Load Balancing in P2P Networks using Random Walk Algorithm

A. S. Syed Navaz[1], A. S. Syed Fiaz[2]

[1]Assistant Professor, Department of Computer Science, Muthayammal College of Arts & Science, Namakkal, Tamilnadu, India

[2]Department of Computer Science & Engineering, Dhirajlal Gandhi College of Technology, Salem, Tamilnadu, India

Abstract: ***Previous analytical studies of unstructured P2P resilience have assumed exponential user lifetimes and only considered age-independent neighbor replacement. In this system, the limitations are overcome by introducing a general node-isolation model for heavy-tailed user lifetimes and arbitrary neighbor-selection algorithms. Using this model, analysis of two age-based neighbor-selection strategies takes place and they significantly improve the residual lifetimes of chosen users, which dramatically reduce the probability of user isolation and graph partitioning compared with uniform selection of neighbors'. In fact, the second strategy based on random walks on age-proportional graphs demonstrates that, for lifetimes with infinite variance, the system monotonically increases its resilience as its age and size grow. Specifically, it shows the probability of isolation converges to zero as these two metrics tend to infinity. This system is finished with simulations in finite-size graphs that demonstrate the effect of this result in practice.***

Keywords: Peer to Peer, Node-Isolation, age-proportional graph.

1. Introduction

Resilience of P2P networks under random user arrival and departure (i.e., churn) has recently become an active research area. One of the primary metrics of resilience is graph disconnection during which a P2P network partitions into several nontrivial sub graphs and starts to offer limited service to its users. However, as shown in, most partitioning events in well-connected P2P networks are single-node isolations, which occur when the immediate neighbours of a node fail before is able to detect their departure and then replace them with other alive users. For such networks, node isolation analysis has become the primary method for quantifying network resilience in the presence of user churn.

In fact, the second strategy based on random walks on age-proportional graphs demonstrates that, for lifetimes with infinite variance, the system monotonically increases its resilience as its age and size grow. Specifically, we show that the probability of isolation converges to zero as these two metrics tend to infinity. We finish the paper with simulations in finite-size graphs that demonstrate the effect of this result in practice.

It consider heavy-tailed user lifetime for improving residual lifetime of chosen users and reducing probability of user isolation and graph partitioning from network. Node isolation analysis has become the primary method for quantifying network resilience in the presence of user churn. A joining node randomly selects alive users from the system and chooses the user with the maximal age. Random walk algorithms have been used to build unstructured P2P systems and replace failed links with new ones. Introduce a new-Neighbor selection strategy that is based on random walks over weighted directed graphs to identify the isolated node. Calculating the age for each node based on sum of weight of in degrees of a node.

1.1 Peer-to-Peer Networks

Peer-to-peer networking is the utilization of the relatively powerful computers (personal computers) that exist at the edge of the Internet for more than just client-based computing tasks. The modern personal computer (PC) has a very fast processor, vast memory, and a large hard disk, none of which are being fully utilized when performing common computing tasks such as e-mail and Web browsing. The modern PC can easily act as both a client and server (a peer) for many types of applications.

The typical computing model for many applications is a client/server model. A server computer typically has vast resources and responds to requests for resources and data from client computers. Client computers initiate requests for resources or data from server computers. A good example of the client/server model of computing is Web browsing. Web servers on the Internet are typically high-end dedicated server computers with very fast processors (or multiple processors) and huge hard disk arrays. The Web server stores all of the content associated with a Web site (HTML files, graphics, audio and video files, etc.) and listens for incoming requests to view the information on a particular Web page. When a page is requested, the Web server sends the page and its associated files to the requesting client.

1.1.1Advantages

- Content and resources can be shared from both the center and the edge of the network. In client/server networking, content and resources are typically shared from only the center of the network.
- A network of peers is easily scaled and more reliable than a single server. A single server is subject to a single point of failure or can be a bottleneck in times of high network utilization.
- A network of peers can share its processor, consolidating computing resources for distributed computing tasks, rather than relying on a single computer, such as a supercomputer.

- Shared resources of peer computers can be directly accessed. Rather than sharing a file stored on a central server, a peer can share the file directly from its local storage.

1.1.2 P2P Network solves the following problems

- Allows the processing resources of edge computers to be utilized for distributed computing tasks.
- Allows local resources to be shared directly, without the need for intermediate servers.
- Allows efficient multipoint communication without having to rely on IP multicast infrastructure.

1.2 Java swing

Swing components facilitate efficient graphical user interface (GUI) development. These components are a collection of light weight visual components. Swing components contain a replacement for the heavyweight AWT components as well as complex user-interface components such as trees and tables. Swing is a set of classes that provides more powerful and flexible components than are possible with the AWT.

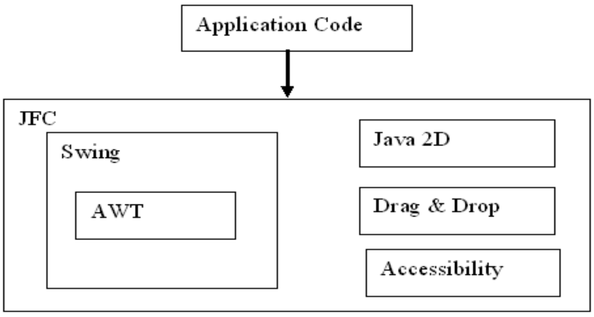

Figure 1: Swing Architecture

In addition to that the familiar components such as buttons, check box and labels swings supplies several exciting additions including tabbed panes, scroll panes, trees and tables. Even familiar components such as buttons have more capabilities in swing.

For example a button may have both an image and text string associated with it. Also the image can be changed as the state of button changes. Unlike AWT components swing components are not implemented by platform specific code instead they are return entirely in JAVA and, therefore, are platform-independent. The term lightweight is used to describe such elements. The number of classes and interfaces in the swing packages is substantial. The Swing architecture is shown in the figure given blow:

2. The Swing Component Classes

Table 1: Swing Component Classes

Class	Description
Abstract Button	Abstract super class for Swing Buttons
Button Group	Encapsulates a mutually exclusive set of Buttons
ImageIcon	Encapsulates an Icon
JApplet	The Swing version of Applet
Jbutton	The Swing Push Button Class
JCheckBox	The Swing CheckBox class
JComboBOx	Encapsulates a combobox
Jlabel	The swing version of a Label
JRadioButton	The Swing version of a RadioButton
JScrollPane	Encapsulates a scrollabel window
JTabbedPane	Encapsulates a Tabbed window
Jtable JTextField	Encapsulates a Table-based control The swing version of a text-field
Jtree	Encapsulates a Tree-based control

1.2.1 Advantages of Swings:

- Wide variety of Components
- Pluggable Look and Feel
- MVC Architecture
- Keystroke Handling
- Action Objects
- Nested containers
- Customized Dialogs
- Compound Borders
- Standard Dialog Classes
- Structured Table and Tree Components
- Powerful Text Manipulation
- Generic Undo Capabilities
- Accessibility Support

3. Problem Definition

2.1 Existing System

In existing system, unstructured p2p network consider only exponential (mean time to failure) user lifetime and age-independent neighbor replacement. Unstructured P2P networks are single-node isolations, which occur when the immediate neighbors' of a node fail before a node is able to detect their departure and then replace them with other alive users. Probability of user isolation is high in existing system and the users cannot communicate with the isolated nodes. When node gets isolated from the network, the links between the nodes get failure. So, graph gets disconnected. Newly joining node unable to detect the heavy-tailed user lifetime to avoid its probability of isolation from the network.

2.1.1Disadvantages

Node isolated from network. So, communication problem occurs. Neighbors node unable to detect the failure previously. Graph- Disconnection between the networks. Resilience of peer to peer networks under random user arrival and departure is difficult. New node cannot be able to identify the heavy-tailed node in the network. Probability of user isolation and graph disconnection is high.

2.2 Proposed System

In proposed system, we consider heavy-tailed user lifetime for improving residual lifetime of chosen users and reducing probability of user isolation and graph partitioning from network. Node isolation analysis has become the primary method for quantifying network resilience in the presence of user churn. A joining node randomly selects alive users from the system and chooses the user with the maximal age. Random walk algorithms have been used to build unstructured P2P systems and replace failed links with new ones. Introduce a new-neighbor selection strategy that is based on random walks over weighted directed graphs to identify the isolated node. Calculating the age for each node based on sum of weight of in degrees of a node.

2.2.1Advantage

Reducing probability of user isolation. Avoids graph-partitioning from network. Overcome Communication problem. Resilience of peer to peer networks for random user arrival and departure is done. Probability of user isolation and graph disconnection is low.

4. P2P Description

3.1 Node information and topology construction

In this module node information is for initializing the number of nodes, giving names to those nodes, initializing the port numbers for a particular node and provision of host name. In node information, node name and port number must be unique for each node. Host name must be given to run nodes in multiple systems. For topology construction we provide the links for the initialized nodes. We also provide cost to the various links.

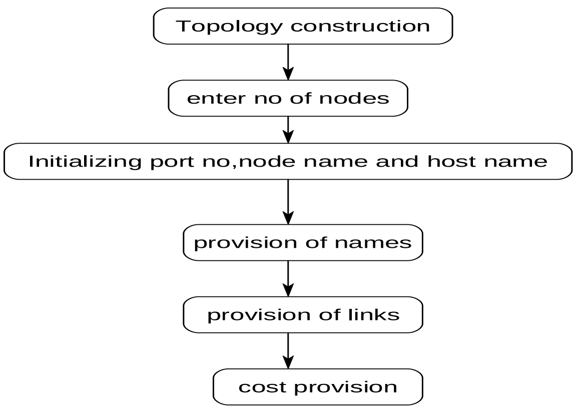

Figure 2: Node Information and Topology construction

We check there is no multiple links for same set of nodes. Cost specification is given to all nodes. Each node in network has capable of sending and receiving information through their unique port number. Based on node information and topology construction, each node can calculate the available path and best path for message transmission.

3.2 Node Isolation

A node which does not have any connection with any other in a network then that node is said to be isolate from network. In unstructured topology, nodes can randomly arrive and departure. When a node get departure, other nodes link gets disconnected. Network resilience is graph disconnection during which a P2P network partitions into several nontrivial sub graphs and starts to offer limited service to its users.

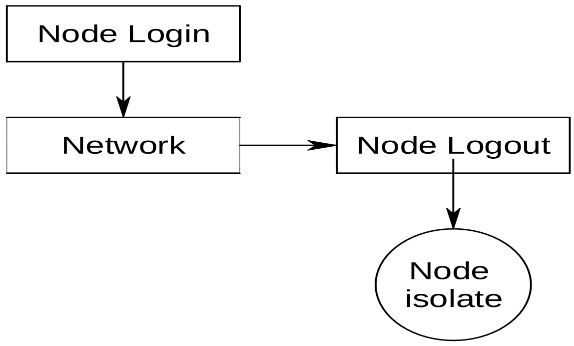

Figure 3: Node Isolation

Due to graph disconnection, isolated node cannot be able to communicate to other nodes in a network. Isolated node cannot make any communication with any other node in network. Message transmission is impossible with isolated node.

3.3 Random Walk Algorithm

A new-neighbor selection strategy that is based on random walks over weighted directed graphs. In this module, node age is calculated by sum of weight of in degrees of each node and age is provided to each node. Age=Σ W_i

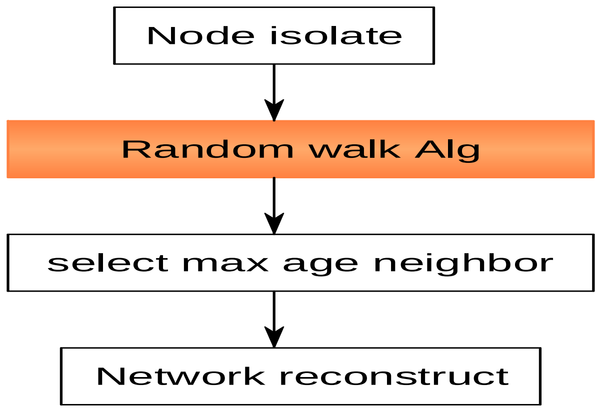

Figure 4: Random walk Algorithm

When nodes get isolated, it selects the maximum node age and starts its offer to its users. Each login node also chooses maximum node age and joins in network for providing its services to increase its residual lifetime. Random walk on age-proportional graphs demonstrates that, for lifetimes with infinite variance, the system monotonically increases its resilience as its age and size grow. Using Random walk algorithm, P2P networks are single-node isolations, can be identified by immediate neighbors of a node to detect their departure and then replace them with other alive users.

Recently, random walks have been proposed as primary algorithmic ingredients in protocols addressing searching and topology maintenance of unstructured P2P networks. In particular:

- Following extensive experimentation, report that searching by simulating random walks has superior performance as compared to the standard approach of searching by flooding. They attribute the improved performance of random walks to their adaptively in termination conditions and hence granularity in coverage of the search space (in flooding, increasing the TTL by 1 may increase the space coverage exponentially).
- Give a distributed algorithm for constructing and maintaining unstructured topologies with very strong connectivity properties, namely constant degree and constant expansion, with O (log n) overhead per addition of a peer, where n is the number of peers. At a very high level, when a new peer arrives, their protocol simulates a random walk on the existing overlay topology which, after O (log n) steps, reaches a nearly uniformly random existing peer to which the new peer attaches.

3.4 Message Transmission

In this module, message gets transmitted between the nodes .when node is isolated it selects random walk algorithm and chooses its neighbor and reconstruct the topology and message is transmitted in alternate path. The source node chooses the destination and passes the message using the method of random walk algorithm to avoid graph disconnection for sending its message in the best path available. Once the client completes its message and sends the message, the client gets the knowledge about the available paths and it also gets the information about the best path and the details regarding the particular path.

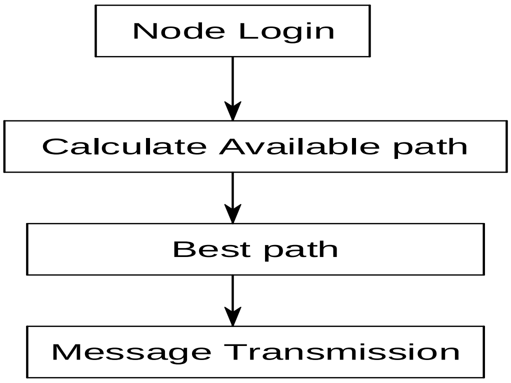

Figure 5: Message Transmission

When the destination system is isolated, it can receive message as isolated node. Then source can use random walk algorithm and resend the message using alternate path. Once the message reached the destination, source can receive the acknowledgement message from the destination.

5. System Implementation

Implementation is the stage in the paper where the theoretical design is turned into a working system and is giving confidence on the new system for the users, which it will work efficiently and effectively. It involves careful planning, investigation of the current System and its constraints on implementation, design of methods to achieve the change over, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

An implementation co-ordination committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Implementation is the final and important phase, the most critical stage in achieving a successful new system and in giving the users confidence. That the new system will work be effective .The system can be implemented only after through testing is done and if it found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

A Random walk, sometimes denoted RW, is a mathematical formalization of a trajectory that consists of taking successive random steps. The results of random walk analysis have been applied to computer science, physics, ecology, economics, and a number of other fields as a fundamental model for random processes in time. For example, the path traced by a molecule as it travels in a liquid or a gas, the search path of a foraging animal, the price of a fluctuating stock and the financial status of a gambler can all be modeled as random walks. The term random walk was first introduced by Karl Pearson in 1905.

Various different types of random walks are of interest. Often, random walks are assumed to be Markov chains or Markov Process, but other, more complicated walks are also of interest. Some random walks are on graphs, others on the line, in the plane, or in higher dimensions, while some random walks are on groups. Random walks also vary with regard to the time parameter. Often, the walk is in discrete time, and indexed by the natural numbers, as in $X_0, X_1, X_2, \ldots$. However, some walks take their steps at random times, and in that case the position X_t is defined for the continuum of times $t \geq 0$. Specific cases or limits of random walks include the drunkard's walk and Levy flight.

Random walks are related to the diffusion models and are a fundamental topic in discussions of Markov Process. Several properties of random walks, including dispersal distributions, first-passage times and encounter rates, have been extensively studied.

6. Conclusion

In this technique, randomized discretization and path delay discretization, to design fast algorithms for computing constrained shortest paths. While the previous approaches (RTF and RTC) build up the discretization error along a path, the new techniques either make the link errors to cancel out each other along the path or treat the path delay as a whole for discretization, which results in much smaller errors. The algorithms based on these techniques run much faster than the best existing algorithm that solves the e–approximation.

Future work includes derivation of residual lifetime distributions in finite systems under age-proportional neighbor selection and analysis of the limiting distribution of neighbor residual lifetimes under max-age selection as the number of sampled users.

References

[1] Feldmann and W. Whitt, "Fitting mixtures of exponentials to longtailed distributions to analyze network performance models," Perform. Eval., vol. 31, no. 3–4, pp. 245–279, Jan. 1998.
[2] B.-G. Chun, B. Zhao, and J. Kubiatowicz, "Impact of neighbor selection on performance and resilience of structured P2P networks," in Proc. IPTPS, Feb. 2005, pp. 264–274.
[3] Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks," in Proc. IEEE INFOCOM, Mar. 2004, pp. 120–130.
[4] E. B. Dynkin, "Some limit theorems for sums of independent random variables with infinite mathematical expectations," Sel. Transl. Math. Statist. Probabil., vol. 1, pp. 171–189, 1961.
[5] Zhongmei Yao and Derek Leonard "Node Isolation Model and Age based neighbor selection in unstructured p2p networks" IEEE/ACM Transactions on Networking, vol. 17, no. 1, February 2009.

Author Profile

A. S. Syed Navaz received M.Sc in Information Technology from K.S.Rangasamy College of Technology, Anna University Coimbatore, M.Phil in Computer Science from Prist University, Thanjavur, and M.C.A from Periyar University, Salem and Pursuing Ph.D in the area of Wireless Sensor Networks. I researched and published in International journals and working as Editorial Board Member & Reviewer for International journals also Member of IACSIT, UACEE, IAENG, IDES, CSTA, SDIWC, SCIEI, IAOE, ISOC, WAYS, NYSCATE & ITEEA. Currently I am working as an Assistant Professor in the Department of Computer Science at Muthayammal College of Arts & Science, Namakkal, India. My Research areas are Wireless Sensor Networks, Mobile Computing & Image Processing.

A. S. Syed Fiaz received his ME (CSE) from Sona College of Technology, Anna University Chennai and BE (CSE) from Sona College of Technology, Anna University Chennai. He has researched and published in International journals and working as Editor Board Member & Reviewer for International journals. Currently he is working as an Assistant Professor in the Department of Computer Science and Engineering at Dhirajlal Gandhi College of Technology, Salem. His areas of interest are Cloud Computing, Computer Networks.