

A novel Distributed P2P Simulator Architecture: D-P2P-Sim

Spyros Sioutas
Department of Informatics
Ionian University
Corfu, Greece
sioutas@ionio.gr

George Papaloukopoulos
Computer Engineering and
Informatics Department
University of Patras
papalukg@ceid.upatras.gr

Evangelos Sakkopoulos
Computer Engineering and
Informatics Department
University of Patras
sakkopul@ceid.upatras.gr

Kostas Tsihclas
Department of Informatics
Aristotle University of
Thessaloniki, Greece
tsihclas@csd.auth.gr

Yannis Manolopoulos
Department of Informatics
Aristotle University of
Thessaloniki, Greece
manolopo@csd.auth.gr

ABSTRACT

In this paper we introduce a novel distributed simulation environment with GUI for P2P simulations (D-P2P-Sim). The key aim is to provide the appropriate integrated set of tools in a single software solution to evaluate the performance of various protocols. The basic architecture of the distributed P2P simulator is based on a multi-threading, asynchronous, message passing and distributed environment with graphical user interface to facilitate ease of use by both researchers and programmers. More information online: <http://students.ceid.upatras.gr/~papalukg>

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

Keywords

P2P simulator, distributed simulator

1. INTRODUCTION

In this section details concerning the developed distributed P2P simulation environment are discussed. There is a number of P2P simulation tools already presented in the extensive P2P simulators survey of S. Naicken et al [1]. According to the survey, the majority of the cases studied included specially created simulation tools for the evaluation of the protocols and algorithms presented. There are cases that the former simulators are reused by the research groups and affiliated ones. Even if one takes the latter into consideration, the survey finds that the custom made simulators far outnumber the use of known ones. As a result, comparison and replication of evaluation results is difficult besides the duplication of effort to develop again the same protocols once more from the beginning. A main outcome of the survey is

that ‘the poor state of existing P2P simulators is the reason that much published research makes use of custom built simulators’. The main drawbacks of the simulation mechanisms found were i) no systematic mechanism to gather statistics of the simulation executions, and ii) absence of graphical user interface support to support user friendliness.

We present a novel distributed P2P simulation environment (Java based), which is an integrated tool to overcome previous drawbacks in P2P simulators and includes several features which are for the first time to the best authors’ knowledge all together in a single integrated P2P distributed simulator. In the following, the architectural features of the distributed P2P simulator proposed are discussed.

Unbiased. The first key feature of the proposed simulator is the unbiased and independent from the protocol implemented mechanism to collect performance data. The mechanism is based on the observation and the filtering of special fields in the message. These fields are controlled by the simulator and are by default inaccessible from the protocol’s processes. This alternative way to collect performance data, in conjunction with the realistic execution environment yields to more accurate performance evaluation. Statistics are integrated for lookup, insert, delete, load balance and routing table length evaluation.

Realism. A simulation model is considered reliable when at least a) it depicts real life execution environment parameters on the simulation, b) it is unbiased and c) it is implemented as close as possible to an application level P2P software (e.g. single PC torrent client software). The execution model of the simulator is asynchronous/event driven and multi threading. These two features are fundamental for a real life P2P application. Although event based simulators have already been presented (e.g. peersim), when it comes to a simulation of large number of nodes then the ‘produce and consume’ model is adopted. We apply a pooling technique to achieve thousands/millions of nodes using the event driven approach which is more realistic. Furthermore, the proposed simulator can be executed as a single P2P network peer per node, behaving as a real life P2P application taking advantage of all the evaluation tools available under the most realistic possible conditions.

Distributed. D-P2P-Sim is a distributed simulator that may take advantage of multiple computer systems interconnected with network and of course clusters. As a result this allows conducting experimental simulations including very

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM Hong Kong, November 2-6, 2009

Copyright 2009 ACM X-XXXXX-XX-X/XX/XX ...\$10.00.

large number of peers, overcoming any limitations posed by a single stand alone computer system. This profit comes at the expense of the communication and coordination processes ($\sim 10\%$ overhead) introduced by the additional functionality.

Pluggable and Extensible. The D-P2P-Sim simulator is available on the Internet and facilitates the development of additional P2P protocols for evaluation. The extensibility is based on the plug in mechanism of Java. An API has been defined and it is available for further development. However, in order to make it easier for researchers and testers to develop a new protocol for the simulator, D-P2P-Sim includes a sample dummy implementation with all code and details for the basic design of a P2P protocol with its message transaction mechanism, routing management and architectural outline.

Efficient. The D-P2P-Sim is practical and efficient as it is supported by an integrated Graphical User Interface and it includes graphical Statistics functionality. Furthermore, the D-P2P-Sim simulator is extremely efficient delivering $>400,000$ peers in a single computer system. While, when it acts in a distributed environment with multiple computer systems with network connection delivers multiple times the former population of peers.

2. FUNCTIONALITY & IMPLEMENTATION

Throughout this section the functional specifications and the overall architecture design of the D-P2P-Sim simulator are described.

The GUI is organized in such a way that can guide someone easily through the simulation process with no need for any prior knowledge about the simulator. One may select the protocol of the overlay to be constructed and enter the number of peers that constitute the overlay network and its key capacity. Perform *step-by-step execution* of the lookup and update operations-algorithms, and verify the algorithm's correctness and proper functionality by observing the messages that are being displayed at real time.

Moreover, the user may design, in batch mode, multiple experiment scenarios to evaluate the protocol's efficiency. For each experiment it should be supplied the number of queries that will be performed and the distribution according to the keys will be picked up for the experiments. After the termination of the experiments the user has available statistical charts which are automatically generated. Additional performance statistics such as the average path length of the lookup, insert and delete algorithms, as the average size of the peers' routing table and the total number of operations that have been conducted at that time, are displayed.

Several options affecting the simulator's behavior are provided for certain tunable parameters concerning three main categories: the application nodes, the distributions and the execution environment. The first set of parameters specifies the applications nodes that constitute the cluster and the percentage of overlay peers that will be managed by each one. The second set specifies the parameters of Beta's and Power Law's distribution functions which affect their skewness, as well as the seed of the random engines. Finally, the third set specifies the number of threads created by the simulator, the debug level and the directories/files that are used by the simulator.

D-P2P-Sim simulator is delivered with code organized carefully in software packages achieving low *coupling* and high

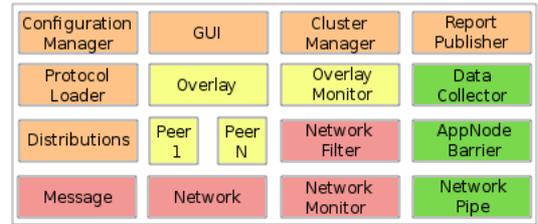


Figure 1: Overview of D-P2P-Sim's architectural modules.

cohesion, allowing in this way a great degree of modularity and a full understanding of its sophisticated design. In figure 1 are depicted with different colors the simulator's architectural modules and the packages that constitute each one. Four modules are distinguished a) the message passing environment, b) the overlay network, c) the remote services and d) the simulator's utilities.

Even if the packages' names are self-explanatory some of them arouse our interest and merits a greater degree of detail.

The *Network*, where the *Messages* are being stored, is continuously observed by the *Network Monitor* notifying the appropriate peer every time it encounters a message destined to it. Furthermore, a copy of that message is filtered by the *Network Filter* in order to get extracted useful statistical data, that will be further processed by the *Overlay Monitor*. Moreover, the *Overlay* is responsible to create and manage the dynamic loaded *Peers* that implement user defined protocols.

If the simulator acts in a distributed environment, messages which are destined to overlay peers residing to a remote application node, are being passed to the appropriate node using the *Network Pipe* service. The statistical data collected from each application node are passed to the main node using the *Data collector* service, while the *AppNode Barrier* service synchronizes the cluster's execution.

Due to space limitation additional technical details, architectural blueprints and multiple snapshots are available online at the simulator's website.

3. INTERACTIVE DEMONSTRATION PLAN

The demonstration includes two parts a) an introductory multimedia presentation that will show off the ease of use of the simulator and its efficiency and effectiveness and b) an interactive "live" event where multiple instances of the simulator will be setup and experiments will be executed for different protocols in a distributed environment over a local area network cluster using typical laptops and PC available on the location. The interactive demo will provide users the chance to execute their own experiments with the protocols available over a distributed network, find out technical details on how to execute simulations using D-P2P-Sim and enjoy the ease of use and statistical output of the proposed environment.

4. REFERENCES

- [1] S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers. The state of peer-to-peer simulators and simulations. *SIGCOMM Comput. Commun. Rev.*, 37(2):95–98, 2007.