UNIVERSITI PUTRA MALAYSIA

CPU AVAILABILITY PREDICTOR AND MACHINE RECOMMENDER FOR A DESKTOP GRID

MOHAMMAD YASER SHAFAZAND

FSKTM 2014 25

CPU AVAILABILITY PREDICTOR AND MACHINE RECOMMENDER FOR A DESKTOP GRID

By

MOHAMMAD YASER SHAFAZAND

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Master of Science

July 2014

DEDICATIONS

To the ones that supported me.

...

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

CPU AVAILABILITY PREDICTOR AND MACHINE RECOMMENDER FOR A DESKTOP GRID

By

MOHAMMAD YASER SHAFAZAND

July 2014

Chairman: Rohaya Binti Latip, Ph.D.

Faculty: Computer Science and Information Technology

Desktop grid computing has emerged from the concept of providing relatively large amounts of computing power for little cost using typical desktop machines. The computing power is stolen by the scheduler from desktop computers while they are in an idle state. Approaches have been taken to utilize the idle CPU cycles of desktop machines using desktop grid computing. However, efficiently using resources has complications due to their volatile state. The scheduler, being responsible for appropriate job dissemination, needs accurate job runtime predictions and thus resource availability predictions for an efficient resource management. Distance, high diversity, distributed resource ownership and intermittent availability of resources are considered challenges which affect forecasting resource availability in desktop grids. Many have successfully researched on processes, techniques and methodologies for accurate resource availability predictions in grid computing, but only some have specifically considered the desktop grid. The overhead of the predictor systems in these environments are also less studied. This has led to more sophisticated predictors sacrificing simplicity and having high resource usage overhead.

Another issue for the desktop grid scheduler is to appropriately match suitable desktop machines (resource providers) to a job. Machines that have enough available resources to satisfy the job requirements and which could also finish the job as soon as possible, must be selected.

As a solution to these problems, two frameworks for a desktop grid computing environment are modelled and proposed. The first framework attends to cluster and model availability behaviour of the machines in the server and predict everyday CPU availability on each client for that particular client. The second framework recommends desktop machines to the scheduler based on resource specifications

and the CPU availability property. We simulated both frameworks by developing software modules and using the CPU availability of 700 desktop machines of an actual desktop grid. We analysed both the accuracy and overhead of our predicting module and compared it with other studies.

The results for our first framework show lower server resource usage overhead compared to similar works while preserving the accuracy. For the second framework, compared to two well known schedulers in desktop grid, our design proved better performance via lower overall job batch completion time. The design of the second framework also supports various resource availability parameters as well as CPU availability. It also has no interference with existing scheduling processes. The overall application of these two frameworks has decreased server resource usage overload and assisted the desktop grid scheduler in better decision making and thus lower overall job completion time.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
sebagai memenuhi keperluan untuk ijazah Master Sains

RAMALAN KETERSEDIAAN CPU DAN CADANGAN MESIN-KERJA BAGI DESKTOP GRID

Oleh

MOHAMMAD YASER SHAFAZAND

Julai 2014

Pengerusi: Rohaya Binti Latip, Ph.D.

Fakulti: Sains Komputer dan Teknologi Maklumat

Pengkomputeran desktop grid wujud dari konsep penggunaan pengkomputeran berkendalian tinggi untuk pengurangan kos dengan menggunakan mesin desktop biasa. Kuasa pengkomputeran digunapakai oleh penjadual dari komputer desktop semasa didalam keadaan melahu.

Pendekatan telah diambil dengan memanfaatkan kitaran CPU melahu mesin-mesin desktop menggunakan pengkomputeran desktop grid. Walaubagaimanapun, penggunaan sumber tersebut secara efisyen mempunyai komplikasi kerana keadaannya yang meruap.

Penjadual, komponen yang bertanggungjawab mengatur dan mengagihkan tugas yang bersesuaian, memerlukan anggaran masa larian tugas dengan tepat untuk memastikan pengurusan sumber yang efisyen. Jarak, kepelbagaian yang tinggi, pemilikan sumber teragih dan kewujudan sumber-sumber berskala antara faktor yang boleh menjejaskan anggaran masa larian tugas dalam grid desktop.

Banyak kajian terdahulu melibatkan proses, teknik dan metodologi telah berjaya menghasilkan ramalan sumber yang tepat dalam pengkomputeran grid, tetapi hanya beberapa elemen spesifik telah mempertimbangkan persekitaran desktop grid dan overhed daripada sistem peramal. Ini telah membawa kepada sistem peramal yang lebih canggih tetapi kompleks dan penggunaan overhed yang tinggi.Selain dari itu, isu penting yang melibatkan operator penjadual desktop grid adalah menyuaipadan mesin desktop yang berperanan sebagai penyedia sumber dengan tugas yang sesuai dan tepat. Mesin-mesin yang mempunyai sumber yang cukup bagi memenuhi keperluan kerja dan dapat mampu menyelesaikan tugas secara efisyen juga perlu dipilih.

Sebagai penyelesaian kepada permasalahan ini, dua modul untuk persekitaran pengkomputeran desktop grid telah dicadangkan. Rangka kerja pertama perlu wujud dalam bentuk kluster dan sifat ketersediaan model setiap mesin dalam pelayan mampu meramalkan ketersediaan CPU setiap hari untuk setiap komputer klien yang tertentu. Rangka kerja kedua perlu mengesyorkan mesin desktop kepada operator penjadual berdasarkan spesifikasi sumber dan harta ketersediaan CPU. Kajian kami melibatkan simulasi kedua-dua rangka kerja dengan membangunkan modul perisian dan menggunakan CPU yang tersedia 700 mesin-mesin desktop sebuah desktop grid yang sebenar. Kami menganalisis dan membandingkan kedua-dua ketepatan rangka kerja dan overhed modul peramal yang dicadangkan dengan kajian-kajian lain.

Keputusan bagi rangka kerja pertama menunjukkan penggunaan sumber pelayan overhed lebih rendah dengan kajian sedia ada disamping mengekalkan ketepatan. Untuk rangka kerja kedua, berbanding dua penjadual popular desktop grid, reka bentuk kajian kami telah membuktikan prestasi yang lebih baik menerusi masa selesai kelompok kerja yang lebih rendah. Reka bentuk rangka kerja kedua juga menyokong pelbagai parameter ketersediaan sumber dan juga ketersediaan CPU. Ia juga tidak mempunyai gangguan dengan proses penjadualan sedia ada. Aplikasi keseluruhan kedua-dua rangka kerja telah mengurangkan penggunaan beban sumber pelayan dan membantu penjadual desktop grid dalam membuat keputusan yang lebih baik disamping masa selesai keseluruhan kerja yang lebih rendah.

ACKNOWLEDGEMENTS

First of all, I would like to thank my dear parents and family for their support throughout my education. I would also like to express my gratitude and appreciation to my supervisor, Dr. Rohaya Latip and my co-supervisor, Dr. Azizol Abdullah who also supported me to finish my thesis. A special thanks to my supervisor, whose help, suggestions and encouragements helped me in finishing this study.

My thanks and appreciation also goes to people who have willingly helped me out.

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of **Master of Science**.

The members of the Supervisory Committee were as follows:

Rohaya Latip, Ph.D.
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairperson)

Azizol Abdulah, Ph.D.
Senior Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

BUJANG BIN KIM HUAT, Ph.D.
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date:

Declaration by graduate student

I hereby confirm that:

- this thesis is my original work;
- quotations, illustrations and citations have been duly referenced;
- this thesis has not been submitted previously or concurrently for any other degree at any other institutions;
- intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- written permission must be obtained from supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before thesis is published (in the form of written, printed or in electronic form) including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- there is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) and the Universiti Putra Malaysia (Research) Rules 2012. The thesis has undergone plagiarism detection software.

Signature: ____________________ Date: ____________

Name and Matric No.: ______________________________

Declaration by Members of Supervisory Committee

This is to confirm that:

- the research conducted and the writing of this thesis was under our supervision;
- supervision responsibilities as stated in the Universiti Putra Malaysia (Graduate Studies) Rules 2003 (Revision 2012-2013) are adhered to.

Signature:	____________________	Signature:	____________________
Name of Chairman of Supervisory Committee:	____________________	Name of Member of Supervisory Committee:	____________________

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial neural network
AR	Auto-Regressive
ARIMA	Auto-Regressive Integrated Moving Average
CBR	Case-based Reasoning
DG	Desktop Grid
DGAP	Desktop Grid Availability Predictor Module
DGRE	Desktop Grid Recommender and Estimator Module
ES	Exponential Smoothing
FLOP (s)	FLoating point OPerations per Second
GA	Genetic Algorithm
HPC	High-performance Computing
HTC	High-throughput Computing
MA	Moving-Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
NN	Nearest Neighbour Algorithm
PSO	Particle Swarm Optimization
RAC	Recent Average Credit
ReqGFlOpH	Requested GigaFlop Operations per Hour
RS	Recommender System
SVM	Support Vector Machine
VC	Volunteer Computing

CHAPTER 1

INTRODUCTION

1.1 Background

Supercomputer's computing power and disk space have given their place to decentralized personal computers with a public ownership (Anderson, 2004). The price drop for desktop machines has made it easy for many institutes and universities to maintain up-to-date computer systems. These systems have high computational performances. Having multi-cores and multi-threads are the common trend of today's desktop machines. A lot of these computers remain turned on most of the day while having a low CPU utilization and a high idle time. While these resource pools contain multi threaded multi core computational capacities, users have shown to not be able to fully utilize these resources in a systems uptime.
Desktop grids (Kondo et al., 2004) are a form of computational grids in which each resource is a desktop machine. It is a paradigm that uses heterogeneous, distributively computing resources (e.g., desktops machines, notebooks) connected through the Internet or Intranet and owned by an organization to provide computing power needed by projects. It has greatly enabled the use of vast amount of computational capacities by stealing cycles while the computer is in idle state. While emerging from the basic fundamentals of grid computing, their dynamic nature created a new branch of research. While grid computing resources are centrally managed, predictable and connected by full-time, high bandwidth network links (Anderson, 2004), Desktop grids belong to separate individuals which manage their own resources. Their uptime is not scheduled and the connection is intermittent. These resources are distributively owned and thus offer no guaranty of availability (Lázaro et al., 2012).

One of the current focuses in desktop grids are to estimate if a resource or machine is available long enough for a job to complete and return its results (Lázaro et al., 2012). The resource allocation or scheduling problem is known as the problem to efficiently assign resources to jobs (Raman et al., 1998). This is a well known challenge in all categories of grid computing. A system model is an abstraction of the underlying resources which provides the properties and availability information at any point of time (Raman et al., 1998). Resources that are distributively owned are mostly used in grid systems concerned with throughput and have no fixed system model. Thus, researches focus on mechanisms to predict properties and availabilities of resources. To have a better utilization, it is important for the scheduler to be able to match jobs to the appropriate machines. It must also be able to forecast resource (i.e CPU) availability. Prediction is a well known technique for resource management (Lázaro et al., 2012). Schedulers must be able to forecast the obtainable performance of a job (Wolski, 2003). The challenges prediction faces in desktop grids are mostly:

1. The resources are federated to the grid by their owners, thus no complete control or management operation is permitted by the grid administration on these resources. This results in a dynamic availability behaviour (Wolski, 2003).

2. Any job running could be interrupted and suspended by the owners work which has a higher priority (Raman et al., 1998).

3. The connection is intermittent and though the resources could be available to the local running jobs, the client machine could be disconnected from the server which manages and schedules jobs. This is due to using the Internet or any unreliable connection between the server and the client machines or between peers.

We realized two main types of availability: resource availability and machine (host) availability (Lázaro et al., 2012). Resource availability discusses the availability of the machines resources (i.e CPU, Disk, RAM) to run jobs. Machine availability discusses availability of the connection between the scheduler/resource manager and the machines which are federated to the grid. Among a computers resources, the CPU is the most important for CPU-bound jobs and there are many studies trying to predict the behaviour of this resource. We focus on online CPU availability prediction for our purpose. CPU availability refers to the CPU of a machine being dedicated to perform computations for the benefit of a distributed system (Lázaro et al., 2012).

1.2 Problem Statement

The need for more computational power is increasing each day due to the increase in information and sophistication of systems. This need has led the scientists to think about using advancement in computers and communication technologies to provide more computational power by aggregating the resources to create a unified resource pool. This is how the computational grids and desktop grids emerged. Researches showing the potential amount of usable computer resources in desktop machines revealed a new resource pool. The desktop grid technology has been introduced to utilize the idle resources in the desktop machines using an Intranet or Internet connection. Aggregating and utilizing these unreliable hosting machines has led to tens and hundreds of TeraFLOPS of computational power (i.e. SETI@HOME) (Anderson, 2004).

There is an increase in the number of desktop machines connected to the Internet every year and the idea to efficiently harvest these idle resources is a concern of current researches in this field. There are many open challenges in the field of grid computing (i.e. the Scheduling problem) which have been inherited in more complicated forms in the field of desktop grid computing. Predicting the availability of resources has long been a challenge in desktop grid computing considering the impact it has on resource management and scheduling (Ren et al., 2007; Javadi et al., 2009; Naseera and Madhu Murthy, 2013; Ramachandran et al., 2012; Lázaro et al., 2012; Hussin and Latip, 2013). There are many aspects in designing a predictor, accuracy is a well known and focused one. The overhead that the predictor creates is less studied (Bey et al., 2011). While recent forecasting studies have improved accuracy, they have also increased complexity and overhead. This overhead reduces the feasibility of the system to be actually applicable to a vast amount of federated desktop machines.

The Scheduling problem also involves the method to effectively assign resources to jobs (Raman et al., 1998). Schedulers which have faster job turnaround times are known to be more efficient (Raman et al., 1998; Lin et al., 2010; Gil et al., 2013). The time taken from submitting the first task to the desktop machines and the time all the jobs have completed and returned is the turnaround time. It is important to intelligently select best suited resource(s) for a job by the scheduler. This would decrease the job completion time and the generally, the turnaround time. The function the scheduler has to improve in this situation is known as matchmaking (Raman et al., 1998). Creating an efficient structured mechanism for matchmaking in desktop grids is also a need. Each new research doesn't completely consider all the advantages of its previous work while improving a metric. In the designing and development of schedulers the same process happens and some advantages of previous systems are lost along the improvement. Solutions must be considered to maintain previous advantages while improving certain metrics.

1.3 Research Questions

The research questions define the main motivations that have triggered the research in the first place. The main questions have been structured as following:

1. How to design a CPU availability predictor for a desktop grid and maintain an acceptable accuracy and a low overhead.

2. How to improve job completion time by introducing more efficient machines to the scheduler and not interfere with its current processes?

   (a) How does CPU availability prediction fit in the problem of finding appropriate machines?

All the processes done in this study are an effort to find answers to these questions.

1.4 Research Objectives

We pursue two main objectives which are:

1. To develop a framework of a predictor based on new features to reduce resource usage overhead on the server while maintaining prediction accuracy.

2. To develop a framework for matchmaking in the scheduler that improves job completion time by utilizing a recommender system that accepts multiple availability features and has no interference with existing scheduling processes.

1.5 Scope and limitations

Our research is conducted in these limited scopes:

- In the field of Desktop Grid computing which is a sub-field of the broader research field of Computational Grids.

- Not in Volunteer Computing. Although Volunteer Computing is similar to Desktop Grid in nature, our research is not generally designed for this area of research. The discussions which include this area would be explicitly mentioned.

- In CPU availability, as the main resource that conducts computations on machines for the purpose of this study.

- Limited ability to run processes rather that the ones sent in the form of jobs from the scheduler on the clients. This means that the clients are not allowed to run tasks that need intensive CPU power and are not sent by the scheduler.

1.6 Contributions

The main contributions of this research are:

1. A resource availability predictor framework implemented as a software module which has a low overhead and maintains accuracy.

2. A framework implemented as a software recommender module which improves overall batch job completion time by recommending appropriate machines to the scheduler.

1.7 Thesis Organization

This thesis is organized based on the standard structure for thesis and dissertations of the Universiti Putra Malaysia. The detail information of how the researches have been conducted and a report on the results have been documented. This thesis consists of six chapters. The next chapters are:

Chapter 2, literature review, which the feasibility knowledge of this research has been obtained by. Research material such as journals, conference proceedings, seminars, thesis, books, computer codes, and online resources are the sources used to obtain the main knowledge to find answers to the research questions. The best practices and effective models are extracted from the works of other researchers to be extended and improved in our model.

Chapter 3 discusses the research methodology. The approaches to search for answers of the research questions are justified in this chapter alongside the metrics used for our evaluations. The data used for evaluation and the methodologies used to answer each research question and the procedures followed are discussed.

In Chapter 4 we describe the simulation and experiments for our CPU availability predictor module. A module and design to predict CPU availability by considering overhead. We have evaluated our modules accuracy and overhead and compare it to other researches in this chapter. The proposed model and design is discussed and justified.

The next chapter, Chapter 5, describes the machine-job recommendation module. This module is set to assist the scheduler in decreasing the turnaround time. Simulations are performed and the results are compared to matchmaking schemes in well known desktop grid middleware.

The final chapter concludes the research and thesis while pointing out the overall achievements, limitations and future potential work to be pursued in following studies.

REFERENCES

AAAI (2012). AITopics / CaseBasedReasoning.

Aamodt, A. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7:39–59.

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749.

Anderson, D. P. (2004). BOINC: a system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pages 4–10.

Andrzejak, A., Domingues, P., and Silva, L. (2006). Predicting Machine Availabilities in Desktop Pools. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 1–4.

Armstrong, J. (2001). *Principles of Forecasting - A Handbook for Researchers and practitioners*. Springer US.

Bey, K. B., Benhammadi, F., Gessoum, Z., and Mokhtari, A. (2011). {CPU} load prediction using neuro-fuzzy and Bayesian inferences. *Neurocomputing*, 74(10):1606–1616.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time series analysis: forecasting and control*. Prentice Hall, Englewood Cliffs, NJ, USA, 3rd edition edition.

Deelman, E., Blythe, J., Gil, Y., and Kesselman, C. (2004). Grid Resource Management. chapter Workflow M, pages 99–116. Kluwer Academic Publishers, Norwell, MA, USA.

Dinda, P. and O'Hallaron, D. (2000). Host load prediction using linear models. *Cluster Computing*, 3(4):265–280.

Dinda, P. A. and O'Hallaron, D. R. (1999). Evaluation of linear models for host load prediction. In *IEEE International Symposium on High Performance Distributed Computing, Proceedings*, pages 87–96, Redondo Beach, CA, USA. IEEE, Piscataway, NJ, United States.

Dooley, K. (2002). Simulation research methods. *Companion to organizations*, pages 829–848.

Elkan, C. (2003). Using the Triangle Inequality to Accelerate k-Means.

Estrada, T., Taufer, M., and Anderson, D. (2009). Performance Prediction and Analysis of BOINC Projects: An Empirical Study with EmBOINC. *Journal of Grid Computing*, 7(4):537–554.

Gil, J.-M., Kim, S., and Lee, J. (2013). Task Replication and Scheduling Based on Nearest Neighbor Classification in Desktop Grids. In Han, Y.-H., Park, D.-S., Jia, W., and Yeo, S.-S., editors, *Ubiquitous Information Technologies and Applications SE - 96*, volume 214 of *Lecture Notes in Electrical Engineering*, pages 889–895. Springer Netherlands.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Graff, M., Escalante, H. J., Cerda-Jacobo, J., and Gonzalez, A. A. (2013). Models of performance of time series forecasters. *Neurocomputing*, 122(0):375–385.

Hellinckx, P., Verboven, S., Arickx, F., and Broeckhove, J. (2009). Predicting Parameter Sweep Jobs: From Simulation to Grid Implementation. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS '09. International Conference on*, pages 402–408.

Hu, L., Che, X.-L., and Zheng, S.-Q. (2012). Online System for Grid Resource Monitoring and Machine Learning-Based Prediction. *Parallel and Distributed Systems, IEEE Transactions on*, 23(1):134–145.

Hussin, M. and Latip, R. (2013). Adaptive resource control mechanism through reputation-based scheduling in heterogeneous distributed systems. *Computer Science*, 9:1661–1668.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender Systems: An Introduction*. Cambridge University Press, Cambridge, UK.

Javadi, B., Kondo, D., Vincent, J.-M., and Anderson, D. P. (2009). Mining for statistical models of availability in large-scale distributed systems: An empirical study of SETI@home. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on*, pages 1–10.

Javadi, B., Kondo, D., Vincent, J.-M., and Anderson, D. P. (2011). Discovering Statistical Models of Availability in Large Distributed Systems: An Empirical Study of SETI@home. *Parallel and Distributed Systems, IEEE Transactions on*, 22(11):1896–1903.

Kondo, D., Javadi, B., Iosup, A., and Epema, D. (2010). The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pages 398–407.

Kondo, D., Taufer, M., Brooks, C., Casanova, H., and Chien, A. (2004). Characterizing and evaluating desktop grids: an empirical study. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, pages 26–.

Lázaro, D., Kondo, D., and Marquès, J. M. (2012). Long-term availability prediction for groups of volunteer resources. *Journal of Parallel and Distributed Computing*, 72(2):281–296.

Lin, Y., Luo, S., and Han, B. (2010). A novel grid resource matching scheme. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 7, pages 3743–3747.

Litzkow, M. J., Livny, M., and Mutka, M. W. (1988). Condor-a hunter of idle workstations. In *Distributed Computing Systems, 1988., 8th International Conference on*, pages 104–111.

Liu, C., Yang, L., Foster, I., and Angulo, D. (2002). Design and evaluation of a resource selection framework for Grid applications. In *High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium on*, pages 63–72.

Lorenzi, F. and Ricci, F. (2005). Case-Based Recommender Systems: A Unifying View. In Mobasher, B. and Anand, S., editors, *Intelligent Techniques for Web Personalization SE - 5*, volume 3169 of *Lecture Notes in Computer Science*, pages 89–113. Springer Berlin Heidelberg.

Mahmood, T. and Ricci, F. (2009). Improving Recommender Systems with Adaptive Conversational Strategies. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, pages 73–82, New York, NY, USA. ACM.

Naseera, S. and Madhu Murthy, K. (2013). Prediction based job scheduling strategy for a volunteer desktop grid. *Communications in Computer and Information Science*, 361 CCIS:25–38.

Nassif, L. N., Nogueira, J. M., Karmouch, A., Ahmed, M., and de Andrade, F. V. (2007). Job completion prediction using case-based reasoning for Grid computing environments. *Concurrency and Computation: Practice and Experience*, 19(9):1253–1269.

Palit, Ajoy K., Popovic, D. (2005). *Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control)*. Advances in industrial control. Springer, 1st editio edition.

Prem, H. and Raghavan, N. (2006). A Support Vector Machine Based Approach for Forecasting of Network Weather Services. *Journal of Grid Computing*, 4:89–114.

Quinson, M. (2002). Dynamic Performance Forcasting for Network-Enabled Servers in a Metacomputing Environment. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, IPDPS '02, pages 124—-, Washington, DC, USA. IEEE Computer Society.

Ramachandran, K., Lutfiyya, H., and Perry, M. (2012). Decentralized approach to resource availability prediction using group availability in a P2P desktop grid. *Future Generation Computer Systems*, 28(6):854–860.

Raman, R., Livny, M., and Solomon, M. (1998). Matchmaking: Distributed Resource Management for High Throughput Computing. In *In Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing*, pages 28–31.

Recio-García, J. A., González-Calero, P. A., and Díaz-Agudo, B. (2012). jcolibri2: A framework for building Case-based reasoning systems. *Science of Computer Programming*, (0):–.

Ren, X., Lee, S., Eigenmann, R., and Bagchi, S. (2007). Prediction of Resource Availability in Fine-Grained Cycle Sharing Systems Empirical Evaluation. *Journal of Grid Computing*, 5(2):173–195.

Resnick, P. and Varian, H. R. (1997). Recommender Systems. *Commun. ACM*, 40(3):56–58.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.

Rood, B. and Lewis, M. J. (2007). Multi-state grid resource availability characterization. In *Proceedings of the 8th IEEE/ACM International Conference on Grid Computing*, GRID '07, pages 42–49, Washington, DC, USA. IEEE Computer Society.

Salinas, S., Garino, C. b., and Zunino, A. (2012). An architecture for resource behavior prediction to improve scheduling systems performance on enterprise desktop grids. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7547 LNCS:186–196.

Seneviratne, S., Levy, D. C., and Buyya, R. (2013). A Taxonomy of Performance Prediction Systems in the Parallel and Distributed Computing Grids. *CoRR*, abs/1307.2.

Singh, K., pek, E., McKee, S. A., de Supinski, B. R., Schulz, M., and Caruana, R. (2007). Predicting parallel application performance via machine learning approaches. *Concurrency and Computation: Practice and Experience*, 19(17):2219–2235.

Smith, W., Foster, I., and Taylor, V. (2004). Predicting application run times with historical information. *Journal of Parallel and Distributed Computing*, 64(9):1007–1016.

Smyth, B. (2007). Case-Based Recommendation. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web SE - 11*, volume 4321 of *Lecture Notes in Computer Science*, pages 342–376. Springer Berlin Heidelberg.

Tannenbaum, T., Wright, D., Miller, K., and Livny, M. (2002). Condor – A Distributed Job Scheduler. In Sterling, T., editor, *Beowulf Cluster Computing with {L}inux*. MIT Press.

Thain, D., Tannenbaum, T., and Livny, M. (2003). Condor and the Grid. In Berman, F., Fox, G., and Hey, T., editors, *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons Inc.

Watson, I. (1999). Case-based reasoning is a methodology not a technology. *Knowledge-Based Systems*, 12(56):303–308.

Wolski, R. (1998). Dynamically forecasting network performance using the Network Weather Service. *Cluster Computing*, 1(1):119–132.

Wolski, R. (2003). Experiences with predicting resource performance on-line in computational grid settings. *SIGMETRICS Perform. Eval. Rev.*, 30(4):41–49.

Wolski, R., Spring, N. T., and Hayes, J. (1999). The network weather service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 15(56):757–768.

Wu, M. and Sun, X.-H. (2006). Grid harvest service: A performance system of grid computing. *Journal of Parallel and Distributed Computing*, 66(10):1322–1337.

Wu, Y., Hwang, K., Yuan, Y., and Zheng, W. (2010). Adaptive Workload Prediction of Grid Performance in Confidence Windows. *Parallel and Distributed Systems, IEEE Transactions on*, 21(7):925–938.

Xia, E., Jurisica, I., Waterhouse, J., and Sloan, V. (2010). Runtime Estimation Using the Case-Based Reasoning Approach for Scheduling in a Grid Environment. In Bichindaritz, I. and Montani, S., editors, *Case-Based Reasoning. Research and Development*, volume 6176 of *Lecture Notes in Computer Science*, pages 525–539. Springer Berlin / Heidelberg.

Yang, L., Schopf, J. M., and Foster, I. T. (2003). Conservative Scheduling: Using Predicted Variance to Improve Scheduling Decisions in Dynamic Environments.

APPENDIX A

ALGORITHMS

A.1 K-means Clustering [1]

A well known algorithm to partition n data nodes $x_i, i = 1...n$ into k clusters. As a result, each data point would be assigned to one of the clusters. To minimize the distance of a cluster center (centroid) to the data points, the positions $\mu_i, i = 1...k$ must be found. The equation solved for the K-means Algorithm is:

$$\arg\min_{c} \sum_{i=1}^{k} \sum_{x \in c_i} d(x, \mu_i) = \arg\min_{c} \sum_{i=1}^{k} \sum_{x \in c_i} \|x - \mu_i\|_2^2$$

where c_i is the set of points that belong to cluster i.

Get the number of clusters k
Initialize the centroids: $\mu_i =$ some value , $i = 1, ..., k$
Until convergence **do**
    Attribute the closest cluster to each data point:
    $c_i = \{j : d(x_j, \mu_i) \leq d(x_j, \mu_l), l \neq i, j = 1, ..., n\}$
    Set the position of each cluster to the mean of all data points belonging to that cluster:
    $\mu_i = \frac{1}{|c_i|} \sum_{j \in c_i} x_j, \forall i$
    $|c| =$ number of elements in c
End Loop

Figure A.1: Algorithm of K-mean Clustering

1. Using algorithm described in: http://www.onmyphd.com/?p=k-means.clustering

APPENDIX B

TABLES

B.1 Tables of Machine for AT module Simulations

Table B.1 shows the specifications of the random machines used to compare and evaluate the Accuracy metric of the AT module (Refer to Section 4.3.4).

Table B.1: Randomly Selected Machines for AT Module Accuracy Evaluation

Number	Machine Name	OS	CPU Arch	Memory	Disk	IP
1	1106	LINUX	INTEL	1,052 MB	2 MB	192.168.221.231
2	2664	SOLARIS5.10	SUN4u	537 MB	3 MB	192.168.154.166
3	3953	SOLARIS5.10	SUN4u	537 MB	1 MB	192.168.170.14
4	5805	NA	NA	537 MB	3 MB	192.168.154.107
5	7445	SOLARIS5.10	SUN4u	537 MB	1 MB	192.168.170.16
6	9574	LINUX	X86_64	1,054 MB	37 MB	192.168.221.137
7	1881	LINUX	INTEL	1,052 MB	2 MB	192.168.221.109
8	2723	LINUX	X86_64	519 MB	53 MB	192.168.154.183
9	3190	LINUX	INTEL	1,062 MB	36 MB	192.168.153.167
10	4521	LINUX	X86_64	1,048 MB	30 MB	192.168.154.210
11	8908	LINUX	INTEL	528 MB	45 MB	192.168.152.122
12	6139	LINUX	INTEL	528 MB	118 MB	192.168.170.110

B.1 Tables of Machine and Jobs for RE Module Simulations

Table B.2 shows the machines table including each machines attributes used for the RE simulation. These machines are randomly selected. All of the attributes are retrieved from the dataset except for Gfloph which is evaluated using Equation 5.1. dfpop_speed is the double precision floating point speed (ops/sec).
Table B.3 shows the randomly created jobs for the RE simulation. The Req Gfloph is the number of giga flop (floating point) operations per an hour the job needs to finish.

Table B.2: Randomly Selected Machines and their attributes for RE Module Simulation

Number	Machine Name	OS	Memory	Disk	dfpop_speed	Gfloph
1	1840	LINUX	1,024 MB	36 MB	1167454	8.4056
2	5933	LINUX	2,048 MB	250 MB	1114719	8.0259
3	4130	LINUX	2,048 MB	326 MB	995169	7.1652
4	1322	LINUX	512 MB	56 MB	984160	7.0859
5	9755	LINUX	1,024 MB	36 MB	1169648	8.4214
6	9103	LINUX	1,024 MB	100 MB	1212967	8.7333
7	7213	LINUX	3,072 MB	13 MB	856717	6.1683
8	1013	LINUX	768 MB	27 MB	771981	5.5582
9	4932	LINUX	384 MB	2 MB	855959	6.1629
10	5020	LINUX	512 MB	5 MB	516291	3.7172
11	3631	LINUX	512 MB	2 MB	506765	3.6487
12	1431	LINUX	1,024 MB	5 MB	466187	3.3565
13	6004	LINUX	1,024 MB	31 MB	843883	6.0759
14	6283	OSX	256 MB	24 MB	389056	2.8012
15	6304	LINUX	768 MB	8 MB	589217	4.2423
16	4671	LINUX	1,024 MB	35 MB	364701	2.6258
17	6139	LINUX	512 MB	118 MB	192707	1.3874
18	1435	SOLARIS	512 MB	4 MB	110334	0.7944
19	8706	LINUX	1,024 MB	228 MB	760732	5.4772
20	3452	SOLARIS	512 MB	1 MB	110532	0.7958
21	2014	SOLARIS	512 MB	5 MB	110285	0.7940
22	7107	SOLARIS	1,024 MB	185 MB	87976	0.6334
23	4192	LINUX	2,048 MB	330 MB	988010	7.1136
24	7411	SOLARIS	512 MB	2 MB	110367	0.7946

Table B.3: Randomly Selected Jobs and their requirements for RE Module Simulation

Number	Req CPU	Req Memory	Req Disk	Req Gfloph
1	1	512 MB	5 MB	100
2	1	1,024 MB	30 MB	144
3	1	1,024 MB	30 MB	200
4	1	512 MB	5 MB	144
5	1	512 MB	5 MB	100
6	1	2,048 MB	30 MB	200
7	1	2,048 MB	30 MB	200
8	1	1,024 MB	5 MB	200
9	1	1,024 MB	30 MB	144
10	1	512 MB	5 MB	144
11	1	2,048 MB	5 MB	100

BIODATA OF STUDENT

The student was born Kerman, Iran in year 1981. He finished his Diploma in Mathematics in 1998. He has a Bachelor degree in Computer Engineering from Bahonar University of Kerman, 2004. Before starting his Masters in University Putra Malaysia, he worked in the industry as a computer developer, project manager, cluster administrator and researcher. He was a member of the high performance computing research center (HPCRC) of Tehran's Amirkabir University of Technology (AUT) which at that time developed a supercomputer known among the world's top 500 supercomputers.

Recently he was involved in the development of the scientific gateway of the Academic Grid Malaysia project cooperating with NovaGlobal Pte Ltd as a grid portlet developer. He also cooperated in the deployment of USM's Hypercube cluster. His research interests include desktop grid computing and applied machine learning.

PUBLICATIONS

1. M. Y. Shafazand, R. Latip, A. Abdullah, and M. Hussin. 2014. A Model for Client Recommendation to a Desktop Grid Server *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, SE - 55, vol. 285, T. Herawan, M. M. Deris, and J. Abawajy, Eds. Springer Singapore, , pp. 491 – 498.