P2P Routing in the Metascheduler Architecture to provide QoS in Cloud Computing

DIONISIO M. L. FILHO, MAYCON L. M. PEIXOTO, MARCOS J. SANTANA, REGINA H. C. SANTANA

e-mail: dionisio@icmc.usp.br, maycon@icmc.usp.br, mjs@icmc.usp.br, rcs@icmc.usp.br

Universidade de São Paulo (USP)
Instituto de Ciências Matemáticas e de Computação
Avenida Trabalhador São-carlense, 400 - Centro
CEP: 13566-590 - São Carlos - SP

Abstract. This paper presents an evaluation of different routing algorithms employed in a logical layer peer-to-peer (P2P). The adopted model uses a Metascheduler that provides quality of services (QoS) in Cloud Computing. The experiments show the response times and the variation between the results of different tests. The various routing algorithms has given the influence of the number of users and type of requested services and how these factors interact between themselves. Based on specific objectives, the results allow determining the impact of the utilization of P2P approaches (Pastry and Chord) by a metascheduler on the search and discovery of services regarding the way the QoS is performed.

Keywords: Cloud Computing, Performance Evaluation, P2P, Monitoring, QoS.

(Received February 20th, 2013 / Accepted August 29th, 2013)

1 Introduction

Cloud Computing is a computational system that allows the delivery of services over the Internet [3]. In general, it provides the change of physical resources into a cloud. In other words, it is the migration of local servers to a cloud to provide services over the Internet facilitating their access from any device with a web connection [3] [28].

Although cloud computing seems to be a model for distributed computing similar to the client/server systems, different fundamentals must be considered. The biggest difference is that there are multiple servers or data centers spread around the world. The user does not know the exact location of their data and where they are processed. The only certainty is that the user's data are in a "cloud" and can be accessed over the Internet.

Cloud computing can be implemented using different approaches, such as public, private or hybrid clouds and the services can be categorized as infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), and so on [8]. Regardless of the implementation of the cloud, a feature that accompanies its concept is that services are always available in the clouds and can be accessed by any Internet-connected device [5].

In order to process or store data on the Internet, it is necessary to consider different approaches of how data are processed and stored and how the communication between clients and servers is performed. An important aspect to be addressed is how to provide quality of service (QoS) in Cloud Computing. Ensuring QoS is not a trivial task as it depends on different parts of the system, as the scheduling and routing of services. Thus, a part that exerts a significant influence to achieve quality of service is the approaches to facilitate the access to servers.

A way to provide the QoS requirements is to use a metascheduler to manage either resources or services [17]. The metascheduler has a function level to perform the service discovery. In this level, the P2P layer can be adopted, as proposed by Peixoto et. al. [17].

This metascheduler was used in this research to be the base of the P2P scheduling and to improve the relation between the P2P system and the cloud. The concern about communication is based on the location of the computational resources that are in a cloud. The innovation presented here is in the use of a metascheduler to provide an interaction between two different technologies. This paper deals with the evaluation of the approaches used to minimize the response times in the cloud activities and the integration over the P2P systems and the cloud to provide good synergy.

The paper is organized as follows: Section 2 presents related studies and the starting point of this project; Section 3 presents the system considered in the simulation as well as the network topologies used; Section 4 shows the results and the behavior of the experiments; In the Section 5 are found the contributions about this paper. Section 6 draws the conclusions of the authors.

2 Related Work

Cloud computing has drawn plenty of attention from both academic and business areas. It is a model for distributed computing that allows sharing resources and minimizing the cost of management.

Rimal et al. [20] present a taxonomy for cloud computing and its utilities, challenges and prospects. The presented challenges related with cloud are safety, load control and interoperability. The authors emphasize that the cloud computing environment provides a solution that is more cost-effective and scalable and allows fault-tolerance. These points are also raised by Armbrust et al. [3], who define cloud computing as a utility that aims to reduce infrastructure costs and provide more agile ways of application development.

Zhang et al. [28] also define the cloud computing term. The main concept presented is that there exists the possibility of using scheduling approaches of other computational systems [5]. A way to perform scheduling mechanisms is to use metaschedulers approach [5] [25] [26].

Metaschedulers are used to distribute services to servers based on the clients' requirements [5]. They can have scheduler layers that do not know the physical resource of an environment. In other words, there may exist scheduler layers that forward the requests, based on the clients' requirements to the scheduler, which will denote the execution of the services [1] [10] [13] [26] [9].

As presented by Chard [9], the resource discovery based on metaschedulers can use centralized and decentralized forms. The centralized way is normally discouraged once the principal problem observed is the central fail point [1] [10] [13] [2].

The decentralized models are presented as a better option for use in metascheduler approaches [9] [29].

An interesting characteristic of decentralized methods is that the use of P2P is found in the literature, but in synergy with grid computing.

Spence and Harris [22] introduced a modification of P2P Pastry system to be used in a specific scenario. The system is used for all components of the grid. Tam et al. [24] also features a study based on the use of P2P and grid. The systems in both cases are based on Pastry system and grid computing. Cai et al. [24] propose a Chord system to manage all the components of a grid. Meng et al. [27] shows a Chord system to manage the grid components.

Ranjan et al. [19] and Lai et al. [12] present studies related to P2P systems and cloud computing. They focus on the management of all components of a cloud improving the service discovery and the load balance [19]. The service discovery is based on all virtual machines of a cloud [12].

3 System Design

As defined by NIST (National Institute of Standards and Technology) [15] and Zhang et al. [28], a hybrid cloud is composed of both private and public clouds. It is utilized to balance the overload of the private cloud sending part of the services to a public cloud. This occurs when the capacity of the private cloud has been reached and there are still requests to be met.

In this study a hybrid scenario is used for the experiments. The workload (synthetic based on exponential) is sent to a public cloud by a private cloud simulating the overhead of the private cloud. To perform the experiments the cloud is used only as infrastructure and no interactivity is considered.

In this work, the metascheduler handles two types of services: light and heavy. The services consist of lightweight requests with low need processing (such as retrieving web pages or transactions online business). Otherside, has the heavy service based on intensive processing, representing the use of cloud computing by scientific applications, for example, an image rendering.

The metascheduler proceeds a search to find the public cloud with the lowest latency to a private cloud. This scheduling is part of the metascheduling because in the first scheduling the metascheduler does not know the available local resources. In the second scheduling the local resources are managed by the metascheduler to better allocate the resources based on the client's request. The QoS considered here is related to the lowest *response time* because the policies in the P2P layer only consider different latencies.

The selection of the provider to be used is the role of the metascheduler, which can use different policies, such as Pastry and Chord.

Pastry is an algorithm that uses a binary search for the decision-making process. It observes the conditions of the network and sends the user's request to the data center with the lowest latency in communication. The use of binary search follows the recommendations found in [21], which show how a node is identified on the network. Thus, the search is performed as a binary search in a region based on the latency considered. The data center elected is the one with the lowest latency to the client. Figure 1 presents the Pastry search engine used in the simulation scenarios.

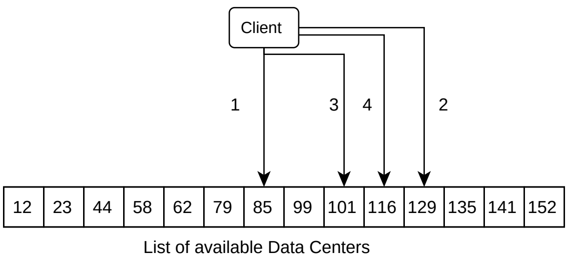

Figure 1: Pastry search engine

Otherside, Chord has a logical composition that forms rings [23]. Its search engine is shown in Figure 2. The search is performed in a specific region and the client looks for the available data center based on the proximity of its region.

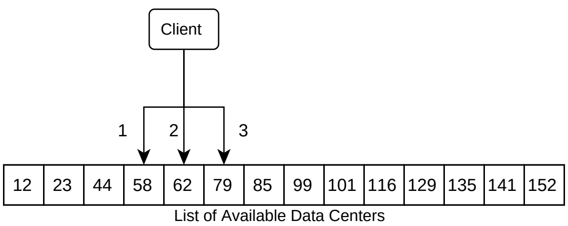

Figure 2: Chord search engine

CloudSim simulator [7] is used for the simulation and implementation of policies, as it allows implementing routing policies. The network standard adopted by CloudSim is Brite [14], which permits the creation of various types of topologies, depending on the application domain addressed. Figures 3 and 4 show the topology of the 30 and 60 users utilized in this study. These numbers of common user represent the low and high overload scenarios.

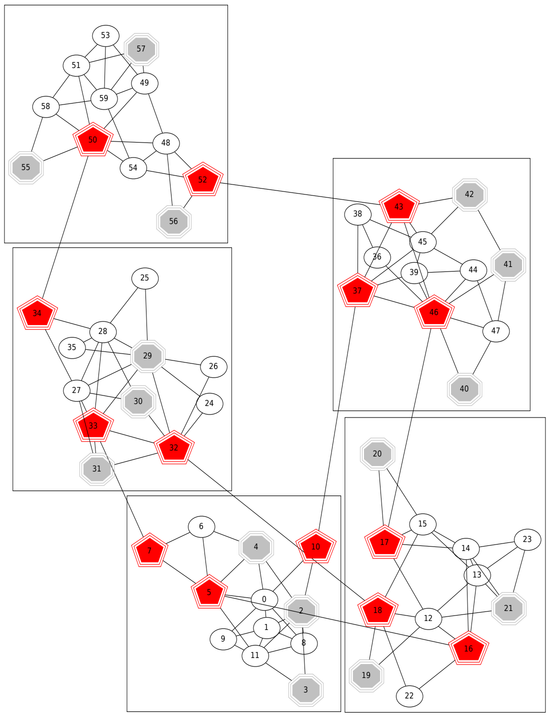

Figure 3: Topology for 30 Users

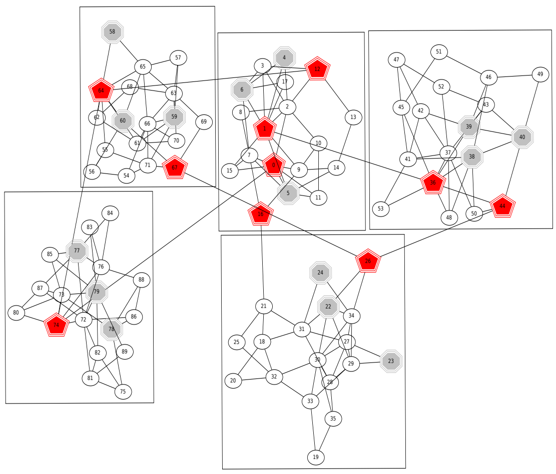

Figure 4: Topology for 60 Users

Users, in this case, are the blank circle, where the metascheduler is present. There are also red nodes in the topology, which are routers that connect two or more regions. The data centers are in red. The nodes and their connections are generated by Brite using the Waxman algorithm, which has been widely employed in network simulations, as its node distribution is similar to those found on physical networks [16].

The cloud environment was simulated considering the network topology obtained from Brite [14]. This information about the network is used by Pastry and Chord systems to determine the data center with the lowest latency in relation to the user that requested services.

An exponential distribution is used, in each topology, to simulate the network behavior. The times considered in the experiments are the response times based on the communication latency between the data center and the client. These topologies (Figure 3 and Figure 4) were utilized to observe the behavior of the routing system in a overloaded scenario. These scenarios can be changed because the experiments are not dependent on a specific topology and the values of latency are obtained by the distance between the nodes. The edge values are based on the seed of the simulation that changes the value of the distance between the nodes.

4 Results

The experiments in this article followed rigidly the concepts involved in the process of performance evaluation discussed in [11]. It has been conducted by means techniques of performance evaluation, so that results could show the real behavior of the environment and bring greater confidence to the decision-making process.

The performance evaluation is required when it is necessary to compare a number of alternatives for the development or improvement of a system and choose the one that best fits the project. The experimental design used here is the full factorial, in which all combinations at all levels of all factors are utilized. Jain [11] and Buragohain [4] present further details about experimental design focusing, respectively, on theory and practice.

For the correct elaboration of the experiments, some points, as the definition of fixed and variable factors have to be well-marked. The fixed factors (Table 1) are those whose influence on the results is not considered and remains the same from the beginning to the end of the experiments.

The number of users, services and routing policies are the factors whose influence is verified in a full factorial experiment. As discussed in the previous section, three routing policies are considered.

Table 1: Fixed Factors

Machine	Values
Processor	Intel Core 2 Quad
Memory	4 GB RAM
Operational System	Ubuntu x64 10. 04
Java	1.6. 0_26
Simulator	Cloudsim 2. 1
Number of Data Centers	15
Host_RAM	16GB
Host_Proc	12000 MIPS
Host_Conn	1Gb

All the results presented in this paper are the average obtained from ten runs of each experiment and a 95% confidence interval. The response variable considered is the response time (RT).

This response time is related to the request, execution and delivery of services. The experiment was defined in Table 2 following the methodology presented by Jain [11].

Table 2: Experiments using Chord and Pastry levels for the Routing Policy

Exp	Routing Policy	Number of Users	Type of Services
1	Chord	30	Light
2	Chord	30	Heavy
3	Chord	60	Light
4	Chord	60	Heavy
5	Pastry	30	Light
6	Pastry	30	Heavy
7	Pastry	60	Light
8	Pastry	60	Heavy

The experimental planning considers the following factors:

- Hybrid Cloud - to simulate the private cloud overload sent to the public cloud. In this context thirty and sixty clients that corresponded to the private clouds were defined. The public clouds comprehend fifteen data centers, as shown in Table 1.
- Type of Services - to define the services the study presented by [6] was utilized. The light services are defined as services web and the heavy services are defined as scientific applications. Here the light services use a probabilistic distribution and vary from 2000 to 2500 MIPS. On the other hand, the heavy services range from 20000 to 25000

MIPS. This huge space was used to simulate an overhead of the system.

- Routing Policies - to understand the differences between the policies considered. Here the policies are based on the P2P systems.

The response times obtained are presented in Figure 5, which shows the behavior of the two P2P systems.

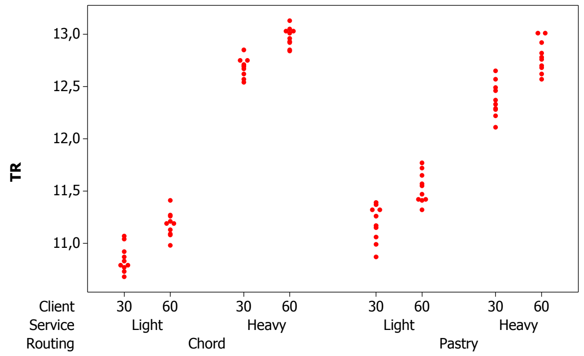

Figure 5: Dispersion of the response times

This graph shows that the dispersion found in the Chord system is less spread than that of the Pastry system. The behavior presented by the Chord system suggests that its standard deviation is low if compared with the Pastry system. Figure 6 shows these differences in another way.

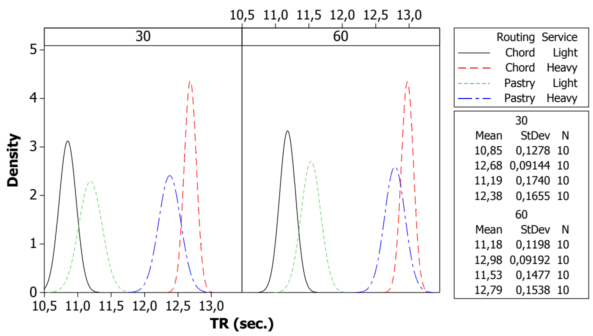

Figure 6: Response times

The Y axis of Figure 6 shows the density related to the standard deviation. The density is more pronounced at a point where the results have low deviation. The X axis shows the response times obtained. The center of the curve is related to the mean response times obtained. The right side of the graph shows the definition of the configuration related with the specific color. The exact values for the mean and the deviations are also presented.

These values are used to obtain the statistical differences between the systems. These difference are showed based on the error obtained, which is based on Equation 1.

$$\frac{DesvPad * Const}{\sqrt{N}} \tag{1}$$

The values for the equation variables are those presented in Figure 6, where $Desvpad$ is related to the standard deviation, $Const$ is the value of the t-student table [11] and N is the number of replications, in this case, the number of executions for each experiment. Figure 7 shows the differences in the response times for light services.

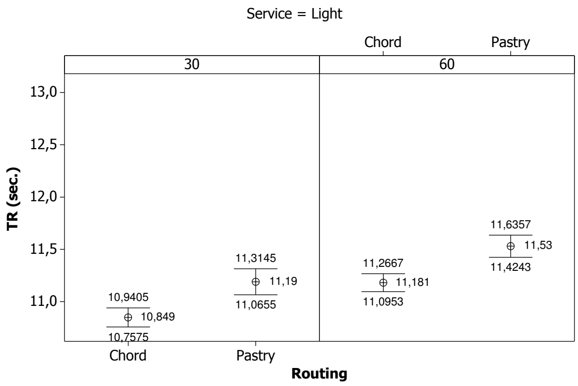

Figure 7: Interval plot for light services

The lower range of the Pastry system is higher than the upper range of the Chord system for both thirty and sixty clients. This behavior shows that the Chord system had better response times based on the statistical values of the experiments.

Figure 8 shows the difference in the response times for the heavy services.

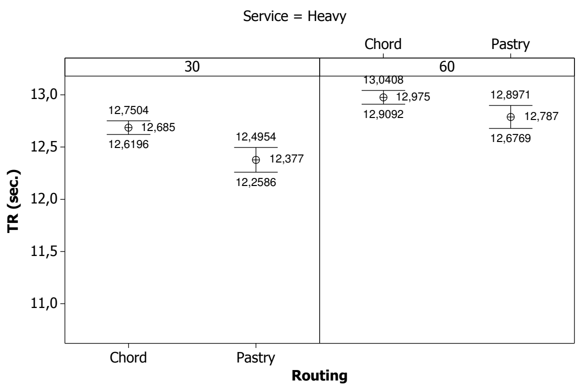

Figure 8: Interval plot for heavy services

Contrarily to the light services, the lower range of Chord is higher than the upper range of Pastry. In this case the Pastry system provided the best response times.

To investigate the differences in the response times, Figure 9 shows the influence of factors on the response times based on the observation of the range of the experiments.

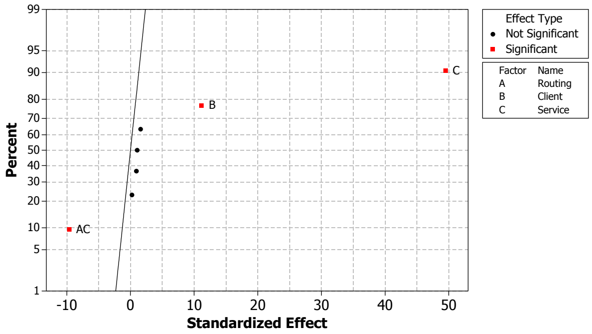

Figure 9: Influence of Factors

The graph of Figure 9 uses a normal distribution to plot the values.

Based on this distribution, a base line is obtained by the sample error and the number of the experiments is presented. It shows where the points are expected to lie if all effects are zero or if they are not significant. The X axis presents the real effect of the factors where the levels are changed.

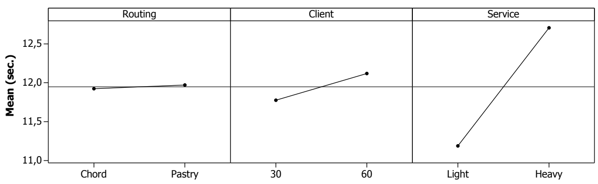

Figure 10: Main differences of mean response times

As shown in Figure 9, the most influential factor is the service (factor C) followed by the number of clients (B) and the combination of P2P systems and type of service utilized (AC). Figure 10 shows the main difference in the mean times based on the specific factor. In other words, each square refers to the mean time for the specific factor based on the sum of the rest of the factors.

The type of service has the biggest change of values considering all the factor changes, causing the major influence of factor C on the response times (Figure 9).

Figure 11 shows the interactions of factors considering the possible combinations. The first square is related to the number of clients and routing, the second is related to the type of services and the routing and the square below is related to the number of clients and type of services.

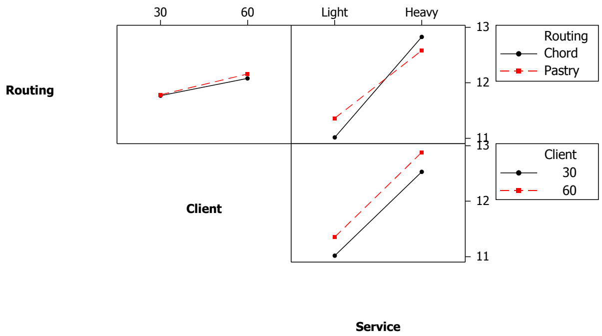

Figure 11: Interaction among the factors.

When the factor changes its levels, i.e, when the number of clients changes, the behavior is almost the same presented by the type of services. In both cases the lines are in parallel, causing no influence between the factors.

In the interaction of routing and services the light services of Chord system have better response times than those of Pastry system. When the service type is heavy, the behavior changes and Pastry becomes better than Chord. In other words, the growth in the response times is bigger in the Chord system, influencing the combination of factors, as presented in Figure 15.

Based on the results it is possible to notice that the major influence in the response times is caused by the type of service.

A second set of experiments (Table 3) was conducted to verify the differences between the P2P systems and the brute force approach.

Table 3: Experiments using Chord and BNC levels for the Routing Policy

Exp	Routing Policy	Number of Users	Type of Services
1	Chord	30	Light
2	Chord	30	Heavy
3	Chord	60	Light
4	Chord	60	Heavy
5	BNC	30	Light
6	BNC	30	Heavy
7	BNC	60	Light
8	BNC	60	Heavy

Here, the difference is in the comparison of the BNC (based on the network capacity) policy and the Chord system. These experiments are helpful to show the losses in the response times when the number of interaction in the search is limited.

Figure 12 shows the plot of individual values. The same behavior is observed in both the Chord system and BNC and no big dispersion is found in the spread of the results.

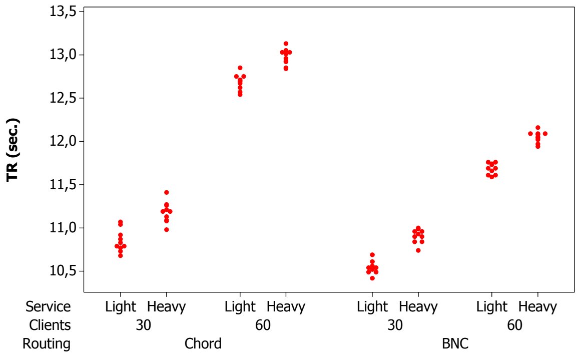

Figure 12: Dispersion of policies response times

The differences in the response times are presented in Figures 13 and 14.

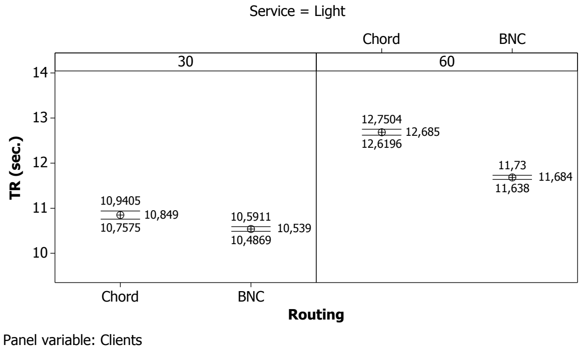

Figure 13: Light services

The upper range of the BNC is lower than the lower range of the Chord system for both thirty and sixty clients. This behavior shows that the BNC system had better response times based on the statistical values of the experiments.

The mean values are close, but statistically different. This difference is approximately $0.31sec.$ for thirty clients and $1.0sec.$ for sixty clients with light services.

For heavy services, the difference is $0.28sec.$ for thirty clients and $0.92sec.$ for sixty clients. This behavior shows that the increase in the response time was proportional for both Chord and BNC.

Figure 15 shows the effects on the response times based on quartiles. Each quartile shows the negative and positive values.

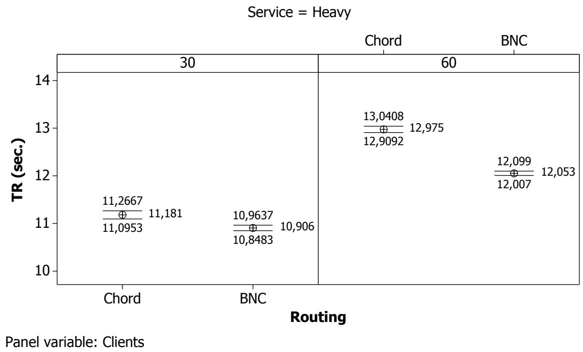

Figure 14: Heavy services

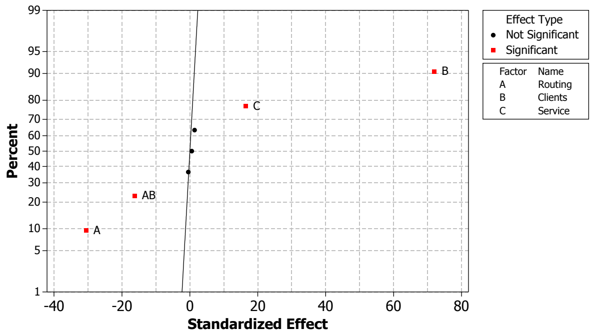

Figure 15: Influence of Factors

The graph of Figure 15 uses a normal distribution to plot the values.

Based on this distribution, a base line is obtained by the sample error and the number of the experiments is presented. It shows where the points are expected to lie if all effects are zero or if they are not significant.

In relation to response times, the policy (Factor A) and the interaction between policy and clients (Factor AB) influenced them negatively. On the other hand, the service (Factor C) and clients (Factor B) had an increase in the response times.

The best response times were achieved by the correct application of policy routing.

Figure 16 shows the main effects of the factors over the response times.

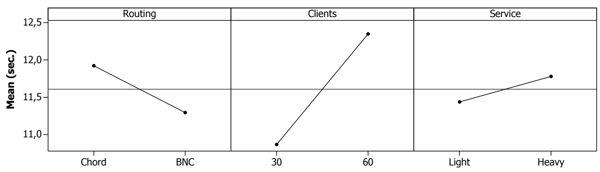

Figure 16: Main differences of the factors

In the first case, i.e policy routing, the best response times were obtained by the brute force approach. In the second square (Clients) the response times had an increase because the system became more utilized with more clients.

The type of services presented an increase in the response times due to the maximization of the system use. However, for the simulation scenario the big changes in the response times were made by both routing policies and clients.

Figure 17 shows the interaction of factors.

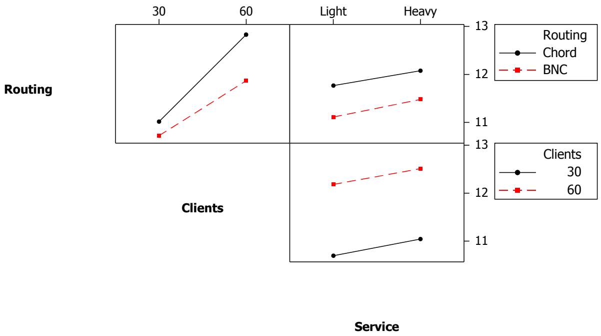

Figure 17: Interaction among the factors.

Clients and services (Factors BC) and services and policy (Factors AC) exert no influence on the response times because the behavior is the same in the level change. The red and black lines are in parallel, representing the same behavior for the level change.

The only interaction presented here is in the clients change (30 to 60) and the routing change (Chord to BNC). The increase in the response times was larger in the Chord System.

5 Discussion of results

The problem to develop these approaches is the necessity of changing the infrastructure to receive the P2P system. In other words, the union of P2P systems and cloud, in the related studies, is a good choice for private clouds.

Based on previous studies, the metascheduler utilized to evaluate the service discovery in cloud computing is the one presented by Peixoto et al. [18] [17]. This metascheduler was selected because it presents a communication model that can work with independent routing policies. This characteristic allows using different policies, including the Pastry policy presented by Ranjan et al. [19] and the Chord policy presented by Lai et al. [12] without changing the infrastructure.

As stated by Peixoto et al. [17], the metascheduler works in different layers. According to Figure 18, the high layer is a P2P, which performs the search by the correct infrastructure based on the lowest latency to the client.

A modification in their model is the characterization of all metaschedulers in the same level. In other words, the SSG (Super Scheduler Gateway), presented in Figure 18, was removed to provide a decentralized scenario.

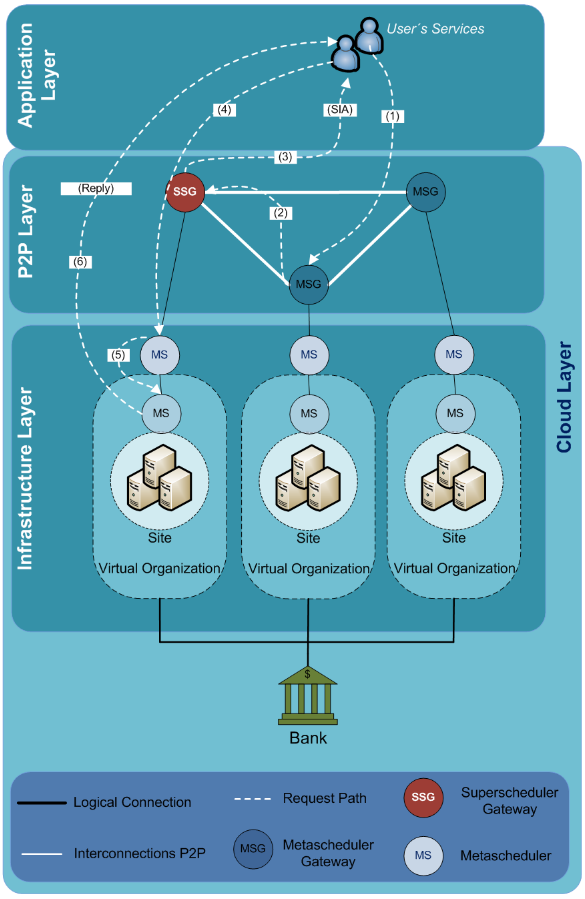

Figure 18: MACC Peering Arrangement [17]

The metascheduler undergoes two steps to process the request: the discovery of the best data center and the local scheduling presented in [17]. In our proposal the second step is not considered, as local scheduling is not the focus. This metascheduler is helpful to simulate tests over hybrid scenarios, which comprise a range of private clouds that send their overloads to a public cloud. The same policies used by Ranjan et al. [19] and Lai et al. [12] were utilized to link the public and private clouds.

6 Conclusions

This paper has presented an evaluation of how different routing policies following peer-to-peer (P2P) approach influence the performance achieved in cloud computing platforms.

A metascheduler model is considered in order to allow the integration between P2P systems and cloud computing. The communication on the cloud computing platform involves several issues, such as the scheduling of the services considering the multiple constraints imposed. Most existing P2P algorithms are not designed to address this problem.

The paper has proposed a statistical analysis comparison among the main P2P algorithms in complex scenarios, which are used in the network overlay to distribute the services over the Internet. It is possible to select the P2P algorithm that can obtain better performance in specific cases.

The results have provided valuable insights into the performance of alternative P2P communication strategies for a cloud computing infrastructure. The main objective was to determine how to guarantee QoS (using the response time as an attribute) considering P2P systems.

As the systems have the same purpose, but a different way to perform the search, a performance evaluation was proposed to identify their differences. In this evaluation the biggest difference is related to the type of services.

The analysis showed that, for light services, the Chord system has better response times, i.e around 6% better than the Pastry system; for heavy services, the Pastry system present better response times (2% better than the Chord system).

A second set of experiments was conducted to determine the difference in the response times between a P2P system and a brute force system.

A clear difference was observed between brute force and the P2P system. However, this difference was around $0.7 sec.$, which is a small difference, if the number of interactions involved is taken into account.

The results showed that it is possible to guarantee good response times using P2P approaches.

Acknowledgements

The authors thank the support of the Distributed System and Concurrent Program, Group of the Computes Systems Department at ICMC-USP, as well as the important contributions from Ph.D. students at ICMC-USP.

References

[1] Andrade, N., Cirne, W., Brasileiro, F., and Roisenberg, P. Ourgrid: An approach to easily assemble grids with equitable resource sharing. In *Job scheduling strategies for parallel processing*, pages 61–86. Springer, 2003.

[2] Androutsellis-theotokis, S. and Spinellis, D. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36:335–371, 2004.

[3] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[4] Buragohain, M. and Mahanta, C. A novel approach for anfis modelling based on full factorial design. *Applied Soft Computing*, 8(1):609–625, 2008.

[5] Buyya, R., Ranjan, R., and Calheiros, R. N. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010, Busan, South Korea, May 21-23), LNCS, Springer, Germany, 2010.*, abs/1003.3920, 2010.

[6] Calheiros, R., Ranjan, R., and Buyya, R. Virtual machine provisioning based on analytical performance and qos in cloud computing environments. In *Parallel Processing (ICPP), 2011 International Conference on*, pages 295–304. IEEE, sept. 2011.

[7] Calheiros, R. N., Ranjan, R., Beloglazov, A., Rose, C. A. F. D., and Buyya, R. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. In *Software: Practice and Experience*, volume 40. Wiley Press, 2010.

[8] Carolan, J., Gaede, S., Baty, J., Brunette, G., Licht, A., Remmell, J., Tucker, L., and Weise, J. Sun microsystems - introduction to cloud computing architecture. In *White paper 1st edition, pp.9-12, june 2009. Retrieved in December 07, 2009, from: http://www.sun.com/featured-articles/CloudComputing.pdf*, 2009.

[9] Chard, K. *DRIVE: A Distributed Economic Meta-Scheduler for the Federation of Grid and Cloud Systems*. Phd thesis - Victoria University of Wellington, Wellington, New Zealand, 2011.

[10] Iosup, A., Epema, D. H. J., Tannenbaum, T., Farrellee, M., and Livny, M. Inter-operating grids through delegated matchmaking. In *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, SC '07, pages 13:1–13:12, New York, NY, USA, 2007. ACM.

[11] Jain, R. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley, 1991.

[12] Lai, K.-C., Huang, K. C., Koong, C. S. C.-S., Yu, Y. F., Huang, P. J., Chen, Q. J., and Hunang, T. L. A P2P Resource Discovery Strategy for Cloud Computing Systems. *Journal of Computers*, 21(1):96–97, 2010.

[13] Leal, K., Huedo, E., and Llorente, I. M. A decentralized model for scheduling independent tasks in federated grids. *Future Gener. Comput. Syst.*, 25:840–852, September 2009.

[14] Medina, A., Lakhina, A., Matta, I., and Byers, J. Brite: An approach to universal topology generation. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pages 346–353. IEEE, 2001.

[15] Mell, P. and Grance, T. The NIST Definition of Cloud Computing. Technical report, National Institute of Standards and Technology, Information Technology Laboratory, 2011.

[16] Naldi, M. Connectivity of waxman topology models. *Computer communications*, 29(1):24–31, 2005.

[17] Peixoto, M. L. M., Santana, M. J., Estrella, J. C., Tavares, T. C., Kuehne, B. T., and Santana, R. H. C. A Metascheduler architecture to provide QoS on the cloud computing. In *ICT '10: 17th International Conference on Telecommunications*, volume 1, pages 650–657, Doha, Qatar, 2010. IEEE Computer Society.

[18] Peixoto, M. L. M., Santana, M. J., and Santana, R. H. C. A p2p hierarchical metascheduler to obtain qos in a grid economy services. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 01*, volume 1 of *CSE '09*, pages 292–297, Washington, DC, USA, 2009. IEEE Computer Society.

[19] Ranjan, R., Zhao, L., Wu, X., Liu, A., Quiroz, A., and Parashar, M. Peer-to-peer cloud provisioning: Service discovery and load-balancing. *Cloud Computing*, pages 195–217, 2010.

[20] Rimal, B. P., Choi, E., and Lumb, I. A taxonomy and survey of cloud computing systems. In *NCM '09: Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 44–51, Washington, DC, USA, 2009. IEEE, IEEE Computer Society.

[21] Rowstron, A. I. T. and Druschel, P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350, London, UK, 2001. Springer-Verlag.

[22] Spence, D. and Harris, T. Xenosearch: Distributed resource discovery in the xenoserver open platform. In *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on*, pages 216–225. IEEE, 2003.

[23] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM 01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 149–160, New York, NY, USA, 2001. ACM.

[24] Tam, D., Azimi, R., and Jacobsen, H. Building content-based publish/subscribe systems with distributed hash tables. *Databases, Information Systems, and Peer-to-Peer Computing*, pages 138–152, 2004.

[25] Xhafa, F. and Abraham, A. Computational models and heuristic methods for grid scheduling problems. *Future Gener. Comput. Syst.*, 26:608–621, 2010.

[26] Yamini, L., LathaSelvi, G., and Mukherjee, S. Efficient metascheduling in a cloud extended grid environment. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 667–672, 2011.

[27] Yong Meng, T., March, V., and Wang, X. A dht-based grid resource indexing and discovery scheme. In *Singapore-MIT Alliance Symposium 2005*. Citeseer, 2005.

[28] Zhang, Q., Cheng, L., and Boutaba, R. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, 2010.

[29] Zhou, J., Abdullah, N., and Shi, Z. A hybrid p2p approach to service discovery in the cloud. *International Journal of Information Technology and Computer Science (IJITCS)*, 3(1):1 – 9, 2011.