειάζεστε για να καταλάβετε τη σημασία του **PTO (Performance Tuning & Optimization)** και πώς επηρεάζει την απόδοση του server. Ο κύριος σκοπός του κεφαλαίου είναι να σας παρέχει έναν καλό λόγο για να βελτιστοποιήσετε το σύστημά σας. Οποιαδήποτε εργασία γίνεται ευκολότερη, όταν έχετε προσωπικό λόγο για να φροντίσετε και τη γνώση για να καταλάβετε γιατί είναι τόσο σημαντική αυτή η εργασία. Το κεφάλαιο αυτό, επίσης, εξηγεί βασικές έννοιες του PTO, εξηγεί τη σημασία των τεστ, στα οποία πρέπει να υποβάλλουμε το σύστημά μας ώστε να καταλήξουμε σε βελτιστοποίηση της απόδοσης του server και τέλος, ορίζει κάποιες παραμέτρους που θα χρησιμοποιηθούν στη συνέχεια.

---

## Σημασία του PTO

Καμιά άλλη τεχνολογία δεν έχει γνωρίσει παρόμοια άνθιση με αυτή που γνώρισε το web. Το 1993, μερικά μόνο sites άρχισαν να εμφανίζονται, ενώ το 1994 υπήρχαν λιγότερα από 1000 sites στο Internet. Τον Ιανουάριο του 1999 υπήρχαν τουλάχιστον 4 εκατομμύρια web servers στο Internet, σύμφωνα με την καταμέτρηση της Netcraft, 79 εκατομμύρια χρήστες στο web και 48 εκατομμύρια web shoppers, σύμφωνα με την καταμέτρηση του Nielsen Media Research and CommerceNet. Ο William W. Priest, διευθυντής του Credit Suisse Asset management, προσπαθώντας σε μια δήλωσή του να εξηγήσει τις υψηλές εκτιμήσεις για το Internet, αναφέρθηκε σε ένα διάγραμμα του Dallas Fed: “Το Internet έκανε 7 χρόνια για να φτάσει το 25% του πληθυσμού, εκεί που κάποιες από τις μεγαλύτερες εφευρέσεις του παρελθόντος έκαναν περίπου 30 χρόνια”.

Ακόμη παραπέρα, το μοντέλο διαμοίρασης πληροφοριών του Internet έχει εισαχθεί σε κάθε οργανισμό με τη μορφή του Intranet. Όλο και περισσότερες εταιρίες χρησιμοποιούν το Intranet για διαμοίραση πληροφορίας και για την επαύξηση της συνεργασίας μεταξύ των διαφόρων τμημάτων τους. Αυτή η τεράστια ανάπτυξη του Internet και του Intranet καθιστά τον web server μια κρίσιμη εφαρμογή σχεδόν για κάθε οργανισμό, οπότε επιβάλλεται να λειτουργούν αυτοί με τη μέγιστη απόδοση. Την ίδια στιγμή, ο φόρτος εργασίας που έχει ανατεθεί στους web servers εκτοξεύεται στα ύψη και καθιστά έτσι πολύ δύσκολη τη ρύθμισή τους ώστε να τρέχουν με τη μέγιστη απόδοση.

Παρακάτω παρατίθενται μερικά πλεονεκτήματα που σας παρέχονται όταν διαθέτετε έναν καλά ρυθμισμένο server στο δίκτυό σας:

- **Μικρότερο κόστος εξοπλισμού**

Προβλήματα απόδοσης εμφανίζονται σε κάθε δίκτυο. Κάποιοι παραπληροφορημένοι διαχειριστές δικτύου επιχειρούν να πετάξουν hardware όταν βλέπουν τέτοια προβλήματα – αυξάνουν RAM, βάζουν πρόσθετο δίσκο ή ακόμη αντικαθιστούν, χωρίς να είναι απαραίτητο, έναν αμφισβητούμενο server. Όλα αυτά τα κάνουν σε μια προσπάθεια να φτιάξουν το πρόβλημα απόδοσης, ενώ το πρόβλημα μπορεί να λυθεί καλύτερα αντικαθιστώντας απλώς μια παλιά εφαρμογή. Τέτοιες λύσεις hardware μπορεί να λύνουν το πρόβλημα βραχυπρόθεσμα, αλλά σπάνια προσφέρουν την καλύτερη λύση σε μια κατάσταση όπου η ακριβής αιτία της φτωχής απόδοσης παραμένει ακόμη άγνωστη.

- **Αυξημένη παραγωγικότητα χρήστη**

Ένας κακώς ρυθμισμένος server μπορεί να οδηγήσει σε μια παύση συστήματος (downtime). Αυτή η παύση συστήματος μεταφράζεται ως δαπάνη στον επιχειρηματικό κόσμο. Όταν ένα ολόκληρο τμήμα δεν μπορεί να έχει πρόσβαση σε πόρους του server, η εταιρία μπορεί να χάσει εκατομμύρια δολάρια μέσα σε μια μόνο μέρα. Ένας server που παρουσιάζει έναν υποβιβασμό του συστήματος μπορεί να φέρει ως αποτέλεσμα μείωση της παραγωγικότητας των χρηστών. Προκειμένου να πετύχετε το στόχο της υψηλής παραγωγικότητας, είναι επιτακτικό να δημιουργήσετε μια στρατηγική. Ακόμη και οι οργανισμοί με υψηλές απαιτήσεις απόδοσης μπορούν να επωφεληθούν από μια στρατηγική βασισμένη στο PTO.

- **Βελτιωμένη στάση χρηστών**

Ένας server με μικρή απόκριση ή ένας server που είναι συνεχώς offline ή απαιτεί τακτική επανεκκίνηση, μπορεί να προκαλέσει απογοήτευση στους τελικούς χρήστες. Οι χρήστες του συστήματος θα πρέπει να είναι σε θέση να συγκεντρώνονται στις εργασίες που τους ανατίθενται, χωρίς να ανησυχούν για το ότι οι διαθέσιμες υπηρεσίες ή οι ατομικοί servers μπορεί να αποτύχουν ανά πάσα στιγμή. Η μείωση της συχνότητας που τίθενται εκτός λειτουργίας οι κρίσιμες υπηρεσίες, βελτιώνει τη στάση των χρηστών. Ως διαχειριστής συστήματος, έχετε την ικανότητα να επηρεάζετε τη στάση των χρηστών μέσω του PTO.

- **Μεγαλύτερη αξιοπιστία λόγω του μειωμένου stress του εξοπλισμού**

Κάθε ένας από εμάς νιώθει stress κάποια στιγμή της ημέρας. Όμως, ο εξοπλισμός νιώθει stress; Στην πραγματικότητα, το stress του εξοπλισμού μπορεί να προέλθει από την υπερχρησιμοποίηση (over utilization) ή από κακή ρύθμιση του server. Ένα παράδειγμα stress εξοπλισμού είναι ένα bottleneck δίσκου που επηρεάζει τους σκληρούς δίσκους του συστήματός σας. Αν προκληθεί ένα bottleneck στον υπολογιστή σας, σημαίνει ότι το λειτουργικό σας σύστημα συνεχώς προσβαίνει στους σκληρούς δίσκους. Αυτό μπορεί να έχει ως αποτέλεσμα τη μειωμένη

αξιοπιστία των οδηγών σκληρού δίσκου. Η αξιοπιστία του οδηγού μπορεί να δοκιμαστεί, αν οι δίσκοι χρησιμοποιούνται παρατεταμένα και ανάρμοστα εξαιτίας απαραίτητων χειροκίνητων σταματημάτων λειτουργίας, με το λειτουργικό σύστημα σε απρόσιτη κατάσταση. Τυπικά, επανειλημμένες 'βίαιες' επανεκκινήσεις προκαλούν stress στο σκληρό δίσκο.

- **Μειωμένα σφάλματα εφαρμογής**

Ο server σας μπορεί να υφίσταται ένα bottleneck επεξεργαστή, δίσκου ή κάποιο άλλο πρόβλημα ρύθμισης απόδοσης, αλλά μπορεί να μην βρίσκεται στην κατάσταση που φοβούνται όλοι οι διαχειριστές συστήματος, στην κατάσταση 'πτώσης συστήματος'. Οι τελικοί χρήστες μπορεί να καταλαβαίνουν ότι το δίκτυο τρέχει λίγο αργά, αλλά μπορούν ακόμη να διεκπεραιώνουν την εργασία τους. Η παραγωγικότητα παύεται ολοκληρωτικά ως αποτέλεσμα των σφαλμάτων εφαρμογής. Το λειτουργικό σύστημα μπορεί να τρέχει και οι χρήστες να μπορούν να συνδεθούν με τους πόρους του server, αλλά η εφαρμογή με την οποία προσπαθούν να επικοινωνήσουν είναι απροσέγγιστη. Αυτός ο τύπος της μη διαθεσιμότητας της εφαρμογής οδηγεί σε απογοήτευση από τη μεριά των τελικών χρηστών και, λανθασμένα, μπορεί να αναφερθεί ως πρόβλημα του server.

---

## Το PTO ως τέχνη

Υπάρχουν τόσοι τρόποι για να ρυθμίσετε την απόδοση ενός web server όσες είναι και οι σελίδες στο Internet. Ανάλογα με τις επιλογές που έχει κάνει η εταιρία σας σχετικά με την παρουσία της στο Web, είστε υπεύθυνοι να ρυθμίσετε τους web servers ώστε να εξυπηρετούν με τον καλύτερο τρόπο στατικές σελίδες ή δυναμικά δημιουργημένες σελίδες εφαρμογών. Κάθε τύπος site απαιτεί διαφορετικό hardware και διαφορετικές ρυθμίσεις απόδοσης του λογισμικού server (στη μελέτη αυτή έχει χρησιμοποιηθεί ο IIS server της Microsoft, οπότε θα αναφερόμαστε σε αυτόν). Ένας ακόμη παράγοντας που πρέπει να ληφθεί υπ' όψη είναι η ποσότητα του traffic που περιμένετε να διαχειρίζεται το site ρεαλιστικά, ιδίως κατά τη διάρκεια περιόδων peak load (αυξημένου φόρτου). Ο φόρτος επηρεάζει την απόδοση του web server και, ανάλογα με τις επιλογές της επιχείρησης, όπως το μέγεθος της διαφημιστικής καμπάνιας της εταιρίας σας, μπορεί να καθορίσει τον αριθμό των αιτήσεων που πρέπει να είναι σε θέση να διαχειρίζονται τα web sites. Πρέπει να έχετε μια καλή ιδέα για το πώς πρόκειται να είναι αυτά τα φορτία και να τα προσομοιώνεται στους server σας πριν τους θέσετε σε λειτουργία. Αυτός είναι ο κύριος λόγος που δεν υπάρχει κάποιος τυποποιημένος τρόπος ρύθμισης της απόδοσης ενός web server.

Το PTO ενός server μπορεί να το δει κανείς τόσο ως επιστήμη όσο και ως τέχνη. Η δοκιμή και το σφάλμα μπορούν να είναι μια σημαντική τεχνική για τον καθορισμό των

ρυθμίσεων και του hardware που λειτουργούν καλύτερα για τις απαιτήσεις του site σας. Ενώ είναι κρίσιμο να καταλάβετε τις τεχνικές ρυθμίσεις, είναι εξ ίσου σημαντικό να καταλάβετε το προφίλ των εφαρμογών ή των sites σας και πώς συμπεριφέρονται αυτά κάτω από διαφορετικές συνθήκες. Όπως ένας ζωγράφος σχεδιάζει με καρβουνάκι για να αναπτύξει μια αίσθηση του πώς θέλει να δημιουργήσει τον πίνακα, έτσι κι εσείς πρέπει να έχετε ένα σχέδιο για την εκτίμηση της απόδοσης του web server σας. Το πρώτο βήμα είναι να διαμορφώσετε ένα ελεγχόμενο περιβάλλον στο οποίο θα τεστάρετε το site σας, θα διεξάγετε την ανάλυση απόδοσης των προβλεπόμενων φορτίων και θα μετράτε την απόδοση σ' αυτό το περιβάλλον πριν εκθέσετε το server σας στο Internet. Έπειτα θα πρέπει να ακολουθήσετε έναν αριθμό ενεργειών, προκειμένου να κάνετε την απόδοση του server σας μέγιστη για τις απαιτήσεις που έχετε θέσει. Η τέχνη του PTO βρίσκεται ακριβώς στην εύρεση αυτής της ακολουθίας ενεργειών, καθώς αυτή είναι διαφορετική για κάθε περίπτωση και απαιτείται αρκετή μελέτη και πειραματισμός προκειμένου να επιτευχθεί η βελτιστοποίηση της απόδοσης του web server.

---

## Τι είναι το Bottleneck;

Σε ένα μπουκάλι μύρας θα έχετε προσέξει πώς ο λαιμός (neck) περιορίζει τη ροή του υγρού. Παρόμοια, τα bottlenecks στους υπολογιστές αναφέρονται σε μέρη του υπολογιστή που περιορίζουν τη ροή των δεδομένων. Ο στόχος σας ως διαχειριστές συστήματος είναι να μειώσετε τα bottlenecks όσο το δυνατόν περισσότερο.

Υπάρχουν τα παρακάτω ήδη bottlenecks:

- **Bottlenecks επεξεργαστή**

Η πιο απλή μορφή bottleneck επεξεργαστή είναι όταν η χρησιμοποίηση του επεξεργαστή είναι η μέγιστη δυνατή. Ωστόσο, bottlenecks επεξεργαστή μπορούν να υπάρξουν και σε χαμηλότερα επίπεδα χρησιμοποίησης, αλλά είναι πολύ δύσκολο να διαγνωστούν και να διορθωθούν τέτοια προβλήματα.

- **Bottlenecks μνήμης**

Τα bottlenecks μνήμης μπορεί να οφείλονται σε έλλειψη RAM και μπορεί να οδηγήσουν στη μείωση της απόδοσης του συνολικού συστήματος.

- **Bottlenecks δίσκου**

Οι σκληροί δίσκοι αποθηκεύουν προγράμματα και δεδομένα που υφίστανται επεξεργασία από τα προγράμματα. Μια μικρή παύση σε μια disk-intensive εφαρμογή που τρέχει, μπορεί να δημιουργήσει ένα bottleneck δίσκου. Επιπλέον, αν δεν υπάρχει αρκετή μνήμη, ο δίσκος κάνει chattering (δηλαδή ακούγεται να δουλεύει παρατεταμένα και ίσως να εμφανίζεται η κλεψύδρα στα Windows που



υποδηλώνει ότι το λειτουργικό εκτελεί κάποια εργασία), που συνήθως είναι αποτέλεσμα των εναλλαγών δεδομένων με την εικονική μνήμη.

- **Bottlenecks δικτύου**

Υπάρχουν τόσα μέρη σε ένα δίκτυο, επομένως είναι πολύ δύσκολο να βρούμε το bottleneck ή ακόμη να βρούμε πόσα bottlenecks υπάρχουν. Μια από τις πολυπλοκότητες κατά το σχεδιασμό ενός δικτύου, είναι το να μπορούμε να κρατάμε τα ξεχωριστά μέρη ώστε να δουλεύουν όλα μαζί. Έτσι λοιπόν, τα bottlenecks δικτύου είναι τα πιο δύσκολα για να εντοπιστούν.

---

## Tests

Τα περισσότερα websites και οι δικτυακές εφαρμογές τρέχουν ομαλά και σωστά εφόσον μόνο ένας χρήστης (π.χ ο αρχικός δημιουργός) ή λίγοι μόνο χρήστες τις χρησιμοποιούν μια δεδομένη στιγμή. Όμως τι συμβαίνει αν χιλιάδες χρήστες προσβαίνουν στο website ταυτόχρονα;

Σήμερα, πολλά websites έχουν μια σοβαρή επιχειρηματική αποστολή και χρησιμοποιούνται κυρίως για να πουλήσουν ένα προϊόν ή μια υπηρεσία, πράγμα που σημαίνει ότι ο στόχος είναι το χρήμα. Οποιοσδήποτε τρέχει μια επιχείρηση στο δίκτυο, θα πρέπει να γνωρίζει τις στατιστικές σχετικά με το πόσο γρήγορα αλλάζει ένας πελάτης σε μια σελίδα κάποιου ανταγωνιστή αν ο webserver αντιδρά πολύ αργά ή καθόλου. Εσείς δεν ακυρώνετε μια συναλλαγή ενός περοϊόντος, αν ο webserver δεν αντιδράσει μέσα σε 5-20 δευτερόλεπτα; Αν οι επισκέπτες της σελίδας σας γνωρίζουν πριν από σας ότι η σελίδα σας είναι αργή, τότε θα αυτό θα σας κοστίσει πολλά χρήματα.

Το μήνυμα, λοιπόν, είναι ξεκάθαρο για όλους τους ιδιοκτήτες websites: **Να τεστάρετε και να παρακολουθείτε το website σας.**

Τα τεστ λειτουργίας (functionality testing), τα τεστ ευκολίας χρήσης (usability testing) και τα τεστ απόδοσης (performance testing) είναι εξίσου σημαντικά αλλά συχνά παραβλέπονται. Οι ιδιοκτήτες των websites θα πρέπει, όχι μόνο να ελέγχουν αν το website κάνει αυτό για το οποίο προορίζεται (functionality testing) και αν ο χρήστης είναι σε θέση να καταλαβαίνει τι γίνεται στον browser του client (usability testing), αλλά θα πρέπει να ελέγχουν και αν ο χρήστης λαμβάνει τα αποτελέσματα από το website μέσα σε αποδεκτό χρόνο (performance testing).

**Θα πρέπει να διασφαλίσετε ότι ο πελάτης σας παίρνει πάντα τη σωστή απάντηση σε κάθε mouse κλικ μέσα σε μερικά δευτερόλεπτα. Διασφαλίστε ότι το 95% των αιτήσεων υφίστανται επεξεργασία σε λιγότερο από 10 δευτερόλεπτα.**

Σύμφωνα με τον Jakob Nielsen:

<b>Χρόνος download</b>	<b>Η άποψη του χρήστη</b>
$<0.1 s$	Ο χρήστης νιώθει ότι το σύστημα αντιδρά στιγμιαία.
$<1.0 s$	Ο χρήστης βιώνει μια μικρή καθυστέρηση αλλά παραμένει συγκεντρωμένος στη συγκεκριμένη σελίδα.
$<10 s$	Μελέτες δείχνουν ότι αυτός είναι ο μέγιστος χρόνος για τον οποίο ο χρήστης παραμένει συγκεντρωμένος στο website, αλλά μπορεί να έχει ήδη αποσπαστεί η προσοχή του.
$>10 s$	Το πιο πιθανό είναι ο χρήστης να έχει ήδη αποσπαστεί από το συγκεκριμένο site και να έχει χάσει το ενδιαφέρον του.

Χρησιμοποιώντας εργαλεία, όπως το Webserver Stress Tool που θα χρησιμοποιήσουμε στην παρούσα μελέτη και το οποίο περιγράφεται εκτενώς στο 6<sup>ο</sup> κεφάλαιο, μπορείτε να είστε σε θέση να γνωρίζετε αν έχετε κάποιο πρόβλημα απόδοσης (λόγω προβλημάτων φόρτου) και να βρίσκετε τρόπους επίλυσης ενός προβλήματος.

---

## Performance, Load ή Stress Testing;

Θα επιμείνουμε να χρησιμοποιούμε τους αγγλικούς όρους για αποφυγή παρεξηγήσεων.

Αυτές οι τρεις λέξεις συχνά χρησιμοποιούνται ως συνώνυμες, αλλά υπάρχουν διαφορές:

### Performance Tests

Τα performance tests χρησιμοποιούνται για να τεστάρουν κάθε μέρος του webserver ή της δικτυακής εφαρμογής ώστε να ανακαλύψουν ποιά μέρη του website είναι αργά και πώς μπορείτε να τα κάνετε γρηγορότερα. Τις περισσότερες φορές αυτό γίνεται τεστάροντας τις διάφορες υλοποιήσεις των σελίδων και των scripts, ώστε να βρεθεί ποιά έκδοση του κώδικα είναι η γρηγορότερη.

Το Webserver Stress tool υποστηρίζει αυτόν τον τύπο των τεστ έχοντας την ικανότητα να τρέχει 20-40 ταυτόχρονες αιτήσεις σε ένα URL και καταγράφοντας το μέσο χρόνο έως ότου απαντηθούν οι αιτήσεις αυτές. Αλλάζοντας τον κώδικα του προγράμματος, είστε σε θέση να ανακαλύψετε όλα τα θέματα που πρέπει να διευθετηθούν προκειμένου να επιτύχετε περισσότερη απόδοση για το web site σας. Συνήθως αυτός ο τύπος τεστ εκτελείται χωρίς να ζητά όλες τις εικόνες, κλπ. σε μια σελίδα, πράγμα που σας επιτρέπει να παίρνετε πιο ακριβή αποτελέσματα.

### **Load Tests**

Στα load tests, το website τεστάρεται χρησιμοποιώντας το καλύτερο εκτιμώμενο traffic που θα πρέπει να υποστηρίζει αυτό το website. Θεωρήστε το κάτι σαν τεστ πραγματικού κόσμου.

Το πρώτο βήμα είναι να καθορίσετε τους μέγιστους χρόνους αιτήσεων που θέλετε να βιώνουν οι χρήστες, το οποίο γίνεται από την επειξηρηματική σκοπιά και την σκοπιά ευκολίας χρήσης, όχι από την τεχνική σκοπιά. Σ' αυτό το σημείο χρειάζεστε να υπολογίσετε τον αντίκτυπο που θα έχει ένα αργό website στις πωλήσεις της εταιρίας σας και στο κόστος υποστήριξης. Ένας καλός κανόνας είναι ότι κανένας επισκέπτης του site δεν πρέπει να περιμένει παραπάνω από δέκα (10) δευτερόλεπτα για να φορτώσει η σελίδα.

Έπειτα πρέπει να υπολογίσετε το προβλεφθέν φορτίο για το website σας, το οποίο αργότερα θα προσομοιώσετε χρησιμοποιώντας το Webserver Stress Tool.

Στο τέλος της λειτουργίας του load test, συγκρίνετε τα αποτελέσματα του τεστ με τους χρόνους αιτήσεων που θέλατε να πετύχετε. Όταν κάποιες αιτήσεις σελίδας παίρνουν περισσότερο χρόνο από τον επιθυμητό, τότε απαιτείται πολύ ώρα για να φορτώσουν αυτές ή μπορεί ακόμη και να αποτυγχάνουν να φορτώσουν, οπότε και παράγουν μηνύματα σφάλματος. Είναι ξεκάθαρο ότι πρέπει να γίνει δουλειά στην εφαρμογή και στο webserver.

### **Stress Tests**

Τα stress tests είναι προσομοιωμένες 'βίαιες επιθέσεις' που εφαρμόζουν υπερβολικό φορτίο στον webserver σας. Στον πραγματικό κόσμο, καταστάσεις όπως αυτή μπορούν να δημιουργηθούν από μία μαζική προσέλευση χρηστών (π.χ. αιχμή σε γράφημα)– πάνω από το κανονικό. Ένα παράδειγμα θα μπορούσε να ήταν μια εκστρατεία marketing μέσω email, που αποστέλλεται σε πιθανούς πελάτες και τους ζητά να επισκεφθούν το website και να εγγραφούν για μια υπηρεσία ή να ζητήσουν πρόσθετες πληροφορίες. Μια ακούσια άρνηση υπηρεσίας στους πιθανούς αυτούς πελάτες που είναι έτοιμοι να

μάθουν περισσότερα για το προϊόν σας, θα μπορούσε τελικά να έχει αντίκτυπο στα εισοδήματα.

Ο σκοπός των stress tests είναι να εκτιμήσουν το μέγιστο φορτίο που μπορεί να υποστηρίξει ο webserver σας, όταν παρέχει τις σελίδες του site σας ή της δικτυακής εφαρμογής στον κόσμο. Το Webserver Stress Tool μπορεί να σας βοηθήσει να μάθετε αν ο server σας θα επανέλθει σε λειτουργία, αφού εξυπηρετήσει μια μεγάλη αύξηση στο traffic.

Τα **Ramp tests** είναι μια παραλλαγή των stress tests. Σ' αυτήν την περίπτωση, ο αριθμός των χρηστών αυξάνεται με την πάροδο του χρόνου από 1 χρήστη σε έναν υψηλό αριθμό από χρήστες. Βλέποντας τα γραφήματα των click times και των σφαλμάτων, μπορεί κάποιος να βρει το μέγιστο φορτίο που μπορεί να εξυπηρετήσει ο server στην επιθυμητή ταχύτητα ή πόσο φορτίο μπορεί να χειριστεί χωρίς σφάλματα.

---

## Χρήστες, συναλλαγές, σελίδες, hits

Υπάρχει μια διαφορά μεταξύ των χρηστών, των συναλλαγών, των page views και των hits:

- Ένας χρήστης δημιουργεί αρκετές συναλλαγές (π.χ. επίσκεψη στην κεντρική σελίδα, εύρεση προϊόντος, αγορά προϊόντος)
- Μια συναλλαγή δημιουργεί αρκετά page views (π.χ. πρόσθεση προϊόντων στο shopping cart, εισαγωγή κάρτας κλπ.)
- Ένα page view δημιουργεί αρκετά hits (π.χ. framesets, εικόνες, applets, κλπ)

---

## Γλωσσάρι

Παρακάτω παρατίθενται μερικοί όροι που χρησιμοποιούνται πολύ συχνά (και στο λογισμικό Webserver Stress Tool):

- **Click**  
Ένα προσομοιωμένο mouse click ενός χρήστη που στέλνει μια αίτηση (ένα URL από τη λίστα των URLs) στο server και που ζητάει αμέσως οποιοσδήποτε απαραίτητες ανακατευθύνσεις, frames και εικόνες (αν όλα αυτά είναι ενεργοποιημένα)
- **Request**  
Μια αίτηση HTTP που αποστέλλεται στο server, χωρίς να περιμένει απάντηση.

- **Hit**  
Μια ολοκληρωμένη αίτηση HTTP (δηλαδή, μια αίτηση που έχει σταλλεί στο server και έχει απαντηθεί).
- **Time for DNS**  
Είναι ο χρόνος που απαιτείται για να ανιχνευθεί το πεδίο ονόματος του URL, χρησιμοποιώντας τον DNS server του συστήματος client.
- **Time to connect**  
Είναι ο χρόνος που απαιτείται για την εγκατάσταση της σύνδεσης με το server.
- **Time to first byte (TFB)**  
Είναι ο χρόνος μεταξύ της στιγμής που αποστέλλεται η αίτηση και της στιγμής που παραλαμβάνεται το πρώτο byte των δεδομένων, προερχόμενο από το server.
- **Click Time**  
Είναι ο χρόνος που έπρεπε να περιμένει ο χρήστης μέχρι να τελειώσει το 'click' του (συμπεριλαμβανομένων και των ανακατευθύνσεων/frames/εικόνων κλπ.).
- **User Bandwidth**  
Το εύρος ζώνης που είναι σε θέση να πετύχει ο χρήστης.
- **Sent Requests**  
Ο αριθμός των αιτήσεων που έχουν σταλλεί στο server κατά τη διάρκεια μιας περιόδου.
- **Received Requests**  
Ο αριθμός των απαντήσεων που έχουν ληφθεί από τον server κατά τη διάρκεια μιας περιόδου.

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 2

**ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΛΕΙΤΟΥΡΓΙΑΣ ΕΝΟΣ SERVER**

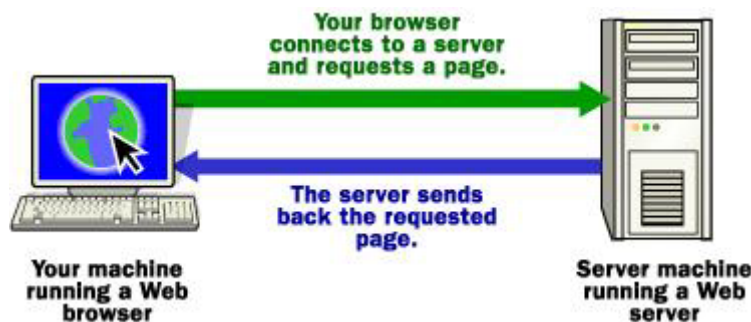
Προκειμένου να μελετήσουμε θέματα βελτίωσης της απόδοσης ενός server, θα πρέπει να εντρυφήσουμε στην αρχιτεκτονική και τις βασικές λειτουργίες του server. Στο κεφάλαιο αυτό περιγράφονται τα βασικά χαρακτηριστικά της λειτουργίας ενός server, καθώς και κάποιοι βασικοί όροι και στοιχεία του Internet, η κατανόηση των οποίων είναι σημαντική για τη συνέχεια. Το κεφάλαιο απευθύνεται σε αρχάριους, οπότε οι πιο έμπειροι μπορούν να παραβλέψουν την ανάγνωσή του.

---

## 📖 Βασική Λειτουργία

Ας πούμε ότι κάθεστε μπροστά σε έναν υπολογιστή και σερφάρετε στο διαδίκτυο, όταν κάποιος φίλος σας σας τηλεφωνεί και σας λέει ότι μόλις διάβασε ένα υπέροχο άρθρο το οποίο βρίσκεται στο site “*http://www.someserver.com/somepage.htm*”. Εσείς λοιπόν πληκτρολογείτε το URL αυτό στον browser σας και αμέσως εμφανίζεται η σελίδα στην οθόνη σας, ανεξάρτητα από το πόσο μακριά βρίσκεται η σελίδα αυτή.

Στο πιο βασικό επίπεδο, το παρακάτω διάγραμμα δείχνει τα βήματα που φέρνουν τη σελίδα στην οθόνη σας:



Ο browser σας δημιούργησε μία σύνδεση με τον απομακρυσμένο server, έκανε αίτηση για το συγκεκριμένο site και το έλαβε.

---

## 📖 Στο Παρασκήνιο

Τα βασικά βήματα που πραγματοποιούνται στο παρασκήνιο είναι:

- Ο browser σπάει το URL σε 3 μέρη:
  1. Το πρωτόκολλο (*http*)
  2. Το όνομα του server (*www.someserver.com*)
  3. Το όνομα του αρχείου (*somepage.htm*)

- Ο browser επικοινωνεί με έναν server ονομάτων για να μεταφράσει το όνομα του server *www.someserver.com* σε μία διεύθυνση IP, την οποία θα χρησιμοποιήσει για να συνδεθεί με τη μηχανή του server.
- Ο browser δημιουργεί μία σύνδεση με το server χρησιμοποιώντας αυτήν τη διεύθυνση IP στο port 80 (θα συζητήσουμε για τα ports παρακάτω).
- Ακολουθώντας το πρωτόκολλο HTTP, ο browser στέλνει μία αίτηση GET στο server, ζητώντας το αρχείο *http://www.someserver.com/somepage.htm* (μπορούν επίσης να σταλούν και cookies από τον browser στο server μαζί με την αίτηση GET – για τα cookies θα μιλήσουμε παρακάτω).
- Ο server μετά στέλνει το HTML text για την web page στο browser (μπορούν επίσης να σταλούν και cookies από το server στο browser μέσα στην επικεφαλίδα της σελίδας).
- Ο browser τέλος διαβάζει τα HTML tags και μορφοποιεί τη σελίδα στην οθόνη.

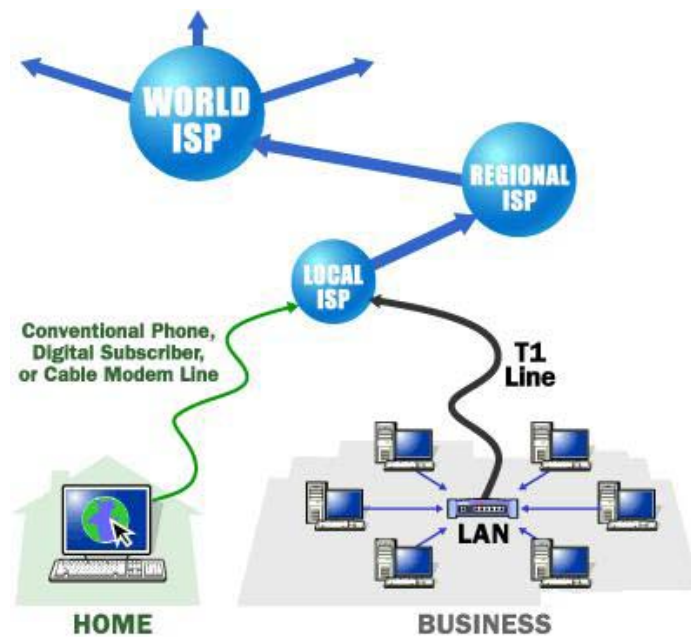
---

## To Internet

Το Internet είναι μία γιαγαντιαία συλλογή από εκατομμύρια υπολογιστές συνδεδεμένους σε ένα δίκτυο υπολογιστών (**computer network**). Το δίκτυο επιτρέπει σε όλους τους υπολογιστές να επικοινωνούν μεταξύ τους. Ένας home υπολογιστής μπορεί να είναι συνδεδεμένος στο Internet χρησιμοποιώντας *dial-up modem*, *DSL* ή *καλωδιακό modem* για να επικοινωνήσει με έναν Παροχέα Υπηρεσιών Ίντερνετ (**ISP**). Ένας υπολογιστής σε μία υπηρεσία ή πανεπιστήμιο συνήθως διαθέτει μία network interface card (**NIC**), η οποία τον συνδέει απευθείας σε ένα δίκτυο τοπικής περιοχής (**LAN**) μέσα στην υπηρεσία. Η υπηρεσία μπορεί να συνδέσει το LAN της σε έναν ISP μέσω μιας γραμμής τηλεφώνου υψηλής ταχύτητας, όπως η **T1**. Μία γραμμή T1 μπορεί να διαχειριστεί περίπου 1.5 εκατομμύριο bits το δευτερόλεπτο (**bps**), ενώ μια απλή γραμμή τηλεφώνου μπορεί να διαχειριστεί 30.000 με 50.000 bps.

Οι ISPs συνδέονται σε μεγαλύτερους ISPs και οι μεγαλύτεροι ISPs διατηρούν “backbones” οπτικών ινών για ένα ολόκληρο έθνος ή περιοχή. Backbones σε όλο τον κόσμο συνδέονται μεταξύ τους μέσω γραμμών οπτικών ινών, υποβρύχιων καλωδίων ή δορυφορικών συνδέσεων. Με αυτόν τον τρόπο, κάθε υπολογιστής στο Internet είναι συνδεδεμένος με οποιονδήποτε άλλον υπολογιστή στο Internet.





## Clients και Servers

Γενικά, όλα τα μηχανήματα στο Internet μπορούν να κατηγοριοποιηθούν σε δύο τύπους: **servers** (εξυπηρέτες) και **clients** (εξυπηρετούμενους). Αυτές οι μηχανές που παρέχουν υπηρεσίες (όπως οι Web servers ή οι FTP servers) σε άλλες είναι οι servers. Οι μηχανές που χρησιμοποιούνται για να συνδέονται σε αυτές τις υπηρεσίες είναι οι clients. Όταν συνδέεστε στο Yahoo! στο site *www.yahoo.com* για να διαβάσετε μία σελίδα, το Yahoo! παρέχει μια μηχανή (πιθανώς ένα σύμπλεγμα από πολύ μεγάλα μηχανήματα που ονομάζεται **cluster**) ώστε να εξυπηρετήσει το αίτημά σας. Το Yahoo! παρέχει έναν server. Το μηχανήμά σας από την άλλη μεριά, πιθανόν δεν παρέχει καμία υπηρεσία σε κανέναν άλλο στο Internet. Γι' αυτόν το λόγο, αποτελεί μία μηχανή χρήστη, επίσης γνωστή και ως client. Είναι δυνατόν και σύνηθες για μία μηχανή να είναι και server και client ταυτόχρονα, αλλά για το δικό μας σκοπό εδώ θα θεωρούμε τις μηχανές είτε του ενός είτε του άλλου τύπου.

Ένας server μπορεί να παρέχει μία ή περισσότερες υπηρεσίες στο Internet. Για παράδειγμα, μία μηχανή server μπορεί να έχει λογισμικό που να τρέχει σ' αυτήν ώστε να τις επιτρέπεται να συμπεριφέρεται ως Web server, e-mail server ή FTP server. Οι clients απευθύνονται σε έναν server έχοντας συγκεκριμένη πρόθεση, κατευθύνοντας έτσι τις αιτήσεις τους σε συγκεκριμένο server λογισμικού που τρέχει στο συνολικό μηχανήμα server.

---

## Διευθύνσεις IP

Σε κάθε μηχανήμα στο Internet ανατίθεται μία μοναδική διεύθυνση που λέγεται **IP διεύθυνση**. Το IP προέρχεται από το Internet Protocol. Οι διευθύνσεις αυτές είναι 32-bit αριθμοί, εκφραζόμενοι συνήθως ως τέσσερις 8-bit αριθμοί που δημιουργούν έναν δεκαδικό χωριζόμενο με τελείες. Μία τυπική IP διεύθυνση είναι η ακόλουθη:

216.27.61.137

Οι τέσσερις διαχωριζόμενοι από τελείες αριθμοί έχουν τιμή από 0-255, που είναι οι  $2^8$  πιθανοί συνδυασμοί των 8-bit αριθμών.

Ένας server έχει μια στατική IP διεύθυνση που δεν αλλάζει πολύ συχνά. Ένας home υπολογιστής ο οποίος συνδέεται στο Internet μέσω dial-up modem, συχνά έχει μια IP διεύθυνση που του ανατίθεται από τον ISP όταν το μηχανήμα συνδέεται. Η διεύθυνση αυτή είναι μοναδική για τη συγκεκριμένη περίοδο – μπορεί να είναι διαφορετική την επόμενη φορά που θα συνδεθεί ο υπολογιστής. Με αυτόν τον τρόπο, ο ISP χρειάζεται μία IP διεύθυνση για κάθε modem που υποστηρίζει, παρά μία διεύθυνση για κάθε πελάτη.

Εάν χρησιμοποιείτε λειτουργικό Windows στον υπολογιστή σας, μπορείτε να δείτε διάφορες πληροφορίες Internet για το μηχανήμα σας, συμπεριλαμβανομένης και της IP διεύθυνσης και του hostname, με την εντολή IPCONFIG.EXE (Windows 2000/XP).

Όσον αφορά στα μηχανήματα Internet, το μόνο που χρειάζεται για να επικοινωνήσουν με τον server είναι η IP διεύθυνση. Για παράδειγμα, μπορείτε να πληκρολογήσετε στον browser το URL *http://209.116.69.66* και να φτάσει στην οθόνη σας το περιεχόμενο του Web server που έχει αυτήν την IP διεύθυνση. Σε κάποιους servers, η IP διεύθυνση από μόνη της δεν είναι αρκετή, αλλά στους περισσότερους μεγάλους servers είναι.

---

## Ονόματα Πεδίου (Domain names)

Επειδή οι περισσότεροι δυσκολεύονται να θυμούνται τις ακολουθίες των αριθμών που σχηματίζουν τις IP διευθύνσεις κι επειδή μερικές φορές οι IP διευθύνσεις χρειάζεται να αλλάζουν, όλοι οι servers στο Internet έχουν επίσης και πραγματικά ονόματα που μπορούν να διαβαστούν, τα οποία ονομάζονται **domain names**. Για παράδειγμα το *www.howstuffworks.com* είναι ένα σταθερό, αναγνώσιμο όνομα. Είναι πιο εύκολο να το θυμόμαστε έτσι από το να θυμόμαστε την IP διεύθυνση 209.116.69.66.

Το όνομα *www.howstuffworks.com* έχει τρία μέρη:

1. Το όνομα host (“www”)
2. Το domain name (“howstuffworks”)
3. Το top-level domain name (“com”)

Το όνομα host δημιουργείται από την εταιρία που φιλοξενεί το πεδίο. Το “www” είναι ένα πολύ κοινό host name, αλλά πολλοί δικτυακοί τόποι είτε το παραλείπουν είτε το αντικαθιστούν με ένα διαφορετικό όνομα που υποδεικνύει μία συγκεκριμένη περιοχή του site. Για παράδειγμα, στο *encarta.msn.com*, το domain name για την Microsoft’s Encarta encyclopedia είναι το “Encarta” και χρησιμοποιείται για host name αντί για το “www”.

---

## Εξυπηρέτες Ονομάτων (Name Servers)

Ένα σύνολο από servers, που καλείται domain name servers (**DNS**), αντιστοιχίζει τα αναγνώσιμα ονόματα των servers σε IP διευθύνσεις. Αυτοί οι servers είναι απλές βάσεις δεδομένων που αντιστοιχίζουν ονόματα σε IP διευθύνσεις και είναι διαμοιρασμένοι σε όλο το Internet. Οι περισσότερες εταιρίες, ISPs και πανεπιστήμια διατηρούν μικρούς name servers για να αντιστοιχίζουν host names σε IP διευθύνσεις. Υπάρχουν επίσης κεντρικοί servers για να αντιστοιχίζουν domain names σε IP διευθύνσεις.

Αν πληκτρολογήσετε το URL *http://computer.howstuffworks.com/web-server.htm* στον browser, ο browser εξάγει το όνομα *www.howstuffworks.com*, το περνάει σε έναν domain name server και ο server αυτός επιστρέφει την κατάλληλη IP διεύθυνση. Ένας αριθμός από servers μπορούν να εμπλακούν προκειμένου να πάρουμε τη σωστή IP διεύθυνση. Για παράδειγμα, στην περίπτωση του *www.howstuffworks.com*, ο name server για το top-level domain “com” θα ξέρει την IP διεύθυνση για το name server που γνωρίζει τα host names και, ένα ξεχωριστό ερώτημα σε αυτόν το name server, τον οποίο διαχειρίζεται ο HowStuffWorks ISP, θα παραδώσει την IP διεύθυνση για τη μηχανή του HowStuffWorks server.

---

## Θύρες (Ports)

Κάθε μηχανή server κάνει τις υπηρεσίες του διαθέσιμες στο Internet χρησιμοποιώντας αριθμημένα ports, ένα για κάθε υπηρεσία που είναι διαθέσιμη στον server. Για παράδειγμα, εάν μία μηχανή server τρέχει έναν Web server κι έναν FTP server, ο Web server τυπικά είναι διαθέσιμος στο port 80 και ο FTP server στο port 21. Οι clients συνδέονται με μια υπηρεσία σε συγκεκριμένη IP διεύθυνση και σε συγκεκριμένο port.

Κάθε μία από τις πλέον γνωστές υπηρεσίες είναι διαθέσιμη σε ένα γνωστό port. Ακολούθως παρατίθενται μερικοί κοινοί αριθμοί port:

- echo 7
- daytime 13
- qotd 17 (Quote of the day)
- ftp 21
- telnet 23
- smtp 25 (Simple Mail Transfer, δηλαδή το e-mail)
- time 37
- nameserver 42
- nickname 43 (Who Is)
- gopher 70
- finger 79
- WWW 80

Αν ο server δεχτεί συνδέσεις σε ένα port από τον έξω κόσμο και αν το port δεν το προστατεύει firewall (δες παρακάτω), μπορείτε να συνδεθείτε στο port από οπουδήποτε στο Internet και να χρησιμοποιήσετε την υπηρεσία. Σημειώστε ότι τίποτε δεν επιβάλλει, για παράδειγμα στον Web server, να βρίσκεται στο port 80. Αν επρόκειτο να εγκαταστήσετε το δικό σας μηχάνημα και να φορτώσετε λογισμικό Web server επάνω, θα μπορούσατε να τοποθετήσετε τον Web server στο port 918, ή σε οποιοδήποτε άλλο μη χρησιμοποιούμενο port. Έπειτα, αν το μηχάνημά σας είναι γνωστό ως xxx.yyy.com, κάποιος στο Internet θα μπορούσε να συνδεθεί στον server σας με το URL *http://xxx.yyy.com:918*. Το “:918” καθορίζει άμεσα τον αριθμό του port και θα πρέπει να συμπεριληφθεί οπωσδήποτε από κάποιον που θέλει να επικοινωνήσει με το server σας. Όταν δεν καθορίζεται κάποιο port στο URL, ο browser απλά υποθέτει ότι ο server χρησιμοποιεί το γνωστό port 80.

---

## Ασφάλεια

Από τα όσα έχουμε πει, ο Web server μπορεί να είναι ένα πολύ μικρό κομμάτι λογισμικού. Λαμβάνει το όνομα του αρχείου μέσω μιας εντολής GET, ανακτά το αρχείο αυτό και το αποστέλλει μέσω σύνδεσης στον browser που έκανε την αίτηση. Λαμβάνοντας υπόψη και τον κώδικα για τη διαχείριση των ports και των συνδέσεών τους, θα μπορούσαμε να δημιουργήσουμε εύκολα ένα πρόγραμμα σε C που να υλοποιεί έναν απλό Web server σε λιγότερες από 500 γραμμές κώδικα. Προφανώς υπάρχουν και πολύ πιο πολύπλοκοι Web servers αλλά, σε γενικές γραμμές, τα βασικά είναι πολύ απλά.

Οι περισσότεροι servers προσθέτουν ένα επίπεδο ασφάλειας στη διαδικασία εξυπηρέτησης. Για παράδειγμα, εάν ποτέ επισκεφτήκατε μια σελίδα και ο browser άνοιξε ένα παράθυρο διαλόγου ζητώντας username και password, τότε είχατε να κάνετε με σελίδα προστατευόμενη από κωδικό πρόσβασης. Ο server επιτρέπει στον ιδιοκτήτη της σελίδας να διατηρεί μία λίστα από ονόματα και κωδικούς πρόσβασης για τα άτομα στα οποία επιτρέπεται να έχουν πρόσβαση στη σελίδα. Ο server αφήνει μόνο όσους γνωρίζουν τον κατάλληλο κωδικό πρόσβασης. Οι πιο προχωρημένοι χρήστες προσθέτουν περισσότερη ασφάλεια ώστε να επιτρέπουν κρυπτογραφημένη σύνδεση μεταξύ server και browser, έτσι ώστε σημαντικές πληροφορίες (όπως αριθμοί πιστωτικών καρτών) να μπορούν να αποστέλλονται μέσω Internet με ασφάλεια.

---

## Δυναμικές σελίδες (Dynamic pages)

Όμως τι γίνεται με τις σελίδες που είναι δυναμικές; Για παράδειγμα:

- Οποιοδήποτε βιβλίο φιλοξενουμένων (guest book) που σας επιτρέπει να εισάγετε ένα μήνυμα σε μία HTML φόρμα και την επόμενη φορά που διαβάζετε το βιβλίο, η σελίδα περιέχει τη νέα καταχώρηση.
- Η φόρμα *whois* στις *Network Solutions*, που επιτρέπει να εισάγετε το domain name στη φόρμα και η σελίδα που επιστρέφεται είναι διαφορετική ανάλογα με το domain name που εισήχθει.
- Οποιαδήποτε μηχανή αναζήτησης που σας επιτρέπει να εισάγετε λέξεις κλειδιά σε μία HTML φόρμα κι έπειτα δημιουργεί δυναμικά μία σελίδα βασισμένη στις λέξεις κλειδιά που εισάγατε.

Σε όλες αυτές τις περιπτώσεις, ο Web server δεν αναζητεί απλά ένα αρχείο. Ακριβέστερα, επεξεργάζεται πληροφορίες και δημιουργεί μια σελίδα βασισμένη σε προδιαγραφές ενός αιτήματος. Σχεδόν σε όλες τις περιπτώσεις, ο Web server χρησιμοποιεί τα λεγόμενα CGI scripts για να ολοκληρώσει την αίτηση. Τα CGI scripts θα τα δούμε παρακάτω.

---

## Cookies

Τα cookies δεν είναι προγράμματα και δεν τρέχουν όπως τα προγράμματα. Γι' αυτόν το λόγο, δεν μπορούν να συλλέξουν οποιαδήποτε πληροφορία από μόνα τους, ούτε μπορούν να συλλέξουν καμία προσωπική πληροφορία για σας και το μηχάνημά σας.

Ένα cookie είναι ένα κομμάτι κειμένου το οποίο μπορεί να αποθηκεύσει ένας Web server στο σκληρό δίσκο ενός χρήστη. Τα cookies επιτρέπουν σε μία σελίδα Web να αποθηκεύει πληροφορίες στο μηχάνημα ενός χρήστη ώστε να τις ανακτά αργότερα. Τα κομμάτια πληροφορίας αποθηκεύονται ως ζεύγη ονόματος-τιμής.

Για παράδειγμα, ένα Web site μπορεί να παράγει ένα μοναδικό αριθμό ID για κάθε επισκέπτη και να αποθηκεύει τον αριθμό ID στο μηχάνημα κάθε χρήστη χρησιμοποιώντας ένα αρχείο cookie.

Αν χρησιμοποιείτε τον Internet Explorer της Microsoft για να surfάρετε στο Web, μπορείτε να δείτε όλα τα cookies που είναι αποθηκευμένα στο μηχάνημά σας. Το πιο κοινό μέρος που μπορούν να βρίσκονται αυτά είναι ο κατάλογος *c:\windows\cookies*. Κάθε αρχείο είναι ένα αρχείο κειμένου που περιέχει ζεύγη ονόματος-τιμής και υπάρχει ένα αρχείο για κάθε Web site που έχει τοποθετήσει cookies στο μηχάνημά σας.

Μπορείτε να δείτε μέσα στον κατάλογο ότι κάθε ένα από τα αρχεία αυτά είναι ένα απλό, κανονικό αρχείο κειμένου. Μπορείτε να δείτε ποιο Web site έχει τοποθετήσει το αρχείο στον υπολογιστή σας κοιτώντας το όνομα του αρχείου (η πληροφορία αυτή είναι επίσης αποθηκευμένη και μέσα στο αρχείο).

Για παράδειγμα, έχω επισκεφθεί τη σελίδα *goto.com* και η σελίδα αυτή έχει τοποθετήσει ένα cookie στο μηχάνημά μου. Το αρχείο cookie για τη σελίδα αυτή περιέχει την εξής πληροφορία:

*UserID A9A3BECE0563982D www.goto.com/*

Η *goto.com* έχει αποθηκεύσει στον υπολογιστή μου ένα μονό ζεύγος ονόματος-τιμής. Το όνομα του ζεύγους είναι *UserID* και η τιμή είναι *A9A3BECE0563982D*. Την πρώτη φορά που επισκέφθηκα την *goto.com*, το site μου ανέθεσε ένα μοναδικό ID και το αποθήκευσε στο μηχάνημά μου.

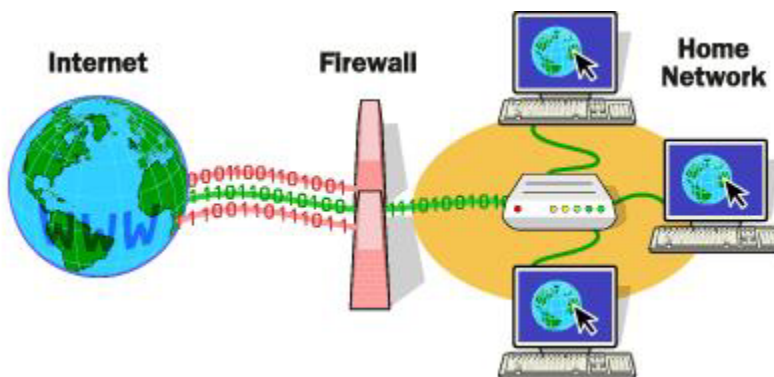
Η πλειοψηφία των sites αποθηκεύουν ένα μόνο κομμάτι πληροφορίας στον υπολογιστή σας – ένα ID χρήστη. Όμως, μια σελίδα μπορεί να αποθηκεύσει και περισσότερα ζεύγη ονόματος-τιμής, αν αυτή το επιθυμεί.

Ένα ζεύγος ονόματος-τιμής είναι απλά ένα κομμάτι δεδομένων με όνομα. Δεν είναι πρόγραμμα και δεν μπορεί να κάνει από μόνο του τίποτα. Ένα Web site μπορεί να ανακτήσει μόνο την πληροφορία που έχει αποθηκεύσει αυτό στον υπολογιστή σας. Δεν μπορεί να ανακτήσει πληροφορία από άλλα cookies, ούτε οποιαδήποτε άλλη πληροφορία από τον υπολογιστή σας.

## Firewalls

Αν χρησιμοποιείτε το Internet για κάποιο διάστημα και ειδικότερα αν δουλεύετε σε μια μεγαλύτερη εταιρία και σερφάρετε στο Web όταν είστε στη δουλειά, τότε πιθανόν να έχετε ακούσει τον όρο **firewall**. Για παράδειγμα, θα ακούτε συχνά άτομα σε εταιρίες να λένε «Δεν μπορώ να χρησιμοποιήσω αυτό το site γιατί δεν θα το αφήσουν να περάσει από το firewall».

Αν έχετε μια γρήγορη σύνδεση Internet στο σπίτι σας (DSL ή καλωδιακό modem), θα έχετε ακούσει σχετικά με τα firewalls για το δίκτυο του σπιτιού σας. Φαίνεται ότι ένα μικρό δίκτυο σπιτιού έχει πολλά κοινά θέματα ασφάλειας με ένα δίκτυο μιας εταιρίας. Μπορείτε να χρησιμοποιήσετε ένα firewall για να προστατέψετε το δίκτυο του σπιτιού σας από επικίνδυνες σελίδες και επίδοξους hackers.



Βασικά, ένα firewall είναι μια ασπίδα για την προστασία της ιδιοκτησίας σας από καταστροφικές επιθέσεις. Για την ακρίβεια, αυτός είναι ο λόγος που ονομάζεται firewall. Η δουλειά του είναι παρόμοια με μία φυσική ασπίδα φωτιάς που αποτρέπει την εξάπλωση της από την μία περιοχή στην άλλη.

Ένα firewall είναι απλά ένα πρόγραμμα ή υλική συσκευή που φιλτράρει την πληροφορία που έρχεται μέσω μιας Internet σύνδεσης στο προσωπικό σας δίκτυο ή υπολογιστικό σύστημα. Αν ένα εισερχόμενο πακέτο είναι μαρκαρισμένο από τα φίλτρα, τότε αυτό μπλοκάρεται.

Ας πούμε ότι εργάζεστε σε μια εταιρία με 500 υπαλλήλους. Η εταιρία για το λόγο αυτόν έχει εκατοντάδες υπολογιστές, οι οποίοι διαθέτουν κάρτες δικτύου ώστε να μπορούν να συνδέονται μεταξύ τους. Επίσης, η εταιρία έχει μία ή περισσότερες συνδέσεις στο Internet μέσω γραμμών όπως η T1 ή η T3. Χωρίς εγκατεστημένο firewall, όλοι αυτοί οι υπολογιστές είναι άμεσα προσβάσιμοι από οποιονδήποτε στο Internet. Ένα άτομο που ξέρει τι κάνει, μπορεί να εξετάσει εκτεταμένα αυτούς τους υπολογιστές, να προσπαθήσει να δημιουργήσει FTP συνδέσεις σε αυτούς, να προσπαθήσει να

δημιουργήσει Telnet συνδέσεις κ.ο.κ.. Εάν ένας υπάλληλος κάνει λάθος και αφήσει μία τρύπα ασφαλείας, οι hackers μπορούν να εκμεταλλευτούν την τρύπα αυτή.

Με ένα firewall εγκατεστημένο, το τοπίο είναι πολύ διαφορετικό. Μια εταιρία μπορεί να εγκαταστήσει ένα firewall σε κάθε σύνδεση στο Internet (π.χ. σε κάθε γραμμή T1 που φτάνει στην εταιρία). Το firewall μπορεί να υλοποιήσει κανόνες ασφαλείας. Για παράδειγμα, ένας από τους κανόνες ασφαλείας μέσα στην εταιρία μπορεί να είναι:

*Από τους 500 υπολογιστές μέσα στην εταιρία, μόνο ένας από αυτούς εξουσιοδοτείται να δέχεται δημόσια FTP κίνηση (traffic). Επιτρέπονται FTP συνδέσεις μόνο σε αυτόν τον υπολογιστή και σε κανέναν άλλον.*

Μια εταιρία μπορεί να θεσπίσει τέτοιους κανόνες για FTP servers, Web servers, Telnet servers κ.ο.κ.. Επιπλέον, η εταιρία μπορεί να ελέγχει το πώς συνδέονται οι υπάλληλοι στα Web sites, αν επιτρέπεται να διανέμονται αρχεία μέσα στο δίκτυο κ.ο.κ.. Το firewall δίνει στην εταιρία τεράστιο έλεγχο στον τρόπο που τα άτομα χρησιμοποιούν το δίκτυο.

---

## CGI Scripts

Στους περισσότερους Web servers, ο μηχανισμός CGI έχει καθιερωθεί με τον ακόλουθο τρόπο. Στον root κατάλογο του server (δηλαδή στον κατάλογο που βρίσκονται τα αρχεία του server τα οποία μπορούν να διατεθούν όταν ο χρήστης συνδέεται με αυτόν), δημιουργούμε έναν υποκατάλογο με το όνομα **cgi-bin**. Ο server καταλαβαίνει ότι όποιο αρχείο ζητηθεί από αυτόν τον κατάλογο δεν πρέπει απλά να διαβαστεί και να αποσταλλεί, αλλά αντί αυτού να **εκτελεστεί**. Η έξοδος του εκτελεσμένου προγράμματος είναι ό,τι ακριβώς αποστέλλεται στον browser που αναζήτησε τη σελίδα. Το εκτελέσιμο είναι γενικά είτε καθαρό εκτελέσιμο, όπως η έξοδος από έναν C compiler, είτε είναι ένα script γραμμένο σε Perl.

Φανταστείτε ότι πληκτρολογείτε το ακόλουθο URL στον browser σας: `http://computer.howstuffworks.com.cgi-bin/search.pl`. Ο server αναγνωρίζει ότι το `search.pl` είναι μέσα στον κατάλογο `cgi-bin`, οπότε εκτελεί το `search.pl` (το οποίο είναι ένα Pearl script) και στέλνει την έξοδό του στον browser.

Η πιο απλή HTML Web σελίδα είναι η ακόλουθη:

```
<html>
  <body>
    <h1>Hello there!</h1>
  </body>
</html>
```



Το πιο απλό CGI script που κατά την εκτέλεσή του θα δημιουργούσε αυτήν την απλή, στατική σελίδα και το οποίο θα μπορούσε να γραφτεί σε γλώσσα C είναι το εξής:

```
#include <stdio.h>

int main()
{
    printf("Content-type: text/html\n\n");
    printf("<html>\n");
    printf("<body>\n");
    printf("<h1>Hello there!</h1>\n");
    printf("</body>\n");
    printf("</html>\n");
    return 0;
}
```

Αφού γράψουμε αυτό το πρόγραμμα, το τοποθετούμε σε ένα αρχείο `simplest.c` κι έπειτα το μεταφράζουμε ως εξής:

```
gcc simplest.c -o simplest.cgi
```

Τοποθετώντας το `simplest.cgi` στον κατάλογο `cgi-bin`, αυτό μπορεί να εκτελεστεί. Εάν από τον browser πληκτρολογήσουμε το κατάλληλο URL ζητώντας αυτό το αρχείο από το server που το κρατά, τότε παράγεται μια σελίδα που απλά γράφει “Hello there!”. Το μοναδικό σημείο που μπορεί να σας φαίνεται περίεργο είναι η γραμμή:

```
printf("Content-type: text/html\n\n");
```

Η γραμμή “Content-type: text/html\n\n” είναι ένα ειδικό κομμάτι κειμένου που θα πρέπει να είναι το πρώτο που θα αποσταλλεί στον browser από ένα CGI script. Εάν το ξεχάσετε, ο browser θα απορρίψει την έξοδο του script.

Αυτά όσον αφορά στις στατικές Web σελίδες. Για τις δυναμικές σελίδες χρησιμοποιούνται πιο πολύπλοκα scripts στα οποία πλέον περνάμε και ορίσματα. Δεν χρειάζεται να επεκταθούμε σε αυτά τα θέματα. Σκοπός αυτής της ενότητας είναι η κατανόηση της βασικής φιλοσοφίας των CGI scripts.

---

## ASP (Active Server Pages)

Οι ASP σελίδες είναι συνηθισμένα HTML αρχεία που έχουν επεκταθεί με πρόσθετα χαρακτηριστικά. Όπως ένα HTML αρχείο, μια ASP σελίδα μπορεί να περιέχει HTML tags που μπορούν να διερμηνευθούν και να εμφανιστούν από έναν web browser. Οτιδήποτε θα μπορούσατε κανονικά να τοποθετήσετε σε ένα HTML αρχείο (όπως Java applets, blinking text, client side scripts, client side ActiveX controls κλπ.), μπορείτε να το τοποθετήσετε και σε μια ASP σελίδα.

Παρακάτω παρατίθεται μια λίστα από τα νέα χαρακτηριστικά που μπορείτε να υλοποιήσετε χρησιμοποιώντας ASP:

1. Τοποθέτηση κυλιόμενου μηνύματος σε web pages
2. Ανάκτηση πληροφορίας που εισάγεται σε μία φόρμα HTML και αποθήκευση της πληροφορίας αυτής σε μια βάση δεδομένων
3. Δημιουργία δυναμικών σελίδων που εμφανίζουν διαφορετικό περιεχόμενο σε διαφορετικούς χρήστες
4. Πρόσθεση μετρητών επιτυχιών (hit counters) σε μία ή περισσότερες web pages
5. Εμφάνιση διαφορετικών σελίδων, ανάλογα με τις δυνατότητες του browser του χρήστη
6. Σύνδεση πολλαπλών web pages με τέτοιον τρόπο ώστε να επιτρέπεται η εύκολη πλοήγησή τους
7. Εύρεση πληροφορίας σχετικά με την δραστηριότητα του χρήστη στη σελίδα σας και καταγραφή της πληροφορίας αυτής σε αρχεία log

Ο καλύτερος τρόπος για να καταλάβετε πως λειτουργούν τα ASPs είναι να συγκρίνετε έναν web server που υποστηρίζει τα ASPs με έναν web server που δεν τα υποστηρίζει. Η Microsoft παρουσίασε τα ASPs με την τρίτη έκδοση του IIS (θα τον μελετήσουμε διεξοδικά παρακάτω). Η παρουσίαση των ASPs μεταμόρφωσε των IIS από server χαμηλών δυνατοτήτων στατικού περιεχομένου, σε server δυναμικού περιεχομένου. Η λειτουργία ενός ASP αρχείου είναι η εξής:

- Ένας χρήστης εισάγει μία διεύθυνση Internet ενός ASP στον browser
- Ο browser στέλνει μια αίτηση για μια σελίδα ASP στον IIS
- Ο web server λαμβάνει την αίτηση και καταλαβαίνει ότι πρόκειται για αίτηση ASP λόγω της κατάληξης “.asp”
- Ο web server ανακτά το κατάλληλο αρχείο ASP από το δίσκο ή τη μνήμη
- Ο web server στέλνει το αρχείο σε ένα ειδικό πρόγραμμα που ονομάζεται “ASP.dll”
- Το αρχείο ASP υποβάλλεται σε επεξεργασία από πάνω προς τα κάτω και εκτελούνται όλες οι εντολές. Το αποτέλεσμα αυτής της επεξεργασίας είναι ένα συνηθισμένο HTML αρχείο
- Το HTML αρχείο αποστέλλεται πίσω στον browser
- Το HTML αρχείο μεταφράζεται από τον Web browser του ατόμου που έκανε την αίτηση και τα αποτελέσματα εμφανίζονται στο παράθυρο του browser

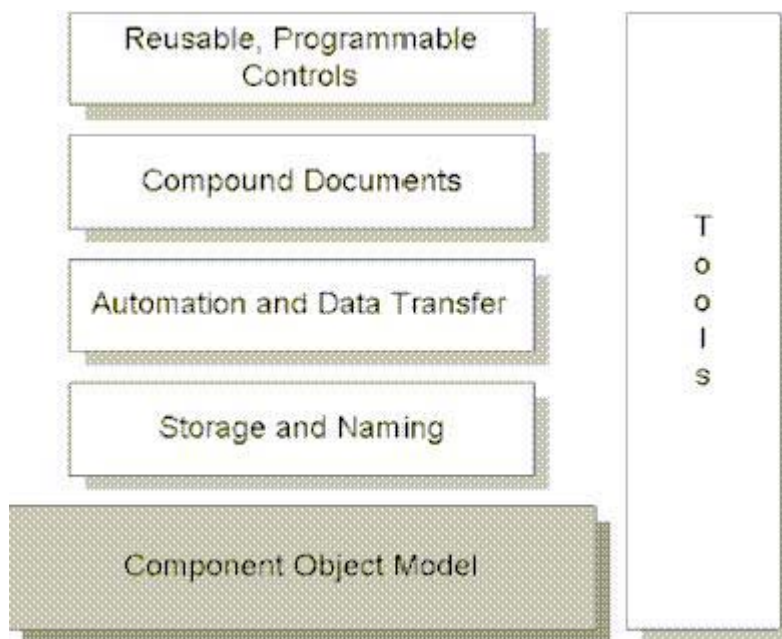
Υπάρχουν τρεις κύριοι λόγοι για να προτιμήσετε τα ASPs από τα CGI scripts. Απόδοση – Με τα CGI, κάθε script τρέχει ως ένα εκτελέσιμο μέσα στη δικιά του διεργασία. Ως αποτέλεσμα κάθε φορά που ζητείται ένα script ο server πρέπει: να δημιουργήσει μια νέα διεργασία, να τρέξει το script, να τερματίσει αυτή τη διεργασία που δημιούργησε. Αυτό

δεν είναι αποδοτικό και μπορεί να έχει αρκετό αντίκτυπο στην απόδοση του Web server. Ο ASP host δεν επανεκκινείται με κάθε πρόσβαση ενός script και γι' αυτόν το λόγο είναι πιο αποδοτικός. Διαχείριση συνόδου – Επειδή τα HTMLs δεν έχουν κατάσταση, το να παρακολουθούνται τα δεδομένα που κυκλοφορούν ανάμεσα στις προσβάσεις σελίδων και στο χρήστη είναι ένα δύσκολο πρόβλημα. Οι ASPs παρέχουν built-in session management λειτουργικότητα που επιτρέπει στους προγραμματιστές να διατηρούν δεδομένα καθώς και στιγμιότυπα COM Components (όπως συνδέσεις βάσεων δεδομένων) κατά τη διάρκεια της συνόδου. Εύκολη ολοκλήρωση των COM Components – τα ASPs είναι σχεδιασμένα ώστε να βασίζονται σε μεγάλο βαθμό στα COM Components για επεκτασιμότητα. Ως αποτέλεσμα, είναι πολύ εύκολο να χρησιμοποιηθεί ένα COM Component μέσα από ένα ASP script.

---

## COM (Component Object Model)

Το COM είναι μία αρχιτεκτονική λογισμικού συστατικών (component software architecture) που επιτρέπει σε εφαρμογές και συστήματα να δημιουργούνται από συστατικά που παρέχονται από διαφορετικούς πωλητές λογισμικού. Το COM είναι η βασική αρχιτεκτονική που διαμορφώνει τη θεμελίωση των υπηρεσιών λογισμικού υψηλότερου επιπέδου, όπως αυτών που παρέχονται από το OLE. Οι υπηρεσίες OLE καλύπτουν διάφορες όψεις του λογισμικού συστατικών (compound documents, custom controls, inter-application scripting, data transfer και άλλες διαδράσεις λογισμικού).



**Υπηρεσίες λογισμικού OLE υψηλού επιπέδου είναι δημιουργημένες πάνω στο Component Object Model**

Αυτές οι υπηρεσίες παρέχουν ευκρινώς διαφορετική λειτουργικότητα στο χρήστη. Ωστόσο, έχουν μια βασική απαίτηση σχετικά με έναν μηχανισμό που επιτρέπει δυαδικά συστατικά λογισμικού, παρεχόμενα από διαφορετικούς πωλητές λογισμικού, ώστε να συνδέονται και να επικοινωνούν το ένα με το άλλο με έναν καλά ορισμένο τρόπο. Αυτός ο μηχανισμός παρέχεται από το COM, μία αρχιτεκτονική λογισμικού συστατικών η οποία:

- Ορίζει ένα δυαδικό standard για τη διαδιεργασία των συστατικών
- Είναι ανεξάρτητη των γλωσσών προγραμματισμού
- Παρέχεται σε πολλαπλές πλατφόρμες (Microsoft® Windows®, Microsoft Windows NT™, Apple® Macintosh®, UNIX®)
- Παρέχεται για την υγιή εξέλιξη εφαρμογών και συστημάτων βασισμένων σε συστατικά
- Είναι επεκτάσιμη

Είναι σημαντικό να σημειώσουμε ότι το COM είναι μια γενική αρχιτεκτονική για λογισμικό συστατικών. Καθώς η Microsoft εφαρμόζει το COM για να απευθυνθεί σε συγκεκριμένες περιοχές όπως οι ρυθμίσεις, τα compound documents, ο αυτοματισμός, η μετάδοση δεδομένων, η αποθήκευση, η ονοματολογία και άλλα, οποιοσδήποτε προγραμματιστής μπορεί να επωφεληθεί από τη δομή και τη θεμελίωση που παρέχει το COM.

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 3

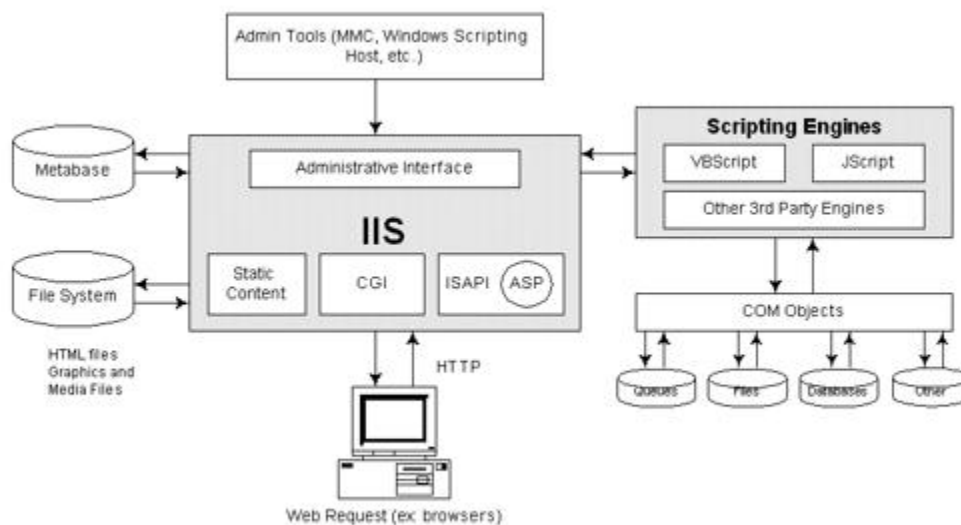
**Ο IIS 5.0 ΚΑΙ Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ**

Στις απαρχές του Web, οι εφαρμογές διαδικτύου δεν ήταν τίποτα παραπάνω από διαμοιρασμένα στατικά έγγραφα. Με την άνθιση του προγραμματισμού CGI, οι Web pages έγιναν λίγο πιο σύνθετες, προσφέροντας χαρακτηριστικά όπως μετρητές, guest books, ίσως μία μερική διάδραση βάσεων δεδομένων, ακόμη όμως δεν μπορούσαν να προσφέρουν τίποτε παραπάνω από έναν τεράστιο online κατάλογο. Το παλιό μοντέλο των εφαρμογών διαδικτύου έχει αλλάξει δραματικά τα τελευταία χρόνια, χάρη στις τεχνολογίες όπως ο IIS που επιτρέπουν τη συμπαγή ολοκλήρωση των υπηρεσιών δικτύου μέσα στο λειτουργικό σύστημα. Οι εφαρμογές διαδικτύου μπορούν τώρα να παρέχουν διάδραση με διάφορες βάσεις δεδομένων, συστήματα αρχείων και άλλες υπηρεσίες που ήταν η κυριαρχία των client-server επιτραπέζιων εφαρμογών. Στο παρόν κεφάλαιο θα μελετήσουμε την αρχιτεκτονική του IIS, του server δηλαδή που θα χρησιμοποιήσουμε σε παρακάτω κεφάλαια προκειμένου να πραγματοποιήσουμε τα πειράματα για τη μελέτη της βελτίωσης της απόδοσης του διαδικτύου.

---

## Τι είναι ο IIS;

Η ομάδα του λειτουργικού συστήματος Windows NT ανέπτυξε ένα μέρος του Microsoft IIS (**I**nternet **I**nformation **S**erver). Αυτό σημαίνει ότι ο IIS είναι στενά συνδεδεμένος με τα Windows NT. Μοιράζεται εφαρμογές, interfaces και εργαλεία μαζί με τις υπηρεσίες των Windows NT, όπως τα User Manager, Performance Monitor και Event Viewer. Ο IIS μπορεί να χρησιμοποιήσει ακόμη και τον SQL Server, για να καταγράψει στατιστικά server, και το Windows NT event log για να παρακολουθήσει πληροφορίες ασφάλειας και πρόσβασης.



Ο IIS χρησιμοποιεί standard πρωτόκολλα υπηρεσιών και δικτύου. Εφαρμογές client δημιουργούν συνδέσεις δικτύου στον IIS χρησιμοποιώντας standard Windows Sockets και TCP/IP Internet πρωτόκολλα. Οποιαδήποτε client εφαρμογή μπορεί να δημιουργήσει μία σύνδεση δικτύου στον IIS αρκεί να χρησιμοποιεί τα standard Internet πρωτόκολλα.

Οι Web browsers είναι οι πιο συχνά χρησιμοποιούμενοι τύποι client εφαρμογών. Επικοινωνούν με μια IIS υπηρεσία χρησιμοποιώντας Standard FTP, Gopher ή HTTP πρωτόκολλα. Η IIS υπηρεσία FTP μεταφέρει αρχεία μεταξύ του IIS και ενός άλλου υπολογιστή στο Internet. Η IIS υπηρεσία Gopher είναι ένα εργαλείο για browsing μεταξύ αρχείων και καταλόγων στο Internet. Η WWW υπηρεσία επεξεργάζεται μια μοναδική συναλλαγή μεταξύ μιας client εφαρμογής και του IIS.

---

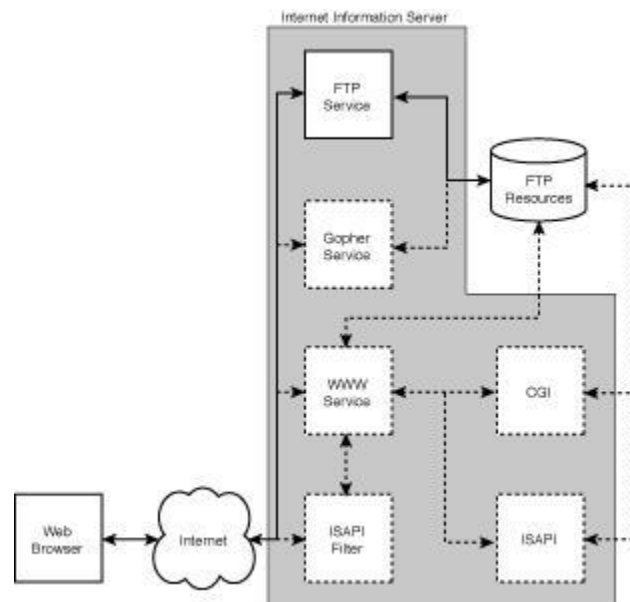
## Πρωτόκολλο TCP/IP

Οι client εφαρμογές δικτύου επικοινωνούν με τις υπηρεσίες του IIS χρησιμοποιώντας το πρωτόκολλο TCP/IP. Μερικές φορές καλείται και Internet Protocol Suite.

Το πρωτόκολλο TCP/IP έχει τέσσερα επίπεδα:

- Επίπεδο σύνδεσης  
Το επίπεδο σύνδεσης δημιουργεί το interface των Windows NT στο δίκτυο. Χειρίζεται όλες τις λεπτομέρειες σχετικά με τη σύνδεση μέσω interface του οδηγού συσκευής δικτύου με την κάρτα interface δικτύου.
- Επίπεδο δικτύου  
Το επίπεδο δικτύου χειρίζεται την κίνηση των πακέτων στο δίκτυο. Το IP είναι σε αυτό το επίπεδο. Αποκρύπτει το βασικό φυσικό δίκτυο δημιουργώντας μια όψη εικονικού δικτύου. Το επίπεδο δικτύου είναι ένα αναξιόπιστο, best-effort connectionless πρωτόκολλο παράδοσης πακέτου. Μετακινεί πακέτα από την πηγή στον προορισμό αλλά δεν υπάρχει εγγύηση ότι αυτά θα φτάσουν εκεί.
- Επίπεδο μεταφοράς  
Το επίπεδο μεταφοράς χειρίζεται τη ροή των δεδομένων μεταξύ της εφαρμογής client και του επιπέδου εφαρμογών. Το TCP (Transmission Control Protocol) είναι σε αυτό το επίπεδο. Το TCP παρέχει ανάκαμψη από σφάλματα και έλεγχο ροής των δεδομένων μεταξύ της client εφαρμογής δικτύου και των υπηρεσιών IIS.
- Επίπεδο εφαρμογών  
Το επίπεδο εφαρμογών χειρίζεται τις συνδέσεις με τα επίπεδα δικτύου και μεταφοράς. Δεν σχετίζεται με το πώς τα δεδομένα κινούνται στο δίκτυο. Το Windows Sockets interface βρίσκεται στο επίπεδο εφαρμογών.

## Βασικές Υπηρεσίες IIS



### Υπηρεσία FTP

Η υπηρεσία FTP του IIS μπορεί να μεταφέρει οποιοδήποτε αρχείο μεταξύ του server κι ενός FTP client. Χειρίζεται τη σύγχρονη πρόσβαση από πολλαπλούς FTP clients. Κάθε FTP client εγκαθιστά μία σύνδεση socket στην υπηρεσία FTP του IIS και συνδέεται με αυτήν. Οι FTP clients χρησιμοποιούν ένα περιορισμένο σύνολο εντολών και έχουν περιορισμένη πρόσβαση αρχείων. Η σύνδεση socket στην υπηρεσία FTP διαρκεί έως ότου ο FTP client αποσυνδεθεί. Το FTP είναι ένα από τα πρώτα πρωτόκολλα TCP/IP. Οι Web browsers και άλλες γραφικές interface εφαρμογές έχουν αντικαταστήσει τις πρώιμες FTP client εφαρμογές. Οι περισσότερες υπηρεσίες FTP δεν παρέχουν περιγραφές αρχείων. Το browsing στους καταλόγους είναι μία αργή διαδικασία.

### Υπηρεσία Gopher

Το Internet Gopher είναι ένα εργαλείο για browsing σε αρχεία και καταλόγους στο Internet. Ένας Gopher client εγκαθιστά μια socket σύνδεση στην υπηρεσία Gopher του IIS. Συνήθως δεν απαιτεί από τον client να συνδεθεί. Ο Gopher client εμφανίζει μια ιεραρχία από αντικείμενα και καταλόγους, όπως περίπου ένα σύστημα αρχείων σε ένα μενού από επιλογές. Μπορεί να είναι μια λίστα από αρχεία, υποκαταλόγους ή συνδυασμό των δύο. Ο Gopher client αντιγράφει ένα επιλεγμένο αρχείο στο δίκτυο και το εμφανίζει. Το μενού Gopher μπορεί να δείχνει σε άλλα αρχεία και καταλόγους που βρίσκονται σε άλλους Gopher servers στο Internet. Ήταν η πρώτη υπηρεσία Internet που προσέφερε αυτό το χαρακτηριστικό. Τέλος το Gopher έχει περιορισμένες



δυνατότητες γραφικής αναπαράστασης. Δεν μπορεί να παρουσιάζει γραφικά και κείμενο ταυτόχρονα.

Το Gopher και το HTTP είναι παρόμοια πρωτόκολλα δικτύου. Έγιναν διαθέσιμα και τα δύο περίπου την ίδια περίοδο. Οι περισσότερες σελίδες στο Internet δεν προσφέρουν τις υπηρεσίες Gopher. Πολλές παλιότερες σελίδες έχουν σταματήσει να τις προσφέρουν. Έχουν μετατρέψει τα έγγραφα Gopher σε έγγραφα HTML επειδή το HTML μπορεί να παρουσιάζει γραφικά και κείμενο μαζί. Τα έγγραφα HTML βασίζονται στο πρωτόκολλο HTTP.

### Υπηρεσία WWW

Ο Web browser που επεξεργάζεται έγγραφα HTML χρησιμοποιεί το πρωτόκολλο HTTP ώστε να συναλλάγεί με την υπηρεσία WWW του IIS. Η υπηρεσία WWW γνωρίζει πώς να αποκριθεί σε μια αίτηση HTML αναλύοντας τα πεδία URL. Μπορεί να καθορίσει αν η αίτηση είναι για μια στατική HTML σελίδα, για μια εφαρμογή ISAPI, για μια εφαρμογή CGI, ή για ένα Internet σύνδεσμο βάσης δεδομένων (Internet Database Connector).

- Στατική HTML σελίδα  
Η υπηρεσία WWW ψάχνει για ένα όνομα αρχείου ή ένα πεδίο μονοπατιού. Π.χ.  
<http://www.infomax.com/welcome.htm>
- Εφαρμογή ISAPI  
Η υπηρεσία WWW ψάχνει για ένα όνομα αρχείου ή ένα πεδίο μονοπατιού με την κατάληξη DLL. Π.χ.  
<http://www.infomax.com/isapiapps/isapidoit.dll>
- Εφαρμογή CGI  
Η υπηρεσία WWW ψάχνει για ένα όνομα αρχείου ή ένα πεδίο μονοπατιού με μια κατάληξη που έχει συνδεθεί με την εφαρμογή. Π.χ.  
<http://www.infomax.com/cgiapps/gcidoit.pl>
- Internet Database Connector  
Η υπηρεσία WWW ψάχνει για ένα όνομα αρχείου ή ένα πεδίο μονοπατιού με την κατάληξη IDC. Π.χ.  
<http://www.infomax.com/idcapps/idcdoit.idc>

Το Hypertext Transmission Protocol είναι ένα πρωτόκολλο χωρίς κατάσταση που σχεδιάστηκε για να επεξεργάζεται μια μοναδική συναλλαγή κατά τη διάρκεια τη σύνδεσης με έναν server. Βρίσκεται στα TCP και IP πρωτόκολλα.

Τα τέσσερα βήματα κατά τη διάρκεια μιας απλής συναλλαγής HTTP είναι:

- Σύνδεση  
Ο client HTTP εγκαθιστά μια σύνδεση socket στον IIS
- Αίτηση  
Ο client HTTP στέλνει μια αίτηση στην υπηρεσία WWW του IIS. Η αίτηση περιέχει τον τύπο της υπηρεσίας HTTP και άλλη πληροφορία
- Απόκριση  
Η υπηρεσία WWW στέλνει μια απόκριση πίσω στον HTTP client. Η απόκριση περιέχει την κατάσταση της συναλλαγής και τα δεδομένα που ζητήθηκαν
- Αποσύνδεση  
Η υπηρεσία WWW σηματοδοτεί τη λήξη της συναλλαγής κλείνοντας τη σύνδεση socket

Ο Web browser δημιουργεί μια σύνδεση με την υπηρεσία WWW για κάθε αρχείο που είναι μέρος του εγγράφου HTML. Δημιουργεί μια σύνδεση πρώτα για το κείμενο HTML. Ο IIS αποσυνδέεται αφού το στείλει. Ο Web browser αναλύει κάθε λέξη του αρχείου κειμένου ψάχνοντας για ονόματα αρχείων γραφικών εικόνων. Έπειτα δημιουργεί μια σύνδεση με τον IIS και ζητά ένα μοναδικό αρχείο γραφικής εικόνας. Ο IIS αποσυνδέεται αφού το στείλει. Αυτή η διεργασία επαναλαμβάνεται για κάθε αρχείο εικόνας.

Το HTTP είναι το περισσότερο χρησιμοποιούμενο Internet πρωτόκολλο. Απασχολεί το 25% των πακέτων του Internet. Το πρωτόκολλο FTP έρχεται δεύτερο απασχολώντας το 15% των πακέτων.

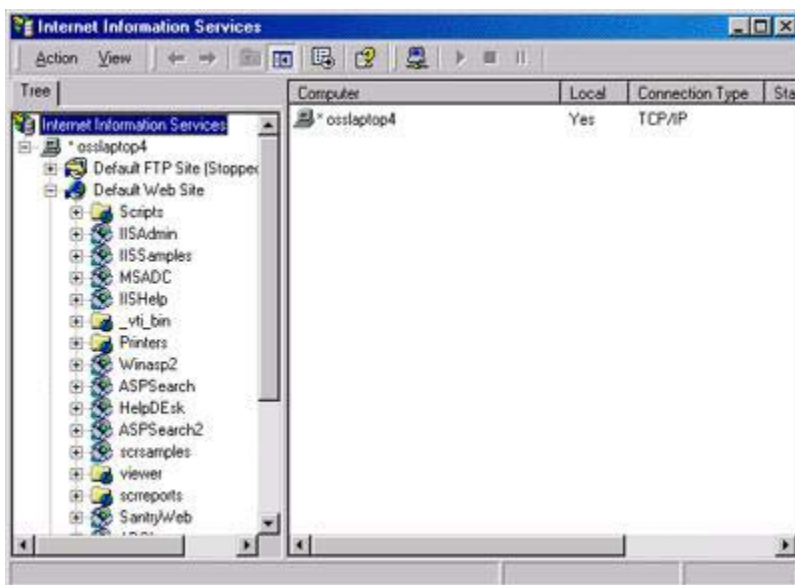
---

## Διαχείριση του IIS

### MMC

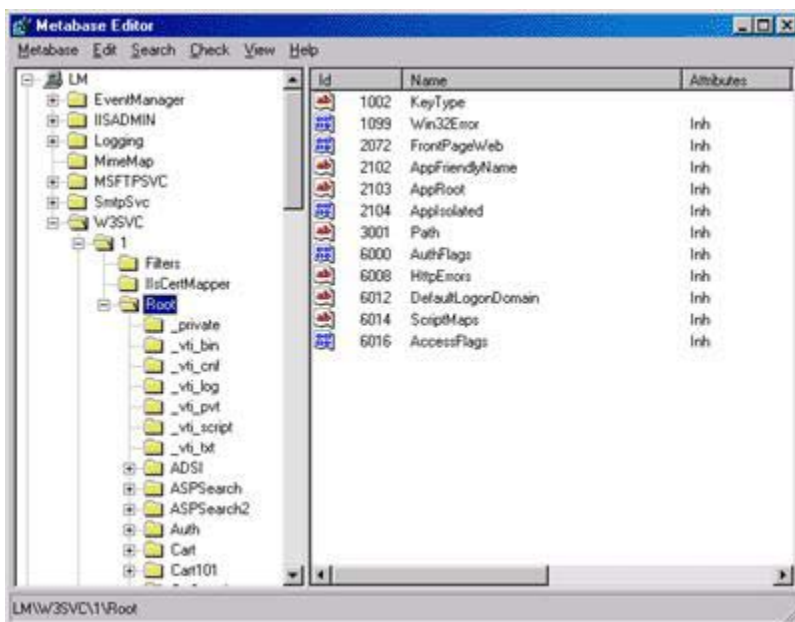
Ο διαχειριστής προσπελαύνει τον IIS μέσω ενός interface όπως το Microsoft Management Control (MMC), το οποίο χρησιμοποιείται για το χειρισμό της IIS Metabase.

Στα Windows XP, μπορούμε να τρέξουμε το MMC επιλέγοντας *Start* → *Control Panel* → *Administrative Tools* → *Internet Information Services*. Θα μελετήσουμε το MMC στη συνέχεια, όπου χρειαστεί αναφορά σε αυτό.



## Metabase

Η metabase είναι μια ψευδο-registry για τον IIS. Όταν ο διαχειριστής κάνει αλλαγές στο δίκτυο δια μέσου του MMC, οι τιμές που περιέχονται στην metabase αλλάζουν. Η metabase είναι παρόμοια με τη δομή της registry των Windows. Είναι μια ιεραρχική δομή που αντικατοπτρίζει τις ρυθμίσεις του IIS. Η metabase έχει κόμβους που ονομάζονται κλειδιά και κάθε κλειδί μπορεί να περιέχει τιμές που επηρεάζουν την ιδιότητα ρύθμισης του IIS. Το παρακάτω σχήμα δείχνει τη δομή της metabase του IIS 5.0 χρησιμοποιώντας ένα εργαλείο που ονομάζεται MetaEdit και το οποίο παρέχεται από το Windows 2000 Server Resource Kit.



## Active Directory Services Interface (ADSI)

Το ADSI παρέχει ένα interface στην metabase του IIS, επιτρέποντας τον προγραμματιστή να χειρίζεται IIS ρυθμίσεις μέσω μιας scripting γλώσσας προγραμματισμού όπως η VBScript. Παρόλο που δεν είναι άμεσα συνδεδεμένο μέρος της αρχιτεκτονικής του IIS, το ADSI μπορεί να ενεργοποιήσει τη δημιουργία εφαρμογών που επιτρέπουν σε διάφορες διαχειριστικές εργασίες να κεντριοκοποιηθούν σε ένα περιβάλλον εφαρμογών. Το ADSI παρέχει αυτό το interface διαχείρισης δημιουργώντας ένα σύνολο από ανοιχτά interfaces σε διάφορους παροχείς καταλόγων. Αυτοί οι παροχείς μπορεί να είναι η IIS metabase, το Windows Active Directory, το Novell NDS, ή άλλες συμβατές υπηρεσίες καταλόγου.

## Άλλες σχετικές υπηρεσίες του IIS

- Υπηρεσίες δεικτοδότησης  
Οι υπηρεσίες δεικτοδότησης παρέχουν ένα μηχανισμό για δεικτοδότηση και εύρεση δύσκολων εγγράφων. Ένας προγραμματιστής μπορεί να δημιουργήσει scripts αναζήτησης και να έχει τη διαβεβαίωση ότι οι υπηρεσίες δεικτοδότησης θα εμφανίζουν την περίληψη ενός εγγράφου μόνο σε αυτούς που έχουν πρόσβαση στο έγγραφο. Αν αποκλείσετε ρητά έναν χρήστη από ένα έγγραφο, τότε οι υπηρεσίες δεικτοδότησης δεν θα το εμφανίζουν στα αποτελέσματα της αναζήτησης.
- Microsoft Message queue (MSMQ)  
Το MSMQ είναι μία τεχνολογία για μηνύματα που κρατώνται σε ουρά και το οποίο επιτρέπει την επικοινωνία εφαρμογών μεταξύ διαφόρων συστημάτων. Το MSMQ είναι ανεξάρτητο από τον τρόπο σύνδεσης του δικτύου, οπότε μπορεί να ανιχνεύει πότε μία σύνδεση δικτύου είναι ενεργή και να συνεχίζει τη μεταφορά μηνυμάτων όταν εγκαθίσταται η σύνδεση. Το MSMQ κάνει αυτήν την connectionless ή αλλιώς ασύγχρονη επικοινωνία επιτρέποντας σε μια εφαρμογή να αποθηκεύει ένα μήνυμα μέσα στην ουρά και επιτρέποντας σε μια άλλη εφαρμογή να ελέγχει την ουρά για ένα μήνυμα που προορίζεται για αυτήν. Το MSMQ μπορεί είτε να περιμένει για μία απόκριση είτε να επιτρέπει στην εφαρμογή να συνεχίσει την επεξεργασία. Γι'αυτόν το λόγο, επιτρέπει στην εφαρμογή να μην ανησυχεί για την μεταφορά των μηνυμάτων και να εμπιστεύεται το MSMQ για την παράδοση των μηνυμάτων.
- Network News Transfer Protocol (NNTP)  
Ο IIS παρέχει έναν NNTP server για διαλόγους με χρήση νημάτων. Όπως και στις άλλες υπηρεσίες που παρέχονται από τον IIS, μπορεί κάποιος να διαχειριστεί την υπηρεσία NNTP μέσω του MMC. Ο NNTP server είναι επίσης ολοκληρωμένος

με την ασφάλεια των Windows επιτρέποντάς σας να διαχειριστείτε την πρόσβαση χρησιμοποιώντας κοινά εργαλεία ασφάλειας των Windows.

- Simple Mail Transport Protocol (SMTP)

Πολλές δικτυακές εφαρμογές χρειάζονται τη δυνατότητα αποστολής mail και ο IIS 5.0 παρέχει αυτό το χαρακτηριστικό με τον SMTP server του. Ο SMTP server είναι εύκολο να εγκατασταθεί, να τον διαχειριστούν και να ολοκληρωθεί σε μια δικτυακή εφαρμογή. Υπάρχουν πολλά αντικείμενα COM διαθέσιμα για αποστολή mail από τον Web server. Έχετε επίσης τη δυνατότητα να δημιουργήσετε αρχεία κειμένου που να συμμορφώνονται με τις προδιαγραφές για την αποστολή e-mail και να τοποθετούν αυτά τα μηνύματα στην ουρά mail του SMTP server. Από τη στιγμή που αυτά τα μηνύματα βρίσκονται στην ουρά, μπορούν να ανακτηθούν και να σταλούν από τον SMTP server.

---

## **Ικανότητα προγραμματισμού (programmability)**

Νωρίτερα μιλήσαμε για μια συναλλαγή HTTP. Ένας client κάνει μια αίτηση, η αίτηση λαμβάνεται από τον IIS και αν πρόκειται για στατικό έγγραφο HTML τότε ανακτάται από το σύστημα αρχείων και αποστέλλεται στον client. Αν η αίτηση απαιτεί να την χειριστεί κάποιο ISAPI φίλτρο όπως το ASP, η αίτηση τότε υφίσταται επεξεργασία από την κατάλληλη scripting μηχανή όπως η VBScript. Το programmability είναι αυτό που κάνει τις δικτυακές εφαρμογές εφικτές. Η επόμενη παράγραφος ορίζει με συντομία τις ατομικές διεργασίες που υλοποιούν το programmability του IIS.

### **Internet Services Application Programming Interface (ISAPI)**

Οι εφαρμογές ISAPI τρέχουν πιο γρήγορα από τα προγράμματα CGI επειδή είναι DLLs (Dynamic Link Libraries) που τρέχουν ως μέρος της λειτουργίας και του ίδιου χώρου διευθύνσεων του IIS. Αυτός είναι ο λόγος που οι εφαρμογές ISAPI είναι γνωστές ως εφαρμογές in-process.

Το ISAPI παρέχει ένα επίπεδο που διακόπτει τις αιτήσεις client πριν εξυπηρετηθεί η απόκριση. Το ISAPI επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές χαμηλού επιπέδου, όπως συνηθισμένους παροχείς ασφάλειας και διερμηνευτές scripts.

Οι εφαρμογές ISAPI κατηγοριοποιούνται ως εξής:

1. Καταλήξεις ISAPI

Αυτές καλούνται με απαίτηση όταν χρειαστούν. Για παράδειγμα έχετε μία φόρμα HTML που περνά τιμές σε ένα ISAPI dll στη μορφή ενός query string. Π.χ.  
<http://www.santry.com/isapi.dll?value1=1&value2=2>

Ένα query string είναι η τιμή που αναγράφεται στο URL μετά το '?’.

## 2. Φίλτρα ISAPI

Φορτώνονται στο χώρο μνήμης του IIS όταν ξεκινά η υπηρεσία “Inetinfo.exe” και παραμένουν στη μνήμη για όσο χρειαστεί. Τα φίλτρα ISAPI αρχικοποιούνται όταν η υπηρεσία WWW ξεκινάει και παραμένουν στη μνήμη μέχρι να σταματήσει η υπηρεσία. Αυτά συχνά παρέχουν συναρτήσεις συστήματος χαμηλού επιπέδου, όπως συνηθισμένο logging, παροχές πιστοποίησης και κρυπτογράφηση δεδομένων.

### Active Server Pages (ASP)

Το ASP για την ακρίβεια είναι μια κατάληξη ISAPI. Οι αιτήσεις για αρχεία ASP τρέχουν μέσω του φίλτρου ISAPI που ονομάζεται asp.dll, το οποίο βρίσκεται στον κατάλογο <systemroot>\system32\inetsrv\ . Το ASP είναι ένα περιβάλλον εκτέλεσης του IIS από τη μεριά του server που σας επιτρέπει να αρχικοποιείτε αντικείμενα COM στον server. Αυτά τα χαρακτηριστικά είναι αυτά που σας επιτρέπουν, ως προγραμματιστές, να δημιουργείτε δυναμικές και οδηγούμενες από τα δεδομένα σελίδες που ανταποκρίνονται σε διάφορες εισόδους που δίνονται από το χρήστη.

Τα αρχεία ASP διακρίνονται από μια κατάληξη .asp. Αυτή η κατάληξη γνωστοποιεί στο φίλτρο ISAPI ότι πρέπει να επεξεργαστεί το αρχείο πριν το στείλει στον client (π.χ. στον Web browser), σε αντίθεση με τα standard έγγραφα HTML που είναι στατικά και αποστέλλονται απευθείας στον client. Ο κώδικας ASP διαφοροποιείται από τον κώδικα HTML που περιέχεται σε ένα έγγραφο με δυο τρόπους:

- Χρησιμοποιώντας διακριτικά script – Το φίλτρο ASP θα ψάξει μέσα στο έγγραφο για ένα “<%” το οποίο δείχνει ότι ξεκινάει ο ASP κώδικας και ότι πρόκειται να γίνει κάποια επεξεργασία, και για ένα “%>” που δείχνει ότι ο ASP κώδικας τελειώνει και αρχίζει ο HTML.
- Προσδιορίζοντας πού θα υποστεί επεξεργασία το script μέσα στο script tag – Αν είστε εξοικειωμένοι στη δημιουργία κώδικα από την πλευρά του client τότε η μέθοδος προσδιορισμού του ASP κώδικα θα φαίνεται γνώριμος:

```
<script language="vbscript" runat=server>  
Some ASP code  
</script>
```

---

## 📖 Νήματα στον IIS

Είναι πολύ σημαντικό να καταλάβει κανείς πώς λειτουργεί το λεγόμενο **thread pooling** (δεξαμενή νημάτων) του IIS. Βλέποντας πώς υφίσταται επεξεργασία μια αίτηση ενός client από τον Web server, θα καταλάβετε γιατί κάποιες τεχνικές είναι πολύ καλύτερες από άλλες όταν θέλετε να βελτιστοποιήσετε το throughput και την κλιμάκωση της εφαρμογής σας.

Η βασική διαδικασία του IIS Web server (δηλαδή η Inetinfo.exe) έχει σχεδιαστεί ώστε να απασχολεί ένα μεγάλο αριθμό ταυτόχρονων χρηστών. Για να επεξεργαστεί εκατοντάδες ή πιθανόν χιλιάδες clients, ο IIS υιοθετεί έναν πολύπλοκο τρόπο διαχείρισης νημάτων για να εξισορροπήσει το χρόνο απίκρισης και το throughput (ωφέλιμο φορτίο) του συστήματος. Αντί να δημιουργεί ένα νέο νήμα για κάθε αίτηση, ο IIS διατηρεί μια δεξαμενή από νήματα-εργάτες που είναι αφοσιωμένα στο να επεξεργάζονται αιτήσεις. Υπάρχουν και κάποια άλλα νήματα που τρέχουν μέσα στον IIS αλλά αυτά δεν θα μας απασχολήσουν. Καθώς εισερχόμενες αιτήσεις καταφθάνουν στον Web server, ο IIS πρέπει με κάποιον τρόπο να στείλει τα νήματα από τη δεξαμενή για επεξεργασία.

Γιατί όμως είναι τόσο σημαντική η δεξαμενή νημάτων; Τα νήματα είναι ακριβά. Κάθε ένα καταναλώνει το μερίδιό του σε πόρους. Αν προσθέσετε και το επιπλέον κόστος για τον προγραμματισμό και τη δρομολόγησή τους από και προς τον επεξεργαστή του συστήματος, μπορείτε να δείτε γιατί μια εφαρμογή που χρησιμοποιεί ένα νήμα για κάθε client αρχίζει και μειώνει την απόδοσή της όταν ο αριθμός των clients φτάνει τους εκατοντάδες ή τους χιλιάδες.

Ένας αλγόριθμος thread-pooling εξοικονομεί σημαντικούς κύκλους επεξεργασίας από τη μεριά του server επειδή δεν υπάρχει καμιά αυξανόμενη ανάγκη για δημιουργία ή καταστροφή νημάτων. Το thread pooling θέτει επίσης ένα άνω όριο στον αριθμό των νημάτων, το οποίο βοηθάει στη συντήρηση άλλων πόρων από τη μεριά του server. Γι'αυτόν το λόγο μία δεξαμενή νημάτων καθιστά δυνατόν για μια εφαρμογή να απασχολεί άνετα ένα μεγαλύτερο αριθμό χρηστών.

Ο IIS δουλεύει με μία default δεξαμενή με μέχρι 10 νήματα εργάτες ανά επεξεργαστή. Ο αριθμός των νημάτων μπορεί να ρυθμιστεί προσθέτοντας εγγραφές στο registry των Windows.

Αφού υπάρχει ένας πεπερασμένος αριθμός νημάτων-εργατών αφοσιωμένα στην επεξεργασία αιτήσεων, ο IIS χρησιμοποιεί μια ουρά για τη δρομολόγηση των εισερχόμενων αιτήσεων των clients όταν όλα τα διαθέσιμα νήματα είναι απασχολημένα με την επεξεργασία άλλων αιτήσεων. Οι αιτήσεις εξυπηρετώνται σε μια βάση «πρώτη

εισερχόμενη – πρώτη εξυπηρετούμενη». Αν ο server δέχεται περισσότερες αιτήσεις ανά δευτερόλεπτο από ότι μπορεί να χειριστεί, η ουρά των αιτήσεων θα αναπτύσσεται όλο και περισσότερο. Όταν η ουρά μεγαλώσει πάρα πολύ, οι χρόνοι απόκρισης στους οποίους εξυπηρετώνται οι εισερχόμενες αιτήσεις θα μειωθούν δραματικά. Ο IIS θέτει μια μέγιστη χωρητικότητα για την ουρά έτσι ώστε το απόθεμα των αιτήσεων να μην αυξάνεται τόσο πολύ.

Στην default του ρύθμιση, ο IIS επιτέπει στην ουρά να μεγαλώσει μέχρι ένα μέγιστο μέγεθος 500 αιτήσεων. Όταν η ουρά φτάσει στη μέγιστη χωρητικότητά της, πρόσθετες εισερχόμενες αιτήσεις απορρίπτονται με ένα σφάλμα *Server Too Busy*. Αυτή η συμπεριφορά είναι πολύ λογική επειδή επιτρέπει στον IIS να απορρίπτει όποια αίτηση δεν μπορεί να χειριστεί μέσα σε κάποιο σύντομο χρονικό διάστημα. Το μέγιστο μέγεθος της ουράς μπορεί επίσης να ρυθμιστεί.

Ρυθμίζοντας το μέγεθος της ουράς σας επιτρέπεται να κανονίσετε το σημείο στο οποίο ο IIS απορρίπτει τις εισερχόμενες αιτήσεις. Καθώς ρυθμίζετε αυτήν την τιμή, κάνετε απόπειρα να πετύχετε μια ισορροπία μεταξύ της υψηλής διαθεσιμότητας και των μικρών χρόνων απόκρισης. Το όλο θέμα είναι να ρυθμίσετε το μέγεθος της ουράς για να χειριστείτε τις λεγόμενες κορυφές (peaks) μικρής περιόδου και να περιορίσετε έτσι τα αποθέματα της ουράς κατά τη διάρκεια ακραίου φόρτου εργασίας. Αν ο Web server σας υπερφορτωθεί, είναι καλύτερο να αποστέλλονται μηνύματα σφάλματος πίσω στους clients αντί να περιμένουν αυτοί τόσο πολύ.

Ας ανακεφαλαιώσουμε λοιπόν πώς λειτουργεί το μοντέλο νημάτων στην default εγκατάσταση του IIS σε ένα μηχάνημα με έναν μοναδικό επεξεργαστή. Υπάρχουν 10 διαθέσιμα νήματα-εργάτες μέσα στη δεξαμενή. Όταν μια αίτηση καταφθάνει, ο IIS την αναθέτει σε ένα ανενεργό νήμα από τη δεξαμενή, αν υπάρχει τέτοιο διαθέσιμο. Σημειώστε ότι ο IIS είναι ελεύθερος να χρησιμοποιεί οποιοδήποτε νήμα είναι διαθέσιμο μέσα στη δεξαμενή για να επεξεργαστεί την αίτηση. Αν όλα τα νήματα είναι απασχολημένα με την επεξεργασία άλλων αιτήσεων, οι εισερχόμενες αιτήσεις μπαίνουν στην ουρά και υφίστανται επεξεργασία σε μια βάση «πρώτη εισερχόμενη – πρώτη εξυπηρετούμενη». Όσο η ουρά δεν φτάνει τη χωρητικότητα των 500 αιτήσεων, όλες οι αιτήσεις θα υφίστανται επεξεργασία στον πρόπονα χρόνο.



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 4

**ΠΑΡΑΚΟΛΟΥΘΗΣΗ ΑΠΟΔΟΣΗΣ**  
**(PERFORMANCE MONITORING)**

**Ο**ποιοδήποτε σύστημα κι αν μελετήσουμε, δεν μπορούμε να βελτιώσουμε την απόδοσή του αν πρώτα δεν το μετρήσουμε. Ωστόσο, το να μετρήσουμε το σύστημα είναι μόνο η μισή δουλειά. Πρέπει να καταλάβουμε πού να ψάξουμε όταν μετράμε το σύστημά μας. Παρατηρώντας απλά κάποιους μετρητές δεν μπορούμε να ολοκληρώσουμε τη δουλειά μας, εάν δεν ξέρουμε να μεταφράσουμε τα δεδομένα που συλλέξαμε. Σ' αυτό το κεφάλαιο μπορείτε να δείτε με ποιον τρόπο μπορείτε να συλλέξετε τα δεδομένα που θα σας βοηθήσουν στη συνέχεια στη δύσκολη δουλειά της βελτίωσης της απόδοσης του συστήματός σας.

---

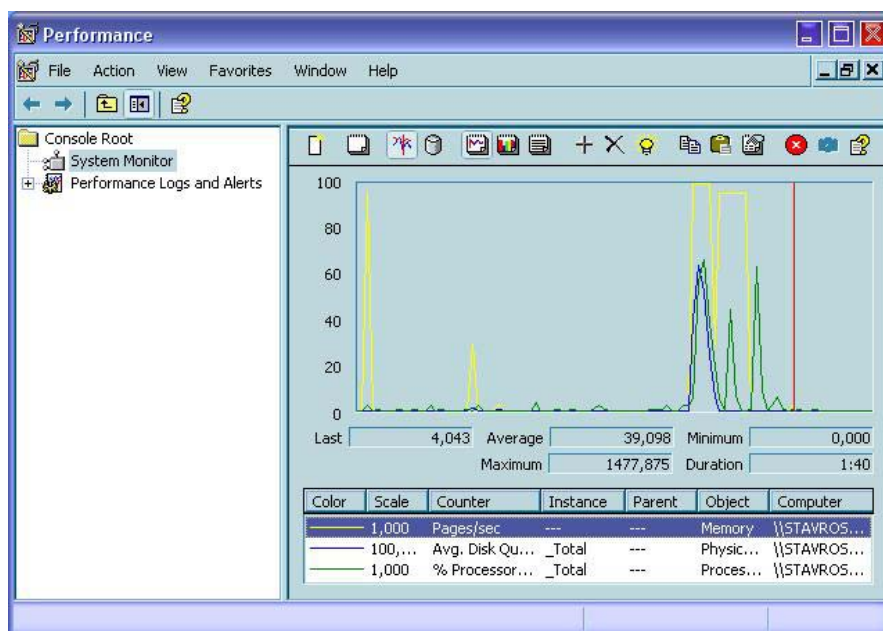
## **Κονσόλα MMC (Microsoft Management Console)**

Τα Windows 2000 (και κατ'επέκταση και τα Windows XP) σας επιτρέπουν να διαμορφώσετε μια κονσόλα MMC, έτσι ώστε να έχετε στη διάθεσή σας έναν αριθμό από διαχειριστικά εργαλεία σε μία μόνο κονσόλα. Με αυτόν τον τρόπο, μπορείτε να βλέπετε την οθόνη συτήματος (System Monitor) και να διαχειρίζεστε τους DHCP (Dynamic Host Configuration Protocol) Servers και τον Microsoft SQL Server 7.0 από την ίδια κονσόλα διαχείρισης.

Στην default ρύθμιση, η Performance console είναι ρυθμισμένη να έχει ως διαχειριστικές εφαρμογές τις **System Monitor** και **Performance Logs and Alerts**. Μπορείτε να τρέξετε την κονσόλα κάνοντας κλικ στο *Start* → *Control Panel* → *Administrative Tools* → *Performance*. Το MMC ανοίγει, παρουσιάζοντας τις εφαρμογές System Monitor και Performance Logs and Alerts. Από εδώ μπορείτε να βλέπετε την τρέχουσα δραστηριότητα του server και να επιλέγετε την πληροφορία που θα συλλεχθεί για ανάλυση.

Όταν τρέχετε το System Monitor για να συλλέξετε δεδομένα για τον υπολογιστή σας, μπορείτε να δείτε πληροφορίες σχετικά με την απόδοση σε πραγματικό χρόνο. Η ανάλυση απόδοσης σε πραγματικό χρόνο είναι ιδιαίτερα χρήσιμη όταν δεν μπορείτε να περιμένετε να συλλεχθούν τα δεδομένα στα logs και τα reports.

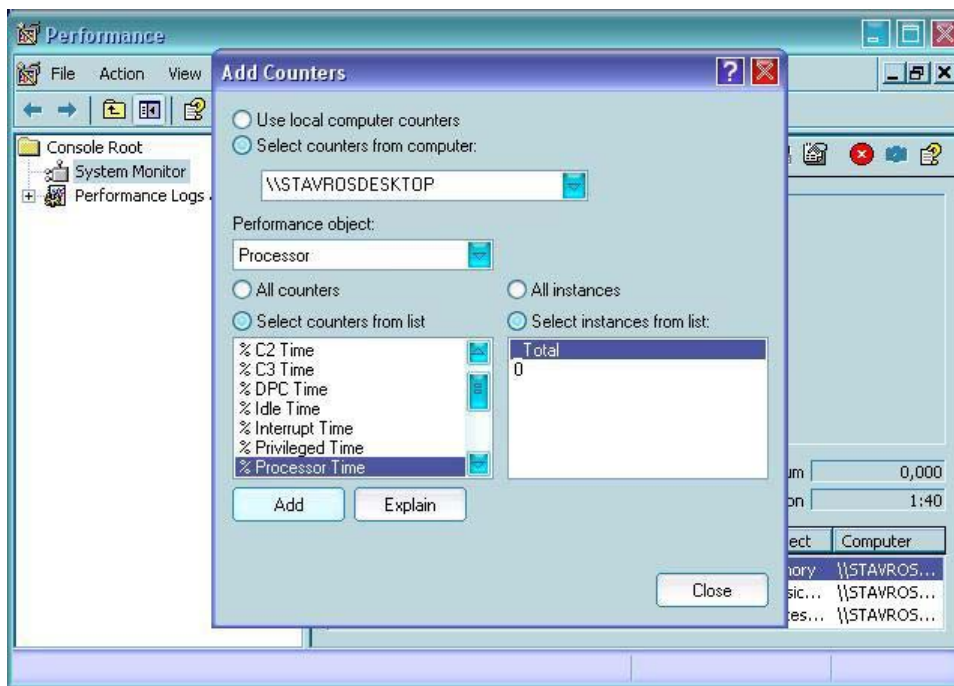
Χρησιμοποιώντας το chart view ή το report view στο System Monitor, είστε σε θέση να ελέγχετε την απόδοση των Windows και των εφαρμογών που τρέχουν στο server. Έτσι εντοπίζοντας ένα spike (δηλαδή μια αιχμή στο γράφημα) στη χρησιμοποίηση (utilization) του επεξεργαστή όταν τρέχει μια συγκεκριμένη εφαρμογή στον server, μπορείτε να κατευθύνθείτε σωστά όταν κάνετε διάγνωση σε ένα πρόβλημα.



Ως network administrator ενός server, θα αντιμετωπίσετε καταστάσεις στις οποίες τα προβλήματα που προκύπτουν δεν μπορούν να διαγνωστούν εύκολα μελετώντας την απόδοση του συστήματος σε πραγματικό χρόνο. Ίσως να χρειαστεί να συλλέξετε δεδομένα μέσα σε ένα χρονικό διάστημα, από μια ώρα μέχρι και μερικές μέρες. Σε αυτές τις περιπτώσεις μπορείτε να βλέπετε τα δεδομένα των μετρητών που συλλέγονται από τα Performance Logs and Alerts, τόσο κατά τη διάρκεια της συλλογής όσο και μετά το τέλος της.

## Objects, Counters και Instances

Οι **counters** (μετρητές) είναι αυτοί που σας επιτρέπουν να μετράτε την απόδοση του συστήματός σας. Σας επιτρέπουν, για την ακρίβεια, να μετράτε τις αποδόσεις των συσκευών hardware και των εφαρμογών software μαζί. Για παράδειγμα, όταν μετράτε τη δραστηριότητα των συσκευών hardware, κάποιοι counters μετράνε τις επισκέψεις του λειτουργικού σας συστήματος στη συσκευή. Άλλοι counters που είναι μέρος των εφαρμογών software που τρέχουν στον server, μετράνε το utilization των πόρων του server από τις εφαρμογές και δίνουν επίσης αναφορά για την απόδοση μιας συγκεκριμένης εφαρμογής. Μπορείτε να προσθέσετε έναν counter κάνοντας κλικ στο '+' που υπάρχει πάνω από το γράφημα, στα δεξιά της Performance console.



Επειδή ο αριθμός των counters που μπορείτε να προσθέσετε είναι πάρα πολύ μεγάλος και επειδή ο καθορισμός του σκοπού του καθενός είναι δύσκολος, οι counters είναι οργανωμένοι σε μια λογική ιεραρχία, η οποία είναι ορισμένη από τη δομή του (μετρήσιμου) hardware εξοπλισμού και των (μετρήσιμων) software στοιχείων. Στην κορυφή της ιεραρχίας αυτής βρίσκεται ο υπολογιστής. Κάθε υπολογιστής έχει διακεκριμένο αριθμό στοιχείων που ονομάζονται **objects**. Τα objects είναι διαθέσιμα για συσκευές hardware όπως ο φυσικός δίσκος, η μνήμη, οι επεξεργαστές και οι προσαρμογείς δικτύου. Άλλα objects περιλαμβάνουν counters για τη μέτρηση άλλων ατομικών διεργασιών (Active Server Pages).

Το σχήμα ονοματολογίας των counters ακολουθεί τη λογική σειρά *object\counter\instance*. Για παράδειγμα, η συμβατική ονοματολογία για ένα object επεξεργαστή είναι *Processor (object)\Utilization (counter)\0 (instance – για έναν μοναδικό επεξεργαστή που αναγνωρίζεται ως 0)*. Σε αυτήν την περίπτωση, ο counter μετράει το ολικό utilization του επεξεργαστή. Επειδή μπορεί να υπάρχουν πολλαπλοί επεξεργαστές σε ένα σύστημα, τα objects μπορεί να έχουν πολλαπλά **instances**. Έτσι, στην περίπτωση ενός server που έχει δύο επεξεργαστές, έχετε να επιλέξετε μεταξύ δύο instances. Για παράδειγμα, το πρώτο instance επεξεργαστή είναι το 0, ενώ το άλλο είναι 1. Μπορείτε να ελέγξετε έναν επεξεργαστή τη φορά ή και τους δυο μαζί. Ομοίως, μπορεί να συναντήσετε πολλαπλά instances στις κάρτες δικτύου, στους σκληρούς δίσκους, κ.λ.π..

---

## Περιγραφή Σημαντικών Performance Objects

Ο παρακάτω πίνακας περιγράφει τα πιο συνηθισμένα objects και το τι μπορεί κάθε object να σας πει σχετικά με το server σας. Χρησιμοποιήστε αυτή τη λίστα ως αφετηρία για τη διάγνωση πρόβλημάτων του συστήματός σας. Παρόλο που οι υπόλοιποι counters θα σας βοηθήσουν να ερευνήσετε με περισσότερη λεπτομέρεια, αυτά τα objects σας παρέχουν την καλύτερη γενική όψη της απόδοσης του server σας.

Όνομα Object	Περιγραφή
<u>Cache</u>	Η cache του συστήματος αρχείων χρησιμοποιείται για την προσωρινή αποθήκευση δεδομένων. Ένας υψηλός cache counter υποδηλώνει καλή απόδοση συστήματος επειδή τα δεδομένα ανακτώνται από την cache και όχι από το δίσκο. Επίσης μπορείτε να καταλάβετε κατά πόσο προσεκτικά έχει γραφτεί μια εφαρμογή χρησιμοποιώντας τους flush counters. Αυτοί οι μετρητές δείχνουν πόσο συχνά η εφαρμογή αναγκάζει τα Windows να κάνουν flush την cache στο δίσκο (δηλαδή πόσο συχνά τον ενημερώνουν). Ένας μεγάλος αριθμός flushes μπορεί να υποδηλώνει ότι ο προγραμματιστής δεν κατάλαβε τα ωφέλη από τη χρήση της cache. Από την άλλη μεριά, πολύ λίγα flushes μπορεί να υποδηλώνουν ότι η εφαρμογή στηρίζεται πολύ στην cache και υπάρχει περίπτωση να χάσει δεδομένα αν πέσει το σύστημα απροσδόκητα.
<u>Μνήμη</u>	Η RAM χρησιμοποιείται από το λειτουργικό σύστημα και τις εφαρμογές. Ο κανόνας λέει ότι όσο περισσότερη μνήμη διαθέτει ένα σύστημα, τόσο γρηγορότερο είναι αυτό. Ωστόσο η RAM δεν είναι το Α και το Ω, για τη ρύθμιση του συστήματός σας. Για έναν πραγματικά αποδοτικό server, όλες οι όψεις του hardware πρέπει να ρυθμιστούν ώστε να συμπίπτουν με την ποσότητα μνήμης που υπάρχει στο μηχάνημα. Ένας υψηλός μετρητής αναγνώσεων σελίδων (page reads) υποδηλώνει ότι σας τελειώνει η διαθέσιμη RAM και ότι το σύστημα καταφεύγει σε ανάγνωση/εγγραφή στο αρχείο σελίδας. Επειδή η ανάγνωση και η εγγραφή στη RAM είναι πολύ πιο γρήγορη από ότι στο δίσκο, το σύστημά σας παρουσιάζει σημάδια χαμηλής απόδοσης αν το αρχείο σελίδας υπερχρησιμοποιείται.
<u>Αντικείμενα</u>	Τα αντικείμενα λογισμικού του συστήματος περιλαμβάνουν

γεγονότα, mutexes, διεργασίες, τμήματα, σημαφόρους, και νήματα. Μετρώντας το thread counter για την απόδοση ενός αντικειμένου μπορείτε να ελέγχετε τον αριθμό των νημάτων που τρέχουν στον υπολογιστή σας την εκάστοτε χρονική στιγμή. Ένας υψηλός thread counter και ένας χαμηλός processes counter μπορούν να δείξουν ότι μια εφαρμογή είναι resource-intensive μόνο από τον αριθμό των νημάτων που χρειάζεται.

### Αρχείο Σελιδοποίησης

Το αρχείο σελιδοποίησης χρησιμοποιείται από το server για να διαβάζει και να γράφει συγκεκριμένα αντικείμενα δεδομένων στην εικονική μνήμη. Ελέγχοντας τη δραστηριότητα του αρχείου σελίδας βλέπετε πληροφορίες σχετικά με τους πόρους της μνήμης στο server. Ένας συνεχόμενα υψηλός usage counter του αρχείου σελίδας υποδηλώνει έλλειψη πόρων μνήμης.

### Φυσικός Δίσκος

Ο φυσικός δίσκος περιλαμβάνει συσκευές σκληρού δίσκου καθώς και RAID συσκευές που μπορεί να έχετε εγκαταστήσει στο server σας. Bottlenecks στο δίσκο είναι συχνά οι αιτίες του χρόνου lag του server. Θα πρέπει να ελέγχετε τη δραστηριότητα των δίσκων σας, ιδιαίτερα στην περίπτωση που έχετε εφαρμογές που απαιτούν συχνή I/O πρόσβαση σε αυτούς (π.χ. βάσεις δεδομένων).

### Διεργασία

Όταν εκκινείται ένα πρόγραμμα, η διεργασία είναι το αντικείμενο που δημιουργείται όταν ένα πρόγραμμα τρέχει. Κάθε πρόγραμμα θεωρείται ως στιγμιότυπο του performance object της διεργασίας. Ένα δεύτερο αντίγραφο ενός προγράμματος φαίνεται με ένα # και έναν αριθμό που υποδηλώνει τον αριθμό του στιγμιότυπου. Αν υποπτευθείτε ότι μια εφαρμογή δεν συμπεριφέρεται σωστά μη παραχωρώντας χρόνο επεξεργαστή, επιλέξτε το performance object της διεργασίας, ώστε να μπορέσετε να ελέγξετε την εφαρμογή.

### Επεξεργαστής

Ο επεξεργαστής είναι ένα κομμάτι hardware που δεν θα θέλατε ποτέ να σας δημιουργήσει bottleneck. Τα bottlenecks επεξεργαστή μπορεί να κάνουν το server να σέρνεται. Επίσης είναι τα πιο δύσκολα bottlenecks στο να τα εντοπίσετε. Σας παρέχονται μετρητές οι οποίοι σας πληροφορούν σχετικά με το πώς τρέχουν οι διάφορες εφαρμογές, πόσο χρόνο επεξεργαστή έχει στη διάθεσή της η κάθε μία, κ.λ.π. .

### Redirector

Ο Redirector είναι το σύστημα αρχείων που εκτρέπει τις αιτήσεις αρχείων σε servers δικτύου. Υψηλή κυκλοφορία στους redirectors μπορεί να προκαλέσει bottlenecks δικτύου. Ένας server που

συμπεριφέρεται ως server αρχείων μπορεί να παρουσιάσει υψηλότερο Bytes Total/sec counter από έναν server που απασχολεί περιορισμένο αριθμό χρηστών.

### Σύστημα

Το performance object του συστήματος έχει μια λίστα από counters που εφαρμόζονται σε όλα τα softwares και hardwares. Το αντικείμενο αυτό σκιαγραφεί μια γενική εικόνα του server. Μπορείτε να ελέγξετε λειτουργίες επιπέδου λειτουργικού συστήματος, για παράδειγμα με τον Context Switches/sec counter.

### Νήμα

Το νήμα είναι η βασική εκτελέσιμη οντότητα που μπορεί να εκτελεί εντολές στον επεξεργαστή. Μπορείτε να ελέγξετε δραστηριότητα επιπέδου λειτουργικού συστήματος και να δείτε έτσι πώς το λειτουργικό σύστημα επηρεάζει τη δραστηριότητα μιας εφαρμογής (ο έλεγχος όμως επικεντρώνεται στα νήματα που χρησιμοποιούνται).

### Active Server Pages

Αν τρέχετε ASP εφαρμογές στο server σας, οι counters που παρέχονται σας δίνουν μια εικόνα για την απόδοση των εφαρμογών αυτών. Μπορείτε έτσι να ελέγξετε, για παράδειγμα, τις αιτήσεις ανά δευτερόλεπτο με τον Request/sec counter. Οι counters αυτοί μπορούν να σας οδηγήσουν στη σωστή κατεύθυνση προκειμένου να πετύχετε βελτιστοποίηση της απόδοσης της εφαρμογής σας.

---

## Περιγραφή Σημαντικών Performance Counters

Ο παρακάτω πίνακας περιγράφει τους πιο συνηθισμένους counters που χρησιμοποιούνται προκειμένου να διαγνωστούν και να αντιμετωπιστούν πιθανά προβλήματα. Ωστόσο, μόνο τα βασικά περιλαμβάνονται για κάθε counter.

<u>Πρόβλημα</u>	<u>Counter</u>	<u>Περιγραφή</u>
<u>Bottleneck επεξεργαστή</u>	1. <i>System\Processor Queue Length</i> (όλα τα instances) 2. <i>Processor\Interrupts/sec</i> 3. <i>System\Conext Switches/sec</i>	Bottlenecks επεξεργαστή μπορούν να προκληθούν από πολλούς παράγοντες. Ωστόσο, το πρώτο βήμα είναι να επιβεβαιωθεί ότι υπάρχει ένα bottleneck επεξεργαστή. Αυτό μπορεί να γίνει μόνο ελέγχοντας τους κατάλληλους counters.

<u>Χρησιμοποίηση Επεξεργαστή</u>	<i>Processor\%Processor Time</i> (όλα τα instances)	Πολλοί διαχειριστές ελέγχουν μόνο τον <i>Processor\%Processor Time</i> όταν καθορίζουν αν υπάρχει bottleneck επεξεργαστή. Αυτός ο μετρητής είναι μόνο ένας καλός δείκτης της χρησιμοποίησης του επεξεργαστή.
<u>Bottleneck δίσκου</u>	<i>Physical Disk\Avg. Disk Queue Length</i> (όλα τα instances)	Καθώς οι εφαρμογές γίνονται όλο και πιο απαιτητικές σε δεδομένα, η χρησιμοποίηση των I/O του δίσκου πλήγεται περισσότερο από ποτέ. Αν δείτε υψηλό <i>Physical Disk\Avg. Disk Queue Length</i> και η χρησιμοποίηση του επεξεργαστή και της μνήμης είναι χαμηλή, τότε υπάρχει μεγάλη πιθανότητα το bottleneck να βρίσκεται στο δίσκο.
<u>Χρησιμοποίηση δίσκου</u>	1. <i>Physical Disk\Disk Reads/sec</i> 2. <i>Physical Disk\Disk Writes/sec</i>	Να ελέγχετε τη χρησιμοποίηση του δίσκου για να προλαμβάνετε τα bottlenecks. Αν βρείτε ότι η χρησιμοποίηση αυξάνεται σταθερά, αρχίστε να κάντε κινήσεις που θα αποτρέψουν ένα πιθανό bottleneck.
<u>Bottlenecks ή Leaks Μνήμης</u>	1. <i>Memory\Pages/sec</i> 2. <i>Memory\Page Reads/sec</i> 3. <i>Memory\TransactionFaults/sec</i> 4. <i>Memory\Pool Paged Bytes</i> 5. <i>Memory\Pool Nonpaged Bytes</i>	Μια διαρροή (leak) μνήμης συμβαίνει έπειτα από μια περίοδο χρόνου, αντίθετα από ένα bottleneck μνήμης, και γι'αυτό είναι πολύ δύσκολο να διαγνωσθούν.
<u>Χρησιμοποίηση Μνήμης</u>	1. <i>Memory\Available Bytes</i> 2. <i>Memory\Cache Bytes</i>	Όπως και στη χρησιμοποίηση δίσκου και επεξεργαστή, ελέγχοντας τη χρησιμοποίηση της μνήμης σας δίνεται ένα μέσο πρόληψης. Αν υπάρχει αυξημένη χρησιμοποίηση μνήμης, ελέγξτε κι άλλους μετρητές προκειμένου να καθορίσετε αν υπάρχει λάθος σε μια διεργασία ή αν φυσικά αίτια, όπως αυξημένος



φόρτος χρηστών, προκάλεσαν το πρόβλημα.

Throughput  
Δικτύου

1. *Network Interface\Bytes Total/sec*
2. *Network Interface\Packets/sec*
3. *Server\Bytes Total/sec*
4. *Server\Bytes Transmitted/sec*
5. *Server\Bytes Received/sec*

Οι counters αυτοί δείχνουν την αποδοτικότητα του δικτύου. Επιπλέον, σας βοηθούν να ανιχνεύσετε προβλήματα σχετικά με το δίκτυο. Για παράδειγμα, αν βλέπετε πολλά ξανασταλμένα πακέτα, ξέρετε ότι υπάρχει λόγος. Ο λόγος μπορεί να είναι ένα κακός πρασαρμογέας δικτύου, ένα χαλασμένο καλώδιο ή μια κακογραμμένη εφαρμογή. Το γεγονός ότι είστε σε θέση να γνωρίζετε ότι όντως υπάρχει πρόβλημα, είναι το πρώτο βήμα για να το διαγνώσετε.

Χρησιμοποίηση  
Δικτύου

*Network Segment\%Net Utilization* (απαιτείται ο Network Monitor driver)

Όπως και παραπάνω, το αντικείμενο αυτό σας βοηθάει να μετρήσετε την αποδοτικότητα του δικτύου. Ωστόσο, σε αυτήν την περίπτωση, εξετάζετε τη χωρητικότητα. Τι ποσοστό από τους διαθέσιμους πόρους δικτύου είναι σε χρήση; Μπορείτε να χρησιμοποιήσετε αυτήν την πληροφορία για να σχεδιάσετε αναβαθμίσεις για το server σας. Όταν αρχίσετε να βλέπετε το *Network Segment\%Net Utilization* counter να ανεβαίνει, ξέρετε ότι περισσότερο από το εύρος ζώνης του δικτύου είναι σε χρήση και είναι ώρα να προσθέσετε πρόσθετο εύρος ζώνης προκειμένου το συστημά σας να υποστηρίξει μελλοντικές απαιτήσεις.

Active Server  
Pages

1. *Active Server Pages\Request Wait Time*
2. *Active Server Pages\Requests Queued*
3. *Active Server*

Αυτοί οι counters δείχνουν τη σχετική απόδοση του IIS όταν δουλεύει με ASPs. Γενικά, δεν θέλετε να βλέπετε αιτήσεις να βρίσκονται στην ουρά και, αν αυτές μπαίνουν

*Pages\Requests Rejected*  
4. *Active Server*  
*Pages\Requests/sec*

στην ουρά, ο χρόνος αναμονής πρέπει να είναι πολύ χαμηλός. Επίσης δεν θέλετε να βλέπετε αιτήσεις να απορρίπτονται επειδή δεν υπάρχουν διαθέσιμοι πόροι. Να κρατάτε αυτά τα προβλήματα σχετικά με τις αιτήσεις που υφίστανται επεξεργασία το δευτερόλεπτο. Μπορεί να παρατηρήσετε κάποια αντίφαση κάτω από φόρτο αιχμής (peak load). Για να λύσετε αυτά τα θέματα θα πρέπει να αναβαθμίσετε ή να προσθέσετε επεξεργαστές.

---

## System Monitor Toolbar

Το System Monitor Toolbar βρίσκεται πάνω ακριβώς από το σύστημα συντεταγμένων (όπου εμφανίζονται τα διαγράμματα των counters) στην κονσόλα MMC. Μπορείτε να πάρετε μια σύντομη περιγραφή του τι κάνει το κάθε εικονίδιο τοποθετώντας τον κέρσορα πάνω στο εικονίδιο. Οι διαθέσιμες επιλογές εντολών (από τα αριστερά προς τα δεξιά) φαίνονται στον παρακάτω πίνακα.

---

<b>Εντολή</b>	<b>Περιγραφή</b>
<i>New Counter Set</i>	Καθαρίζει την τρέχουσα οθόνη, ώστε να μπορέσετε να προσθέσετε μία νέα λίστα μετρητών.
<i>Clear Display</i>	Καθαρίζει την οθόνη, αλλά κρατάει την τρέχουσα λίστα μετρητών και αμέσως επανεκκινεί την παρακολούθηση των καταχωρημένων μετρητών.
<i>View Current Activity</i>	Αν βρίσκεστε στην log view, μπορείτε να αλλάξετε πίσω στην παρακολούθηση των δεδομένων σε πραγματικό χρόνο ή στο διάστημα που έχετε θέσει.
<i>View Log File Data</i>	Αν τα δεδομένα καταγράφονται, μπορείτε να αλλάξετε από την επισκόπηση της τρέχουσας δραστηριότητας στα δεδομένα καταγραφής.

<i>View Chart</i>	Αλλάζει στην chart view, δηλαδή στην επισκόπηση των διαγραμμάτων.
<i>View Histogram</i>	Αλλάζει στην επισκόπηση των ιστογραμμάτων.
<i>View Report</i>	Αλλάζει στην επισκόπηση των καταγραμμένων δεδομένων.
<i>Add</i>	Προκειμένου να αρχίσετε να προσθέτετε μετρητές σε ένα νέο ή υπάρχον διάγραμμα ή αναφορά, κάντε κλικ στο εικονίδιο Add για να ανοίξετε το παράθυρο διαλόγου Add Counters.
<i>Delete</i>	Διαγράφει τους επιλεγμένους counters.
<i>Highlight</i>	Κάνει τους επιλεγμένους counters να ξεχωρίζουν από τους υπόλοιπους.
<i>Copy Properties</i>	Σας επιτρέπει να επαναχρησιμοποιήσετε τις τρέχουσες ρυθμίσεις για ένα νέο System Monitor session.
<i>Paste Counter List</i>	Σας επιτρέπει να επικολλήσετε τους counters που χρησιμοποιήσατε σε ένα προηγούμενο διάγραμμα ή αναφορά, βγάζοντάς σας από τον κόπο να προσθέσετε ξανά τους ίδιους counters.
<i>Properties</i>	Ανοίγει το παράθυρο διαλόγου των ιδιοτήτων του System Monitor.
<i>Freeze Display</i>	Παγώνει την οθόνη, ενώ τα δεδομένα εξακολουθούν να καταγράφονται στο παρασκήνιο.
<i>Update Data</i>	Αν παρακολουθείτε δεδομένα και έχετε θέσει manual update, κάντε κλικ στο εικονίδιο αυτό για να ενημερώσετε το διάγραμμα ή την αναφορά.
<i>Help</i>	Το online help του System Monitor.

---

## Chart View

Η chart View είναι μια γραφική απεικόνιση του System Monitor. Αποτελείται από το graph mode και το histogram mode. Όλες οι άλλες απεικονίσεις (report, alert και log) παράγουν έξοδο κειμένου είτε σε μορφή πραγματικού χρόνου είτε προσαρτημένο σε συγκεκριμένο αρχείο καταγραφής.

Η chart view σας προσφέρει τη δυνατότητα να συλλέγετε δεδομένα σε πραγματικό χρόνο και στα δύο modes. Αν θέλετε να αναλύσετε παράγοντες απόδοσης στο χρόνο, τότε κοιτάζετε τα δεδομένα στο graph mode. Αν πάλι χρειάζεστε να παρακολουθήσετε πολλαπλά στιγμιότυπα ενός δοθέντος μετρητή την ίδια στιγμή, τότε κοιτάζετε το histogram mode.

### Graph Mode

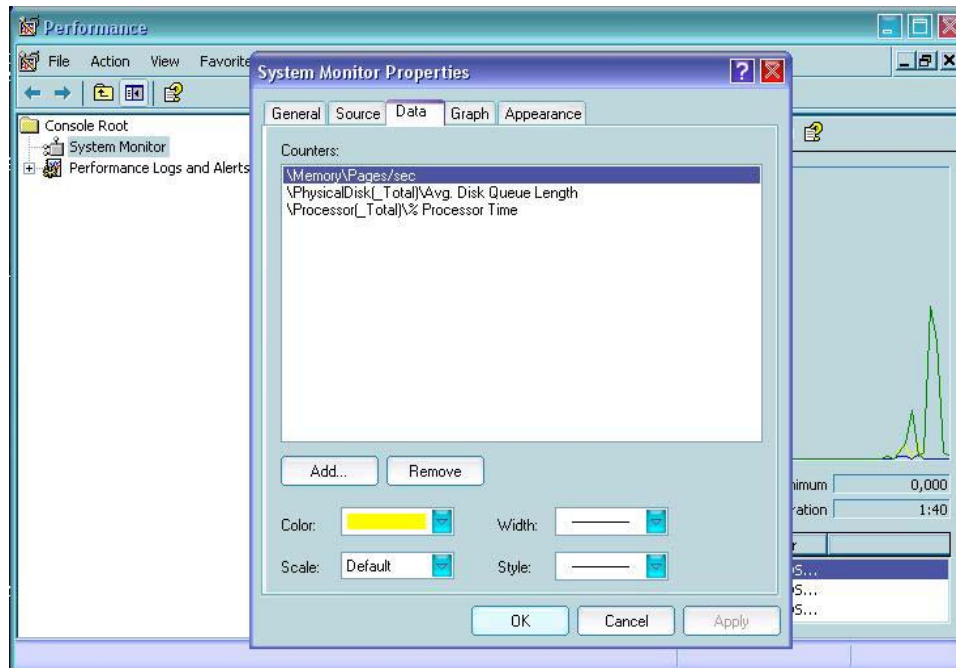
Όταν ξεκινάτε την παρακολούθηση της απόδοσης με το System Monitor, σας παρουσιάζεται ένα κενό γράφημα. Το graph mode είναι επιλεγμένο ως default, ενώ μπορείτε να το αλλάξετε από τα εικονίδια των επιλογών εντολών του toolbar που εξηγήθηκαν παραπάνω. Για να ξεκινήσετε να παρακολουθείτε έναν counter στο graph mode, κάντε κλικ στο σύμβολο '+' στο toolbar, ώστε να προσθέσετε τον μετρητή που επιθυμείτε. Καθώς ο μετρητής ξεκινά να παρακολουθεί το σύστημα, προσέξτε την κατακόρυφη γραμμή στη μέση του γραφήματος. Αυτή είναι η γραμμή χρόνου. Η γραμμή χρόνου είναι πάντα κόκκινη και ένα διάστημα πέρα από την τελευταία παρατηρημένη τιμή. Η γραμμή χρόνου κινείται προς τα δεξιά όταν η οθόνη ενημερώνεται σύμφωνα με το διάστημα που έχετε θέσει για το γράφημα. Η γραμμή αυτή αναδιπλώνεται προς τα αριστερά όταν φτάσει στο τέλος του ορίου του γραφήματος.

Κάνοντας δεξί κλικ μέσα στο γράφημα, εμφανίζεται ένα μενού και από εκεί μπορείτε να επιλέξετε την επιλογή 'Properties'. Στη συνέχεια θα περιγράψουμε τις ιδιότητες και τις ρυθμίσεις που μπορείτε να κάνετε σχετικά με το γράφημα.

General Tab: Αρχικά μπορείτε να καθορίσετε την απεικόνιση που θέλετε (graph, histogram, report). Έπειτα μπορείτε να αποφασίσετε αν θα εμφανίζονται κάποια στοιχεία, όπως η λεζάντα, το toolbar και η μπάρα τιμών. Ο καθορισμός της εμφάνισης ως 3D ή flat γίνεται επίσης από εδώ. Τέλος μπορείτε να καθορίσετε το χρονικό διάστημα στο οποίο θα γίνεται η δειγματοληψία των δεδομένων από τους counters.

Source: Από εδώ μπορείτε να επιλέξετε μια πηγή από όπου θα εμφανιστούν τα δεδομένα σας στο γράφημα. Έτσι μπορείτε να επιλέξετε τα δεδομένα σας να

εμφανίζονται σε πραγματικό χρόνο, ή να φορτώσετε δεδομένα από παλαιότερες καταγραφές από log files ή βάσεις δεδομένων.



**Data:** Εδώ φαίνονται οι counters που έχετε επιλέξει. Μπορείτε να προσθέσετε άλλους counters και να αλλάξετε τον τρόπο (χρώμα, πλάτος, στυλ, κλίμακα) με τον οποίο εμφανίζονται στο γράφημα.

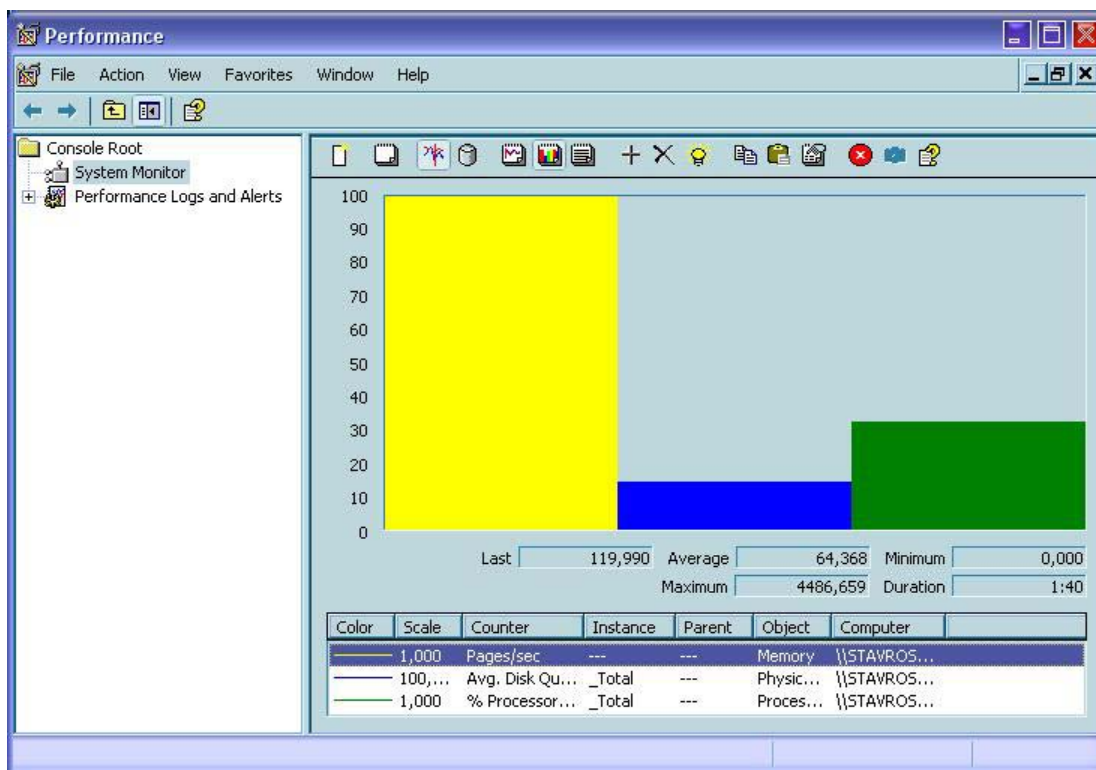
**Graph:** Μπορείτε να εισάγετε έναν τίτλο για το γράφημα και μια ετικέτα για τον κατακόρυφο άξονα. Μπορείτε επίσης να κρύψετε τους αριθμούς από τον κατακόρυφο άξονα ή να εισάγετε πλέγμα. Τέλος εδώ γίνεται ο καθορισμός της κλίμακας των τιμών που θα παίρνει ο κατακόρυφος άξονας.

**Appearance:** Εδώ μπορείτε να αλλάξετε το χρώμα κάποιων στοιχείων (όπως το background), καθώς και τη γραμματοσειρά του κειμένου.

### **Histogram Mode**

Το ιστόγραμμα είναι η άλλη μέθοδος γραφικής απεικόνισης των μετρητών. Για να αλλάξετε στο histogram mode, κάντε κλικ στο εικονίδιο 'Histogram' στο toolbar ή επιλέξτε 'Histogram' στο General Tab του παραθύρου διαλόγου Properties. Αν παρακολουθείτε πολλαπλά στιγμιότυπα του ίδιου μετρητή ταυτόχρονα, είναι προτιμότερο να βλέπετε αυτό το mode. Παρακολουθώντας στο ιστόγραμμα την απόδοση των επεξεργαστών που είναι εγκατεστημένοι σε έναν server, σας επιτρέπεται

να βλέπετε σε πραγματικό χρόνο την απόδοσή τους, ‘σώμα με σώμα’. Αν έχετε το φόβο ότι το hardware σε έναν server με δύο επεξεργαστές υποχρησιμοποιείται, μπορείτε να επιβεβαιώσετε αυτόν το φόβο παρακολουθώντας τη δραστηριότητα και των δύο επεξεργαστών ταυτόχρονα. Στο graph mode δεν μπορείτε να έχετε μια σίγουρη απάντηση, ιδίως όταν παρακολουθείτε δεδομένα σε πραγματικό χρόνο.

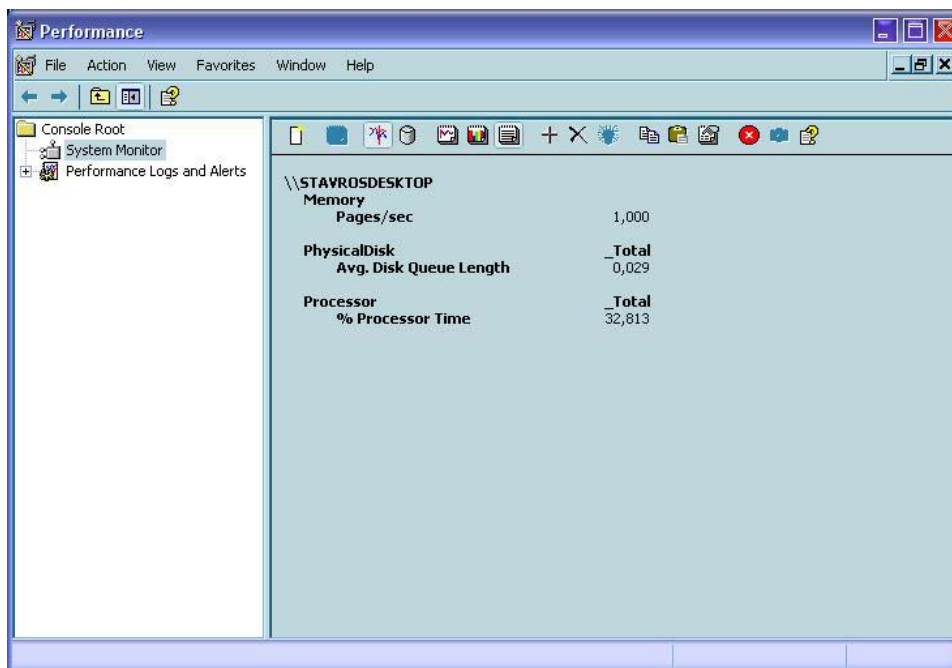


## Report View

Υπάρχουν φορές, όπως όταν πρέπει να παρακολουθήσετε πολλούς counters, όπου οι γραφικές απεικονίσεις τείνουν να δουλεύουν εναντίον σας παρά προς όφελός σας. Για παράδειγμα όταν παρακολουθείτε 11 counters που όχι μόνο είναι εξαιρετικά ενεργοί, αλλά λειτουργούν και σε διαφορετικές κλίμακες.

Στο report view, δεν υπάρχει ο παράγοντας της κλίμακας επειδή παρατηρείτε πάντα ολόκληρη την τιμή. Όταν παρακολουθείτε πολλούς counters στα graph και histogram modes, τις περισσότερες φορές θέλετε να παρατηρείτε κάθε counter ατομικά, παρά να βλέπετε συνολικά το γράφημα για να διαπιστώσετε κάποια τάση ή πρόβλημα.

Για να ανοίξετε το report view, κάντε κλικ στο εικονίδιο ‘View Report’ στο toolbar ή επιλέξτε ‘Report’ στο General Tab του παραθύρου διαλόγου των Properties. Μπορείτε να αρχίσετε να προσθέτετε μετρητές στην αναφορά όπως ακριβώς και στο γράφημα. Καθώς προσθέτετε μετρητές στην αναφορά, αυτοί τοποθετούνται στο κάτω μέρος της.



Το καλύτερο θα ήταν να συλλέγετε τα δεδομένα απόδοσης σε ένα log file, ή να εξάγετε τα δεδομένα σε μια εξωτερική εφαρμογή. Παρακάτω, θα δείτε πώς γίνεται η συλλογή των δεδομένων.

---

## Performance Logs and Alert Views

Τα Performance Logs and Alert Views είναι δύο απεικονίσεις οι οποίες μπορεί να είναι οι πιο χρήσιμες από όλες τις υπόλοιπες απεικονίσεις. Αυτό δεν σημαίνει ότι πρέπει να χρησιμοποιούνται πάντα και ότι δεν συγκρίνονται με καμιά άλλη απεικόνιση. Ποκειμένου να είναι αποτελεσματικές, θα πρέπει να χρησιμοποιηθούν στη σωστή κατάσταση. Ο μόνος τρόπος για να καταλάβετε ποια είναι η σωστή κατάσταση για την υλοποίηση αυτών των εργαλείων, είναι να καταλάβετε πλήρως τις δυνατότητες και τους περιορισμούς τους.

Οι δύο αυτές απεικονίσεις σχεδιάστηκαν για σοβαρή συλλογή δεδομένων και για εκείνες τις στιγμές όταν δεν μπορείτε απευθείας να εντοπίσετε το πρόβλημα με το chart ή το report view. Όταν θέλετε να συλλέξετε δεδομένα απόδοσης του server μέσα σε μέρες ή ακόμη και μήνες, η καταγραφή είναι ο μόνος τρόπος. Διαφορετικά, σε μεγάλες περιόδους σάς μένουν γραφήματα που απορροφούν πολύτιμους πόρους server και είναι κουραστικό να αναλυθούν.

Υπάρχουν δύο τρόποι καταγραφής που μπορείτε να παράγετε κάτω από τα Performance Logs and Alerts, τους counter και trace. Δημιουργείτε ένα counter log προσθέτοντας μετρητές στο log, όμοια με τις προηγούμενες απεικονίσεις, θέτοντας ένα διάστημα χρόνου και αρχίζοντας την καταγραφή. Το log file αποτελείται από δεδομένα μετρητών που συλλέχθηκαν για κάθε μετρητή που καθορίσατε, μέσα στο διάστημα που θέσατε. Τα trace logs γράφουν δεδομένα συγκεκριμένων δραστηριοτήτων, όπως λειτουργία I/O δίσκου ή δικτύου TCP/IP. Όταν συμβαίνει ένα τέτοιο γεγονός, ο παροχέας στέλνει τα δεδομένα στα Performance Logs and Alerts.

Τα Performance Logs and Alerts σας ειτρéπουν να βλέπετε τα δεδομένα κατά τη συλλογή ή αφού τελειώσει η συλλογή. Όταν βλέπετε τα δεδομένα κατά τη συλλογή δεν σημαίνει ότι αυτά σταματούν να καταγράφονται. Η υπηρεσία συνεχίζει να γράφει στο log file έτσι ώστε να μη χάσετε κάποια κρίσιμη τάση που μπορεί να συμβεί στο server.

### **Counter Logs**

Το Performance Log παρέχει ένα δείγμα counter log που ονομάζεται System Overview. Αυτό το counter log αποτελείται από μετρητές που σας δίνουν μια γενική εποπτεία του server σας. Κάντε δεξί κλικ στο log file και επιλέξτε Properties. Αυτό ανοίγει το παράθυρο διαλόγου των System Overview Properties. Προσέξτε ότι κάτω από τους Counters στο General tab, οι μετρητές που είναι καταχωρημένοι εκεί είναι μέρος αυτού του συγκεκριμένου log file. Για να προσθέσετε νέους μετρητές, κάντε κλικ στο Add και θα ανοίξει το παράθυρο διαλόγου Select Counters. Από εδώ, μπορείτε να προσθέσετε μετρητές στο log file όπως ακριβώς στο chart και report view. Για να αφαιρέσετε μετρητές από το log, κάντε κλικ στο Remove στο παράθυρο διαλόγου System Overview Properties.

Για να δημιουργήσετε ένα νέο counter log, ακολουθήστε τα παρακάτω βήματα:

1. Κάντε δεξί κλικ οπουδήποτε στην πλευρά όπου αναγράφονται οι λεπτομέρειες των Counter Logs και επιλέξτε New Log Settings.
2. Στο παράθυρο διαλόγου New Log Settings, πληκτρολογήστε το όνομα του νέου counter log και κάντε κλικ στο OK.
3. Εμφανίζεται το παράθυρο διαλόγου των ιδιοτήτων. Κάντε κλικ στο Add για να αρχίσετε να προσθέτετε μετρητές στο log file.
4. Αφού προσθέσετε τους επιθυμητούς μετρητές, καθορίστε ένα διάστημα ενημέρωσης.
5. Κάνοντας κλικ στο Log Files tab, σας επιτρέπεται να αλλάξετε το όνομα του log file πληκτρολογώντας ένα νέο όνομα στο πεδίο File Name. Μπορείτε επίσης να καθορίσετε μια συγκεκριμένη κατάληξη για το αρχείο, που να υποδηλώνει δευτερεύουσα σημασία για αυτό. Οι επιλογές περιλαμβάνουν χχχχμμ (μια χρονολογία τεσσάρων ψηφίων και ένα μήνα δύο ψηφίων. Κάνοντας τα αρχεία να



τελειώνουν με μια σφραγίδα ημερομηνίας ή ώρας σας βοηθάει στο να ξεχωρίζετε τα log files μεταξύ τους.

6. Μπορείτε να επιλέξετε τι τύπο log file να δημιουργήσετε – δυαδικό, κειμένου .csv, κειμένου .tsv, ή δυαδικό κυκλικό αρχείο. Χρησιμοποιήστε ένα δυαδικό αρχείο αν θέλετε να είστε σε θέση να καταγράφετε στιγμιότυπα που είναι ασυνεχή, δηλαδή που σταματάνε και συνεχίζουν από τη στιγμή που το log έχει αρχίσει να τρέχει. Ένα δυαδικό κυκλικό αρχείο συνεχίζει να γράφει στο ίδιο log file, γράφοντας επάνω σε προηγούμενα δημιουργημένες εγγραφές. Μπορείτε να πληκτρολογήσετε και ένα σχόλιο για το αρχείο, που είναι ακόμη ένας τρόπος για να ξεχωρίζετε τα log files μεταξύ τους.
7. Επιλέξτε ένα μέγεθος για το log file αν θέλετε να περιορίσετε το μέγεθος του αρχείου σε μια συγκεκριμένη ποσότητα σε KB. Θυμηθείτε, όσο περισσότερους μετρητές επιλέγετε, τόσο συντομότερο είναι το διάστημα ενημέρωσης. Επίσης, όσο περισσότερα δεδομένα καταγράφετε, τόσο μεγαλύτερα είναι τα log files. Αν έχετε περιορισμένο χώρο στο δίσκο, περιορίστε το μέγεθος του αρχείου. Κάντε κλικ στο Schedule tab για να θέσετε ένα χρόνο αρχής και τέλους.
8. Το Schedule tab σας επιτρέπει να καθορίσετε ένα χρόνο αρχής, ή μπορείτε να επιλέξετε χειροκίνητα να ξεκινήσει το log file από το μενού συντόμευσης. Μπορείτε να σταματήσετε τη συλλογή δεδομένων σε ένα συγκεκριμένο χρόνο ή όταν γεμίσει το log. Μπορείτε επίσης να σταματήσετε τη συλλογή δεδομένων έπειτα από ένα διάστημα χρόνου, όπως μια ώρα, μια μέρα ή ένα δευτερόλεπτο. Αφού κλείσει το log, μπορείτε να διαλέξετε να αρχίσει ένα νέο log file ή να τρέξετε μια εντολή, όπως μια εντολή να αντιγράψετε το log file σε μια συγκεκριμένη τοποθεσία. Κάντε κλικ στο OK όταν τελειώσετε.

Αν επιλέξετε να ξεκινήσει η συλλογή δεδομένων χειροκίνητα, μπορείτε να κάνετε δεξί κλικ πάνω στο log file και μετά κλικ στο Start από το μενού συντομεύσεων. Διαφορετικά, το log file ξεκινάει τη συλλογή δεδομένων μόλις κλείσετε το παράθυρο των ιδιοτήτων.

### **Trace Logs**

Όπως αναφέρθηκε προηγουμένως, τα trace logs γράφουν σε ένα log file όταν ένας παροχέας στέλνει δεδομένα στην υπηρεσία Performance Logs and Alerts.

Τα trace logs δημιουργούνται με τον ίδιο τρόπο όπως τα counter logs. Ωστόσο, υπάρχουν κάποιες διαφορές που πρέπει να προσέξετε στην οθόνη ιδιοτήτων για το trace log. Πρώτον, πρέπει να καθορίσετε το γεγονός που θα καταγραφεί από τον παροχέα συστήματος. Δεύτερον, ο τύπος του αρχείου που μπορείτε να δημιουργήσετε είναι ένα .etl αρχείο. Απαιτείται ένα εργαλείο ανάλυσης (parse tool) για να διερμηνεύσει την έξοδο του trace log. Μπορείτε επίσης να καθορίσετε την ποσότητα της μνήμης που

θέλετε να αναθέσετε έτσι ώστε η υπηρεσία καταγραφής να μπορεί να αποθηκεύει προσωρινά τα δεδομένα του trace log σε buffers μνήμης πριν τα μεταφέρει στο log file.

### **Βλέποντας δεδομένα από ένα καθορισμένο χρονικό διάστημα**

Το Performance tool σας επιτρέπει να χειρίζεστε διαστήματα χρόνου μέσα στα logs, έτσι ώστε να βλέπετε τις διακυμάνσεις των δεδομένων που συμβαίνουν ανάμεσα στα διαστήματα ενημέρωσης που καθορίζετε.

Για να δείτε δεδομένα από ένα καθορισμένο χρονικό διάστημα, ακολουθήστε τα παρακάτω βήματα:

1. Ανοίξτε την κονσόλα Performance.
2. Κάντε δεξιά κλικ στο χώρο των λεπτομερειών του System Monitor και κάντε κλικ στο Properties.
3. Κάντε κλικ στο Source tab.
4. Κάτω από το Data Source, καθορίστε την τοποθεσία για το log file που θέλετε να χρησιμοποιήσετε.
5. Κάντε κλικ στο Time Range. Σύρετε (drag) την αριστερή ή τη δεξιά πλευρά της μπάρας για να μικρύνετε την περίοδο χρόνου που θέλετε να δείτε. Κάντε κλικ στο OK όταν τελειώσετε.
6. Τώρα πρέπει να προσθέσετε τους μετρητές που έχουν υποστεί παρακολούθηση στο γράφημά σας ή στο report view για να δείτε την έξοδο. Έχετε την επιλογή της πρόσθεσης ενός μοναδικού μετρητή από το log ή όλων μαζί. Μπορείτε να αλλάξετε τις απεικονίσεις ώστε να πάρετε διαφορετικές απόψεις των δεδομένων.

**(Σημείωση:** Όταν βλέπετε τα καταγραμμένα δεδομένα στο report view, βλέπετε μόνο το μέσο όρο του κάθε μετρητή)

### **Alert View**

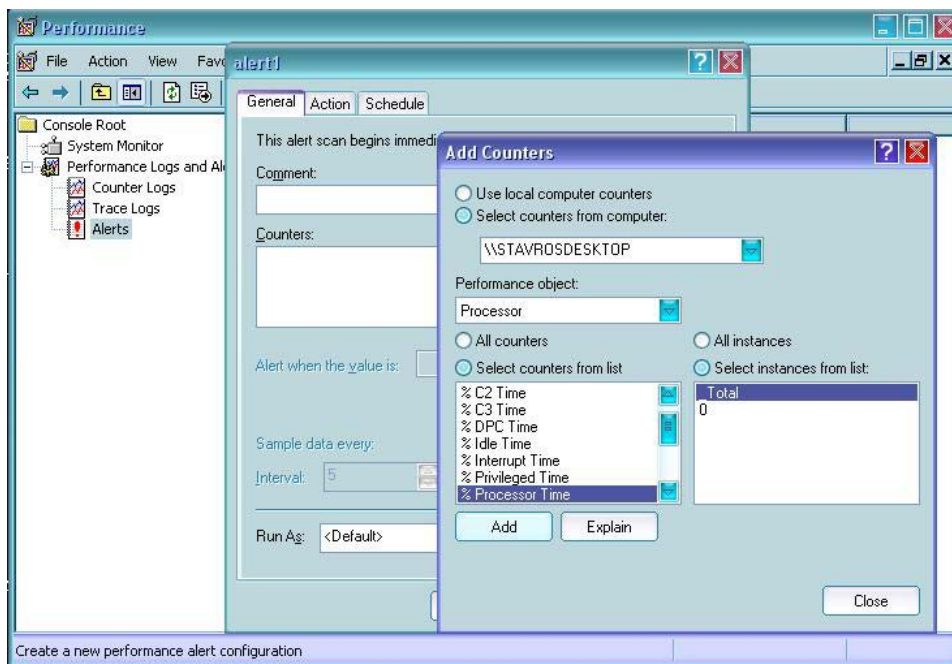
Αυτή η απεικόνιση, όπως και οι υπόλοιπες, σας επιτρέπει να παρακολουθήσετε μετρητές για πολλαπλά αντικείμενα, αλλά έχει και μερικά μοναδικά χαρακτηριστικά. Η Alert View χρειάζεται από εσάς να καθορίσετε μια τιμή κατωφλίου για κάθε μετρητή που θέλετε να παρακολουθήσετε. Με αυτήν την τιμή καθορισμένη, η απεικόνιση μπορεί να σας στείλει μια προειδοποίηση όταν ο μετρητής πλησιάσει αυτήν την τιμή κατωφλίου. Μια προειδοποίηση μπορεί να προκληθεί πριν ο μετρητής πλησιάσει το κατώφλι ή αφού το περάσει.

Με αυτόν τον τρόπο, μπορείτε να κάθεστε αναπαυτικά χωρίς να χρειάζεται να έχετε στραμμένη την προσοχή σας σε πολλούς μετρητές την ίδια στιγμή, αφού γνωρίζετε ότι θα ειδοποιηθείτε όταν ένας μετρητής συμπεριφερθεί με συγκεκριμένο τρόπο. Για μια

πλειοψηφία μετρητών, θέλετε να ειδοποιηθείτε αν ο μετρητής γίνει μεγαλύτερος από μια τιμή. Για άλλους, θέλετε να ειδοποιηθείτε αν η τιμή πέσει κάτω από μια συγκεκριμένη τιμή.

Η πρόσθεση μετρητών στην απεικόνιση γίνεται ακολουθώντας τα παρακάτω βήματα:

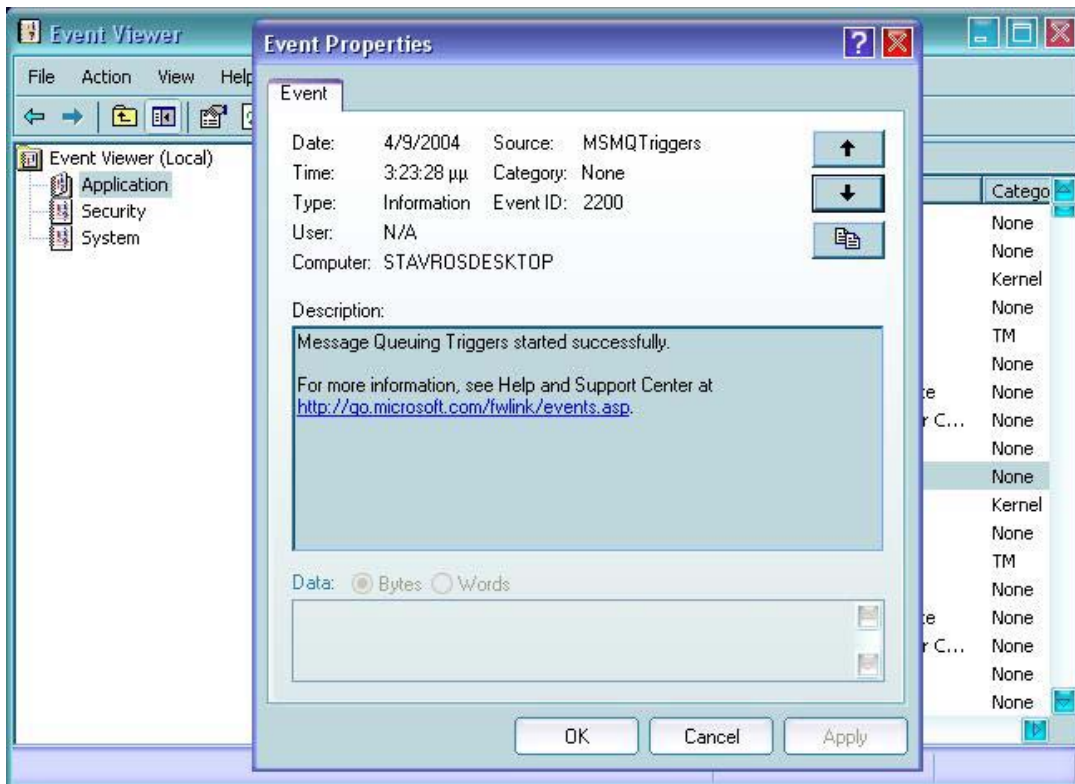
1. Ανοίξτε την κονσόλα Performance.
2. Κάντε διπλό κλικ στο Performance Alerts and Logs και δεξί κλικ στο Alerts.
3. Από το drop-down μενού, επιλέξτε New Alert Settings
4. Πληκτρολογήστε ένα όνομα για την προειδοποίηση και κάντε κλικ στο OK.
5. Στο παράθυρο διαλόγου Properties, πληκτρολογήστε ένα σχόλιο που θα περιγράφει την προειδοποίηση και αρχίστε να προσθέτετε μετρητές προς παρακολούθηση. Αν θέλετε η προειδοποίηση να δημιουργείται όταν ο μετρητής είναι πάνω από μια τιμή κατωφλίου, επιλέξτε Over στο drop-down box Alert When The Value Is. Διαφορετικά, επιλέξτε Under ώστε η προειδοποίηση να δημιουργείται όταν η τιμή πέσει κάτω από την τιμή κατωφλίου.



6. Για να καθορίσετε ένα διάστημα ενημέρωσης, επιλέξτε ένα διάστημα που ταιριάζει καλύτερα για τους μετρητές που μπορεί να παρακολουθείτε. Αν θέσετε μια περίοδο που είναι πολύ μεγάλη, μπορεί να χάσετε δραστηριότητα που έχετε θέσει να ελέγχει η προειδοποίηση.

7. Κάντε κλικ στο Action tab για να καθορίσετε τι ενέργεια θα κάνει η υπηρεσία Performance όταν δημιουργείται η προειδοποίηση. Υπάρχουν τέσσερις πιθανές ενέργειες που μπορεί να κάνει η κονσόλα Performance:

- Καταγραφή μιας εγγραφής στο event log της εφαρμογής:  
Η κονσόλα Performance γράφει στο Application log του εργαλείου Event Viewer που βρίσκεται στα Administrative tools.



- Αποστολή μηνύματος μέσω δικτύου σε:  
Μπορείτε να ηλεκτρολογήσετε το όνομα ενός υπολογιστή στο δίκτυό σας στον οποίο θα σταλεί το μήνυμα προειδοποίησης.
- Εκκίνηση καταγραφής δεδομένων απόδοσης:  
Μπορείτε να κάνετε το σύστημα να αρχίσει να γράφει σε ένα προκαθορισμένο log όταν δημιουργηθεί η προειδοποίηση.
- Εκτέλεση προγράμματος:  
Διαλέγοντας αυτήν την επιλογή, σας επιτρέπεται να τρέξετε ένα πρόγραμμα στο server. Μπορείτε επίσης να περάσετε ορίσματα στο πρόγραμμα που τρέχει ως απόκριση προς την προειδοποίηση.

Το τελευταίο tab είναι το Schedule tab. Από εδώ, μπορείτε να καθορίσετε το χρόνο έναρξης και λήξης για την προειδοποίηση.

Αν η προειδοποίηση δημιουργήθηκε ώστε να τρέχει αμέσως, το εικονίδιο που αναπαριστά την προειδοποίηση γίνεται από κόκκινο πράσινο, συμβολίζοντας έτσι ότι τρέχει. Αν αυτή ρυθμίστηκε ώστε να τρέχει χειροκίνητα, ή σε κάποια στιγμή αργότερα, έχει κόκκινο χρώμα. Μπορείτε να εκκινήσετε την προειδοποίηση κάνοντας δεξί κλικ πάνω στο εικονίδιο και επιλέγοντας Start. Με τον ίδιο τρόπο μπορείτε να σταματήσετε την προειδοποίηση επιλέγοντας Stop, ή να τη διαγράψετε επιλέγοντας Delete.

---

## Εξάγοντας τα δεδομένα απόδοσης

Το System Monitor σας επιτρέπει να εξάγετε δεδομένα έτσι ώστε δημοφιλή προγράμματα (Excel, Microsoft Notepad και γενικά οποιαδήποτε εφαρμογή μπορεί να εισάγει tab-delimited ή comma-delimited ASCII αρχεία) να μπορούν να τα διαχειριστούν. Το System Monitor σας επιτρέπει επίσης να εξάγετε δεδομένα που συλλέχθηκαν είτε με την chart είτε με τη report view σε .htm αρχείο. Χρησιμοποιώντας τον browser σας, βλέπετε τα δεδομένα σε απεικονίσεις graph, histogram ή report. Όταν ανοίγετε το .htm αρχείο με τον browser, το System Monitor toolbar εμφανίζεται στην κορυφή της οθόνης. Χρησιμοποιώντας το toolbar μπορείτε να προσθέσετε κι άλλους counters για παρακολούθηση, να αλλάξετε τις ιδιότητες του System Monitor ή να αλλάξετε τις απεικονίσεις κάτω από τις οποίες παρακολουθείτε τα δεδομένα.

Για να εξάγετε τα δεδομένα είτε από απεικόνιση chart είτε από report, ακολουθήστε τα παρακάτω βήματα:

1. Κάντε δεξί κλικ οπουδήποτε στην απεικόνιση chart ή report και επιλέξτε Save As...
2. Το αρχείο αποθηκεύεται σε .htm από default. Μπορείτε να το αλλάξετε σε format .tsv επιλέγοντας Report από την drop-down λίστα στο Save As.
3. Αφού επιλέξετε το format του αρχείου, πληκτρολογήστε το όνομα του αρχείου και κάντε κλικ στο Save.

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 5

**ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ**

Τα προηγούμενα κεφάλαια μάς παρέχουν τις θεωρητικές γνώσεις που χρειαζόμαστε για να περάσουμε, πλέον, σε πιο πρακτικούς τομείς. Στο κεφάλαιο αυτό θα παρουσιάσουμε τα τεχνικά χαρακτηριστικά του συστήματος που χρησιμοποιήσαμε στα πειράματα που ακολουθούν στο δεύτερο μέρος της εργασίας αυτής, ώστε να βελτιώσουμε την απόδοση του server μας.

---

## Τεχνικά χαρακτηριστικά υπολογιστών

Client	Server
Intel Pentium 4 2.53GHz	Intel Pentium M 1.5GHz
512MB DDR	512MB DDR SDRAM
HDD 80GB 7200rps	HDD 60GB 7200rps
GeForce 4 MX440 64MB	ATI Radeon 9700 64MB
Broadcom 440x 10/100 Integrated Controller	VIA VT6105 Rhine III Fast Ethernet

---

### Software

Microsoft Windows XP Professional  
Microsoft Internet Information Services 5.0  
IIS 6.0 Resource Kit  
WebServer Stress Tool 6 (Enterprise Edition)

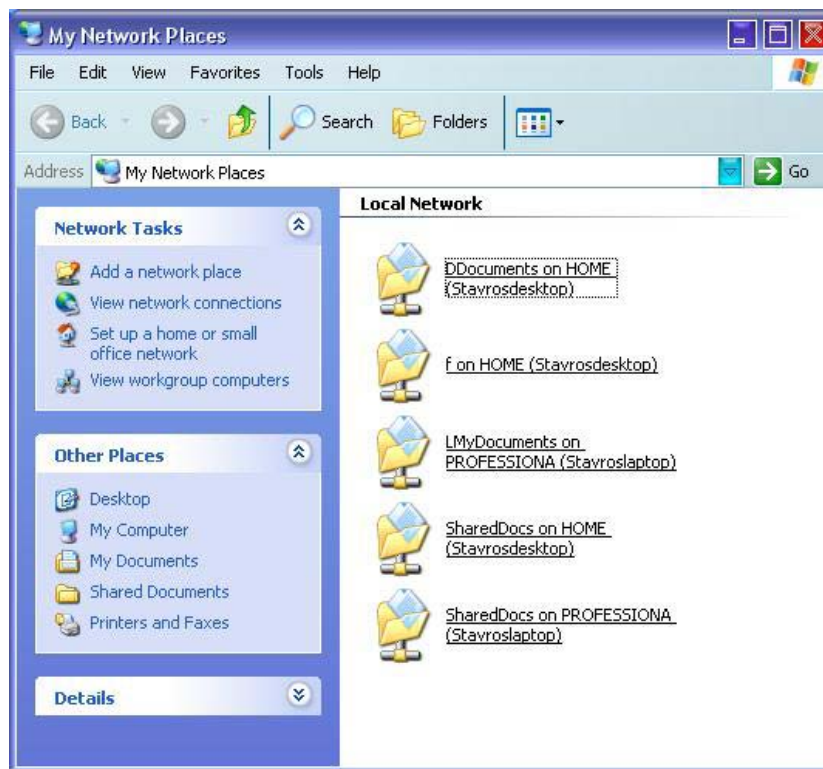
---

## Δημιουργία LAN

Προαπαιτούμενα: Κάρτες δικτύου εγκατεστημένες και στους δύο υπολογιστές, σύνδεση των δύο υπολογιστών μέσω ανεστραμμένου (cross-connect) καλωδίου.

Για να δημιουργήσουμε ένα LAN μεταξύ του client και του server ακολουθήσαμε τα παρακάτω βήματα:

1. Κάνουμε κλικ στο *Start* → *My Network Places* στον πρώτο υπολογιστή, π.χ στον client.
2. Από το πλαίσιο *Network Tasks* στα αριστερά του παραθύρου επιλέγουμε 'Set up a home or small office network'.

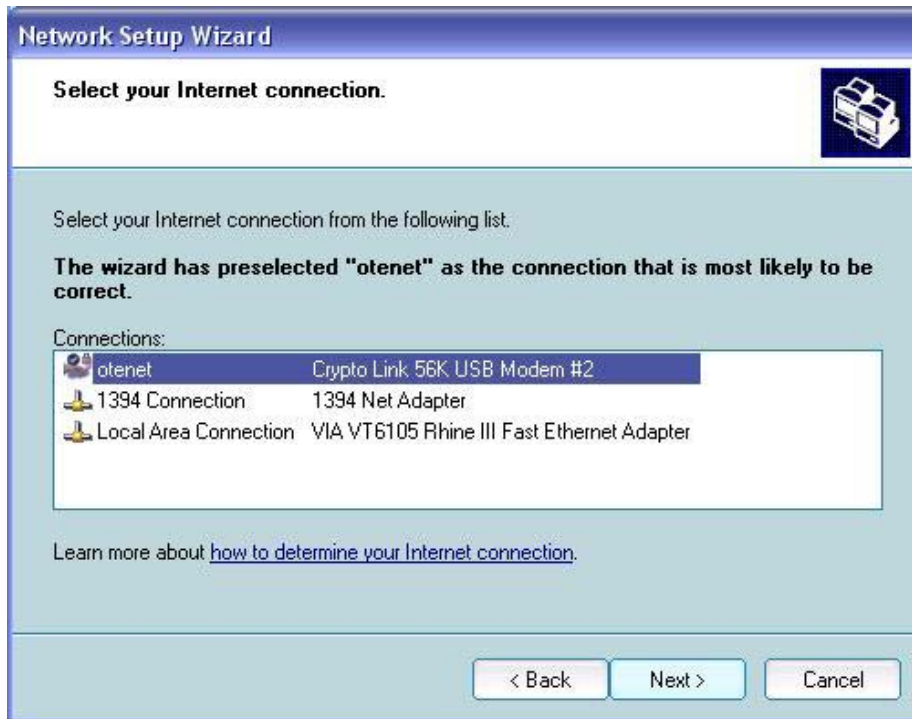


3. Ανοίγει ο Network Setup Wizard και πατάμε δύο φορές το Next.
4. Επιλέγουμε το πρώτο radio button αν ο συγκεκριμένος υπολογιστής συνδέεται απευθείας στο Internet. Όταν θα ακολουθήσουμε αυτά τα βήματα και για το δεύτερο υπολογιστή (δηλαδή το server στην προκειμένη περίπτωση) θα επιλέξουμε το δεύτερο radio button ώστε να μπορεί να συνδέεται στο Internet μέσω του client. Πατάμε Next.

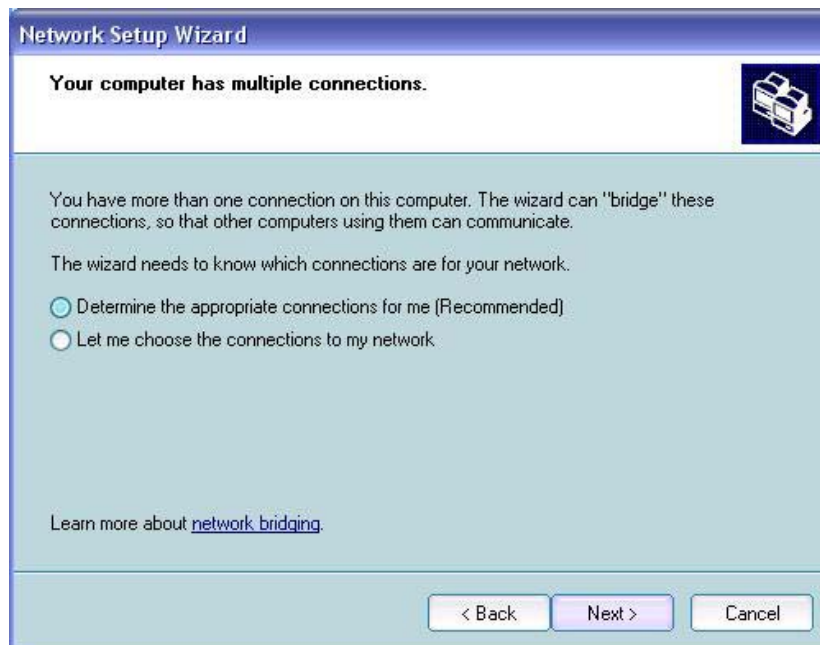




5. Στο επόμενο παράθυρο πρέπει να επιλέξουμε τη σύνδεση μέσω της οποίας θα συνδέεται ο client στο Internet. Επιλέγουμε όπως στο παρακάτω σχήμα και πατάμε Next.



6. Αφήνουμε την επιλογή ως έχει (Recommended) ώστε να καθορίσει αυτόματα ο wizard τις πολλαπλές συνδέσεις και να τις γεφυρώσει (bridge). Πατάμε Next.



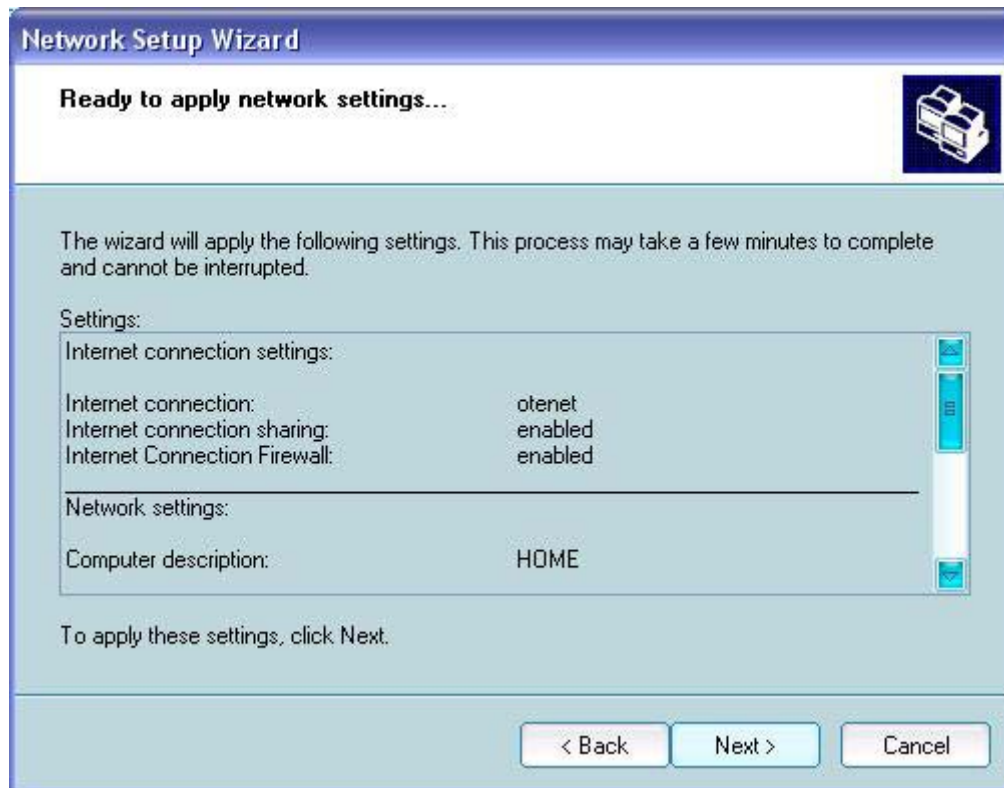
7. Δίνουμε μια περιγραφή κι ένα όνομα στον client.

The screenshot shows the 'Network Setup Wizard' dialog box. The title bar reads 'Network Setup Wizard'. The main heading is 'Give this computer a description and name.' with a computer icon to the right. Below this, there are two input fields: 'Computer description:' with the text 'HOME' and 'Examples: Family Room Computer or Monica's Computer'; and 'Computer name:' with the text 'STAVROSDESKTOP' and 'Examples: FAMILY or MONICA'. Below the input fields, there is a paragraph: 'The current computer name is STAVROSDESKTOP. Some Internet Service Providers (ISPs) require that you use a specific computer name. This is often true for computers with a cable modem. If this is the case for your computer, do not change the computer name provided by your ISP.' At the bottom, there is a link: 'Learn more about [computer names and descriptions](#).' and three buttons: '< Back', 'Next >', and 'Cancel'.

8. Δίνουμε ένα όνομα για το WORKGROUP. Αυτό το όνομα πρέπει να χρησιμοποιήσει και ο server όταν θα ακολουθήσει την ίδια διαδικασία. Πατάμε Next.

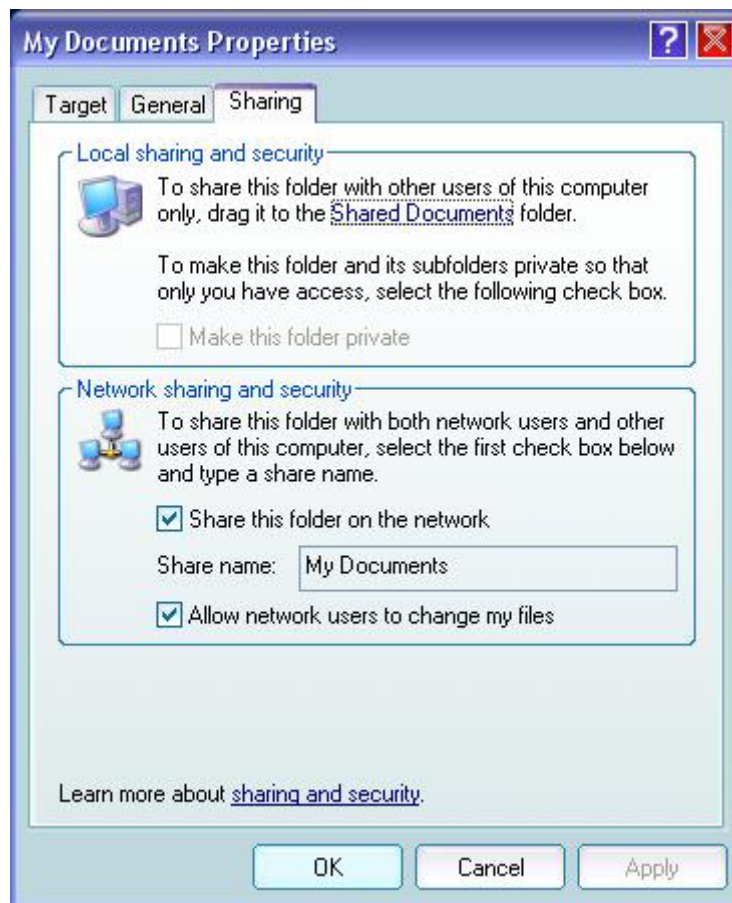
The screenshot shows the 'Network Setup Wizard' dialog box. The title bar reads 'Network Setup Wizard'. The main heading is 'Name your network.' with a computer icon to the right. Below this, there is a paragraph: 'Name your network by specifying a workgroup name below. All computers on your network should have the same workgroup name.' Below the paragraph, there is an input field for 'Workgroup name:' with the text 'MSHOME' and 'Examples: HOME or OFFICE'. At the bottom, there are three buttons: '< Back', 'Next >', and 'Cancel'.

9. Αφού ακολουθήσουμε σωστά όλα τα βήματα, εμφανίζεται ένα παράθυρο που μας δίνει πλήρη αναφορά όλων των ρυθμίσεων που κάναμε. Πατώντας Next οι ρυθμίσεις εφαρμόζονται στον υπολογιστή. Κάνουμε επανεκκίνηση και ο client έχει πλέον δημιουργήσει μια LAN σύνδεση. Ακολουθούμε τα ίδια βήματα και στο server με τη διαφορά του βήματος 4.



Για να μπορέσει ο ένας υπολογιστής να 'δει' φακέλους του άλλου θα πρέπει να τεθούν οι κατάλληλες άδειες διαμοίρασης (sharing permissions). Έστω ότι για παράδειγμα θέλουμε ο server να βλέπει τον φάκελο My Documents του client. Κάνουμε τα εξής:

1. Κάνουμε δεξί κλικ στον φάκελο My Documents στην επιφάνεια εργασίας του client και επιλέγουμε Properties.
2. Στο Sharing tab βάζουμε τικ όπως στο παρακάτω σχήμα και επιλέγουμε ένα όνομα με το οποίο θα αναγνωρίζει ο server αυτόν το φάκελο.



Όσοι φάκελοι είναι διαμοιραζόμενοι εμφανίζουν μια ανοιχτή παλάμη στην κάτω αριστερή γωνία του εικονιδίου τους.

Για να δει ο client τα διαμοιραζόμενα αρχεία (ή αντιστοίχως ο server του client) αρκεί να πατήσει Start→My Network Places. Εκεί εμφανίζονται όλοι οι διαμοιραζόμενοι φάκελοι στους οποίους έχει πρόσβαση ο υπολογιστής (βλέπε το πρώτο σχήμα αυτού του κεφαλαίου).

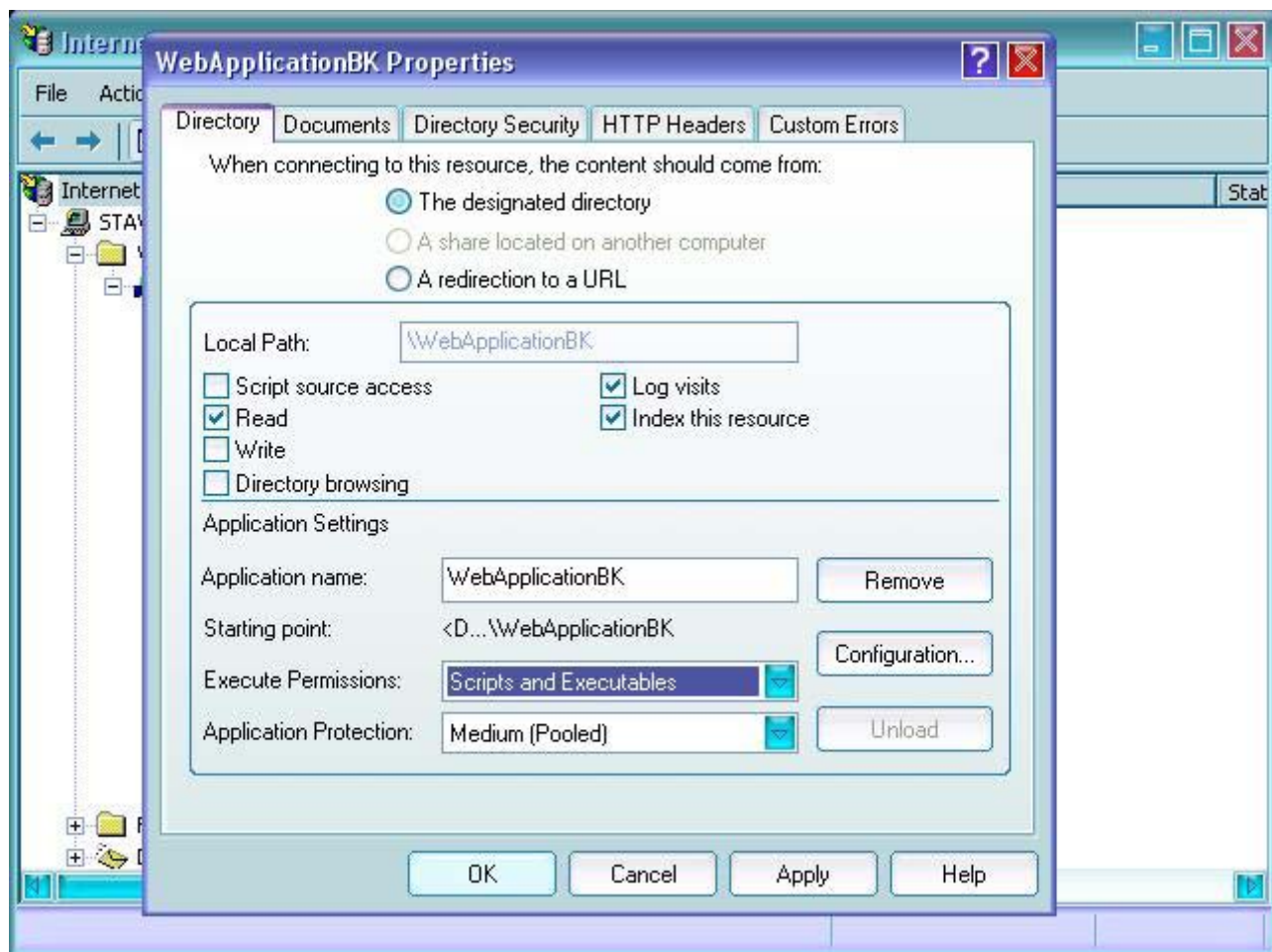
---

## Εγκατάσταση μιας ASP εφαρμογής

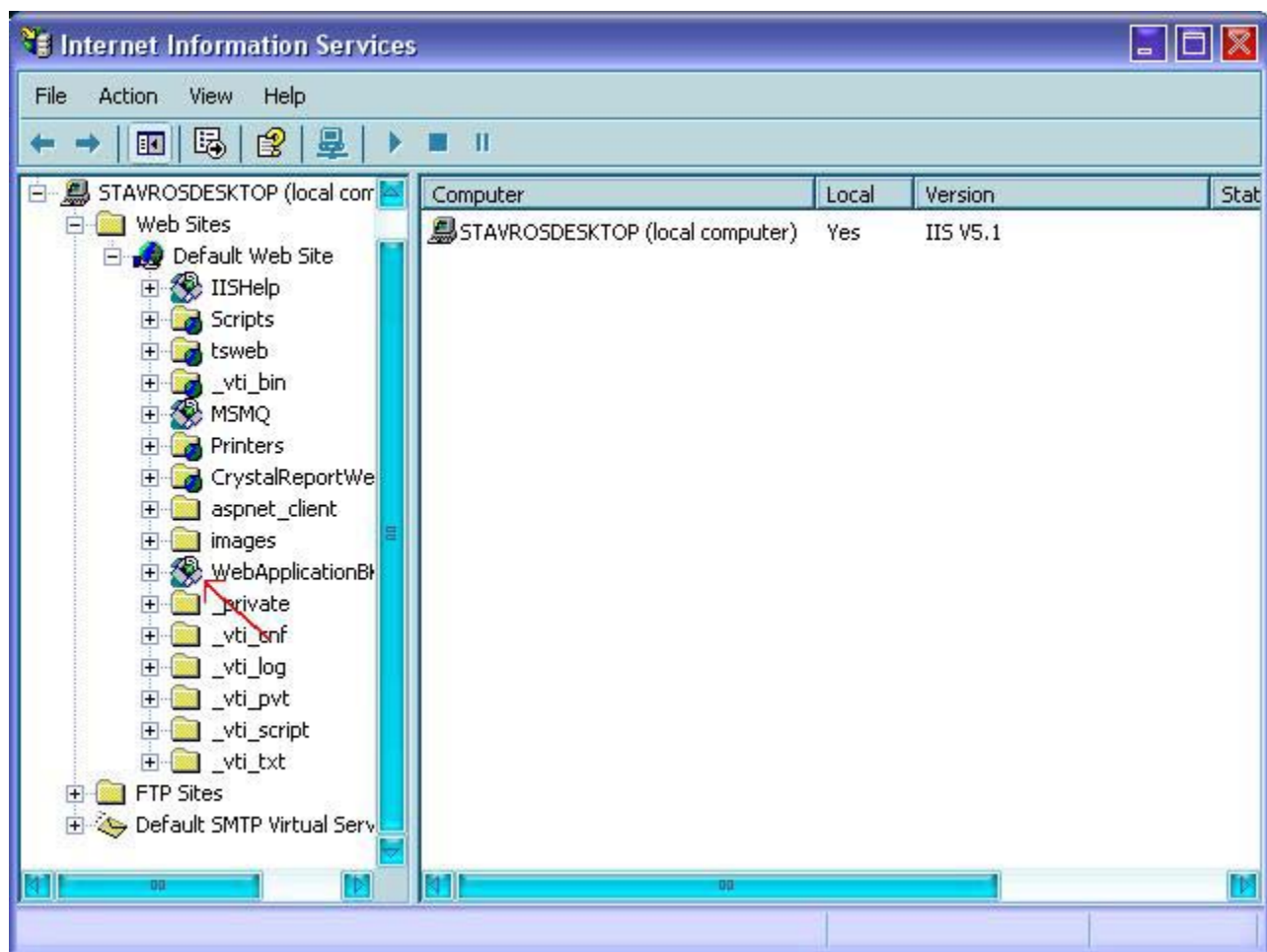
Δεδομένης μιας ASP εφαρμογής, πρέπει να ακολουθήσουμε τα παρακάτω βήματα προκειμένου να εγκαταστήσουμε την εφαρμογή αυτή στο server. Στο παράδειγμά μας θα χρησιμοποιήσουμε την εφαρμογή 'WebApplicationBK'. Σημειώστε ότι αυτό είναι το όνομα που έχει δοθεί στο όλο project του .NET που χρησιμοποιήσαμε:

1. Δημιουργούμε μέσα στο φάκελο '<Disk Drive>:\inetpub\wwwroot' ένα φάκελο με το όνομα του project της εφαρμογής (εδώ το 'WebApplicationBK') και τοποθετούμε εκεί όλα τα αρχεία του project.

2. Ανοίγουμε τον IIS από *Start* → *Control Panel* → *Administrative Tools* → *Internet Information Services*.
3. Πατάμε το '+' στα αριστερά του ονόματος του υπολογιστή μας, μετά το '+' στα αριστερά των 'Web Sites' και τέλος το '+' στο 'Default Web Site'.
4. Στους καταλόγους που υπάρχουν εκεί θα συμπεριλαμβάνεται και ο κατάλογος 'WebApplicationBK'. Πατάμε δεξί κλικ πάνω στον φάκελο αυτόν κι επιλέγουμε Properties.
5. Στα 'Application Settings' πατάμε 'Create' και στα 'Execute Permissions' επιλέγουμε 'Scripts and Executables'.



6. Πατώντας Apply το εικονίδιο της εφαρμογής αλλάζει από φάκελος στο εικονίδιο που φαίνεται στο παρακάτω σχήμα.



7. Αν θέλουμε τώρα ο server να δει την εφαρμογή αυτήν, τότε μπαίνουμε στον browser του server (Internet Explorer στο παράδειγμα) και στο πεδίο 'Address' πληκτρολογούμε  
*http://localhost/WebApplicationBK.*
8. Αν θέλουμε ο client να συνδεθεί στην εφαρμογή του server τότε μπαίνουμε στον browser του client και στο πεδίο 'Address' πληκτρολογούμε  
*http://<server\_name>/WebApplicationBK.*

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 6

**WEBSERVER STRESS TOOL 6.0**

**T**ο Webserver Stress Tool 6.0 (της εταιρίας Paessler) είναι ένα πολύ χρήσιμο εργαλείο, το οποίο μπορεί να προσομοιώσει σχεδόν οποιονδήποτε αριθμό από χρήστες που προσβαίνουν σε ένα website μέσω HTTP/HTTPS. Το λογισμικό αυτό ‘φορτώνει’ εικονικά με αιτήσεις έναν server, έτσι ώστε να μπορέσουμε να μετρήσουμε την απόδοσή του. Το εργαλείο τρέχει σε έναν client και με κατάλληλα scripts αποστέλλονται εικονικές αιτήσεις στο server. Με αυτόν τον τρόπο, ενώ διαθέτουμε μόνο έναν server κι έναν client συνδεδεμένους σε LAN, μπορούμε να δημιουργήσουμε ένα εικονικό περιβάλλον, στο οποίο πολλαπλοί εικονικοί clients στέλνουν αιτήσεις στο server. Στο κεφάλαιο αυτό θα παρουσιάσουμε ένα tutorial του εργαλείου, για να περάσουμε έπειτα στο πειραματικό μέρος.

---

## Είδη test

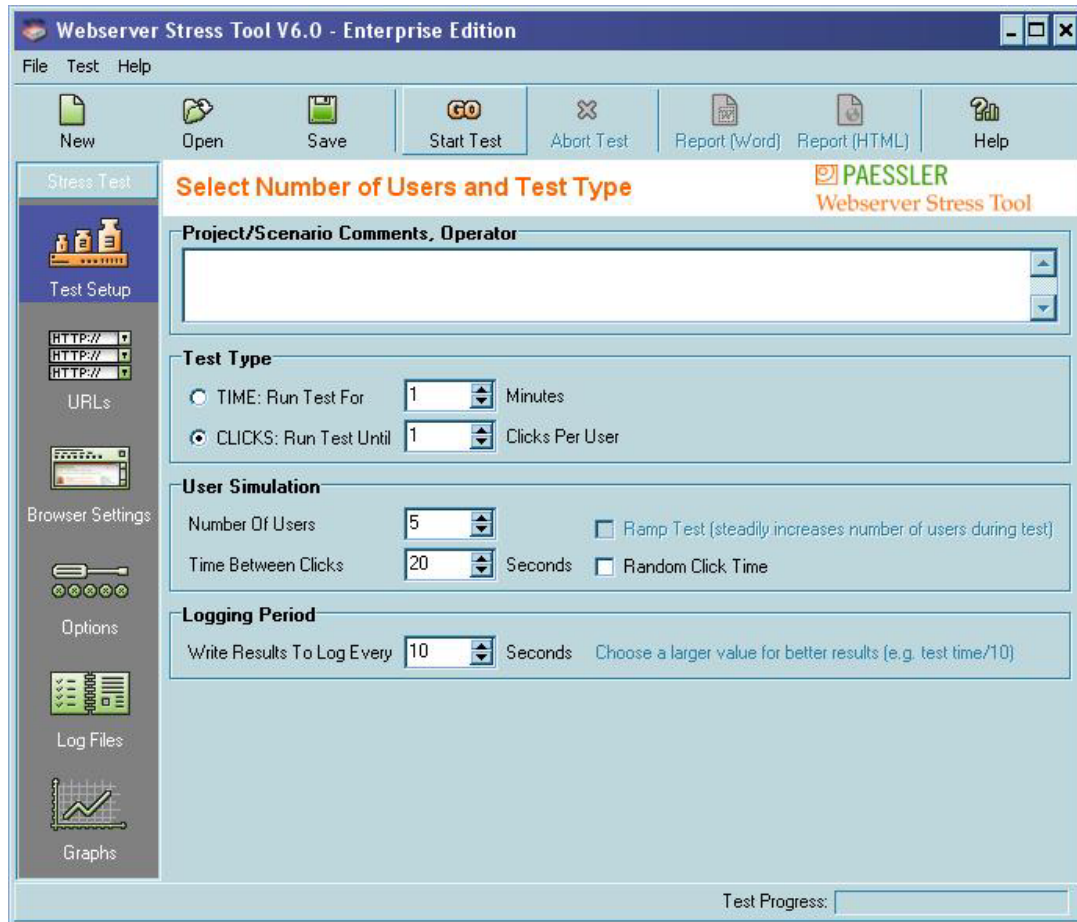
Τα είδη των tests που μπορούν να διενεργηθούν με το εργαλείο είναι τα εξής:

- **Performance tests:**  
Χρησιμοποιούνται για να ελέγχουν κάθε μέρος του server ή της δικτυακής εφαρμογής για να ανακαλύψουν ποια μέρη του website είναι αργά και πώς μπορείτε να τα κάνετε γρηγορότερα. Τις περισσότερες φορές, αυτό γίνεται ελέγχοντας διάφορες υλοποιήσεις μοναδικών web σελίδων/scripts προκειμένου να διαπιστωθεί ποια έκδοση του κώδικα είναι η γρηγορότερη.
- **Load tests:** τεστάρουν το website χρησιμοποιώντας την καλύτερη εκτίμηση για την κίνηση (traffic) που πρέπει να υποστηρίζει το website σας. Θεωρήστε το αυτό ως ‘τεστ πραγματικού κόσμου’ του website.
- **Stress tests:** είναι προσομοιωμένες ‘βάνουσες επιθέσεις’ (‘brutal attacks’) που εφαρμόζουν υπερβολικό φόρτο στον webserver σας. Στον πραγματικό κόσμο, τέτοιες καταστάσεις μπορούν να δημιουργηθούν από μαζικές αιτήσεις χρηστών (αιχμές στα διαγράμματα), που είναι πέρα από την κανονική χρήση.
- **Ramp tests:** χρησιμοποιούνται για να βρουν τον μέγιστο αριθμό των χρηστών που μπορεί να εξυπηρετήσει ο server χωρίς μηνύματα σφάλματος.
- **Άλλα** συνηθισμένα τεστ είναι επίσης πιθανά, όπως τεστ για να διασφαλίσουμε ότι οι σελίδες μπορούν να ζητηθούν ταυτόχρονα χωρίς προβλήματα όπως αδιέξοδα βάσεων δεδομένων, σημαφόροι κλπ.



## Ρυθμίσεις τεστ

Με το που ξεκινάτε το Webserver Stress Tool, σας εμφανίζεται το παράθυρο ‘Select Number Of Users and Test Type’:



Αυτό το παράθυρο σας επιτρέπει να ρυθμίσετε το σενάριο φόρτου που θέλετε να προσομοιώσετε στο τεστ.

### **Project/Scenario Comments, Operator**

Εδώ μπορείτε να εισάγετε ελεύθερο κείμενο για τον σχολιασμό του τεστ. Αυτό το σχόλιο θα εισαχθεί στις αναφορές του τεστ αργότερα και θα σας βοηθήσει στην επαναδημιουργία του, αν αυτό χρειαστεί.

### **Test Type**

Το Webserver Stress Tool προσφέρει δύο κύριους τύπους τεστ οι οποίοι διαφέρουν στη μέθοδο καθορισμού της λήξης του τεστ:

- **TIME:** τα τεστ τρέχουν για έναν καθορισμένο αριθμό λεπτών
- **CLICKS:** το τεστ τερματίζεται όταν κάθε χρήστης έχει διαπράξει έναν συγκεκριμένο αριθμό clicks.

Για τον τύπο TIME έχετε επιπλέον την επιλογή να τρέξετε το λεγόμενο *Ramp Test*, το οποίο αυξάνει τον αριθμό των ταυτόχρονων χρηστών κατά την πάροδο του χρόνου του τεστ.

### User Simulation

Αυτό το τμήμα καθορίζει το πώς προσομοιώνονται οι χρήστες. Πρώτα από όλα πρέπει να εισάγετε τον **αριθμό των χρηστών** που θέλετε να προσομοιώσετε. Θυμηθείτε ότι αυτός είναι ο αριθμός των χρηστών που χρησιμοποιούν ταυτόχρονα το website σας για το φόρτο που θέλετε να προσομοιώσετε.

Ο αριθμός των χρηστών μπορεί να είναι από 1 μέχρι μερικές χιλιάδες. Ο μέγιστος αριθμός των ταυτόχρονων χρηστών που μπορούν να προσομοιωθούν επιτυχώς εξαρτάται σχεδόν απόλυτα από το εύρος ζώνης του δικτύου και την υπολογιστική ισχύ του client που τρέχει το Webserver Stress Tool.

Ένας κανόνας εύρεσης του μέγιστου αριθμού ταυτόχρονων χρηστών που μια μηχανή INTEL/AMD μπορεί εύκολα να προσομοιώσει είναι:

*(συχνότητα επεξεργαστή σε MHz) χρήστες*

Αφού κάθε χρήστης τρέχει στο δικό του νήμα το τεστ δουλεύει καλύτερα αν είναι εγκατεστημένη αρκετή μνήμη. Προσπαθήστε να έχετε 50-200KB ελεύθερα στη RAM για κάθε προσομοιωμένο χρήστη.

Επίσης πρέπει να εισάγετε το **χρόνο μεταξύ των Clicks** των χρηστών. Αυτή η ρύθμιση είναι το ίδιο σημαντική όσο και ο αριθμός των χρηστών. Αν εισάγετε μια τιμή που είναι σωστή κατά το ήμισυ, θα προσομοιώσετε το διπλό traffic.

**Σημαντικό: Αυτές οι δύο τιμές είναι οι πιο κρίσιμες τιμές που θα εισάγετε!**

Προκειμένου να δημιουργήσετε τον υψηλότερο δυνατό φόρτο, θέστε το χρόνο μεταξύ των clicks στο 0. Με αυτόν τον τρόπο, το Webserver Stress Tool θα στέλνει την επόμενη αίτηση ενός χρήστη αμέσως μετά το τέλος της προηγούμενης αίτησης. Παρκαλώ σημειώστε ότι, σε αυτήν την περίπτωση, ο αριθμός των 40-80 χρηστών είναι αρκετός για τα περισσότερα τεστ (υψηλότερες τιμές μπορούν να μειώσουν το φόρτο λόγω της επιβάρυνσης του πολυνηματισμού).

Ενεργοποιώντας τον **τυχαίο χρόνο click** μπορείτε να πείτε στο Webserver Stress Tool να χρησιμοποιεί τυχαία ένα χρόνο μεταξύ των clicks, ο οποίος είναι μεταξύ του 0 και

του αριθμού των δευτερολέπτων που εισάγατε στο **χρόνο μεταξύ των clicks**. Αυτό θα κάνει το τεστ περισσότερο δυναμικό, αλλά δυσκολότερο να αναπαραχθεί.

Για τεστ απόδοσης και πίεσης (stress) μπορείτε να ενεργοποιήσετε το χαρακτηριστικό **Ramp Test** (μόνο για τεστ τύπου **TIME**). Σ' αυτό το μοτίβο, το Webserver Stress Tool αυξάνει τον αριθμό των ενεργών χρηστών συνεχώς, από το μηδέν μέχρι τον **αριθμό των χρηστών**, κατά την πάροδο του χρόνου. Με αυτόν τον τρόπο μπορείτε εύκολα να καταλάβετε ποιοι είναι οι περιορισμοί ενός webserver.

### Logging Period

Χρησιμοποιώντας τη ρύθμιση **γράψε αποτελέσματα στο log κάθε x δευτερόλεπτα** μπορείτε να πείτε στο Webserver Stress Tool κάθε πότε να καταγράφει τα αποτελέσματα μιας περιόδου σε ένα log και να τα μαρκάρει στα γραφήματα. Επιλέγοντας μια μικρή τιμή, τα αποτελέσματα που θα πάρετε θα είναι ασταθή (π.χ. επειδή οι περίοδοι καταγράφουν διαφορετικό αριθμό αιτήσεων), ενώ επιλέγοντας μια μεγάλη τιμή, σημαντική πληροφορία μπορεί να κρυφτεί (π.χ. στα ramp tests δεν θα δείτε ακριβώς κάτω από τι φόρτο ξεκίνησαν τα προβλήματα).

Ως κανόνας, μια τιμή μεταξύ των 10 και 30 δευτερολέπτων πρέπει να είναι καλή για τα περισσότερα τεστ. Για μεγαλύτερα τεστ (π.χ. με χρόνο τεστ μεγαλύτερο των 5 λεπτών), διαλέξτε ως τιμή το χρόνο του τεστ διαιρούμενο με το 10-50.

---

## **Επιλέγοντας τα URLs**

Πατήστε στο 'URLs' στο αριστερό toolbar και θα εμφανιστεί ο διάλογος των ρυθμίσεων των URL.

### URL Sequencing

Αυτή η ρύθμιση καθορίζει τον τρόπο με τον οποίο το Webserver Stress Tool αναθέτει τα URLs στους χρήστες κατά τη διάρκεια του τεστ. Υπάρχουν 4 επιλογές:

- **Users select URL for each click randomly:**

Χρησιμοποιώντας την ενσωματωμένη συνάρτηση random, το Webserver Stress Tool απλά επιλέγει ένα από τα URLs για κάθε click. Ανάλογα με το website σας, αυτή μπορεί να είναι μια καλή ρύθμιση για να δημιουργήσετε φόρτο πραγματικού κόσμου.

- **Users always click the same URL:**

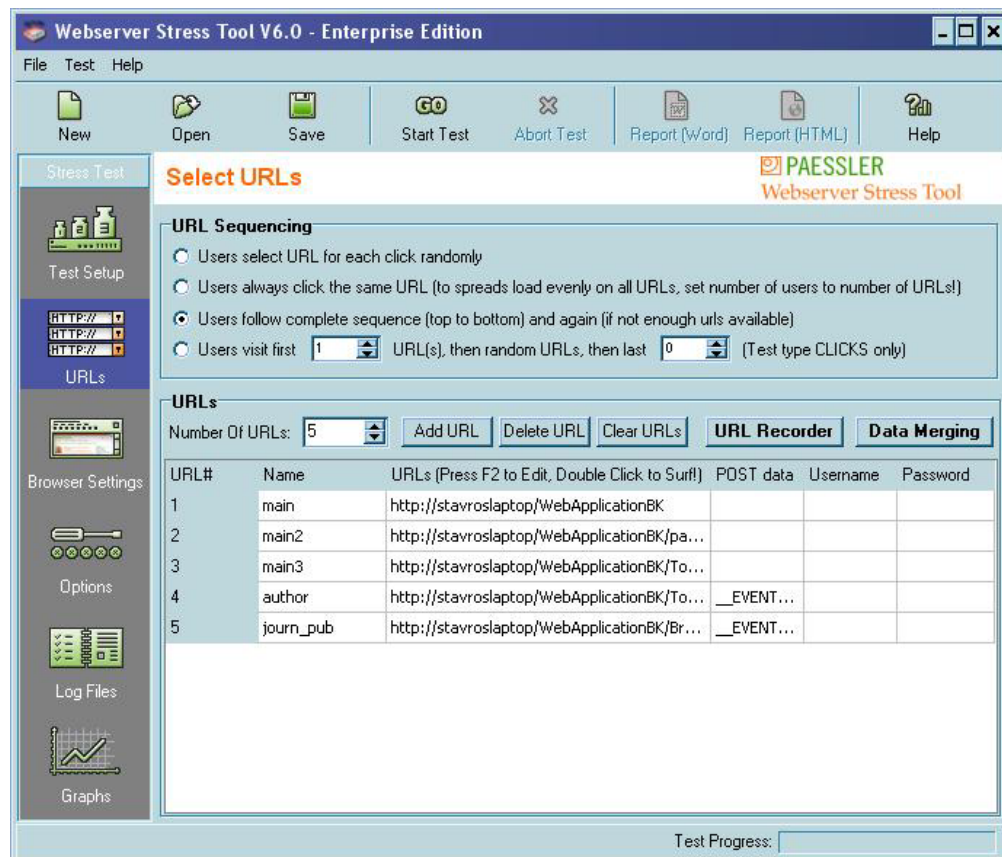
Στο ξεκίνημα του τεστ, κάθε χρήστης επιλέγει ένα URL και χρησιμοποιεί μόνο αυτό για όλη τη διάρκεια του τεστ. Για να διαμοιράσετε με ίσο τρόπο το φόρτο σε όλα τα URLs, θέστε τον αριθμό των χρηστών να είναι πολλαπλάσιος ή ίσος με τον αριθμό των URLs. Αυτή η ρύθμιση είναι πολύ χρήσιμη για τη σύγκριση των χρόνων απόκρισης διαφορετικών web pages (π.χ. με διαφορετικές υλοποιήσεις ενός χαρακτηριστικού) ή για την εύρεση των σελίδων που είναι πιο αργές από άλλες.

- **Users follow complete sequence:**

Όλοι οι χρήστες θα χρησιμοποιήσουν το URL#1 για το πρώτο click, το URL#2 για το δεύτερο κ.ο.κ.. Αν ένας χρήστης φθάσει στο τελευταίο URL θα ξαναρχίσει πάλι με το URL#1. Χρησιμοποιήστε αυτή τη ρύθμιση για να προσομοιώσετε μονοπάτια μέσα στη σελίδα σας.

- **Users visit first X URLs, then random, then last Y URLs:**

Όλοι οι χρήστες θα χρησιμοποιήσουν τα πρώτα X URLs (από πάνω προς τα κάτω) για αρχή. Έπειτα, τα υπόλοιπα URLs ανατίθενται τυχαία. Για τα CLICK τεστ έχετε την πρόσθετη επιλογή του να θέσετε έναν αριθμό URLs στο κάτω μέρος της λίστας των URLs, που θα πρέπει να επισκεφθούν οι χρήστες στο τέλος του τεστ. Αυτό είναι το σωστό μοτίβο αν έχετε ένα website στο οποίο οι χρήστες πρέπει να κάνουν log in χρησιμοποιώντας μερικά URLs, μετά να σερφάρουν και τέλος να κάνουν log out.



## URLs

Στο πεδίο ‘**Number of URLs**’ μπορείτε να επιλέξετε πόσα URLs θα συμπεριληφθούν στο τεστ. Είναι αποδεκτός οποιοσδήποτε αριθμός από 1 μέχρι μερικές χιλιάδες.

Για κάθε URL μπορείτε να εισάγετε ένα **όνομα**, το **URL**, τα επιθυμητά **δεδομένα POST**, ένα **username** κι ένα **password**.

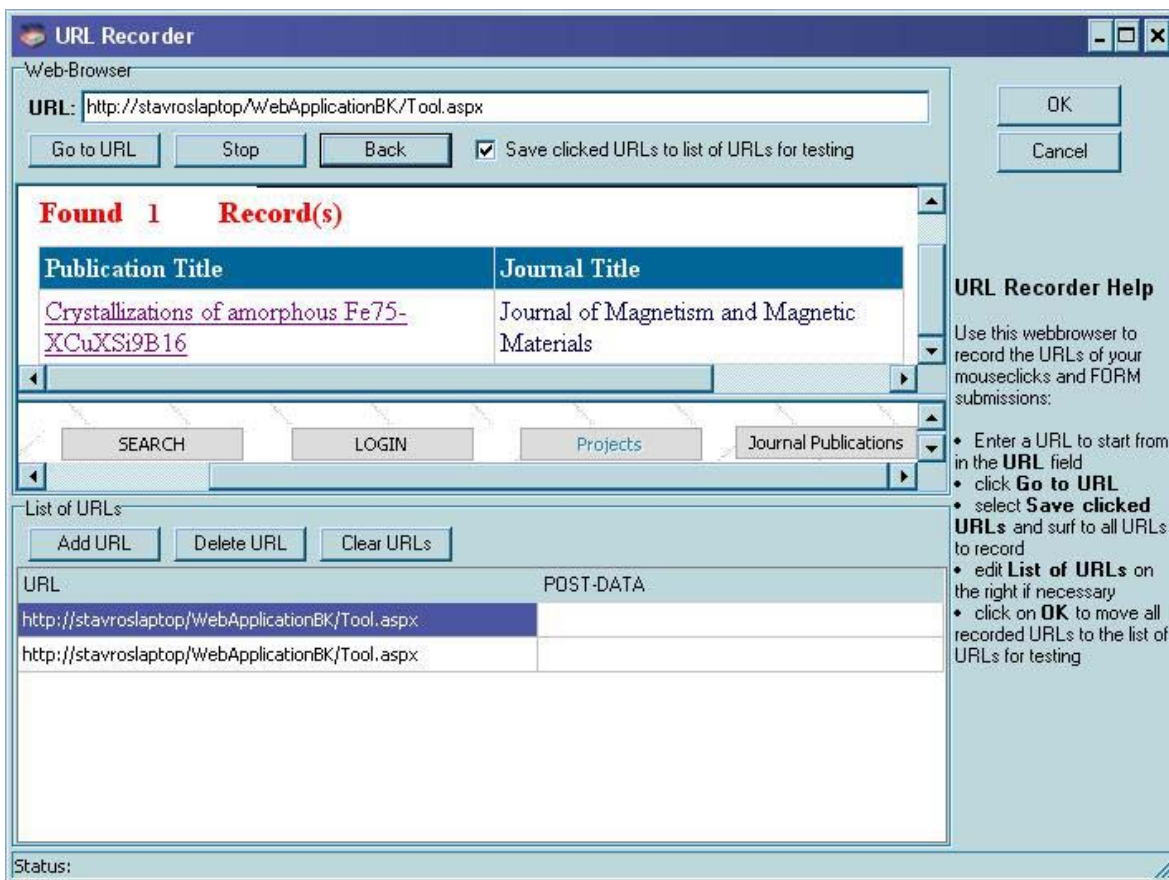
Χρησιμοποιήστε το ‘**Add URL**’ για να προσθέσετε μία ακόμη εγγραφή στη λίστα και το ‘**Delete URL**’ για να διαγράψετε τις επιλεγμένες εγγραφές. Το ‘**Clear URLs**’ καθαρίζει όλη τη λίστα. Μπορείτε να εισάγετε τα URLs χειροκίνητα ή χρησιμοποιώντας το ‘**URL Recorder**’.

Ακολουθεί μια περιγραφή του κάθε πεδίου:

- Επιλέξτε ένα απλό **όνομα** για κάθε εγγραφή. Χρησιμοποιείται στα γραφήματα και στα log files.
- Το URL θα πρέπει να είναι στην standard URL μορφή  
Π.χ. *http://yourservername:port/path/pagename?GET-params*  
*http://www.paessler.com:8080/login.asp?username=dirk)*
- Συνήθως οι αιτήσεις HTTP αποστέλλονται χρησιμοποιώντας τη μέθοδο GET. Αν υπάρχει κείμενο στο πεδίο δεδομένων POST της λίστας URL, η αίτηση αποστέλλεται χρησιμοποιώντας την POST μέθοδο. Το κείμενο θα πρέπει να είναι κωδικοποιημένο ως URL. Συνήθως η GET χρησιμοποιείται για απλές φόρμες και η POST για φόρμες με πολλά δεδομένα ή φόρμες που πρέπει να μεταφερθούν με ασφάλεια χρησιμοποιώντας SSL. Η μέθοδος GET ενσωματώνει όλα τα δεδομένα στο URL (διαχωριζόμενα από ένα ‘?’). Για μια αίτηση POST τα δεδομένα αποστέλλονται ως μέρος της επικεφαλίδας της αίτησης HTTP.
- Αν χρησιμοποιείτε BASIC πιστοποίηση, εισάγετε εδώ το **Username** και το **password** για ένα URL. Με την BASIC πιστοποίηση τα δεδομένα του log in αποστέλλονται ως μέρος της επικεφαλίδας του HTTP σε καθαρό κείμενο. Αυτό, προφανώς, δεν δουλεύει για μηχανισμούς log in που χρησιμοποιούνται στις φόρμες HTML. Πρέπει να προσομοιώσετε αυτά τα log ins χρησιμοποιώντας δεδομένα GET/FORM.

## URL Recorder

Το Webserver Stress Tool προσφέρει έναν ‘click recorder’ που σας βοηθάει να φτιάξετε την λίστα των URLs του τεστ. Κάντε κλικ στο ‘**URL Recorder**’ στο αριστερό toolbar:



Απλώς εισάγετε το πρώτο URL στο πεδίο **URL** και κάντε κλικ στο **‘Go to URL’**. Παρακολουθήστε τη λίστα των URLs στο κάτω μέρος του παραθύρου. Κάθε φορά που κάνετε κλικ σε ένα link στο παράθυρο του browser ή υποβάλλετε μια φόρμα, το URL προστίθεται στη λίστα. Αν θέλετε να κάνετε browse χωρίς να καταγράφεται το click, απλά απενεργοποιήστε το **‘Save click URLs to list of URLs’**.

Αν μια αίτηση POST υποβάλλεται, τα δεδομένα POST επίσης σώζονται. Αν μια σελίδα είναι ένα frameset, όλα τα URLs του frameset προστίθενται στη λίστα.

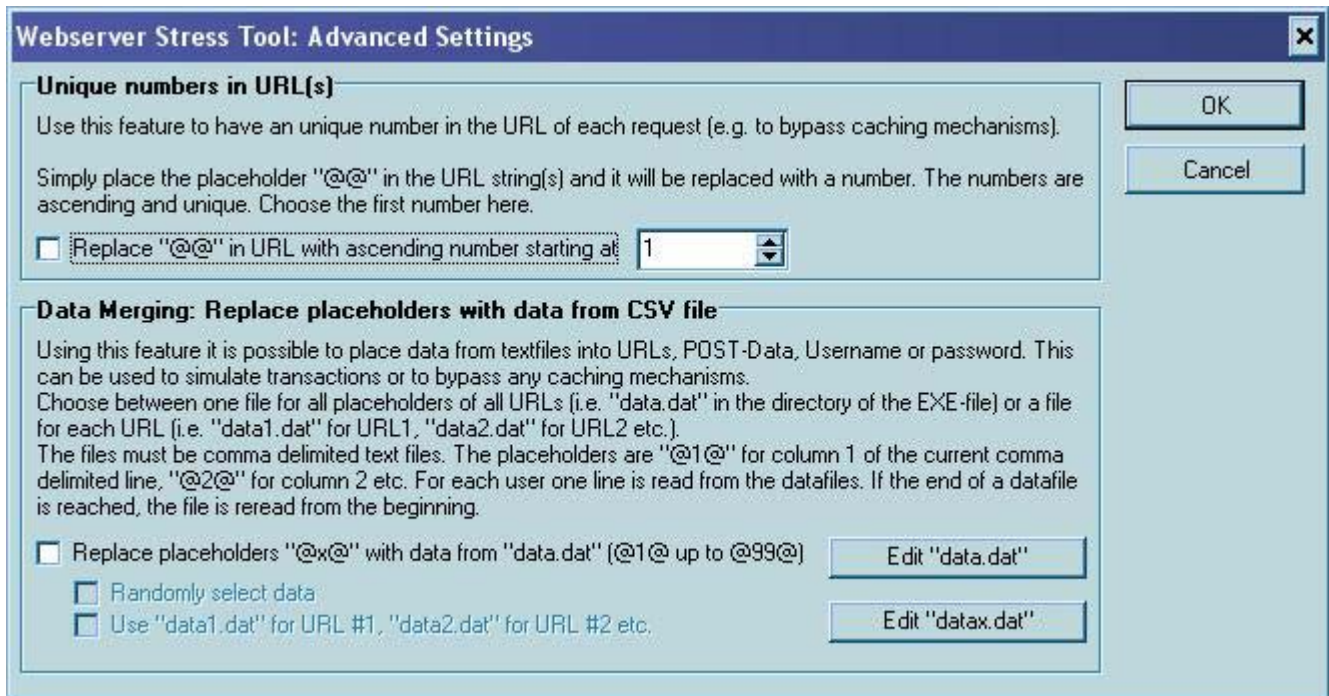
Αν ένα click ανοίγει ένα νέο παράθυρο, το Webserver Stress Tool επίσης ανοίγει ένα νέο παράθυρο και καταγράφει τα περαιτέρω clicks.

Μπορείτε να τροποποιείτε τη λίστα των URLs χρησιμοποιώντας τα **‘Add URL’**, **‘Delete URL’** και **‘Clear URL’**. Επίσης τα URLs μπορούν να αλλάξουν (διπλό κλικ πάνω σε ένα URL).

Παρακαλώ σημειώστε ότι ο browser που χρησιμοποιείται από το Webserver Stress Tool είναι η εγκατεστημένη έκδοση του Internet Explorer των Windows.

## Data Merging

To Webservice Stress Tool, προκειμένου να είναι πιο ευέλικτο με τα URLs ή για να προσπερνά μηχανισμούς εναποθήκευσης (ενός web, proxy ή εφαρμογών server), προσφέρει τη δυνατότητα συγχώνευσης πρόσθετων δυναμικών δεδομένων μέσα στα URLs που ζητήθηκαν.



**Unique numbers in URL(s):** Χρησιμοποιήστε αυτό το χαρακτηριστικό για να έχετε έναν μοναδικό αριθμό μέσα στο URL κάθε αίτησης, ώστε να προσπεράσετε τους μηχανισμούς εναποθήκευσης. Απλά τοποθετήστε ένα '@@' (placeholder) μέσα στα URL string(s) και αυτό θα αντικατασταθεί με έναν αριθμό. Οι αριθμοί είναι αυξανόμενοι και μοναδικοί για όλους τους χρήστες και τα clicks.

**Replace placeholders with data from file:** Χρησιμοποιώντας αυτό το χαρακτηριστικό, είναι δυνατό να τοποθετήσετε δεδομένα από αρχεία κειμένου μέσα στα URLs, στα δεδομένα POST, στο username και το password. Αυτό μπορεί να χρησιμοποιηθεί για να προσομοιώσει συναλλαγές ή ακόμη για να προσπεραστεί οποιοσδήποτε μηχανισμός εναποθήκευσης.

Διαλέξτε μεταξύ ενός αρχείου για όλα τα placeholders όλων των URLs (όπως 'data1.dat' στον κατάλογο του αρχείου EXE) και ενός αρχείου για κάθε URL (όπως 'data1.dat' για το URL1, 'data2.dat' για το URL2 κ.ο.κ.).

Μπορείτε να αλλάξετε τα αρχεία αυτά κάνοντας κλικ στο **Edit “data.dat”** και **Edit “datax.dat”** αντίστοιχα.

Αν χρειάζεστε να αλλάξετε τα αρχεία αυτά χειροκίνητα τότε: Το περιεχόμενο των αρχείων θα πρέπει να είναι διαχωριζόμενο από κόμμα (.). Τα placeholders είναι ‘@1@’ για την πρώτη στήλη, ‘@2@’ για τη δεύτερη κ.ο.κ.. Για κάθε χρήστη διαβάζεται μια γραμμή από τα αρχεία δεδομένων. Αν φτάσει το τέλος ενός αρχείου δεδομένων, το αρχείο διαβάζεται ξανά από την αρχή.

Σημείωση: αν υπάρχουν διαστήματα σε μια στήλη τότε χρησιμοποιήστε “ (quotation marks) γύρω από τα δεδομένα και “” όταν θέλετε να συμπεριλάβετε ένα”.

Π.χ. 1,”one and two”,3,4,”four and five”,”This “” is a quotation mark”.

### **Data Merging – Ένα Tutorial**

Ας υποθέσουμε ότι έχετε τρία URLs, τα οποία χρειάζεται να επισκεφθεί ένας χρήστης προκειμένου να κάνει log in στο site σας και να κάνει κάτι.

- <http://myserver/homepage.htm> (μια αρχική σελίδα με φόρμα για log in)
- <http://myserver/login.htm> (οι παράμετροι POST είναι login/password, αυτή η σελίδα θέτει ένα cookie)
- <http://myserver/data.htm> (μερικές παράμετροι GET)

Θέλετε να προσομοιώσετε 10 διαφορετικούς χρήστες που κάνουν log in στο site περνώντας από τα 3 URL.

Χρησιμοποιώντας το Webserver Stress Tool, αυτό μπορεί να γίνει ως εξής:

1. Στη σελίδα **Test Setup**
2. Επιλέξτε **Test Type CLICKS**, θέστε **Number of clicks** στο 3, θέστε **Number of users** στο 10
3. Στη σελίδα **URLs**
4. Θέστε **Number of URLs** στο 3, επιλέξτε **All Users follow complete Sequence**, εισάγετε τα τρία URLs.
5. Για το URL#2 εισάγετε το ακόλουθο στην στήλη POSTDATA:  
**username=@1@&password=@2@**
6. Για το URL#3 αλλάξτε το URL κάπως έτσι:  
**http://myserver/data.htm?data=@1@**
7. Κάντε κλικ στο **Data Merging**
8. Ενεργοποιήστε το **Replace Placeholders...** και το **Use data1.dat for URL#1...**
9. Κάντε κλικ στο **Edit “Datax.dat”**



10. Απαντήστε στο “**Edit data for what URL**” με την τιμή 2
11. Εισάγετε τους συνδυασμούς username και password στις στήλες @1@ και @2@ και κάντε κλικ στο **OK**.
12. Κάντε κλικ ξανά στο **Edit “Datax.dat”**
13. Απαντήστε στο “**Edit data for what URL**” με την τιμή 3
14. Εισάγετε τα δεδομένα για το URL#3 στη στήλη @1@ και κάντε κλικ στο **OK**.
15. Κάντε κλικ στο **OK** ξανά
16. Ελέγξτε όλες τις άλλες ρυθμίσεις
17. Τρέξτε το τεστ

Αρχικά και οι 10 χρήστες στέλνουν μια αίτηση στο URL#1, το οποίο είναι η απλή αρχική σελίδα.

Έπειτα, για κάθε χρήστη ζητείται το URL login.htm, αλλά τα @1@ και @2@ στο πεδίο POSTDATA αντικαθίστανται με τις τιμές του αρχείου data2.dat, έτσι ώστε κάθε αίτηση να αποστέλλεται με διαφορετικά δεδομένα log in (το @1@ αντικαθίσταται με δεδομένα από τη στήλη 1, το @2@ από τη στήλη 2).

Αν το login.htm στείλει ένα cookie, το cookie αποθηκεύεται ατομικά για κάθε χρήστη και αποστέλλεται ξανά με τις υπόλοιπες αιτήσεις.

Για το τρίτο click, το Webserver Stress Tool αντικαθιστά το placeholder @1@ με τα strings από το αρχείο data3.dat, επομένως στέλνει 10 διαφορετικά GET URLs στο server μαζί με τα cookies που λήφθηκαν κατά το τρέξιμο του δεύτερου URL.

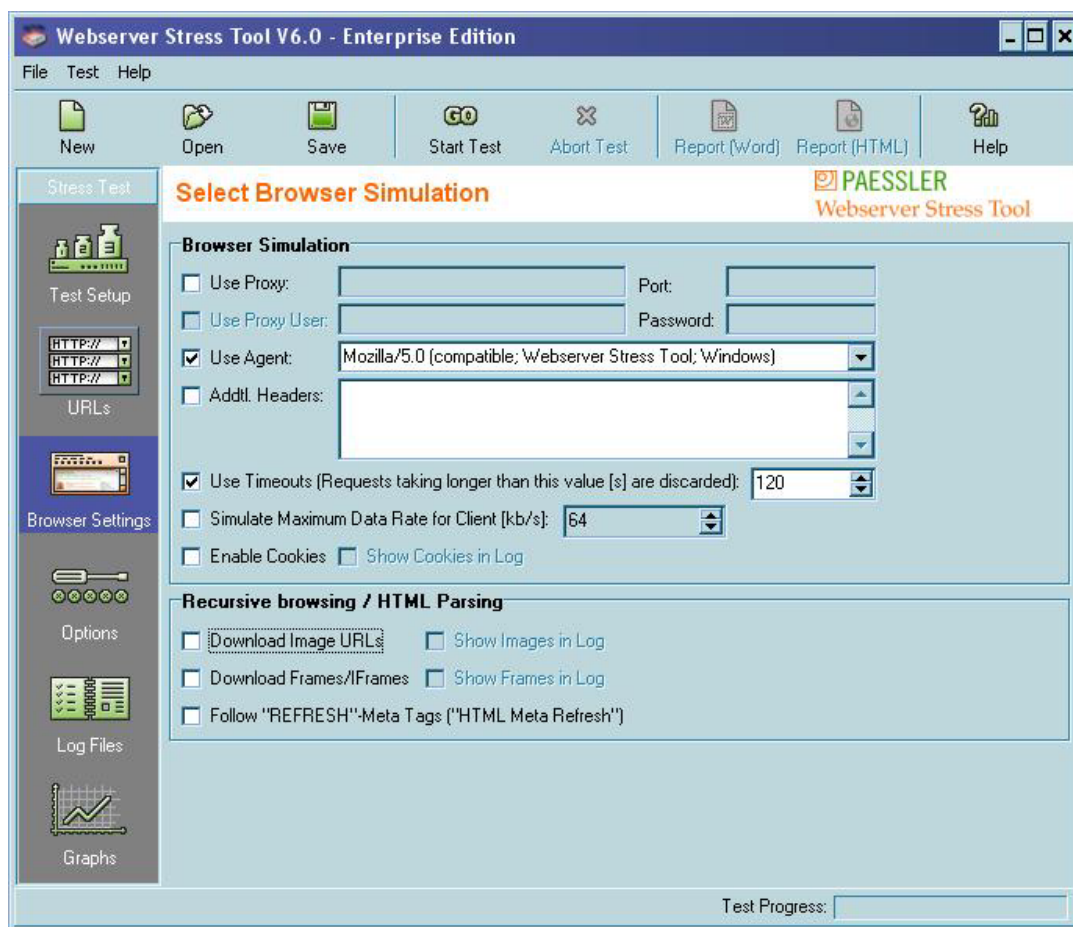
Μορείτε να δείτε το λεπτομερές log file για να διαπιστώσετε αν τα δεδομένα στάλθηκαν όπως ακριβώς επιθυμούσατε.

---

## Ρυθμίσεις Browser

Πολλά χαρακτηριστικά του προσομοιωμένου browser μπορούν να ρυθμιστούν από το χρήστη.

Πρώτα από όλα είναι σημαντικό να θεωρήσετε ότι το Web Server Stress Tool προσομοιώνει έναν browser από την ‘άποψη’ του server (π.χ. αποστολή αιτήσεων), αλλά δεν προσομοιώνει το rendering στην οθόνη του client ή την εκτέλεση scripts στην πλευρά του client. Γι’αυτόν το λόγο τα Java applets, τα javascripts κλπ., δεν εκτελούνται (η εκτέλεση scripts πολλών χρηστών θα προκαλέσει επίσης μεγάλο φόρτο στη CPU του client). Η απόδοση τέτοιων client-side τμημάτων της web εφαρμογής δεν μπορούν να μετρηθούν με εργαλεία όπως το Web server Stress Tool.



## Προσομοίωση browser

Αν χρησιμοποιείτε proxy server, επιλέξτε **Use Proxy** και εισάγετε τη διεύθυνση και το **Port** του proxy. Αν ο proxy server σας χρειάζεται πιστοποίηση, επιλέξτε **Use Proxy User** και εισάγετε τα **Username** και **Password**.

Ένα συγκεκριμένο User Agent String μπορεί να σταλλεί στο server όταν είναι επιλεγμένο το **Use Agent**. Μπορείτε να επιλέξετε ένα user agent string από τη λίστα ή να εισάγετε το string από μόνοι σας.

Για να εισάγετε τις δικές σας παραμέτρους στις επικεφαλίδες των HTTP αιτήσεων, ενεργοποιήστε το **Addtl. Headers** και εισάγετε τα δεδομένα σας στο πεδίο κειμένου.

Μπορείτε να θέσετε ένα μέγιστο χρονικό διάστημα για τον τερματισμό των HTTP αιτήσεων επιλέγοντας το **Use Timeout**. Εισάγετε το διάστημα αυτό σε δευτερόλεπτα. Μια καλή τιμή για να ξεκινήσετε είναι 60 δευτερόλεπτα, κανένας χρήστης δεν θα περιμένει παραπάνω από αυτό.

Προκειμένου να εμποδίσετε το ρυθμό δεδομένων με τον οποίο προσβαίνει ένας χρήστης στο server, μπορείτε να ενεργοποιήσετε το **Simulate Maximum Data Rate for Client** και να εισάγετε μια τιμή σε Kb/s. Αυτό μπορεί να χρησιμοποιηθεί για την προσομοίωση χρηστών που προσπελούν το server μέσω γραμμών modem (π.χ. 56 Kb/s).

Αν ο web server σας ή η web εφαρμογή χρειάζεται cookies, επιλέξτε **Enable cookies**. Τα cookies φαίνονται στο λεπτομερές log file όταν το **Show Cookies in Log** είναι επιλεγμένο. Τα cookies αποθηκεύονται για κάθε χρήστη και αποστέλλονται ξανά στο server για τις ακόλουθες αιτήσεις, έως ότου ο server στείλει ένα cookie invalidation (ακύρωση).

### **Αναδρομικό Browsing (Recursive Browsing)**

Τα χαρακτηριστικά αυτής της ομάδας θα πρέπει να χρησιμοποιηθούν μόνο σε πολύ ισχυρά client συστήματα, ειδικά για τα τεστ με συνθήκες υψηλού φόρτου, επειδή το HTML κάθε αίτησης πρέπει να αναλυθεί ολόκληρο για να αναγνωριστούν τα image URLs, link URLs κλπ.. Τα χαρακτηριστικά αυτά μπορεί να οδηγήσουν σε σημαντικό φόρτο επεξεργαστή στον client, πράγμα που μπορεί να επιφέρει ανακριβή αποτελέσματα για τη απόδοση του μετρούμενου server. Να προσέχετε το φόρτο του επεξεργαστή του client κατά τη διάρκεια του τεστ και να τρέξετε το τεστ με και χωρίς αυτά τα χαρακτηριστικά ώστε να συγκρίνετε τα αποτελέσματα.

Επιλέγοντας το **Download Image URLs**, λέτε στο Webserver Stress Tool να αναλύσει όλα τα <IMG RC="url"> tags από τον κώδικα HTML και να στείλει μια αίτηση για κάθε IMG URL στο server, αμέσως μόλις ανακτηθεί ολόκληρο το HTML. Αν μια εικόνα χρησιμοποιείται αρκετές φορές μέσα στη σελίδα, τότε ζητείται μόνο μια φορά.

Επιλέξτε το **Show Images in Log** αν θέλετε να έχετε μια εγγραφή στο log file για κάθε εικόνα, διαφορετικά μόνο μια εγγραφή παράγεται, η οποία δηλώνει πόσες εικόνες βρέθηκαν και ζητήθηκαν.

Αν το site σας χρησιμοποιεί HTML Frame tags πρέπει να επιλέξετε **Download Frames/IFrames**. Το Webserver Stress Tool τότε αναλύει τον κώδικα HTML για <FRAME> και <IFRAME> tags. Για κάθε Frame URL αποστέλλεται μια αίτηση στο server. Αν ένα frame είναι ένα frameset, ο μηχανισμός αυτός δουλεύει αναδρομικά και ζητάει τα subframes επίσης.

Επιλέξτε **Show Frames in Log** αν θέλετε να έχετε μια εγγραφή στο log file για κάθε frame, διαφορετικά παράγεται μόνο μια εγγραφή, η οποία δηλώνει πόσα frames βρέθηκαν και ζητήθηκαν.

Μερικά sites χρησιμοποιούν τα Refresh meta tags ως μέσο ανακατεύθυνσης. Για να ακολουθήσετε αυτές τις ανακατευθύνσεις επιλέξτε **Follow “Refresh” Meta Tag**. Το HTML αναλύεται για <meta name=“refresh” content=“time;url”> tags. Μόλις βρεθεί ένα tag, αποστέλλεται μια αίτηση στο server (η τιμή χρόνου δεν χρησιμοποιείται). Σημείωση: Οι ανακατευθύνσεις των επικεφαλίδων HTTP υφίστανται πάντα επεξεργασία.

---

## **Επιλογές (Options)**

### *Advanced Settings*

Όπως και με το αναδρομικό browsing, τα χαρακτηριστικά αυτής της ομάδας πρέπει να χρησιμοποιηθούν μόνο σε ισχυρά μηχανήματα client. Μπορεί να προκαλέσουν εξαιρετικά μεγάλο φόρτο επεξεργαστή και να αποφέρουν ανακριβή αποτελέσματα σχετικά με την απόδοση του μετρούμενου server.

Τα αποτελέσματα HTML όλων των αιτήσεων μπορούν να γραφτούν σε ένα αρχείο δίσκου το καθένα επιλέγοντας **Save All HTML files**. Τα ονόματα των αρχείων δημιουργούνται από τον αριθμό των χρηστών και τον αριθμό των κλικ και των αιτήσεων των χρηστών.

Το **Link-Checker** αποθηκεύει όλα τα μοναδικά URLs από όλες τις HTML σελίδες που έχουν ζητηθεί κατά τη διάρκεια του τεστ και ελέγχει όλα αυτά τα URLs για χαλασμένα links αφού τελειώσει το τεστ φόρτου/πίεσης. Τα αποτελέσματα μπορούν να βρεθούν στα log files.

Σε αργά μηχανήματα client, μπορεί να βοηθήσει η ενεργοποίηση του **Hide Stress Tool Window** για να δώσετε λίγο ακόμη ισχύ στο τεστ. Στα γρήγορα μηχανήματα (500 MHz και παραπάνω) δεν θα φανεί διαφορά.

### *Logging*

Το Webserver Stress Tool πάντα γράφει ένα περιληπτικό log file κατά τη διάρκεια του τεστ. Για περισσότερο λεπτομερή log files ενεργοποιήστε το **Write Detailed and User Logs**. Σ' αυτήν την περίπτωση θα γραφτεί στο δίσκο ένα λεπτομερές log και ένα log για κάθε δραστηριότητα χρήστη.

**Να προσέχετε ότι για τα τεστ φόρτου με εκατοντάδες ή ακόμη και με χιλιάδες χρήστες, το λεπτομερές logging μπορεί να έχει σοβαρές επιπτώσεις στην απόδοση του client και γ' αυτό οι μετρούμενες τιμές να είναι ανακριβείς.** Είναι πάντα μια

καλή ιδέα να τρέχετε τέτοια τεστ με και χωρίς λεπτομερές logging ώστε να συγκρίνετε τα αποτελέσματα.

Χρησιμοποιώντας το Write Request Log (CSV) μπορεί να δημιουργηθεί ένα επιπλέον log file σε μορφή αναγνώσιμη από μια μηχανή, το οποίο να έχει μια γραμμή δεδομένων για κάθε αίτηση του τεστ. Αυτό μπορεί να είναι καλό αν χρειάζεστε να επεξεργάζεστε τα αποτελέσματα μόνοι σας. Επίσης το request log μπορεί να επηρεάσει την απόδοση του client.

Το Webserver Stress Tool μπορεί επίσης να γράφει όλα τα παραλαμβανόμενα δεδομένα σε ένα log file. Για να γίνει αυτό επιλέξτε **Write Header to Log** για όλα τα δεδομένα μέσα στις επικεφαλίδες HTTP και **Write data to Log** για όλα τα δεδομένα HTML των αιτήσεων.

Όταν χρησιμοποιείτε το **Write on Error Only**, στο log γράφονται μόνο τα δεδομένα των αιτήσεων που κατέληξαν σε σφάλμα.

### Local IP addresses to use

Αν το μηχάνημα στο οποίο τρέχετε το Webserver Stress Tool έχει περισσότερες από μια IP διευθύνσεις, μπορείτε να επιλέξετε ποια IP διεύθυνση θα χρησιμοποιηθεί για να προσομοιώσει τις αιτήσεις. Βρήκαμε ότι, για τις περισσότερες περιπτώσεις των HTTP τεστ φόρτου/πίεσης, δεν χρειάζεται να έχετε παραπάνω από μια διεύθυνση, καθώς ο server απαντάει σε όλες τις αιτήσεις με τον ίδιο τρόπο ανεξάρτητα από την IP διεύθυνση. Είναι απαραίτητο να χρησιμοποιήσετε παραπάνω από μια διεύθυνση μόνο αν το website σας ή η εφαρμογή χρησιμοποιεί την IP διεύθυνση για να ακολουθήσει τα sessions του χρήστη κλπ..

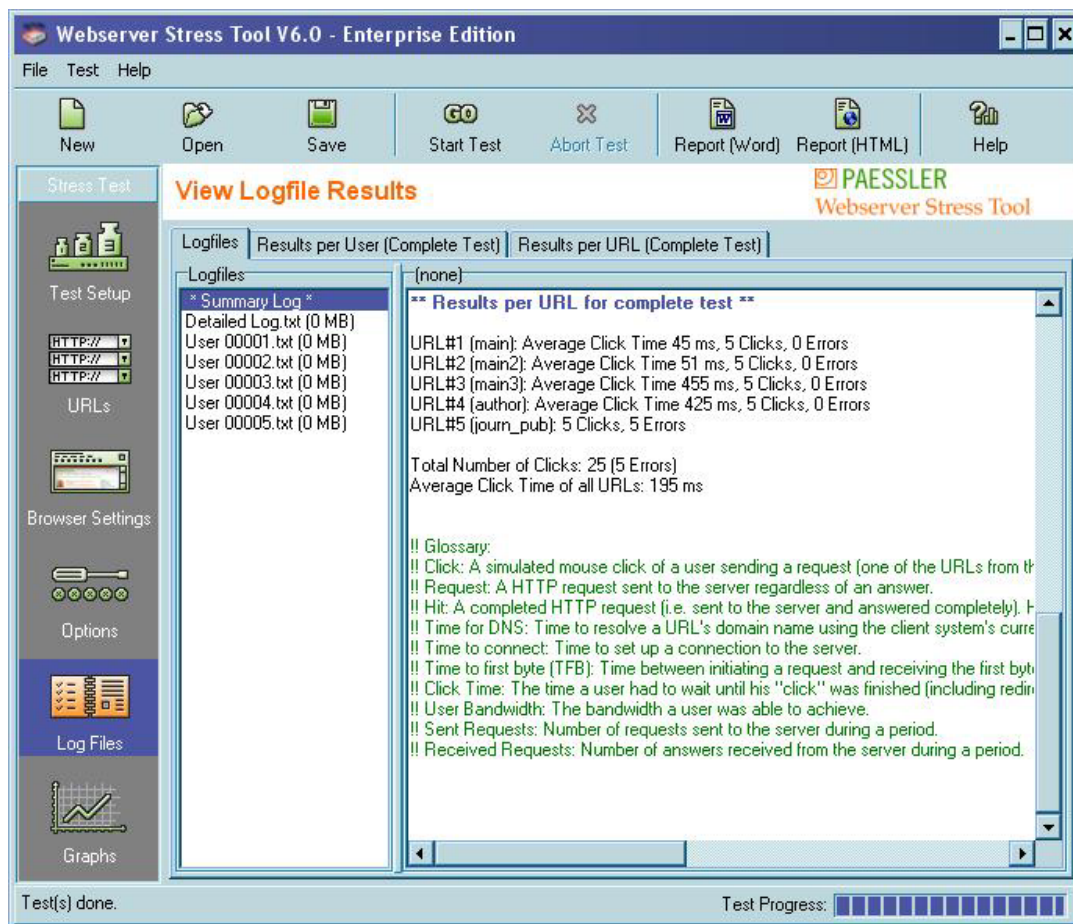
Αν έχουν επιλεγεί παραπάνω από μία IP διευθύνσεις, οι διευθύνσεις αυτές χρησιμοποιούνται κυκλικά για κάθε προσομοιωμένο χρήστη. Ο πρώτος χρήστης χρησιμοποιεί την πρώτη διεύθυνση για όλες του τις αιτήσεις, ο δεύτερος τη δεύτερη διεύθυνση κ.ο.κ..

### Timer

Χρησιμοποιώντας το χαρακτηριστικό **Start test at** μπορείτε να αναβάλλετε το τεστ για μια συγκεκριμένη ημερομηνία και ώρα.

## 📖 Τρέχοντας το τεστ

Αφού θέσετε τις επιθυμητές ρυθμίσεις, κάντε κλικ στο **Start Test** ώστε το Webserver Stress Tool να ξεκινήσει την εκτέλεση του τεστ.



Κατά τη διάρκεια του τεστ μπορείτε να κάνετε κλικ τριγύρω στο πρόγραμμα, αλλά δεν μπορείτε να αλλάξετε τις ρυθμίσεις του προγράμματος και του τεστ. Μπορείτε να βλέπετε επίσης τα αποτελέσματα του τεστ και κατά τη διάρκεια της εκτέλεσής του.

Όσην ώρα το τεστ είναι ενεργό, υπάρχει μια γραφική άποψη των προσομοιωμένων χρηστών στο κάτω μέρος του παραθύρου. Κάθε χρήστης απεικονίζεται με ένα ορθογώνιο, το οποίο χρωματίζεται ανάλογα με την κατάσταση του. Αυτό το γράφημα ενημερώνεται κάθε λίγα δευτερόλεπτα και γι' αυτόν το λόγο δεν μπορεί να δείχνει όλες τις δυνατές καταστάσεις για όλους τους χρήστες (αυτό θα μείωνε την επεξεργασία). Αλλά παρόλ' αυτά δίνει μια καλή εντύπωση του τι γίνεται.

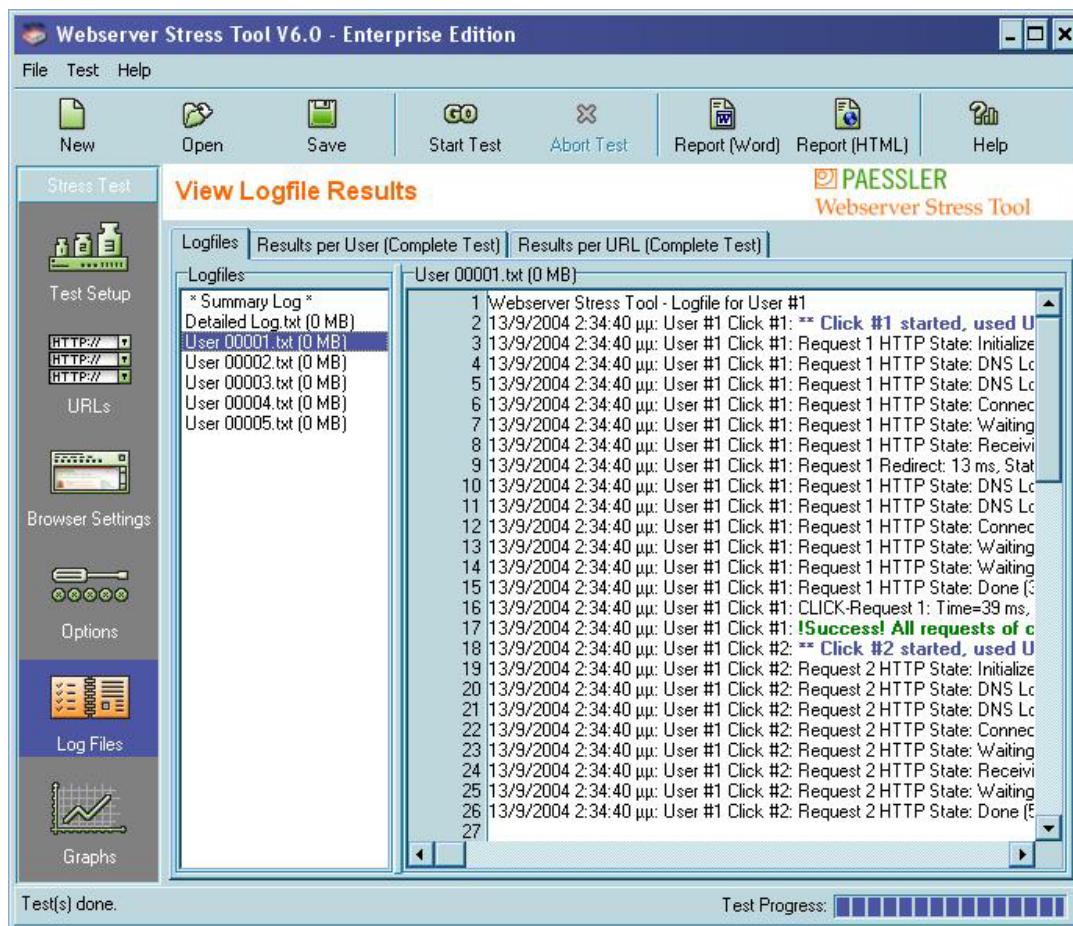
Επίσης παρατηρήστε τη γραμμή κατάστασης στο κάτω μέρος του παραθύρου για να πάρετε πληροφορίες κατάστασης σχετικά με το τεστ.

Κάνοντας κλικ στο **Abort Test** μπορείτε να σταματήσετε το τεστ σε οποιαδήποτε στιγμή.

Όταν τελειώσει το τεστ ακούγεται το ‘μπιπ’ του συστήματος και μπορείτε να δείτε τα αποτελέσματα.

## 📖 Επιθεώρηση των αποτελεσμάτων των Logfiles

Κάντε κλικ στο **Log Files** για να ανοίξετε το logfile browser.



Στα αριστερά θα βρείτε μια λίστα με τα διαθέσιμα logfiles. Απλώς κάντε κλικ σε μια από τις εγγραφές για να δείτε τα περιεχόμενα στα δεξιά.

Αν έχετε ενεργοποιημένη την επιλογή **Write HTML Files to disk**, μπορείτε επίσης εδώ να επιλέξετε όλα τα HTML αρχεία.

Όλα τα logfiles αποθηκεύονται στον υποκατάλογο “logs” του καταλόγου όπου περιέχεται το EXE αρχείο (συνήθως *C:\Program Files\Webservers Stress Tool 6\logs*).

## Summary Log και Detailed Log

Υπάρχουν δύο κύρια log files: το **Summary Log** και το **Detailed Log**.

Το Summary Log περιέχει μόνο τα πιο σημαντικά αποτελέσματα:

- Ώρα και ημερομηνία του τεστ
- Συνοπτικά αποτελέσματα για όλες τις περιόδους
- Συνοπτικά αποτελέσματα για ολόκληρο το τεστ
- Γλωσσάρι

Το Detailed Log (πρέπει να ενεργοποιηθεί στη σελίδα επιλογών) περιέχει όλα τα παραπάνω και επιπροσθέτως:

- Δεδομένα εγκατάστασης του τεστ (URLs, αριθμός χρηστών κλπ.)
- Πληροφορία επεξεργασίας του τεστ (π.χ. χρόνος αναμονής)
- Λεπτομερή αποτελέσματα για όλες τις περιόδους
- Ανεπιτυχείς αιτήσεις
- Γλωσσάρι
- Τοποθεσίες όπου έχουν αποθηκευτεί τα logfiles

Αυτό το logfile μπορεί να μεγαλώσει πάρα πολύ. Ανάλογα με το λειτουργικό σας σύστημα, το Webserver Stress Tool μπορεί να μην είναι σε θέση να εμφανίσει το logfile. Σε αυτήν την περίπτωση χρησιμοποιήστε έναν εξωτερικό editor.

Σημείωση: μεγάλα log files δεν μπορούν να ανοιχτούν σε μηχανήματα με Windows 95/98/ME.

Το Detailed Log γράφεται στιγμιαία στο δίσκο κατά τη διάρκεια του τεστ και γι' αυτόν το λόγο μπορεί να βοηθήσει στη διάγνωση προβλημάτων, σε περίπτωση ασυνήθιστου τερματισμού του τεστ.

Το Summary Log γράφεται στο δίσκο στο τέλος του τεστ.

## User Logs

Αν είναι ενεργοποιημένη αυτή η επιλογή στη σελίδα επιλογών, το Webserver Stress Tool γράφει ένα logfile για κάθε χρήστη. Αυτά τα logfiles περιέχουν:

- Δεδομένα για όλα τα clicks, frames, εικόνες, ανακατευθύνσεις, αιτήσεις κλπ.
- Time to first byte, Time to connect και άλλα παρόμοια δεδομένα για κάθε αίτηση



- Εναλλακτικά δεδομένα επικεφαλίδας, HTML και cookie, καθώς και image URLs και frame URLs

Επιλέξτε ένα user log που είναι στη λίστα των log files κάνοντας το κλικ και το αρχείο θα φανεί στο δεξί μέρος του παραθύρου.

### Αποτελέσματα για κάθε χρήστη

Η σελίδα **Results per User (Complete Test)** εμφανίζει τα αποτελέσματα για κάθε προσομοιωμένο χρήστη:

User No.	Clicks	Hits	Errors	Avg. Click Time [ms]	Bytes	kB/s	Cookies
1	5	5	1	196	31116	0,032	
2	5	5	1	12327	29674	0	
3	5	5	1	9727	31116	0,001	
4	5	5	1	12327	29674	0	
5	5	5	1	196	31116	0,032	

Τα δεδομένα που εμφανίζονται είναι τα συναθροισμένα αποτελέσματα του ολοκληρωμένου τεστ.

### Αποτελέσματα για κάθε URL

Ο αριθμός των hits, σφαλμάτων και χρήσης χρόνου κάθε URL εμφανίζεται στη σελίδα **Results per URL (Complete Test)**:

The screenshot shows the 'View Logfile Results' window in the Webrserver Stress Tool. The window title is 'Webrserver Stress Tool V6.0 - Enterprise Edition'. The menu bar includes 'File', 'Test', and 'Help'. The toolbar contains icons for 'New', 'Open', 'Save', 'Start Test', 'Abort Test', 'Report (Word)', 'Report (HTML)', and 'Help'. The main area displays a table of test results for a completed test. The table has columns for 'URL No.', 'Name', 'Clicks', 'Errors', 'Errors [%]', 'Time Spent [ms]', and 'Avg. Cl'. The data is as follows:

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Cl
1	main	5	0	0	225	45
2	main2	5	0	0	254	51
3	main3	5	0	0	2277	455
4	author	5	0	0	2125	425
5	journ pub	5	5	100	0	

The status bar at the bottom shows 'Test(s) done.' and a 'Test Progress' indicator.

Τα δεδομένα που εμφανίζονται είναι τα συναθροισμένα αποτελέσματα του ολοκληρωμένου τεστ.

## 📖 Ανάλυση Γραφικών Αποτελεσμάτων

Αυτή η παράγραφος περιγράφει τα διάφορα γραφήματα με τα αποτελέσματα του τεστ. Το Webrserver Stress Tool απεικονίζει τα αποτελέσματα κάθε τεστ με αρκετές γραφικές εικόνες, οι οποίες σας βοηθούν να βλέπετε και να καταλαβαίνετε τα αποτελέσματα του τεστ γρήγορα.

Μερικά από τα γραφήματα είναι σχεδιασμένα για την περιγραφή αποτελεσμάτων από περισσότερες από μια περιόδους (δηλαδή περισσότερα από ένα σημεία δεδομένων), έτσι αν βρείτε γραφήματα με πολύ λίγα σημεία δεδομένων (datapoints) πρέπει είτε να αυξήσετε το χρόνο του τεστ είτε να μειώσετε το χρόνο της καταγραφής των αποτελεσμάτων στο log. Σημειώστε ότι για τα Ramp tests με τον αυξανόμενο αριθμό χρηστών, ο οριζόντιος άξονας στα περισσότερα διαγράμματα είναι ο αριθμός των χρηστών. Διαφορετικά, στον οριζόντιο άξονα χρησιμοποιείται ο χρόνος που έχει περάσει από την έναρξη του τεστ.

## Χρήση των γραφημάτων

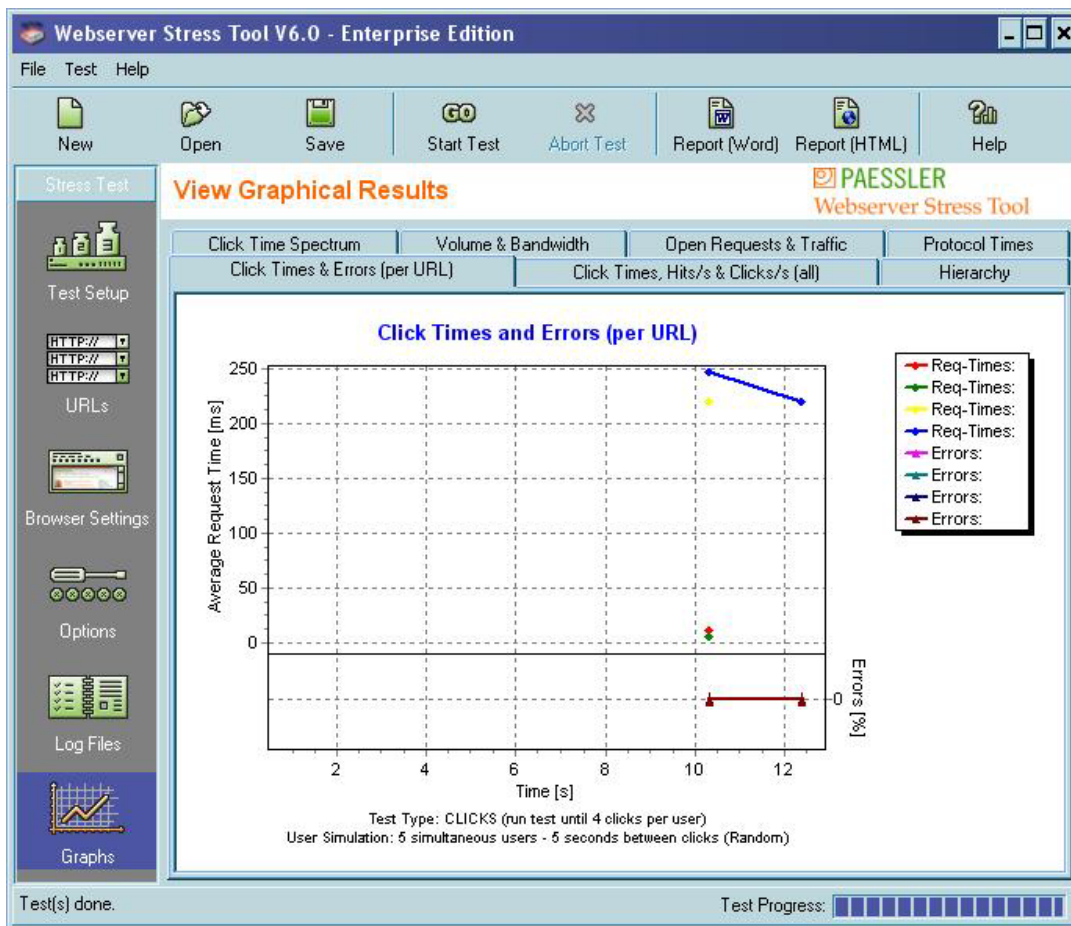
Στα περισσότερα γραφήματα μπορείτε να κάνετε zoom in κάνοντας drag το ποντίκι (με αριστερό κλικ πατημένο), από πάνω αριστερά έως κάτω δεξιά. Για να κάνετε zoom out, κάντε drag το ποντίκι από κάτω δεξιά έως πάνω αριστερά.

Χρησιμοποιώντας το context menu (δεξί κλικ στο γράφημα) κάθε γραφήματος, μπορείτε να αντιγράφετε το γράφημα στο clipboard και να το σώζετε στο δίσκο ως αρχείο εικόνας.

## Click Times & Errors (per URL)

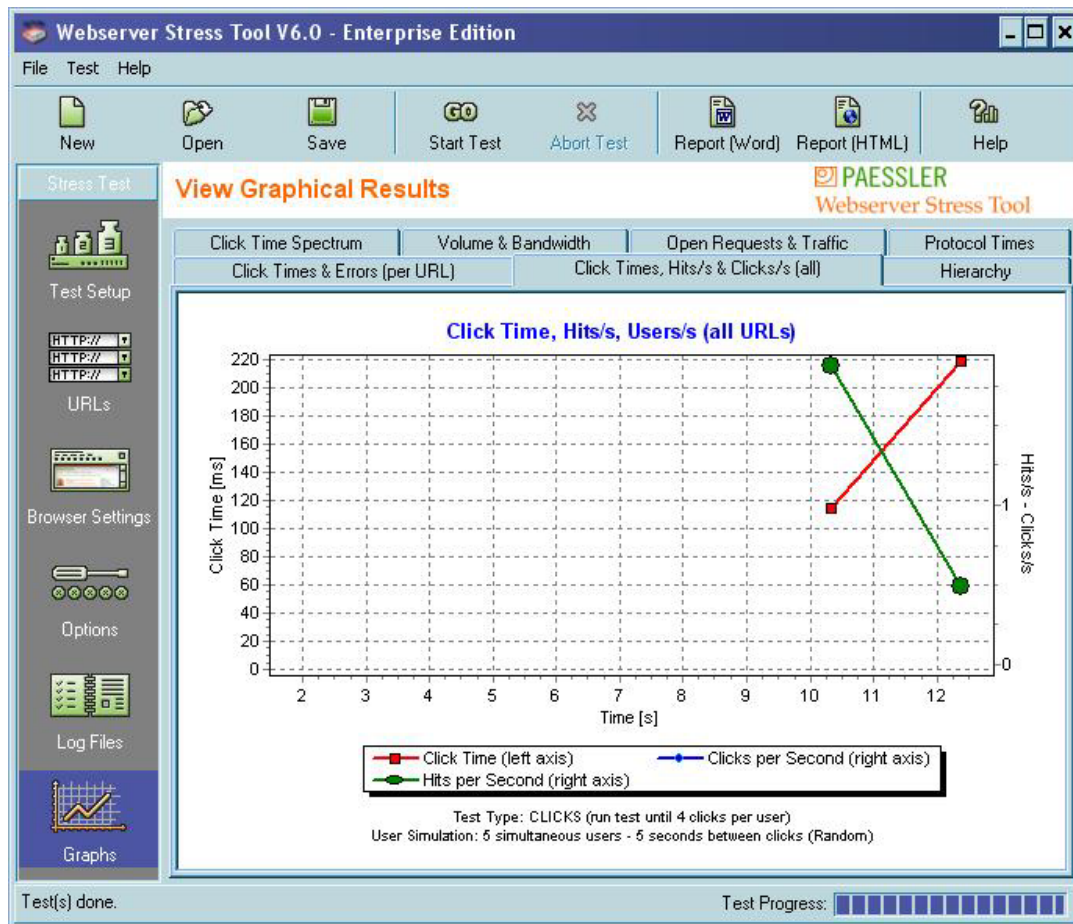
Για κάθε URL αυτό το γράφημα δείχνει τους χρόνους αίτησης των clicks και το ποσοστό των σφαλμάτων.

Το γράφημα δείχνει μια γραμμή για τους μέσους χρόνους click για κάθε URL και για κάθε περίοδο και μια γραμμή για κάθε URL που δείχνει το ποσοστό των εσφαλμένων clicks (δηλαδή των clicks με τουλάχιστον μια εσφαλμένη αίτηση).



## Click Times, Hits/s, Clicks/s

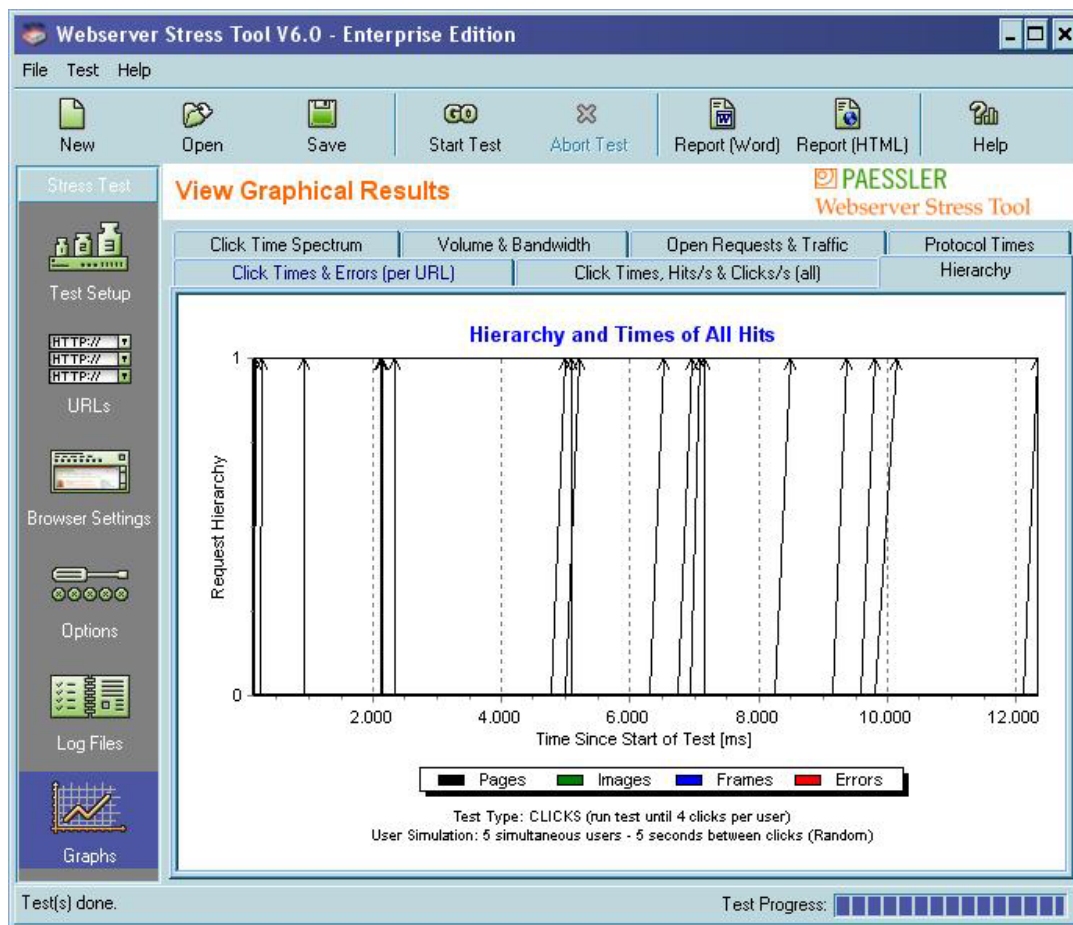
Ο μέσος χρόνος που περίμενε ένας χρήστης για την αίτησή του ώστε να υποστεί επεξεργασία (συμπεριλαμβανομένων των ανακατευθύνσεων, εικόνων, frames κλπ.), τα hits ανά δευτερόλεπτο και οι χρήστες ανά clicks ανά δευτερόλεπτο φαίνονται σ' αυτό το γράφημα. Η διαφορά με το προηγούμενο γράφημα είναι ότι αυτήν τη φορά οι τιμές υπολογίζονται για όλα τα URLs μαζί.



## Ιεραρχία Γραφήματος

Για κάθε προσομοιωμένο hit που στέλνει το Webservers Stress Tool στο server εμφανίζεται ένα τόξο σ' αυτό το διάγραμμα.

Κάθε τόξο συμβολίζει ένα hit (δηλαδή μια αίτηση HTTP). Τα μαύρα τόξα είναι σελίδες (δηλαδή αρχεία HTML), τα πράσινα τόξα συμβολίζουν εικόνες, τα μπλε frames και τα κόκκινα τόξα δείχνουν εσφαλμένες αιτήσεις.

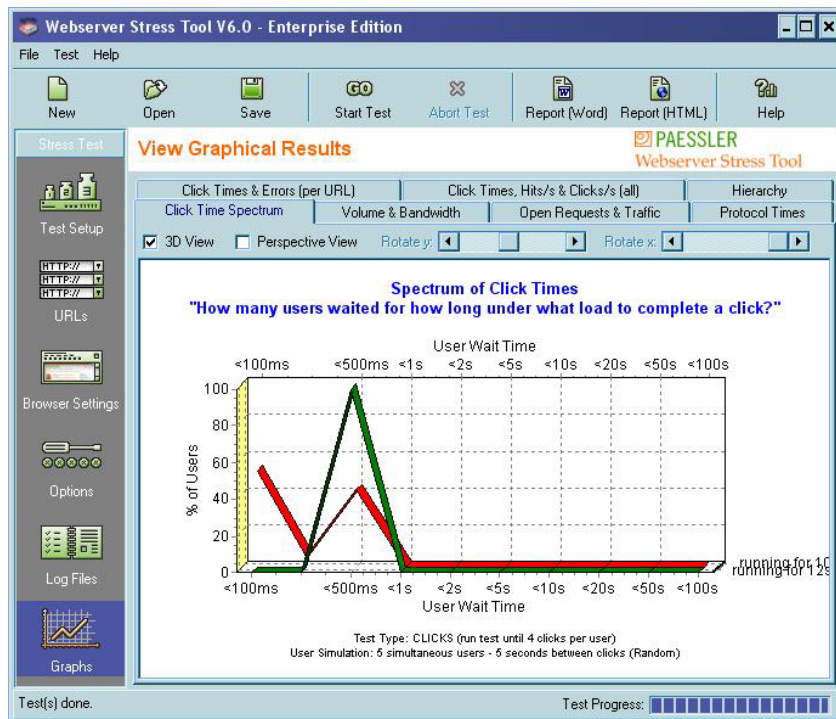


Όσο περισσότερο διαρκέσει μια αίτηση, τόσο μακρύτερα καταλήγει το τόξο. Μόλις ληφθεί το κείμενο HTML μιας αίτησης σελίδας, ζητώνται οι εικόνες από το server, οι οποίες φαίνονται στο διάγραμμα με πράσινα τόξα.

### **Spectrum (φάσμα) of Click Times**

Αυτό το γράφημα δείχνει, για κάθε τρέξιμο, την κατανομή των χρόνων αναμονής χρήστη. Δηλαδή δείχνει σε ποιο ποσοστό χρηστών (κάθετος άξονας) αντιστοιχεί ένα διάστημα χρόνου αναμονής (οριζόντιος άξονας) και ακόμη πιο απλά, πόσο χρόνο πρέπει να περιμένει το μεγαλύτερο ποσοστό των χρηστών.

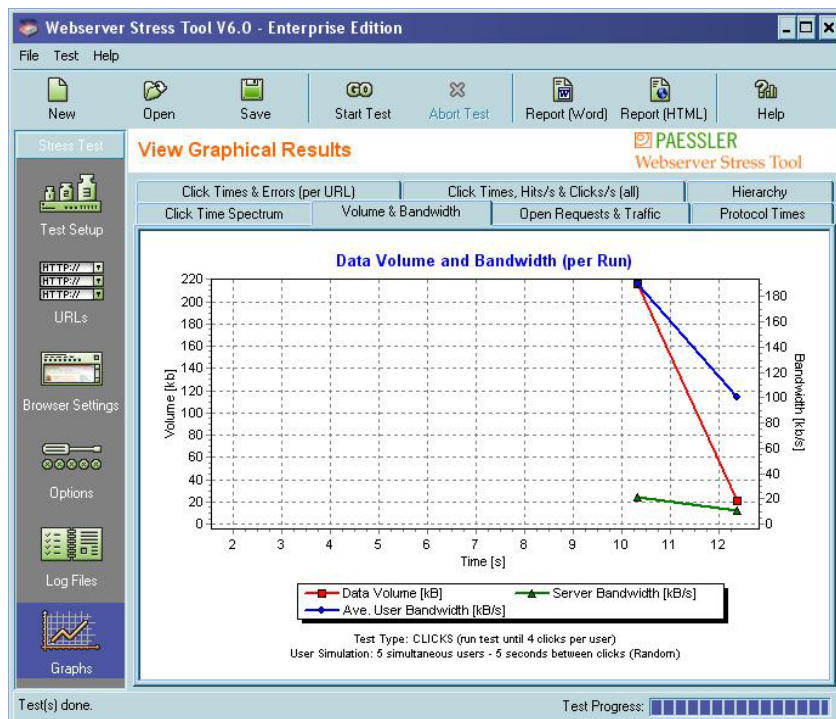
Το όφελος αυτού του τεστ στο λεγόμενο capacity planning (σχεδιασμός χωρητικότητας) είναι ξεκάθαρο. Θεωρήστε ως στόχο, ο μέγιστος χρόνος απόκρισης κάθε χρήστη να είναι 10 δευτερόλεπτα ή μικρότερος. Έχοντας αυτόν το στόχο στο μυαλό, πρέπει να σιγουρευτείτε ότι το γράφημά σας έχει το μέγιστό του στο “<10s” ή σε μικρότερο.



## Data Volume & Bandwidth

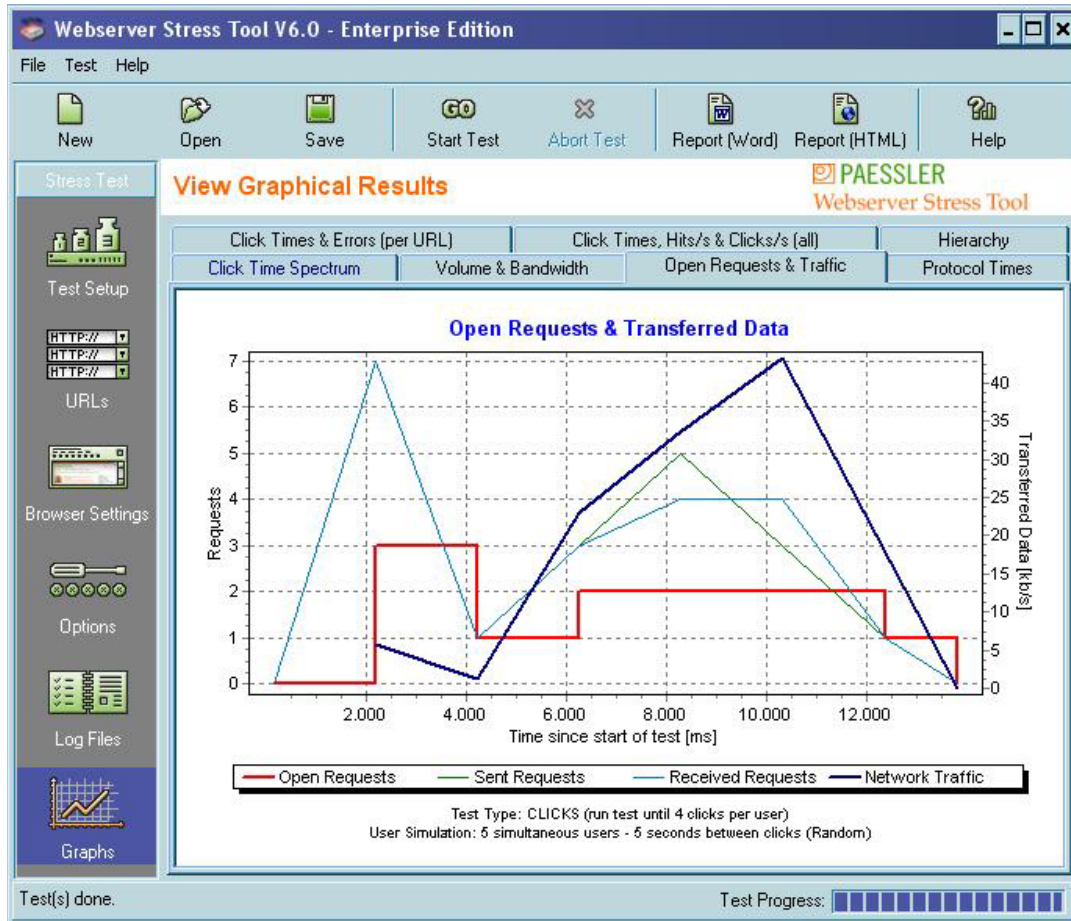
Ο όγκος των δεδομένων που στάλθηκαν από το σερβερ και το εύρος ζώνης που απαιτείται για να μεταφερθεί αυτός ο όγκος φαίνεται σ' αυτό το διάγραμμα.

Επίσης φαίνεται και το μέσο εύρο ζώνης που έλαβαν όλοι οι χρήστες.



## Open Requests

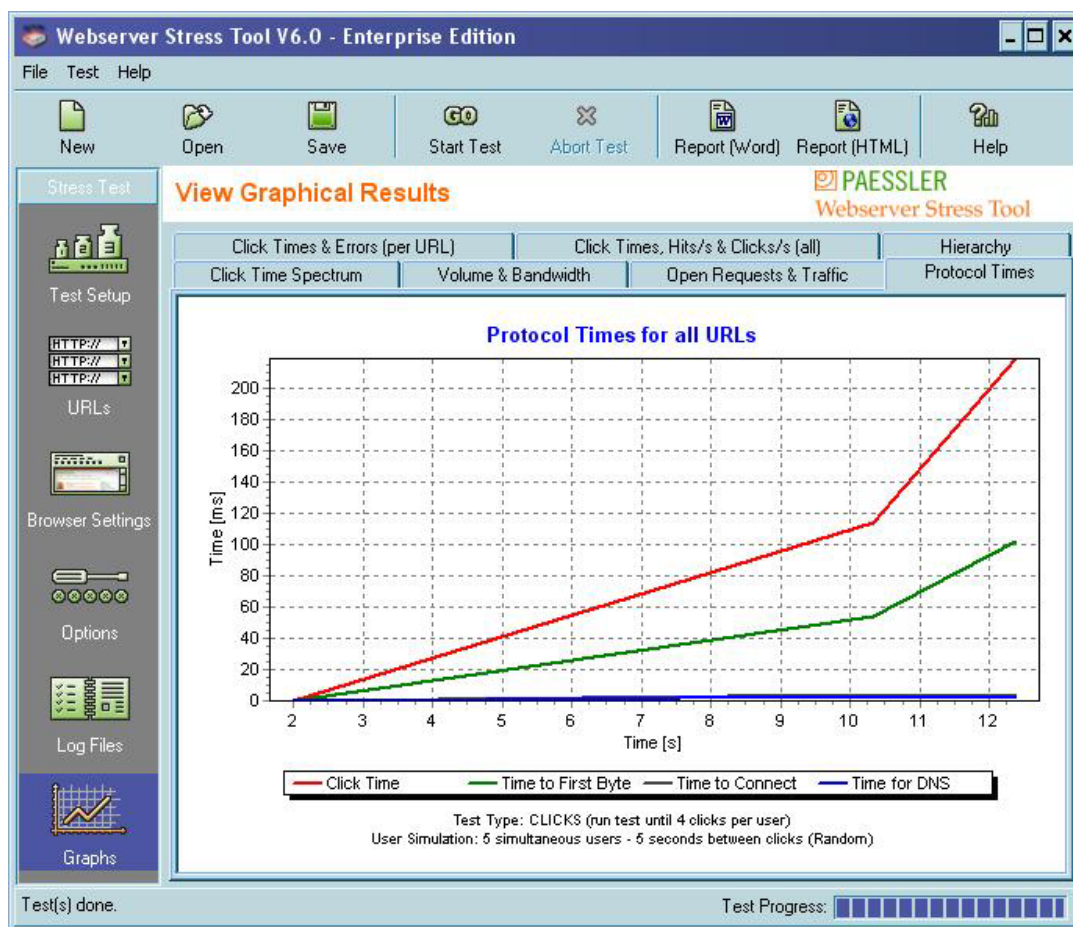
Ο αριθμός των ανοιχτών (=αναπάντητων) αιτήσεων, ο αριθμός των σταλμένων και ληφθέντων αιτήσεων και η κυκλοφορία του δικτύου φαίνονται σ' αυτό το γράφημα.



## Protocol Times

Μια αίτηση HTTP αποτελείται από πολλά στάδια. Πρώτα το όνομα του server πρέπει να αντιστοιχηθεί σε μια IP διεύθυνση χρησιμοποιώντας το DNS (χρόνος για DNS), έπειτα ανοίγεται ένα IP Port στο server από τον client για να στείλει την επικεφαλίδα της αίτησης (Time to Connect). Ο server μετά απαντά στην αίτηση (Time to First Byte) και στέλνει όλα τα δεδομένα. Όταν όλα τα δεδομένα έχουν μεταφερθεί η αίτηση τερματίζεται (Click Time).

Ο μέση τιμή αυτών των 4 αναγνώσεων για όλες τις αιτήσεις φαίνεται σ' αυτό το διάγραμμα.



## 📖 Δημιουργία Αναφορών (Reports)

Το Webserver Stress Tool προσφέρει δύο μεθόδους εξαγωγής των αποτελεσμάτων.

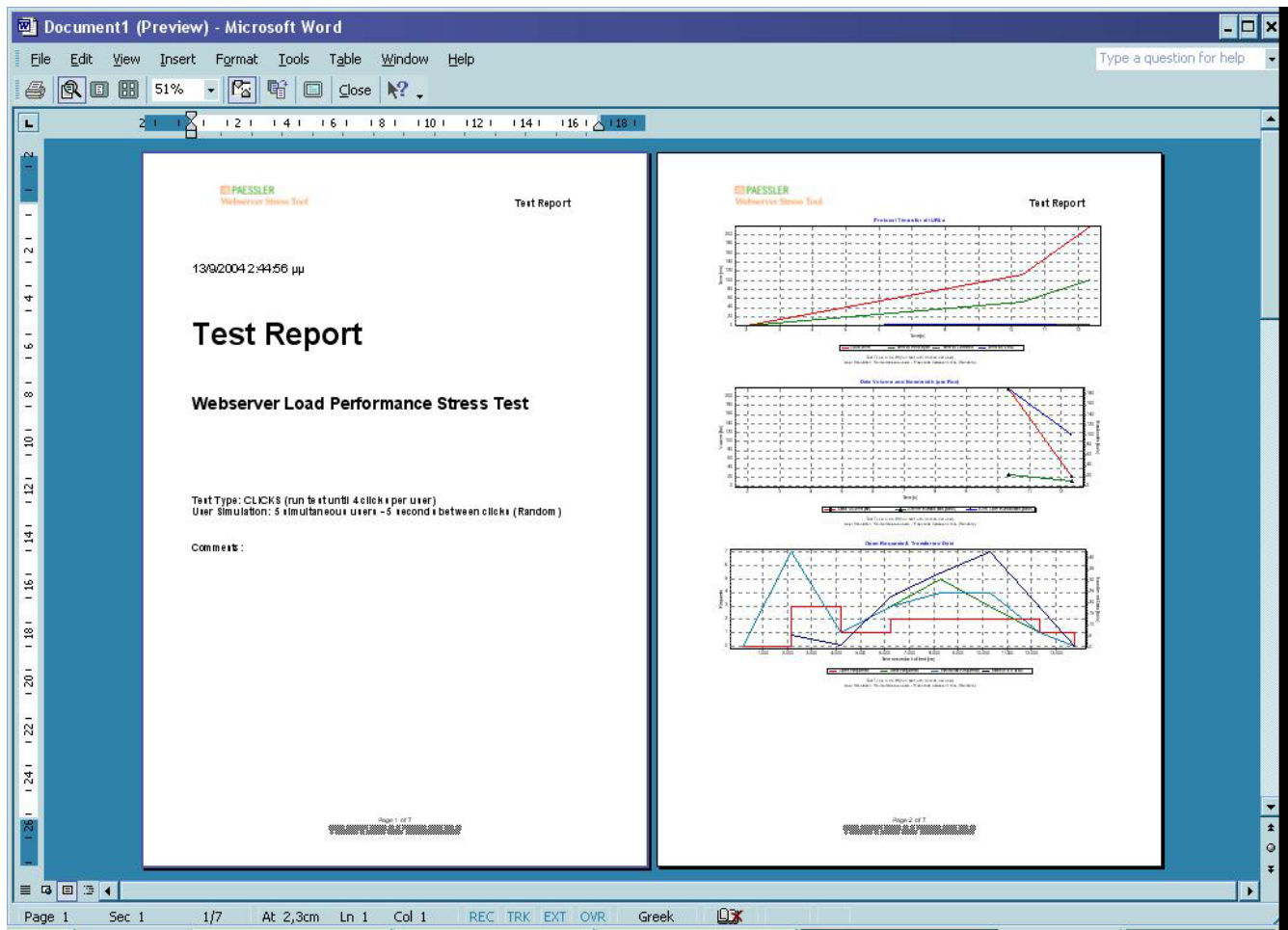
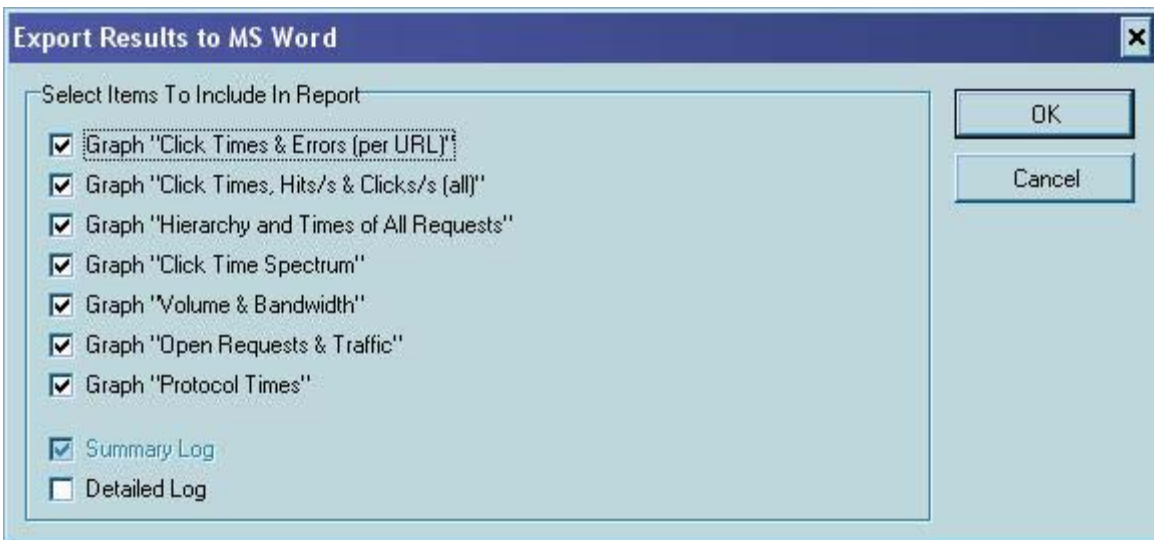
Μπορείτε να εξάγετε όλη την πληροφορία από τα αποτελέσματα σε ένα έγγραφο MS Word και μπορείτε να δημιουργήσετε και έναν αριθμό αρχείων HTML.

### Αναφορά (Word)

Αν έχετε εγκατεστημένο το MS Office στον υπολογιστή σας, κάντε κλικ στο **Report (Word)** μόλις τελειώσει το τεστ.

Επιλέξτε τα δεδομένα που θέλετε να συμπεριληφθούν στην αναφορά. Μόλις κάνετε κλικ στο **OK**, το Microsoft Word ξεκινάει χρησιμοποιώντας το OLE και δημιουργείται η αναφορά. Μερικά δευρόλεπτα αργότερα μπορείτε να τροποποιήσετε, να εκτυπώσετε και να αποθηκεύσετε την αναφορά χρησιμοποιώντας όλες τις λειτουργίες του Microsoft Word.





## Αναφορά (HTML)

Κάντε κλικ στο **Report (HTML)** για να δημιουργήσετε ένα σύνολο αρχείων HTML και εικόνας με τα αποτελέσματα του τεστ. Θα βρείτε ένα μενού στα αριστερά, κάντε κλικ σε μια εγγραφή για να δείτε την λεπτομερή πληροφορία.

The screenshot shows a Microsoft Internet Explorer browser window displaying the 'Webservers Stress Tool - Report' in HTML format. The browser's address bar shows the file path: `C:\Program Files\Webserver Stress Tool 6 (Enterprise Edit.)\logs\report.html`. The page content is organized into several sections:

- Test Report**: Includes sub-sections for Settings, Test Configuration, URLs, Results and Graphs, Click Times and Errors, Click Time, Hits/s & Clicks/s, Protocol Times, Hierarchy, Open Requests & Traffic, Volume & Bandwidth, Click Time Spectrum, Results per User, Results per URL, and Logfiles. Under Logfiles, there are links to 'Detailed Log.txt (0 MB)', 'Summary Log.txt (0 MB)', and five individual user log files (User 00001.txt to User 00005.txt, each 0 MB).
- Project and Scenario Comments, Operator**: Contains a 'Test Setup' section with details like 'Test Type: CLICKS (run test until 4 clicks per user)', 'User Simulation: 5 simultaneous users - 5 seconds between clicks (Random)', and 'Logging Period: Log every 10 seconds'. It also includes 'URLs' and 'Browser Settings'.
- Client System**: Provides system information such as 'System: Windows XP V5.1 (Build 2600) Service Pack 1, CPU Proc. Type 586 (Rev. 516) at 2533 MHz', 'Memory: 230 MB available RAM of 536 MB total physical RAM, 1046 MB available pagefile, 49769 MB free disk space on C:', 'Client's MAC address: B2-5A-2B-B1-45-DC', and 'Time measurement resolution: 2,793651 µsec, clock runs at 4MHz'.

The browser's status bar at the bottom shows 'Done' and 'My Computer'.

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# **ΜΕΡΟΣ ΔΕΥΤΕΡΟ**

**ΠΕΙΡΑΜΑΤΙΚΟ ΜΕΡΟΣ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 7

**BOTTLENECKS ΕΠΕΞΕΡΓΑΣΤΗ**

**Α**υτό το κεφάλαιο περιέχει την πληροφορία που χρειάζεστε για να καταλάβετε τη σημασία του **PTO (Performance Tuning & Optimization)** και πώς αυτό επηρεάζει την απόδοση του server. Ο κύριος σκοπός του κεφαλαίου είναι να σας παρέχει έναν καλό λόγο για να βελτιστοποιήσετε το σύστημά σας.

---

## Τι είναι το bottleneck επεξεργαστή;

Ένα bottleneck επεξεργαστή προκύπτει όταν μία ή περισσότερες διεργασίες καταλαμβάνει σχεδόν όλο το χρόνο του επεξεργαστή στον υπολογιστή. Κατά το bottleneck, τα νήματα διεργασίας που είναι έτοιμα να εκτελεστούν πρέπει να περιμένουν σε μια ουρά για χρόνο επεξεργαστή. Οποιαδήποτε άλλη δραστηριότητα παύεται μέχρι να αδειάσει η ουρά. Με άλλα λόγια, άλλα υποσυστήματα, όπως ο σκληρός δίσκος, μένουν αδρανή όσο περιμένουν τον επεξεργαστή να εκτελέσει ένα δοθέν σύνολο εργασιών. Τα bottlenecks επεξεργαστή επηρεάζουν ολόκληρο το σύστημα, καθώς ο επεξεργαστής είναι απαραίτητος για την επεξεργασία σχεδόν κάθε εργασίας.

**Σημείωση:** Δεν μπορούμε να δημιουργήσουμε εκούσια ένα bottleneck επεξεργαστή, επομένως δεν μπορούμε και να το προσομοιώσουμε. Έτσι, θα προσπαθήσουμε απλά να σας κατευθύνουμε ώστε να μπορέσετε να αντιμετωπίσετε ένα bottleneck επεξεργαστή, αν και όταν αυτό σας παρουσιαστεί.

---

## Ενδείξεις για ανίχνευση ενός bottleneck επεξεργαστή

Σε ένα μηχάνημα ενός μοναδικού χρήστη, μπορείτε να ανιχνεύσετε κατά ένα μέρος ένα bottleneck επεξεργαστή από το ποσοστό καθυστέρησης που βιώνετε στο χρόνο απόκρισης του συστήματος. Σε ένα server, ο χρόνος απόκρισης μπορεί να μην είναι ένας σωστός ενδείκτης του φορτίου επεξεργαστή του server αυτού. Εξωτερικές επιρροές, όπως η διαθεσιμότητα του εύρους ζώνης του δικτύου, μπορεί να δώσουν λάθος εντύπωση σ'έναν διαχειριστή. Έτσι, επειδή κάποιες ενδείξεις μπορεί να είναι λανθασμένες σχετικά με την καθυστέρηση του συστήματος, οι περισσότεροι διαχειριστές 'νιώθουν' ένα bottleneck επεξεργαστή εμπειρικά. Κάποιοι εμπειρικοί κανόνες που θα σας βοηθήσουν να ανιχνεύσετε ένα bottleneck επεξεργαστή, παρατίθενται παρακάτω:

- **Συνέχεια (Consistency)**

Σε αντίθεση με ένα σταθμό εργασίας, όπου οι λίστες των εφαρμογών συνεχώς αλλάζουν, οι servers έχουν να κάνουν κανονικά με μια μόνο εφαρμογή η οποία

τρέχει συνεχώς. Ένας διαχειριστής δικτύου μπορεί να ανιχνεύσει bottlenecks επεξεργαστή ψάχνοντας για συνεχή προβλήματα αργής επεξεργασίας. Το επόμενο βήμα είναι να διαπιστώσει αν η εφαρμογή αυτή είναι του τύπου που δημιουργεί προβλήματα επεξεργαστή. Οι εφαρμογές αυτές λέγονται processor-bound και περιγράφονται στην επόμενη υποενότητα.

- **Προσδιορισμός εφαρμογής**

Είναι σημαντικό να προσδιοριστούν ποιες εφαρμογές έτρεχαν όταν οι χρήστες ανέφεραν προβλήματα σχετικά με την ταχύτητα επεξεργασίας του server. Σε πολλές περιπτώσεις θα δείτε μια προβληματική εφαρμογή. Φυσικά, πρέπει να καθορίσετε αν η εφαρμογή έχει σφάλματα (π.χ. αν έχει φτωχή διαχείριση πόρων), αν είναι υπερφορτωμένη από πάρα πολλούς χρήστες, ή αν όντως έχετε bottleneck επεξεργαστή. Ο προγραμματιστής της εφαρμογής μπορεί συνήθως να σας παρέχει προδιαγραφές και οδηγίες για την εφαρμογή, ώστε να γίνει η διαδικασία της ανίχνευσης πιο εύκολη.

- **Προσδιορισμός εργασίας**

Η τεχνολογία των components επιτρέπει σε μια εφαρμογή να εκτελείται σε περισσότερα από ένα μηχανήματα. Το COM+ διευρύνει αυτήν την ιδέα επιτρέποντας σε μια μοναδική εφαρμογή να εκτελείται σε περισσότερους από έναν servers. Έτσι λοιπόν δεν είστε πλέον σε θέση να αναγνωρίσετε εύκολα μια εφαρμογή ή τα μηχανήματα στα οποία τρέχει η εφαρμογή. Θα πρέπει να καταλάβετε πώς ολοκληρώνουν οι χρήστες τις εργασίες τους, ποια components χρησιμοποιούνται στις επηρεαζόμενες εφαρμογές και πού βρίσκονται αυτά τα components στους servers, ώστε να ανιχνεύσετε ένα bottleneck επεξεργαστή.

- **Εξωτερικοί ενδείκτες**

Αυτοί είναι οι ενδείκτες ότι κάτι πάει στραβά στο δίκτυο. Για παράδειγμα, μπορεί να έχετε μια εφαρμογή που κανονικά τρέχει καλά, αλλά ξαφνικά αρχίζει να έχει προβλήματα αφού εγκαταστήσετε ένα νέο server ή αναβαθμίσετε μια άλλη εφαρμογή. Οποιαδήποτε αλλαγή, θετική ή αρνητική, είναι ενδείκτης ότι κάτι πάει στραβά στο background το οποίο πρέπει να γνωρίζετε.

- **Χαρακτηριστικά εφαρμογών (features)**

Ένα bottleneck επεξεργαστή μπορεί να μην εμφανιστεί αμέσως αφού εγκαταστήσετε την εφαρμογή ή αφού αλλάξετε τις ρυθμίσεις του δικτύου. Σε πολλές περιπτώσεις, μόνο ένα ή δύο χαρακτηριστικά μιας εφαρμογής μπορούν να προκαλέσουν προβλήματα απόδοσης που αλλάζουν σημαντικά τον τρόπο που τρέχει το δίκτυο. Για να σας δώσουμε μια ιδέα σχετικά με το πώς δουλεύει αυτό σε μια κατάσταση πραγματικού κόσμου, σκεφτείτε τι συμβαίνει αφού εγκαταστήσετε μια πολύπλοκη εφαρμογή στο δίκτυο. Το πιο πιθανό είναι οι

χρήστες της εφαρμογής να ξεκινήσουν δουλεύοντας με τα πιο απλά χαρακτηριστικά. Ωστόσο, ενώ οι χρήστες αποκτούν κάποια πείρα σχετικά με την εφαρμογή ή εκπαιδεύονται, χρησιμοποιούν μερικά από τα πιο πολύπλοκα χαρακτηριστικά της εφαρμογής. Είναι σε αυτό το σημείο όπου μπορεί να αρχίσετε να βλέπετε μια καθυστέρηση στο δίκτυο.

---

## Εφαρμογές process-bound

Μερικές εφαρμογές απαιτούν περισσότερο χρόνο επεξεργαστή από άλλες. Ένα πολύ κοινό παράδειγμα είναι η σύγκριση μιας λογιστικής εφαρμογής και μιας εφαρμογής διαχείρισης βάσης δεδομένων. Η λογιστική εφαρμογή χρησιμοποιεί πάντα πολλούς κύκλους επεξεργασίας επειδή εκτελεί πολλούς υπολογισμούς, έτσι χαρακτηρίζεται ως process-bound. Η εφαρμογή βάσης δεδομένων, αντίθετα, χαρακτηρίζεται ως disk-bound επειδή χρησιμοποιείται για να διαχειρίζεται δεδομένα. Ο περισσότερος χρόνος της εφαρμογής αυτής ξοδεύεται μεταφέροντας δεδομένα από και προς το δίσκο.

**Σημείωση: Η ASP εφαρμογή που θα χρησιμοποιήσουμε παρακάτω στα πειράματα είναι disk-bound αφού βασίζεται στη διαχείριση μιας βάσης δεδομένων και δεν απαιτεί υπολογισμούς.**

Τα παρακάτω βήματα μπορούν να σας βοηθήσουν να διαπιστώσετε αν μια εφαρμογή είναι process-bound ή όχι, ανεξάρτητα από το πόσο πολύπλοκη είναι αυτή:

1. Σπάστε την εφαρμογή στα επιμέρους components της.
2. Καθορίστε το τι κάνει η κάθε εργασία. Ένας καλός προγραμματιστής θα μπορούσε να σας βοηθήσει να καταλάβετε την εφαρμογή (π.χ. με ένα καλώς ορισμένο block diagram).
3. Καθορίστε την τοποθεσία στην οποία βρίσκεται το κάθε component.
4. Αποφασίστε αν οι εργασίες που εκτελούνται από ένα σύνολο components σε έναν συγκεκριμένο server είναι process-, memory-, ή disk-bound. Οι processor-bound εργασίες απαιτούν συνήθως την εκτέλεση κάποιων υπολογισμών. Οποιοδήποτε component εκτελεί ανάλυση δεδομένων, δουλεύει με γραφικά ή χειρίζεται δεδομένα στη μνήμη, είναι processor-intensive.
5. Επιβεβαιώστε τις υποθέσεις σας παρακολουθώντας τα χαρακτηριστικά απόδοσης των components, κάνοντας χρήση του System Monitor.
6. Παρακολουθήστε την απόδοση της εφαρμογής μέσα σε μια περίοδο για να δείτε τη χρησιμοποίηση του επεξεργαστή.

---

## 📖 Ανάλυση δραστηριότητας επεξεργαστή

Ο παρακάτω πίνακας παρουσιάζει τους μετρητές που πρόκειται να χρησιμοποιήσουμε στα πειράματα αυτής της υποενότητας:

Μετρητές	Ενδείξεις
<i>System\ Processor Queue Length</i>	Τα threads περιμένουν για χρόνο επεξεργαστή. Αν η τιμή αυτή υπερβαίνει το 2 για παρατεταμένη περίοδο, τότε ο επεξεργαστής μπορεί να έχει bottleneck.
<i>Processor\ % Processor Time</i>	Το ποσοστό της CPU που καταναλώνεται τη συγκεκριμένη στιγμή.
<i>Processor\ % Privileged Time</i>	Τμήμα του χρόνου επεξεργαστή που καταναλώνεται κατά το privileged mode του συστήματος. Στο privileged mode εκτελούνται μόνο οι ρουτίνες του λειτουργικού συστήματος (π.χ. interrupts, I/O routines, κλπ.) και όχι του χρήστη (π.χ. εφαρμογές).
<i>Processor\ %DPC Time</i>	Το ποσοστό του χρόνου επεξεργαστή που καταναλώνεται από τα DPCs.
<i>Processor\ %Interrupt Time</i>	Το ποσοστό του χρόνου επεξεργαστή που καταναλώνεται από τα Interrupts.
<i>Processor\ Interrupts/sec</i>	Ο αριθμός των interrupts ανά δευτερόλεπτο.
<i>Web Service\ Total Method Requests/sec</i>	Ο αριθμός των HTTP αιτήσεων το δευτερόλεπτο.

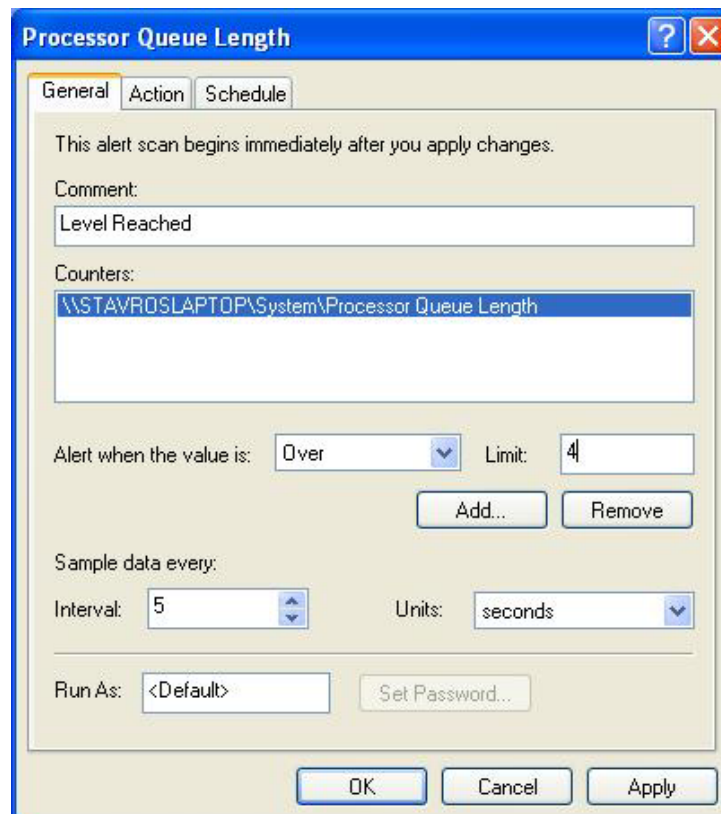
Από τους παραπάνω μετρητές, ο **System\Processor Queue Length** είναι μάλλον ο πιο σημαντικός για την ανάλυση της δραστηριότητας του επεξεργαστή. Αυτός ο μετρητής δείχνει τον αριθμό των νημάτων που περιμένουν για επεξεργασία μέσα στην μοναδική ουρά που διαμοιράζονται όλοι οι επεξεργαστές. Όταν ο μετρητής αυτός παρουσιάζει υψηλές τιμές (πάνω από 2 νήματα) για παρατεταμένο διάστημα, τότε αυτό σημαίνει ότι τα νήματα παραμένουν στην ουρά και περιμένουν επειδή ο επεξεργαστής δεν μπορεί να χειριστεί το φορτίο που του έχει ανατεθεί.



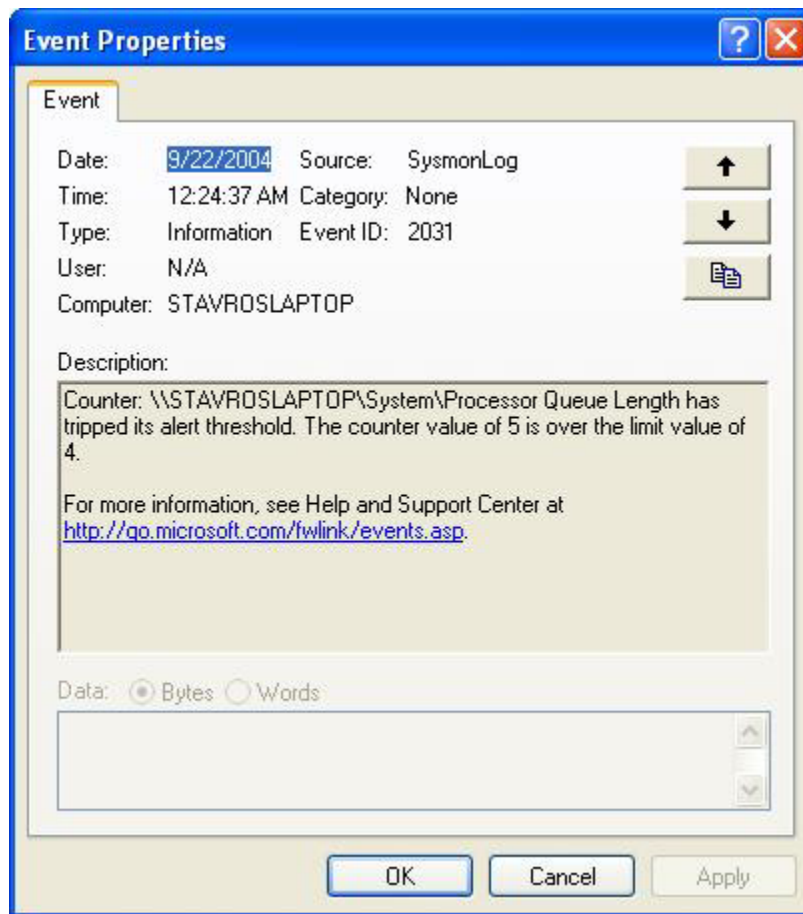
Έτσι λοιπόν, όταν υπάρχει μεγάλος αριθμός νημάτων στην ουρά για μεγάλο διάστημα, ενώ ταυτόχρονα η χρησιμοποίηση της CPU είναι κοντά στο 100%, τότε το πιο πιθανό είναι να υπάρχει bottleneck επεξεργαστή. Μπορείτε να θέσετε ένα Alert στο System Monitor ώστε να σας ειδοποιεί όταν η ουρά του επεξεργαστή ξεπερνά μια μη αποδεκτή τιμή.

Για να θέσουμε ένα Alert ακολουθούμε τα εξής βήματα:

1. Ανοίγουμε το System Monitor από *Start* → *Control Panel* → *Administrative Tools* → *Performance*
2. Κάνουμε κλικ στο σταυρό αριστερά του Performance Logs and Alerts και κάνουμε κλικ στο Alerts.
3. Δεξιά στο παράθυρο κάνουμε δεξί κλικ, επιλέγουμε New Alert Settings... και βάζουμε ένα όνομα, π.χ. Processor Queue Length.
4. Εισάγουμε ένα σχόλιο στο νέο παράθυρο διαλόγου που ανοίγει, π.χ. Level Reached.
5. Εισάγουμε (Add...) τον μετρητή System\Processor Queue Length.
6. Βάζουμε ένα όριο 4 και αφήνουμε την επιλογή Over ώστε το Alert να μας ειδοποιήσει όταν η τιμή του μετρητή ξεπεράσει το όριο.
7. Εισάγουμε ένα χρονικό διάστημα δειγματοληψίας, π.χ. 5 δευτερόλεπτα.
8. Πατάμε OK.



9. Για να δούμε τα όποια Alerts που μπορεί να έχουν πυροδοτηθεί, ανοίγουμε το Event Viewer που βρίσκεται στο ίδιο path με το Performance.



Για κάθε αίτηση σύνδεσης που φθάνει στο server μέσω του δικτύου, η CPU πρέπει να επεξεργαστεί τα πακέτα της πληροφορίας, μεταφέροντάς τα από το ένα επίπεδο στο άλλο, μέχρις ότου αυτά φθάσουν στο επίπεδο εφαρμογών, όπου η πληροφορία περνάει στον IIS. Αυτό σημαίνει ότι η CPU χρησιμοποιείται από τον IIS έμμεσα, καθώς πρέπει να επεξεργάζεται κάθε πακέτο που φθάνει στον IIS μέσω του δικτύου.

Καθώς αυξάνεται το φορτίο, αυξάνεται επίσης και η χρησιμοποίηση της CPU. Ο περισσότερος χρόνος της CPU ξοδεύεται στο privileged mode. Για κάθε αίτηση HTTP, ο IIS αναλύει την αίτηση αυτή, εντοπίζει το αρχείο και το στέλνει στο δίκτυο. Το διάβασμα του αρχείου και το γράψιμο στη γραμμή δικτύου διεκπεραιώνεται από το λειτουργικό σύστημα στο privileged mode.

---

## Test #1

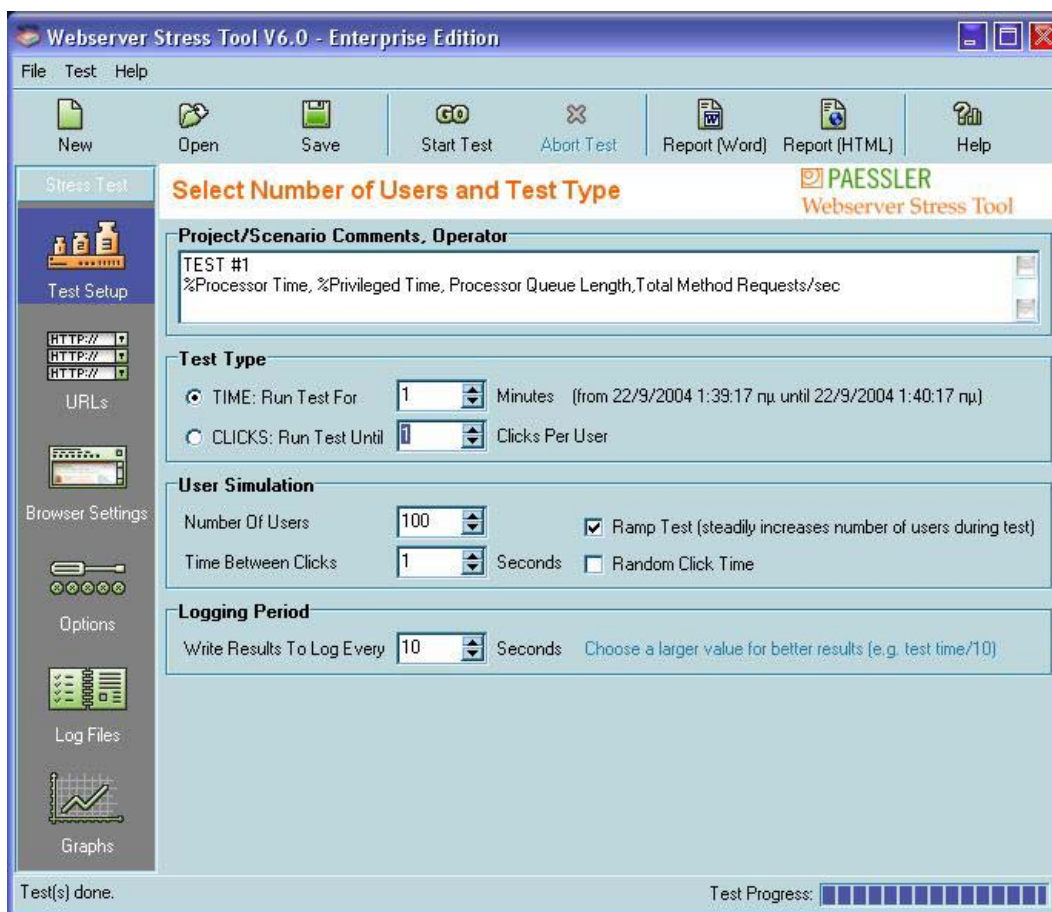
Θα πραγματοποιήσουμε ένα Ramp test στον client και θα μετρήσουμε τους εξής μετρητές στο System Monitor του server:

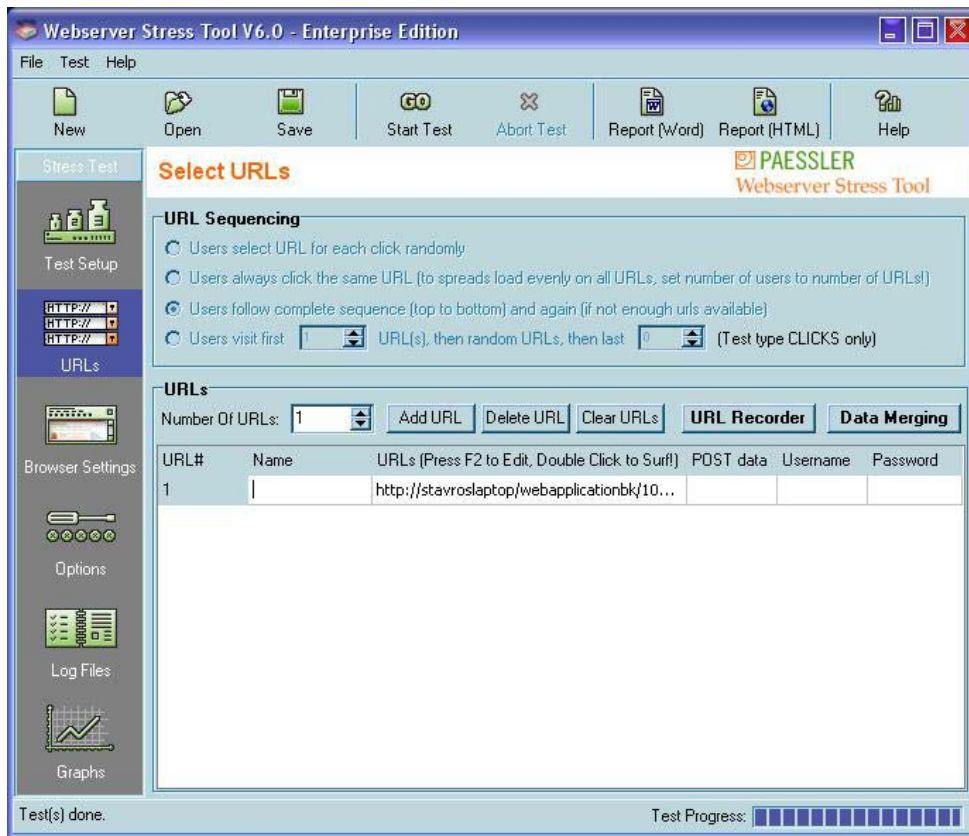
*Processor\%Privileged Time*  
*Processor\%Processor Time*  
*System\Processor Queue Length*  
*Web Service\Total Method Requests/sec*

Στο test αυτό, οι ταυτόχρονοι χρήστες (δηλαδή κάνουν κλικ για αίτηση ταυτόχρονα στον server) ξεκινάν από 0 και σταδιακά φθάνουν τους 100 μέχρι το τέλος του test.

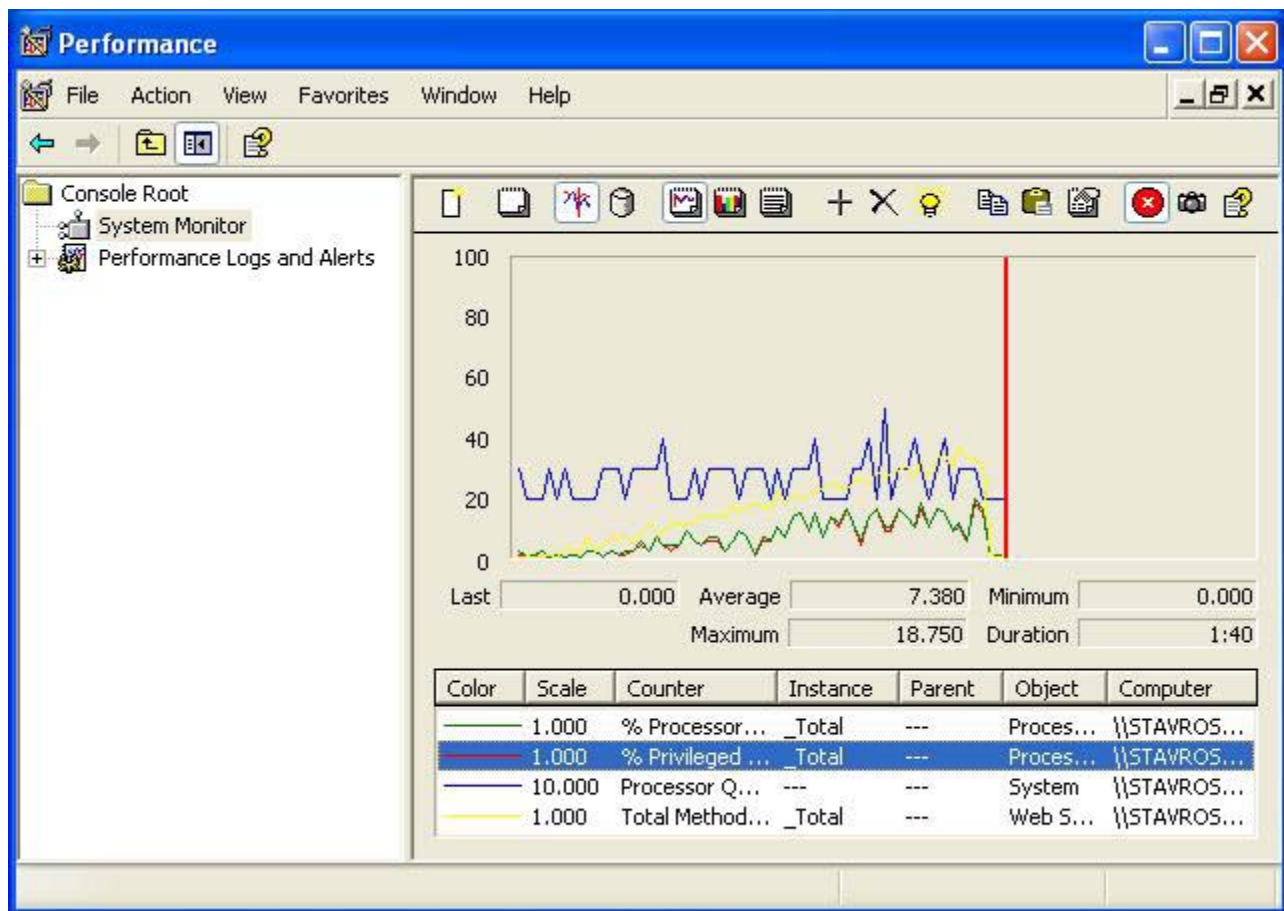
Το URL που χρησιμοποιήσαμε είναι στατικού περιεχομένου μεγέθους 100KB. Όλοι οι χρήστες κάνουν αίτηση για αυτό. Έχουμε χρησιμοποιήσει bandwidth throttling ώστε να προσομοιώσουμε σύνδεση χρηστών μέσω modem 56Kbps. Οι επιτρεπόμενες ταυτόχρονες συνδέσεις είναι 10 (μεταβλητή MaxConnections της metabase), ενώ ο επιτρεπόμενος αριθμός των ταυτόχρονων εκτελούμενων νημάτων είναι 4 (μεταβλητή της registry). Για τις προαναφερθείσες μεταβλητές θα μιλήσουμε αργότερα.

Παραθέτουμε τα παράθυρα με τις ρυθμίσεις του συγκεκριμένου test:





Ακολουθούν τα αποτελέσματα του test, όπως τα έλαβε το System Monitor του server:



Καταρχήν βλέπουμε ότι η ουρά δεν παραμένει σε υψηλά επίπεδα για μεγάλο διάστημα και το Processor Time βρίσκεται σε χαμηλά επίπεδα, επομένως μπορούμε να πούμε ως ένα σημείο ότι ο server μας είναι υγιής.

Βλέπουμε ότι κατά την πάροδο του χρόνου, ο αριθμός των HTTP αιτήσεων ανά δευτερόλεπτο αυξάνεται. Αυτό οφείλεται στο γεγονός ότι ο αριθμός των ταυτόχρονων χρηστών αυξάνεται (Ramp test).

Όσο αυξάνονται όμως οι αιτήσεις, τόσο αυξάνεται και το Processor Time, καθώς επίσης και το Privileged Time. Έτσι επιβεβαιώνονται αυτά που αναφέραμε παραπάνω.

### **Συμπέρασμα:**

Όσο αυξάνεται το φορτίο του server (δηλαδή οι αιτήσεις προς αυτόν), η χρησιμοποίηση του επεξεργαστή αυξάνεται.

## Συμβουλή:

Όταν παρατηρείτε μια μεγάλη ουρά, ενώ όλοι οι επεξεργαστές είναι απασχολημένοι, δημιουργήστε ένα ιστόγραμμα του μετρητή Process\%Processor Time για κάθε διεργασία, με τον τρόπο που έχουμε περιγράψει στο 4<sup>ο</sup> κεφάλαιο. Το ιστόγραμμα δείχνει το χρόνο επεξεργαστή που καταναλώνει η κάθε διεργασία.

1. Αν ένα bar του ιστογράμματος είναι υψηλότερο και ξεχωρίζει από όλα τα άλλα, τότε ίσως η διεργασία που αναπαριστά να χρησιμοποιεί παραπάνω χρόνο επεξεργαστή από όσο θα έπρεπε, προκαλώντας έτσι το bottleneck επεξεργαστή. Σ' αυτήν την περίπτωση αντικαταστήστε την εφαρμογή που τρέχει σ' αυτήν τη διεργασία ή μεταφέρετε τη διεργασία σε άλλον server.
2. Αν πάλι όλες οι διεργασίες διαμοιράζονται εξίσου τον επεξεργαστή, τότε θα ήταν καλή ιδέα να αναβαθμίσετε τον επεξεργαστή σας ή να προσθέσετε έναν νέο.

Προκειμένου να μελετήσουμε τη δραστηριότητα του επεξεργαστή σε σχέση με τα **interrupts** και τα **DPCs**, καλό είναι να ορίσουμε λίγο αυτά τα δύο χαρακτηριστικά. Ένα interrupt είναι μια ρουτίνα που καλείται από το λειτουργικό σύστημα, όταν απαιτείται η αναστολή της λειτουργίας του επεξεργαστή, προκειμένου αυτός να αναλάβει μια συγκεκριμένη διεργασία, όπως μια λειτουργία I/O (π.χ. διάβασμα από το δίσκο). Τα DPCs είναι παρόμοια με τα interrupts με τη μόνη διαφορά ότι αυτά μπορούν να ανασταλούν για λίγο, μέχρις ότου ο επεξεργαστής ολοκληρώσει διεργασίες υψηλότερης προτεραιότητας. Εμείς θα αντιμετωπίζουμε τα DPCs ως τις ρουτίνες που καλούνται από μια κάρτα δικτύου για την αντιγραφή ενός ληφθέντος πακέτου στην κύρια μνήμη. Τα Interrupt Time και DPC Time είναι μέρη του Privileged Time.

Έτσι λοιπόν, τόσο η κάρτες δικτύου όσο και οι σκληροί δίσκοι, παράγουν interrupts ως αποτέλεσμα των εισερχόμενων αιτήσεων του IIS. Βλέποντας τη χρησιμοποίηση της CPU σε σχέση με τα interrupts και το ρυθμό με τον οποίο παράγονται τα interrupts αυτά, μπορείτε να διαπιστώσετε αν ο server σας μπορεί να χειριστεί την ένταση του φορτίου που του έχει ανατεθεί τη συγκεκριμένη χρονική στιγμή. Επίσης, αν γνωρίζουμε ότι ένα μεγάλο κομμάτι της CPU καταναλώνεται από τα interrupts, τότε υπάρχουν μερικά πράγματα που μπορούμε να κάνουμε ώστε να ανακουφίσουμε κάποιο φορτίο της CPU.

---

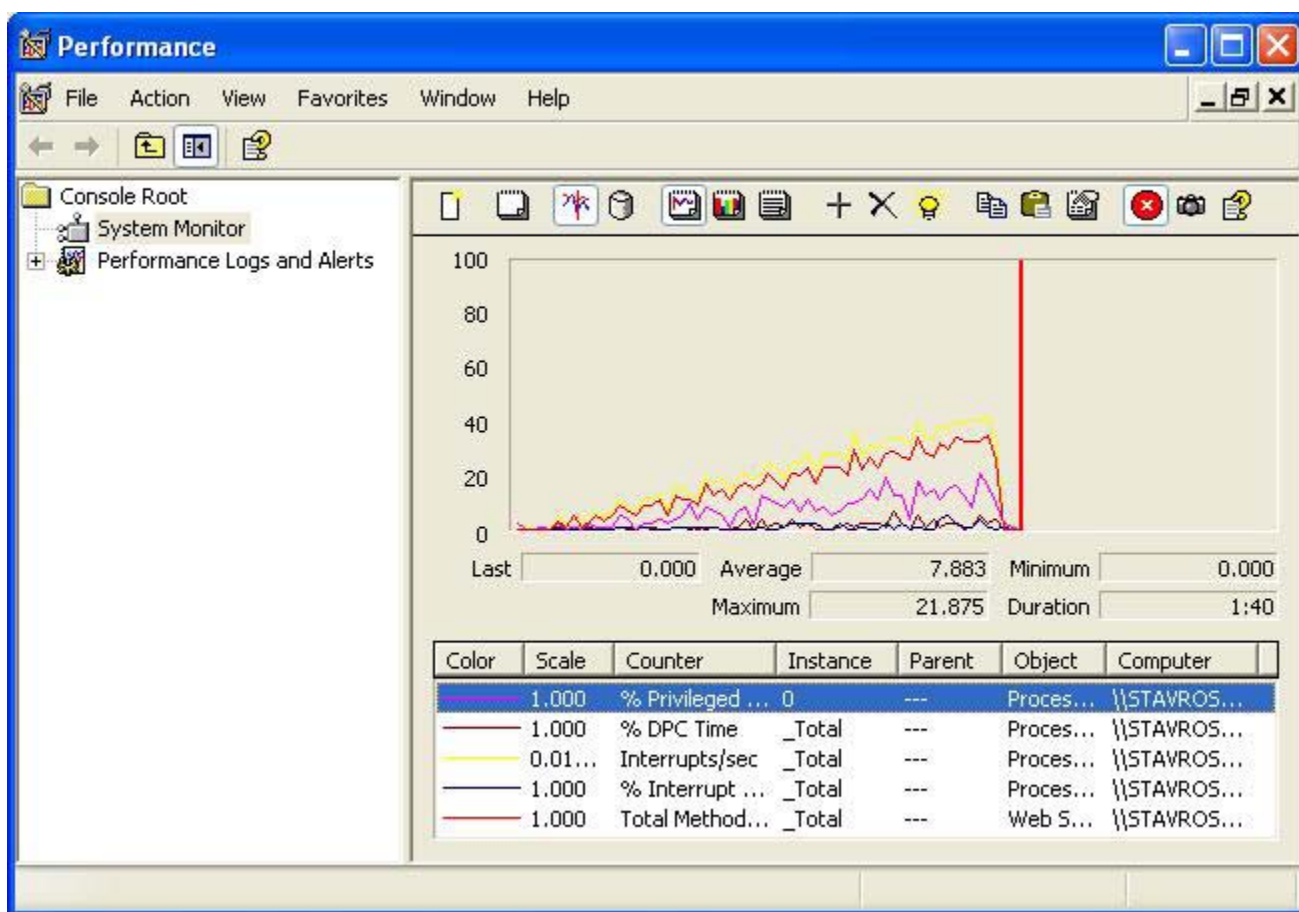
## Test #2

Για να μην επαναλαμβάνουμε τα παράθυρα, το test είναι ακριβώς το ίδιο με το test #1

Οι μετρητές που θα παρακολουθήσουμε σ' αυτό το test είναι οι εξής:

*Processor\%Privileged Time*  
*Processor\%DPC Time*  
*Processor\%Interrupt Time*  
*Processor\Interrupts/sec*  
*Web Service\Total Method Requests/sec*

Τα αποτελέσματα φαίνονται στο παρακάτω σχήμα:



Παρατηρούμε πάλι πως καθώς αυξάνονται οι αιτήσεις, αυξάνεται τόσο το Privileged Time όσο και τα Interrupt Time και DPC Time, με το DPC Time να είναι μεγαλύτερος από το Interrupt Time.

### **Συμπέρασμα:**

Καθώς αυξάνονται οι αιτήσεις προς τον IIS, αυξάνεται και ο χρόνος CPU που αφιερώνεται στα interrupts και τα DPCs. Το DPC Time είναι μεγαλύτερο από το Interrupt Time, κάτι που δικαιολογείται αφού, μιας και ανακτηθεί ένα αρχείο από το δίσκο, παρακρατείται σε έναν buffer, οπότε οι προσβάσεις στο δίσκο περιορίζονται, ενώ τα DPCs αυξάνονται ούτως ή άλλως με την αύξηση των πακέτων που καταφθάνουν στην κάρτα δικτύου.

### **Συμβουλή:**

Αν έχετε προσδιορίσει ότι η χρησιμοποίηση της CPU είναι πολύ υψηλή λόγω των interrupts και DPCs, τότε μπορείτε:

1. Να αγοράσετε μια κάρτα δικτύου που υποστηρίζει *interrupt moderation*, ένα χαρακτηριστικό που υπάρχει στους οδηγούς συσκευής των περισσότερων σύγχρονων καρτών δικτύου. Αντί να παράγεται ένα interrupt για κάθε πακέτο που φθάνει στην κάρτα δικτύου, τα interrupts αποθηκεύονται προσωρινά σε έναν buffer και παρουσιάζονται στο server σε ένα batch (ομάδα). Ενεργοποιώντας αυτό το χαρακτηριστικό, η χρησιμοποίηση της CPU μειώνεται σημαντικά.
2. Να αγοράσετε μια πρόσθετη κάρτα δικτύου. Το ιδανικό θα ήταν να διαθέτατε μια κάρτα δικτύου για κάθε επεξεραστή. Προσθέτοντας πολλαπλές κάρτες δικτύου στο server, μπορείτε να διανέμετε τα interrupts στους επεξεργαστές, μειώνοντας έτσι το επιπλέον φορτίο της επεξεργασίας των interrupts αυτών από τη CPU.

---

## **Συνδέσεις (Connections)**

Είναι σημαντικό να προσδιορίσετε πώς αντιδράει ο server σας όταν χειρίζεται διαφορετικό αριθμό συνδέσεων. Όταν έχετε συλλέξει δεδομένα σχετικά με τις τάσεις των συνδέσεων, μπορείτε να συσχετίσετε τα δεδομένα σχετικά με τη γενική απόδοση του server με τον αριθμό των συνδέσεων που εξυπηρετώνται ταυτόχρονα.

**Σημείωση: Τα Windows XP υποστηρίζουν μέχρι 10 συνδέσεις ταυτόχρονα στο server. Αυτό μας περιορίζει κάπως όσον αφορά στα πειράματα που θα θέλαμε να διενεργήσουμε.**

Κάθε σύνδεση που εγκαθιστά ο IIS 5.0 καταναλώνει κάποιο χρόνο επεξεργαστή. Ο προσαρμογέας της κάρτας δικτύου διακόπτει τον επεξεργαστή, σηματοδοτώντας ότι ένας client έχει κάνει μια αίτηση. Περαιτέρω επεξεργασία απαιτείται για την εγκατάσταση και τη διατήρηση της σύνδεσης, για τη διεκπεραίωση των αιτήσεων των clients που αποστέλλονται μέσω αυτής της σύνδεσης και, όταν η σύνδεση κλείνει, απαιτείται επεξεργασία για τη διαγραφή των δομών που εξυπηρέτησαν τη σύνδεση. Έτσι, κάθε φορά που εγκαθίσταται μια σύνδεση, το φορτίο του server αυξάνεται. (αυτό είναι κάτι πολύ εύκολο να διαπιστωθεί, οπότε δεν θα διενεργήσουμε κάποιο πείραμα για να το δείξουμε).

Ο IIS 5.0 πριλαμβάνει αρκετά χαρακτηριστικά για τη βελτιστοποίηση του χειρισμού των συνδέσεων. Ανάμεσα σ'αυτά τα χαρακτηριστικά είναι τα **HTTP Keep-Alives**, τα οποία είναι διαφορετικά από τα *TCP/IP Keep-Alives*. Ενώ τα τελευταία είναι μηνύματα



που αποστέλλονται προκειμένου να προσδιοριστεί αν μια σύνδεση είναι ακόμη ενεργή, τα πρώτα διατηρούν τη σύνδεση ακόμη και αφού η πρώτη αίτηση της σύνδεσης ολοκληρωθεί. Τα HTTP Keep-Alive κρατούν τη σύνδεση ενεργή και διαθέσιμη για επόμενες αιτήσεις, έτσι ώστε να αποφευχθεί το πρόσθετο κόστος της εγκατάστασης και του τερματισμού των συνδέσεων. Τα HTTP Keep-Alive είναι ενεργοποιημένα στον IIS 5.0 από default. Παρόλο που βελτιώνουν σημαντικά την απόδοση του εύρους ζώνης στους περισσότερους servers και βελτιώνουν τους χρόνους απόκρισης για τους clients, μπορείτε να τις τροποποιήσετε ή να τις απενεργοποιήσετε αν νομίζετε ότι δεν χρειάζονται. Καλό είναι πάντως να τις έχετε πάντα ενεργοποιημένες.

Τα HTTP Keep-Alive του IIS θα τα βρείτε ως εξής: Ακολουθήστε το path *Start → Control Panel → Administrative Tools → Internet Information Services*. Κάντε δεξί κλικ στο 'Default Web Site' και επιλέξτε Properties. Τα HTTP Keep-Alive θα τα βρείτε στο Web Site tab.

**Σημείωση: Το Webserver Stress Tool, για κάθε αίτηση ενός από τους χρήστες εγκαθιστά και μία νέα σύνδεση. Επομένως τα HTTP Keep-Alive είναι αχρείαστα για τα δικά μας πειράματα. Έτσι, δυστυχώς, δεν μπορούμε να δούμε τη διαφορά αν τα απενεργοποιήσουμε!**

Οι μετρητές που μπορούμε να χρησιμοποιήσουμε για να παρακολουθήσουμε τις συνδέσεις παρατίθενται παρακάτω:

<b>Μετρητής</b>	<b>Περιγραφή</b>
Web Service\Current Connections	Ο αριθμός των συνδέσεων που διατηρήθηκαν ταυτόχρονα από τον IIS κατά τη διάρκεια του πιο πρόσφατου δειγματικού διαστήματος.
Web Service\Maximum Connections	Ο μεγαλύτερος αριθμός των συνδέσεων που διατηρήθηκαν ταυτόχρονα από τη στιγμή που εκκινήθηκε ο server για τελευταία φορά.

Μπορείτε να αναγνωρίσετε ένα bottleneck επεξεργαστή κατά τη διάρκεια ενός δοθέντος διαστήματος ως εξής:

- Παρατηρείτε παραπάνω από δυο threads στην ουρά για παρατεταμένο χρόνο.
- Παρατηρείτε υψηλή χρησιμοποίηση σε έναν ή περισσότερους επεξεργαστές

- Παρατηρείτε μια καμπύλη στο γράφημα του μετρητή Current Connections που φθάνει μέχρι μια υψηλή τιμή και μετά διαμορφώνει υψίπεδο. Αυτό σημαίνει ότι πρόσθετες συνδέσεις μπλοκάρονται ή απορρίπτονται.

### Συμβουλή:

Προκειμένου να αποτρέψετε bottlenecks επεξεργαστή, διασφαλίστε ότι δεν δημιουργείται μια μεγάλη ουρά επεξεργαστή, όταν εξυπηρετείτε μεγάλους αριθμούς συνδέσεων. Αν ο επεξεργαστής συνηθίζει να υπομένει bottlenecks όταν εξυπηρετεί μεγάλο αριθμό συνδέσεων, τότε θα ήταν καλό να αναβαθμίσετε ή να προσθέσετε επεξεργαστές, ή να περιορίσετε το μέγιστο αριθμό των συνδέσεων στο server. Περιορίζοντας τις συνδέσεις, ο server μπορεί να μπλοκάρει ή να απορρίπτει συνδέσεις, αλλά αυτό βοηθά στο να διασφαλίζεται ότι οι αποδεχόμενες συνδέσεις υφίστανται επεξεργασία γρήγορα. Όμως, θα πρέπει να ληφθεί σοβαρά υπ'όψη το γεγονός ότι περιορίζοντας τις συνδέσεις, είναι δυνατόν να αυξηθεί η χρησιμοποίηση της CPU.

Κάποιος θα μπορούσε να σκεφτεί λανθασμένα ότι, αν περιορίσουμε το μέγιστο αριθμό των ταυτόχρονων συνδέσεων που μπορεί να εξυπηρετεί ο server, τότε θα μειώσουμε το CPU Time. Κάτι τέτοιο δεν είναι αλήθεια. Όταν μια αίτηση σύνδεσης καταφθάνει μέσω του δικτύου στο server, την παίρνει η CPU και τη μεταφέρει μέσω των επιπέδων δικτύου, μέχρις ότου αυτή φτάσει στο επίπεδο εφαρμογών, όπου και την αναλαμβάνει ο IIS. Σ' αυτό το σημείο, βασισμένος στο μέγιστο αριθμό των ταυτόχρονων αιτήσεων, ο IIS αποφασίζει αν θα δεχτεί την αίτηση ή όχι. Καθώς ο server καθυστερεί την έναρξη της επεξεργασίας των εισερχόμενων αιτήσεων σύνδεσης, αυτές αρχίζουν και συσσωρεύονται. Αυτό γίνεται ξεκάθαρο, αν παρακολουθήσετε το μετρητή Connection Attempts/sec, ο οποίος αρχίζει και αυξάνεται. Έτσι, περιορίζοντας το μέγιστο αριθμό των ταυτόχρονων συνδέσεων, ο IIS έχει περισσότερη δουλειά να κάνει, επιφορτίζοντας τη CPU έμμεσα με πρόσθετο φόρτο για την αποδοχή ή απόρριψη των αιτήσεων, με αποτέλεσμα το CPU Time να αυξάνεται.

Το καλύτερο θα ήταν να πειραματιστείτε αρκετά, ώστε να βρείτε τον ιδανικό αριθμό των μέγιστων συνδέσεων που θα πρέπει να επιτρέπει ο server σας, προκειμένου να πετύχετε μικρότερη χρησιμοποίηση επεξεργαστή και μέγιστο throughput.

Η μεταβλητή της metabase του IIS 5.0 που είναι υπεύθυνη για το μέγιστο αριθμό των συνδέσεων είναι η **MaxConnections**. Μπορούμε να τροποποιήσουμε τη μεταβλητή αυτήν από το εργαλείο Metabase Explorer που βρίσκεται μέσα στο IIS 6.0 Resource Kit. Ανοίγουμε το εργαλείο αυτό και κάνουμε κλικ στο <Computer Name> → Schema → Properties → Defaults για να εμφανιστούν στα δεξιά τα entries της metabase. Κάνοντας δεξί κλικ στο MaxConnections μπορούμε να θέσουμε μια τιμή της αρεσκείας μας.

The screenshot shows the IIS Metabase Explorer window. The left pane displays a tree view with the following structure:

- STAVROSLAPTOP ( local )
  - LM
  - Schema
    - Classes
    - Properties
      - Defaults
      - Names
      - Types

The right pane displays a table of properties:

ID	Name	Data Type	Data
1000	MaxBandwidth	DWORD	4294967295
1002	KeyType	String	
1003	MaxBandwidthBlocked	DWORD	4294967295
1013	ConnectionTimeout	DWORD	900
1014	MaxConnections	DWORD	10
1015	ServerComment	String	
1016	ServerState	DWORD	4
1017	ServerAutoStart	DWORD	4294967295
1018	ServerSize	DWORD	1
1019	ServerListenBacklog	DWORD	40
1020	ServerListenTimeout	DWORD	120
1022	DirectoryLevelsToScan	DWORD	2
1023	ServerBindings	MultiString	
1024	MaxEndpointConnections	DWORD	4294967295
1025	ClusterEnabled	DWORD	0
1027	ServerConfigFlags	DWORD	0
1029	DisableSocketPooling	DWORD	0
1099	Win32Error	DWORD	0
2021	SecureBindings	MultiString	
2040	FilterLoadOrder	String	

The status bar at the bottom indicates the path `/Schema/Properties/Defaults/`, the number of items `486 items`, and the server name `STAVROSLAPTOP ( local )`.

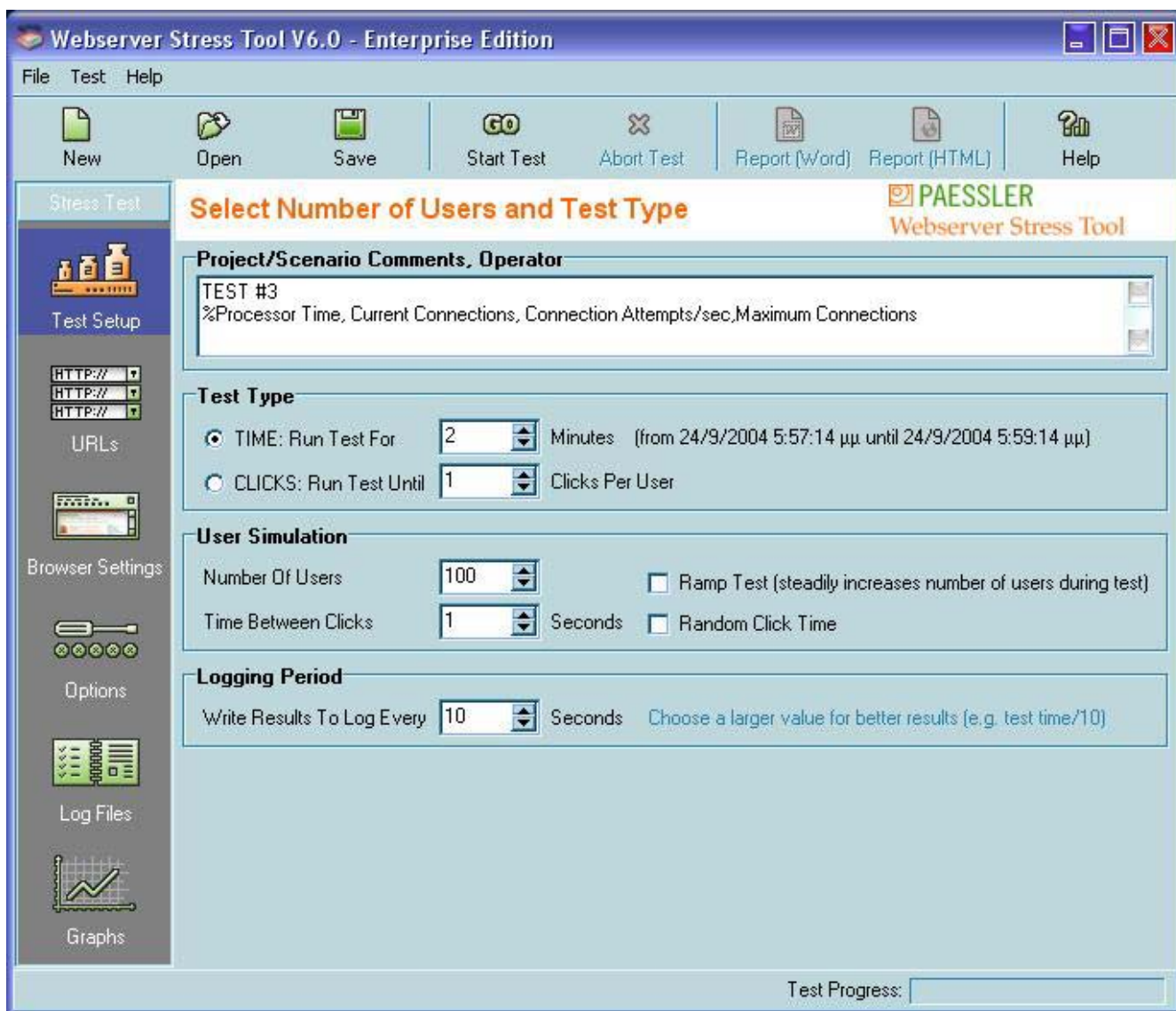
The screenshot shows the **MaxConnections Properties** dialog box. It has a title bar with a close button. The dialog contains two tabs: **Value** and **General**. The **Value** tab is selected, and it displays a text box with the value `10` next to the label **Value:**. At the bottom of the dialog, there are three buttons: **OK**, **Cancel**, and **Apply**.

## Test #3

**Σημείωση:** Στο τεστ αυτό αναγκαστήκαμε να εγκαταστήσουμε το λειτουργικό Windows 2000 Server ώστε να μπορούμε να επιτρέπουμε στο server να δέχεται απεριόριστο αριθμό συνδέσεων.

Πραγματοποιήσαμε το εξής τεστ:

Θέσαμε στον αριθμό των ταυτόχρονων χρηστών την τιμή 100, επιλέξαμε σαν URL ένα στατικό site μεγέθους 1MB και διενεργήσαμε το πείραμα για 2 λεπτά. Επίσης χρησιμοποιήσαμε bandwidth throttling 56Kbps. Οι 100 χρήστες κάνουν ταυτόχρονα κλικ στο URL ανά ένα δευτερόλεπτο. Επιλέξαμε οι επιτρεπόμενες συνδέσεις να είναι απεριόριστες (default επιλογή του IIS στα Windows 2000 Server).



Επιλέξαμε ένα τόσο μεγάλο site ώστε να μπορέσουμε να επιτύχουμε μεγάλο αριθμό ανοιχτών συνδέσεων, καθώς το Webserver Stress Tool με κάθε κλικ ανοίγει και μια νέα σύνδεση. Κάνοντας ένας χρήστης κλικ σε ένα μεγάλο site, δεν μπορεί να εξυπηρετηθεί αμέσως από το server κι επομένως συσσωρεύονται οι επόμενες αιτήσεις σύνδεσης.

Στο server θα μετρήσουμε τους παρακάτω μετρητές για να δούμε πώς συμπεριφέρεται η CPU ανάλογα με τον αριθμό των ταυτόχρονων συνδέσεων που επιτρέπει ο server:

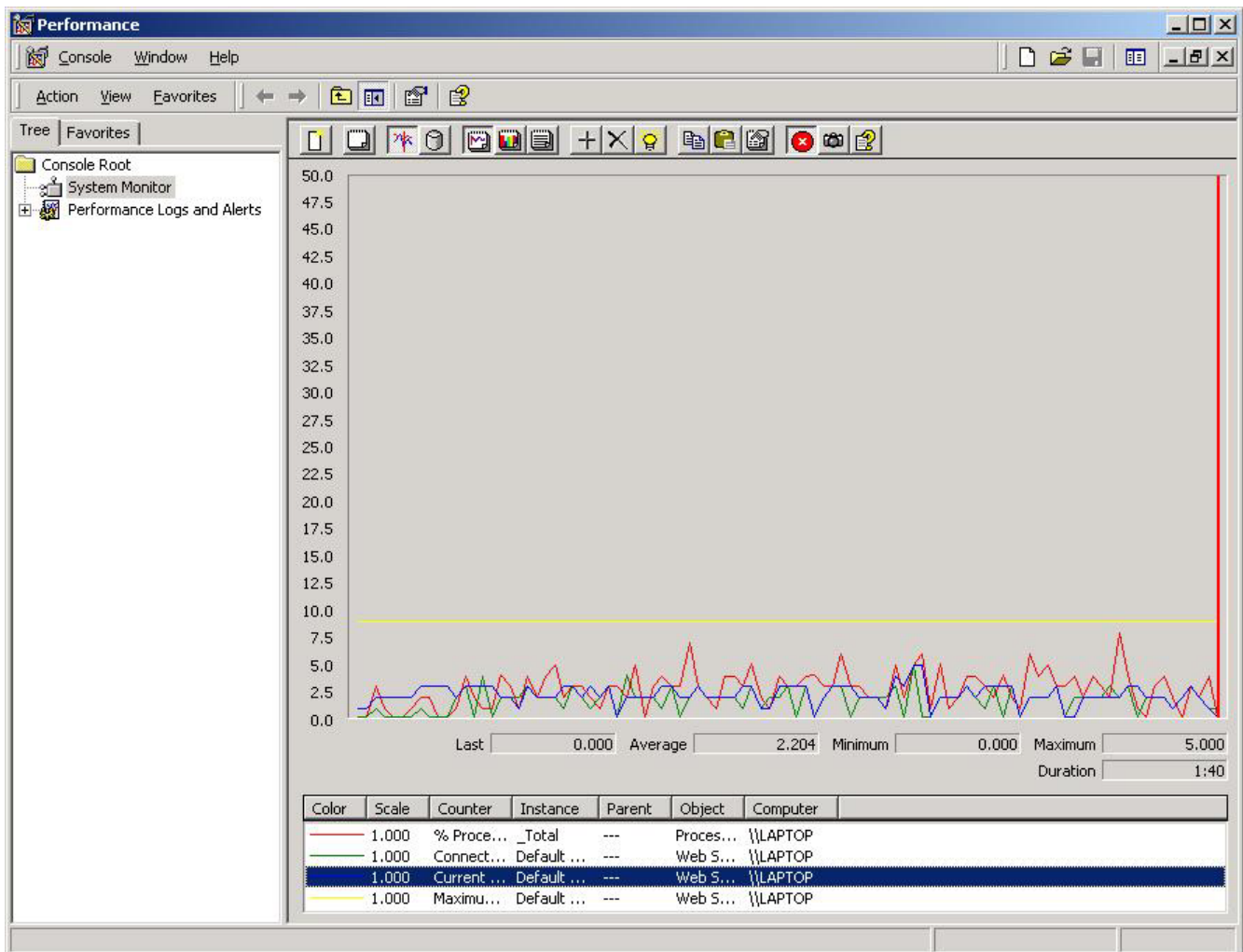
*Processor\%Processor Time*

*Web Service\Connection Attempts/sec*

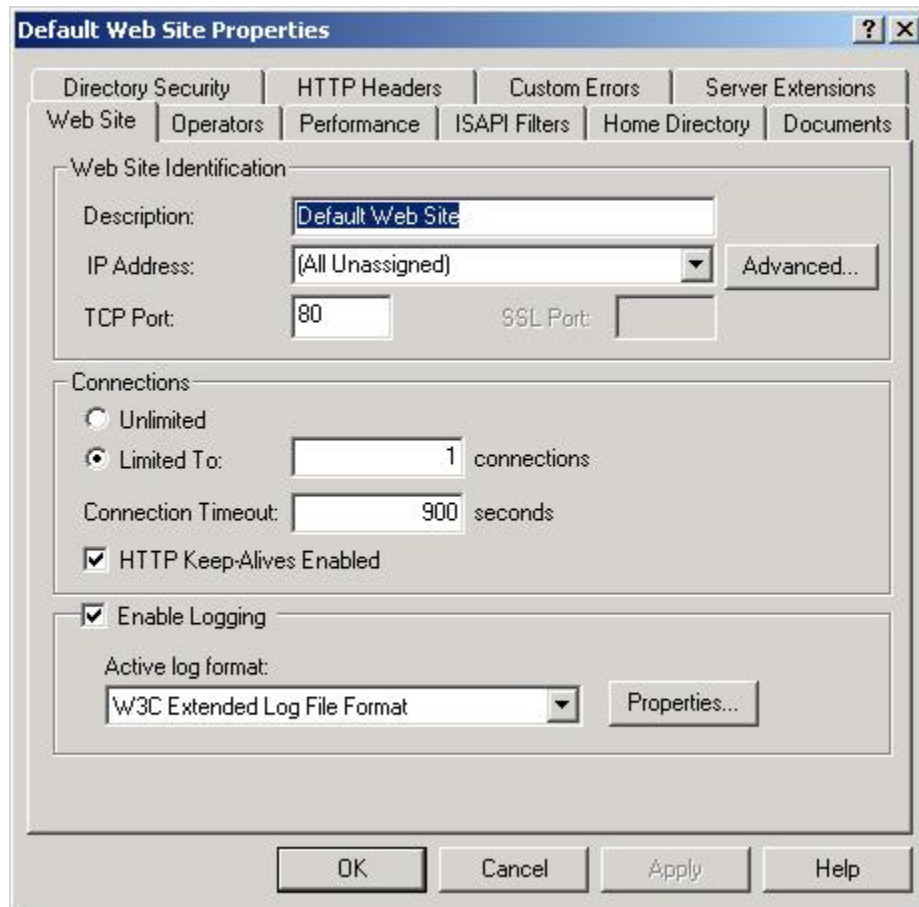
*Web Service\Current Connections*

*Web Service\Maximum Connections*

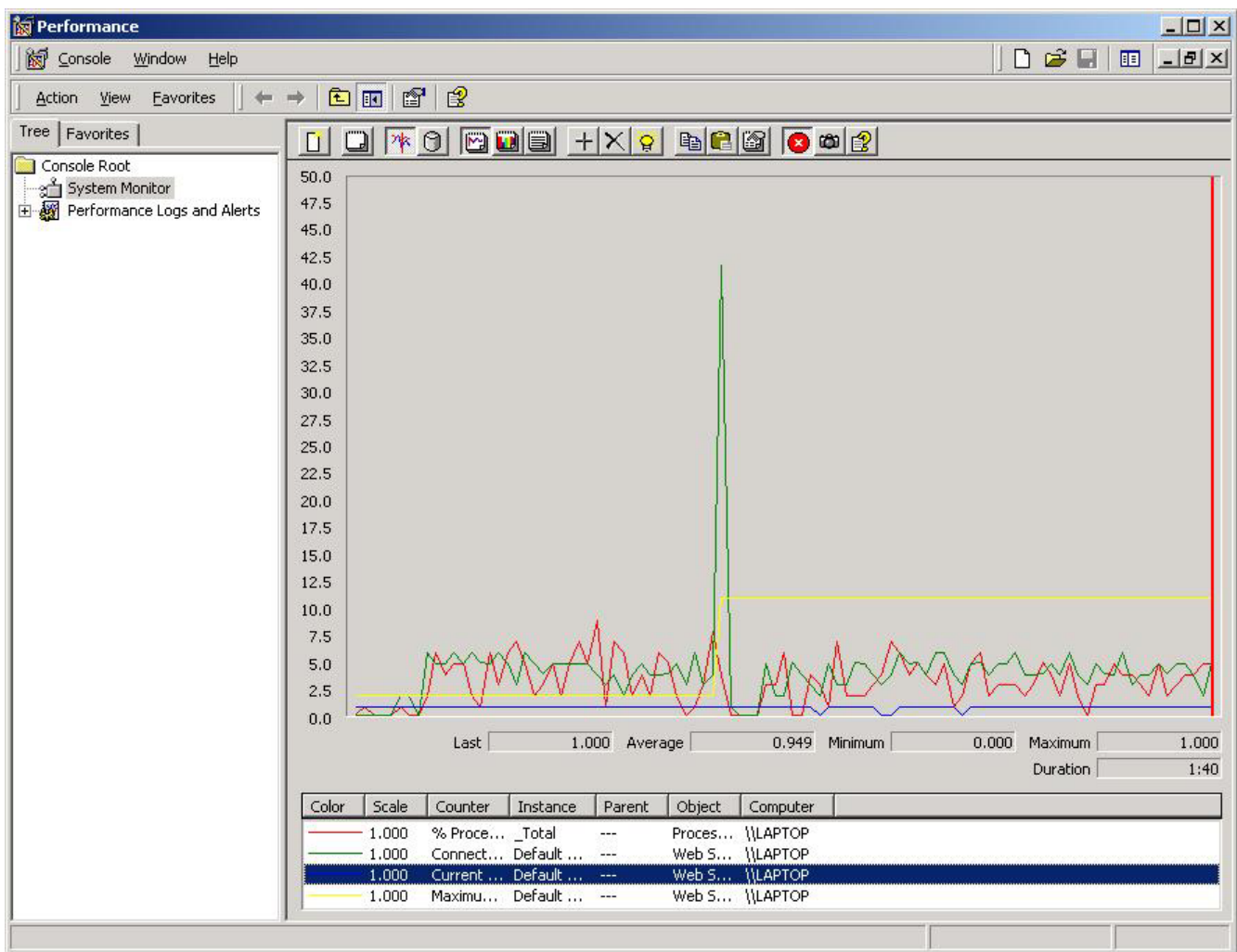
Μετά τη διενέργεια του τεστ είχαμε την ακόλουθη έξοδο στο System Monitor του server:



Τώρα θέτουμε ως μέγιστο αριθμό συνδέσεων τη 1 σύνδεση. Αυτό το κάνουμε ανοίγοντας το *Start* → *Programs* → *Administrative Tools* → *Internet Services Manager*. Εκεί κάνουμε δεξί κλικ στο Default Web Site κι επιλέγουμε Properties. Στο web Site tab του παραθύρου διαλόγου επιλέγουμε Limited To 1 connections. (Θα μπορούσαμε εναλλακτικά να αλλάξουμε τη μεταβλητή MaxConnections της metabase με τον τρόπο που δείξαμε παραπάνω). Τέλος κάνουμε Restart τον IIS.



Διενεργούμε ξανά το ίδιο πείραμα και παράγεται η ακόλουθη έξοδος στο System Monitor του server:



Θα περίμενε κανείς ότι στο δεύτερο πείραμα, επειδή ο server έχει να επεξεργαστεί μόνο μια αίτηση, το Processor Time θα ήταν μικρότερο. Παρόλ'αυτά, όπως είπαμε, αυξάνονται τα Connection Attempts/sec και ο IIS χρησιμοποιεί έμμεσα τη CPU για να δεχτεί ή να απορρίψει μια αίτηση, οπότε το Processor Time αυξάνεται στο δεύτερο πείραμα.

Οι Maximum Connections ουσιαστικά μπορεί να μην έχουν τις τιμές που φαίνονται στα γραφήματα, επειδή ίσως να μην πρόλαβε ο server να διαγράψει τις δομές τους την ώρα της δειγματοληψίας. Οι Maximum Connections κανονικά πρέπει να είναι μικρότερες. Ο μετρητής Current Connections είναι πιο κατατοπιστικός για το τι γίνεται με τις συνδέσεις.

### **Συμπέρασμα:**

Όταν μειώνεται ο αριθμός των μέγιστων επιτρεπόμενων ταυτόχρονων συνδέσεων, το CPU utilization αυξάνεται!

---

## 📖 Νήματα

Παρόλο που δεν μπορούμε να αντιστοιχίσουμε ατομικά νήματα με συνδέσεις ή ατήσεις, μπορούμε:

- Να μετρήσουμε τον αριθμό των νημάτων που επεξεργάζεται ο IIS
- Να μετρήσουμε το ποσοστό του χρόνου επεξεργαστή που χρησιμοποιεί κάθε νήμα
- Να αντιστοιχίσουμε τον αριθμό των νημάτων (και τη δραστηριότητα του επεξεργαστή) με τον αριθμό των συνδέσεων, τον αριθμό των αρχείων που εξυπηρετώνται, καθώς και με άλλα μεγέθη της δραστηριότητας και της απόδοσης του server.

Παρακάτω παραθέτουμε έναν πίνακα με τους μετρητές που μπορείτε να χρησιμοποιήσετε προκειμένου να παρακολουθήσετε τη συμπεριφορά των νημάτων:

---

Μετρητής	Περιγραφή
<i>Process\Thread Count: Inetinfo</i>	Ο αριθμός των νημάτων που δημιουργούνται από τη διεργασία. Αυτός ο μετρητής δεν δείχνει ποια νήματα είναι απασχολημένα και ποια είναι αδρανή. Δείχνει την τελευταία τιμή που παρατηρήθηκε, όχι το μέσο όρο.
<i>Thread\%Processor Time: Inetinfo =&gt; Thread #</i>	Δείχνει πόσο χρόνο επεξεργαστή χρησιμοποιεί κάθε νήμα της διεργασίας Inetinfo.
<i>Thread\Context Switch/sec: Inetinfo =&gt; Thread #</i>	Δείχνει πόσες φορές τα νήματα της υπηρεσίας IIS εναλλάσσονται στον επεξεργαστή. Αυτός ο μετρητής δείχνει τη δραστηριότητα των νημάτων της υπηρεσίας IIS.

Μπορείτε να φτιάξετε ένα γράφημα για τον μετρητή *Process\Thread Count: Inetinfo* στο χρόνο για να δείτε πόσα νήματα δημιουργεί η Inetinfo και πώς ποικίλουν οι αριθμοί των νημάτων. Έπειτα, παρατηρήστε το χρόνο επεξεργαστή για κάθε νήμα της διεργασίας κατά τη διάρκεια περιόδων υψηλής, μέσης και χαμηλής δραστηριότητας του server.

Πρέπει επίσης να παρατηρείτε και τα context switces στο χρόνο, τα οποία δείχνουν ότι ο πυρήνας του λειτουργικού έχει αναγκάσει τον επεξεργαστή να αναστείλλει την επεξεργασία ενός νήματος για να αναλάβει κάποιο άλλο. Ένας μεγάλος αριθμός νημάτων είναι πολύ πιθανό να αυξήσει τον αριθμό των context switches. Ενώ επιτρέπουν σε πολλαπλά νήματα να μοιράζονται τον επεξεργαστή, τα context switces διακόπτουν επίσης τον επεξεργαστή και μπορεί να επηρεάσουν την απόδοσή του,



κυρίως σε multiprocessor υπολογιστές. Για όσο η χρησιμοποίηση του επεξεργαστή είναι κάτω από 70%, δεν τίθεται θέμα προβλήματος.

Ως default, ο IIS δημιουργεί μέχρι 4 νήματα ανά επεξεργαστή (για εξυπηρέτηση στατικού περιεχομένου). Ο IIS συνεχίζει να προσαρμόζει δυναμικά τα νήματα στη διεργασία του σε απόκριση προς τη δραστηριότητα του server. Για τα περισσότερα συστήματα, αυτή η αυτόματη ρύθμιση του IIS είναι επαρκής για να διατηρηθεί ο βέλτιστος αριθμός των νημάτων, αλλά μπορείτε να αλλάξετε το μέγιστο αριθμό των νημάτων ανά επεξεργαστή, αν αυτό κρίνεται απαραίτητο.

### Συμβουλή:

Σχετικά με τον αριθμό των νημάτων:

1. Αν σχεδόν όλα τα νήματα της διεργασίας του IIS είναι απασχολημένα σχεδόν όλη την ώρα και όλοι οι επεξεργαστές είναι πολύ κοντά στη μέγιστη χωρητικότητά τους, τότε θα ήταν καλό να διανέματε το φορτίο σε περισσότερους servers. Μπορείτε να προσθέσετε και άλλους επεξεργαστές, αλλά προσεκτικά. Αχρείαστοι ή υποχρησιμοποιούμενοι επεξεργαστές θα μειώσουν την απόδοση, δεν θα τη βελτιώσουν.
2. Αν σχεδόν όλα τα νήματα φαίνονται απασχολημένα, ενώ οι επεξεργαστές δεν είναι πάντα ενεργοί, τότε θα ήταν καλό να αυξάνατε το μέγιστο αριθμό των νημάτων ανά επεξεργαστή. Μην αυξήσετε το μέγιστο αριθμό των νημάτων, αν οι επεργαστές σας δεν έχουν πολλή μεγάλη χωρητικότητα. Περισσότερα νήματα σε έναν επεξεργαστή προκαλούν περισσότερες διακοπές και context switches, με αποτέλεσμα να αναλογεί λιγότερος χρόνος επεξεργαστή ανά νήμα και άρα μείωση της απόδοσης.

Για να αυξήσετε το μέγιστο αριθμό των νημάτων ανά επεξεργαστή, πρέπει να ρυθμίσετε τη μεταβλητή **MaxPoolThreads** της registry των Windows. Για να το κάνετε αυτό ακολουθήστε τα παρακάτω βήματα:

1. Ανοίξτε τον Registry Editor από Start→Run... κι εκεί πληκτρολογήστε regedt32 (για τα Windows XP).
2. Ανοίξτε τον κατάλογο Parameters στο μονοπάτι *HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Inetinfo\Parameters*
3. Στο δεξί παράθυρο φαίνονται τα περιεχόμενα του φακέλου αυτού. Κάντε δεξί κλικ και επιλέξτε New→DWORD Value και εισάγετε το όνομα MaxPoolThreads.
4. Κάνοντας διπλό κλικ μπορείτε να θέσετε πλέον την τιμή για τη μεταβλητή αυτή.
5. Πρέπει να επανεκκινήσετε την υπηρεσία του IIS. Για να το κάνετε αυτό ανοίξτε το Internet Information Services. Κάντε δεξί κλικ στο όνομα του υπολογιστή και

επιλέξτε All Tasks→Restart IIS...Τέλος πατήστε OK και μετά End Now και περιμένετε να επανεκκιήσει η υπηρεσία.

### Συμβουλή:

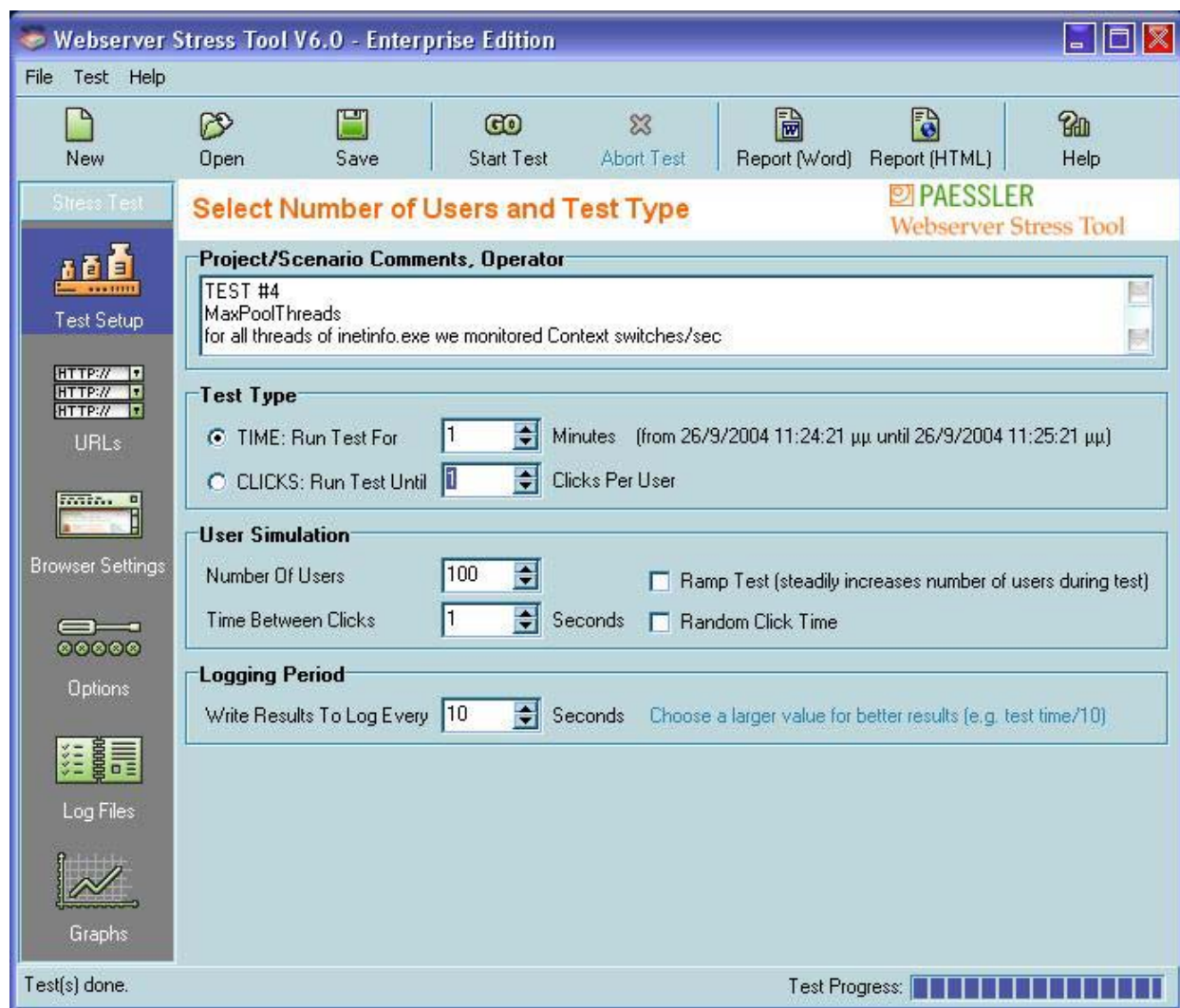
Μην θέτετε ποτέ την τιμή της μεταβλητής MaxPoolThreads κάτω από 2 ή πάνω από 20.

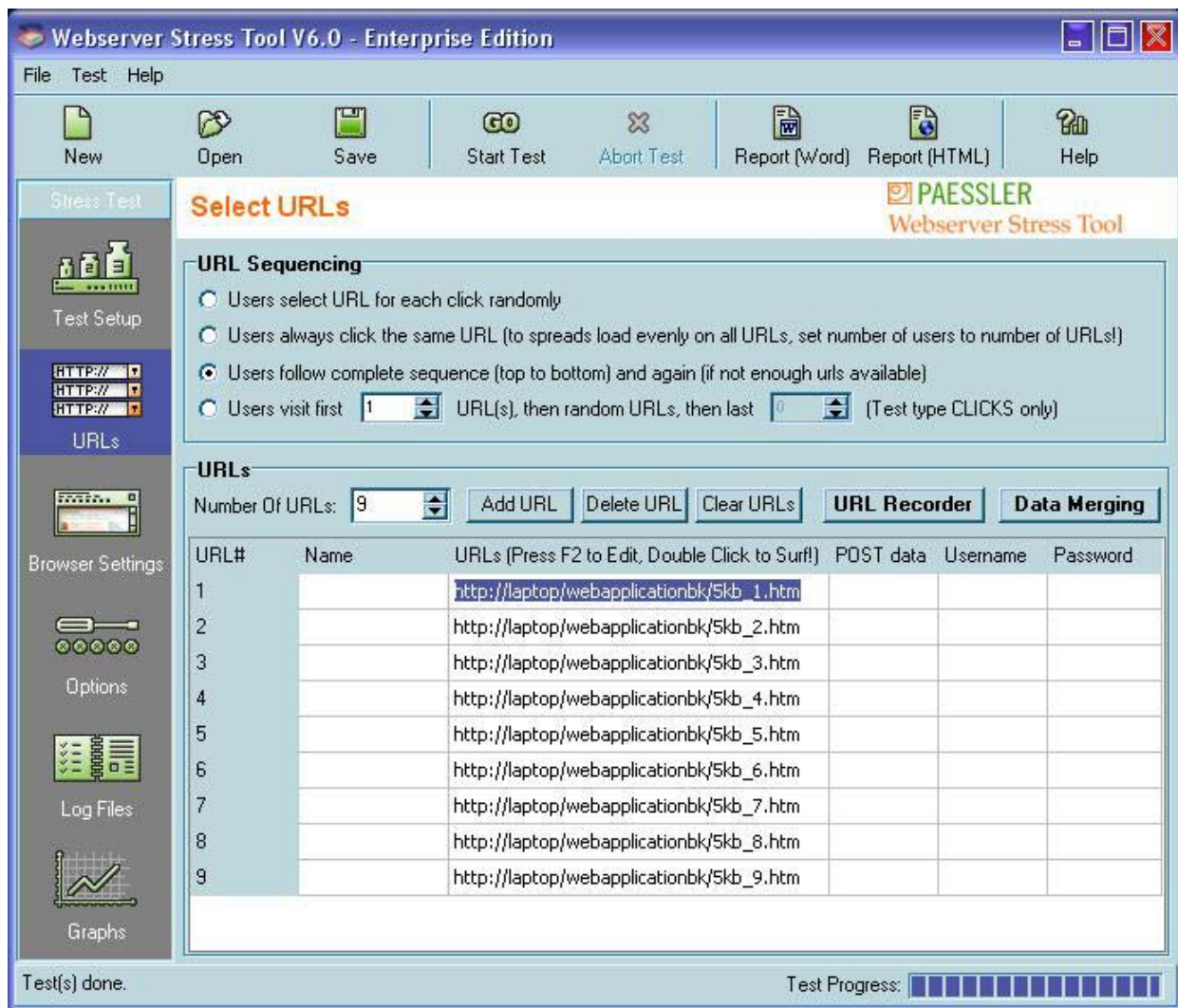
---

## Test #4

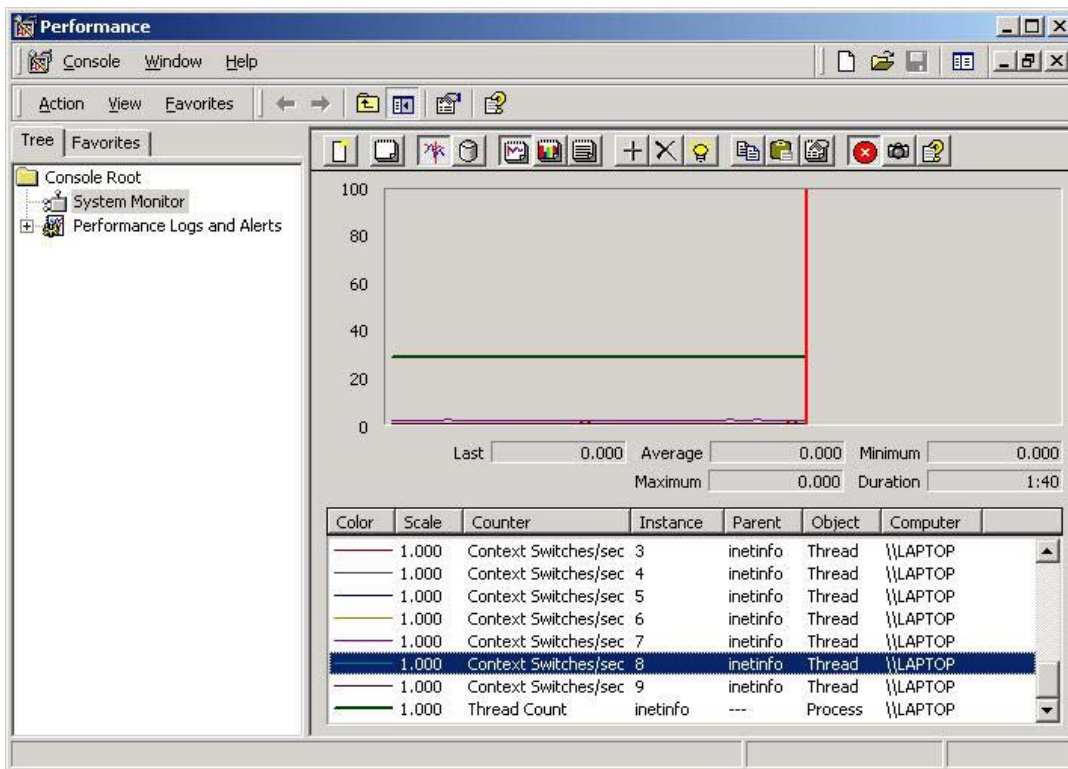
Κάναμε το παρακάτω πείραμα για να δείξουμε ότι όταν αυξάνουμε τον αριθμό των μέγιστων νημάτων στην πισίνα του IIS για την επεξεργασία αιτήσεων στατικού περιεχομένου, αυξάνεται ο αριθμός των context switces/sec, κάτι που έχει αντίκτυπο στην συνολική απόδοση του server.

Οι παράμετροι του πειράματός μας φαίνονται στις παρακάτω εικόνες:



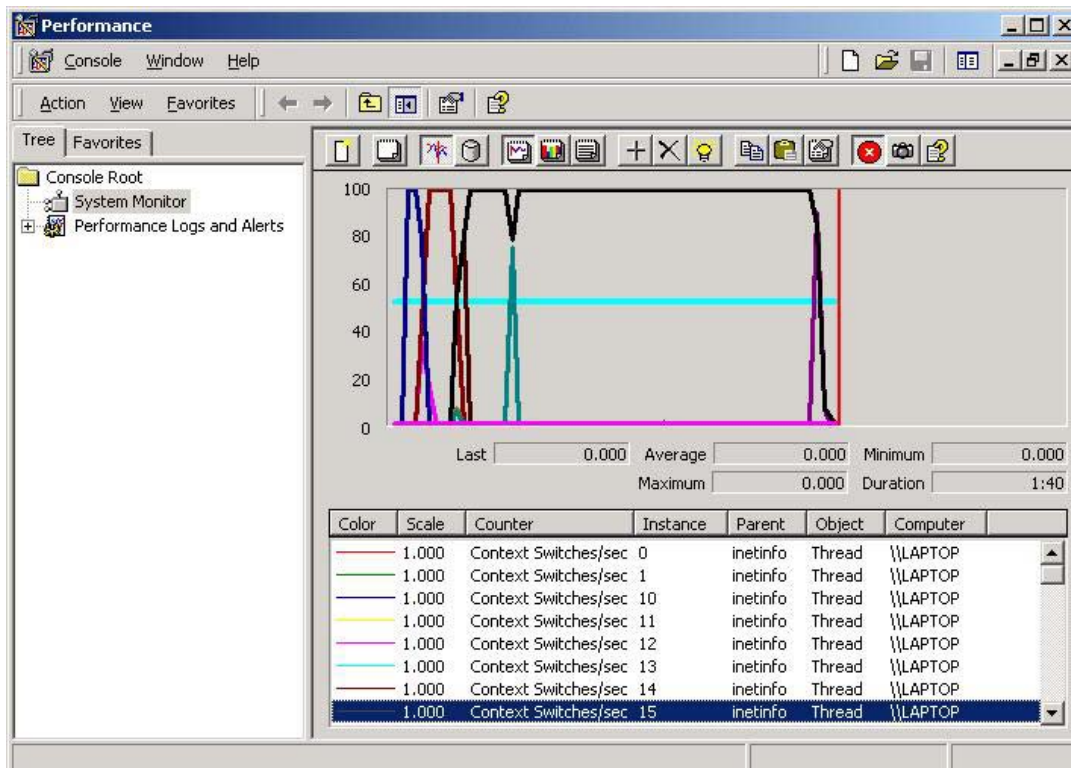


Αρχικά χρησιμοποιήσαμε την default τιμή (δηλαδή 4) της μεταβλητής *MaxPoolThreads* της registry και πήραμε τα παρακάτω αποτελέσματα στο System Monitor του server.



Βλέπουμε ότι ο αριθμός των νημάτων της inetinfo.exe είναι περίπου 28.

Διενεργήσαμε ένα δεύτερο πείραμα, μεταβάλλοντας αυτήν τη φορά την τιμή της μεταβλητής MaxPoolThreads στην τιμή 20. Είχαμε τα παρακάτω αποτελέσματα:



Βλέπουμε ότι ο αριθμός των νημάτων της inetinfo.exe αυξήθηκε και ότι τα context switches/sec μερικών από αυτών των νημάτων (προφανώς αυτών που επεξεργάζονται τις αιτήσεις) αυξήθηκαν κατά πολύ. Κάτι τέτοιο έχει μεγάλο αντίκτυπο στην απόδοση του server, καθώς τα context switches επιφέρουν επιπλέον φόρτο στον επεξεργαστή. Πρέπει να κάνετε πολλές δοκιμές, προκειμένου να βρείτε το σωστό αριθμό των μέγιστων νημάτων του IIS, ώστε να εξυπηρετούνται οι αιτήσεις σας με τον πιο αποδοτικό τρόπο.

### **Συμπέρασμα:**

Όταν αυξάνεται ο μέγιστος αριθμός των νημάτων που χρησιμοποιεί ο IIS για την εξυπηρέτηση αιτήσεων στατικού περιεχομένου, αυξάνονται κατά πολύ τα context switches/sec της διεργασίας inetinfo.exe, με αποτέλεσμα τη μείωση της απόδοσης του server.

---

## Εφαρμογές ASP

Όταν μια αίτηση HTTP φτάνει για ένα ASP script (ο πόρος έχει κατάληξη .asp) ο IIS προωθεί την αίτηση στο DLL asp.dll. Το DLL αυτό διαβάζει τον αιτούμενο πόρο από το δίσκο, διερμηνεύει το script και αποστέλλει την αίτηση πίσω στον client. Κάθε αίτηση που καταφθάνει στον IIS τοποθετείται στην ουρά. Αν υπάρχει ένα νήμα-εργάτης διαθέσιμο, τότε θα πάρει την πρώτη έτοιμη αίτηση από την ουρά για να την επεξεργαστεί.

Αν υπάρχει συμφόρηση στο σύστημα, το μήκος της ουράς θα αρχίσει να αυξάνει και θα παραμείνει σε υψηλές τιμές για αρκετό χρονικό διάστημα. Θα πρέπει να παρακολουθείτε το μετρητή *Active Server Pages: Requests Queued* ώστε να βεβαιωθείτε ότι δεν υπάρχει συμφόρηση. Όταν η ουρά φτάσει σε κάποιο άνω όριο, το οποίο καθορίζεται από μια συγκεκριμένη μεταβλητή στη registry, αποστέλλεται η απόκριση ‘Server Too Busy’ πίσω στον client.

Η μεταβλητή της metabase **ASPProcessorThreadMax** καθορίζει το μέγιστο αριθμό των νημάτων-εργατών που υπάρχουν μέσα στην πισίνα του IIS. Ως default η τιμή της μεταβλητής αυτής είναι 10 στον IIS 5.0. Η μεταβλητή της metabase **ASPRequestQueueMax** καθορίζει το μήκος της ουράς και ως default έχει τιμή 500 στον IIS 5.0.

Ο IIS 5.0 περιλαμβάνει έναν μηχανισμό που προσαρμόζει την τιμή της ASPProcessorMax “on the fly” (δηλαδή η πραγματική της τιμή δεν αλλάζει). Όταν η χρησιμοποίηση πέσει κάτω από ένα ποσοστό, τότε αυτό δείχνει ότι κάποια νήματα πιθανόν να μπλοκάρονται. Ο IIS 5.0 τότε αυξάνει τον αριθμό των ενεργών νημάτων έτσι

ώστε άλλες αιτήσεις να μπορούν να εξυπηρετηθούν. Όταν η χρησιμοποίηση του επεξεργαστή ξεπεράσει κάποιο ποσοστό, κάτι που δείχνει ότι υπάρχουν ελεύθερα νήματα, ο IIS 5.0 απενεργοποιεί κάποια νήματα ώστε να μειώσει το ποσοστό του context switching.

Για να ενεργοποιήσετε το μηχανισμό αυτόν θα πρέπει να θέσετε TRUE στη μεταβλητή της metabase **ASPThreadGateEnabled** (default FALSE). Για να καθορίσετε το κάτω όριο της χρησιμοποίησης, τροποποιήστε τη μεταβλητή **ASPThreadGateLoadLow** (default 75). Για να καθορίσετε το άνω όριο της χρησιμοποίησης, τροποποιήστε τη μεταβλητή **ASPThreadGateLoadHigh** (default 90).

### **Συμβουλές:**

Σας παραθέτουμε κάποιες συμβουλές σχετικά με τη βελτιστοποίηση των Web εφαρμογών, όπως είναι τα ASPs:

- 1. Αναβαθμίστε τους επεξεργαστές** Οι Web εφαρμογές επωφελούνται από τους γρηγορότερους επεξεργαστές.
- 2. Προσθέστε Επεξεργαστές** Τα συστατικά που καλούνται από τα ASPs μπορούν και πρέπει να είναι πολυνηματικά, το οποίο σημαίνει ότι μπορούν να τρέχουν ταυτόχρονα σε πολλαπλούς επεξεργαστές. Η πρόσθεση ενός επεξεργαστή σε ένα μονοεπεξεργαστικό σύστημα φέρνει το περισσότερο όφελος. Πάντως, δεν πρέπει να χρησιμοποιείτε περισσότερους από τέσσερις επεξεργαστές ανά σύστημα.
- 3. Προσθέστε Μνήμη** Προσθέτοντας μνήμη μπορείτε να βοηθηθείτε αν οι εφαρμογές τρέχουν μέσα στις δικές τους διεργασίες (ως default τα ASPs τρέχουν μέσα στη διεργασία του IIS 5.0, αλλά μπορείτε να επιλέξετε να τρέχουν μέσα στους δικούς τους χώρους μνήμης).
- 4. Ανασυγκροτήστε τους δίσκους σας** Αν η απόδοση της cache σας μειώνεται στο χρόνο, ανασυγκροτήστε τους δίσκους σας.

---

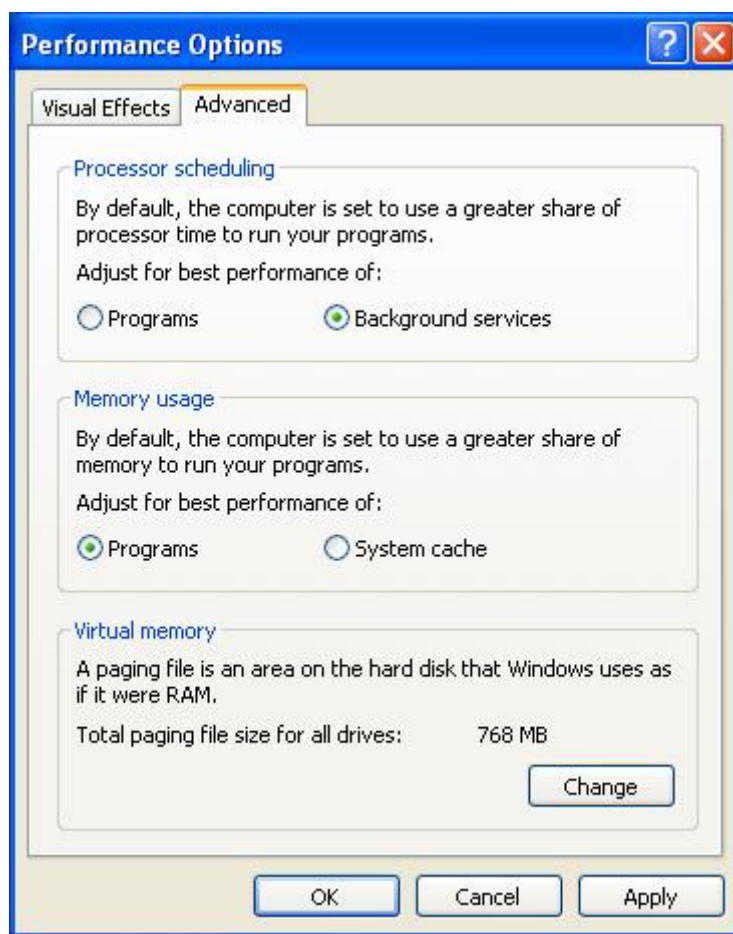
## **Συμβουλές για την επίλυση των bottlenecks επεξεργαστή**

### **1. Αναβαθμίστε την L2 Cache**

Όταν προσθέτετε ή αναβαθμίζετε επεξεργαστές, να επιλέγετε επεξεργαστές με μεγάλη δευτερεύουσα (L2) cache. Εφαρμογές αρχείων server, όπως είναι ο IIS, επωφελούνται από μια μεγάλη cache επεξεργαστή επειδή τα μονοπάτια εντολών τους περιλαμβάνουν πολλά διαφορετικά συστατικά. Προκειμένου να βελτιώσετε την απόδοση των servers που τρέχουν τον IIS 5.0, σας προτείνουμε να αγοράσετε μια μεγάλη cache επεξεργαστή (2MB ή περισσότερη αν είναι εξωτερική και μέχρι το μέγιστο διαθέσιμο αν είναι πάνω στο τσιπ της CPU).

## 2. Ελέγξτε πρώτα για απλά προβλήματα

Υπάρχουν κάποιες απλές ρυθμίσεις που μπορείτε να κάνετε στο server σας, οι οποίες μπορούν να οδηγήσουν σε μεγάλη βελτίωση της απόδοσης. Κάντε δεξί κλικ στο My Computer και επιλέξτε Properties. Θα δείτε το παράθυρο διαλόγου System Properties. Κάντε κλικ στο Advanced tab κι έπειτα στο κουμπί Performance Settings. Στο νέο παράθυρο διαλόγου κάντε κλικ στο Advanced tab. Λογικά πρέπει να βλέπετε το παρακάτω παράθυρο διαλόγου.



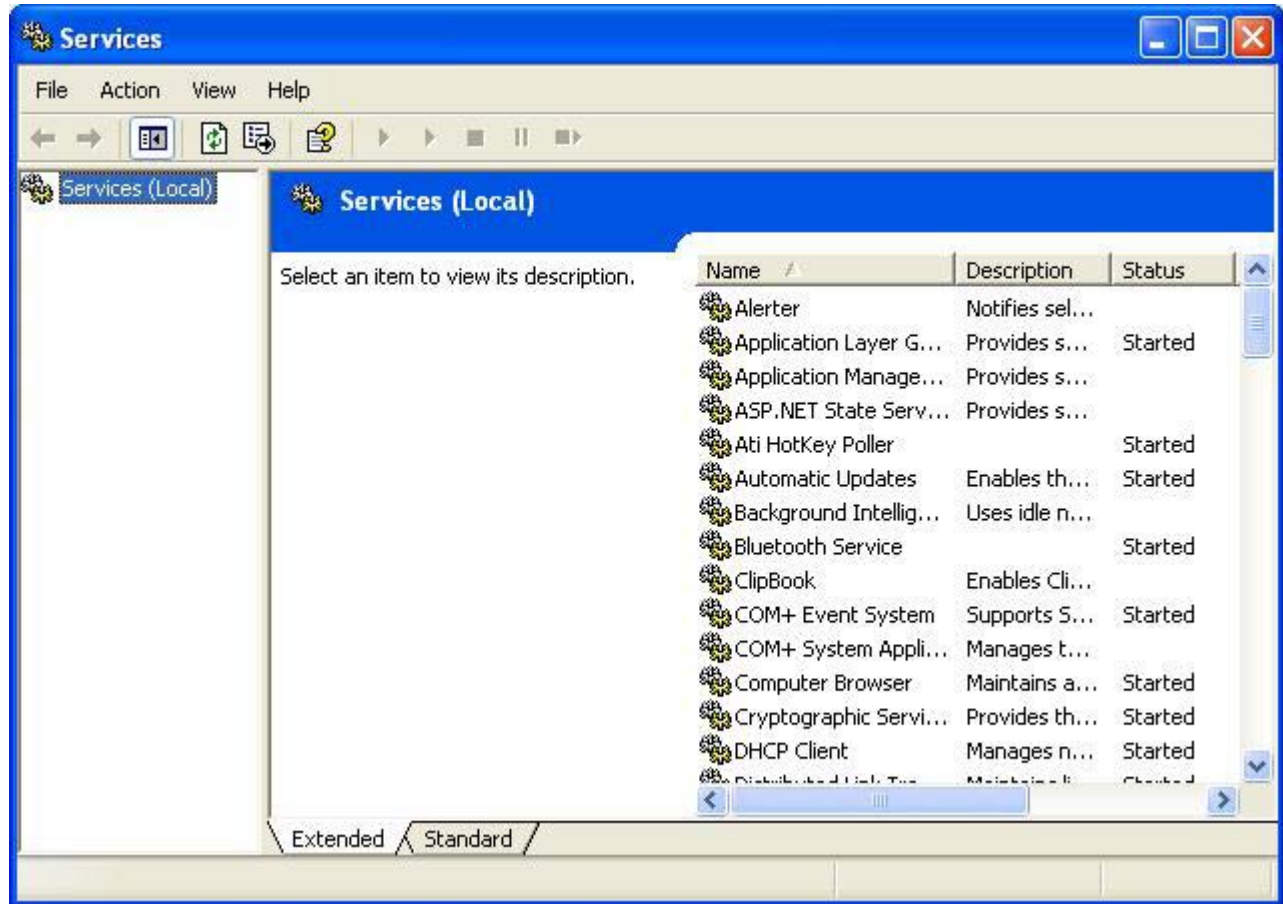
Δεν υπάρχει λόγος να έχετε επιλεγμένο το Programs στο Processor Scheduling εκτός κι αν ο server σας χρησιμοποιείται και ως workstation. Αφού οι περισσότερες εργασίες server τρέχουν στο background, θα θέλατε να έχετε background processor scheduling, ώστε να δίνεται προτεραιότητα στις εφαρμογές αυτές προκειμένου να τρέχουν καλύτερα.

## 3. Απομακρύνετε όσες υπηρεσίες δεν χρειάζονται

Κάντε κλικ στο *Start* → *Control Panel* → *Administrative Tools* → *Services*

Προκειμένου να επιτευχθεί η βελτιστοποίηση του server, δεν χρειάζεται να έχει κάποιος μειωμένη λειτουργικότητα. Το θέμα εδώ είναι να απομακρύνετε τις υπηρεσίες που δεν σας χρειάζονται καθόλου, έτσι ώστε άλλες υπηρεσίες να

χρησιμοποιούν τους πόρους που χρειάζονται καλύτερα. Για παράδειγμα, αν οι εταιρία σας δεν χρειάζεται καθόλου την υπηρεσία FTP, τότε το καλύτερο θα ήταν να την απενεργοποιήσετε. Για να το κάνετε αυτό, κάντε δεξί κλικ σε μια υπηρεσία που θέλετε να σταματήσετε και επιλέξτε Stop.





**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 8

**BOTTLENECKS ΜΝΗΜΗΣ**

**Σ**το κεφάλαιο αυτό δίνονται συμβουλές σχετικά με την αντιμετώπιση αυτού του είδους των bottlenecks. Εφόσον ο IIS χρειάζεται μνήμη για να τρέξει εφαρμογές που εξυπηρετούν μεγάλο αριθμό χρηστών και, αφού η μνήμη κοστίζει, οι τρόποι αντιμετώπισης των bottlenecks μνήμης που σας προτείνουμε μπορούν να σας βοηθήσουν να επιλύσετε τα προβλήματά σας, εξοικονομώντας χρήματα από πιθανή αγορά μεγαλύτερης μνήμης.

**Bottlenecks μνήμης μπορούν να προκύψουν καθαρά από έλλειψη φυσικής μνήμης, με αποτέλεσμα την αυξημένη πρόσβαση στο σκληρό δίσκο, πράγμα που με τη σειρά του οδηγεί σε μειωμένη απόδοση των εφαρμογών.**

---

## To Working Set του IIS

Θα δούμε πώς χρησιμοποιεί ο IIS τη μνήμη. Πρώτα απ'όλα, ο IIS πρέπει να φορτωθεί στη μνήμη. Όταν όλες οι υπηρεσίες του βρίσκονται στη μνήμη, ο IIS απαιτεί 2.5 MB μνήμης. Οι πραγματικές απαιτήσεις μνήμης, όμως, μπορεί να ποικίλουν, από τη στιγμή που το working set μπορεί να γίνεται paged in και out από τη μνήμη, όταν αυτό απαιτείται. Το μέγεθος του working set (δηλαδή του τμήματος της διεργασίας που βρίσκεται στη μνήμη) κι επομένως και οι απαιτήσεις για μνήμη του IIS για τη σελίδα σας, θα εξαρτηθούν από τον αριθμό των χαρακτηριστικών του IIS που χρησιμοποιούνται καθώς και από το ποσό της διαθέσιμης μνήμης.

Οι μετρητές που θα σας βοηθήσουν να παρακολουθήσετε το working set του IIS είναι οι ακόλουθοι:

Μετρητές	Ενδείξεις
<i>Process\Working Set : inetinfo.exe</i>	Το συνολικό μέγεθος σε bytes του working set του IIS.
<i>Process\ Pool Paged Bytes : inetinfo.exe</i>	Το κομμάτι εκείνο της διεργασίας inetinfo.exe (μετρημένο σε bytes) το οποίο είναι δυνατό να μετακινείται από τη μνήμη στο δίσκο, όταν αντιμετωπίζονται καταστάσεις έλλειψης μνήμης.
<i>Process\ Pool Nonpaged Bytes : inetinfo.exe</i>	Το κομμάτι εκείνο της διεργασίας inetinfo.exe (μετρημένο σε bytes) το οποίο είναι τοποθετημένο <u>μόνιμα</u> στη μνήμη.

**Το κείμενο, τα δεδομένα και τα τμήματα της στοίβας της διεργασίας inetinfo.exe που κατοικούν στη μνήμη συνθέτουν το working set του IIS.**

Προκειμένου ο IIS να επεξεργαστεί την κάθε αίτηση, δημιουργεί και διατηρεί κάποιες εσωτερικές δομές δεδομένων, με πληροφορία σχετική με την αίτηση και την κατάστασή της. Κάθε αίτηση απαιτεί περίπου 10 KB δεδομένων του working set. Όταν η επεξεργασία της αίτησης ολοκληρωθεί, η μνήμη που καταλάμβανε απελευθερώνεται. Γι' αυτόν το λόγο, το ολικό ποσό της μνήμης που καταναλώνεται εξαρτάται από τον αριθμό των ταυτόχρονων συνδέσεων που υφίστανται επεξεργασία από το server.

---

## ASP Caching

Κάθε αίτηση για μια σελίδα ASP έχει τις δικές της ASP Template Cache και ASP Script Engine Cache. Η ASP Template Cache αποθηκεύει templates (=προμεταγλωττισμένες σελίδες ASP σε μορφοποίηση κειμένου). Το μέγεθός της καθορίζεται από τη ρύθμιση της μεταβλητής **AspScriptCacheSize** της metabase, που ως default έχει τιμή 250. Η ASP Script Engine Cache κρατάει ASP templates που δεν έχουν μεταγλωττιστεί σε δυαδικό κώδικα. Το μέγεθός της καθορίζεται από τη ρύθμιση της μεταβλητής **AspScriptEngineCacheMax** της metabase, που ως default έχει τιμή 125. Η συσχέτιση μεταξύ των δύο είναι ότι μια σελίδα ASP εναποθηκεύεται μια φορά στην Template Cache, αλλά μπορεί να εμφανιστεί πολλές φορές στην Script Engine Cache, αν η σελίδα αυτή εκτελείται σε πολλά νήματα ταυτόχρονα.

Ένα site με πολύ μνήμη και πολλές ξεχωριστές ASP σελίδες που τυγχάνουν πολλών hits, απαιτούν μια αύξηση της μεταβλητής **AspScriptFileCacheSize**. Υπάρχει πολύ μικρότερη ανάγκη για αύξηση της **AspScriptEngineCacheMax**. Γενικά θα πρέπει να ισχύει  $AspScriptEngineCacheMax \geq (CPUs + 1) * AspProcessorThreadMax$ .

---

## IIS Object Cache

Η IIS Object Cache είναι ένα πολύ σημαντικό τμήμα του working set του IIS. Ο IIS λαμβάνει συνεχώς αιτήσεις για αρχεία που πρέπει να ανακτηθούν από το δίσκο (ή από την cache του συστήματος αρχείων, όπως θα δούμε παρακάτω) και στέλνει τα αρχεία αυτά πίσω στους clients που έκαναν τις αιτήσεις. Για να είναι σε θέση να ανακτήσει ένα αρχείο από το δίσκο, ο IIS χρειάζεται να αποκτήσει ένα handle για το αρχείο αυτό από το σύστημα αρχείων. Προκειμένου να αποκτήσει το handle, το σύστημα αρχείων πρέπει να εντοπίσει τις δομές δεδομένων που παριστάνουν το αρχείο μέσα στο σύστημα αρχείων και να στείλει το handle αυτό πίσω στον IIS, ώστε να μπορέσει αυτός να το χρησιμοποιήσει σε μεταγενέστερες κλήσεις API για αυτόν τον πόρο. Η μετάφραση του

ονόματος ενός πόρου σε ένα handle είναι μια χρονοβόρος διαδικασία και, από τη στιγμή που τον περισσότερο χρόνο οι χρήστες ζητούν ένα μικρό υποσύνολο του περισσότερο δημοφιλούς περιεχομένου στο site, οι σχεδιαστές του IIS αποφάσισαν να εναποθηκεύουν αυτά τα handles. Η IIS Object Cache χρησιμοποιείται για την εναποθήκευση τέτοιων handles, directory listings και άλλων αντικειμένων στα οποία ο IIS χρειάζεται να προσβαίνει κατά την επεξεργασία των εισερχόμενων αιτήσεων.

### **Ο IIS αναθέτει το 10% της φυσικής μνήμης του υπολογιστή για την Object Cache.**

Οι μετρητές που θα σας βοηθήσουν να παρακολουθήσετε την IIS Object Cache είναι οι ακόλουθοι:

<b>Μετρητές</b>	<b>Ενδείξεις</b>
<i>Internet Information Service Global\ Cache Size</i>	Το συνολικό μέγεθος της cache.
<i>Internet Information Service Global\ Cache Used</i>	Το κομμάτι της cache που χρησιμοποιείται.
<i>Internet Information Service Global\ File Cache Flushes</i>	Ο αριθμός των φορών που ένα αντικείμενο απομακρύνθηκε από την cache επειδή έληξε ο χρόνος ζωής του (TTL) ή επειδή αυτό τροποποιήθηκε.
<i>Internet Information Service Global\ File Cache Hits</i>	Ο αριθμός των hits της cache από την έναρξη των υπηρεσιών του IIS.
<i>Internet Information Service Global\ %File Cache Hits</i>	Το ποσοστό των αιτήσεων που ικανοποιήθηκαν από την cache.
<i>Internet Information Service Global\ File Cache Misses</i>	Ο αριθμός των αποτυχιών της cache από την έναρξη των υπηρεσιών του IIS.
<i>Internet Information Service Global\ Directory Listings</i>	Ο συνολικός αριθμός των directory listings στην cache τη συγκεκριμένη στιγμή.
<i>Internet Information Service Global\ Objects</i>	Ο συνολικός αριθμός των αντικειμένων που βρίσκονται στην cache τη συγκεκριμένη στιγμή. Αυτός ο αριθμός ισούται με το άθροισμα των Cached File

Handles, των directory listings και άλλων αντικειμένων (π.χ. Database Connector ερωτήματα και αποκρίσεις).

Όταν ένα αντικείμενο προστίθεται στην cache για πρώτη φορά, αντιστοιχίζεται σε αυτό μια τιμή time-to-live (TTL). Κάθε φορά που το αντικείμενο προσπελαύνεται, το TTL ανανεώνεται. Αν το αντικείμενο δεν προσπελαστεί για αρκετή ώρα, το TTL λήγει. Ένα νήμα *object cache scavenger* (εργάτης καθαριότητας) τρέχει περιοδικά και απομακρύνει από την cache τα αντικείμενα, το TTL των οποίων έχει λήξει ή αντικείμενα που έχουν αλλάξει από τότε που εναποθηκεύτηκαν. Ο μετρητής **Cache Flushes** αναφέρει τον συνολικό αριθμό των αντικειμένων που απομακρύνθηκαν από την cache από τον *object cache scavenger*.

Το TTL ελέγχεται από ένα κλειδί της registry. Το κλειδί αυτό ονομάζεται **ObjectCacheTTL** και έχει default τιμή 30 δευτερόλεπτα. Αν θέλετε να τροποποιήσετε την τιμή αυτήν, θα πρέπει να δημιουργήσετε το κλειδί αυτό (REG\_DWORD) μέσα στη registry, κάτω από το path *HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Inetinfo\Parameters* με τον τρόπο που έχει δείχτει στην προηγούμενη ενότητα. Η τιμή που εισάγετε είναι σε δευτερόλεπτα.

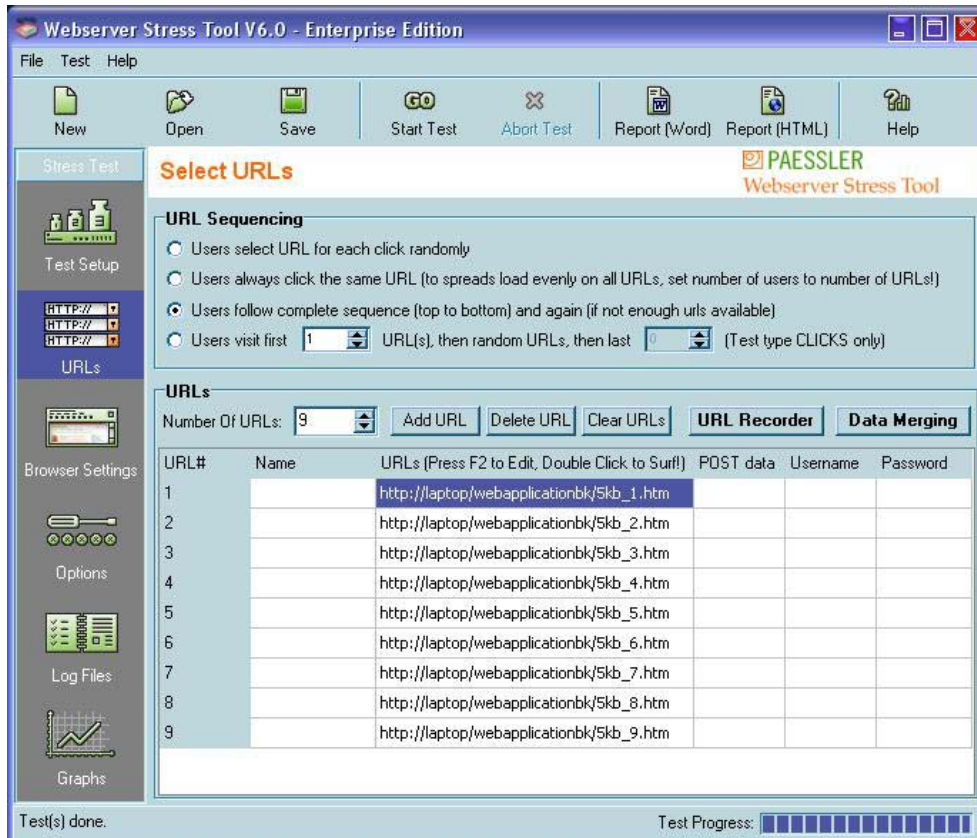
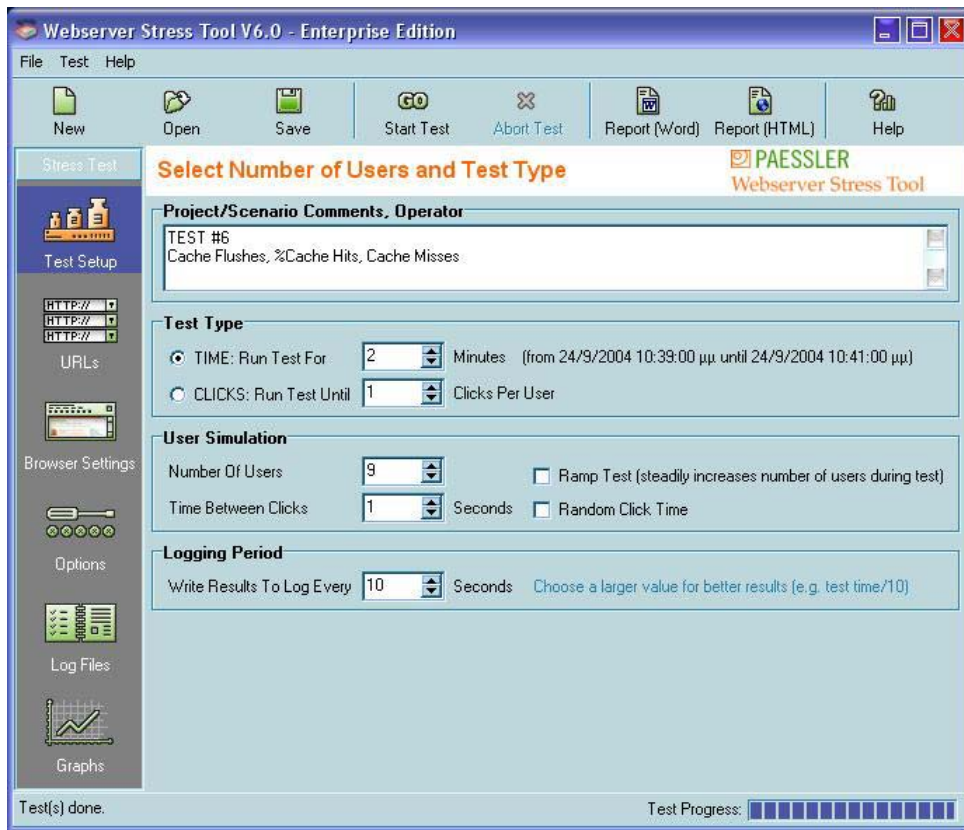
Από τη στιγμή που δεν μπορείτε να έχετε τον έλεγχο του μεγέθους της IIS Object Cache, ο μονος τρόπος για να ρυθμίσετε την αποδοτικότητα είναι να ρυθμίσετε την τιμή TTL. Η cache χάνει την αποδοτικότητά της όταν γίνεται paged out. Αφού είναι μέρος του working set του IIS, μπορεί να γίνει paged out όταν δεν υπάρχει αρκετός χώρος στη μνήμη του συστήματος. Είναι πολύ σημαντικό να προσπαθήσετε να διατηρήσετε το working set του IIS μέσα στη φυσική μνήμη.

---

## Test #5

Προκειμένου να δούμε πώς συμπεριφέρεται η Object Cache σε μεταβολές του TTL, διενεργήσαμε το παρακάτω πείραμα:

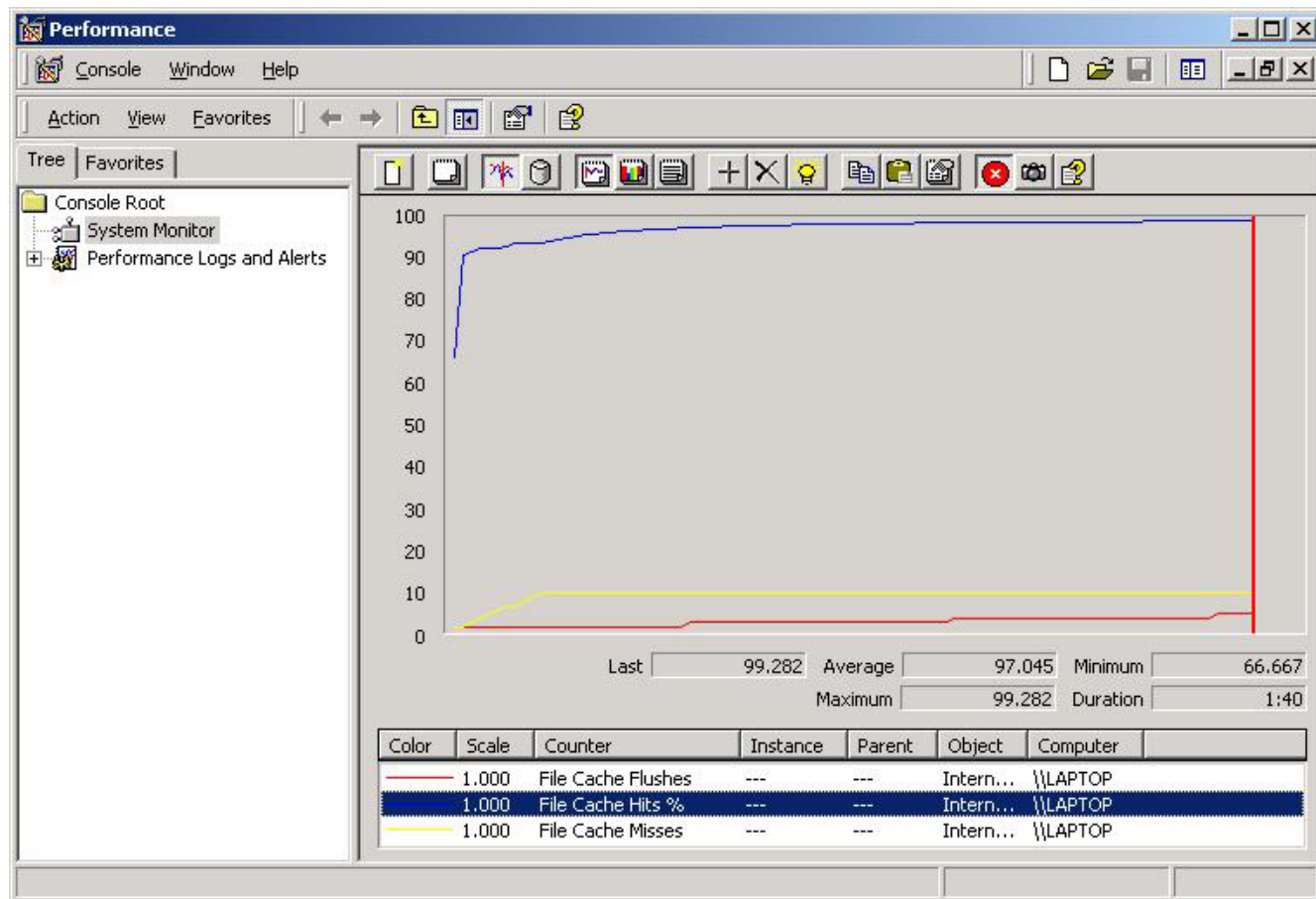
Θέσαμε 9 ταυτόχρονους χρήστες, οι οποίοι κλικάρουν ταυτόχρονα σε 9 διαφορετικά στατικά URLs με τη σειρά που τα τοποθετήσαμε (δηλαδή όλοι στο πρώτο, μετά όλοι στο δεύτερο κ.ο.κ.). Το πείραμα διενεργήθηκε για 2 λεπτά και ο χρόνος ανάμεσα στα κλικς είναι ένα δευτερόλεπτο. Παραθέτουμε τις ρυθμίσεις του τεστ:



Θέλουμε να παρακολουθήσουμε τους εξής μετρητές:

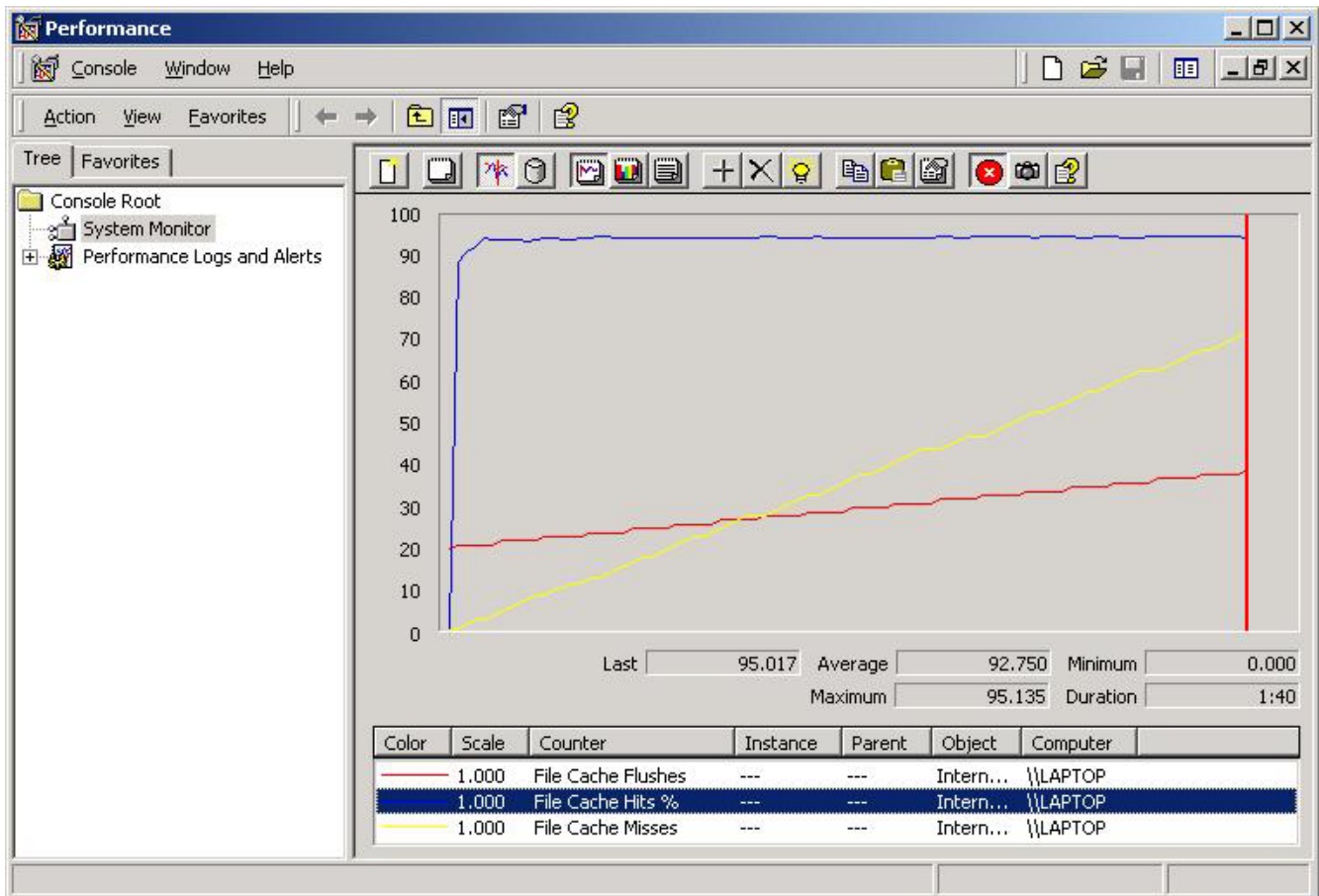
*Internet Information Service Global\File Cache Flushes*  
*Internet Information Service Global\File Cache Hits*  
*Internet Information Service Global\File Cache Misses*

Αρχικά τρέχουμε το πείραμα με την default τιμή για το TTL (30 δευτερόλεπτα). Τα αποτελέσματα του System Monitor του server ήταν τα εξής:



Αρχικά η Object Cache κάνει ένα miss κάθε δευτερόλεπτο, αφού κάθε δευτερόλεπτο οι 9 χρήστες κάνουν κλικ σε μια νέα σελίδα που δεν υπάρχει στην cache. Έτσι από την έναρξη του IIS η cache αποτυγχάνει 9 φορές μέχρι να εναποθηκευθούν και οι 9 σελίδες στην cache και από εκεί κι έπειτα δεν αποτυγχάνει καμιά φορά, αφού κάθε 9 δευτερόλεπτα, οι σελίδες που βρίσκονται στην cache ανανεώνουν το TTL τους πριν αυτό λήξει (30 δευτερόλεπτα). Επίσης η cache κάνει flush σπάνια (και αυτό οφείλεται σε ληγμένα TTLS άλλων αντικειμένων – όχι των εν λόγω σελίδων). Έτσι, όπως είναι αναμενόμενο, το ποσοστό των hits στην cache ολοένα αυξάνεται και τείνει να φτάσει το 100%.

Τώρα επαναλαμβάνουμε το πείραμα, αφού μεταβάλλουμε πρώτα την τιμή της μεταβητής ObjectCacheTTL της registry, δίνοντάς τη την τιμή 2. Τα αποτελέσματα αυτήν τη φορά ήταν τα εξής:



Τώρα βλέπουμε ότι κάθε δύο δευτερόλεπτα η cache κάνει flush. Αυτό συμβαίνει γιατί κάθε σελίδα που μπαίνει στην cache μένει μόνο 2 δευτερόλεπτα και όταν ο πρώτος χρήστης προσπαθεί να την ξαναπροσπελάσει μετά από 9 δευτερόλεπτα, αυτή έχει ήδη απομακρυνθεί από την cache. Έτσι τα cache misses αυξάνουν γραμμικά, όπως φαίνεται και από το γράφημα. Τέλος το ποσοστό των hits της cache μένει σταθεροποιημένο κοντά στο 90%, κάτι που είναι λογικό, αφού πάντα 1/9 χρήστες κάνει miss, η σελίδα μπαίνει στην cache και οι υπόλοιποι 8/9 χρήστες κάνουν hit.

### **Συμπέρασμα:**

Όταν μειώνουμε το TTL είναι πολύ πιθανό να απομακρυνθούν πρόωρα κάποια δημοφιλή αντικείμενα, με αποτέλεσμα να έχουμε περισσότερες αποτυχίες της cache, οπότε και μειωμένη απόδοση. Πρέπει να δοθεί ιδιαίτερη προσοχή και μελέτη για να βρεθεί η ιδανική τιμή TTL για τις συγκεκριμένες σελίδες που παρέχει ο server σας.



### Συμβουλή:

Σε sites που βιώνουν μεγάλο φόρτο, ένα μικρό διάστημα TTL είναι αρκετό, ενώ σε sites με μικρότερο φόρτο θα χρειαστείτε μεγαλύτερο διάστημα. Αν ο μετρητής Cache Flushes παρουσιάσει μείωση και ο μετρητής %Cache Hits παρουσιάσει αύξηση, τότε έχετε επιτύχει βελτίωση της απόδοσης του server σας.

Ένα άλλο συστατικό της χρησιμοποίησης της μνήμης που αυξάνεται καθώς αυξάνεται ο φόρτος στο server είναι η ανάθεση των μπλοκ ελέγχου για τη διατήρηση της πληροφορίας κατάστασης για κάθε σύνδεση. Κάθε αίτηση σύνδεσης προς τον IIS (ή προς κάθε εφαρμογή που χρησιμοποιεί το TCP) προκαλεί στο επίπεδο TCP μέσα στο λειτουργικό σύστημα να αναθέσει ένα μπλοκ ελέγχου μετάδοσης (TCB) και να το προσθέσει σε έναν πίνακα κατακερματισμού. Όταν η σύνδεση τερματίζεται, το TCB παραμένει για κάποιο διάστημα χρόνου για λόγους ασφαλείας και μετά, είτε ανατίθεται σε άλλη αίτηση σύνδεσης είτε απελευθερώνεται, ανάλογα με την ένταση του φόρτου. Αυτό το κομμάτι της μνήμης που ανατίθεται στο TCB είναι nonpaged, επομένως βρίσκεται πάντα στη μνήμη. Κάθε σύνδεση απαιτεί 10KB μνήμης για το TCB. Μπορείτε να καθορίσετε τον μέγιστο αριθμό ταυτόχρονων συνδέσεων που 'αντέχει' η μνήμη σας, υπολογίζοντας το μέγεθος της μνήμης που έχετε στη διάθεσή σας και πολλαπλασιάζοντάς το επί 10KB.

---

## File System Cache

Η File System Cache είναι το working set του συστήματος αρχείων. Αυτή η cache είναι μια περιοχή μέσα στη φυσική μνήμη, όπου το σύστημα αρχείων αποθηκεύει τα πιο πρόσφατα και πιο συχνά χρησιμοποιούμενά του δεδομένα. Ως default, το σύστημα δεσμεύει περίπου το 50% της φυσικής μνήμης για την File System Cache, αλλά το σύστημα μπορεί να μεταβάλλει δυναμικά το μέγεθος της μνήμης αυτής, σε περίπτωση που αντιμετωπίσει έλλειψη μνήμης.

Ο IIS βασίζεται στο λειτουργικό σύστημα για την αποθήκευση και την ανάκτηση των συχνά χρησιμοποιούμενων Web σελίδων και άλλων αρχείων από τη File System Cache. Η File System Cache είναι ιδιαίτερα χρήσιμη για servers στατικών σελίδων Web, επειδή οι σελίδες Web έχουν την τάση να χρησιμοποιούνται σε επαναλαμβανόμενα, προβλέψιμα μοτίβα.

Ο IIS 5.0 διαβάζει πάντα σειριακά. Η σειριακή ανάγνωση επωφελείται από ένα χαρακτηριστικό του Cache Manager που καλείται *read ahead*. Αυτό συμβαίνει όταν ο προφητικός αλγόριθμος του Cache Manager ανιχνεύει σειριακές αναγνώσεις και αρχίζει να διαβάζει μεγαλύτερα μπλοκ δεδομένων σε κάθε λειτουργία ανάγνωσης. Τα read aheads μπορούν να παρέχουν μια σημαντική ώθηση στην απόδοση μιας διεργασίας.

Ο IIS 5.0 χρησιμοποιεί και τη File System Cache και την IIS Object Cache και μάλιστα μερικές φορές τις χρησιμοποιεί και ταυτόχρονα. Όταν ένα νήμα που ανήκει σε μια υπηρεσία του IIS χρειάζεται να ανοίξει ένα αρχείο, το νήμα ζητάει ένα file handle από το λειτουργικό σύστημα. Όταν λάβει το handle, το νήμα το χρησιμοποιεί για να ανοίξει το αρχείο. Έπειτα, αν ο χώρος της μνήμης το επιτρέπει, το νήμα αποθηκεύει το handle στην IIS Object Cache και το σύστημα αποθηκεύει τα δεδομένα του αντίστοιχου αρχείου στη File System Cache. Αργότερα, αν το νήμα αυτό (ή όποιο άλλο νήμα) χρειαστεί το αρχείο, το file handle μπορεί να ανακτηθεί από την IIS Object Cache και τα περιεχόμενα του αρχείου μπορούν να ανακτηθούν από τη File System Cache.

Οι μετρητές που θα σας βοηθήσουν να μελετήσετε τη File System Cache είναι οι εξής:

<b>Μετρητές</b>	<b>Ενδείξεις</b>
<i>Memory\Cache Bytes</i>	Το μέγεθος της cache σε bytes. Αυτός ο μετρητής δίνει την τελευταία παρατηρημένη τιμή, όχι το μέσο όρο.
<i>Memory\Cache Faults/sec</i>	Δείχνει το πόσο συχνά τα δεδομένα που αναζητώνται στη File System Cache, δεν βρίσκονται εκεί. Αυτός ο μετρητής περιλαμβάνει και αποτυχίες για δεδομένα που βρίσκονται σε κάποιο άλλο μέρος στη μνήμη, καθώς επίσης και αποτυχίες που απαιτούν λειτουργίες δίσκου προκειμένου να ανακτήσουν τα δεδομένα που ζητήθηκαν.
<i>Cache\Copied Reads/sec</i>	Η συχνότητα των αναγνώσεων από σελίδες της File System Cache που μεταφέρουν ένα αντίγραφο δεδομένων μνήμης από την cache στους buffers της εφαρμογή.
<i>Cache MDL Reads/sec</i>	Ο μετρητής αυτός δείχνει πόσο συχνά το σύστημα επιχειρεί να διαβάσει μεγάλα μπλοκ δεδομένων από την cache. Τα Memory Descriptor List (MDL) Reads είναι λειτουργίες ανάγνωσης στις οποίες το σύστημα χρησιμοποιεί μια λίστα των φυσικών διευθύνσεων για κάθε σελίδα για την εύκολη εύρεση τους. Μέσω του MDL interface τα δεδομένα πηγαίνουν από το δίσκο στην cache και από εκεί κατευθείαν στην κάρτα δικτύου.
<i>Memory\Page Faults/sec</i>	Αν μια διεργασία ζητήσει μια σελίδα από τη μνήμη και το σύστημα δεν μπορέσει να τη βρει στην ζητούμενη

τοποθεσία, τότε έχουμε ένα page fault. Αν η σελίδα είναι κάπου αλλού μέσα στη μνήμη το page fault λέγεται soft, ενώ αν η σελίδα πρέπει να ανακτηθεί από το δίσκο, το page fault λέγεται hard. Ο μετρητής δεν κάνει διάκριση σε hard και soft page faults.

*Memory\Page Reads/sec*

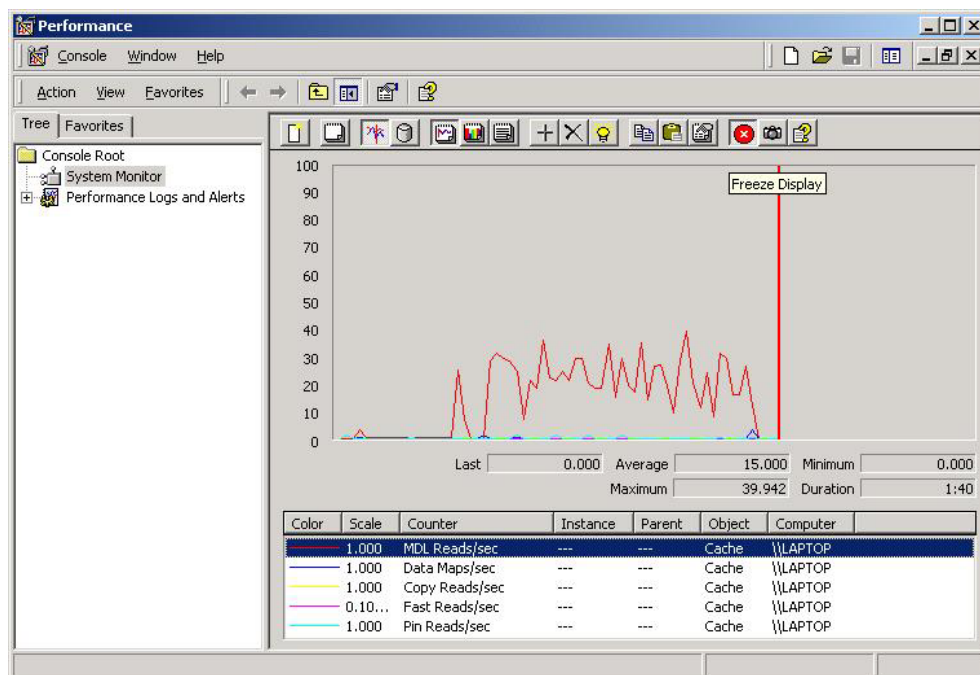
Δείχνει τον αριθμό των αναγνώσεων σελίδων από το σκληρό δίσκο.

---

## Test #6

Ο Cache Manager, στο επίπεδο εκτέλεσης μέσα στο λειτουργικό σύστημα, παρέχει πέντε μηχανισμούς για την πρόσβαση στις υπηρεσίες του. Έχουμε ήδη συμπεριλάβει στους παραπάνω μετρητές το σημαντικότερο μηχανισμό, που είναι ο MDL. Οι υπόλοιποι τέσσερις είναι οι εξής: Data Map, Pin, Copy και Fast Read mechanisms. Ο IIS 5.0 κάνει ευρύτατη χρήση του MDL μηχανισμού και γι' αυτό δεν θα παρουσιάσουμε καθόλου τους υπόλοιπους. Με το παρακάτω τεστ θα δείξουμε ότι η χρήση του MDL interface ξεπερνάει κατά πολύ τη χρήση των interfaces των άλλων μηχανισμών.

Διενεργήσαμε ένα Ramp test 500 χρηστών, οι οποίοι κάνουν κλικ όλοι σε μια στατική σελίδα μεγέθους ενός MB. Το System Monitor του server εμφάνισε τα εξής αποτελέσματα:



## Συμπέρασμα:

Ο IIS 5.0 κάνει ευρεία χρήση του MDL μηχανισμού για πρόσβαση στις υπηρεσίες του Cache Manager, η οποία ξεπερνάει κατά πολύ τη χρήση άλλων μηχανισμών.

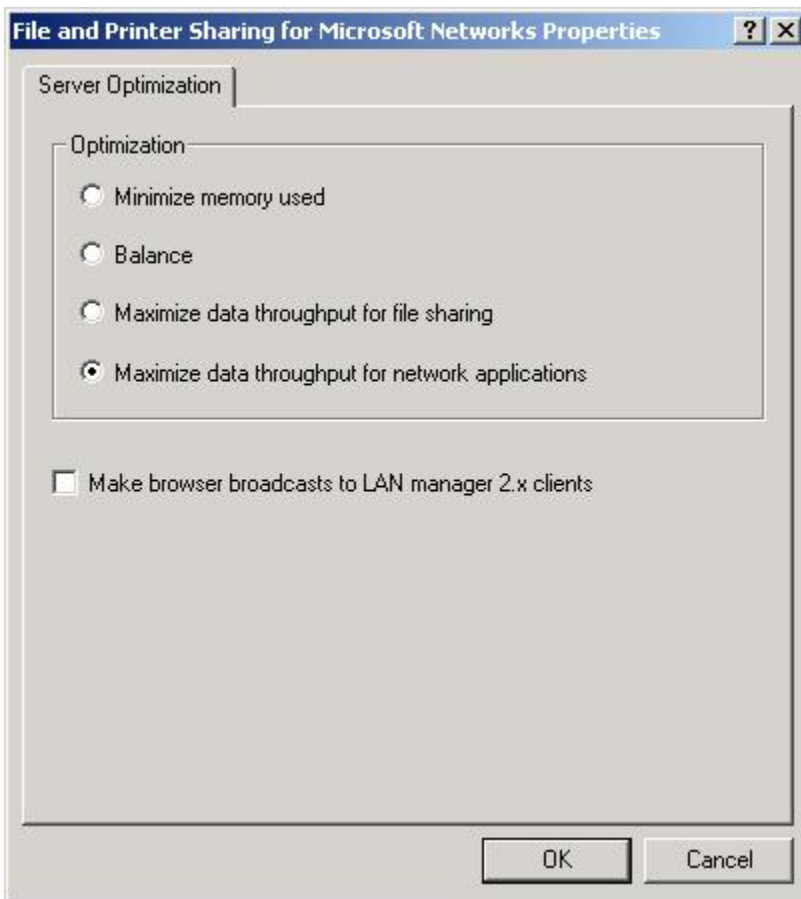
---

## Συμβουλές για τη βελτιστοποίηση της χρήσης μνήμης

1. Οι servers που τρέχουν τον IIS 5.0, όπως άλλοι file servers υψηλής απόδοσης, επωφελούνται από την ύπαρξη επαρκούς μνήμης. Γενικά, όσο περισσότερη μνήμη προσθέτετε, τόσο αποδοτικότερα δουλεύουν οι servers. Ο IIS 5.0 απαιτεί 64 MB μνήμης το ελάχιστο, ενώ προτείνεται να χρησιμοποιεί τουλάχιστον 128MB. Αν τρέχετε εφαρμογές memory-intensive (που απαιτούν δηλαδή πολύ μνήμη), τότε μπορεί ο server σας να απαιτεί πολύ μεγαλύτερη ποσότητα μνήμης για να τρέχει ιδανικά (για παράδειγμα, οι περισσότεροι servers που εξυπηρετούν το Web site microsoft.com διαθέτουν τουλάχιστον 512MB μνήμης). Εν κατακλείδι, θα πρέπει να παρέχετε αρκετή φυσική μνήμη για το working set της διεργασίας *inetinfo.exe* ώστε να αποφευχθούν σε μεγάλο βαθμό τα page faults.
2. Θα πρέπει να παρέχετε αρκετή μνήμη για την IIS Object Cache ώστε να επιτευχθεί ένα hit ratio τουλάχιστον 80%. Είσης θα πρέπει να παρέχετε αρκετή μνήμη και για τη File System Cache. Για να καθορίσετε πόση μνήμη χρειάζεστε για αυτήν την cache, μετρίστε από τα log files το μέγεθος των αρχείων που αποτελούν το 70-90% των συνολικών αιτήσεων.
3. **Βελτιώστε την οργάνωση των δεδομένων** Να κρατάτε τα σχετικά Web αρχεία στα ίδια λογικά partitions ενός δίσκου. Κρατώντας τα αρχεία μαζί, μπορείτε να βελτιώσετε την απόδοση της File System Cache. Επίσης, κάντε defragment τους δίσκους σας. Ακόμη και τα καλά οργανωμένα αρχεία κάνουν αρκετή ώρα να ανακτηθούν αν είναι τεμαχισμένα.
4. **Αντικαταστήστε ή μετατρέψτε τις εφαρμογές CGI** Οι CGI εφαρμογές χρησιμοποιούν πολύ περισσότερο χρόνο επεξεργαστή και χώρο μνήμης απ'ότι οι ισοδύναμες εφαρμογές ASP ή ISAPI.
5. **Επαναπρογραμματίστε την IIS Object Cache** Προσπαθήστε να επιμηκύνετε την περίοδο που μπορεί ένα χρησιμοποιημένο αντικείμενο να παραμείνει στην cache (χρησιμοποιήστε τη ρύθμιση ObjectCacheTTL με τον τρόπο που περιγράψαμε προηγουμένως).
6. **Περιορίστε τις συνδέσεις** Αν ο server σας δεν έχει αρκετή μνήμη, το να περιορίσετε τον αριθμό των συνδέσεων στο server μπορεί να σας βοηθήσει να

ανακουφίσετε την έλλειψη μνήμης, επειδή κάποιο μέρος της φυσικής μνήμης (περίπου 10KB ανά σύνδεση) καταναλώνεται από τις δομές δεδομένων που χρησιμοποιεί το σύστημα για να ελέγχει τις συνδέσεις.

7. **Τερματίστε αχρείαστα χαρακτηριστικά** Μπορείτε επίσης να απενεργοποιήσετε την ενίσχυση απόδοσης για τις εφαρμογές στο προσκήνιο. Επιπλέον, τις φορές που δεν θέλετε να παρακολουθείτε την απόδοση του server σας, καλό είναι να απενεργοποιείτε το logging γιατί κι αυτό έχει μια μικρή επίπτωση στην απόδοση του server.
  
8. **Αλλάζτε την ισορροπία από την File System Cache στο IIS 5.0 working set** Ως default, οι servers που τρέχουν σε λειτουργικό σύστημα Windows 2000 είναι ρυθμισμένοι να δίνουν προτεραιότητα στην File System Cache παρά στα working sets των διεργασιών, κατά τη διαδικασία ανάθεσης χώρου μνήμης. Παρόλο που οι servers που χρησιμοποιούν τον IIS 5.0 επωφελούνται από μια μεγάλη File system Cache, η ρύθμιση **Maximize Throughput for File Sharing** συχνά έχει ως αποτέλεσμα pageable κώδικας να γράφεται στο δίσκο, που με τη σειρά του οδηγεί σε μεγάλες καθυστερήσεις στην επεξεργασία. Προκειμένου να αποφύγετε αυτές τις καθυστερήσεις, επιλέξτε στα Server properties την επιλογή **Maximize data throughput for network applications**. Για να το κάνετε αυτό:
  - Στο desktop, ανοίξτε το **My Computer** και επιλέξτε **Network and Dial-up Connections**.
  - Κάντε δεξί κλικ στο **Local Area Connection** και ανοίξτε τα **Properties**.
  - Επιλέξτε **File and Printer Sharing for Microsoft Networks** και επιλέξτε **Properties**.
  - Στο κουτί **Server Optimization** επιλέξτε **Maximize data throughput for network applications**.



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΚΕΦΑΛΑΙΟ 9

**BOTTLENECKS ΔΙΣΚΟΥ**

**Α**φού πρώτα έχετε ήδη αποκλείσει την περίπτωση ύπαρξης των δύο προηγούμενων bottlenecks και ενώ το πρόβλημα απόδοσης εξακολουθεί να υπάρχει, τότε είναι πολύ πιθανό να αντιμετωπίζετε ένα bottleneck δίσκου. Στο κεφάλαιο αυτό, όπως και στα προηγούμενα, παρατίθενται συμβουλές επίλυσης προβλημάτων απόδοσης που οφείλονται στο σκληρό σας δίσκο.

---

## Τι είναι το bottleneck δίσκου;

Έχετε ποτέ παρακολουθήσει το φωτάκι του σκληρού σας δίσκου να τρεμοπαίζει για πάρα πολύ ώρα; Ίσως να το ακούτε κιάλας καθώς το λειτουργικό σύστημα ‘χτυπάει’ επανειλημμένα το δίσκο. Τελικά, αν το σύστημα του drive δεν σταματήσει, τότε αυτό θα αποτύχει, αφού εξ’άλλου είναι μηχανική συσκευή. Πριν ακόμη προλάβετε να αντιληφθείτε μια τέτοια κατάσταση, οι χρήστες του δικτύου σας θα αρχίσουν να παραπονιούνται σχετικά με τους αργούς χρόνους επεξεργασίας και αναμονής, κυρίως για εφαρμογές, όπως συστήματα βάσεων δεδομένων. Μπορεί να δείτε ακόμη και το server να παγώνει, λόγω της έλλειψης του μοναδικού πόρου που πρέπει αυτός να χειρίζεται, δηλαδή του ελεύθερου χώρου.

Αυτά είναι όλα συμπτώματα ενός bottleneck δίσκου (και μάλιστα ενός μεγάλου bottleneck δίσκου στις περισσότερες περιπτώσεις). Ακόμη και μικρά bottlenecks μπορούν να προκαλέσουν καταστροφικές συνέπειες σε ένα server δικτύου. Ακόμη και στη λιγότερο αργή απόκριση μιας εφαρμογής, μπορείτε να δείτε από ένα μικρό τεχνικό προβληματάκι μέχρι και μια ανεξήγητη απώλεια δεδομένων. Φυσικά, το πρώτο ερώτημα που άρεται είναι το εξής: Τι προκαλεί την εμφάνιση ενός bottleneck δίσκου;

Ο πιο απλός ορισμός για το bottleneck δίσκου είναι ότι τα δεδομένα που χρειάζεστε να γράψετε στο δίσκο καταβάλλουν τη λειτουργία του υποσυστήματός του. Αν το drive σας δεν μπορεί να αντιδράσει αρκετά γρήγορα, τότε ο προσαρμογέας SCSI (Small Computer System Interface) είναι πολύ αργός, το καλώδιο έχει βλάβη ή ο ίδιος ο server καταπονείται, οπότε είναι σίγουρο ότι βιώνετε ένα bottleneck δίσκου.

---

## Ανιχνεύοντας τα bottlenecks δίσκου

Τα bottlenecks δίσκου, δυστυχώς, είναι πολύ δύσκολο να ανιχνευθούν, επειδή μπορούν να ‘μεταμφιεστούν’ σε πολλά άλλα είδη προβλημάτων. Σας προτείνουμε λοιπόν δύο εργαλεία ώστε να μπορέσετε κατά ένα μέρος να αρχίσετε να υποψιάζεστε ένα bottleneck δίσκου: το **Disk Defragmenter** και το **System Monitor**.

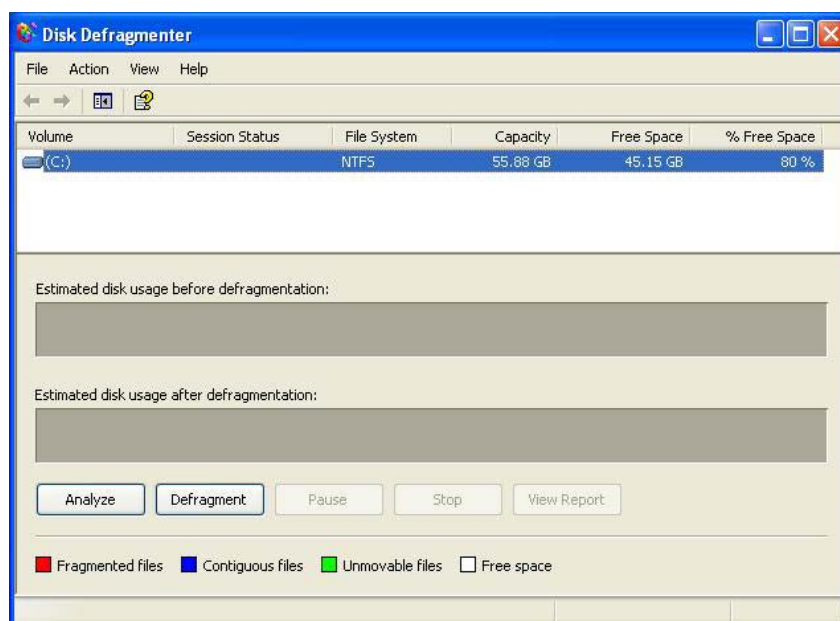


## Disk Defragmenter

Παρακολουθήστε το φωτάκι του drive. Αν αυτό παραμένει αναμμένο για σχεδόν όλη την ώρα, τότε ξέρετε ότι το σύστημα προσβαίνει στο σκληρό δίσκο πολλές φορές για κάποιο λόγο. Το φωτάκι δεν σας λέει γιατί, αλλά το μόνο που σας λέει είναι το πόσο. Δεν είναι φυσιολογικό να παραμένει το φωτάκι αναμμένο για πολύ ώρα, ανεξάρτητα από το πόσο φορτωμένος είναι ο server τη συγκεκριμένη στιγμή. Το φωτάκι θα πρέπει είτε να παραμένει κλειστό για αρκετά μεγάλο διάστημα παρουσιάζοντας μικρά επίπεδα δραστηριότητας (ιδανική περίπτωση) είτε να ανάβει και να σβήνει σε συχνά διαστήματα (πιο συνηθισμένη περίπτωση).

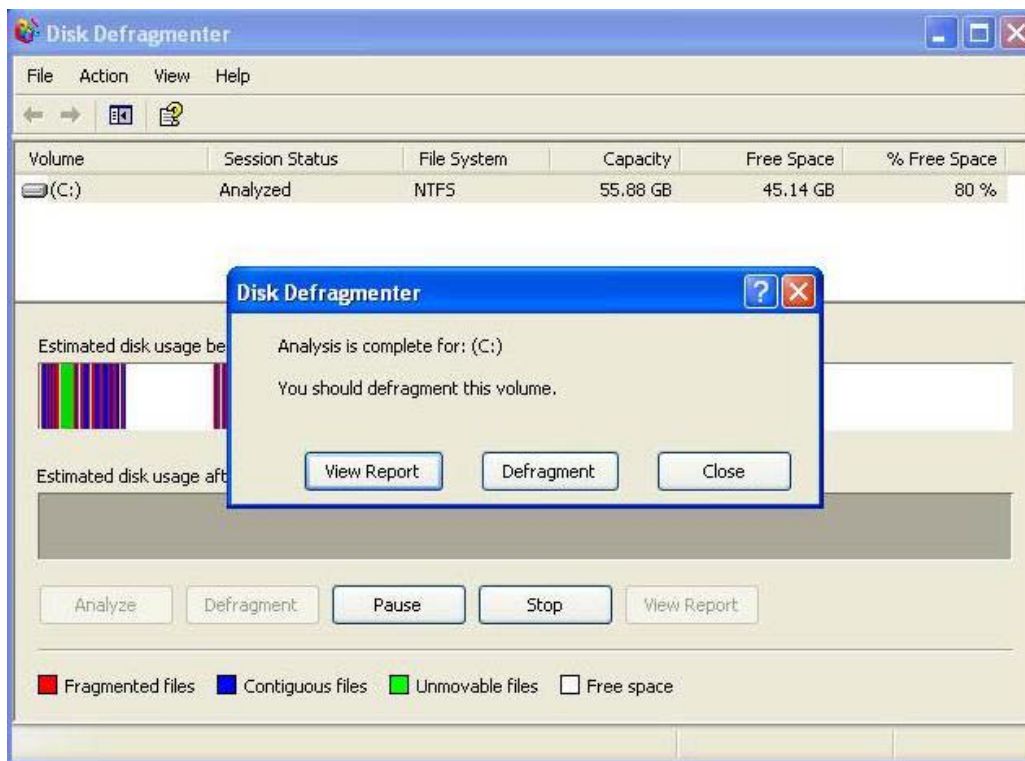
Το πρώτο πράγμα που πρέπει να κάνετε είναι να αποκλείσετε την περίπτωση bottleneck επεξεργαστή και μνήμης, γιατί και οι δύο πηγές προβλημάτων μπορούν να επηρεάσουν τη δραστηριότητα του drive.

Αφού προσδιορίσετε ότι το πρόβλημα σχετίζεται με το σκληρό δίσκο, μπορείτε να χρησιμοποιήσετε τον **Disk Defragmenter** για να διαπιστώσετε αν το πρόβλημα έχει να κάνει με τον κατακερματισμό των αρχείων σας. Για να ανοίξετε το εργαλείο αυτό, κάντε κλικ στο *Start* → *Programs* → *Accessories* → *SystemTools* → *DiskDefragmenter*. Ο κατακερματισμός του δίσκου είναι ένα πολύ σοβαρό πρόβλημα για πολλούς λόγους. Ένα αρχείο που είναι τεμαχισμένο σε πολλές τοποθεσίες προκαλεί πιο συχνή μετακίνηση της κεφαλής του drive. Η κίνηση της κεφαλής επιφέρει επιπλέον κόστος στην απόδοση, επειδή οι κεφαλές των drives κινούνται σχετικά αργά εν συγκρίσει με τα υπόλοιπα μέρη του υπολογιστή. Είναι πολύ γρηγορότερη η πρόσβαση στα δεδομένα, αν ολόκληρο το αρχείο βρίσκεται τοποθετημένο σε μια θέση στο δίσκο, επειδή η κεφαλή του drive δεν χρειάζεται να κινηθεί.

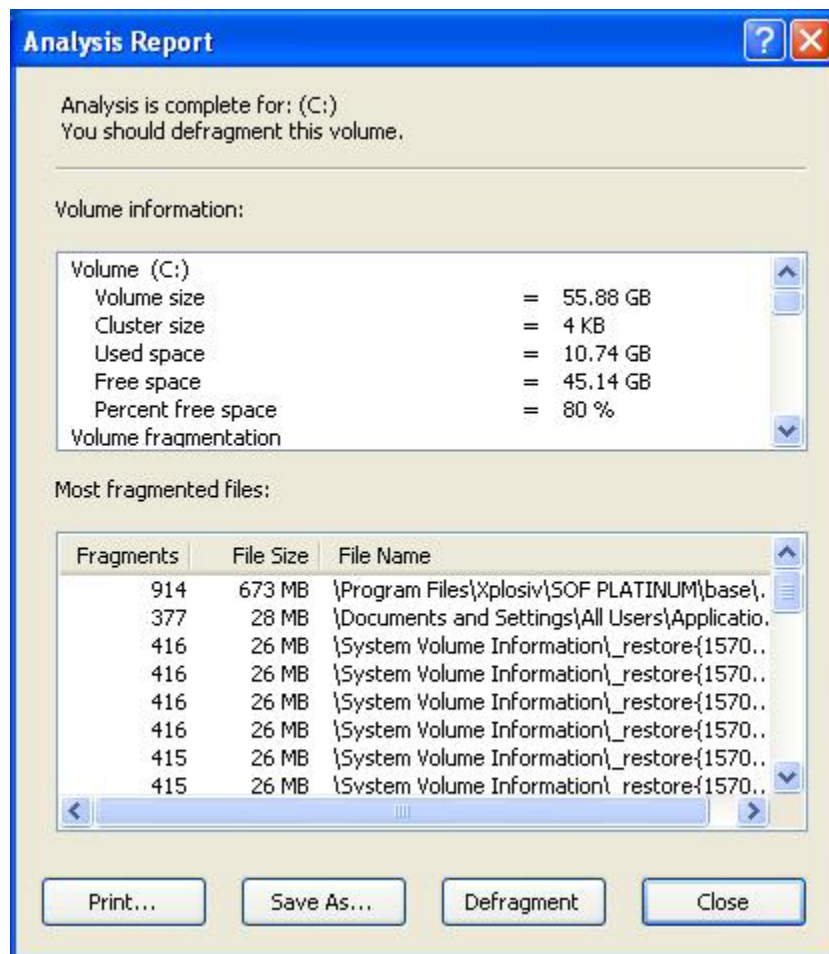


Ο κατακερματισμός του δίσκου δημιουργεί διάφορα άλλα προβλήματα. Για παράδειγμα, αν ο δίσκος σας έχει πολλά μικρά ανοιχτά μέρη στα οποία μπορούν να γραφτούν δεδομένα, αλλά καθόλου μεγάλα μέρη, το λειτουργικό σύστημα αναγκάζεται να διενεργεί εγγραφές σε περισσότερα από ένα μέρη, προκειμένου να αποθηκεύσει μεγάλα αρχεία. Έτσι, κατά τη διάρκεια μιας λειτουργίας ανάγνωσης, η κεφαλή κινείται σε περισσότερα μέρη, καταναλώνοντας περισσότερο χρόνο.

Για να ελέγξετε τον κατακερματισμό του δίσκου σας, στην κονσόλα του Disk Defragmenter κάντε κλικ στο *Analyze*. Μετά από κάποιο ποσό χρόνου, ο disk Defragmenter εμφανίζει ένα παράθυρο διαλόγου, όμοιο με αυτόν που φαίνεται στην παρακάτω εικόνα. Το λειτουργικό σύστημα θα σας πληροφορήσει αν ο δίσκος σας χρειάζεται defragmentation ή όχι. Παρόλ' αυτά, το λειτουργικό σύστημα μπορεί να κάνει λάθος, ενώ ένα αρχείο να είναι πολύ κατακερματισμένο.

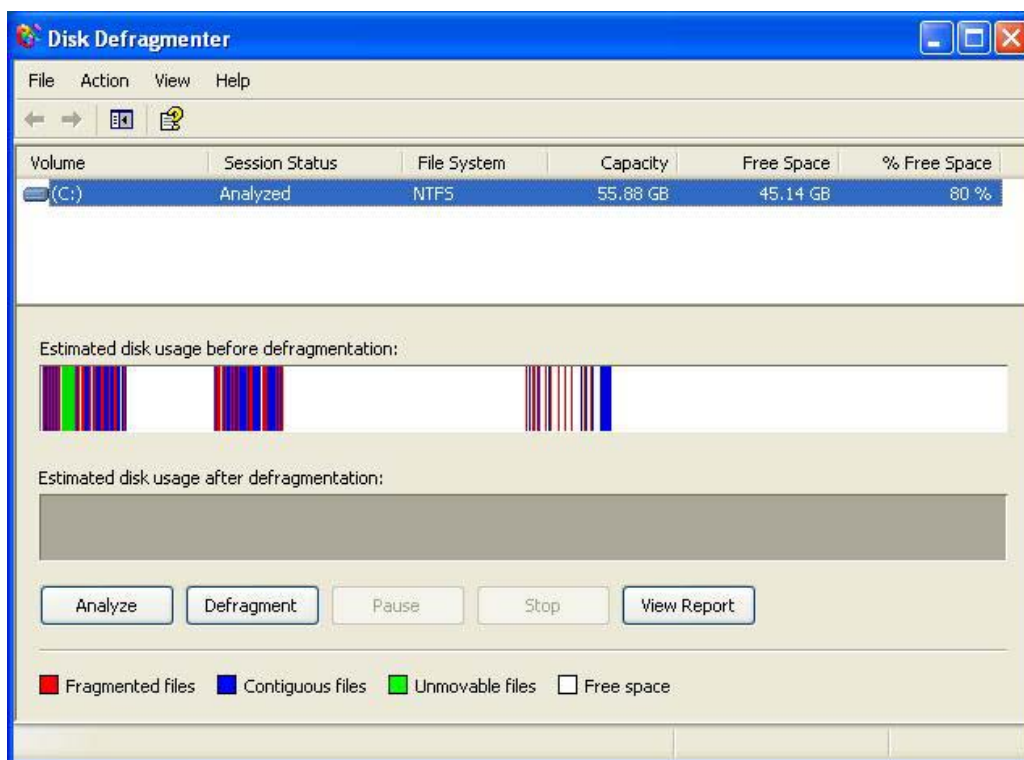


Κάντε κλικ στο *View Report* για να δείτε το παρακάτω παράθυρο αναφοράς της ανάλυσης. Αυτή η απεικόνιση έχει δύο παράθυρα στατιστικών. Το πάνω παράθυρο σας παρέχει γενικά στατιστικά στοιχεία σχετικά με το drive. Εκεί μπορείτε να δείτε πληροφορία σχετικά με το ποσό του ελεύθερου χώρου στο drive και το πραγματικό ποσό του κατακερματισμού. Παρόλο που αυτό το παράθυρο παρέχει σημαντικές πληροφορίες, δεν σας λέει τίποτα για την κατάσταση του δίσκου σας. Αν έχετε έναν πολύ μεγάλο δίσκο, το ποσό του ολικού κατακερματισμού μπορεί να είναι μικρό, αλλά ο κατακερματισμός κάποιων ξεχωριστών, κρίσιμων αρχείων μπορεί να είναι πολύ υψηλός. Σε αυτήν την περίπτωση, ο δίσκος σας χρειάζεται defragmentation.



Το πιο σημαντικό παράθυρο είναι το *Most Fragmented Files*. Περιέχει μια λίστα αρχείων, τα οποία είναι τα πιο κατακερματισμένα στο δίσκο σας. Αν κάνετε κλικ στο File Size, τα αρχεία θα ταξινομηθούν από το πιο κατακερματισμένο στο λιγότερο κατακερματισμένο. Αν δείτε ότι ένα κρίσιμο για σας αρχείο είναι πολύ κατακερματισμένο, κάντε defragmentation ακόμη κι αν το λειτουργικό δεν σας το προτείνει.

Κλείνοντας το διάλογο Analysis Report, εμφανίζεται ο disk Defragmenter, όπως φαίνεται στην εικόνα που ακολουθεί. Το παράθυρο *Analysis display* απεικονίζει με γραφικό τρόπο την κατάσταση του σκληρού σας δίσκου. Οι κόκκινες περιοχές είναι τα κατακερματισμένα αρχεία. Η ύπαρξη πολλών τέτοιων κόκκινων περιοχών δηλώνει ότι υπάρχει πρόβλημα κατακερματισμού.



## System Monitor

Με τους σημερινούς δίσκους υψηλής ταχύτητας, ο βαθμός του throughput του δίσκου είναι σπάνια η αιτία του bottleneck. Ωστόσο, η πρόσβαση στη μνήμη είναι πολύ ταχύτερη από την πρόσβαση στο δίσκο. Έτσι, αν ο server έχει να κάνει πολλές αναγνώσεις και εγγραφές στο δίσκο, η συνολική απόδοση του server παρουσιάζει μείωση. Προκειμένου να μειώσετε το ποσό των λειτουργιών I/O του δίσκου, θέλετε ο server σας να διαχειρίζεται τη μνήμη πολύ αποδοτικά και να κάνει paging στο δίσκο μόνο όταν είναι απαραίτητο.

Οι μετρητές που θα σας βοηθήσουν να ανιχνεύσετε ένα bottleneck δίσκου στο System Monitor του server σας, είναι οι εξής:

Θέμα	Μετρητές	Λεπτομέρειες
<u>Συνολική απόδοση του drive</u>	<i>Physical Disk\%Disk Time</i> σε συνύπαρξη με τους <i>Processor\%Processor Time</i> <i>Network Interface\Bytes Total/sec</i>	Αν η τιμή %Disk Time είναι υψηλή και οι τιμές του Network Interface δεν είναι υψηλές, τότε οι σκληροί δίσκοι του συστήματος μπορεί να προκαλούν bottleneck.

Disk I/O

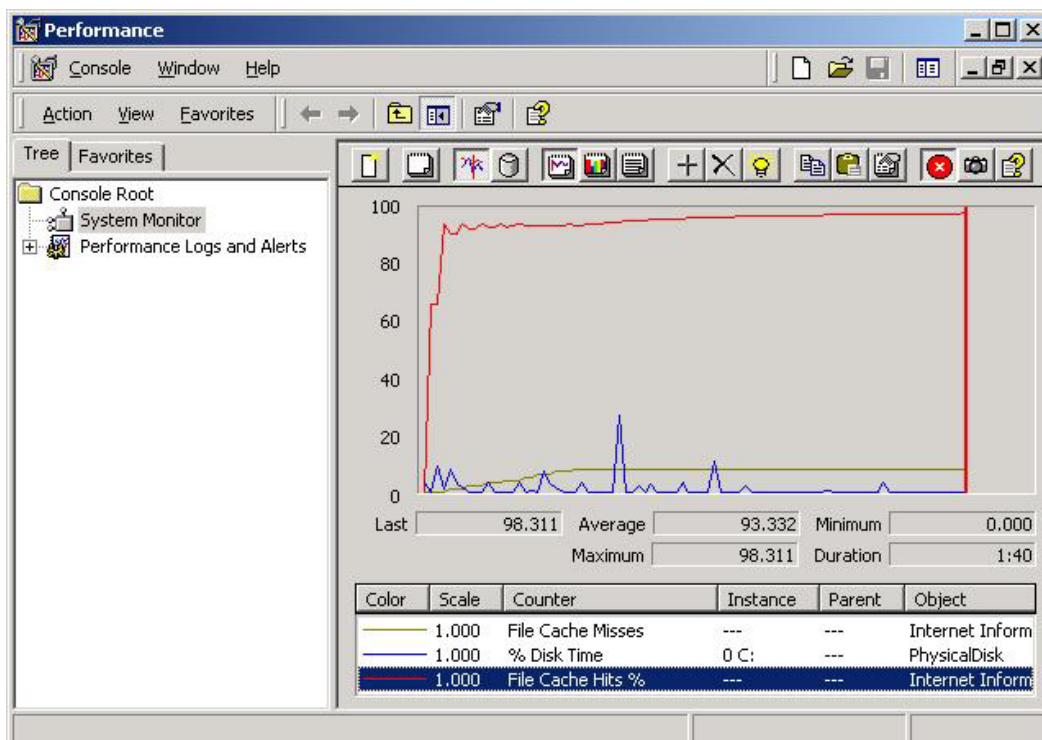
- PhysicalDisk\DiskWrites/sec*
- PhysicalDisk\Disk Reads/sec*
- PhysicalDisk\Avg. Disk Write Queue Length*
- PhysicalDisk\Avg. Disk Read Queue Length*
- Physical Disk\Current Disk Queue Length*

Ο αριθμός των αναγνώσεων και των εγγραφών ανά δευτερόλεπτο σας λένε πόση δραστηριότητα I/O δίσκου υπάρχει. Οι ουρές εγγραφής και ανάγνωσης σας λένε πόσες αιτήσεις εγγραφής και ανάγνωσης περιμένουν για επεξεργασία. Γενικά, θέλετε να υπάρχουν πολύ λίγες αιτήσεις στις ουρές.

## Test #7

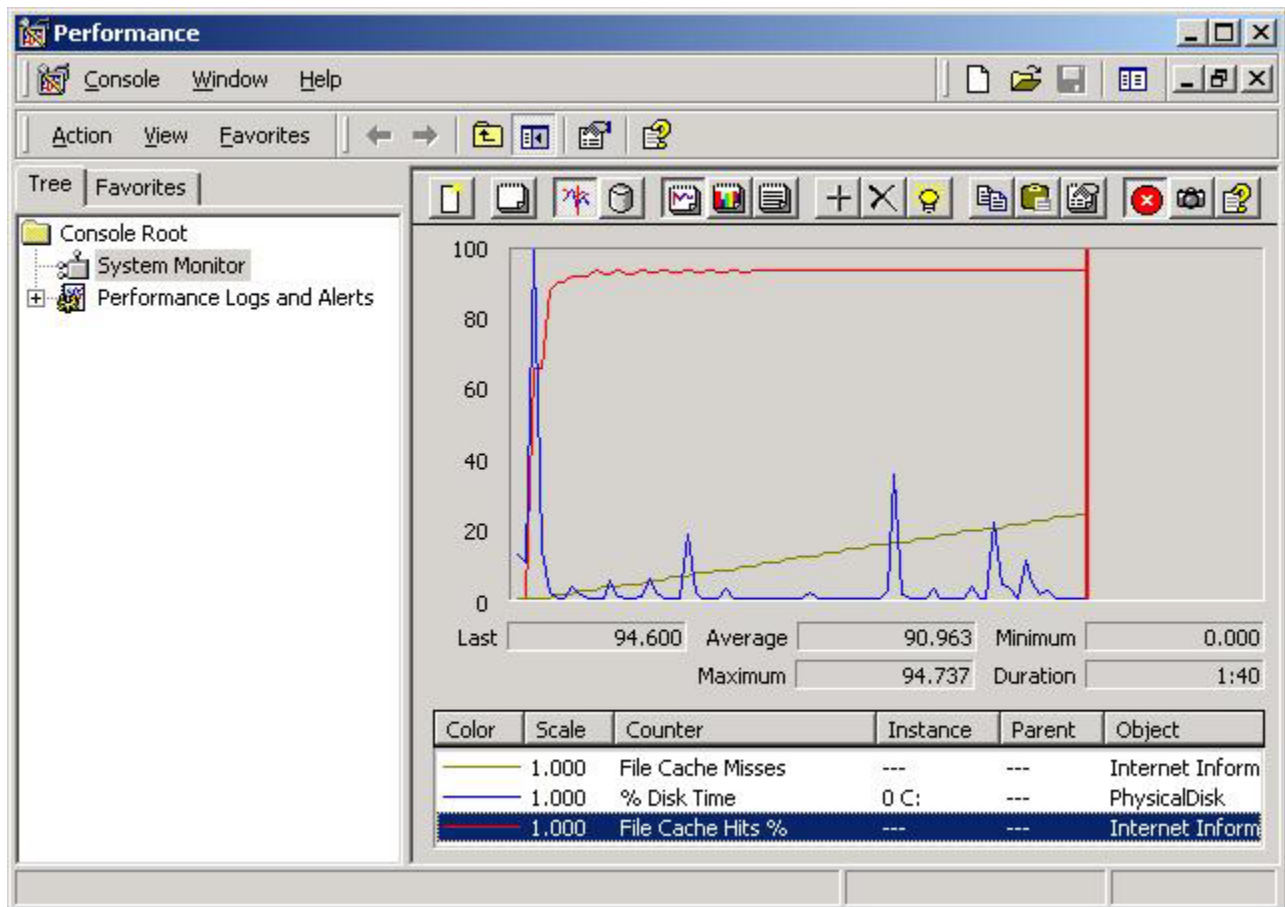
Θα διενεργήσουμε ένα πείραμα για να δείξουμε τι επίδραση έχει στη χρησιμοποίηση του δίσκου η αλλαγή στο χρόνο ζωής (TTL) των αντικειμένων της cache, έτσι ώστε να γίνει κατανοητή η σχέση μεταξύ δίσκου και μνήμης.

Το πείραμα αυτό είναι ίδιο με το test #5 που περιγράψαμε νωρίτερα, με τη διαφορά ότι οι 9 στατικές σελίδες που χρησιμοποιήσαμε είναι μεγέθους 100 KB. Αρχικά εκτελέσαμε το πείραμα χρησιμοποιώντας την default τιμή για την ObjectCacheTTL (30 δευτερόλεπτα). Το System Monitor του server έδειξε τα ακόλουθα αποτελέσματα:



Τα σχόλιά μας για τους μετρητές File Cache Hits % και File Cache Misses είναι τα ίδια με το test #5. Προσέξτε ότι ο μετρητής %Disk Time παίρνει κάποιες τιμές μέχρι να ανακτηθούν και οι 9 σελίδες από το δίσκο και να τοποθετηθούν στη μνήμη και από εκεί κι έπειτα η χρησιμοποίηση μειώνεται.

Διενεργούμε τώρα το ίδιο πείραμα, μεταβάλλοντας την τιμή της ObjectCacheTTL στο 2 (δευτερόλεπτα). Τα αποτελέσματα του System Monitor ήταν τα εξής:



Τα σχόλιά μας για τους μετρητές File Cache Hits % και File Cache Misses είναι τα ίδια με το test #5. Προσέξτε ότι ο δίσκος συνεχίζει να δουλεύει καθώς η cache αποτυγχάνει συνεχώς και ανακτώνται συνεχώς οι σελίδες από αυτόν.

### **Συμπέρασμα:**

Μειώνοντας την τιμή της μεταβλητής ObjectCacheTTL αυξάνονται οι αποτυχίες στην cache, με αποτέλεσμα να έχουμε περισσότερες αναγνώσεις από το δίσκο, πράγμα που με τη σειρά του οδηγεί σε μείωση της απόδοσης.

---

## Τεχνολογίες disk-formatting

Θα παρουσιάσουμε σύντομα τέσσερις διαφορετικές τεχνικές ρύθμισης και διαμόρφωσης δίσκου που βοηθούν στη βελτίωση της απόδοσης του server.

### **Disk Mirroring**

Το disk mirroring είναι ένας από τους πιο εύκολους τρόπους για τη δημιουργία ανοχής σφαλμάτων μέσα στο server. Η αρχή τους είναι απλή. Το μόνο που χρειάζεστε είναι να προσθέσετε δύο σκληρούς δίσκους με ακριβώς τα ίδια χαρακτηριστικά και μέγεθος στο σύστημά σας. Όταν επανεκκινήσετε το server, το Disk Management σας επιτρέπει να τους κάνετε mirroring. Αυτό θα έχει ως αποτέλεσμα και οι δύο δίσκοι να έχουν ακριβώς την ίδια πληροφορία. Αν ένας δίσκος αποτύχει, αναλαμβάνει ο άλλος αμέσως, πράγμα που σημαίνει ότι ο χρήστης δεν καταλαβαίνει τη διαφορά. Το μόνο που χρειάζεστε για να διορθώσετε το πρόβλημα είναι να εγκαταστήσετε ένα νέο drive με το ίδιο μέγεθος και ίδια χαρακτηριστικά και να το ξανακάνετε mirroring με το υπάρχον drive. Το μειονέκτημα με αυτήν την τεχνική είναι ότι χάνεται πάντα η μισή χωρητικότητα των δίσκων λόγω του mirroring.

### **Disk Duplexing**

Το disk duplexing είναι σαν το disk mirroring με τη διαφορά ότι εδώ χρησιμοποιούνται δύο host adapters, δηλαδή δύο ξεχωριστά κανάλια δεδομένων στο ίδιο μηχάνημα. Αυτό παρέχει μια ενίσχυση στην ανοχή σφαλμάτων (επειδή παρέχεται πλέον ανοχή και σε σφάλματα του host adapter). Έχει όμως ως μειονέκτημα το υψηλότερο κόστος συν το γεγονός ότι η μισή χωρητικότητα χάνεται εξ αιτίας του mirroring.

### **Drive Striping**

Σε αυτήν την τεχνική μπορούν να συνδυαστούν δύο δίσκοι ίδιας χωρητικότητας σε ένα volume. Τα Windows 2000 μπορούν να εναλλάσσουν τα δεδομένα μεταξύ των δύο δίσκων σε λωρίδες. Το πλεονέκτημα είναι ότι δίνεται ενίσχυση στην απόδοση, χρησιμοποιείται ολόκληρη η χωρητικότητα των δίσκων, αλλά δεν υπάρχει καθόλου ανοχή σφαλμάτων, καθώς αν αποτύχει ο ένας δίσκος, τότε χάνονται και τα αντίστοιχα δεδομένα στον άλλον.

### **RAID-5**

Είναι η καλύτερη τεχνική και για την απόδοση και για την ανοχή σφαλμάτων για μεγάλα δίκτυα με πολύ μεγάλη χωρητικότητα σε δίσκο. Η διάταξη RAID-5 απαιτεί την ύπαρξη τριών τουλάχιστον σκληρών δίσκων.

Η εγκατάσταση είναι παρόμοια με την τεχνική drive striping. Ωστόσο, αντί να εναλλάσσονται μόνο τα δεδομένα μεταξύ των δύο δίσκων, χρησιμοποιείται επιπλέον κι ένα τρίτο drive που χρησιμοποιεί parity bit. Τα δεδομένα εγγράφονται κυκλικά, οπότε και τα τρία drives περιέχουν parity bits. Χρησιμοποιώντας αυτήν τη μεθοδολογία, αν ένας δίσκος αποτύχει, οι άλλοι δύο επιτρέπουν στο σύστημα να ανακύψει χρησιμοποιώντας τα parity bits. Η ανάκυψη γίνεται αυτόματα (δηλαδή δεν χρειάζεται remirroring).

Η τεχνική RAID-5 παρουσιάζει πολύ καλύτερη απόδοση από τις άλλες τεχνικές, καθώς και πολύ μεγάλη ανοχή σε σφάλματα. Αυτός είναι και ο λόγος που οι περισσότεροι διαχειριστές δικτύων χρησιμοποιούν τις διατάξεις αυτές.

---

## Συμβουλές

Μπορείτε να σκεφτείτε τον IIS ως έναν file server ειδικού σκοπου. Αιτήσεις για στατικό και δυναμικό περιεχόμενο έχουν αρχικά ως αποτέλεσμα μια πρόσβαση στο δίσκο. Οι πόροι που προσπελούνται συχνά αποθηκεύονται στην cache από τον Cache Manager. Υπάρχουν δύο προσεγγίσεις για τη βελτίωση της απόδοσης του IIS server όσον αφορά στο υποσύστημα του δίσκου. Η πρώτη προσέγγιση είναι η μείωση του αριθμού των προσβάσεων στο δίσκο, αποθηκεύοντας στην κύρια μνήμη τα αρχεία που προσπελούνται πολύ συχνά. Η δεύτερη προσέγγιση είναι η αύξηση του throughput του υποσυστήματος του δίσκου.

Συλλέγοντας στατιστικά στοιχεία σε σχέση με τις σελίδες που ζητώνται από τους χρήστες, θα δείτε ότι υπάρχει ένα πολύ μεγάλο ποσοστό αιτήσεων που αφορούν ένα μικρό μόνο ποσοστό αρχείων. Μέσω των log files μπορείτε να προσδιορίσετε ποια είναι τα πιο δημοφιλή αρχεία και ποιο είναι το σύνολό τους σε bytes. Αυτό θα ήταν ένας καλός δείκτης σχετικά με το ποιο θα ήταν το ιδανικό ποσό μνήμης που θα πρέπει να έχετε για το server σας.

Για πολύ μεγάλα sites με τρομερά ποσά περιεχομένου, είναι πολύ ουτοπικό να περιμένετε ότι ένα μεγάλο τμήμα των πιο δημοφιλών αρχείων πρέπει να βρίσκεται πάντα στη μνήμη. Σε αυτές τις περιπτώσεις, η λύση για τη βελτίωση της απόδοσης του υποσυστήματος του δίσκου είναι η δεύτερη προσέγγιση, δηλαδή η αύξηση του throughput του δίσκου.

Αν κατά την παρακολούθηση της χρήσης των φυσικών σας δίσκων παρατηρήσετε ότι οι δίσκοι στους οποίους είναι αποθηκευμένο το περιεχόμενο του server έχουν μεγάλη χρησιμοποίηση, προσπαθήστε να διανέμετε το φορτίο, προκειμένου να μειώσετε τη χρησιμοποίηση. Αν υπάρχουν άλλες εφαρμογές που παράγουν αυτό το φορτίο στον ίδιο



δίσκο, τότε να μεταφέρετε αυτές τις εφαρμογές ή το περιεχόμενο σε άλλον δίσκο. Με αυτόν τον τρόπο θα μειώσετε τα page faults του δίσκου στον οποίο περιέχεται το περιεχόμενο του server σας. Επίσης, φροντίστε να διατηρείτε τα log files σας σε διαφορετικό δίσκο από το περιεχόμενο του server.

Το επόμενο βήμα για την ενίσχυση του throughput του υποσυστήματος του δίσκου, είναι η χρήση της διάταξης RAID-5. Το RAID-5, όπως έχουμε ήδη πει, παρέχει υψηλή απόδοση και καλή ανοχή σφαλμάτων σε μικρό κόστος.

**Ό,τι και να κάνετε, να έχετε πάντα στο μυαλό ότι τη χρησιμοποίηση του δίσκου την επηρεάζουν επίσης η μνήμη και ο επεξεργαστής. Φροντίστε να αποκλείσετε την ύπαρξη ενός bottleneck μνήμης ή επεξεργαστή, πριν προχωρήσετε στη μελέτη για bottleneck δίσκου!**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**  
**ΠΑΠΑΔΟΠΟΥΛΟΣ ΣΤΑΥΡΟΣ**



# ΒΙΒΛΙΟΓΡΑΦΙΑ

Οι πηγές που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας είναι οι παρακάτω:

## **BIBΛΙΑ**

*Microsoft Windows 2000 Performance Tuning Technical Reference*, John Paul Muler – Irfan Shaudhry

*Web Performance Tuning*, Patrick Killelea, O'REILLY 1998

## **ΑΡΘΡΑ**

*Managing the Performance of Microsoft's Internet Information Server*, Odysseas Pentakalos

## **ΠΗΓΕΣ INTERNET**

*IIS – The Basics of Performance Tuning*

<http://www.winnetmag.com/Web/Article/ArticleID/16144/16144.html>

*Optimizing Internet Information Services 5.0*

<http://support.microsoft.com/default.aspx?scid=kb;EN-US;305313>

*All About Cookies*

<http://www.paulding.net/FAQ/cookies.html>

*Performance Tuning and Optimization*

<http://labmice.techtarget.com/BackOffice/IIS5/performance.htm>

*Monitoring and Optimizing Internet Information Server*

[http://library.mobrien.com/Manuals/MPRM\\_Group/IIS1.htm](http://library.mobrien.com/Manuals/MPRM_Group/IIS1.htm)

*Monitoring your IIS Web Servers*

<http://www.zdnet.com.au/insight/0,39023731,20265586,00.htm>

*Optimizing and Tweaking Internet Information Services*

<http://techrepublic.com.com/5100-6268-1059916.html>

*Advanced Basics – IIS Threading and State Management*

<http://www.microsoft.com/mind/0299/basics/basics0299.asp>

*HowStuffWorks – How Web Servers Work*

<http://computer.howstuffworks.com/web-server1.htm>

*IIS 5 Architecture Overview*

<http://www.wwwcoder.com/main/parentid/170/site/1515/68/default.aspx>

*The Art and Science of Web Server Tuning with Internet Information Services 5.0*

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/iis/maintain/optimize/iis5tune.mspix>

*HOW TO: Optimize Web Server Performance in Windows 2000*

<http://support.microsoft.com/default.aspx?scid=KB;EN-US;q308186&>

*IIS 5.0 Resource Guide: Chapter 5 - Monitoring and Tuning Your Server*

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/iis/reskit/iischp5.mspix>

*Shared IIS registry entries*

<http://www.jsiinc.com/SUBA/tip0300/rh0392.htm>