ON MEASURING THE ROBUSTNESS OF CLOUD COMPUTING SYSTEMS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Saeeda Usman

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Electrical and Computer Engineering

June 2015

Fargo, North Dakota

North Dakota State University
Graduate School

Title

On Measuring the Robustness of Cloud Computing Systems

By

Saeeda Usman

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Samee U. Khan
Chair

Jacob S. Glower

Sudarshan Srinivasn

Ying Huang

Approved:

06/04/2015
Date

Scott C. Smith
Department Chair

ABSTRACT

The diverse computing services offered by the cloud computing paradigm have escalated the interest in cloud deployment to a great extent. Cloud systems need to be resilient to uncertainties and perturbations. However, the perturbations in a cloud environment may cause the performance to degrade and violate the Service Level Agreements (SLAs). Therefore, it is imperative to adhere to the performance assurance by guaranteeing reliability in diverse and unexpected conditions. In our research, we focused on measuring and analyzing the robustness of a cloud based scheduling system. To mitigate the negative effects of the perturbations and uncertainties existing in the system working environment we present a robust resource allocation system. In our study, we focused on a two-step line of action: (a) measurement of robustness and (b) achieving an optimized Pareto front of the scheduler system is Cloud.

To address the aforesaid challenge and fulfill the required Quality of Service (QoS), this research work employs a robustness analysis of resource allocation schemes in cloud on the basis of multiple performance parameters. Due to the high number of parameters' comparison criterion, decision of the most robust allocation scheme is quite challenging. Therefore, a dimension reduction mechanism is employed to reduce the problem complexity. Thereafter, the resource allocation schemes are evaluated for guaranteeing the systemwide performance to ensure reliability and ascertain promising performance. The experimental results depict that the order of parameter selection in the reduction process has a significant impact on the selection of the most robust allocation scheme.

The performance demands of modern computing applications have led to an exponential increase in power density of on-chip devices. Not only the operational budget of the system has increased substantially, but also the temperature has experienced an alarming increase rate. The

aforementioned challenges necessitate the requirement of realizing efficient mapping methodologies to overcome the resource exploitation issue in Cloud computing. This study attempts the optimization of performance, power, and temperature of multi core systems by varying the frequency of operation of the core. Our proposed resource scheduler efficiently adheres to the optimized Pareto front to address the aforementioned challenges.

ACKNOWLEDGEMENTS

With immense gratitude I would like to thank ALLMIGHTY ALLAH for blessing me with the opportunity of attaining such a great achievement. I am deeply indebted to Him for all the knowledge, skills, strength, and unlimited blessings I have been bestowed. The completion of this degree could not have been possible without the persistent support of Allah (SWT).

I would like to thank my Advisor, Dr. Samee U. Khan for his sincere appreciation, encouragement, and support during the pursuance of my dissertation. He indeed has the substance and attitude of a genius that propagated a spirit of adventure for research. His respect is even more increased by the persistent guidance and help that he is always ready to offer.

I am very grateful to take this opportunity to thank Dr. Jacob S. Glower, Dr. Sudarshan Srinivasan, and Dr. Ying Huang for serving on my graduate committee. I am grateful to them for their guidance and encouragement. I am thankful to the Electrical and Computer Engineering staff members Jeffrey Erickson, Laura D. Dallman, and Priscilla Schlenker for all their help and kindness during my graduate studies and research at North Dakota State University.

Finally, I would especially like to thank the most important person in my life, my husband, who helped and supported me by all means to fulfill my career goals. It is due to his cooperation, patience, and support that I am able to achieve this milestone of attaining a Ph.D. degree with two very young kids.

Above all, I would like to thank my parents who endured very hard time, both physically and financially to make me the person I am today. Their unconditional love, encouragement, and faith have been a continuous source of tremendous support for me. Finally, I owe my heartiest thanks to my siblings, friends and colleagues who always facilitated me in the time of need.

DEDICATION

To my parents and husband.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Introduction

In this chapter, we aim to discuss the introduction of the research we have performed during Ph.D. We carried out our research on the improvement of robustness of a cloud based scheduling system for subdue the perturbations present in the system environment. In our research studies, we focused on the enhancement of resource allocation of tasks to a set of machines. In the first case a robustness measurement and analysis methodology is devised. Nevertheless, in the following case, we obtained a Pareto optimized set of solutions for the enhancement of a resource allocation system. Based on our study, we devise a formulation that unveils bounds on the desired objectives for the achievement of optimization in the system working environment. We analyzed that the frequency of operation when constrained to certain limit of operating domain can benefit the scheduler in optimizing the power and temperature. Therefore, by adhering to mapping configuration the resource utilization is adjusted dynamically.

1.2. Robustness Measurement of a Scheduling System

With the tremendous growth in demand of cloud deployment, the computing services offered by cloud providers are expected to guarantee effective performance along with resource provisioning. The cloud service providers, such as Google, Amazon, Yahoo, and Cisco aggregate the pool of computing resources to adjust the exponentially increasing demand of computing resources by enterprise businesses and scientific research areas [1.1]. To comply with the client defined Service Level Agreements (SLAs), the cloud infrastructure consolidates the computing and storage resources in an “on-demand” manner to preclude the high operational costs [1.2].

Perhaps the sharing of resources makes the cloud susceptible to perturbations and erroneous functionality [1.3]. Therefore, to address this impediment, the cloud service providers need to consider the uncertainty in the working environment and ascertain robustness to ensure the desired level of performance. Moreover, the cloud framework must orchestrate the resource consolidation such that the SLA is satisfied and the agreed level of Quality of service (QoS) is rendered.

The Information and Communication Technology (ICT) has witnessed an exponential increase in the adoption of cloud services in the recent years. According to the Gartner report published in January 2015, the cloud market is expected to reach $143 billion in 2015 reflecting 1.80 percent increase from 2014 [1.4]. The pervasive and convenient access to the cloud raises anomalies ranging from hardware bottlenecks to component failures that are challenging to predict and diagnose [1.1]. The aforementioned obstacles pose a serious threat to the performance and functionality of cloud [5]. Moreover, the shared pool of resources makes the cloud framework vulnerable to perturbations and failures [1.3]. Therefore, to achieve effective functionality, all of the above mentioned issues necessitate the assurance of robustness in the cloud framework.

Provision of a robustness guarantee is required to ensure proper functionality of cloud in the presence of uncertainties [1.5]. Most of the existing approaches define robustness as a measure of acceptable and expected operation in the presence of perturbations and uncertainties [1.3], [1.6]. The IEEE standard glossary of software engineering lexicon [1.7] defines robustness as "The degree to which the system or component can function correctly or as expected in the presence of invalid inputs or stressful environmental conditions." Various studies in literature recognize the adverse effect of uncertainties in the cloud's working environment that degrades the performance. Bilal et al. [1.3] performed an extensive analysis of the robustness metrics of data center network in the cloud infrastructure to improve service reliability and overall performance. Zhang et al. [1.8]

proposed a Vectorized Ordinal Optimization (VOO) approach to handle the uncertainties in the cloud resource allocation schemes. Nevertheless, the work presented here focuses on robustness measurement considering a multiparamter environment. The selection of a robust resource allocation in cloud emerges as a challenging problem when the range of parameters' evaluation increases to a high number [1.9]. Typically, researchers have been working on the problems considering a limited number of comparison parameters [1.6], [1.10]. However, when the parameter comparison criterion increases to a large number, say $n >> 0$, the selection of one unique solution becomes unachievable for the scheduler in cloud [1.11].

Due to the significance of robustness in a cloud framework, the presented research work hinges on prescribing a mechanism to measure robustness. The resource allocation schemes in the cloud are evaluated for the magnitude of robustness exhibited to procure metrics that render a promising performance. The evaluation is performed based on numerous parameters. The solution is approached by first reducing the problem complexity. The goal is to first efficiently employ the dimension reduction procedure to transform the data belonging to higher dimensions into a lower dimensional space. The dimension reduction process is to be performed such that the information pertaining to data properties is preserved. Intactness of data properties appear as a significant obstacle when the data lying in higher dimensions is reduced to its lower counterpart [1.12]. The data set after convergence is analyzed for the robustness measure.

In this paper, we explore the procedure of dimension reduction using a geometrical approach. The mathematical formulation for the robustness analysis of allocation schemes in a scheduling system. Data lying in a higher or n-dimensional (dim) hyperspace is projected on to a low- dimensional linear or non-linear space. The projection unveils low-dimensional structures that can be used for the data analysis as well as for data visualization [1.13]. Therefore, a feasible

solution to the above stated problem is to reduce the data at first place and then perform the comparison of the robustness. The key to convergence is a dimension reduction procedure [1.14]. The data is mapped onto a lower dimension space as a result of employing the reduction process. The dimension reduction approach employed in this work is a geometrically flavored procedure. A step wise dimension reduction is performed by taking projection and retaining the impact of the reduced coordinate. The geometrical reduced surface (distribution of data) attained as a result, retains a non-linear relation to the hypersurface it represents [1.12]. The reduced dimension version is subsequently evaluated for the robustness measure and the allocation schemes are then categorized on the basis of the robustness quantified. Based on the robustness measure a comparison among the allocation schemes is performed to find the most effective and suitable scheduling scheme. The immense advantage of the reduction incorporated robustness analysis besides low complexity is that we can guarantee robustness despite of the high number of performance features considered for the comparison.

1.3. Realization of Pareto Front for Cloud Scheduler

The dynamic and promising services delivered by Cloud computing paradigm have strikingly elevated the demand of Cloud deployment (models). The paradigm orchestrates the computing resources, such as the processing cores, I/O resource, and storage to meet "on demand" client requirements. The aforementioned characteristic of Cloud has extensively scaled the service offering to leverage and productize functionality. However, to ensure that the agreed Service Level Agreement (SLA) is met the Clouds needs to offer metering services to avoid resource exploitation.

To provide a single pane view of the resources status and achieve high levels of granular visibility, intelligent monitoring should by realized to track resource utilization. Due to the increase

in chip power density, the offered computing resources are prone to predicaments, such as hardware failure, low reliability, and insecure multi-tendency. Indeed, task completion is the foremost priority of schedulers in Cloud. Nevertheless, thermal management and power consumption hold pivotal importance in achieving high-end functionality. Moreover, cost minimization can be accelerated by avoiding over-provisioning of the aforementioned resources.

Recently, a wide range of hardware and software based technique [1.15]-[1.17] have been proposed to control the power consumption of Chip Multi-Processors (CMPs). Although the management schemes could effectively reduce power depletion, they incur performance overhead in the form of thermal runaway. Motivated by this fact, the work presented in this paper address the abovementioned issue by considering the run-time information. Therefore, frequent monitoring of core temperature and operating frequency is required to lower the risk of chip overheating. We provide a methodology to mitigate the violation of peak power and temperature constraints, respectively. The objective of this work is to optimize the cumulative performance of the resource allocation system. Intuitively, a convex optimization approach is devised to minimize the makespan, temperature, and power utilization of the scheduler. Our contribution circumvent the efficient management of power/temperature exploitation without comprising the task completion deadline. The solutions that adhere to all of the constraints of power, makespan, and temperature constitute to the set of efficient or Pareto optimized solutions. Despite of the contradicting nature of the objectives, we perform efficient mapping of resources to fulfill the end user demands without the violation of any timing constraints. The relevant Pareto front of high quality is obtained for the optimization of the three objectives.

1.4. References

[1.1] F. Gu, K. Shaban, N. Ghani, S. Khan, M. Rahnamay-Naeini, M. Hayat, and C. Assi, "Survivable cloud network mapping for disaster recovery support," IEEE Transactions on Computers, vol. PP, no. 99, pp. 1–1, 2014.

[1.2] B. Guan, J. Wu, Y. Wang, and S. Khan, "Civsched: A communication-aware inter-vm scheduling technique for decreased network latency between co-located vms," IEEE Transactions on Cloud Computing, vol. 2, no. 3, pp. 320–332, July 2014.

[1.3] K. Bilal, M. Manzano, S. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 1–1, Jan 2013.

[1.4] G. Group, Worldwide IT Spending Forecast, Jan 2015. [Online]. Available: http://www.gartner.com/newsroom/id/2959717

[1.5] C. Hewwit, "Orgs for scalable, robust, privacy-friendly client cloud computing," IEEE Internet Computing, vol. 12, no. 5, pp. 96–99, 2008.

[1.5] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim, "Measuring the robustness of a resource allocation," IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 7, pp. 630–641, 2004.

[1.7] IEEE Std 610.12-1990-Glossory of Software Engineering Technologies.

[1.8] F. Zhang, J. Cao, W. Tan, S. Khan, K. Li, and A. Zomaya, "Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud," IEEE Transactions on Emerging Topics in Computing, vol. 2, no. 3, pp. 338–351, Sept 2014.

[1.9] J. Apodaca, D. Young, L. Briceo, J. Smith, S. Pasricha, A. A. Maciejewski, H. J. Siegel, and S. B. B. Khemka, "Stochastically robust static resource allocation for energy minimization

with a makespan constraint in a heterogeneous computing environment," in 9th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 11, 2011, pp. 22–31.

[1.10] B. Dorronsoro, P. Bouvry, J. Caero, A. Maciejewski, and H. Siegel, "Multi-objective robust static mapping of independent tasks on grids," in 2010 IEEE Congress on Evolutionary Computation (CEC), July 2010, pp. 1–8.

[1.11] F. Zhang, J. Cao, K. Li, S. U. Khan, and K. Hwang, "Multi-objective scheduling of many tasks in cloud platforms," Future Generation Computer Systems, vol. 37, pp. 309 – 320, 2014.

[1.12] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in First IEEE Conference on Visualization, Oct 1990, pp. 361–378.

[1.13] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," SIAM Journal of Scientific Computing, vol. 26, no. 1, pp. 313–338, Jan. 2005.

[1.14] H. L. Yap, M. Wakin, and C. Rozell, "Stable manifold embeddings with structured random matrices," IEEE Journal of Selected Topics in Signal Processing, vol. 7, no. 4, pp. 720–730, Aug 2013.

[1.15] K. Bilal, A. Fayyaz, S. U. Khan, and S. Usman, "Power-aware resource allocation in computer clusters using dynamic threshold voltage scaling and dynamic voltage scaling: comparison and analysis," Cluster Computing, pp. 1–24, 2015.

[1.16] J. Kołodziej, S. U. Khan, L. Wang, and A. Y. Zomaya, "Energy efficient genetic-based schedulers in computational grids," Concurrency and Computation: Practice and Experience, 2012.

[1.17] A. Abbas, M. Ali, A. Fayyaz, A. Ghosh, A. Kalra, S. U. Khan, M. U. S. Khan, T. De Menezes, S. Pattanayak, A. Sanyal et al., "A survey on energy-efficient methodologies and

architectures of network-on-chip," Computers & Electrical Engineering, vol. 40, no. 8, pp. 333–347, 2014.

2. RELATED WORK

In this chapter we discuss some of the work that is related to the research we have performed during Ph.D.

2.1. Correlation between Robustness and Reliable Performance

In this section, we present some of the research works that pertains to the Robustness measurement addressed in literature. The framework proposed in the paper [2.1] is generic and focuses on handling data lying in higher manifolds. By using dimension reduction, we perform data convergence in a stage wise manner and then test the acquired result for robustness to attain high end effective performance. Maxwell *et al.* [2.2] and Ali *et al.* [2.3] proposed robustness metrics for the quantification of robustness for a given resource allocation environment. Ghoshal *et al.* [2.4] applied a data management strategy in distributed transient environments like cloud for handling both virtual machine failure and variations in network performance. The aforementioned technique is unable to handle and overcome failures that occur during run-time. Nevertheless, our methodology is more focused on achieving high level performance in the cloud environment to overcome the threats and challenges in an effective manner. Guaranteeing performance is of utmost importance in the implementation of a cloud paradigm.

Larsen *et al.* [2.5] used a syntactic transformation approach that employs classical analysis techniques and tools to achieve robustness. Moreover, to achieve the required QoS level authors in [2.6] applied a fuzzy control logic for the resource management. Nevertheless, the application of fuzzy control is effective only in systems with a simple architecture. To handle the resource allocation problem effectively for a variety of scenarios a great deal of knowledge about the rules and parameters involved is required and extensive simulation needs to be carried

out before designing fuzzy system. Macias *et al.* [2.7] proposed an SLA improvement strategy by utilizing a two way communication path between the market brokers and resource managers. The aforementioned technique improves the SLA violation only when prior knowledge about the reputation of the system is available. Otherwise no significant difference can be brought using this methodology. On the other hand, our methodology can be adapted for any possible scenario. Scheduling and managing of resource with QoS maintained according to the SLA specifications is a major challenge in a cloud computing environment. The perturbations present in the system environment make the aforementioned tasks even more challenging and opens new paradigms in resource scheduling of cloud. To accomplish the above mentioned goal, the researchers in [2.8] presented a scheduling heuristic that caters multiple SLA parameters. The parameters considered are limited to CPU time, network bandwidth, and storage capacity. However, performance parameters such as response time, temperature, and processing time are not considered in improving the system's performance.

Li *et al.* [2.9] proposed a customizable cloud model for resource scheduling. An additional aspect of trust is incorporated in the system architecture along with the QoS targets for performance up-gradation. Although the QoS parameters considered in this approach includes response time, bandwidth, storage, reliability, and cost. However, the QoS delivery is restricted to the average values of the above mentioned performance aspects. Moreover, guarantee of the service delivery is not provided despite of the predetermined confidence level. The aforementioned approaches may cater users' preferences, but are unable to guarantee a QoS satisfaction level according to the SLA requirements. The problem we deal here is different from the existing work since it takes into account multiple parameters to optimize the system's performance, despite of the uncertainty present in the system environment.

We employ the dimensionality reduction technique as our solution to handle the impact of a parameter number as high as n, where $n \gg 0$, while meeting the QoS requirements. Relative to the ambient dimension n, the dimensionality reduction techniques aim to extract concise information about the data lying in a high-dimensional space [2.10]. The data lying in higher manifolds can be converged using manifold-reduction techniques. Current state- of-the-art techniques for dimensionality reduction can be broadly bifurcated into linear and non-linear dimensionality reduction. The prior includes classical methods like Principal component Analysis (PCA) and Multi-Dimensional Scaling (MDS). Nevertheless, the linear techniques outperformed the non-linear techniques due to the incapability to handle non-linear data structures [2.11]. On contrary non-linear manifold learning techniques, such as the Locally Linear Embedding (LLE), Laplacian Eigenmaps, and Isomap are efficient in handling non-linear data structures. However, the aforementioned methodologies are computationally expensive to handle and are not scalable due to their time and memory complexity [2.12]. Nevertheless, we emphasize on preserving the critical relationship among the data-set elements and to discover critical information about the data preserved, under the mapping Φ keeping the computational cost minimum.

2.2. Pareto Front Optimization

A large number of hardware and software techniques, for example [2.13], [2.14] and [2.15] have proposed by researchers to improve the energy profile of multi-core systems. The traditional power saving strategies focus on scaling the voltage and frequency of the core to meet the allowable power level. However, temperature received less attention. Consequently, reliability and decrease in the life-time of the chip resulted as a trade-off. Therefore, researchers over the last decade, emphasize the need of *Dynamic Thermal Management (DTM)* [2.16] and [2.17] for safe chip operation and to reduced cooling cost.

The work presented by authors in [2.13]-[2.18] perform optimization of power consumption while guaranteeing the required performance. Nevertheless, the aforementioned methodologies optimizes the power and performance, but during the optimization. Authors in [2.19] speculates the chip thermal management requirement and devised methodologies to attain the chip temperature optimization. In Ukhov *et. Al* [2.20] the authors propose a Steady State Dynamic Temperature Profile (SSTDP) to realize temperature-aware reliability model. The technique considers mitigating the thermal cycling failure. However, transient faults and their management is not catered. Moreover, power optimization is not entertained while achieving reliability. Significantly, different from the above listed work, this research work explore the scheduling decision space to optimize the performance of multi-core system. The temperature and power utilization is capped and dynamically adjusted while meeting the performance requirement of the system to generate a set of Pareto optimized solutions.

2.3. References

[2.1] K. Bilal, M. Manzano, S. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 1–1, Jan 2013.

[2.2] P. Maxwell, A. A. Maciejewski, H. J. Siegel, J. Potter, G. Pfister, J. Smith, and R. Friese, "Robust static planning tool for military village search missions: model and heuristics," The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, vol. 10, no. 1, pp. 31–47, 2013.

[2.3] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim, "Measuring the robustness of a resource allocation," IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 7, pp. 630–641, 2004.

[2.4] D. Ghoshal and L. Ramakrishnan, “Frieda: Flexible robust intelligent elastic data management in cloud environments,” in 2012 SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC), Nov 2012, pp. 1096–1105.

[2.5] K. G. Larsen, A. Legay, L.-M. Traonouez, and A. Wsowski, “Robust synthesis for real-time systems,” Theoretical Computer Science, vol. 515, pp. 96–122, Jan. 2014.

[2.6] J. Rao, Y. Wei, J. Gong, and C.-Z. Xu, “Qos guarantees and service differentiation for dynamic cloud applications,” IEEE Transactions on Network and Service Management, vol. 10, no. 1, pp. 43–55, March 2013.

[2.7] M. Macias, J. Fito, and J. Guitart, “Rule-based sla management for revenue maximization in cloud computing markets,” in International Conference on Network and Service Management (CNSM), Oct 2010, pp. 354–357.

[2.8] V. Emeakaroha, I. Brandic, M. Maurer, and I. Breskovic, “Sla-aware application deployment and resource allocation in clouds,” in 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW), July 2011, pp. 298–303.

[2.9] W. Li, Q. Zhang, J. Wu, J. Li, and H. Zhao, “Trust-based and qos demand clustering analysis customizable cloud workflow scheduling strategies,” in IEEE International Conference on Cluster Computing Workshops (CLUSTER WORKSHOPS), Sept 2012, pp. 111–119.

[2.10] A. Inselberg and B. Dimsdale, “Parallel coordinates: a tool for visualizing multi-dimensional geometry,” in First IEEE Conference on Visualization, Oct 1990, pp. 361–378.

[2.11] L. K. Saul and S. T. Roweis, “Think globally, fit locally: Unsupervised learning of low dimensional manifolds,” Journal of Machine Learning Research, vol. 4, pp. 119–155, Dec. 2003.

[2.12] A. Najafi, A. Joudaki, and E. Fatemizadeh, “Nonlinear dimensionality reduction via path-based isometric mapping,” Machine Learning, vol. 4, pp. 119–155, 2013

[2.13] K. Bilal, A. Fayyaz, S. U. Khan, and S. Usman, “Power-aware resource allocation in computer clusters using dynamic threshold voltage scaling and dynamic voltage scaling: comparison and analysis,” Cluster Computing, pp. 1–24, 2015.

[2.14] J. Shuja, K. Bilal, S. A. Madani, and S. U. Khan, “Data center energy efficient resource scheduling,” Cluster Computing, vol. 17, no. 4, pp. 1265–1277, 2014.

[2.15] D. Kliazovich, P. Bouvry, and S. U. Khan, “Dens: data center energy-efficient network-aware scheduling,” Cluster computing, vol. 16, no. 1, pp. 65–75, 2013.

[2.16] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, “Temperature-aware microarchitecture: Modeling and implementation,” ACM Trans. Archit. Code Optim., vol. 1, no. 1, pp. 94–125, Mar. 2004.

[2.17] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini, and G. De Micheli, “Temperature control of high-performance multi-core platforms using convex optimization,” in Design, Automation and Test in Europe, 2008. DATE’08, 2008, pp. 110–115.

[2.18] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-z. Xu, and A. Zomaya, “Energy-aware parallel task scheduling in a cluster,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1661–1670, 2013.

[2.19] J. Zhou and T. Wei, “Stochastic thermal-aware real-time task scheduling with considerations of soft errors,” Journal of Systems and Software, 2015.

[2.20] I. Ukhov, M. Bao, P. Eles, and Z. Peng, “Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems,” in Proceedings of the 49th Annual Design Automation Conference, 2012, pp. 197–204.

3. ON MEASURING THE ROBUSTNESS OF CLOUD COMPUTING SYSTEMS[1]

This paper is submitted to *IEEE Transactions on Parallel and Distributed Systems*. The authors of the paper are Saeeda Usman, Kashif Bilal, Samee U. Khan, Keqin Li, and Albert Y. Zomaya.

3.1. Introduction

With the tremendous growth in demand of cloud deployment, the computing services offered by cloud providers are expected to guarantee effective performance along with resource provisioning. The cloud service providers, such as Google, Amazon, Yahoo, and Cisco aggregate the pool of computing resources to adjust the exponentially increasing demand of computing resources by enterprise businesses and scientific research areas [3.1]. To comply with the client defined Service Level Agreements (SLAs), the cloud infrastructure consolidates the computing and storage resources in an "on-demand" manner to preclude the high operational costs [3.2]. Perhaps the sharing of resources makes the cloud susceptible to perturbations and erroneous functionality [3.3]. Therefore, to address this impediment, the cloud service providers need to consider the uncertainty in the working environment and ascertain robustness to ensure the desired level of performance. Moreover, the cloud framework must orchestrate the resource consolidation such that the SLA is satisfied and the agreed level of Quality of service (QoS) is rendered.

[1] The material in this chapter was co-authored by Saeeda Usman Kashif Bilal, Samee U. Khan, Keqin Li, and Albert Y. Zomaya. Saeeda Usman had primary responsibility for conducting experiments and collecting results. Saeeda Usman was the primary developer of the conclusions that are advanced here. Saeeda Usman also drafted and revised all versions of this chapter.

The Information and Communication Technology (ICT) has witnessed an exponential increase in the adoption of cloud services in the recent years. According to the Gartner report published in January 2015, the cloud market is expected to reach $143 billion in 2015 reflecting 1.80 percent increase from 2014 [3.4]. The pervasive and convenient access to the cloud raises anomalies ranging from hardware bottlenecks to component failures that are challenging to predict and diagnose [3.1]. The aforementioned obstacles pose a serious threat to the performance and functionality of cloud [3.5]. Moreover, the shared pool of resources makes the cloud framework vulnerable to perturbations and failures [3.3]. Therefore, to achieve effective functionality, all of the above mentioned issues necessitates the assurance of robustness in the cloud framework.

Provision of a robustness guarantee is required to ensure proper functionality of cloud in the presence of uncertainties [3.5]. Most of the existing approaches define robustness as a measure of acceptable and expected operation in the presence of perturbations and uncertainties [3.3], [3.6]. The IEEE standard glossary of software engineering lexicon [3.7] defines robustness as "The degree to which the system or component can function correctly or as expected in the presence of invalid inputs or stressful environmental conditions." Various studies in literature recognize the adverse effect of uncertainties in the cloud's working environment that degrades the performance. Bilal et al. [3.3] performed an extensive analysis of the robustness metrics of data center network in the cloud infrastructure to improve service reliability and overall performance. Zhang et al. [3.8] proposed a Vectorized Ordinal Optimization (VOO) approach to handle the uncertainties in the cloud resource allocation schemes. Nevertheless, the work presented here focuses on robustness measurement considering a multiparamter environment. The selection of a robust resource allocation in

cloud emerges as a challenging problem when the range of parameters' evaluation increases to a high number [3.9]. Typically, researchers have been working on the problems considering a limited number of comparison parameters [3.6], [3.10]. However, when the parameter comparison criterion increases to a large number, say n >> 0, the selection of one unique solution becomes unachievable for the scheduler in cloud [3.11].

Due to the significance of robustness in a cloud framework, the presented research work hinges on prescribing a mechanism to measure robustness. The resource allocation schemes in the cloud are evaluated for the magnitude of robustness exhibited to procure metrics that render a promising performance. The evaluation is performed based on numerous parameters. The solution is approached by first reducing the problem complexity. The goal is to first efficiently employ the dimension reduction procedure to transform the data belonging to higher dimensions into a lower dimensional space. The dimension reduction process is to be performed such that the information pertaining to data properties is preserved. Intactness of data properties appear as a significant obstacle when the data lying in higher dimensions is reduced to its lower counterpart [3.12]. The data set after convergence is analyzed for the robustness measure.

In this paper, we explore the procedure of dimension reduction using a geometrical approach. Data lying in a higher or n-dimensional (dim) hyperspace is projected on to a low-dimensional linear or non-linear space. The projection unveils low-dimensional structures that can be used for the data analysis as well as for data visualization [3.13]. Therefore, a feasible solution to the above stated problem is to reduce the data at first place and then perform the comparison of the robustness. The key to convergence is a dimension reduction procedure [3.14]. The data is mapped onto a lower dimension space as a result of employing the

reduction process. The dimension reduction approach employed in this work is a geometrically flavored procedure. A step wise dimension reduction is performed by taking projection and retaining the impact of the reduced coordinate. The geometrical reduced surface (distribution of data) attained as a result, retains a non-linear relation to the hypersurface it represents [3.12]. The reduced dimension version is subsequently evaluated for the robustness measure and the allocation schemes are then categorized on the basis of the robustness quantified.

The salient contributions of the paper are:

- We provide the mathematical formulation for the robustness analysis of allocation schemes in a scheduling system. Based on the robustness measure a comparison among the allocation schemes is performed to find the most effective and suitable scheduling scheme. The most compelling attribute of the comparison is the multi-dimensional nature of the resource allocation schemes (the detail can be found in Section 3.4). To reduce the complexity of the comparison, we employ a projectivity based dimension reduction process.
- Using the dimension reduction procedure, particularly, the geometrical reduction methodology, we transform the multi-parameter high dimensional data into a low-dimension workspace. The advantage of the geometric reduction process besides the low complexity is that the decision of the most appropriate and robust allocation schemes for a scheduler becomes fault resilient in a multi-dimensional environment. We will provide the details of above stated contribution in Section 3.5.
- Because of the uncertainties in the cloud working environment, we expect that the estimated parameter values may deviate from the actual. Therefore, the formulation

takes into consideration the perturbations that might exist in the systems' operational environment. A variation in the expected feature value is introduced to make the scheduler aware of imprecision. Moreover, besides merely reducing the dimensionality, we ensure that the mapping process should include the effect of uncertainty in the input parameters (see Theorem 1). Furthermore, the recovery of original data is also guaranteed by the use of tranversality property of projective maps.

- We employ rigorous mathematical machinery to devise a robustness boundary (see Theorem 2 and Section 3.3). We show that the reduced parameter affects the orientation of the data (hypersphere) undergoing the reduction process thereby affecting the robustness measure (see Section 3.4). Followed by the dimension reduction procedure, a robustness measurement methodology is employed that narrows down the choice of the best allocation scheme. The immense advantage of the reduction incorporated robustness analysis besides low complexity is that we can guarantee robustness despite of the high number of performance features considered for the comparison.
- We elaborate the dimension reduction based robustness measurement methodology with the help of an example scenario to demonstrate the system functionality. The details of the methodology are provided in Section 3.5.
- To benchmark the performance of robustness based cloud scheduling system, both synthetic and real-world workloads are used. The details are presented in Section 3.7. A comprehensive experimental evaluation is performed to observe the impact of reduction order selection. The results depict that the selection of reduction order holds a pivotal role in the choice of most robust allocation scheme. Based on the reduction

order, the choice of suitable scheduling scheme is preferentiated. The aforementioned findings can be used to gear the user specified SLA in a cloud based resource scheduler.

The remainder of the paper is organized as follows. Section 3.2 presents preliminary, mathematical concepts, and terminologies employed in the paper. The problem formulation and proposed methodology is provided in Section 3.3. Section 3.4 provides details of the dimension reduction procedure along with the verification of the proposed model, followed by an example in Section 3.5. Section 3.6 presents discussion on the customized version of dimension reduction procedure to fulfill the SLA. Performance evaluation and simulation results are presented in Section 3.7. Section 3.8 discusses the related work, and Section 3.9 concludes the paper.

3.2. Preliminaries

Before a detailed discussion of the dimension reduction employed analysis of resource allocation robustness, we present a brief description of certain preliminary concepts. The significant steps of the methodology presented in this study are depicted in Fig. 3.1. Moreover, the mathematical model and prototypes used in our work are discussed below to help the readers get a better understanding of the paper.

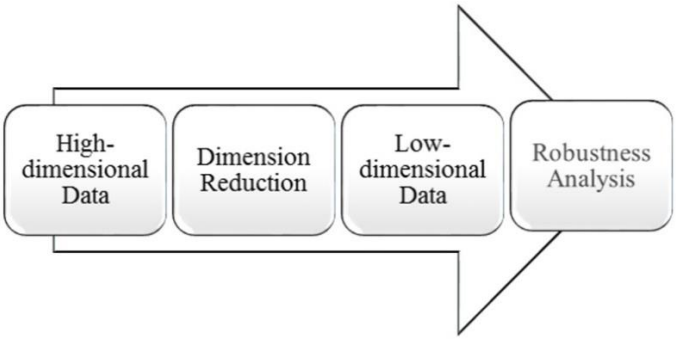


Fig. 3.1. Flow of procedural steps for robustness analysis

3.2.1. n-Dimensional Sphere

The purpose of this sub-section is to provide an overview of the amount of information exhibited by elements belonging to higher dimension space. To quantize the multivariate dataset a dimension reduction procedure is employed for convergence purpose. Data in the hyper sphere is collapsed down to low-dimensional manifolds using projective mapping and transformations. Consider Fig. 3.2 representing a 5-dimensional sphere in R^6. The variables $\alpha, \beta, \gamma, \Phi, \eta$ corresponds to the coordinates of the sphere in which the sphere is lying. Each co-ordinate signifies information regarding the data points of the sphere.

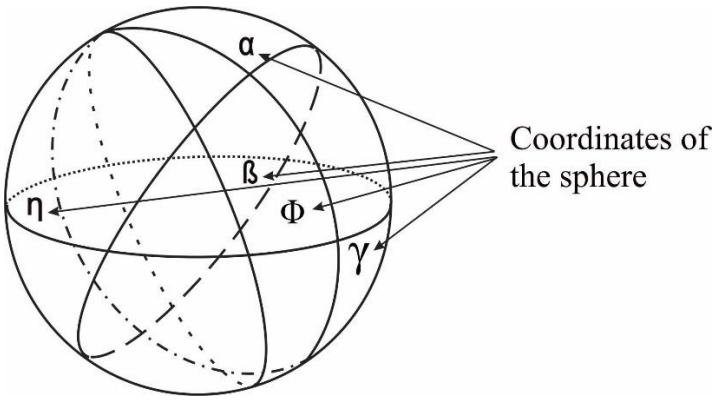


Fig. 3.2. A 5-Dimensional sphere with coordinates labeled

Definition 1 [3.7]. *The n-dimensional sphere of radius r with center at the point (a_1, a_2, …., a_n) is a collection of all points and, the co-ordinates x = (x_1, x_2, …., x_n), such that:*

$$(x_1 + a_1)^2 + (x_2 + a_2)^2 + \cdots + (x_n + a_n)^2 = r^2, \tag{3.1}$$

or

$$\sum_{i=1}^{n} (x_1 + a_1)^2 = r^2 \qquad \forall_i = 1, 2, \ldots., n. \tag{3.2}$$

A projective transformation of the hypersphere is an immersion of R^N into R^M, such that $M < N$. Each projection is a reduced map of the original sphere and is called a *submanifold*. The values *M* and *N* correspond to the dimensions of the sphere in the higher dimensional space. The projection in *M* is a submanifold of *N*.

3.2.2. N-Dimensional Reduction

Dimension reduction is a popular and efficient information transformation (convergence) technique used for the mapping of data to a lower dimensional space [3.15]. Moreover, besides providing a method for data visualization, the dimension reduction mechanism is also used for extracting key low dimensional features and attain better models for inference [3.14]. Based on reduction in data, a projective approach of dimension reduction is employed to reduce complexity of high dimensional data set. Numerous projections (linear and non-linear) of data points are mapped from a higher-dimensional space to a lower counterpart to analyze and process the data. Consider the following set of equations that represents mapping of *n*-dimensional surface y to a three-dimensional projection, such that:

$$y = (x_1, x_2, \ldots, x_n), \tag{3.3}$$

$$\pi_{ijk}, \tag{3.4}$$

$$\Phi = (x_i, x_j, \ldots, x_k), \tag{3.5}$$

where Φ is the projection of *y* in (*i*, *j*, *k*) coordinates and π_{ijk} is the projection plane. Figure 3.3 represents a mapping of a sphere in R^4 to a sphere in R^3. In reduction process the projections are taken across various planes, removing one co-ordinate axis in each of the projections. For example, in Fig. 3.3 three projections are taken for mapping a sphere from R^4 to R^3. In each of the projections one axis is set to zero. We take one projection across the *xyz*-plane setting $w=0$, the second one across *wyz*-plane, setting $x=0$, and the last one across the *wxy*-plane,

setting $z=0$. Each projection contains the information about the co-ordinates retained. Distribution of data points across the projected planes reveals information about the properties of the hyper sphere. Each submanifold retains information of three planes and removes information about one of the planes. The dimension information loss can be retrieved by taking a combination (possible non-unique) of two planes. That is, using the following;

$$dim(D_1 + D_2) = dim\, D_1 + dim\, D_2 - (D_1 \cap D_2) \tag{3.6}$$

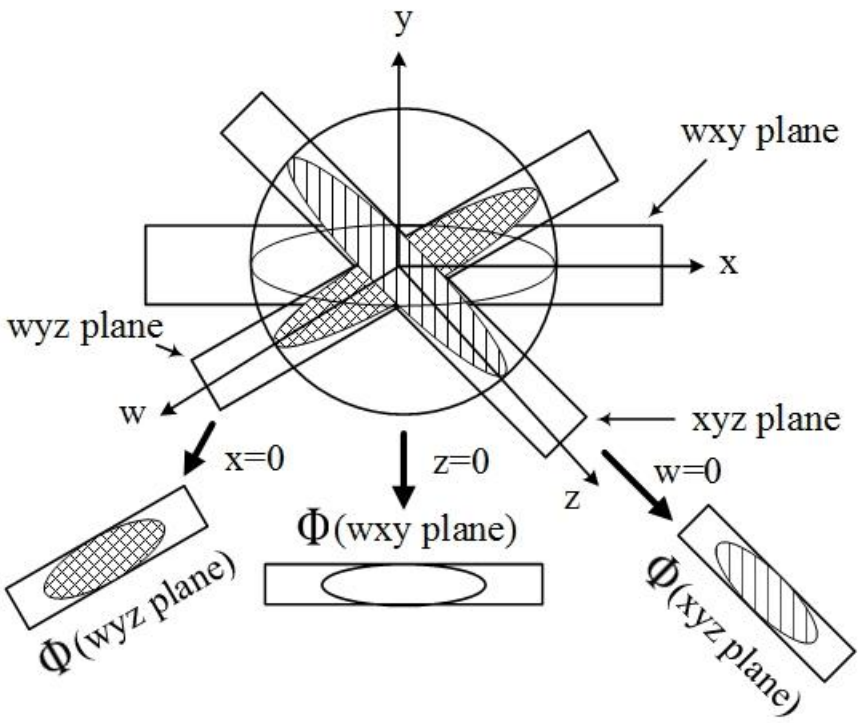


Fig. 3.3. A 4-D sphere projected on R^3 planes

Definition 2 (Transversality Property). *Suppose C, D are regular submanifolds of B, such that* $C, D, \subset B$ *and satisfies the property:*

$$TS_P C + TS_P D = TS_P B, \tag{3.7}$$

where $T\,S$ represents the transversality parameter and P represents all of the points that belong to $C \cap D$, normally, a linear combination of C and D. Transversality property is use to relate sub-spaces of the same vector space. The submanifolds C and D are then called as transverse submanifold and we denote the relationship as $C \pitchfork D$. The glossary of notations used in the paper are listed in Table 3.1.

Table 3.1. Glossary of notations

Variable	**Meaning**	**Variable**	**Meaning**
ℓ	Set of co-ordinates	τ_i	i-th resource allocation scheme
π	Projection mapping parameter	φ	Set of perturbations
Φ	Projection	ϵ_i	Allowable variation in Ω_i
a	Center point of co-ordinates	ϵ_i^{min}	Minimum variation allowed in Ω_i
r	Radius of n-sphere	ϵ_i^{max}	Maximum variation allowed in Ω_i
T_p	Transversality Parameter	m_{ij}	Mapping function for Ω_i to φ_j
$\pitchfork$	Transversality Relationship	ρ_μ	Robustness radius
Ω	Set of performance parameters	L_p	L_p-norm distance function

3.2.3. Transversality

The sub-spaces $P, Q \subset V$, where V, represents a vector space, are said to be transversal, when there exists a vector V in which every vector can be expressed as:

$$dim\, V = dim\, P + dim\, Q - dim\, P \cap Q. \tag{3.8}$$

Given that a linear combination of vectors in P and Q results in vectors that belong to the vector space V.

Proposition 1 [3.16]:

If $K, L \subset M$ are transverse regular submanifolds then $K \cap L$ also a regular submanifold of dimension $dim\, K + dim\, L - dim\, M$.

Let X and Y be the smooth submanifolds of a finite dimensional vector space O, then according to the following theorem a surjective map of the two manifolds is stable.

Theorem 1 (Stability of bounded projections). *If $X_{(n,\, p)}$ is in the acceptable range and X is bounded, then all linear maps from X to Y are stable.*

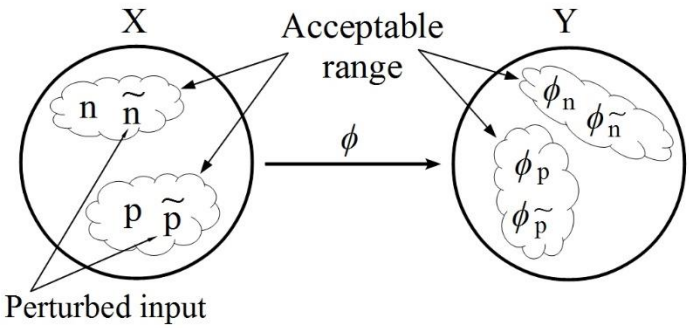


Fig. 3.4. A surjective map of X to Y with perturbed inputs

Proof. Let n and p be two points in the submanifold $X \subset O$, where O is a vector space. If a projection, Φ, of X is taken, such that the projection is continuous. Then for a perturbation of amount, $\epsilon > O$ selected for X there exists a slight deviation, $\delta > 0$, in the projected value, such that:

$$||(n,p) - (n^{\sim}, p^{\sim})|| < \delta \Rightarrow ||\Phi(n,p) - \Phi(n^{\wedge}, p^{\wedge})|| < \epsilon. \qquad (3.9)$$

The variables ϵ and δ in Eq. 3.9 represents negligible values, slightly greater than zero. Figure 3.4 represents a mapping of X to Y. To deal with the uncertainty in the system environment,

perturbation in input parameters is considered. The perturbed inputs are represented by ~ in Fig. 3.4. Figure 3.4 depicts that the worst case scenario does not cause any trouble as far as the perturbation is within the acceptable range.

Theorem 1 indicates a certain "stability" of the two projected points (specifically, when the preservation of ambient distances is desired) during the projectivity process. The stability guaranteed is useful in recovering the original sphere using the random projections. Moreover, a stable system ensures that a slight difference in expected conditions does not produce a remarkable disturbance in the system. Suppose, *a* and *b* be two data points lying in the original sphere ℓ, and let ℓ^* be the projected lower dimension image of ℓ,

$$\ell = arg\ min||a - b||_p, \tag{3.10}$$

$$a, b \in \ell. \tag{3.11}$$

Supposing that *a*, *b* are uniquely defined, then the projected space, ℓ^* is,

$$\ell^* = arg\ min||\Phi a - \Phi b||_p, \tag{3.12}$$

$$a, b \in \ell. \tag{3.13}$$

The distance between the data points projected as a result of the projectivity operation is almost similar to the corresponding difference in the original vector space. Therefore a satisfactory recovery of the original problem (planes/data) is assured.

3.2.4. Robustness Metric

Robustness can be defined as the degree to which a system can function correctly despite of uncertainties in the system parameters [3.17]. The salient steps for attaining the robustness metric includes: **(a)** identification of the performance attributes that need to be preserved, **(b)** perturbation variables, **(c)** effect of the perturbations on the system performance, and **(d)** an analysis process to ascertain the robustness. The above mentioned procedure is

illustrated in detail with the help of an example from the cloud paradigm in Section 3.4. The goal is to analyze and compare the allocation schemes and the scheme that despite changes in various parameters delivers reliable functionality is marked as the more robust scheme.

3.3. Problem Formulation and Proposed Methodology

The system model under consideration is a resource allocation computing system that adheres to a set of performance parameters required to make the system performance robust. Based on the parameters identified for efficient performance the desired level of SLA needs to be met by the cloud platform. The QoS performance features comprise of parameters, such as the energy consumed, makespan, temperature, and network delay for an allotted set of tasks. To ensure that the SLA is maintained, the acceptable divergence from the expected values must be within a tolerable range. The perturbations are caused due to various factors, such as estimation error, hardware failure, and network delay. Consequently, the system parameters encounter inaccuracies in the estimated values and the actual ones that makes the overall system failure prone.

Let Ω represents the set of performance parameters that the system must take into account. Each element of the set $\Omega(\Omega_i \in \Omega)$, must be bounded in variation to comply with the system requirements. We consider a resource allocation, τ, that needs to be evaluated on certain performance parameters. For the proposed resource allocation, Ω contains the following elements:

Ω = {makespan, queue waiting time, turnaround time, response time, temperature}.

Figure 3.5 depicts a resource allocating cloud based system architecture. Level 1 represents the consumers utilizing the services of the cloud. The requirements and expectations are considered at Level 2 that is perturbation prone. The elements of set Ω are depicted in the

Level 3 of Fig. 3.5. Performance is categorized on the basis of adherence to the elements of set Ω. The cloud at Level 4 ensures that clients are provided with high-performance functionality. Due to the multiple performance features, the decision of finding the most robust resource allocation is a complex endeavor. Consider the case when a categorization of more robust scheme is to be made by the scheduling system in the cloud. Figure 3.6, for instance, represents the response of two resource allocation schemes to each of the performance features depicted in set Ω. The resource allocation schemes are represented by the symbols $*$ and •. To find the more robust and suitable resource allocation scheme out of the two aforementioned allocation schemes is a difficult task. Therefore, to reduce the complexity of the problem in hand, we perform a dimension reduction procedure. The resource allocation that is more successful in preserving most of the performance attributes to the desired domain that promises effective functionality is regarded as the more robust.

As discussed earlier in Section 3.2, that the dimension reduction mechanism will employ a series of projections. The reduction procedure maps a set of data points $S \subset R^n$ to $f(s)$, represents the dimensionally reduced subspace of S. The projection to a four co-ordinate plane is given as: $R^n \rightarrow R^4$

$$\pi_{ij}(a_1, a_2 \ldots\ldots, a_n = \big(a_i, a_j, a_k, a_l\big). \tag{3.14}$$

Mathematically, a diagnostics use of projections is performed to reduce the problem complexity and then the robustness of each of the resource allocation scheme depicted in Fig. 3.6.

Theorem 2 (Robustness boundary). *Let* $S \subset R^n$ *be a set of data points and* $\hat{S} = f(s)$. Then $x \in S$ is robust if and only if under some projection π_{ij} each entry satisfies $|a_i| < 1$.

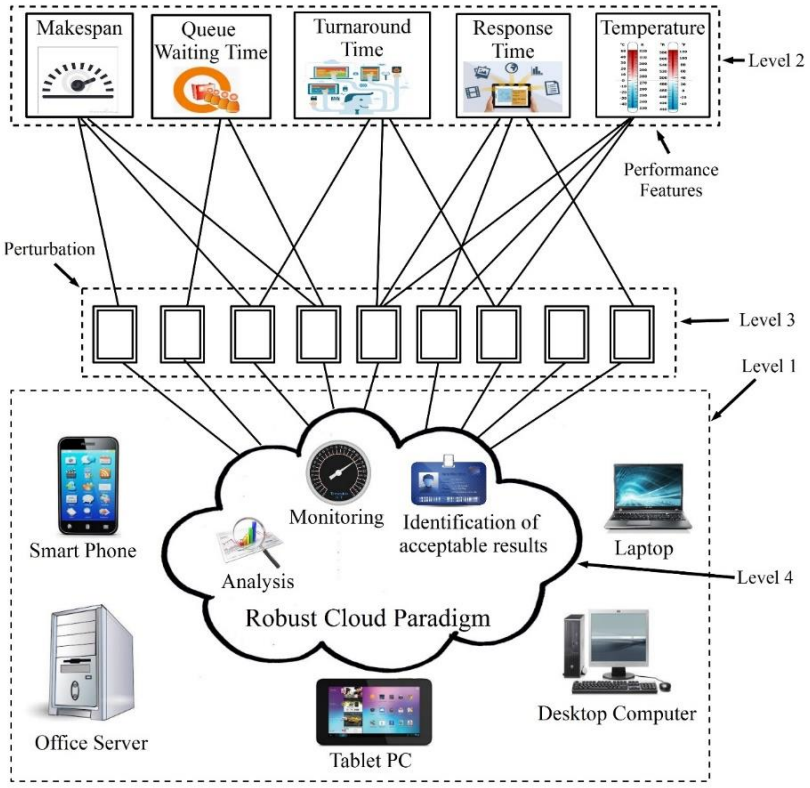


Fig. 3.5. Robustness architecture in a cloud paradigm

Proof. $x \in S$ is not robust, if and only if there exists an entry in $f(x)$ with absolute value greater than 1. Such a point will lie outside of the unit cube I_1^n. Therefore, there must exist at least one projection such that the projection $\pi_{ij}(a)$ lies outside the cube I_1^2. On the other hand, any point with $\pi_{ij}(a)$ outside the cube I_1^2 must stem from a data point that lies outside I_1^n, as some entry will have absolute value larger than 1.

If for some point $a \in \hat{S}$ the projection $\pi_{ij}(a)$ lies outside the cube I_1^2, then the point a is not robust. In particular, any point that has a projection outside of the robustness sphere boundary is said to be non-robust and vice versa. Moreover, the procedure can be reversed

to regain the original data set [3.16]. Furthermore, adding the labels to the data would be beneficial in retaining the original data by tracking the data labels throughout the procedure.

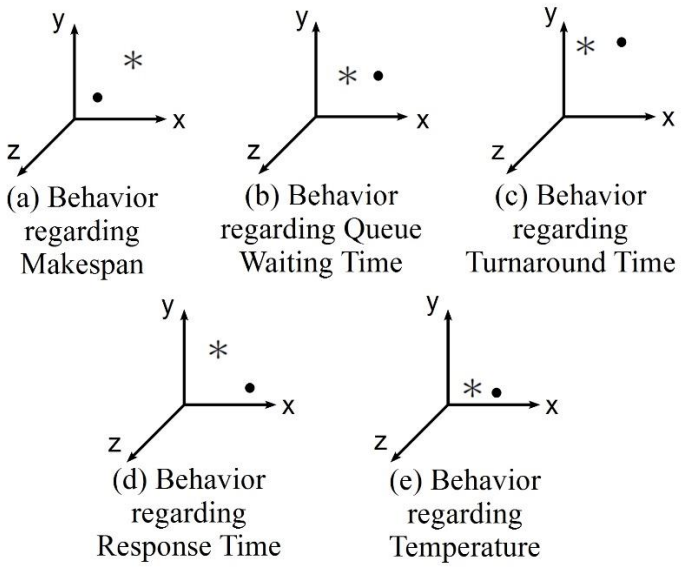


Fig. 3.6. Comparison of two resource allocations for five different performance features

3.4. Perturbations and Robustness Analysis

The environmental factors in which a cloud computing paradigm works are susceptible to fluctuate from its expected values [3.3]. Therefore, a change in the expected conditions is possible in a cloud working environment. In this study, we are considering multiple perturbations that affect the performance of the cloud. The perturbation when crosses a bound of the maximum value of the tolerable limit, the system might produce results that are undesirable. The system is consequently assumed to be operating in a non-robust infrastructure.

The robustness analysis is performed by identifying all of the system parameters that affect the desired QoS. These parameters are called as the uncertainty parameters and are represented by the vector set phi Φ in the paper. The set of uncertainties is represented by vector

phi, φ. The elements of φ may be heterogeneous (that is, φ_1 is energy value and φ_2 is the value of temperature). The variation in the system performance features Ω_i is bounded by $< \epsilon_i^{min}$, $\epsilon_i^{max} >$. The boundary of the set Ω_i is considered as < 0, 1.3 × *(estimated feature s value)* >. For simplification purpose, we consider the perturbations to be independent of each other. In a cloud computing paradigm perturbation vector, φ can assume the following expected attributes:

φ = {Machine failure, VM failure, estimation error (expected deviation from original values of parameters)}.

Different perturbation parameters make different impact on the efficient working of the system. For illustration purpose, the system behavior in response to a single perturbation parameter is observed at a time. Each individual perturbation causes the system performance to violate from normal operation. The combined effect of all of the perturbations (such that $\varphi_i \, \epsilon \, \varphi, \forall_i$) affecting a performance parameter is shown in Fig. 3.7. The dotted lines illustrate the outlier uncertainty expected across each individual parameter. The outer n-sphere depicts a net effect of three parameter perturbations across the system. The *n*-sphere is bounded by a *n-Dim* convex surface such that the surface bounds the variation of the *n*-sphere. Every point on the *n*-sphere has a minimum distance from the *n-Dim* surface. The acceptable values of parameters are identified by adherence to the bounds of variations.

The perturbations are mapped onto the co-ordinate axis. Each of the perturbation is allowed to vary in a limited range such that the performance remains in tolerable range. As discussed earlier, a *n-Dim* surface will take the form of an *n*-sphere that will bound the maximum value of the underlying spheres to an acceptable or desired range.

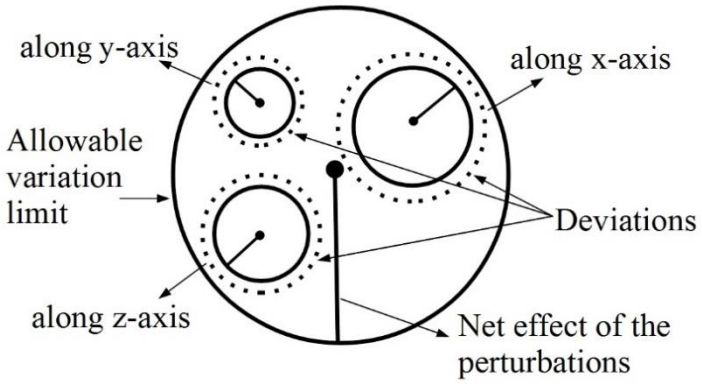


Fig. 3.7. Effect of all of the perturbations on the performance features

3.4.1. Robustness Objectives

In the presented work, robustness is a reflection and quantification of the allowable inaccuracies before the system exceeds the bounds of expected value of operation. Mathematically, for every performance feature $\Omega_i \, \epsilon \, \Omega$, the boundary values of perturbation must lie within the bounds of ϵ_i^{min} and ϵ_i^{max}.

To investigate a relationship between the performance features Ω and perturbation parameters φ, a mapping function m_{ij} is introduced. The m_{ij} maps $\Omega_i \, \epsilon \, \Omega$ to $\varphi_j \, \epsilon \, \varphi$, such that $\Omega_i = m_{ij}(\varphi_j)$. Thereafter, we observe the uncertainty parameters in such a way that if the system performance violates the desired range of operation, then the variation in values of the φ_i is recorded. If φ_j is assumed to be a discrete variable, then the boundary values, $m_{ij}(\varphi_j) = \epsilon_i^{min}$ and$m_{ij}(\varphi_j) = \epsilon_i^{max}$, correspondence to the nearest values that is enclosed by the boundary limits. The aforementioned relationships bifurcates the region of robust operation from the non-robust one. The goal is to find the slight uncertainty in φ_j that causes any of the performance feature $\Omega_i \, \epsilon \, \Omega$ to surpass the limits$< \epsilon_i^{min}, \epsilon^{max} >$, desired for a robust functionality of the system.

More specifically, let φ_j^{orig} of φ_j is assumed as the estimated value of orientation. Because of some inaccuracies in the predicted environmental changes, the resultant value of φ_j may vary from the anticipated value. Due to the fact that we are dealing with multiple parameters, the trade off in φ_j can be "multi-dimensional" and thus the resulting shift from assumed values can occur in different directions. Assuming a system where no prior information about distribution layout, the variable φ_j can demonstrate any value. Figure 3.8 depicts a simplified version of the problem under discussion. The robustness concept is illustrated for a single feature Ω_i, and a three element uncertainty vector $\varphi_j \in R^3$. The sphere presented in Fig. 3.8 plots the boundary points for the given application, such that:

$$\frac{\varphi_j}{m_{ij}(\varphi_j)} = \epsilon_i^{max} \; and \; \frac{\varphi_j}{m_{ij}(\varphi_j)} = \epsilon_i^{min}. \tag{3.15}$$

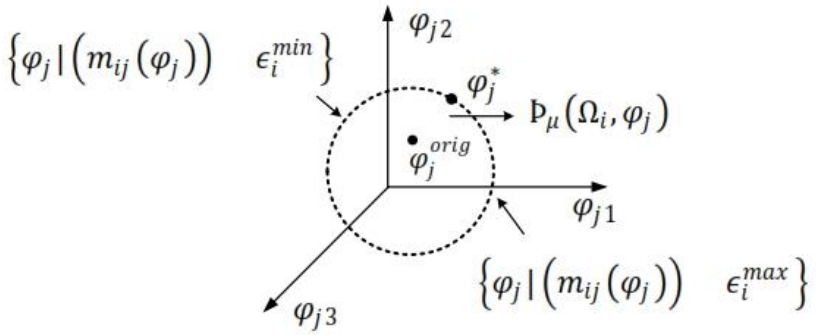


Fig. 3.8. Possible directions of variation in uncertainty parameter φ_j for performance feature

The region inside the sphere is characterized as robust region. All of the values of φ_i lying inside the sphere yield the system in acceptable mode of operation for a particular Ω_i and vice versa. The point on the boundary of the sphere marked as $\varphi_j^*(\Omega_i)$ is uniquely defined as the

smallest distance from φ_j^{orig} to any boundary point. A significant information is revealed by the L_p-norm value of the two variables $||\varphi_j^*(\Omega_i) - \varphi_j^{orig}||_p$, that is the maximum allowable variation in the space around φ_j^{orig} [3.18]. The L_p-norm keeps the φ_j in the acceptable performance range for Ω_i.

The above mentioned value also depicts the tolerable variation in φ_j. Let the distance function $||\varphi_j^*(\Omega_i) - \varphi_j^{orig}||_p$ be called as *robustness radius*, $p_u(\Omega_i, \varphi_j)$ of Ω_i in the presence of φ_j, given as:

$$\rho_\mu(\Omega_i, \varphi_j) = L_\rho(\varphi_j, \varphi_j^{orig}) =_{min}^{\rho} \sqrt{\sum_{i=1}^{n} |\varphi_j[i] - \varphi_j^{orig}[i]|^P}, \tag{3.16}$$

where ρ_μ is the robustness of the application under process and $1 \leq \rho \leq \infty$. Moreover when $\rho = \infty$, the L_∞-norm distance function $L_\infty(\varphi_j, \varphi_j^{orig})$ is given as:

$$L_\infty(\varphi_j, \varphi_j^{orig}) = max_{i=1}^{n} |\varphi_j[i] - \varphi_j^{orig}[i]|, \tag{3.17}$$

where φ_j and φ_j^{orig} are two *n*-dimensional data objects. The robustness of the system against uncertainties can be extended and generalized for all $\Omega_i \, \epsilon \, \Omega$. Without loss of generality, the robustness metric is defined as:

$$\alpha_\mu(\Omega, \varphi_j) =_{(\Omega_i \epsilon \Omega)}^{min} \left(\rho_\mu(\Omega_i, \varphi_j)\right), \tag{3.18}$$

where the $\alpha_\mu(\Omega, \varphi_j)$ is the robustness measure for the evaluation of a cloud in accordance with the performance feature set Ω against the uncertainty parameter φ_j. The choice of Norm can be altered depending upon the particular scenario of operation. The distance calculation can be modified so that the significance of an element can be varied. Thereby, by varying the probability of the weight of an element, the distance calculation changes indirectly.

3.5. Evaluation of Robustness with an Example

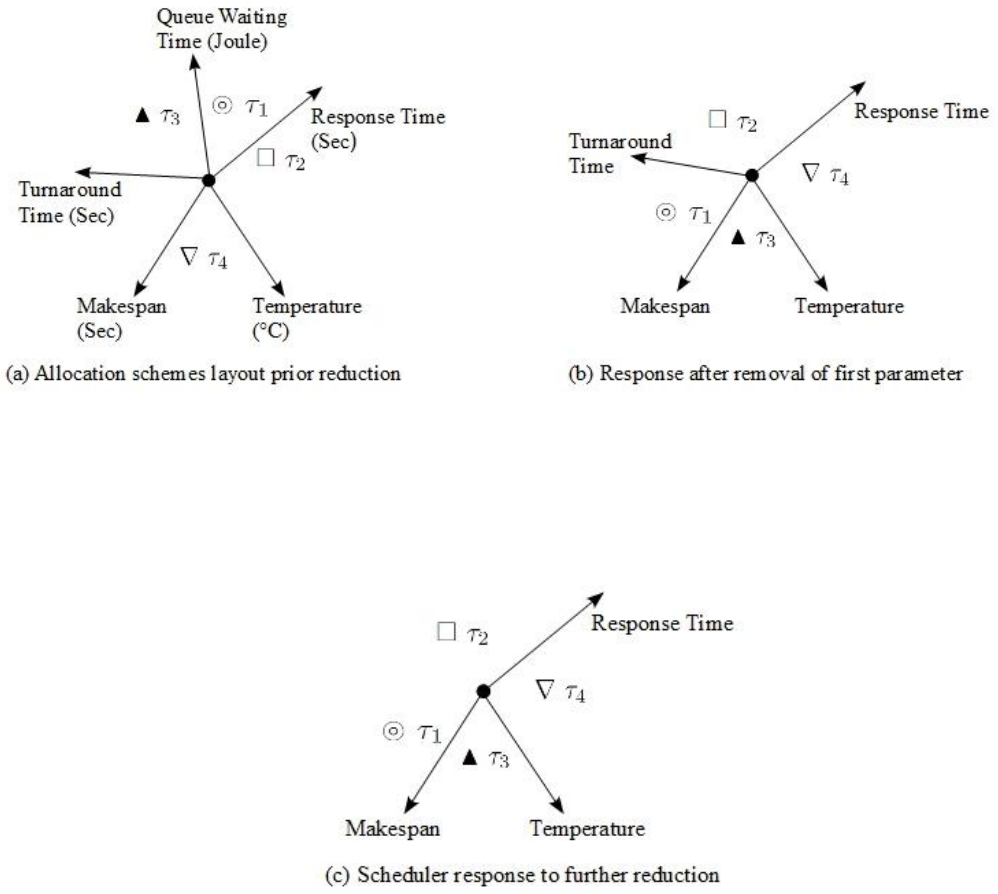
We consider a resource scheduling system for the derivation of a robustness measure using the dimension reduction methodology. The resource scheduling system allocates a set of independent applications *A* to set of machines, *M*. The mapping of applications to machines is characterized by the perseverance of the identified performance parameters impact. The perturbation parameters in the system may cause a variation in the expected values resulting in system operation beyond the estimated range. As discussed earlier, the resource allocation considers multiple parameters as performance attributes to evaluate the system's functionality. The vector Ω, consider the same set of parameters as considered in Section 3.4, that is,

Ω = {makespan, queue waiting time, turnaround time, response time, temperature}.

A useful representation of the Multi-dimensional Scaling (MDS) methodology can be realized with the help of resource scheduling example. The *n* performance features considered are dependent on *k* perturbations. The predominant perturbation parameters can be written in the form of a vector Φ,

φ = {VM failure, estimation error, network congestion}.

Figure 3.9 depicts the impact of the dimension reduction procedure on the layout of the allocation schemes. The elements of set Ω are represented as co-ordinate axis in Fig. 3.9(a). Each co-ordinate in-turn represents a dimension of the scheduling system. To explore the effect of the dimension reduction procedure four resource allocation schemes are considered for comparison purpose. The resource allocations labeled as τ_1, τ_2, τ_3 and τ_4 are represented by the symbols ⊙, □, ▲, and ∇, respectively. The convergence process employs the reduction of one dimension (performance feature) at a time.



(a) Allocation schemes layout prior reduction

(b) Response after removal of first parameter

(c) Scheduler response to further reduction

Fig. 3.9. Impact of dimension reduction on the orientation of allocation schemes in scheduling system of cloud

Each time when a parameter is reduced, the orientation of the allocation schemes is affected, accordingly. Intuitively, the reduced dimension co-ordinates have an impact on the distribution behavior in terms of robustness on the allocation schemes. The system layout after the reduction of the performance feature "queue waiting time" is shown in Fig. 3.9(b). Similarly, a further reduction of one more parameter, that is "turnaround time" further reduces the complexity. Figure 3.9(c) depicts the layout of response of the allocation schemes to the reduction of the turnaround time parameter.

A distinction (robustness based) among the allocation techniques can be made by calculating distance of the points from origin. The robustness estimate in this case prefers a wider radii for efficient performance. The resource allocation scheme that still lies inside the vicinity of the robustness sphere is regarded as robust and vice versa. The robustness boundary is represented by a dotted sphere in Fig. 3.10. Note that the behavior of a resource scheduling technique to reduction procedure is independent of the behavior of any other scheduling techniques evaluated by the scheduler for robustness.

The cloud paradigm may possibly work in uncertain environmental conditions, that may cause the system to violate the robustness boundary. If the scheduler performance lies within the bounds of the robustness sphere, the resource allocation response is in the acceptable region of operation, otherwise not. For distance calculation of the data point from the robustness boundary, $l_2 - norm$(Euclidean norm) can be used in the above discussed example, such that:

$$|r| = \sqrt{\sum_{k=1}^{n} |r_k|^2}, \tag{3.19}$$

where

$$r = [r_1, r_2, r_3 \dots \dots . r_n]^t. \tag{3.20}$$

Note that the Euclidean distance obtained is a measure of difference between the estimated value of operation and the actual value of operation. The $l_2 - norm$ gives the better approximation by rendering a unique and smooth representation of the approximated data. Moreover, influence of error is minimized using $l_2 - norm$ [3.18]. To ascertain robustness, the robustness radius must adhere to $|r| \leq \rho_\mu(\Omega_i \varphi_j)$ for the resource allocation schemes.

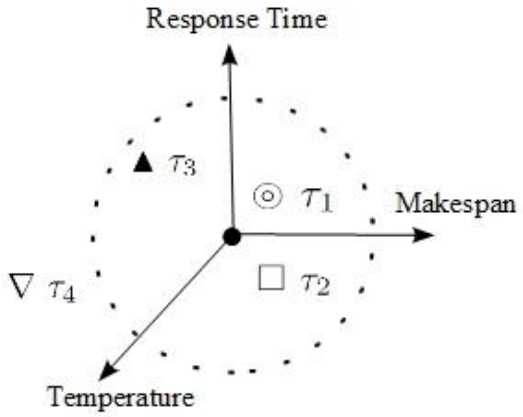


Fig. 3.10. Robustness boundary in the presence of perturbations

3.6. Honoring the SLA using User-Defined Priority Measures

Due to the promising performance that the cloud rendered over the last decade, vendors are facing urgent need to provide customized, reliable, and QoS rendering computing services [3.11]. A problem of substantial recent interest faced by the cloud service providers is the achievement of customers' satisfaction according to the specified SLA. The customers satisfaction is assured by the delivery of the required QoS as specified in the SLA. To meet the user defined level of performance the cloud service providers must deliver the agreed level of SLA.

The customer specifies the required features (benchmarks, targets, and metrics) along with the significance level of each of the aforementioned features. To implement a client specified QoS based cloud architecture and avoid the SLA breaches the service provider should consider following the five step procedure, given in Fig. 3.11. The features that are required should be identified along with the uncertainty parameters that can cause a negative effect on the system performance. Priority of the performance feature is defined by the user and the reduction procedure

takes into account user's performance. The problem complexity is reduced by employing dimension reduction methodology to attain the desired SLA level.

The SLA based dimension reduction follows the same pattern of reduction (geometric) as detailed earlier in the paper, except for the parameters priorities defined aswell. The client's desired set of performance aspects are recorded in the set Ω, similar to the one is given in Section 3.5. Let γ represents the set of values signifying the importance of each of the parameter enlisted in Ω. To illustrate the idea, consider that the most significant parameter takes the highest numeric value in ranking out of ten, such that:

$$\gamma = \{7, 4, 8, 6, 9\} \tag{3.21}$$

mapped to, Ω such that: Ω = {makespan, queue waiting time, turnaround time, response time, temperature}, in the following manner:

Makespan= 7, Queue Waiting Time= 4, Turnaround Time= 8, Response Time= 6 and Temperature=9.

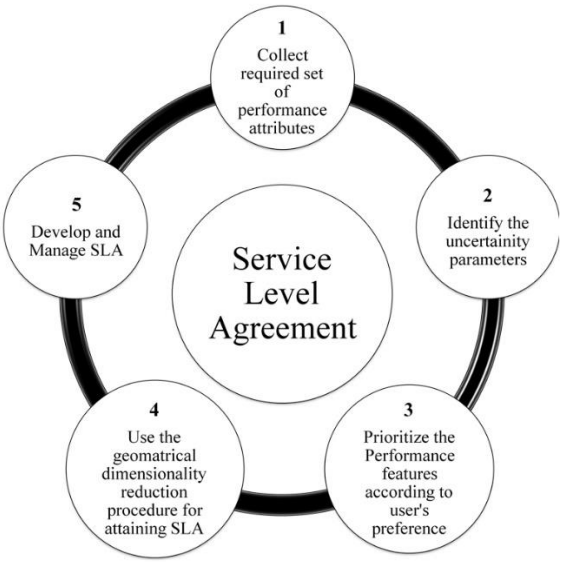


Fig. 3.11. SLA procedure

Let us consider two resource allocations, τ_1 and τ_2. For the purpose of demarcation between the more robust and less robust allocation schemes we employ the dimension reduction procedure. The reduction decision is made, based on the priorities defined by the client. In the quest to achieve the clients desired SLA the order of dimension reduction is carried in a manner that the most significant parameter is reduced first. For example, we reduce the energy parameter first as it attains the highest priority according to the client's requirements. Likewise, the next parameter to be reduced is the throughput. Figure 3.12 depicts the behavior of two scheduling schemes after the dimension reduction of the three most important performance features. The resource scheduling schemes, τ_1 and τ_2 are represented by • and ⋆ respectively in the Fig. 3.12.

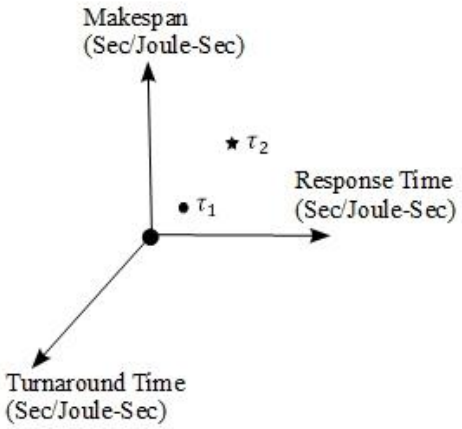


Fig. 3.12. Response of two scheduling schemes after SLA dependent dimension reduction

The robustness ranking quantification favors the allocation scheme that is farthest from the origin. Therefore, the resource scheduling scheme represented by an asterisk (⋆) has a clear distinction of providing a better SLA level than the one represented by a dot (•) in Fig. 3.12.

This follows that the resource allocation τ_2 is more promising and better than the resource allocation τ_2 according to the client defined requirement of QoS to ensure the SLA.

3.7. Performance Evaluation

In this section, we perform a simulation study of the dimension reduction procedure defined in Section 3.5. Using the dimension reduction procedure, we will evaluate the performance of a collection of resource allocation schemes. Each of the resource allocation methodologies is evaluated based on various performance parameters. The set of performance parameters take into account the real-world performance evaluation features, such as the makespan, throughput, turnaround time, and response time. The reduction procedure is verified for the effect of sequence of the parameters selected for reduction in the decision of most robust allocation scheme. The order of selection of parameters in the dimension reduction process has an impact on the robustness measure that each of the allocation scheme exhibits. The robustness measure is subsequently used to determine the most appropriate resource allocation scheme depending on the order of the parameters chosen in the reduction procedure. Therefore, the most suitable resource allocation scheme, in response to one particular performance parameter may be altogether different when another performance parameter is selected for reduction.

As explained earlier in the paper, the dimension reduction procedure reduces the dimension by one level, each time the procedure is applied. The impact on the allocation schemes due to the dimension chosen for reduction is recorded and the dimension selected is then removed from the set of parameters/dimensions to be reduced for convergence purpose. In this manner, the dimension reduction process is carried until the required level of dimensions (coordinates) is achieved. Once the reduction process is complete, the robustness radius is calculated. A comparison of the robustness radii is performed among the allocation schemes in

the evaluation process to find out the most suitable scheme. The order of the performance parameters selected for reduction in the reduction procedure is customized when a certain level of SLA is to be fulfilled. The purpose of using a customized parameter reduction is to achieve the user specified QoS level to guarantee the SLA and procure users' satisfaction.

To perform the evaluation of dimension reduction method on a diverse environment, the procedure is evaluated on three different sets of workload: **(a)** randomly generated workload, **(b)** real-time workload, and **(c)** the SLA based reduction. Each of the plots depicts the results of an average of 1000 iterations per simulation. In the following subsections, we discuss the evaluation results for the above mentioned scenarios.

3.7.1. Random Workload Response to Dimension Reduction

This subsection presents a detailed analysis and experimental evaluation of the dimension reduction procedure for the randomly generated dataset. Initially, a set of six resource allocations along with a data set of performance parameters are generated randomly. The resource allocations are then compared and evaluated on the basis of response to nine performance parameters selected for the reduction process. The analysis of dimension reduction procedure based on the random data starts with the random generation of the performance parameters taken in consideration. Each time the dimension reduction procedure is simulated, a parameter is chosen for reduction and is removed after recording its impact on the allocation schemes. This recorded response is then used to segregate the results into most appropriate allocations depending on the robustness radius provided by each allocation scheme.

Once the anticipated level of dimensionality is attained, a mathematical comparison of the robustness radius that each of the allocation schemes depicts is performed, as presented in Section 3.4 of the paper. Figure 3.13 presents the distribution pattern of the number of times a resource

allocation is selected as the best, based on the robustness radius exhibited. It can be seen in Fig. 3.13 that Alloc. 3 outperformed all other allocation schemes in producing better results in adherence to robustness.

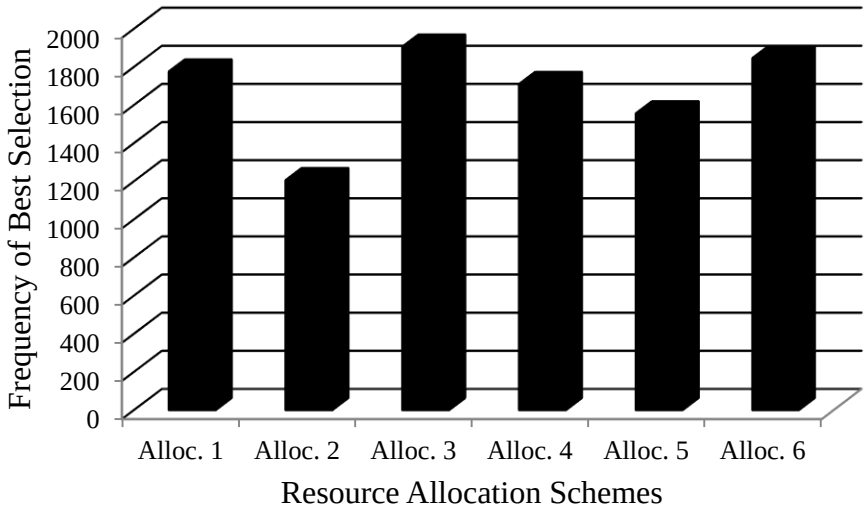


Fig. 3.13. Best allocation scheme distribution pattern for a randomly generated workload

Table 3.2 illustrates a comparison of the selection of the most suitable allocation scheme based on the variation in the reduction order of performance parameters. In Table 3.2, *P* represents the performance parameter and *RA* depicts the resource allocation scheme. The 1st, 2nd, and 3rd columns of the table present the performance parameters that are kept invariant in successive rows. However, the right part of the Table 3.2 describes the performance parameters that are varied during the reduction process. The results illustrate that the order in which a parameter is chosen in the reduction process has a significant impact on the robustness measure of resource allocation schemes. Subsection (A) of the Table 3.2, depicts the results for the case when the first three performance parameters are kept identical and variation is incorporated in the last

three parameters only. At each level, a different sequence of the order of performance features results in a different best allocation strategy. Whereas, the subsection (B) of the Table 3.2 depicts results for the case when the similarity extent is increased to a level four. Despite, an increase in the extent of the level of similarity, for every new combination of the last two parameters in the reduction order, a different allocation scheme is selected as most promising. Therefore, from the above discussion we deduce that the reduction order has an impact on the results in terms of selection of the best allocation scheme.

Table 3.2. Response of performance parameters to variation in last two levels in the reduction

Order of Reduction Comparison in Random Data Distribution						
(A)						
Parameter Selection Order						**Best Allocation Scheme**
1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	
P_2	P_7	P_4	P_5	P_3	P_1	RA_5
P_2	P_7	P_4	P_1	P_6	P_9	RA_3
P_2	P_7	P_4	P_8	P_5	P_3	RA_4
P_2	P_7	P_4	P_5	P_1	P_6	RA_6
(B)						
Parameter Selection Order						**Best Allocation Scheme**
1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	
P_5	P_8	P_1	P_6	P_3	P_4	RA_6
P_5	P_8	P_1	P_6	P_9	P_2	RA_1
P_5	P_8	P_1	P_6	P_4	P_3	RA_5
P_5	P_8	P_1	P_6	P_7	P_9	RA_2

To explore the impact of a parameter's order placement in the dimension reduction process, consider Fig. 3.14. In Fig. 3.14 *RA* depicts the resource allocation scheme and *P* represents the performance parameter. It can be observed that the RA_1 exhibits highest frequency of yielding best results when P_1 is selected in the reduction process. Likewise, when P_9 is chosen first in the reduction process, the RA_3 outperforms rest in compliance to best results. Similarly, RA_6 depicts the highest robustness when the P_7 is selected in dimension reduction procedure. Nonetheless, the same allocation scheme produces a nominal ratio of best results when any other performance parameter is chosen at the first place in the dimension reduction process. Therefore, from the above observations, we conclude that the frequency of times a resource allocation is selected as the most appropriate varies with a change in the parameter reduction order. The above attained outcomes can help the cloud in finding the most appropriate allocation scheme based on the most prioritized performance parameters.

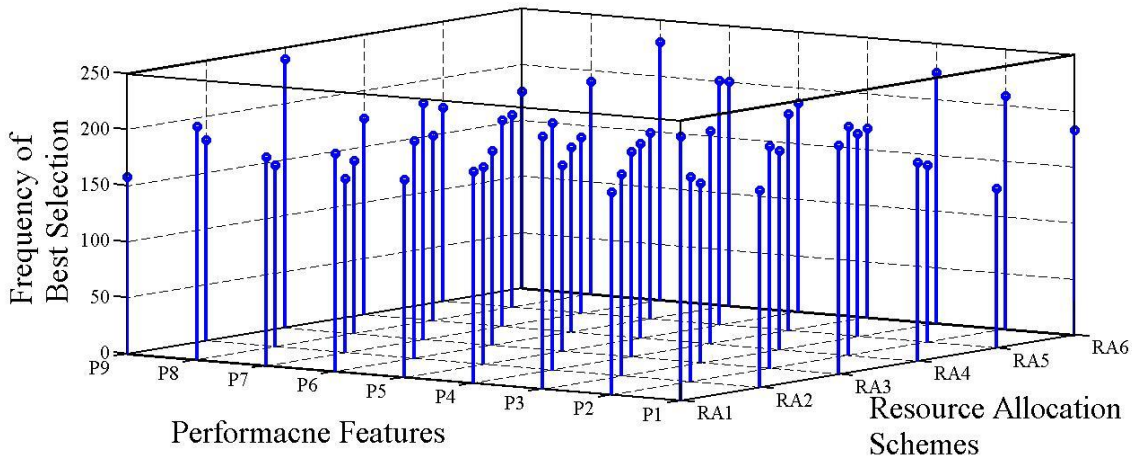


Fig. 3.14. Comparison of best allocation scheme under various performance parameters for random workload

3.7.2. Results and Discussion based on Real-time Workload

In this section, we evaluate the dimension reduction procedure on a real-time workload. To test the impact of the variation in reduction order of parameters, we employed five resource allocation schemes that are evaluated on the basis of nine performance parameters. The simulations devised for the reduction methodology are iterated on the real-time data-set to evaluate the scheduling schemes for robustness observance. The resource allocations considered for the robustness adherence comparison are: **(a)** Longest Job First (LJF), **(b)** Shortest Job First (SJF), **(c)** Shortest Remaining Time First (SRTF), **(d)** First Come First Serve (FCFS), and **(e)** Greedy Thermal Aware (GTA) [3.19]. The performance parameters considered for the evaluation and comparison are: **(a)** makespan, **(b)** average queue waiting, **(c)** total queue wait, **(d)** average turnaround, **(e)** total turnaround, **(f)** average response time, **(g)** total response time, **(h)** average temperature, and **(i)** average temperature difference among pods.

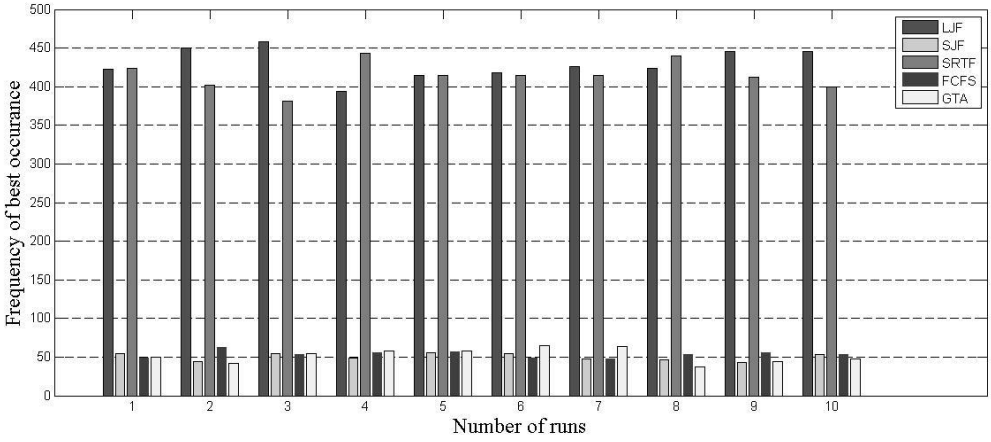


Fig. 3.15. Most robust allocation selection based on ten iterations

To study the selection behavior of best allocation scheme under the dimension reduction methodology, we simulate the procedure one thousand times. The procedure is run on the above

mentioned five scheduling techniques under the same configuration of parameters. To grasp the impact of the dimension reduction methodology the simulations are performed for a total of one thousand times during each run. Figure 3.15 presents the results in terms of most robust allocation strategy achieved in each run for all of the nine performance parameters. It can be observed that in almost all of the iterations, LJF and SRTF outperform the other scheduling approaches considered for the evaluation.

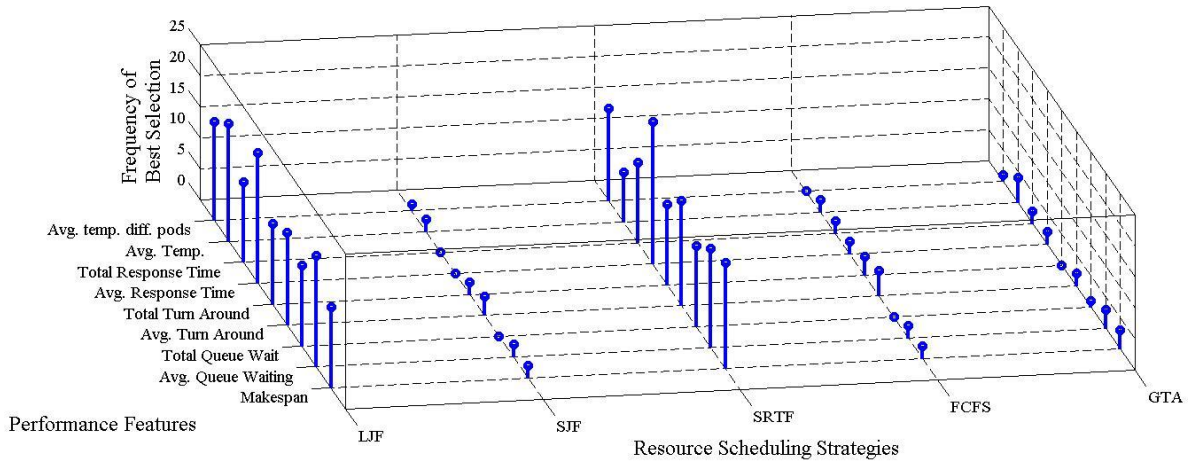


Fig. 3.16. Comparison of allocation scheme response to the performance parameters for real-time workload

The impact of a parameter's selection order in the dimension reduction process on the five scheduling techniques enlisted above is presented in Fig. 3.16. The graph signifies a 3-dimensional behavior of the response of the allocation techniques to the performance feature selected at the first place in the dimension reduction procedure. For each iteration, a performance feature is randomly chosen to begin the reduction process and the response of the allocation schemes is recorded. The frequency of a particular resource allocation scheme selected as the most suitable and robust scheme with the reduction of certain performance feature is depicted in Fig. 3.16.

Although among the available resource scheduling strategies, the LJF and SRTF are the two most frequently selected best resource scheduling schemes. Moreover, Fig. 3.16 reflects a picture of the overall selection criteria based on the individual performance feature. As the parameter reduction order is changed, the resulting best resource allocation distribution changes accordingly. Therefore, we conclude that the order of reduction gears the selection of the most promising allocation schemes.

For a clearer understanding of the effect of order in the dimension reduction technique consider Table 3.3. The table clearly indicates that as a particular feature is altered in the reduction process, the resulting best scheduling changes accordingly. In upper half of Table 3.3, first four performance features in the reduction order are identical in all of the three successive rows; the average turnaround is followed by average temperature parameter, then total turnaround, and finally average response time in Table 3.3. However, a change in the selection of last two parameters alters the decision of the scheduler. At each level, a different sequence of the order of performance feature results in a different best scheduling strategy. Similarly, the second part of Table 3.3 portrays an identical behavior of the response of the system to a variation in the order of reduction. Therefore, the overall distribution of the results is dependent on the order in which a performance feature is selected and has a significant influence on the final results attained.

3.7.3. SLA based Dimension Reduction

To deliver the desired level of QoS, the cloud service providers must satisfy the required SLA level. Moreover, to avoid uncertainties that may cause an abnormal system response, the QoS needs to be observed to guarantee the SLA under all circumstances. For the system model given in Section 3.5, the experiments were performed with the same parameters as for Section

7.2 simulation scenario (real-time workload). To ensure that the SLA is met, we customized the scheduler performance in the cloud by altering the priorities of the performance parameters. Therefore, weights based on precedence of the performance metrics have been assigned. The performance parameter with the highest priority is reduced first, so as to put a significant impact on the values of the remaining dimensions reduced. Consider the case when the reduction order is predefined by the client to steer the reduction procedure according to the priorities. For the set of performance parameters presented in sub-section 3.7.2, we consider the ranking order, γ_1, such that:

$$\gamma_1 = \{9, 1, 4, 8, 5, 7, 2, 6, 3\} \tag{3.22}$$

Table 3.3. Response of performance parameters to variation in last two levels

Order of Reduction Comparison							
(A)							
	First parameter	**Second parameter**	**Third parameter**	**Fourth parameter**	**Fifth parameter**	**Sixth parameter**	**Best Allocation Scheme**
(i)	Average Turnaround	Average Temperature	Total Turnaround	Average Response Time	Total Queue Waiting	Makespan	LJF
(ii)	Average Turnaround	Average Temperature	Total Turnaround	Average Response Time	Makespan	Average Queue Waiting	FCFS
(iii)	Average Turnaround	Average Temperature	Total Turnaround	Average Response Time	Average Queue Waiting	Total Response Time	SRTF
(B)							
	First parameter	**Second parameter**	**Third parameter**	**Fourth parameter**	**Fifth parameter**	**Sixth parameter**	**Best Allocation Scheme**
(i)	Average Temperature	Total Response Time	Average Response Time	Total Queue Waiting	Average Turnaround	Average Temperature Difference among pods	GTA
(ii)	Average Temperature	Total Response Time	Average Response Time	Total Queue Waiting	Average Temperature Difference among pods	Average Turnaround	SRTF

The value of γ_1 depicts that the performance parameter makespan is given the highest priority, followed by the average turnaround, average response time, average temperature, total turnaround, and total queue waiting. The above defined precedence orders marks LJF as the best allocation scheme as shown in Table 3.4. On the contrary, when the user specification changes the precedence order to, γ_2, such that:

$$\gamma_2 = \{7, 2, 3, 10, 8, 6, 1, 5, 4, \} \tag{3.23}$$

The reduction order is preceded by the average turn around, followed by total turnaround makespan, average response time, average temperature, and average temperature difference among pods. The results obtained depicts that the GTA schemes received the maximum number of hits as the best allocation scheme. Table 3.4 presents results for the SLA based dimension reduction.

Table 3.4. SLA based dimension reduction

Ranking Order	**Order of Reduction Comparison**						
	First parameter	**Second parameter**	**Third parameter**	**Fourth parameter**	**Fifth parameter**	**Sixth parameter**	**Best Allocation Scheme**
γ_1	Makespan	Average Turnaround	Average Response Time	Average Temperature	Total Turnaround	Total Queue Waiting	LJF
γ_2	Average Turnaround	Total Turnaround	Makespan	Average Response Time	Average Temperature	Average Temperature Difference among pods	GTA

In summary, for every user, the desired SLA is different. Based on the user priority of a particular performance parameter, the best resource allocation is determined. For instance, if the user prefers the response time parameter the reduction process yields the SRTF as the more robust allocation scheme. Nonetheless, when the user preference changes to average temperature, the SJF turns out to be the best allocation scheme. Therefore, we conclude that for a cloud to satisfy

the SLA, the user preference has a significant impact on the selection of the most robust allocation strategy.

3.8. Related Work

Robustness measurement has been widely studied and addressed in literature, for example [3.3] for the benefit of a particular framework. The framework proposed in the paper is generic and focuses on handling data lying in higher manifolds. By using dimension reduction, we perform data convergence in a stage wise manner and then test the acquired result for robustness to attain high end effective performance. Maxwell *et al.* [3.20] and Ali *et al.* [3.6] proposed robustness metrics for the quantification of robustness for a given resource allocation environment. Ghoshal *et al.* [3.21] applied a data management strategy in distributed transient environments like cloud for handling both virtual machine failure and variations in network performance. The aforementioned technique is unable to handle and overcome failures that occur during run-time. Nevertheless, our methodology is more focused on achieving high level performance in the cloud environment to overcome the threats and challenges in an effective manner. Guaranteeing performance is of utmost importance in the implementation of a cloud paradigm.

Larsen *et al.* [3.22] used a syntactic transformation approach that employs classical analysis techniques and tools to achieve robustness. Moreover, to achieve the required QoS level authors in [3.23] applied a fuzzy control logic for the resource management. Nevertheless, the application of fuzzy control is effective only in systems with a simple architecture. To handle the resource allocation problem effectively for a variety of scenarios a great deal of knowledge about the rules and parameters involved is required and extensive simulation needs to be carried out before designing fuzzy system. Macias *et al.* [3.24] proposed an SLA improvement strategy by utilizing a two way communication path between the market brokers and resource managers.

The aforementioned technique improves the SLA violation only when prior knowledge about the reputation of the system is available. Otherwise no significant difference can be brought using this methodology. On the other hand, our methodology can be adapted for any possible scenario. Scheduling and managing of resource with QoS maintained according to the SLA specifications is a major challenge in a cloud computing environment. The perturbations present in the system environment make the aforementioned tasks even more challenging and opens new paradigms in resource scheduling of cloud. To accomplish the above mentioned goal, the researchers in [3.25] presented a scheduling heuristic that caters multiple SLA parameters. The parameters considered are limited to CPU time, network bandwidth, and storage capacity. However, performance parameters such as response time, temperature, and processing time are not considered in improving the system's performance.

Li *et al.* [3.26] proposed a customizable cloud model for resource scheduling. An additional aspect of trust is incorporated in the system architecture along with the QoS targets for performance up-gradation. Although the QoS parameters considered in this approach includes response time, bandwidth, storage, reliability, and cost. However, the QoS delivery is restricted to the average values of the above mentioned performance aspects. Moreover, guarantee of the service delivery is not provided despite of the predetermined confidence level. The aforementioned approaches may cater users' preferences, but are unable to guarantee a QoS satisfaction level according to the SLA requirements. The problem we deal here is different from the existing work since it takes into account multiple parameters to optimize the system's performance, despite of the uncertainty present in the system environment.

We employ the dimensionality reduction technique as our solution to handle the impact of a parameter number as high as n, $\text{where } n \gg 0$, while meeting the QoS requirements. Relative

to the ambient dimension n, the dimensionality reduction techniques aim to extract concise information about the data lying in a high-dimensional space [3.12]. The data lying in higher manifolds can be converged using manifold-reduction techniques. Current state- of-the-art techniques for dimensionality reduction can be broadly bifurcated into linear and non-linear dimensionality reduction. The prior includes classical methods like Principal component Analysis (PCA) and Multi-Dimensional Scaling (MDS). Nevertheless, the linear techniques outperformed the non-linear techniques due to the incapability to handle non-linear data structures [3.27]. On contrary non-linear manifold learning techniques, such as the Locally Linear Embedding (LLE), Laplacian Eigenmaps, and Isomap are efficient in handling non-linear data structures. However, the aforementioned methodologies are computationally expensive to handle and are not scalable due to their time and memory complexity [3.28]. Nevertheless, we emphasize on preserving the critical relationship among the data-set elements and to discover critical information about the data preserved, under the mapping Φ keeping the computational cost minimum.

3.9. Conclusion and Future Work

In this paper, we analyzed and implemented a geometrical dimension reduction mathematical model for the evaluation of robustness of resource allocation schemes in the cloud. The presence of uncertainty in the system parameters is considered and *n*-number of performance parameters are entertained that depicts the wideness of the approach. Our results reveal that the process of dimension reduction is dependent on the order of the parameter selected during the convergence procedure. The novelty of this work is the freedom of incorporation of the performance parameters required for robustness evaluation. The results achieved after reduction retain a reflection of all of the parameters utilized in the convergence process. The proposed method can be used to gauge robustness and observe the most effective allocation scheme among

a group of allocation schemes that are apparently hard to distinguish. Moreover, the proposed methodology can be extended to a customized scenario to meet the QoS according to the required SLA, in a cloud environment. We have presented two theorems that strengthen our reduction approach linked to the robustness measurement procedure.

As our future work, we will focus on the optimization of the results by generating a pareto optimal set of results. The pareto optimal set will be realized into a pareto front that would enable the cloud to generate the most optimized findings for a multi-parameter system environment.

3.10. References

[3.1] F. Gu, K. Shaban, N. Ghani, S. Khan, M. Rahnamay-Naeini, M. Hayat, and C. Assi, "Survivable cloud network mapping for disaster recovery support," *IEEE Transactions on Computers*, vol. PP, no. 99, pp. 1–1, 2014.

[3.2] B. Guan, J. Wu, Y. Wang, and S. Khan, "Civsched: A communication-aware inter-vm scheduling technique for decreased network latency between co-located vms," *IEEE Transactions on Cloud Computing*, vol. 2, no. 3, pp. 320–332, July 2014.

[3.3] K. Bilal, M. Manzano, S. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 1–1, Jan 2013.

[3.4] G. Group, *Worldwide IT Spending Forecast*, Jan 2015. [Online]. Available: http://www.gartner.com/newsroom/id/2959717

[3.5] C. Hewwit, "Orgs for scalable, robust, privacy-friendly client cloud computing," *IEEE Internet Computing*, vol. 12, no. 5, pp. 96–99, 2008.

[3.6] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim, "Measuring the robustness of a resource allocation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 7, pp. 630–641, 2004.

[3.7] *IEEE Std 610.12-1990-Glossory of Software Engineering Technologies*.

[3.8] F. Zhang, J. Cao, W. Tan, S. Khan, K. Li, and A. Zomaya, "Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 338–351, Sept 2014.

[3.9] J. Apodaca, D. Young, L. Briceo, J. Smith, S. Pasricha, A. A. Maciejewski, H. J. Siegel, and S. B. B. Khemka, "Stochastically robust static resource allocation for energy minimization with a makespan constraint in a heterogeneous computing environment," in *9th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 11*, 2011, pp. 22–31.

[3.10] B. Dorronsoro, P. Bouvry, J. Caero, A. Maciejewski, and H. Siegel, "Multi-objective robust static mapping of independent tasks on grids," in *2010 IEEE Congress on Evolutionary Computation (CEC)*, July 2010, pp. 1–8.

[3.11] F. Zhang, J. Cao, K. Li, S. U. Khan, and K. Hwang, "Multi-objective scheduling of many tasks in cloud platforms," *Future Generation Computer Systems*, vol. 37, pp. 309 – 320, 2014.

[3.12] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," in *First IEEE Conference on Visualization*, Oct 1990, pp. 361–378.

[3.13] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal of Scientific Computing*, vol. 26, no. 1, pp. 313–338, Jan. 2005.

[3.14] H. L. Yap, M. Wakin, and C. Rozell, “Stable manifold embeddings with structured random matrices,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 4, pp. 720–730, Aug 2013.

[3.15] I. Fodor, “A survey of dimension reduction techniques,” Tech. Rep., 2002.

[3.16] [Online]. Available: www.math.toronto.edu/mgualt/MAT1300/

[3.17] A. Vosoughi, K. Bilal, S. U. Khan, N

+. Min-Allah, J. Li, N. Ghani, P. Bouvry, and S. Madani, “A multidimensional robust greedy algorithm for resource path finding in large-scale distributed networks,” in *8th International Conference on Frontiers of Information Technology*, 2010, pp. 16:1–16:6.

[3.18] T. Marosevic, “A choice of norm in discrete approximation,” *Mathematical Communications*, vol. 1, no. 2, pp. 147–152, 1996.

[3.19] J. Li, M. Qiu, J.-W. Niu, L. T. Yang, Y. Zhu, and Z. Ming, “Thermal-aware task scheduling in 3d chip multiprocessor with real-time constrained workloads,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 2, pp. 24:1–24:22, Feb. 2013.

[3.20] P. Maxwell, A. A. Maciejewski, H. J. Siegel, J. Potter, G. Pfister, J. Smith, and R. Friese, “Robust static planning tool for military village search missions: model and heuristics,” *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 10, no. 1, pp. 31–47, 2013.

[3.21] D. Ghoshal and L. Ramakrishnan, “Frieda: Flexible robust intelligent elastic data management in cloud environments,” in *2012 SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC)*, Nov 2012, pp. 1096–1105.

[3.22] K. G. Larsen, A. Legay, L.-M. Traonouez, and A. Wsowski, “Robust synthesis for real-time systems,” *Theoretical Computer Science*, vol. 515, pp. 96–122, Jan. 2014.

[3.23] J. Rao, Y. Wei, J. Gong, and C.-Z. Xu, “Qos guarantees and service differentiation for dynamic cloud applications,” *IEEE Transactions on Network and Service Management*, vol. 10, no. 1, pp. 43–55, March 2013.

[3.24] M. Macias, J. Fito, and J. Guitart, “Rule-based sla management for revenue maximization in cloud computing markets,” in *International Conference on Network and Service Management (CNSM)*, Oct 2010, pp. 354–357.

[3.25] V. Emeakaroha, I. Brandic, M. Maurer, and I. Breskovic, “Sla-aware application deployment and resource allocation in clouds,” in *2011 IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW)*, July 2011, pp. 298–303.

[3.26] W. Li, Q. Zhang, J. Wu, J. Li, and H. Zhao, “Trust-based and qos demand clustering analysis customizable cloud workflow scheduling strategies,” in *IEEE International Conference on Cluster Computing Workshops (CLUSTER WORKSHOPS)*, Sept 2012, pp. 111–119.

[3.27] L. K. Saul and S. T. Roweis, “Think globally, fit locally: Unsupervised learning of low dimensional manifolds,” *Journal of Machine Learning Research*, vol. 4, pp. 119–155, Dec. 2003.

[3.28] A. Najafi, A. Joudaki, and E. Fatemizadeh, “Nonlinear dimensionality reduction via path-based isometric mapping,” *Machine Learning*, vol. 4, pp. 119–155, 2013.

4. THERMAL-AWARE, POWER EFFICIENT, AND MAKESPAN REALIZED PARETO FRONT FOR CLOUD SCHEDULER[2]

This paper is submitted to *IEEE CloudNA 2015*. The authors of the paper are Saeeda Usman, Kashif Bilal, Nasir Ghani, Samee U. Khan, and Laurence T. Yang.

4.1. Introduction

The dynamic and promising services delivered by Cloud computing paradigm have strikingly elevated the demand of Cloud deployment (models). The paradigm orchestrates the computing resources, such as the processing cores, I/O resource, and storage to meet "on demand" client requirements. The aforementioned characteristic of Cloud has extensively scaled the service offering to leverage and productize functionality. However, to ensure that the agreed Service Level Agreement (SLA) is met, the Clouds needs to offer metering services to avoid resource exploitation.

To provide a single pane view of the resources status and achieve high levels of granular visibility, intelligent monitoring should by realized to track resource utilization. Due to the increase in chip power density, the offered computing resources are prone to predicaments, such as hardware failure, low reliability, and insecure multi-tendency. Indeed, task completion is the foremost priority of schedulers in Cloud. Nevertheless, thermal management and power consumption hold pivotal importance in achieving high-end functionality. Moreover, cost minimization can be accelerated by avoiding over-provisioning of the aforementioned resources.

[2] The material in this chapter was co-authored by Saeeda Usman, Kashif Bilal, Nasir Ghani, Samee U. Khan, and Laurence T. Yang. Saeeda Usman had primary responsibility for conducting experiments and collecting results. Saeeda Usman was the primary developer of the conclusions that are advanced here. Saeeda Usman also drafted and revised all versions of this chapter.

Recently, a wide range of hardware and software based technique [4.1]-[4.3] have been proposed to control the power consumption of Chip Multi-Processors (CMPs). Although the management schemes could effectively reduce power depletion, they incur performance overhead in the form of thermal runaway. Motivated by this fact, the work presented in this paper address the abovementioned issue by considering the run-time information. Therefore, frequent monitoring of core temperature and operating frequency is required to lower the risk of chip overheating. We provide a methodology to mitigate the violation of peak power and temperature constraints, respectively.

To improve the performance of a scheduler in Cloud, we propose a temperature-aware power efficient methodology that judicially maximize performance and system reliability. The objective of this work is to optimize the cumulative performance of the resource allocation system. Intuitively, a convex optimization approach is devised to minimize the makespan, temperature, and power utilization of the scheduler. Our contribution circumvent the efficient management of power/temperature exploitation without comprising the task completion deadline.

Our major contributions are listed as follows:

- We develop a resource mapping heuristic that optimizes the performance using rigorous mathematical modeling. The scheduling decision space is constrained with a set of system specifications to attain the desired results.
- We model the problem to demonstrate the relationship between the frequency and the power consumption of the scheduling system in Cloud. The formulation unveils bounds on power and temperature utilization to dynamically adjust the resource utilization.
- The solutions that adheres to all of the constraints of power, makespan, and temperature constitute to the set of efficient or Pareto optimized solutions. Despite of the contradicting

nature of the objectives, we perform efficient mapping of resources to fulfill the end user demands without the violation of any timing constraints.

- The chip temperature is kept into consideration while scaling the operating frequency to avoid cooling challenges and ensure safe operating temperature.
- The relevant Pareto front of high quality is obtained for the optimization of the three objectives. The power and temperature management is judicially performed using the proposed heuristic. Moreover, the deadlines of the tasks are preserved to ensure efficient performance.

The paper is organized as follows. Section 4.2 presents the system architecture. Section 4.3 provides the details of the system model followed by the preliminaries of the proposed model in Section 4.4. The problem formulation is presented in Section 4.5. Section 4.6 presents a discussion on the methodology adopted for the Pareto front approximation. Performance evaluation and simulation results are presented in Section 4.7. Section 4.8 discusses the related work, and Section 4.9 concludes the paper.

4.2. Service Architecture

The Cloud service provider must ensure that the clients perpetually receive the Cloud services (resources) according to the agreed Quality of Services (QoS) level. Concurrently, the resources must be distributed efficiently and intelligently minimizing the resource wastage. Our goal is to optimally schedule the Cloud resources to fulfill the requirements. Allocation is mapped in a manner that the over provisioning of resources is prohibited. The scheduling mechanism detailed here promises to minimize the power consumption, temperature, and makespan of a scheduling within a Cloud environment. The objective is to optimize the task completion time such that the power conservation is not sacrificed while adhering to the temperature bound to overcome

hot-spots. The scheduling mechanism is routed to generate a set of Pareto solutions, the *Pareto front*, to achieve the most efficient optimization for the Multi-Objective Problem (MOP) under consideration.

Definition (Pareto front). *A point* $x^* \in X$ *is called Pareto optimal if there is no* $x \in X$ *such that* $F(x) < F(x^*)$*. Then,* $F(x^*)$ *is said to be globally efficient. The set of all such optimal points contours the image (curve) called as the Pareto front.*

The goal of MOP performed in the work presented here is to identify the set of efficient points, $F(x_i)$ for $\forall x_i \in x^*$, that is able to represent the Pareto front, as shown in Fig. 4.1. The key concept is to find the desired optimal operating point that guarantees all the objectives without violating the set of constraints. Consequently, a set of Pareto efficient solutions, are acquired that characterize the improvement all of the objectives without worsening any.

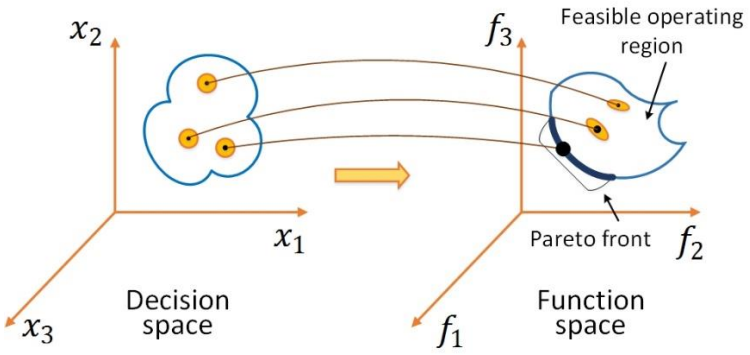


Fig. 4.1. Desired Pareto front for the set of efficient solutions

4.3. System Model

Consider a scheduling system in Cloud. The scheduler performs the task allocation. The tasks are mapped on the set of machines that satisfy the task requirements.

4.3.1. Machines

The resource scheduler allocates the incoming tasks to a set of machines, $M = \{M_1, M_2, \ldots\ldots, M_k\}$. The machines are assumed to be equipped with a Dynamic Voltage/ Frequency Scaling (DVFS) module. A constant and negligible transition time between successive levels of the DVFS is assumed for the problem considered in this paper. Each machine of the set $M(M_j \epsilon M)$ is characterized by the following attributes:

- The operations frequency of the machine, f_j, measured in hertz or cycles per unit time. By employing the DVFS, the frequencyf_j can varied from f_j^{min} to f_j^{max}. The hierarchy of the frequency bounds is defined by the relation, $0 < f_j^{min} < f_j^{max}$. The frequency holds a linear relationship with the speed of the machine [4.4].
- The machine architecture, $A(M_j)$, that comprises of the storage specifications, speed rendered, and the kind of CPU utilized.

4.3.2. Tasks

Consider a metaset of tasks, $T = \{\tau_1, \tau_2, \ldots\ldots, \tau_n\}$. Each task, $\tau_i \epsilon T$, is characterized by the following requirements:

- The time, t_i, required to complete the execution of the task. The Expected Time to Complete (ETC) is presumed to be known a priori.
- The machine architectural requirements, $A(\tau_i)$, that entails the task execution.
- The deadline, d_i, specifies the time at which the task execution must be performed. A successful mapping of tasks happen when all the constituting tasks of the set T are executed before the assigned deadlines.

4.4. Preliminaries

In this section, we present the modeling basics of power and temperature management. As explained in the previous section, real-time tasks in a task set are supposed to execute on a set of machines. The machines used in the scheduling system are assumed to be equipped with the DVFS methodology. Therefore, each machine is enabled to switch between discrete levels of normalized frequencies, that is, $\{f_1, f_2, \dots f_L\}$. Where $0 < f_j^{min} = f_1 < f_2 < \cdots < f_L = f_j^{max}$.

4.4.1. Power Model

The power requirement is a cumulative sum of the idle and active mode expected power consumption. Such that:

$$P_{Total} = P_{Dynamic} + P_{Static} + P_{Constant} = (\alpha . C_{EFF} . f . V^2) + (I_0 . V) + P_c, \qquad (4.1)$$

where $V, freq, C_{EFF}, I_{0,}$ and α are the supply voltage, clock frequency, effective switch capacitance, leakage current and activity rate of the computing device, respectively. The dynamic power consumption refers to the power consumed in the active/dynamic mode. The static power consumption corresponds to the power dissipated regardless of switching activity. The term P_c relates to the power expended by various system component activities, such as memory/disk accesses.

The dynamic mode power reflects a quadratic relationship between the supply voltage and the power consumption of the system. Assuming a linear relationship between voltage and frequency, given as:

$$vlotage \propto f, \qquad (4.2)$$

The dynamic power dissipation becomes,

$$Power \propto Voltage^3, \qquad (4.3)$$

or

$$Power \propto f^3. \tag{4.4}$$

The key design idea of DVFS is governed by the power-frequency proportionality relationship, such that a reduction in the clock frequency or supplied voltage, results in a cubic decrease in the power consumed. It is to be understood that the time to finish an operation is inversely proportional to the clock frequency. Such that:

$$time = {}^{1}/_{f}. \tag{4.5}$$

Therefore, lowering the supply voltage also decreases the maximum achievable clock speed. Running the machine at a slower frequency can significantly reduce a computing devices' dynamic power consumption. On the contrary, reducing the frequency/voltage would substantially slow down the time to complete an operation. It is apparent from the equations listed above that one can reduce cubically the instantaneous power consumption, at the expense of linearly increased delay (reduced speed). Owing to this analysis, we adopt the DVFS-based frequency selection scheme to maximize the processor power savings.

4.4.2. Temperature Model

To model the temperature realization of a machine in a scheduling system, we follow the dynamic thermal model proposed by Skadron *et al.* [4.5] to characterize the thermal behavior of the processor. The model unifies the Resistance-Capacitance (RC) model and the temperature $Temp$ at a time instance t, such that:

$$Temp(t) = Temp_{stsd} \times (Temp_{stsd} - Temp_{start}) \times e^{-t/RC}, \tag{4.6}$$

where $Temp_{stsd}$ is the steady state temperature, $Temp_{start}$ is the initial temperature, R is the thermal resistance, and C is the thermal capacitance. The thermal resistance R and capacitance C are constants depending on the machine architecture. The steady state temperature of a task is the

temperature that will be touched if vast number of occurrences of the task execute continually on the machine. It bears an almost linear relationship with the power consumed, and is given by:

$$Temp_{stsd} = (Power \times R) + Temp_{amb}, \quad (4.7)$$

where R is the thermal resistance, as explained earlier and $Temp_{amb}$is the ambient temperature of the machine/core. The power consumption of tasks differ significantly depending on the nature of the task. Therefore, we can say that the steady state temperatures of tasks in a task set are different. In the quest to improve performance, continuous scaling of supplied voltage has been the focal point. Consequently, high operational frequency is exercised to meet high power needs. The dynamic power remains unaffected with the change in temperature. Nevertheless, the static power loss increases exponentially with temperature. The leakage current, I_0, is given as:

$$I_0 = I_s\left(ATemp^2 e^{(\mu_1 V+\mu_2)/Temp} + Be^{(\mu_3 V+\mu_4)}\right), \quad (4.8)$$

where $I_s, V,$ and $Temp$ is the initial leakage current, voltage supplied, and the operating temperature, respectively. Whereas, $A, B, \mu_1, \mu_2, \mu_3\ and\ \mu_4$ are constants with values determined empirically. The leakage current thereby increases exponentially with temperature, as shown in Fig. 4.2.

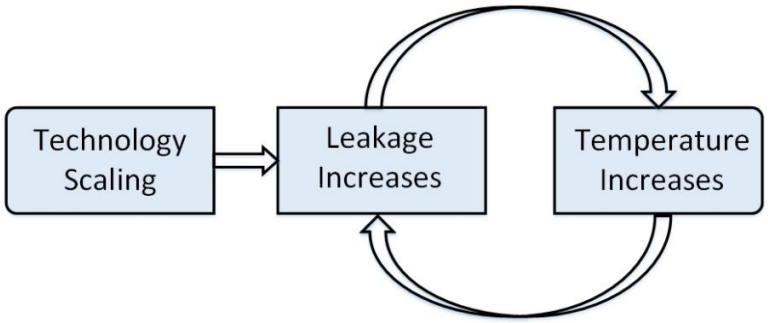


Fig. 4.2. Impact of technology scaling

Therefore, to avoid hotspots and thermal runaway temperature effects cannot be neglected. The total power consumption of a task running on a machine is given by;

$$P_{Total} = C_{EFF} f^3 + C_1 freq + C_2 Temp_f, \quad (4.9)$$

where C_1 and C_2 are the curve fitting constants [4.6]. Based on the above mentioned mathematical model to optimize the power consumption of a machine we adopt the DVFS-based power scaling. The temperature control is gauged by an optimum temperature bound check given in the next section.

4.5. Problem Formulation

Consider a given a resource allocation system that comprises a set of machines, M, and a set of tasks, T. The scheduler is required to map tasks on the machine set, such that all the characteristics of the tasks and the deadline constraint of T are fulfilled. We term this assignment as a feasible task to machine mapping. A feasible task to machine mapping happens when each task $\tau_i \epsilon T$ can be mapped to at least one M_j subject to all of the constraints associated with each task, such that the computational time, architecture, and deadline. The aforementioned requirements of the tasks are recorded as a Boolean operator (x_{ij}).

The task to machine mapping is performed such that, a minimization of the cumulative instantaneous power (P_{ij}) consumed by the machines in the scheduling system, the temperature and the makespan of the set of tasks, MS_{ij} is ensured. Power management is achieved by regulating the voltage and frequency supplied using the DVFS. The DVFS methodology exploits the convex relation between the power expended by a machine to the voltage and frequency exploited. The motivation of using the DVFS technique is to expand the task execution time using frequency and voltage reduction to minimize the overall power consumption. Table 4.1 presents the legend explanation.

Table 4.1. Linear program terminology

Variable	Description
i	Task indication variable/subscript
j	Indicates node/machine
x_{ij}	Binary variable
MS_{ij}	Makespan (completion time of task set "i" on machine "j")
t_i	Time to complete task "i"
P_{ij}	Power consumption of machine "j" when task "i"is performed
ψ_{ij}	Time to execution of task "i" on machine "j"
f	Frequency of the core/node
S	Storage memory required
F	Finishing time
C	Job consolidation resources
τ_i	Task $i \in T$
M	Set of machines
$Temp$	Temperature of machine
d_i	Deadline of task "i"
T	Set of tasks to be performed

Objective function

$$O = x_{ij}\left[Min\left(MS_{ij} + P_{ij} + Temp_{ij}\right)\right]$$

$$= \left(\sum_i \sum_j MS_{ij} x_{ij}\right)\alpha_{MS} + \left(\sum_i \sum_j P_{ij} x_{ij}\right)\alpha_p$$

$$+ \left(\sum_i \sum_j Temp_{ij} x_{ij}\right)\alpha_{Temp}. \quad (4.10)$$

s.t. $\forall i, j, where\ (i > 0\ and\ j > 0)$.

Bounding weight parameter

$$\sum_i \sum_j \alpha_{ij} \leq 1, \tag{4.11}$$

where α = proportional weight parameter

$$\sum_i \sum_j x_{ij} = 1, \quad \forall j \in M \quad and \quad \forall i \in T \tag{4.12}$$

$x_{ij} = 1$ *if task "i" is allocated to node "j" otherwise* 0.

Bounding Power Consumption (A Constraint)

$$\sum_i \sum_j p_{ij} \leq P_j^{max} \qquad \{P_j > 0\}. \tag{4.13}$$

Set of Constraints

(a) Execution time of task '*i*' on node '*j*', $\psi_{ij} \leq deadline_i \qquad \forall_i = tasks$

(b) Frequency of node, $f_j^{min} \leq f_{ij} < f_j^{max} \qquad (0 < f_j^{min})$

(c) Resource Consolidation, $C_{ij} \leq C_j$

(d) $Stoarge\ capacity, S_{ij} \leq S_j$

(e) Finish time of machine, $F_j \leq MS_{max}$

(f) $temp_{min} \leq temp_j \leq temp_{max}$

Assumptions in the formulation:

- The task characteristics are not expected to change during the execution course. That is, the deadline of task completion and resource requirement remains the same.
- The expected execution time of a task is considered as a decision criteria for scheduling a task on a node.
- P_{ij}, represents the power consumption of all of the components of a node.

- A task in execution process is not stopped until completion. However, the assigned resources may change, e.g. power consumption of the node.
- The cooling power of the scheduler is not included in calculations.

4.6. Pareto Front Approximation

We focus on the optimization problem with three objective functions, $f_1(x), f_2(x)$, and $f_3(x)$,. Although, there are variety of computational methods for procuring the aforementioned objective. The methodology incorporated for the approximation of the Pareto front used in our work is the dual simplex procedure. Each objective function is assigned a certain weight and the point of optimization is adaptively determined. At each iteration the non-dominated points are identified to construct the set of Pareto optimal points.

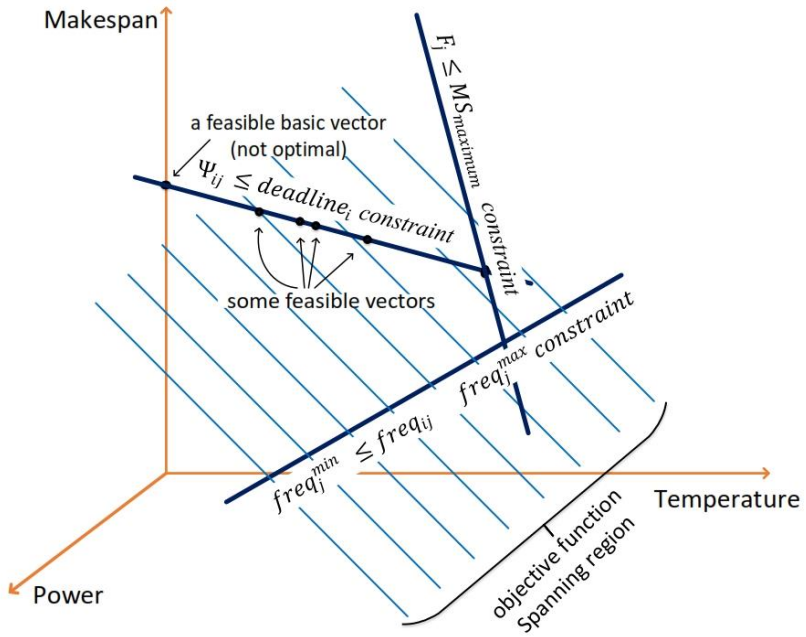


Fig. 4.3. Linear programming model

Figure 4.3 depicts the linear programming model considered in our paper. The co-ordinates represent the independent variables, depicted by the objective functions. The constraints restricts the allowable feasible region to a specified operating area, as shown in the Fig. 4.3. The dual simplex methodology will enable the system to operate in the optimal region. The desired region is identified by imposing the constraints and bounding the objective function by the feasible vectors. The feasible solutions that are dominated are discarded in the quest to find solution that after better optimization to make the set of non-dominated solutions. The Pareto front approximation seeked in this work, is given as:

$$\forall k \qquad f_k(x^*) \leq f_k(x), \tag{4.14}$$

$$\exists k \qquad f_k(x^*) \leq f_k(x), \tag{4.15}$$

where $f_k(.)$ represents the k^{Th} objective function. Nevertheless, x^* depicts the non-dominated solutions and x constitute the dominated solution. The elements of the set $f_k(x^*)$ indicates the desired optimized solution set and compose the coveted Pareto front.

4.6.1. Dual Simplex Method for the Linear Programming

We assume that the mapping (x_{ij}) is only fulfilled when the resource consolidation/architectural constraints of the tasks are satisfied. The resource consolidation refers to the desired level of storage, memory, power, and Virtual Machines (VMs) required to perform a particular task or set of task on a machine. The constraint optimization problem is resolved with the simplex method of linear programming. The feasible intersection points are enumerated using the complex (Simplex) method. Nevertheless, the worst point will be replaced by a new and better point using the aforementioned methodology. A variation in the parameter f is used to achieve an optimized solution to the problem.

Table 4.2. Generalized simplex tableau

Iteration	R_i	P_{ij}	$R_{mij}sr_j$	t_{ij}	F_j	$freq_j^{min}$	$freq_j^{max}$	S_1	S_2	S_3	S_4	S_5	S_6	S_7	B*
1								1							
		Values shall be attained according to the task to machine assignment							1						
										1					
											1				
												1			
													1		
														1	
	Values of the objective function														

** Values corresponding to the maximum limit (Boundary or Extreme)*

The systems' modeling is initiated by normalizing the constraints. The inequalities in the constraints list are converted into equations by adding a slack, and surplus respectively, such as:

$$Constraint_i + slack_i = feasible\ limit/bound \tag{4.16}$$

The simplex tableau for the objective function is generated comprising of the objectives, constraints, and artificial variables. The elements of the objective function play a pivotal role in the optimization methodology. The simplex method operations work in the form of a tableau. For every iteration a new tableau is generated to indicate the convergence. Moreover, the new tableau highlights the objectives function values that needs to optimized to achieve the overall optimization goal. The optimization process is characterized by the evaluation of feasible intersection points. Table 4.2 represents a generalized simplex tableau.

In the Table 4.2, the most negative co-efficient in the objective function row determines the pivot column. The columns pertaining to the variables S_i simply record the slack and surplus term of each of the constraint. First, the simplex procedure is employed to find the pivotal element. The pivot identifies the next intersection point to be evaluated that improves the objective. Thereafter, Gaussian elimination step is followed to attain the next simplex tableau.

The entries of the objective function row in the Table 4.2 determines the decision of generation a new tableau. The iterations stop until every element of the objective function takes a non-negative value, while adhering to the bounds of operation. Algorithm 4.1 details the procedure to achieve the non-dominated solution for the linear convex problem. A weighted sum approach is employed to generate the Pareto front. The admissible feasible solutions are generated. The solution that form the desired convex combination of the objectives are retained and vice versa. Formally, the vertices that restrict the objective function to the optimal corner point, make the set required basic solution. The algorithm initiates with the test of optimality that checks the current state of objective parameters. That is, if the elements constituting the last row, are positive, the optimality condition is already reached. Otherwise, the procedure evaluates the identification of the pivotal element that is triggered by the most negative entry of the simplex tableau. Using the elimination methodology the new transformed tableau is generated. The same check and do procedure is followed until the optimality is reached.

4.7. Simulation Results

To validate our proposed methodology, the scheduling operations performed are implemented in Matlab. To evaluate the proposed algorithm, we used a core i-7 desktop PC with 3.4 GHz of CPU speed and 8 GB of RAM. The dimensions of the tasks executed are as large as 10,000 tasks by a total of 20 computing nodes. The task mapping is restricted to the constraints and operational bounds listed in Section 4.4. The objective of the simulations performed is to maximize the performance while minimizing the power and temperature factor. The algorithm 4.1 depicts the weight parameter, α, for each of the desired objective. The purpose of utilizing α is to define the importance of a specific objective in a controllable manner.

Algorithm 4.1. Round to the nearest integer solution while maintaining the constraints

INPUT: The number of tasks, τ to perform, the number of machine, M, and objective to minimize;

OUTBUT: Optimal solution for the problem by executing the benchmark of required performance level;

INITIALIZATION: The control parameters of the objective function are provided;

1: **Step 1**
2: **if** $f_1, f_2, \dots, f_n \geq 0$ **then**
3: L_P problem is optimail
4: **else**
5: choose the most negative $f_1 < 0$
6: **Step 2**
7: compute the pivot element $\min_{1 \leq k \leq m}\{b_k/a_{kl}| > 0\}$
8: obtain a basic solution and update the objective function row as:
9: **Step 3**
10: $f_N = f_N - \beta(\gamma_p)$
11: where $\beta = b_\gamma / a_{\gamma l}$
12: and $\gamma_p =$ pivotal row element
13: update and generate the new simplex tableau
14: **end if**
15: Go to Step 1

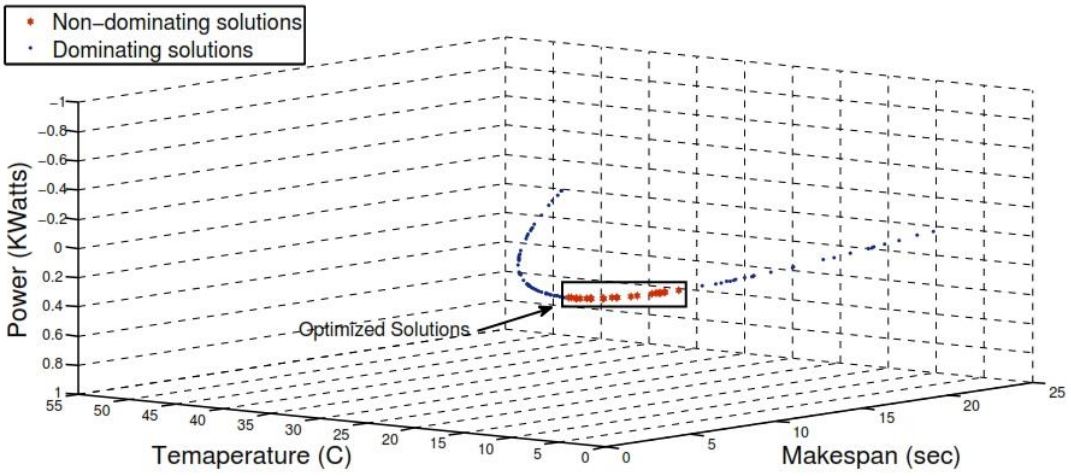


Fig. 4.4. Pareto front of the optimized solution set

Performance is quantified by the completion of a task by a machine in the targeted deadline. Note that, if the task completion is performed in the range of allowable power and temperature constraints, it is specified as efficient mapping. The solutions that adhere to the system specifications of defined objectives and constraints are used to construct the coveted Pareto front. Figure 4.4 depicts the ability of the proposed algorithm to efficiently minimize the desired objectives.

In Fig. 4.5 and Fig. 4.6, we validate the effectiveness of the linear programming model presented in the paper. The Fig. 4.5 depicts the temperature distribution of five machines for a makespan time ranging to 3000 secs. The peak temperature is constraint by a dotted line at 85 C. The temperature of each machine is below the aforementioned bounded limit to ensure the safe region operation. The adherence to the thermal constraint avoids the occurrence of hot-spots. Similarly, the Fig. 4.6 show the power consumption of the five machines for the similar specifications of makespan. The peak power of each machine is under the constraint of 250 Watts. From the results obtained we conclude that the proposed methodology is efficient in adhering to the system constraints of power and temperature.

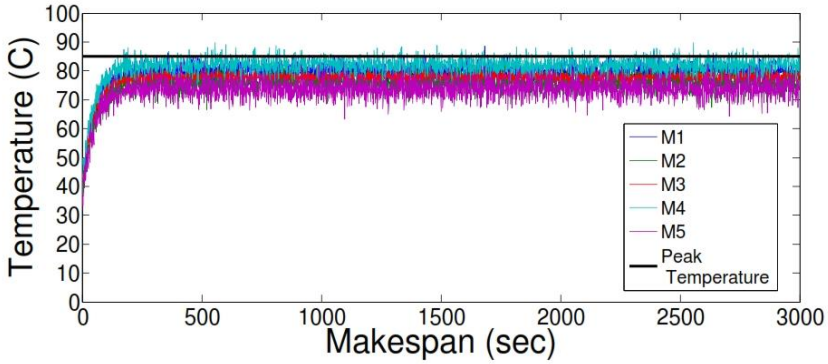


Fig. 4.5. Optimization of temperature

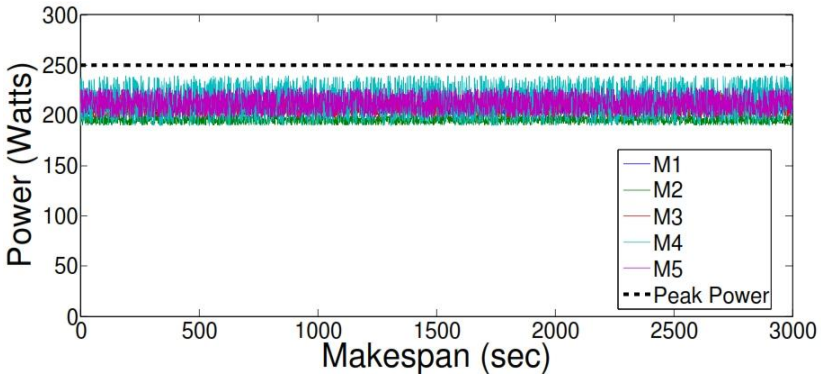


Fig. 4.6. Optimization of power consumption

In Fig. 4.4, the non-dominating solutions assume a computationally calculated Pareto front for the optimized solutions. However, the sub-optimal solutions are represented as dominating solutions. The non-dominating solution set sweeps out the dominated solutions from the knee region of the curve. The solutions that are less optimized form the tails of the Pareto-optimized graph. Nevertheless, the solutions that comprise the knee region are comparatively better in minimizing all of the three objectives. Consequently, the overall performance of a scheduler is improved by adhering to the abovementioned details. The allocated tasks when are performed by the available set of machines following the methodology depicted in the paper, a Pareto optimized front curve may be obtained.

4.8. Related Work

A large number of hardware and software techniques, for example [4.1], [4.7] and [4.8] have proposed by researchers to improve the energy profile of multi-core systems. The traditional power saving strategies focus on scaling the voltage and frequency of the core to meet the allowable power level. However, temperature received less attention. Consequently, reliability and decrease in the life-time of the chip resulted as a trade-off. Therefore, researchers over the last

decade, emphasize the need of *Dynamic Thermal Management (DTM)* [4.5] and [4.9] for safe chip operation and to reduced cooling cost.

The work presented by authors in [4.1]-[4.12] perform optimization of power consumption while guaranteeing the required performance. Nevertheless, the aforementioned methodologies optimizes the power and performance, but during the optimization. Authors in [4.13] speculates the chip thermal management requirement and devised methodologies to attain the chip temperature optimization. In Ukhov *et. Al* [4.14] the authors propose a Steady State Dynamic Temperature Profile (SSTDP) to realize temperature-aware reliability model. The technique consider mitigating the thermal cycling failure. However, transient faults and their management is not catered. Moreover, power optimization is not entertained while achieving reliability.

Significantly, different from the above listed work, this study explore the scheduling decision space to optimize the performance of multi-core system. The temperature and power utilization is capped and dynamically adjusted while meeting the performance requirement of the system.

4.9. Conclusions and Future Work

The exponential increase in the demand of Cloud deployment is constrained by the prohibitively high operational cost. To address the abovementioned issue, the work presented in our paper combined the benefit of power- and temperature-awareness in multi-core systems. A coherent framework of power, temperature, and makespan optimization is proposed to attain promising performance. Using the frequency of operation as selection criteria the task allocation is mapped to simultaneously minimize the aforementioned objective function entities. We proposed a formulation that caters the heterogeneity among resources and proposes bounds of operation to adjust to the varying demand of power, frequency, and temperature. Moreover, to

define precedence a weighted approach is utilized to significantly impact the desired objective and guarantee desired results.

The results reveal that the algorithm proposed is efficient in obtaining the trade-off front and removing the dominated solutions. The trade-off comprises the Pareto front and comprises of the non-dominated solutions. For future work, we plan to investigate the methodology on an extended scale of performance objectives. The particular domains of interest encompass throughput maximization and reduction of network congestion.

4.10. References

[4.1] K. Bilal, A. Fayyaz, S. U. Khan, and S. Usman, "Power-aware resource allocation in computer clusters using dynamic threshold voltage scaling and dynamic voltage scaling: comparison and analysis," *Cluster Computing*, pp. 1–24, 2015.

[4.2] J. Kołodziej, S. U. Khan, L. Wang, and A. Y. Zomaya, "Energy efficient genetic-based schedulers in computational grids," *Concurrency and Computation: Practice and Experience*, 2012.

[4.3] A. Abbas, M. Ali, A. Fayyaz, A. Ghosh, A. Kalra, S. U. Khan, M. U. S. Khan, T. De Menezes, S. Pattanayak, A. Sanyal et al., "A survey on energy-efficient methodologies and architectures of network-on-chip," *Computers & Electrical Engineering*, vol. 40, no. 8, pp. 333–347, 2014.

[4.4] S. U. Khan and I. Ahmad, "A cooperative game theoretical technique for joint optimization of energy consumption and response time in computational grids," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 3, pp. 346–360, 2009.

[4.5] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, "Temperature-aware microarchitecture: Modeling and implementation," *ACM Trans. Archit. Code Optim.*, vol. 1, no. 1, pp. 94–125, Mar. 2004.

[4.6] H. Huang, V. Chaturvedi, G. Quan, J. Fan, and M. Qiu, "Throughput maximization for periodic real-time systems under the maximal temperature constraint," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 2s, p. 70, 2014.

[4.7] J. Shuja, K. Bilal, S. A. Madani, and S. U. Khan, "Data center energy efficient resource scheduling," *Cluster Computing*, vol. 17, no. 4, pp. 1265–1277, 2014.

[4.8] D. Kliazovich, P. Bouvry, and S. U. Khan, "Dens: data center energy-efficient network-aware scheduling," *Cluster computing*, vol. 16, no. 1, pp. 65–75, 2013.

[4.9] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini, and G. De Micheli, "Temperature control of high-performance multi-core platforms using convex optimization," in *Design, Automation and Test in Europe, 2008. DATE'08*, 2008, pp. 110–115.

[4.10] P. Lindberg, J. Leingang, D. Lysaker, S. U. Khan, and J. Li, "Comparison and analysis of eight scheduling heuristics for the optimization of energy consumption and makespan in large-scale distributed systems," *The Journal of Supercomputing*, vol. 59, no. 1, pp. 323–360, 2012.

[4.11] J. SiYuan, "A novel energy efficient algorithm for cloud resource management," *International Journal of Knowledge and Language Processing*, vol. 4, no. 2, 2013.

[4.12] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C.-z. Xu, and A. Zomaya, "Energy-aware parallel task scheduling in a cluster," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1661–1670, 2013.

[4.13] J. Zhou and T. Wei, "Stochastic thermal-aware real-time task scheduling with considerations of soft errors," *Journal of Systems and Software*, 2015.

[4.14] I. Ukhov, M. Bao, P. Eles, and Z. Peng, “Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems,” *in Proceedings of the 49th Annual Design Automation Conference*, 2012, pp. 197–204.

5. CONCLUSIONS

In this chapter, we discuss the conclusion of the research we have performed during Ph.D. We carried out our research on the measurement and analysis of robustness of resource allocation system in Cloud for finding fault resilient solutions. In our research studies, we focused on the enhancement of resource allocation of tasks to a set of machines. In the first case a robustness measurement and analysis methodology is devised. Nonetheless, in the succeeding case, we obtained a Pareto optimized set of solutions for the improvement of a resource allocation system. Based on our study, we devise a formulation that unveils bounds on the desired objectives for the achievement of optimization. We analyzed that the frequency of operation when constrained to certain limit of operating domain can benefit the scheduler in optimizing the power and temperature.

We analyzed and implemented a geometrical dimension reduction mathematical model for the evaluation of robustness of resource allocation schemes in the cloud. The presence of uncertainty in the system parameters is considered and n-number of performance parameters are entertained that depicts the wideness of the approach. Our results reveal that the process of dimension reduction is dependent on the order of the parameter selected during the convergence procedure. The novelty of this work is the freedom of incorporation of the performance parameters required for robustness evaluation. The results achieved after reduction retain a reflection of all of the parameters utilized in the convergence process. The proposed method can be used to gauge robustness and observe the most effective allocation scheme among a group of allocation schemes that are apparently hard to distinguish. Moreover, the presented framework can be extended to a customized scenario to meet the QoS according to the required SLA, in a cloud environment. We

have presented two theorems that strengthen our reduction approach linked to the robustness measurement procedure.

In this thesis, we also presented the optimization of power- and temperature-awareness in multi-core systems. A coherent framework of power, temperature, and makespan optimization is proposed to attain promising performance. The relevant Pareto front of high quality is obtained for the optimization of the abovementioned objectives. Using the frequency of operation as selection criteria the task allocation is mapped to simultaneously minimize the aforementioned objective function entities. We proposed a formulation that caters the heterogeneity among resources and proposes bounds of operation to adjust to the varying demand of power, frequency, and temperature. Moreover, to define precedence a weighted approach is utilized to significantly impact the desired objective and guarantee desired results.

In future, we intend to extend our model by incorporating more number of objective parameters to address and optimize for attaining high-end performance. For instance, the improvement of throughput can increase the efficiency of the scheduler. Moreover, a reduction in network congestion also plays an important role in expediting the task completion. All such real-life parameters are having significance and must be considered in the design of an efficient resource scheduling models.