Digital Technologies, 2015, Vol. 1, No. 1, 17-21
Available online at http://pubs.sciepub.com/dt/1/1/4

DOI:10.12691/dt-1-1-4

A Case Study on P2P Computing System for Digital Network and Different Simulation Tools

Anwar Mohd. Mansuri*

Computer Science & Engineering, MIT Mandsaur, Mandsaur, India
*Corresponding author: anwar.iter@gmail.com

Received April 04, 2015; Revised May 28, 2015; Accepted May 31, 2015

Abstract Computer network technologies have been growing explosively and the study in computer networks is being challenging task system participants want to get the final result, In this research work central node may become overloaded, especially if many nodes request the result at the same time. This work present a proposal to build a high throughput computing system totally based upon the Peer-to-Peer (P2P) paradigm. In this work discuss the general characteristics of P2P systems this way we avoid overloading the central node, as well as network congestion we assume that the computing system works on the top of an overlay network. Here present a complete description of the P2P computing system shows a classification of the tools used in communications networks. And we discuss how different simulation tools used in different area like educational research, commercial research is done on P2P system.

Keywords: *network, P2P, NS, distributed system, computing systems, simulation tools*

Cite This Article: Anwar Mohd. Mansuri, "A Case Study on P2P Computing System for Digital Network and Different Simulation Tools." *Digital Technologies*, vol. 1, no. 1 (2015): 17-21. doi: 10.12691/dt-1-1-4.

1. Introduction

Distributed computing systems play a very significant role in today's academic and business world. These systems are applied to compute tasks requiring huge computation power, which is not available on a single machine. They are mainly divided into two categories: grid computing systems and Peer-to-Peer computing systems. Grids are constituted by organizations and institutions and contain a small number of machines connected by a computer network of high capacity. Grids may share many kinds of resources: computing power, disk space, data, sensors, etc. Peer-to-Peer (P2P) systems, initially designed for file sharing are gaining popularity in industry and academia as a powerful mechanism to support other types of applications. Some proposals are massive file storage systems, name indexing and voice calls over IP. The computational science & engineering community is paying attention to P2P systems as the foundation of large computational resources. Communications protocols and link technologies, traffic flows and routing algorithms. One of the obvious approaches to deal with complexity is the use of simulation techniques. The network design process is a challenging task, requiring designers to balance user performance expectations with costs and capacities. Peer-to-peer systems are distributed systems in which there is neither a central control point, nor a hierarchical structure among its members. In a P2P system, all Nodes in the system have the same role, and is interconnected using some kind of network defining an application-level virtual network, also called overlay. Nodes communicate using this overlay, in order to find information, share resources or communicate human users. Network analysis, architecture and the design process have been considered art, combining an individual's particular rules on evaluating and choosing network technologies together with knowledge about how technologies, services and protocols can be meaningfully combined. P2P systems do not interconnect well defined groups using highly reliable networks. They are based on large numbers of unreliable users using unreliable network connections. Even under these limitations, they manage to provide some interesting services and features such as scalability, fault tolerance, efficient information search, highly redundant storage, persistence or anonymity. Researchers typically conduct the simulation using only one simulation package. Different tools show different results with the same simulation models. Peer-to-Peer (P2P) computing systems have emerged in recent years. These systems are built using many private machines, which are most often home computers (PC or Macintosh) or even gaming consoles. The user installs special software on her/his machine and registers into a selected computing project. Then, she/he receives data chunks to compute and sends the results back to the central node, where partial results are combined into the final result. Network connections used in P2P computing systems are regular home access links such as DSL or cable. This approach is much simpler than grids, since the only requirement is to provide suitable software. The P2P computing approach allows for unreliability of participant they may freely join and leave the computing system, which is not used in grid systems.

There is a wide variation in their results. It is very hard to assume which tool's results are accurate.

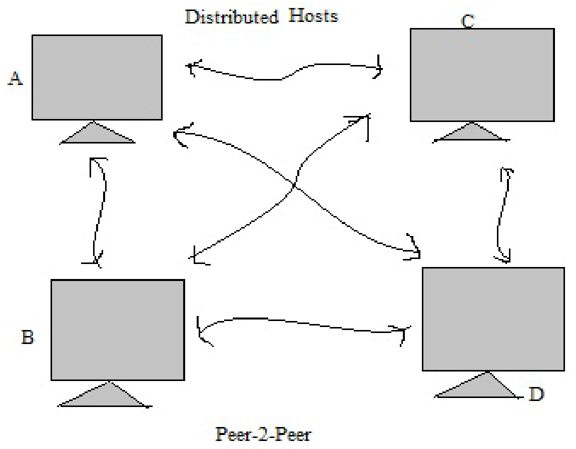

Figure 1. Basic P-2-P Structure

P2P computing systems may also use this model, but as home users contribute their resources, they may also want to participate in the results. This entails the problem of distributing the complete result to each participant. In the case when the result is combined at one central node, a huge number of participants interested in the results and requesting it from one central machine may cause a server overload or even denial of service. The system consists of many machines connected into one logical structure. It takes a computational task as the input, which is then computed by participants. As delivery of all results to each participant introduces significant network traffic, it is essential to provide effective distribution algorithms.

2. Distributed Computing and P2P Network

A distributed computing system is able to perform data computation and distribution of results at the same time. There are two major types of distributed computing systems: grids and Peer-to-Peer (P2P) computing systems. The input task is divided into blocks, which are then sent to system participants that offer their resources in order to perform calculations. The obtained results show the system performance expressed by the operation cost for various types of network flows: unicast, any cast and Peer-to-Peer. Moreover, the simulations prove that our computing system provides about 66% lower cost compared to a centralized computing system. Network is a complex mix of applications, communications protocols and link technologies, traffic flows and routing algorithms. It is immensely complex. Network design process is a challenging task, requiring designers to balance user performance Expectations with costs and capacities. One of the obvious approaches to deal with complexity is the use of modeling and simulation (MS) techniques. The network design process is a challenging task, requiring designers to balance user performance expectations with costs and capacities. External factors, such as government policies and regulations, the competitive situation, available technological services and products are adding complexity to the design process. And at present and upcoming future the data is more and more so security of the data in entire cluster is hike. So when you creating the cluster for the particular geographically area the challenges in security of network as well as congestion is hike. Peer-to-peer systems are distributed systems in which there is neither a central control point, nor a hierarchical structure among its members. The user installs special software on her/his machine and registers into a selected computing project. Then, she/he receives data chunks to compute and sends the results back to the central node, where partial results are combined into the final result. Network connections used in P2P computing systems are regular home access links such as DSL or cable. This approach is much simpler than grids, since the only requirement is to provide suitable software. The P2P computing approach allows for unreliability of participants— they may freely join and leave the computing system, which is not used in grid systems. In a P2P system, all nodes in the system have the same role, and are interconnected using some kind of network (usually, Internet), defining an application-level virtual network, also called overlay. Nodes communicate using this overlay, in order to find information, share resources or communicate human users.

3. Architecture for P2P Computing System

Let us now describe the details of our system architecture. Node requests the source block from the agent when it has free computation resources available. The agent responds with the source block if available or signals that no more source blocks are available for computation.

It contains two types of elements:

- Nodes- regular machines that do the computation and exchange results between each other.
- Agent- A central element, which assigns source blocks to nodes. It is also used as a database including the location of all results, what is similar to the idea of the agent node in the Bit Torrent protocol

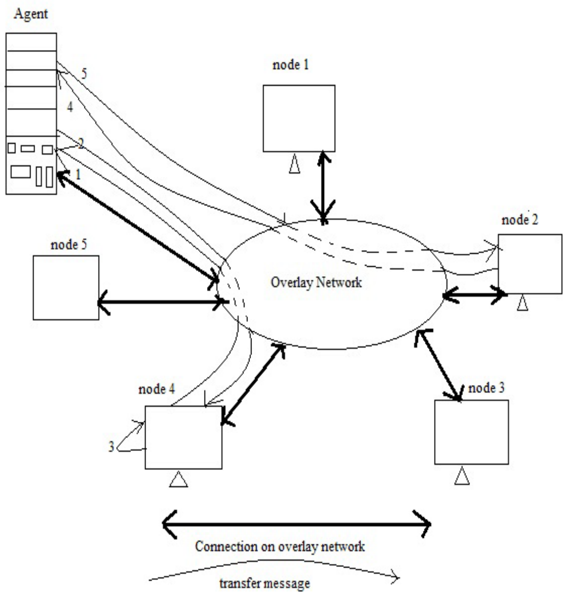

Figure 2. Operation simulation result

The agent stores information about all participants and does not provide information about result locations for nodes, which are not present on the participants' list. This way the system protects itself against unfairness-every node has to contribute to the system in order to obtain final output results. A node that wants to get the result, which was computed by another node, sends a location request to the agent in step 4. Then, the agent responds with known locations step 5. The node selects one of them according to the selection policy. To make the agent locations' list complete, the node sends an update to the agent every time it acquires a new block available to send. This happens in two cases: the node has finished computing the source block or the node has finished downloading the result block from another node. Nodes and the tracker may interact with each other by sending signaling messages. We distinguish the following kinds of messages in the system:

- Source block request- is sent to the tracker from a node, which wants to obtain a new source block for computing.
- Tracker update-is sent to the tracker from a node that starts to possess a new result block available for download, this message updates the tracker's location list.
- Block location request-is sent to the tracker from a node, which wants to receive a current list of nodes that posses result blocks.
- Download request-is sent from a node that misses a result block to another node, which has the requested result block available for download. It must be confirmed by a download acknowledgement message
- Download acknowledgement-this message is a positive reply to the download request message.
- Block location list-is sent from the tracker to a node, requesting the current list of block locations; this is a reply to the block location request message.

The information is updated by the peer according to completed operations (both processing and downloading of blocks). The other data is monitored and provided by the tracker.

4. Analytical Tools for Network Design

The analytical tool typically formulates the network planning problems as optimization models were given cost functions are minimized (maximized) under a set of constraints. Normally a single model tries to capture many relevant aspects of the problem. When an optimal solution to the network model is found, the values of the decision variables can be used to decide the optimal action to be taken. There are seven files that are handled by the Delite tool for each network design. Links between nodes can be generated using a few design algorithms. Thus, various designs may have different costs, delays, reliability, and average number of hops etc. educational and practical wide area network (WAN) design tool, which can produce network designs of limited size using a set of the embedded network design algorithms. At each node there is a scheduling process in charge of searching jobs to run.

This tool is helpful for the users interested for node cost analysis and can be used with integration with other network design and simulation tool.

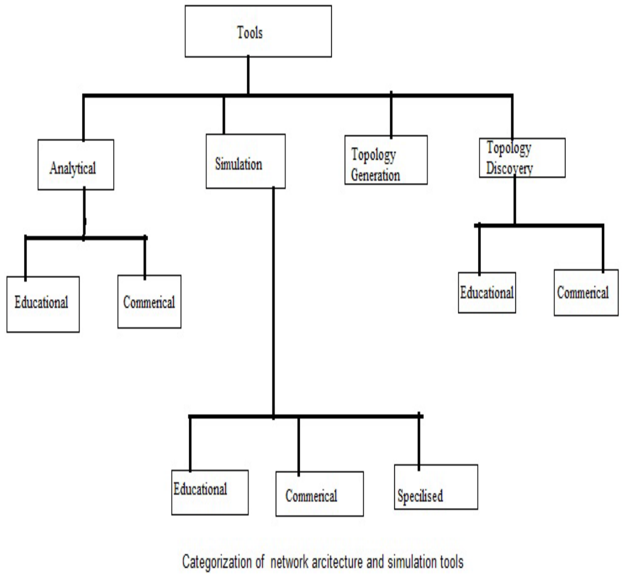

Figure 3. Architecture of network tools

5. Network Simulation Tools

Simulation is the discipline of designing a model of an actual or theoretical or physical system and manipulating the modelb in such a way that it operates on time or place to compress it, thus enabling one to practice the interaction. Many unimportant details can be abstracted away and simulations can be completely repeatable. In a simulation, a mathematical/logical model is numerically evaluated over the period of interest and performance measures are estimated from model-generated data. Simulations are complementary to analysis, not only by providing a check on the assumptions of the models and on the correctness of the analysis, but by allowing exploration of complicated scenarios that would be either difficult or impossible to analyze. Simulation plays a vital role in helping researchers to develop intuition. Simulation analysis is applicable to systems of almost any level of complexity.

- NS-2: (Network Simulator), a VINT (Virtual Inter-Network Testbed) project from U.C. Berkeley/LBL/Xerox PARC, is a discrete event simulator targeted at network research, which provides substantial support for simulation of TCP, routing and multicast protocols. The simulator is using a Tcl/Otcl (Tool Command Language/Object Oriented Tcl) as a command and configuration interface. There are four types of files related to ns-2 simulator.
- Network Workbench: is a discrete event network simulator developed for the academic investigation of Internet protocol. It contains a complete protocol stack, abstracted from the Internet stack and a set of exercise that focus on critical protocol algorithms in the Internet stack. Network Workbench has several version, which are: version 0(1994), version 1(1994), version 2(1997), and version 3(1998). This tool supports several network topology, DLC error control, CSMA/CD collision back off, optimal route computation, reliable transport, multicast, and LAN/MAN integration. It supports fiber distributed data interface (FDDI), open shortest-path-first

(OSPF). Workbench is abstracted from the internet (TCP/IP) stack. It contains five layers (application, transport, network, data link control, and physical).

- Netsim (M.I.T.'s Network Simulator): is a single process discrete event simulator used for the investigation of many aspects of local area network (LAN). Netsim has three main goals: (1) flexibility of experiment of specifying the network and traffic; (2) simulate the accurate behavior of Ethernet; and (3) has features that make running sequences of related experiments easier. The experimental data of Netsim simulation is stored in an experiment description file which contains: the layout of the network to be simulated, the traffic generation behavior of the station on the network, information about repetitive of each run and about the sequence of parameters to be used for a series of experiments, etc.
- MaRS (Maryland Routing Simulator): is a discrete event simulator proving a flexible platform for the evaluation and comparison of network routing algorithms. MaRS allows physical network, routing algorithm and traffic sources. MaRS is structured in two parts: a simulation engine, which manages the event list and user interface and a set of components for modeling the network configuration and handling certain simulation functions. MaRS has been used to evaluate and compare several next-hop routing algorithms
- Pdns (Parallel/Distributed ns): simulator consists of extensions to the widely used and publicly available ns network simulator. The extensions allow many existing ns simulations to be run in a distributed environment with minimal changes. The pdns implementation also takes advantages of the large body of existing models found in ns and uses those without modification. Also includes in the pdns extensions is novel packet routing method called NIx-vector that allow routing decisions without the necessity of routing tables, resulting in substantial memory saving.

Simulation models for a number of different random number generators are provided, including exponential, Pareto, normal, uniform, empirical and constant.

6. Results

Below table shows summary of the small and large-scale network simulation tools. There are not too many works which are focused on the overview of topology generation tools and if they do, it mostly relates to the algorithms on which tools are based, currently, networks are monitored, maintained and diagnosed using discovery tools that rely on network protocols like internet control message protocol (ICMP) and simple network management protocol (SNMP).

Table 1. Summery of network simulation tools

Tools	Usages	Target network model	Outputs	Stored data format	Platform
Ns-2	Education research	WAN	Simulation of several protocol	Text file (ns/tcl format)	Unix
Network Workbench	Education research	LAN/MAN	Simulation of network summary of interlayer statistics	Text file	Window/Linux
NetSim	Educational	LAN (Ethernet	Simulation,histogram of packet queuing delay, offered load, overall throughput	Text file	Windows
MaRS	Educational	WAN, link-state and distance-vector routing SPF, ExBF, Segal routing	Simulation of application traffic-ftp, telnet, simple, various meters	Not knownrequire further investigation	Unix/Windows
OPNET	Commercial	LAN Satellite, radio modelling	Simulation of several protocols, delay, utilisation etc. performance analysis	Text file (C/C++ format)	XWindows
GloMoSim	Specialised	Wireless network	Simulation of wireless network	Text file	Windows Unix

These tools support network discovery and provide the means to remotely query and control network devices, such as routers and hosts. It would be useful, if a model could be analyzed using different tools simultaneously. One tool can be complemented by other tools for their distinctive features. In this respect, there is a need for integrating analytical, simulation, topology discovery and generation tools altogether. As future work, we propose to extend the computation system with new constraints; like varied number of nodes and replica statuses changed during the simulation. Other future directions are new decision policies that can be evaluated by using simulator.

7. Conclusions

In this paper, we have studied about P2P networks. We have explained the main characteristics of design, and have implemented a prototype based on the P2P storage system. In this paper, a survey of the research work on network MS tools with a classification is presented. The system can use three kinds of network flows for data distribution unicast, Peer-to-Peer and anycast. We plan to enhance the system by adding tools to simplify its utilization, allowing users to submit not only individual jobs but also more sophisticated workflows.

References

[1] Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, A. Silbershatz, Topology discovery in heterogenous IP networks: the NetInventory system, IEEE/ ACM Transactions on Networking 12 (3) (2004) 401-414.

[2] Stuart Kurkowski, Tracy Camp, Michael Colagross, MANET simulation studies: the incredibles, ACM SIGMOBILE MCCR 9 (4) (2005) 50-61.

[3] M.A. Rahman, A. Pakstas, F.Z. Wang, Network modelling and simulation tools, in: Proceedings of the Eighth EPSRC Annual Postgraduate Symposium on the Convergence of telecommunications, Networking and Broadcasting (EPSRC PGNet 2007), Liverpool John Moores University, June 28-29, Liverpool, UK, 2007.

[4] Humayun Akhtar, An Overview of some network modeling, simulation & performance analysis tools, in: Proceedings of the Second IEEE Symposium on Computers and Communications (ISCC'97), 1997, pp. 344-348.

[5] Jukka K. Nurminen, Models and algorithms for network planning tools – practical experiences, System Analysis Laboratory Research Reports, Helsinki University of Technology, May 2003.

[6] R.S. Cahn, Wide Area Network Design: Concepts and Tools for Optimization, Morgan Kaufman Publishers, San Francisco, 1989.

[7] Ma H. (Bob), Network Design Tool, Computer Science and Engineering, Washington University in St. Louis, MO, 1989.

[8] D.W. Bachmann, M.E. Segal, M.D. Srinivasan, T.J. Teorey, NetMod: A Design Tool for Large–Scale Heterogeneous Campus Network. Center for Information Technology Integration (CITI), University of Michigan, IEEE Journal on Selected Areas in Communications, January 1991, vol. 9(1),pp-15-24.

[9] Davis C.M. Wood, Sean S. Coleman, Michael F. Schwartz, Fremont: a system for discovering network characteristics and problems, in: Proceedings of the USENIX Winter Conference, 25-29 January, San Diego, CA, 1993, pp. 335-348.

[10] R. Samanta, T. Funkhouser, K. Li, Parallel rendering with K-way replication, in: Proceedings of the IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics, 2001, pp. 75-84.

[11] D. Hughes, J. Walkerdine, Distributed video encoding over a Peer-to-Peer network, in: The Proceedings of PREP 2005, vol. 1, 2005.

[12] S. Draves, The interpretation of dreams, an explanation of the electric sheep distributed screen-saver.

[13] A. Legout, G. Urvoy-Keller, P. Michiardi, Understanding BitTorrent: an experimental perspective, Technical Report, INRIA-00000156, 2005.

[14] B. Godfrey, K. Lakshminarayanan, S. Surana, R.M. Karp, I. Stoica, Load balancing in dynamic structured p2p systems, in: INFOCOM.

[15] I.T. Foster, C. Kesselman, The globus project: a status report, in: Heterogeneous Computing Workshop, pp. 4-18.

[16] R. Gupta, V. Sekhri, A.K. Somani, Compup2p: an architecture for internet computing using peer-to-peer networks, IEEE Transactions on Parallel and Distributed Systems 17 (2006) 1306-1320.

[17] G. Chmaj, K. Walkowiak, A p2p computing system for overlay networks, Future Generation Computer Systems 29 (1) (2013) 242-249.

[18] J.-C. Charr, R. Couturier, D. Laiymani, Jacep2p-v2: a fully decentralized and fault tolerant environment for executing parallel iterative asynchronous applications on volatile distributed architectures, Future Generation Computer Systems 27 (2011) 606-613.

[19] Giovanni Vigna, Fredrik Valeur, Jingyu Zhou, Richard A. Kemmerer, Composable tool for network discovery and security analysis, in: 18th Annual Computer Security Application Conference, 09–13 December, San Diego CA, 2002, pp. 14-24.

[20] M.A. Rahman, A. Pakstas, F.Z. Wang, An approach to integration of network design and simulation tools, in: Proceedings of the Eighth International IEEE Conference on Telecommunications (ConTEL 2005), Zagreb, Croatia, June 15–17, 2005, pp. 173–180

[21] J.M. Pullen, The network workbench: network simulation software for academic investigation of internet concepts, Computer Networks. Elsevier Science Ltd, March 2000, vol. 3(3), 2000, pp. 365-378.

[22] B.L. Barnett III, An ethernet simulator for undergraduate networking, ACM SIGCSE Bulletin, in: Proceedings of the 24th SIGCSE Technical Symposium on Computer Science Education, vol. 25(1), Indianapolis, IN, USA, 1993, pp. 145-150.

[23] D. Griffin, Traffic engineering for quality of services in the internet, at large-scale, Technical Report, CEC Deliverable Number: 201/UCL/bl, TEQUILA Consortium, 2000.

[24] C. Alaettinoglu, I. Dussa-Zieger, A.U. Matta, Shankar, O. Gudmundsson, Introducing MaRS, a Routing Testbed, ACM SIGCOMM, pp. 95-96.

[25] G.F. Riley, M.H. Ammar, Simulation large networks: how big is big enough? in: Proceedings of First International Conference on Grand Challenge for Modeling and Simulation, San Antonio, TX, 2002, pp. 39-45.

[26] M.A. Rahman, A. Pakstas, F.Z. Wang, An approach to integration of network design and simulation tools, in: Proceedings of the Eighth International IEEE Conference on Telecommunications (ConTEL 2005), Zagreb, Croatia, June 15-17, 2005, pp. 173-180.

[27] Y. Breitbart, M. Garofalakis, B. Jai, C. Martin, R. Rastogi, A. Silbershatz, Topology discovery in heterogenous IP networks: the NetInventory system, IEEE/ACM Transactions on Networking 12 (3) (2004) 401-414.

[28] S. Park, A. Savvides, M.B. Srivastava, SensorSim: a simulation framework for sensor networks, in: International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, 2000, pp. 104-111.

[29] S. Penz, Wireless multicast support for the NS-2 emulation environment, in: Proceedings of the Fifteenth IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2007, pp. 267-273.

[30] Giovanni Vigna, Fredrik Valeur, Jingyu Zhou, Richard A. Kemmerer, Composable tool for network discovery and security analysis, in: 18th Annual Computer Security Application Conference, 09–13 December, San Diego CA, 2002, pp. 14-24.