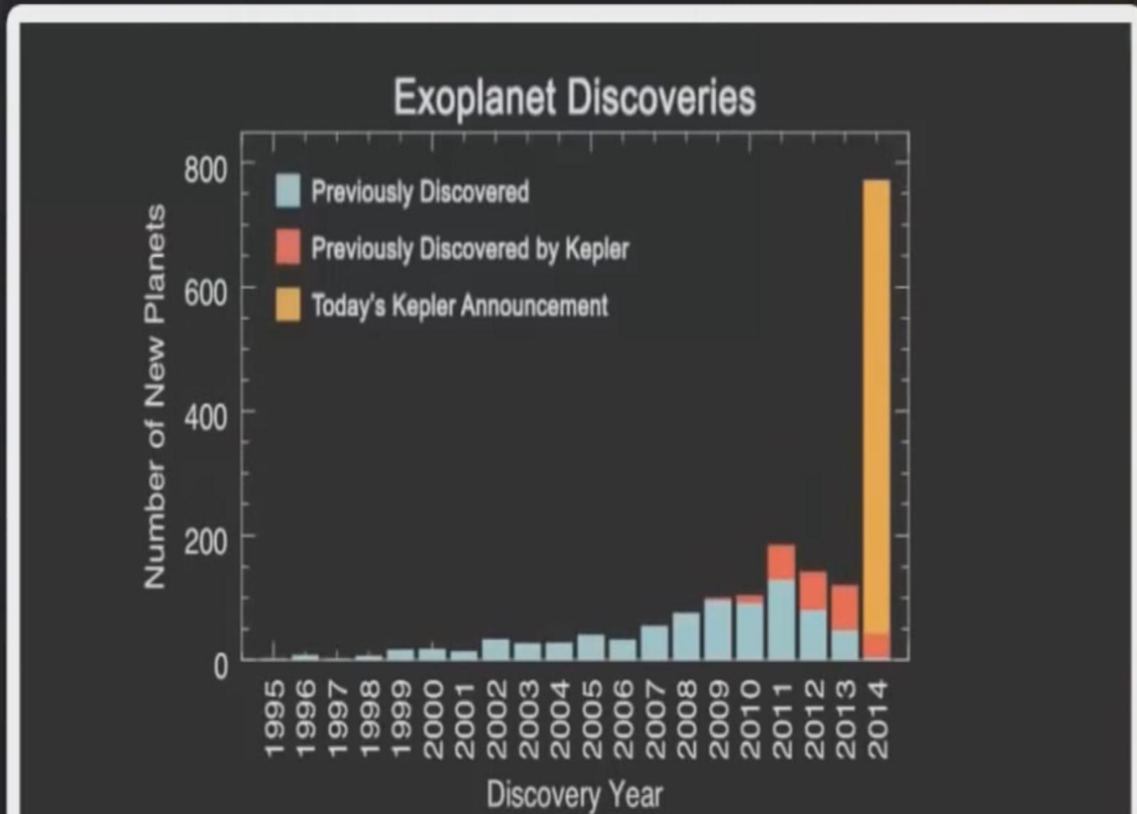
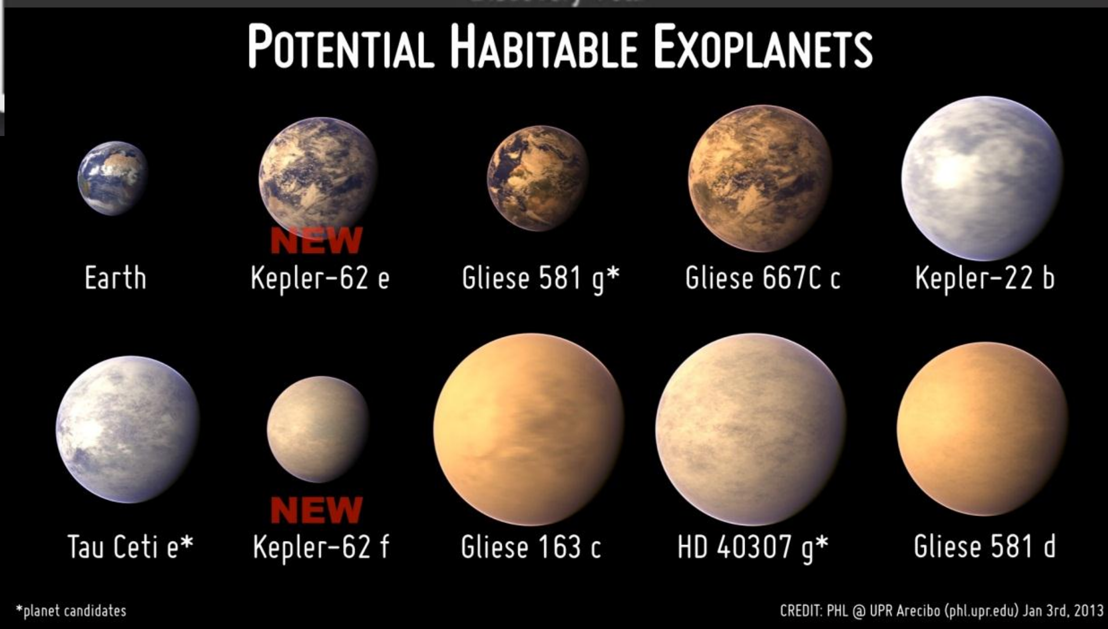




Advanced Data Indexing

(Προηγμένη ευρετηρίαση δεδομένων)

Ροές Δεδομένων (1ο Μέρος)

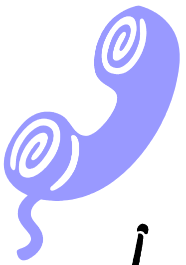
Κίνητρα και Εφαρμογές Διαχείρισης Ροών

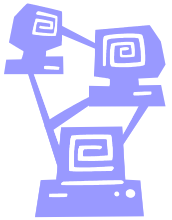
Ροές – Ένας Νέος Κόσμος

- Παραδοσιακά Συστήματα DBMS: τα δεδομένα είναι αποθηκευμένα σε *πεπερασμένα, προσδιορισμένα σύνολα.*
- Ροές: τα δεδομένα είναι κατανεμημένα, έχουν συνεχή ροή, είναι μη φραγμένα, έρχονται γρήγορα, μεταβάλλονται με το χρόνο, έχουν θόρυβο, κλπ.
- Διαχείριση των Ροών: απαραίτητη σε μία μεγάλη ποικιλία σύγχρονων εφαρμογών:
  - Παρακολούθηση Δικτύων, Βελτιστοποίηση Κυκλοφορίας
  - Δίκτυα αισθητήρων (sensors)
  - Ασφάλεια Δικτύων
  - Οικονομικές Εφαρμογές
  - Παρακολούθηση επισκέψεων σελίδων και web logs
  - κλπ.

Ροές Μεγάλου Όγκου Δεδομένων

- Τα δεδομένα πληθαίνουν συνεχώς και γρηγορότερα από την ικανότητά μας να τα αποθηκεύσουμε ή να τα δεικτοδοτήσουμε:
- Στις ΗΠΑ γίνονται 3 δις τηλεφωνικές κλήσεις, 30 δις emails, 1 δις SMS, κάθε μέρα!
- Επιστημονικά Δεδομένα: Οι δορυφόροι της NASA καταγράφουν και στέλνουν δισεκατομμύρια δεδομένα κάθε μέρα.
- Κυκλοφορία Δικτύου: Πάνω από 1 δις πακέτων ανά ώρα σε κάθε router. Κάθε ISP έχει εκατοντάδες routers!





Ροές Μεγάλου Όγκου Δεδομένων

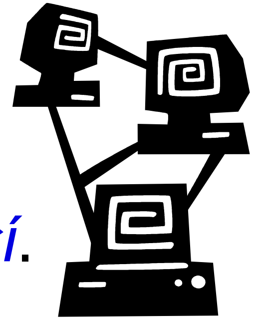
Γιατί πρέπει να αναλύσουμε αυτά τα δεδομένα;

- Για Επιστημονική Έρευνα (έλεγχος περιβαλλοντικών συνθηκών, κλπ.)
- Για Βελτίωση Συστημάτων (εντοπισμός λαθών, αποτυχιών, κακής λειτουργίας, κλπ.)
- Για Επιχειρηματικό Σχεδιασμό (κανόνες αγοράς, προσφορά/ζήτηση, κλπ.)
- κλπ.

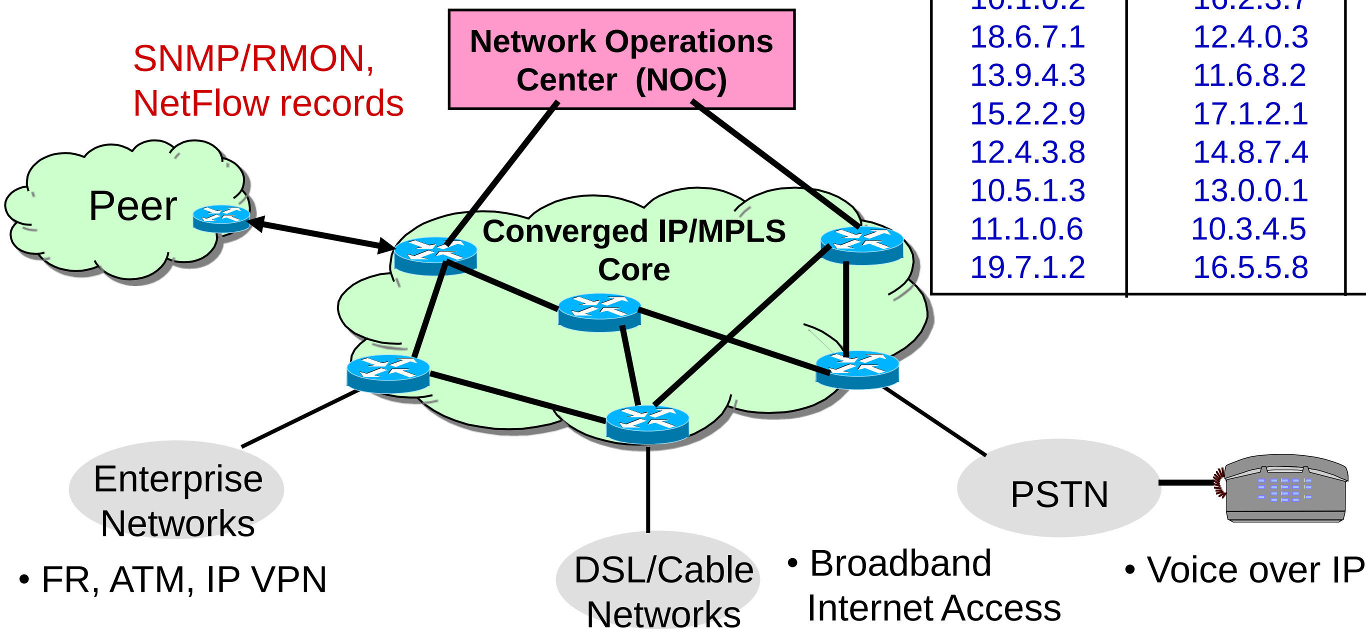
Αλλιώς, θα υπήρχε κάποιος λόγος ακόμα και για την ύπαρξη τους;

Παράδειγμα: Ροή Δεδομένων Δικτύου

- Τα δίκτυα είναι πηγές μεγάλου όγκου δεδομένων: τα μετα-δεδομένα ανά ώρα και ανά IP router είναι της τάξης των gigabytes.
- Το θεμελιώδες πρόβλημα της ανάλυσης της ροής των δεδομένων είναι *ο πολύ μεγάλος όγκος της πληροφορίας για να αποθηκευτεί ή να αναμεταδοθεί*.
- Συνεπώς τα δεδομένα πρέπει να επεξεργάζονται *καθώς αυτά καταφθάνουν*, *διαβάζοντάς τα μόνο μία φορά* και *χρησιμοποιώντας μικρό χώρο* → **μοντέλο των ροών**.
- *Προσεγγιστικές απαντήσεις* σε πολλά ερωτήματα είναι αποδεκτές, αρκεί **να εγγυώνται την ποιότητα του αποτελέσματος**.



Εφαρμογή: Παρακολούθηση Δικτύου (IP)



Source	Destination	Duration	Bytes	Protocol
10.1.0.2	16.2.3.7	12	20K	http
18.6.7.1	12.4.0.3	16	24K	http
13.9.4.3	11.6.8.2	15	20K	http
15.2.2.9	17.1.2.1	19	40K	http
12.4.3.8	14.8.7.4	26	58K	http
10.5.1.3	13.0.0.1	27	100K	ftp
11.1.0.6	10.3.4.5	32	300K	ftp
19.7.1.2	16.5.5.8	18	80K	ftp

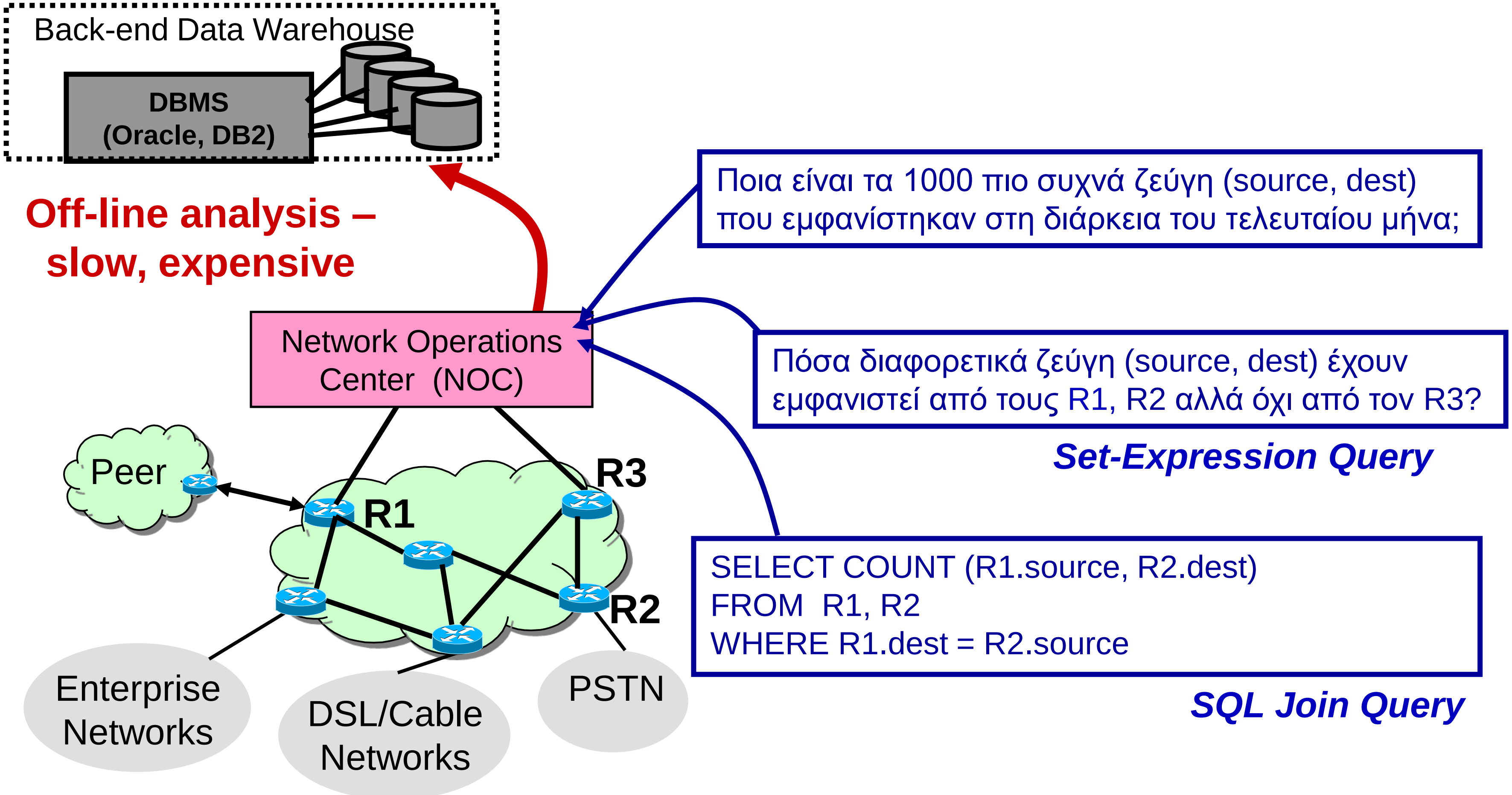
Παράδειγμα ροής IP Session Data

- Καταφθάνουν πραγματικά ροές μεγάλου όγκου δεδομένων με ταχύτατους ρυθμούς.
  - Η AT&T συλλέγει *~1 Terabyte* δεδομένα ροών δικτύου κάθε μέρα.
- Πολλές φορές κατευθύνονται σε μεγάλες αποθήκες δεδομένων για να αναλυθούν εκτός δικτύου.

Ροή δεδομένων σε επίπεδο πακέτων

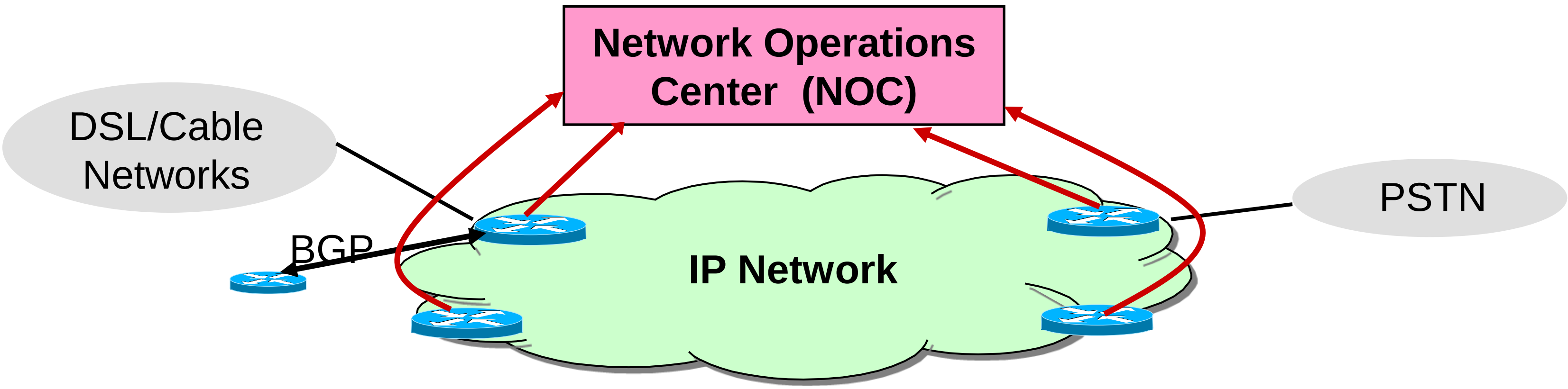
- Έστω ότι:
  - Η ταχύτητα σύνδεσης στο δίκτυο είναι ***2Gb/sec*** και το μέσο μέγεθος πακέτου είναι ***50bytes***.
  - Το πλήθος των πακέτων ανά δευτερόλεπτο που μεταδίδονται είναι ***5 εκατομμύρια***.
  - Ο μέσος χρόνος μετάδοσης είναι: ***0.2 microsec***.
- Αν κρατούσαμε μόνο βασικές πληροφορίες (το **header**) από το κάθε πακέτο: **src/dest IP,time,no. of bytes**,κλπ. δηλαδή περίπου ***10bytes***, τότε:
  - **Ο χώρος που θα χρειαζόμασταν ανά δευτερόλεπτο θα ήταν *50Mb***.
  - **Ανά ημέρα θα ήταν *4.5Tb* ανά σύνδεση (link).**
  - **Και οι ISPs τυπικά έχουν εκατοντάδες συνδέσεις!**
- Το να αναλύσουμε το ***περιεχόμενο των πακέτων*** σε λογικό χώρο και χρόνο, είναι ένα τελείως απίθανο πράγμα!!

Ερωτήματα Παρακολούθησης Δικτύου



- Υπάρχει αυξημένη πολυπλοκότητα για τέτοια ερωτήματα λόγω του *περιορισμένου χώρου και χρόνου*.
- Όμως υπάρχουν λύσεις για τέτοια όπως και για άλλα ερωτήματα.

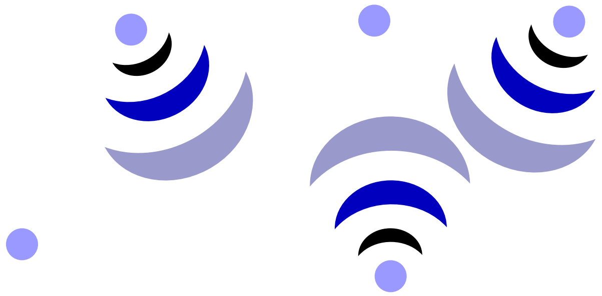
Ανάλυση Ροής Δεδομένων σε πραγματικό χρόνο



- Πρέπει να επεξεργαστούμε τις ροές δικτύων σε *πραγματικό χρόνο* και με *μία μόνο ανάγνωση* των δεδομένων τους.
- Κρίσιμοι λόγοι για ανάλυση: fraud, DoS attacks, SLA violations, και άλλοι.
  - Επίσης για την καταγραφή και την μελέτη της κίνησης στο δίκτυο με σκοπό την βελτίωσή του.
- *Ανταλλάσσεται η ποιότητα του αποτελέσματος σε σχέση με τον διαθέσιμο χώρο/χρόνο ή την επικοινωνία. Θέλουμε:*
  - Γρήγορες αποκρίσεις, μικρός χώρος/χρόνος.
  - Ελαχιστοποίηση της χρήσης των πηγών επικοινωνίας.

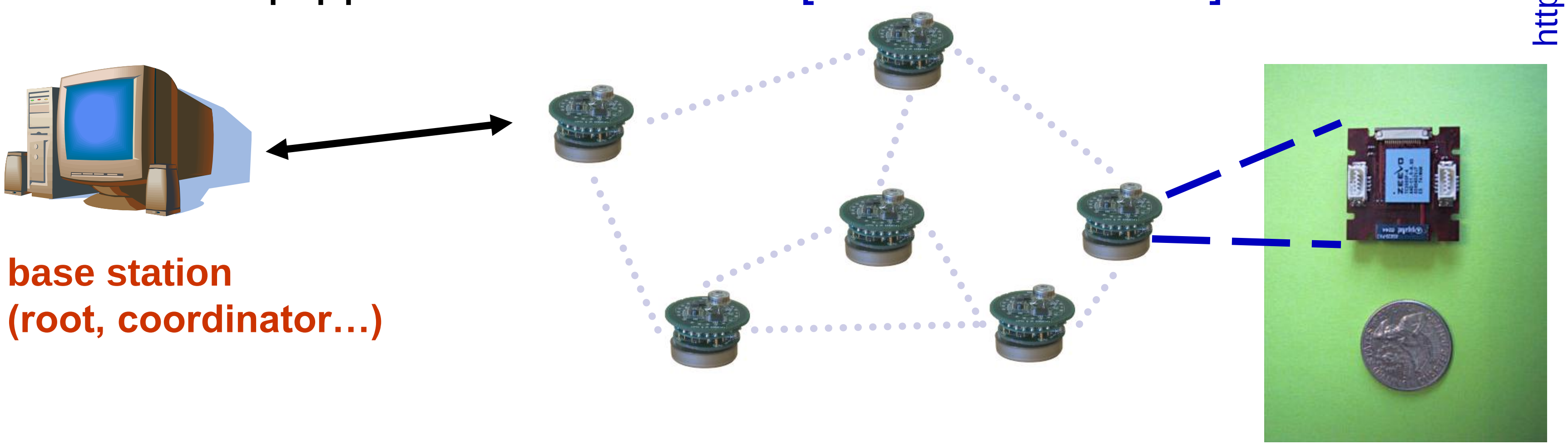
Δίκτυα Αισθητήρων (Sensor Networks)

- Τα ασύρματα δίκτυα αισθητήρων έχουν γίνει απαραίτητα σε εφαρμογές ελέγχου του περιβάλλοντος, σε στρατιωτικές εφαρμογές, κλπ.
- Αρκετοί (100, 10^3, 10^6?) αισθητήρες τοποθετούνται στην περιοχή ενδιαφέροντος.
- Οι αισθητήρες παρατηρούν, καταγράφουν και μεταδίδουν σε μία τοπική ροή δεδομένα:
  - Μετρήσεων φωτός, θερμοκρασίας, πίεσης,…
  - Εντοπισμού σημάτων, κίνησης, ακτινοβολίας,…
  - Καταγραφής ήχων, εικόνας, κίνησης,…



Εφαρμογή σε Δίκτυα Αισθητήρων

- Ανάθεση ενός ερωτήματος σε δίκτυο αισθητήρων μέσω ενός τερματικού σταθμού.
- Οι αισθητήρες έχουν πολλούς περιορισμούς:
  - Περιορισμένη ενέργεια (μπαταρία), μνήμη, επεξεργαστή, εύρος ακτίνας,…
  - Η *επικοινωνία* είναι η μεγαλύτερη πηγή κατανάλωσης της ενέργειάς τους.
  - Η αναμετάδοση ενός απλού bit δεδομένων είναι ισοδύναμη με 800 instructions [Madden et al.'02].



http://www.intel.com/research/exploratory/motes.htm

Μοντέλα Ροών

Γενικό Μοντέλο Ροών

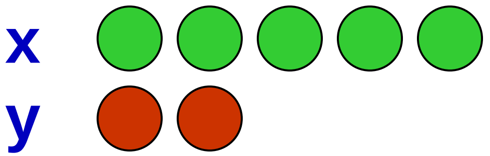
- Μεταδιδόμενο Σήμα: Κάθε μορφή ροής μπορεί να παρασταθεί έχοντας ως βάση έναν μονοδιάστατο πίνακα A[1…N] με αρχικές τιμές A[i] όλες μηδέν.
  - Μπορούν να χρησιμοποιηθούν και πολυδιάστατοι πίνακες.
- Το μεταδιδόμενο σήμα μπορεί να παρασταθεί από μία *ροή ενημερώσεων.*
  - Η j-στή ενημέρωση είναι το ζεύγος <x, c[j]> που σημαίνει:
    - A[x] := A[x] + c[j]   (το c[j] μπορεί να είναι >0, <0)
- Στόχος: Ο υπολογισμός συναρτήσεων f στον A[] ώστε:
  - Να γίνεται σε μικρό χώρο.
  - Να γίνεται γρήγορη επεξεργασία των ενημερώσεων.
  - Να γίνεται γρήγορος υπολογισμός της f.
  - … (μπορεί να υπάρχουν και έξτρα απαιτήσεις).
- Η πολυπλοκότητα αυξάνεται όταν έχουμε ογκώδεις ροές με μεγάλο μήκος (N).

Παράδειγμα Ροής Πακέτων Δικτύου IP

- Πλήθος πακέτων (packets) που στέλνονται από μία διεύθυνση IP κατά τη διάρκεια μίας ημέρας:
  - Τα δεδομένα σε $\mathbf{2}^{32}$ μεγέθους μονοδιάστατο πίνακα. Οι ενημερώσεις γίνονται αυξητικά.
- Πλήθος ροών μεταξύ μίας διεύθυνσης IP (πηγής) και μίας άλλης διεύθυνσης IP (προορισμού) κατά τη διάρκεια μίας ημέρας:
  - Τα δεδομένα σε $\mathbf{2}^{64}$ μεγέθους δισδιάστατο πίνακα, συγκεντρωμένα σε πακέτα μέσα στις ροές. Οι ενημερώσεις γίνονται αυξητικά.
- Πλήθος ενεργών ροών ανά διεύθυνση IP:
  - Τα δεδομένα σε $\mathbf{2}^{32}$ μεγέθους μονοδιάστατο πίνακα. Οι ενημερώσεις γίνονται και με αύξηση και με μείωση.

Μοντέλα Ροών: Βασικές Περιπτώσεις

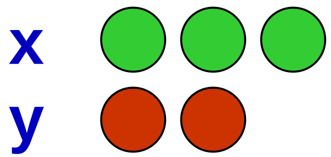
- Μοντέλο Χρονοσειρών (Time Series Model):
  - Η x-στή ενημέρωση **αντικαθιστά** την τιμή A[x] (A[x]:=c[x])
  - Παράδειγμα: Μετοχές.
- Μοντέλο Ταμειακής Μηχανής (Cash-Register Model):
  - Ροές όπου οι ενημερώσεις γίνονται μόνο **αυξητικά**: (A[x]:=A[x]+c[j], c[x]>0 πάντα)
  - Τυπική περίπτωση είναι για c[x]=1 και τότε η ροή εκφράζει ένα πολλαπλό σύνολο (multi-set) αντικειμένων.
  - Παράδειγμα, η ροή: <x,3>, <y,2>, <x,2> εκφράζει την άφιξη 3 αντικειμένων τύπου x, 2 αντικειμένων τύπου y, και μετά πάλι 2 αντικειμένων τύπου x.



  - Παραδείγματα τέτοιων ροών: ροή πακέτων σε ένα δίκτυο, ροή χρήσης ενέργειας.

Μοντέλα Ροών: Βασικές Περιπτώσεις

- Μοντέλο Μύλου (Turnstile Model):
  - Οι ενημερώσεις γίνονται και **αυξητικά** και **μειωτικά**. (A[x]:=A[x]+c[j], με c[x]>0 ή c[x]<0)
  - Είναι το πιο γενικό μοντέλο ροών.
- Προσομοιώνει το μοντέλο αφίξεων και αναχωρήσεων:
  - Παράδειγμα, η ροή: <x,3>,<y,2>,<x,-2> θα έχει την τελική κατάσταση: <x,1>,<y,2>.



  - Παραδείγματα τέτοιων ροών: έλεγχων επιβατών που φτάνουν και φεύγουν από έναν σταθμό, έλεγχος διαφοράς τιμών μεταξύ δύο κατανομών, κλπ.
- Η δυσκολία κάθε προβλήματος εξαρτάται από το μοντέλο πάνω στο οποίο εφαρμόζεται:
  - Παράδειγμα, η εύρεση MIN/MAX έχει διαφορετική δυσκολία στο μοντέλο Time-Series από ότι στο μοντέλο Turnstile.

Ένα απλό Πρόβλημα ροών

- Ο Παύλος αντιμεταθέτει τους αριθμούς **1,2,…,N** (μεγάλο πλήθος) και τους δείχνει όλους εκτός από έναν στην Κάρολ με τη σειρά της αντιμετάθεσης, το έναν μετά το άλλο.
- Η Κάρολ πρέπει να βρει τον αριθμό που λείπει.
- Η Κάρολ δεν μπορεί να θυμάται όλους τους αριθμούς που έχει δει.

Πώς;;;;

- Η Κάρολ όμως τον βρίσκει!

Η λύση για τον αριθμό που λείπει

- Η Κάρολ υπολογίζει το άθροισμα όλων των αριθμών που έχει δει.
- Κρατάει ένα τρέχον άθροισμα **S** και προσθέτει κάθε αριθμό **j** που έρχεται σ' αυτό: **S = S + j**.
- Στο τέλος, το αφαιρεί από το συνολικό άθροισμα των αριθμών **1,2,…,N** που είναι γνωστό και ίσο με: **N(N+1)/2**.
- Δηλαδή, αν **x** ο αριθμός που λείπει, τότε ισχύει:

$$x = \frac{N(N+1)}{2} - S$$

Οι απαιτήσεις χώρου-χρόνου

- Η Κάρολ χρησιμοποιεί χώρο **O(logN)** bits για την αποθήκευση του αθροίσματος **S**.
- Εκτελεί μία πρόσθεση κάθε φορά που ο Παύλος της δείχνει έναν αριθμό, η οποία απαιτεί χρόνο υπολογισμού **O(logN)**.
- Στο τέλος κάνει μία αφαίρεση από το **N(N+1)/2** η οποία απαιτεί χώρο **O(logN)** bits και χρόνο υπολογισμού **O(logN)**, οπότε υπολογίζει με απόλυτη ακρίβεια τον αριθμό που λείπει.
- Συνολικός χώρος και χρόνος: **O(logN)**
- Αυτό το πρόβλημα ροής ανήκει στο cash-register μοντέλο (αλλά χωρίς επαναλήψεις αντικειμένων).
- Αν λείπουν **δύο** αριθμοί;

Η λύση για δύο αριθμούς

- Θα μπορούσε η Κάρολ να κρατάει και το τρέχον γινόμενό τους **P**, δηλαδή: **P = P * j**
- Τότε αν **x** και **y** οι δύο αριθμοί που λείπουν θα ισχύουν οι ισότητες:

$$x + y = \frac{N(N+1)}{2} - S \qquad\qquad x \cdot y = \frac{N!}{P}$$

- Από τη λύση του συστήματος μπορούν να υπολογιστούν με απόλυτη ακρίβεια τα **x, y**.
- Όμως η τιμή του **P** μεγαλώνει πολύ απότομα (σχεδόν όσο και το **N!**) και έτσι δεν θα χωρέσει στην μνήμη.
- Άλλη εναλλακτική;

Η λύση για δύο αριθμούς

- Η Κάρολ είναι έξυπνο κορίτσι και τελικά αποφασίζει να κρατήσει το τρέχον άθροισμα των τετραγώνων τους **Sq**, δηλαδή: **Sq = Sq + j²**
- Τότε αν **x** και **y** οι δύο αριθμοί που λείπουν θα ισχύουν οι ισότητες:

$$x + y = \frac{N(N+1)}{2} - S \qquad x^2 + y^2 = \frac{N(N+1)(2N+1)}{6} - Sq$$

- Από τη λύση του συστήματος μπορούν να υπολογιστούν με απόλυτη ακρίβεια τα **x, y**.
- Ο χώρος και ο χρόνος που απαιτείται είναι πάλι **O(logN)**.
- Όμως αν έχουμε **k>>2** αριθμούς που λείπουν το πρόβλημα δυσκολεύει και οι απαιτήσεις για την ακριβή λύση του είναι πλέον πολύ μεγάλες.

Υπολογισμοί με Προσέγγιση ή/και Τυχαιότητα

- Πολλά άλλα προβλήματα είναι δύσκολο να απαντηθούν με απόλυτη ακρίβεια στην περίπτωση των ροών. Π.χ.
  - Είναι το πλήθος όλων των αντικειμένων ίδιο σε δύο διαφορετικές ροές;
    - Απαιτείται γραμμικός χώρος **O(N)** για να απαντήσουμε με απόλυτη ακρίβεια στο πρόβλημα αυτό.

- **Υπολογισμός με Προσέγγιση:** Να βρεθεί μία απάντηση που να προσεγγίζει το σωστό αποτέλεσμα μέσα σε κάποιο εύρος τιμών. Π.χ.
  - Μας ενδιαφέρει να βρεθεί μία απάντηση που να αποκλίνει το πολύ **±10%** από το σωστό αποτέλεσμα.
  - Γενικά αν **ε** ο παράγοντας της προσέγγισης τότε το διάστημα τιμών για την απάντηση θα είναι: $(1 \pm \varepsilon)$.

Υπολογισμοί με Προσέγγιση ή/και Τυχαιότητα

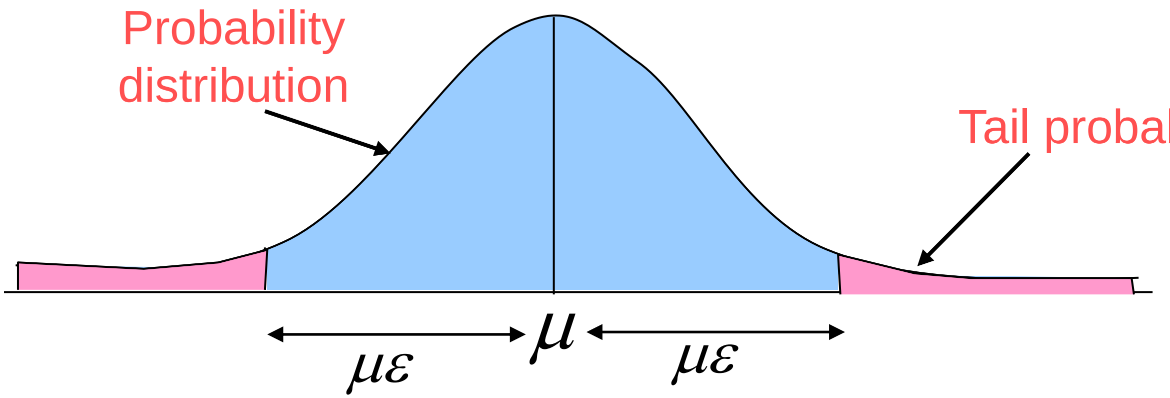
- **Υπολογισμός με Τυχαιότητα:** Να βρεθεί η απάντηση που να είναι η πιο πιθανή επιτρέποντας μία μικρή πιθανότητα αποτυχίας. Π.χ.
  - Μας ενδιαφέρει να βρεθεί η σωστή απάντηση με πιθανότητα λάθους 1 προς 10000.
  - Γενικά αν **δ** η πιθανότητα της αποτυχίας τότε η πιθανότητα επιτυχίας της απάντησης θα είναι: **(1 - δ)**.
- **Υπολογισμός με Προσέγγιση και με Τυχαιότητα:** Συνδυασμός και των δύο τρόπων με παραμέτρους τα **ε**,**δ**.
  - Συμβολίζουμε τις προσεγγίσεις αυτές ως (**ε**,**δ**)-προσεγγίσεις.
  - Οι τιμές των παραμέτρων ρυθμίζονται από τον χρήστη.
  - Παράδειγμα: Ο χρήστης ενδιαφέρεται να βρει μία απάντηση που να αποκλίνει το πολύ **±20% (ε=0.2)** από το σωστό αποτέλεσμα με πιθανότητα λάθους **10% (δ=0.1).** Αν η σωστή απάντηση είναι **5** τότε η απάντηση που θα πάρουμε θα είναι **5 ± 1** με πιθανότητα **≥0.9**.

Εγγυήσεις πιθανοτήτων (Probabilistic Guarantees)

- Αλγόριθμοι που υπολογίζουν με τυχαιότητα (Randomized algorithms): Η τιμή της απάντησης που επιστρέφεται θεωρείται ότι προέρχεται από μία ειδικά κατασκευασμένη ***τυχαία μεταβλητή***.
  - Είναι *unbiased* (δηλαδή η αναμενόμενη/μέση τιμή της είναι και η σωστή απάντηση στο πρόβλημα)
  - Συνδυάζει αρκετά ανεξάρτητα και ιδανικά κατανεμημένα στιγμιότυπα της απάντησης (*Independent Identically Distributed*) (μέση τιμή = διάμεσος)
- Χρησιμοποιούνται οι *ανισότητες ουρών* (*Tail Inequalities*) για να δοθούν όρια πιθανοτήτων στην απάντηση:
  - *Markov Inequality*
  - *Chebyshev Inequality*
  - *Chernoff Bound*
  - *Hoeffding Bound*

Βασικά εργαλεία: Tail Inequalities

- Οριοθετούν την πιθανότητα της ουράς *tail probability* μίας τυχαίας μεταβλητής (δηλαδή την πιθανότητα η τυχαία μεταβλητή να αποκλίνει μακριά από την μέση τιμή της)



- <u>Βασικές Ανισότητες:</u> Αν **X** μία τυχαία μεταβλητή με μέση τιμή **μ** και διακύμανση **Var[X]**, τότε για κάθε **ε>0** ισχύει:

Markov:

$$\Pr(X \geq (1+\varepsilon)\mu) \leq \frac{1}{1+\varepsilon}$$

Chebyshev:

$$\Pr(|X-\mu| \geq \mu\varepsilon) \leq \frac{\mathrm{Var}[X]}{\mu^2\varepsilon^2}$$

Tail Inequalities για αθροίσματα

- Μπορούμε να πάρουμε ισχυρότερα όρια στις πιθανότητες ουρών (tail probabilities) για αθροίσματα ανεξάρτητων τυχαίων μεταβλητών.
- *Hoeffding Bound:* Έστω $\mathbf{X_1,\ldots,X_m}$ ανεξάρτητες τυχαίες μεταβλητές με τιμές $\mathbf{0} \le \mathbf{X_i} \le \mathbf{r}$. Έστω $\bar{X} = \frac{1}{m}\sum_i X_i$ και $\boldsymbol{\mu}$ η μέση τιμή της μεταβλητής $\bar{X}$.

  Τότε για κάθε $\boldsymbol{\varepsilon>0}$ ισχύει:

$$\Pr(|\bar{X} - \mu| \ge \varepsilon) \le 2\exp^{\frac{-2\mu\cdot\varepsilon^2}{r^2}}$$

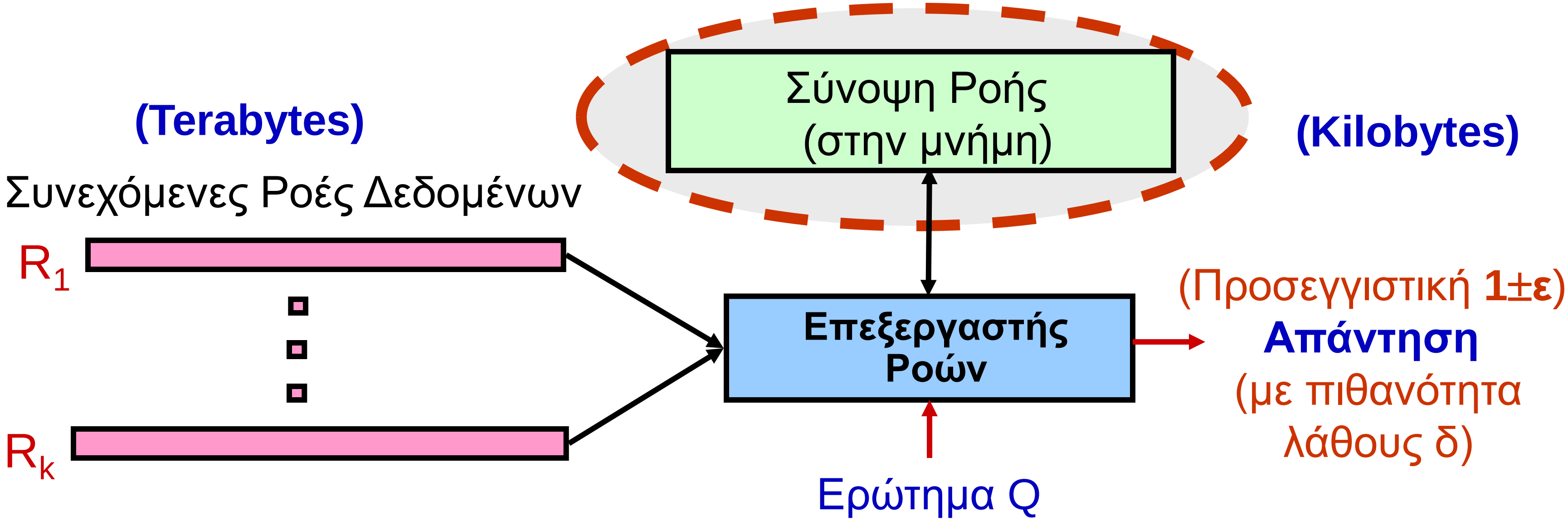
Tail Inequalities για αθροίσματα

- Μπορούμε να πάρουμε και ακόμα πιο ισχυρά όρια στις πιθανότητες ουρών (tail probabilities) για το άθροισμα ανεξάρτητων δοκιμών *Bernoulli.*
- *Chernoff Bound:* Έστω $X_1,\ldots,X_m$ ανεξάρτητες δοκιμές Bernoulli τέτοιες ώστε $\Pr[X_i=1] = p$ (οπότε $\Pr[X_i=0] = 1-p$). Έστω $X = \sum_i X_i$ και $\mu = mp$ η μέση τιμή της **X**. Τότε για κάθε **ε>0** ισχύει:

$$\Pr(|X-\mu| \geq \mu\varepsilon) \leq 2\exp^{\frac{-\mu\varepsilon^2}{2}}$$

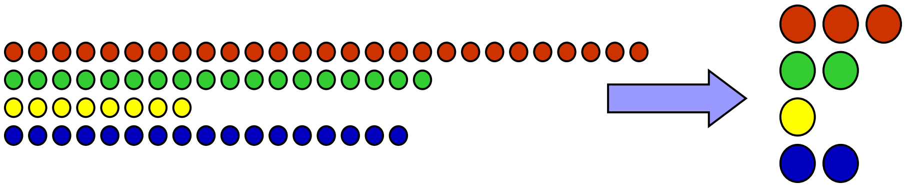
- Παρατήρηση: Το όριο Chernoff είναι πιο ισχυρό σε ερωτήματα απαρίθμησης (count queries) σε σχέση με το όριο Hoeffding.

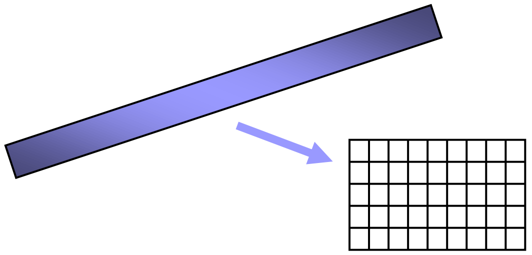
Γενικό Μοντέλο Αλγορίθμων Ροών



- Απαιτήσεις για την σύνοψη της ροής:
  - *Μονό Πέρασμα (Single Pass):* Κάθε στοιχείο ελέγχεται το πολύ μία φορά σε προκαθορισμένη σειρά με βάση την άφιξή του.
  - *Μικρός χώρος:* Log ή poly-log με βάση το μέγεθος της ροής
  - *Μικρός χρόνος:* Ο χρόνος επεξεργασίας του κάθε στοιχείου αλλά και της ενημέρωσης της σύνοψης να είναι μικρός.
  - Και ίσως επιπλέον απαιτήσεις, …

Δειγματοληψία και Sketches





Δειγματοληψία: Βασικές Αρχές

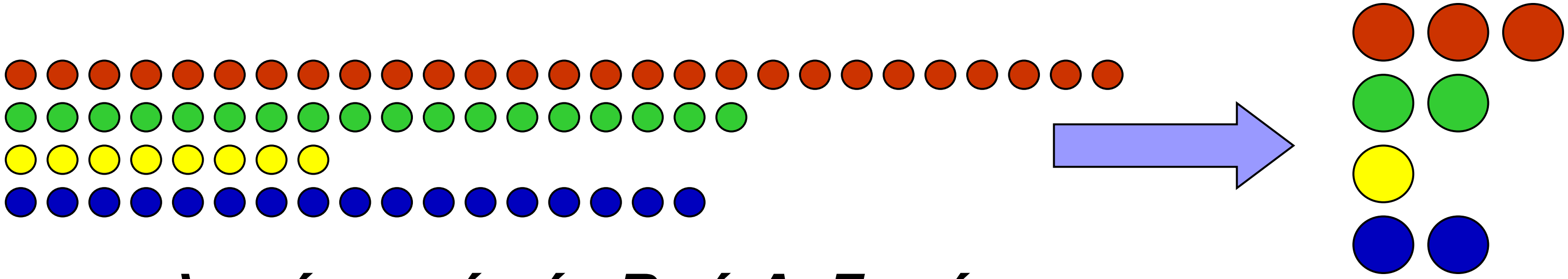
- **Ιδέα:** Ένα μικρό **τυχαίο** δείγμα **S** των δεδομένων συχνά αντιπροσωπεύει καλά όλα τα δεδομένα.
  - Για μία γρήγορη απάντηση με προσέγγιση, επεξεργαζόμαστε το ζητούμενο ερώτημα (τροποποιημένο) στο δείγμα **S**.
  - Παράδειγμα: (**N=12,** τροποποίηση: μόνο περιττά στοιχεία)

Data stream:	9	3	5	2	7	1	6	5	8	4	9	1
Sample **S**:	9	5	1	8								

  - Αν το ερώτημα είναι ο μέσος όρος (avg), τότε θα βρούμε τον μέσο όρο μόνο των περιττών στοιχείων στο **S** και η απάντηση θα είναι: **(9+5+1)/3 = 5.**
  - Αν το ερώτημα είναι η καταμέτρηση (cnt) τότε θα αντικαταστήσουμε κάθε περιττό στοιχείο στο **S** με **N** και κάθε άλλο με **0** και θα βρούμε τον μέσο όρο τους, δηλ. **(N+N+N+0)/4 = 3*12/4 = 9.**

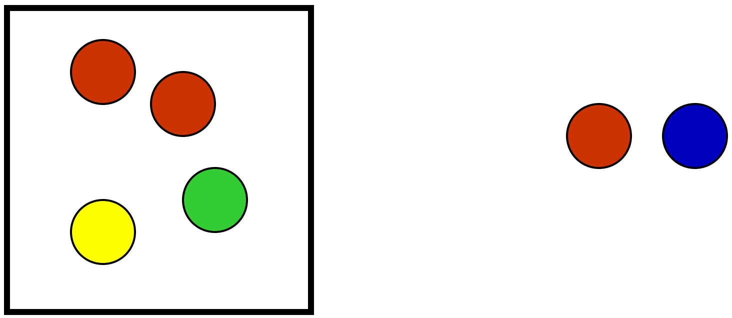
Δειγματοληψία: Βασικές Αρχές

- ***Unbiased Εκτιμητής*** (για ερωτήματα count, avg, sum, etc.)
  - Όρια λάθους από *Hoeffding* (sum, avg) ή *Chernoff* (count)



- ***Δειγματοληψία από μία Ροή Δεδομένων:***
  - Θεμελιώδες πρόβλημα: η δειγματοληψία **m** αντικειμένων **ομοιόμορφα** από μία ροή.
    - Χρήσιμη: προσέγγιση ενός υπολογισμού μεγάλου κόστους με βάση μόνο ένα μικρό δείγμα.
  - Πρόβλημα: όταν δεν γνωρίζουμε το συνολικό μέγεθος της ροής.
    - Τότε πότε και πόσο συχνά θα παίρνουμε δείγματα;
  - Δύο λύσεις που εφαρμόζονται σε διαφορετικές καταστάσεις:
    - Reservoir sampling (χρονολογείται από τη δεκαετία του ‘80)
    - Min-wise sampling (χρονολογείται από τη δεκαετία του ‘90)

Reservoir Sampling



- Παίρνουμε τα πρώτα **m** αντικείμενα (αρχικό δείγμα)
- Το **i**-στο αντικείμενο (**i>m**) επιλέγεται με πιθανότητα **m/i**
- Αν επιλεχθεί, αντικαθιστά τυχαία ένα αντικείμενο που βρίσκεται ήδη στο δείγμα.

- Optimization: Αν το **i** γίνει πολύ μεγάλο, υπολογίζεται ποιο αντικείμενο θα επιλεχθεί για το δείγμα, παρακάμπτοντας ορισμένα κυρίαρχα αντικείμενα [Vitter'85]

Reservoir Sampling - Ανάλυση

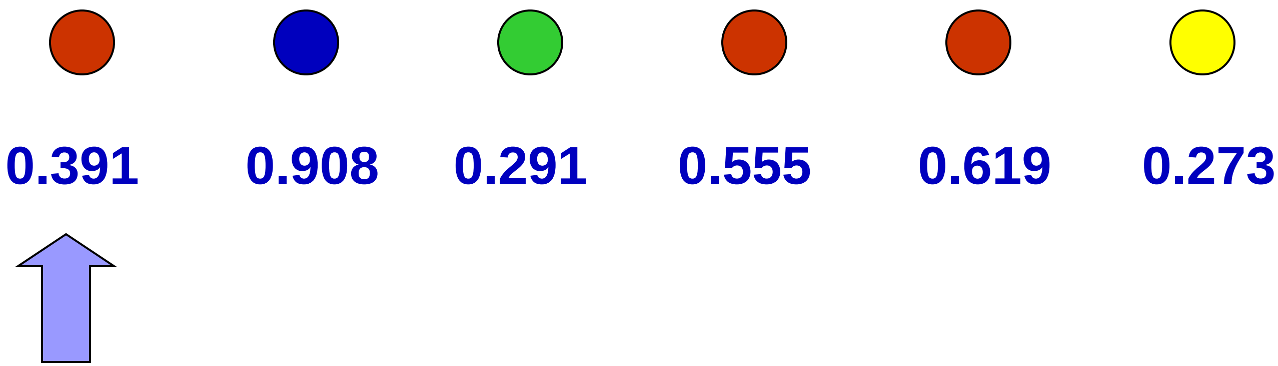
- Ανάλυση για την πιο απλή περίπτωση όπου το μέγεθος του δείγματος είναι **m = 1**
- Η πιθανότητα του **i**-στού αντικειμένου να βρίσκεται στο δείγμα από μία ροή μεγέθους **n** είναι:
  - Ίση με την πιθανότητα το **i** να επιλεχθεί στο δείγμα κατά την άφιξή του επί την πιθανότητα το **i** να παραμείνει μέχρι το τέλος της ροής στο δείγμα, δηλαδή:

$$\frac{1}{\cancel{i}} \times \frac{\cancel{i}}{\cancel{i+1}} \times \frac{\cancel{i+1}}{\cancel{i+2}} \cdots \frac{\cancel{n-2}}{\cancel{n-1}} \times \frac{\cancel{n-1}}{n} = 1/n$$

- Η περίπτωση για **m > 1** είναι παρόμοια.
- Έτσι οι πιθανότητες ακολουθούν την ομοιόμορφη κατανομή (uniform).
- Μειονέκτημα: Είναι δύσκολη η παραλληλοποίηση της μεθόδου αυτής.

Min-wise Sampling

- Σε κάθε αντικείμενο τοποθετούμε μία ετικέτα που έχει έναν τυχαίο δεκαδικό αριθμό μεταξύ 0 και 1.
- Η δειγματοληψία γίνεται με βάση τον μικρότερο αριθμό ετικέτας [Nath et al.'04].



- Κάθε αντικείμενο έχει την ίδια πιθανότητα να αποκτήσει την μικρότερη ετικέτα, άρα έχουμε ομοιόμορφη κατανομή πιθανότητας.
- Μπορεί να εφαρμοστεί και σε πολλαπλές ροές χωριστά και μετά να ενωθούν τα δείγματα.

Άλλες μορφές δειγματοληψίας

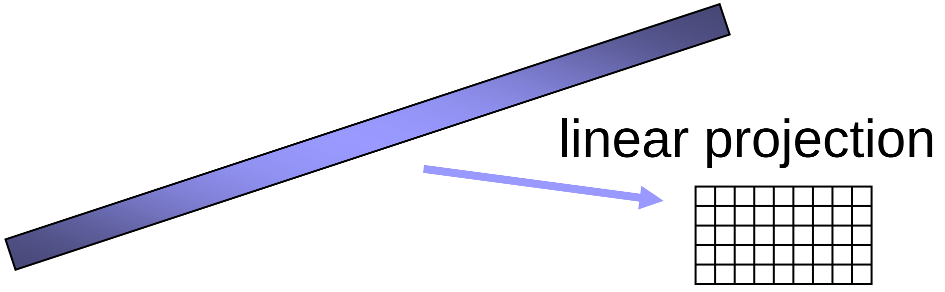
- Έχουν προταθεί πολλές ακόμα μορφές δειγματοληψίας:
  - Domain sampling
  - Universe sampling
  - Priority sampling
  - Quantile sampling
  - Frequent Elements/Heavy Hitters Sampling
  - Distinct sampling
  - Subset-Sum Sampling
  - κλπ.
- Γενικά η δειγματοληψία έχει αρκετά πλεονεκτήματα καθώς οι αλγόριθμοι υπολογισμών εφαρμόζονται σε μικρά δείγματα μόνο αγνοώντας τον τεράστιο όγκο των δεδομένων.

Προβλήματα

- Δεν γίνεται όμως όλα τα προβλήματα να αντιμετωπιστούν με την δειγματοληψία.
  - Παράδειγμα: *Πόσα διαφορετικά αντικείμενα υπάρχουν μέσα σε μία ροή*;
  - Αν ένα μεγάλο ποσοστό αντικειμένων αγνοηθούν, λόγω του ότι ασχολούμαστε μόνο με τα στοιχεία του μικρού δείγματος, δεν είμαστε σίγουροι αν όλα τα υπόλοιπα στοιχεία είναι ίδια με αυτά που έχουμε στο δείγμα ή (στο άλλο άκρο) είναι όλα διαφορετικά από αυτά που έχουμε στο δείγμα.
- Άλλες τεχνικές έχουν το πλεονέκτημα ότι μπορούν να «δουν» όλα τα δεδομένα ακόμα και αν δεν μπορούν να τα «θυμούνται» όλα.

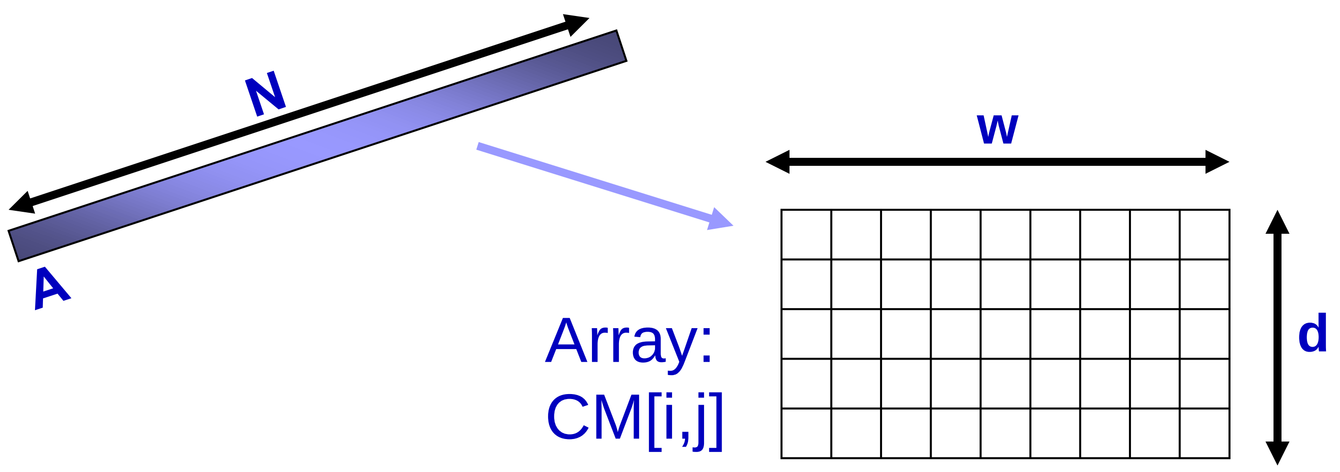
Sketches

- *"Sketch":* είναι **γραμμικός μετασχηματισμός** των δεδομένων της ροής καθώς διαβάζονται (input)
  - Αν θεωρήσουμε την ροή ως ένα διάνυσμα, τότε το sketch είναι το αποτέλεσμα του πολλαπλασιασμού του διανύσματος των στοιχείων της ροής με έναν πίνακα (implicit matrix)
  - Αν ο πίνακας εμπεριέχει τυχαιότητα τότε τα sketches λέγονται και random projections.



Count-Min Sketch [Cormode, Muthukrishnan’04]

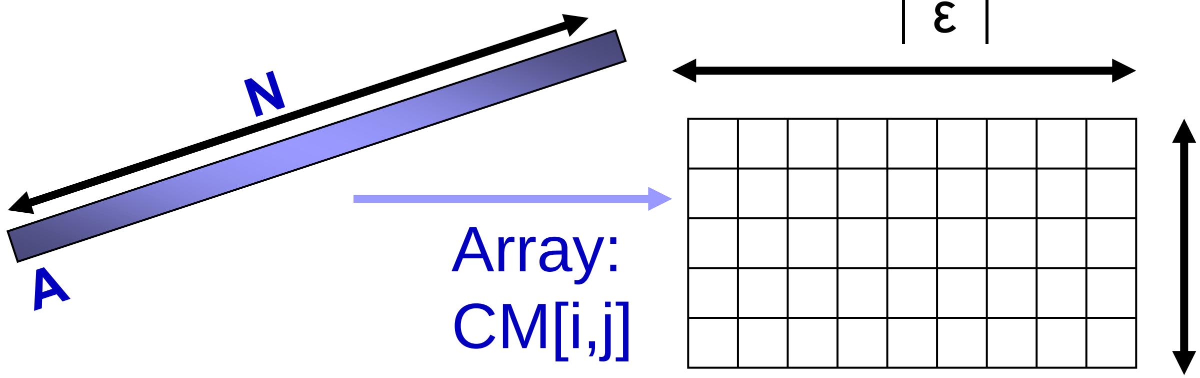
- Είναι μία απλή μέθοδος sketch που χρησιμοποιείται ως βάση για πολλά διαφορετικά ερωτήματα και προβλήματα πάνω σε ροές:
  - Join aggregates, range queries, moments, …
- Η ροή μοντελοποιείται ως ένα διάνυσμα **A** διάστασης **N**.
- Δημιουργεί μία συνοπτική εικόνα της ροής με τη μορφή ενός πίνακα **CM** διάστασης **w × d**. Αρχικά όλα τα στοιχεία του είναι **0**.
- Χρησιμοποιούνται **d** συναρτήσεις hash που απεικονίζουν τις τιμές του διανύσματος στο διάστημα **[1..w]**.



Count-Min Sketch [Cormode, Muthukrishnan'04]

- Οι συναρτήσεις hash κάνουν απεικονίσεις όσο το δυνατόν με τυχαιότητα (είναι από ομοιόμορφη οικογένεια), και είναι **ανά δύο ανεξάρτητες μεταξύ τους**. Αυτό σημαίνει ότι κάθε στοιχείο της ροής θα απεικονίζεται στις **d** γραμμές αλλά όχι απαραίτητα στην ίδια θέση.
- Αν οι παράμετροι προσέγγισης και τυχαιότητας είναι οι **δ,ε**, τότε οι παράμετροι του sketch ορίζονται ως εξής:

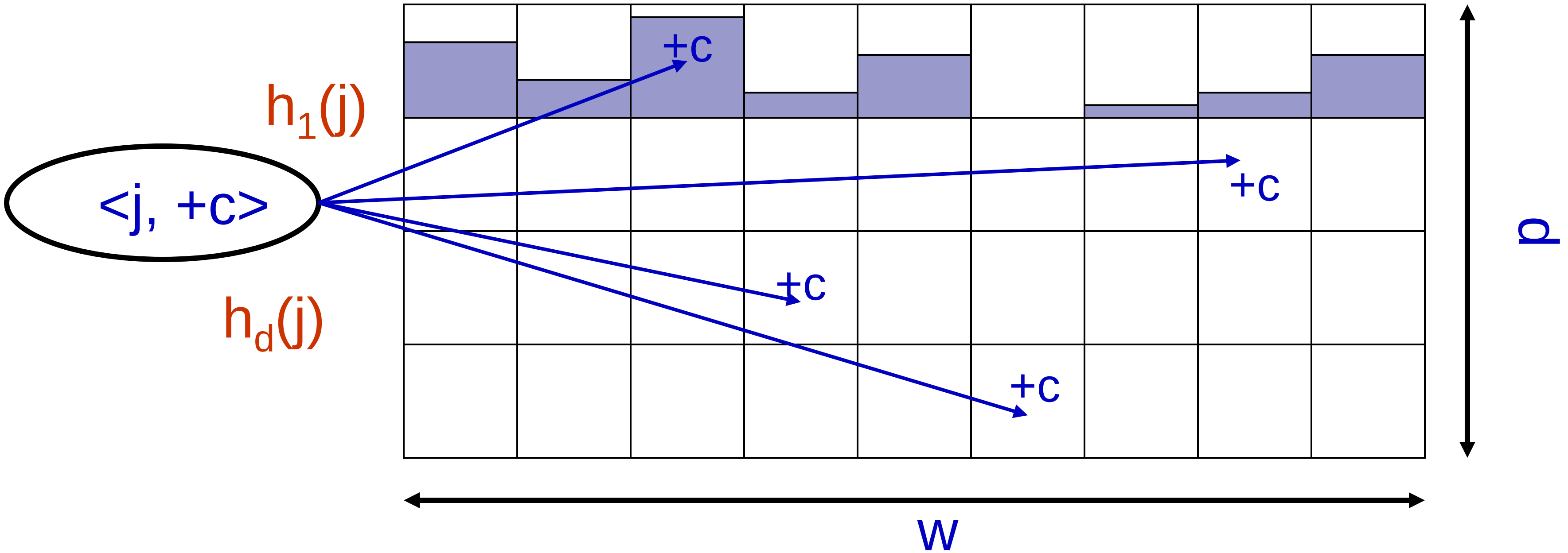
$$w = \left\lceil \frac{2}{\varepsilon} \right\rceil$$



$$d = \left\lceil \log_2 \frac{1}{\delta} \right\rceil$$

- Συνεπώς οι τιμές των **w,d** ρυθμίζουν το μέγεθος της ακρίβειας της προσέγγισης και την πιθανότητα λάθους αντίστοιχα.

Δομή του Count-Min Sketch



- Κάθε στοιχείο που διαβάζεται από την ροή **A[ ]** απεικονίζεται σε ένα κελί ανά γραμμή, από τις συναρτήσεις hash **h()**.
- Η εκτίμηση μιας τιμής **A[j]** (point query) γίνεται υπολογίζοντας την τιμή:

$$A'[j] = \min_k \{ CM[k,h_k(j)] \}$$

Εγγυήσεις της μεθόδου CM Sketch

- *[Cormode, Muthukrishnan'04]* Η μέθοδος CM sketch εγγυάται ότι το σφάλμα της προσέγγισης σε ερωτήματα αναζήτησης μίας τιμής **A[j]** (point queries) είναι μικρότερο από **$\varepsilon \cdot \|A\|_1$** όπου $\|A\|_1 = \sum_i A[i]$, ενώ απαιτεί χώρο: **O(1/ε log 1/δ)**.
  - Η πιθανότητα για μεγαλύτερο σφάλμα είναι μικρότερη από **1-δ**.
  - Παρόμοιες εγγυήσεις υπάρχουν και για ερωτήματα τύπου: range queries, quantiles, join size,κλπ.
- Ορισμένα σημαντικά σημεία:
  - Οι καταμετρήσεις παραβιάζονται *(γίνονται υπερεκτιμήσεις)* λόγω των συγκρούσεων από τις συναρτήσεις hash.
    - Μπορούμε να περιορίσουμε το μέγεθος του έξτρα «όγκου» που μαζεύεται σε κάθε κελί;
  - Η ανεξαρτησία μεταξύ των γραμμών περιορίζει το διάστημα εμπιστοσύνης για την εκτίμηση του min{ } (καλύτερη ακρίβεια).
    - Βασίζεται στην ανεξαρτησία μεταξύ των συναρτήσεων hash.

Ανάλυση της μεθόδου CM Sketch

Έστω ότι η εκτίμηση του A[j] είναι: $A'[j] = \min_k \{ CM[k,h_k(j)] \}$

- Τότε στην k-στη γραμμή θα έχουμε: $CM[k,h_k(j)] = A[j] + X_{k,j}$
  - όπου $X_{k,j} = \sum_{h_k(i)=h_k(j)} A[i]$. Άρα:

    $$E\left[X_{k,j}\right] = \sum_{h_k(i)=h_k(j)} A[i] * \Pr[h_k(i) = h_k(j)] \leq \frac{\varepsilon}{2}\sum_i A[i] = \frac{\varepsilon}{2}\|A\|_1$$

    (λόγω του ότι οι h είναι ανά δύο ανεξάρτητες)
  - Άρα:

    $$\Pr[X_{k,j} \geq \varepsilon\|A\|_1] = \Pr[X_{k,j} \geq 2E[X_{k,j}]] \leq \tfrac{1}{2}$$ (ανισότητα Markov)

- Έτσι, $\Pr[A'[j] \geq A[j] + \varepsilon||A||_1] = \Pr[\forall k,\ X_{k,j} \geq \varepsilon||A||_1] \leq 1/2^{\log 1/\delta} = \delta$

- Συνεπώς με βεβαιότητα έχουμε $A[j] \leq A'[j]$ και

  με πιθανότητα το λιγότερο 1-δ θα ισχύει: $A'[j] < A[j] + \varepsilon\ ||A||_1$

Εφαρμογές της μεθόδου CM Sketch

- Η μέθοδος CM Sketch είναι βασικό εργαλείο για:
  - range-sum queries, quantiles, heavy hitters, moments, …
- Είναι ένα απλό sketch το οποίο μπορεί να χρησιμοποιηθεί σε μία μεγάλη ποικιλία διαφόρων αναλύσεων.
- Εφαρμόζεται στο Gigascope data stream system, όπου επεξεργάζεται 2 με 3 εκατομμύρια ενημερώσεις το δευτερόλεπτο.
- Λειτουργεί πάνω στο μοντέλο του μύλου (turnstile) κατασκευάζοντας μία σύνοψη των δεδομένων.
- Δύο ή περισσότερα sketches μπορούν να συγχωνευτούν με απλή άθροιση των καταχωρήσεών τους (εφαρμογή σε κατανεμημένα συστήματα).

Η εκτίμηση των διακριτών τιμών

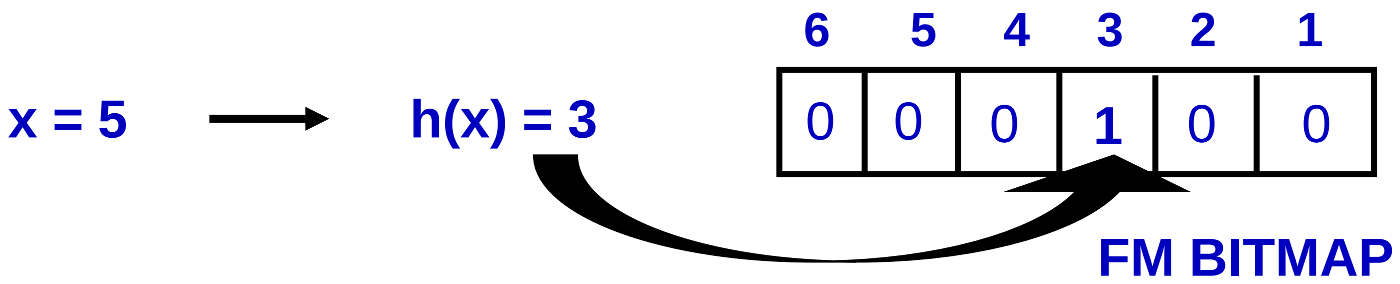
- Πρόβλημα: Να βρεθεί το *πλήθος των διακριτών τιμών* σε μία ροή δεδομένων που παίρνουν τιμές στο διάστημα **[1,...,N]**.
  - Σημαντικό στη στατιστική: καταμέτρηση πλήθους ειδών ή ομάδων σε έναν πληθυσμό.
  - Σημαντικό στη βελτιστοποίηση ερωτημάτων.
  - *Παρακολούθηση δικτύου* (καταμέτρηση): διακριτών διευθύνσεων IP προορισμού, ζευγών διευθύνσεων πηγής/προορισμού, URLs που ζητήθηκαν, κλπ.
- Παράδειγμα: (**N**=64)   *πλήθος διακριτών τιμών: 5*

Ροή δεδομένων: 3 2 5 3 2 1 7 5 1 2 3 7

- Δύσκολο πρόβλημα για να αντιμετωπιστεί με τυχαία δειγματοληψία! [Charikar et al.'00]
  - Πρέπει να πάρουμε δείγμα σχεδόν όσο το συνολικό μέγεθος της ροής για να εγγυηθούμε ότι η εκτίμηση είναι σωστή κατά ± 10 με πιθανότητα > 1/2, ανεξάρτητα με το είδος του εκτιμητή που θα χρησιμοποιηθεί!
- Άρα;

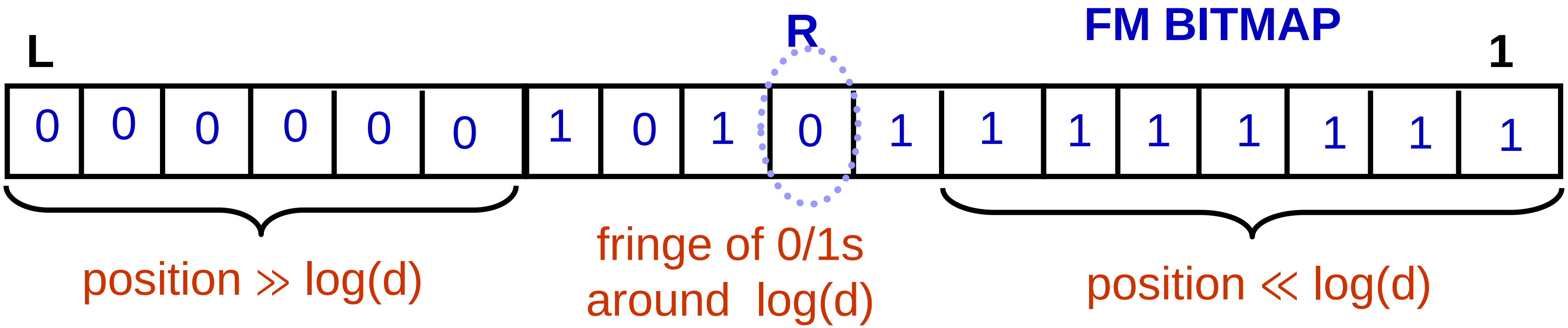
Η μέθοδος FM Sketch [Flajolet, Martin'85]

- Εκτιμά το πλήθος των διακριτών τιμών (**count distinct**).
- Χρησιμοποιεί μία συνάρτηση hash $\mathbf{h:[1,N] \rightarrow \{1,2,\log_2 N\}}$ που απεικονίζει τα στοιχεία εισόδου στο **i** με πιθανότητα $\mathbf{2^{-i}}$.
  - **Pr[h(x) = 1] = ½, Pr[h(x) = 2] = ¼, Pr[h(x)=3] = 1/8** …
  - Η κατασκευή της **h()** γίνεται εύκολα από μία ομοιόμορφη συνάρτηση hash καταμετρώντας τα trailing zeros.
- Το FM Sketch είναι ένα bitmap των **L = log N** bits.
  - Το bitmap αρχικοποιείται με μηδενικά.
  - Για κάθε εισερχόμενη τιμή **x**, θέτουμε: **FM[h(x)] = 1**.



Ανάλυση της μεθόδου FM Sketch

- Αν υπάρχουν **d** διακριτές τιμές, αναμένονται οι **d/2** να αντιστοιχούν στο **FM[1]**, οι **d/4** στο **FM[2]**, κλπ.



  - Έστω **R** η θέση του δεξιότερου μηδενικού στο bitmap **FM**. Τότε η θέση αυτή είναι η εκτίμηση του **log(d)**.
  - Τελική εκτίμηση: $\mathbf{d = c2^R}$ με σταθερά $\mathbf{c \approx 1.3}$.
  - Μπορούν να γίνουν διαφορετικά bitmaps (με διαφορετικές συναρτήσεις hash) και να πάρουμε το μέσο όρο των τελικών εκτιμήσεων ώστε να αυξήσουμε την ακρίβεια.

Ιδιότητες της μεθόδου FM Sketch

- Με **$O(1/\varepsilon^2 \log 1/\delta)$** διαφορετικά bitmaps, παίρνουμε **(1±ε)** ακρίβεια με πιθανότητα τουλάχιστον **1-δ** [Bar-Yossef et al'02], [Ganguly et al.'04]
  - 10 bitmaps ≈ 30% σφάλμα, 100 bitmaps < 10% σφάλμα
- *Υποστήριξη Διαγραφών:* Χρησιμοποιούνται μετρητές αντί για bits στις θέσεις του sketch:
  - +1 για εισαγωγές, -1 για διαγραφές
- *Υποστήριξη κατανεμημένων sketches:* με χρήση του τελεστή **OR** μπορούν να συγχωνευτούν:

6	5	4	3	2	1		6	5	4	3	2	1		6	5	4	3	2	1
0	0	1	0	1	1	+	0	1	1	0	0	1	=	0	1	1	0	1	1

  - Εκτίμηση του: **$|S_1 \cup \ldots \cup S_k|$** = πλήθος στοιχείων ένωσης συνόλων.

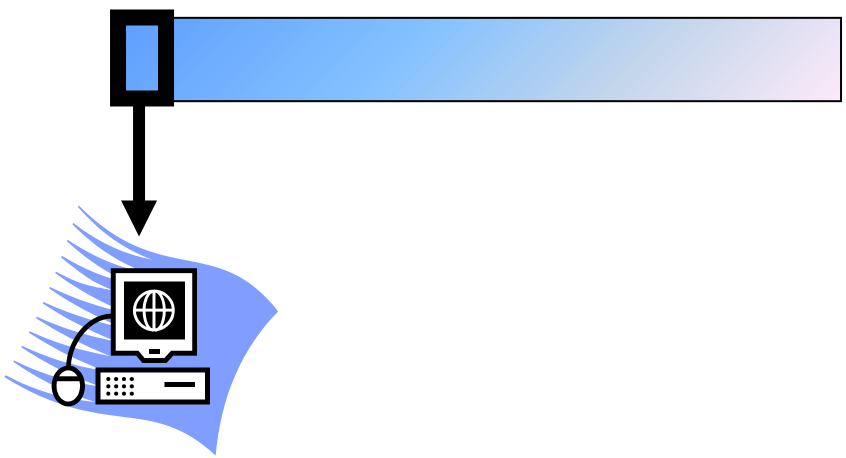
Δειγματοληψία και Sketching

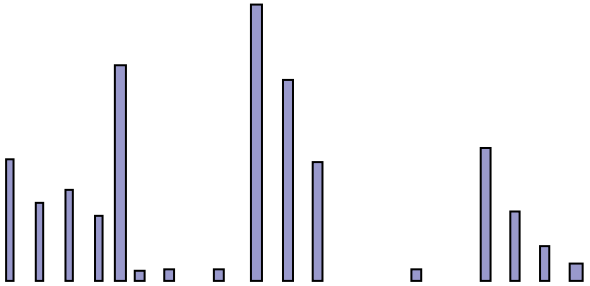
- Οι μέθοδοι δειγματοληψίας και sketching είναι στον πυρήνα πολλών αλγορίθμων ροών δεδομένων
  - Moments/join aggregates, histograms, wavelets, top-k, frequent items, other mining problems, …
- Ένα δείγμα είναι ένας γενικός αντιπρόσωπος των δεδομένων, ενώ τα sketches τείνουν να είναι *εξειδικευμένα για έναν ειδικό σκοπό*
  - FM sketch for count distinct, CM/AMS sketch for joins / moment estimation, …
- Η παραδοσιακή δειγματοληψία δεν μπορεί να εφαρμοστεί στο μοντέλο του μύλου (turnstile: arrivals & departures)
  - Αλλά… υπάρχουν γενικεύσεις για διακριτή δειγματοληψία [Ganguly et al.'04], [Cormode et al.'05], καθώς και [Gemulla et al.'08]

Πρακτικότητα

- Οι αλγόριθμοι που παρουσιάστηκαν είναι από τους πιο απλούς και πιο γρήγορους:
  - Τα Sketches μπορούν εύκολα να επεξεργαστούν χιλιάδες ενημερώσεις ανά δευτερόλεπτο σε τυπικούς Η/Υ.
  - Περιοριστικοί παράγοντες είναι στην πράξη μόνο αυτοί που σχετίζονται με I/O.
- Υλοποιήθηκαν σε αρκετά πρακτικά συστήματα:
  - AT&T's Gigascope system on live network streams
  - Sprint's CMON system on live streams
  - Google's log analysis
- Υπάρχουν υλοποιήσεις διαθέσιμες στο δίκτυο:
  - **http://www.cs.rutgers.edu/~muthu/massdal-code-index.html**
  - ή αναζήτηση του 'massdal'

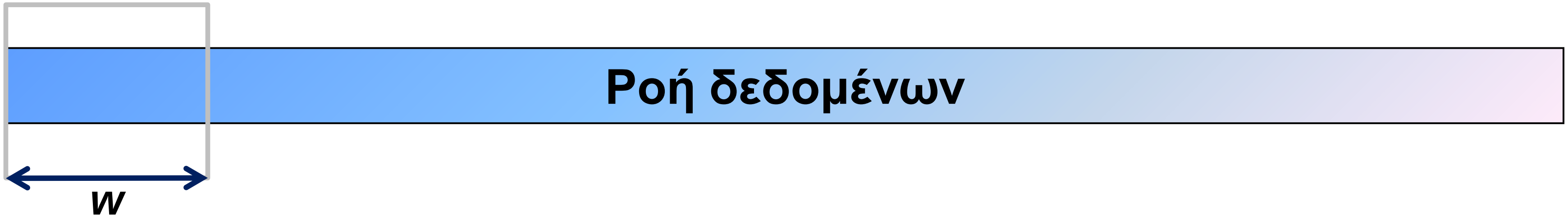
Άλλες μεθοδολογίες για ροές





Μοντέλο Κυλιόμενου Παράθυρου

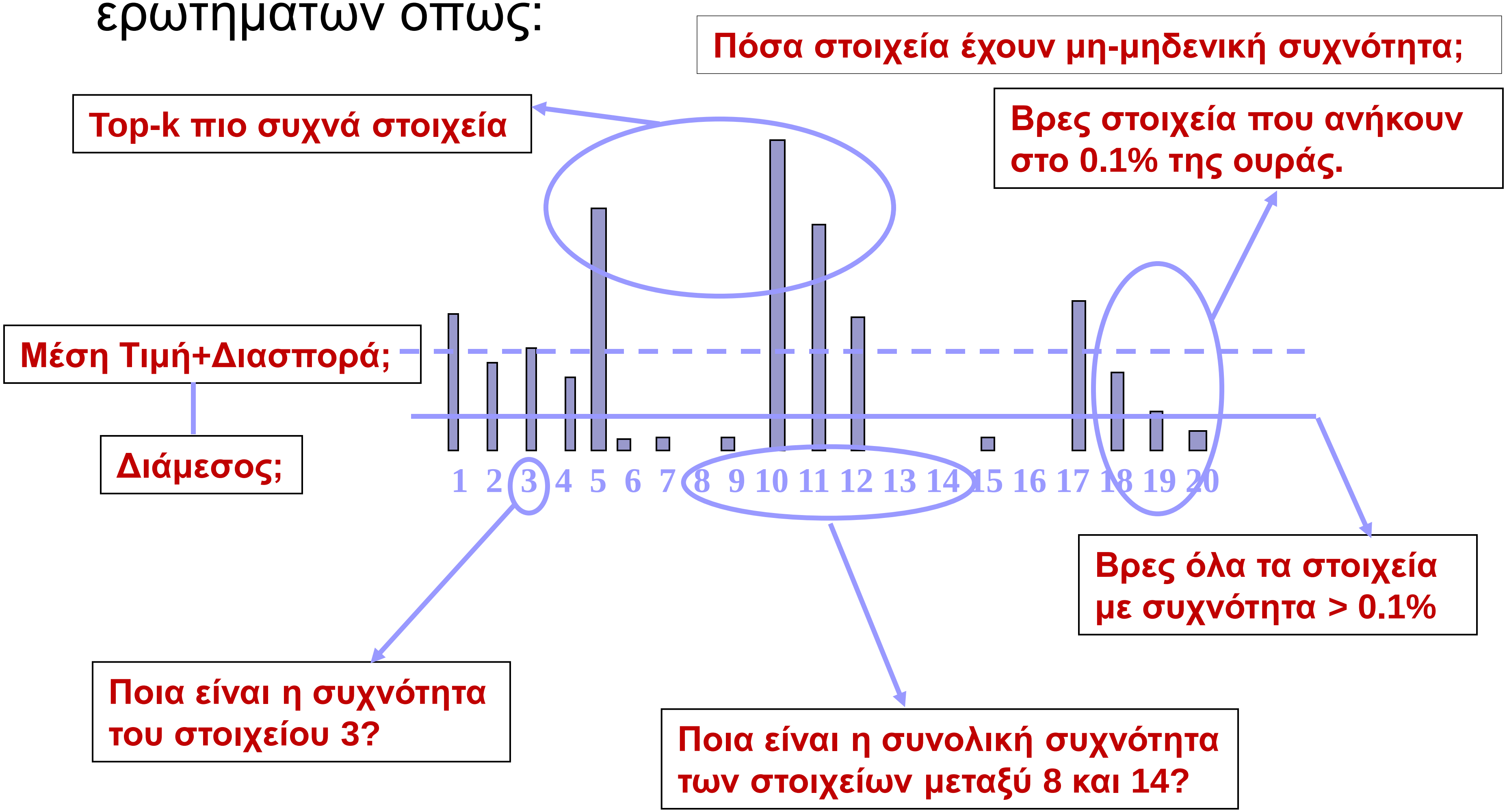
- Είναι ένα δημοφιλές μοντέλο ροών όπου η ροή δεδομένων είναι μία χρονοσειρά και μπορεί να είναι άπειρη.
- Ασχολούμαστε μόνο με τα **w** πιο πρόσφατα δεδομένα, όπου **w** το μέγεθος του παραθύρου (window size).
- Καθώς ο χρόνος περνά, έρχονται καινούργια δεδομένα ενώ «λήγουν» τα παλιά.
- Γίνεται διαγραφή των παλιών δεδομένων.



- Η διαχείριση των δεδομένων στο παράθυρο εξαρτάται από τον αλγόριθμο και το είδος των ερωτημάτων που διαχειρίζεται.

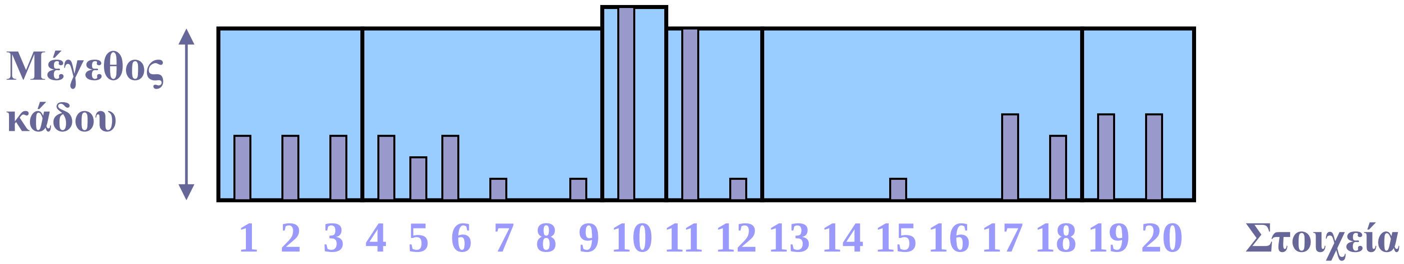
Ιστογράμματα

- Είναι βασικά εργαλεία για την καταμέτρηση συχνοτήτων σε δεδομένα ροών. Χρησιμοποιούνται για πολλά είδη ερωτημάτων όπως:



Ιστογράμματα

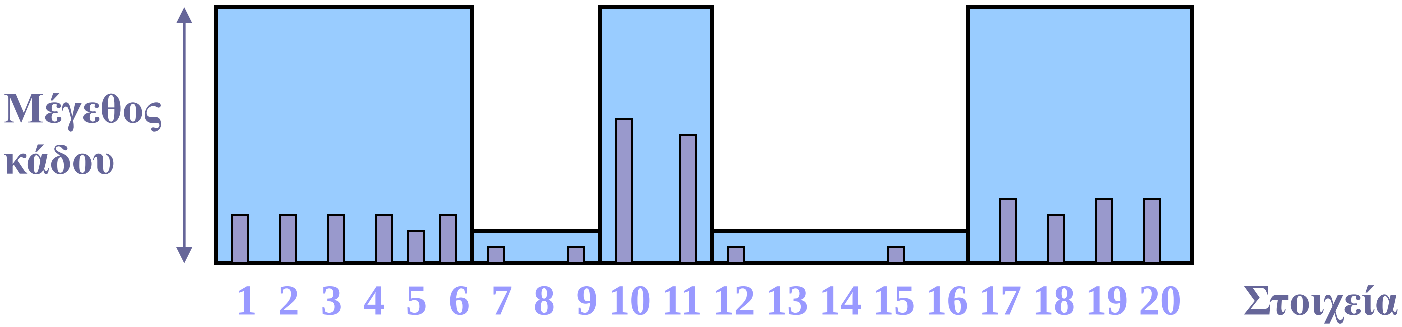
- Σε μερικές περιπτώσεις τα τυπικά ιστογράμματα δεν μπορούν να μας δώσουν καλή εικόνα των δεδομένων.
- **Ίσου βάθους Ιστογράμματα**
  - <u>Ιδέα:</u> Επιλέγονται κατάλληλα διαστήματα κλάσεων έτσι ώστε να υπάρχει ίδιο πλήθος στοιχείων ανά κλάση.
  - Δεν γνωρίζουμε εκ των προτέρων τα διαστήματα των κλάσεων, αλλά διαμορφώνονται κατά την ανάγνωση και επεξεργασία των δεδομένων της ροής.



Ιστογράμματα

- **V-Βέλτιστα Ιστογράμματα**
  - Ιδέα: Επιλέγονται κατάλληλα διαστήματα κλάσεων που να ελαχιστοποιούν τη διασπορά συχνότητας μεταξύ των κλάσεων
  - C_B → πλήθος στοιχείων στην κλάση Β
  - V_B → πλήθος διακριτών στοιχείων στην κλάση Β
  - f_v → συχνότητα στοιχείου v στην κλάση Β

$$\text{minimize} \quad \sum_B \sum_{v \in B} \left(f_v - \frac{C_B}{V_B} \right)^2$$



Άλλα είδη Ιστογραμμάτων

- Άλλα χρήσιμα είδη ιστογραμμάτων για εφαρμογή τους σε αλγορίθμους ροών είναι:
  - Set histograms
  - Multidimensional histograms
  - Range-sum histograms
  - B-Bucket histograms
  - κλπ.

Βασικά Στοιχεία για Wavelets

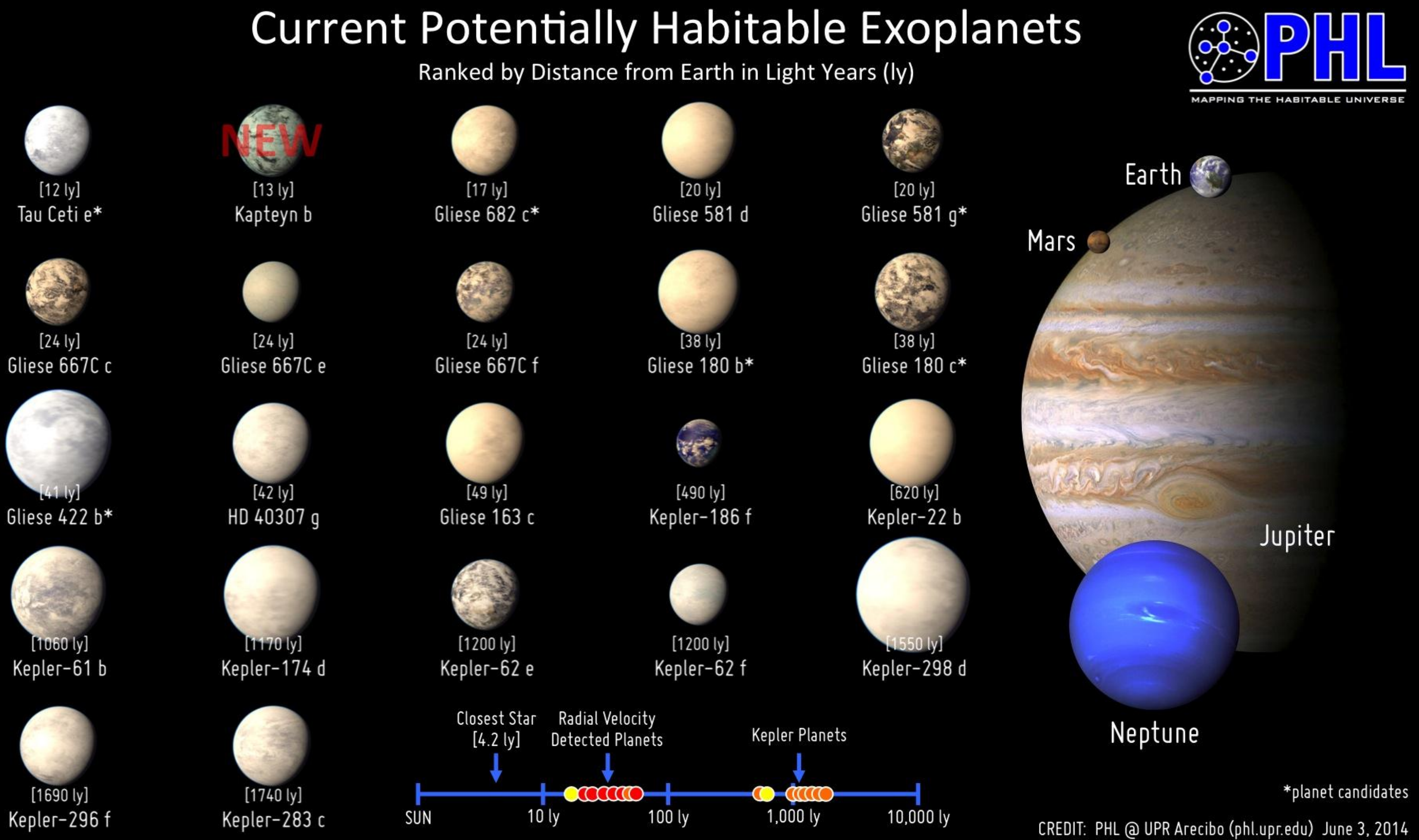
- Χρησιμοποιούνται σε αρκετές εφαρμογές ροών δεδομένων (χρονοσειρές, σήματα, ήχος, κλπ). Δίνουν μία σύνοψη των δεδομένων.
- Εφαρμόζονται σε ροές του μοντέλου των χρονοσειρών.
- Ο μετασχηματισμός Wavelet δείχνει τις κυρίαρχες τάσεις στο σήμα (στα δεδομένα της ροής).
- Ένας τυπικός μετασχηματισμός Wavelet απαιτεί **log(N)** επαναλήψεις (περάσματα) και **N-1** συναρτήσεις μετασχηματισμού οι οποίες καθορίζουν τις τιμές των συντελεστών του wavelet.
- Έξοδος Wavelet: διάνυσμα βάσης, μέσος όρος και οι τελικοί συντελεστές.

Βασικά Στοιχεία για Wavelet

- Οι Wavelet συντελεστές είναι προβολές του σήματος σε ένα ορθογώνιο σύνολο διανυσμάτων βάσης.
- Μία κατηγορία με μεγάλη εφαρμογή είναι τα Haar wavelets που χρησιμοποιούνται στις ΒΔ λόγω ευκολίας υπολογισμού τους.
- Το ανακατασκευασμένο σήμα από λίγους σημαντικούς wavelet συντελεστές, προσεγγίζει με τον καλύτερο τρόπο το αρχικό σήμα.

Συμπεράσματα

- Ροές δεδομένων: Μεγάλη απόκλιση από τα παραδοσιακά συστήματα των βάσεων δεδομένων.
  - Απαιτούν θεμελιώδεις επανασχεδιασμούς των μοντέλων, των υποθέσεων, των αλγορίθμων, των αρχιτεκτονικών των συστημάτων, κλπ.
- Πολλά καινούργια προβλήματα σε ροές διατυπώθηκαν από τις αναπτυσσόμενες τεχνολογίες.
- Τα απλά εργαλεία όπως η προσέγγιση ή/και η τυχαιότητα παίζουν σημαντικό ρόλο στο να έχουμε αποτελεσματικές λύσεις:
  - Sampling, sketches (CM, FM, …), …
  - Απλές αλλά ισχυρές ιδέες με μεγάλο εύρος εφαρμογής.
  - Μπορούν συχνά να συνδυαστούν μεταξύ τους.



Current Potentially Habitable Exoplanets
Ranked by Distance from Earth in Light Years (ly)
PHL
MAPPING THE HABITABLE UNIVERSE
NEW
[12 ly] Tau Ceti e*
[13 ly] Kapteyn b
[17 ly] Gliese 682 c*
[20 ly] Gliese 581 d
[20 ly] Gliese 581 g*
[24 ly] Gliese 667C c
[24 ly] Gliese 667C e
[24 ly] Gliese 667C f
[38 ly] Gliese 180 b*
[38 ly] Gliese 180 c*
[41 ly] Gliese 422 b*
[42 ly] HD 40307 g
[49 ly] Gliese 163 c
[490 ly] Kepler-186 f
[620 ly] Kepler-22 b
[1060 ly] Kepler-61 b
[1170 ly] Kepler-174 d
[1200 ly] Kepler-62 e
[1200 ly] Kepler-62 f
[1550 ly] Kepler-298 d
[1690 ly] Kepler-296 f
[1740 ly] Kepler-283 c
Closest Star [4.2 ly]
Radial Velocity Detected Planets
Kepler Planets
SUN
10 ly
100 ly
1,000 ly
10,000 ly
Earth
Mars
Jupiter
Neptune
*planet candidates
CREDIT: PHL @ UPR Arecibo (phl.upr.edu) June 3, 2014

ΤΕΛΟΣ