www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 3 March 2017, Page No. 20500-20503
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i3.18

Robust Performance-Based Resource Provisioning

Ms.Sreeja B P[1], Ms.SarathaDevi[2],Ms.Narmatha K[3]
[1,2,3] Assistant Professor,
Department of Information Technology,
Karpagam College of Engineering,Coimbatore,Tamilnadu,India.
sreejabp@gmail.com
sarathadevi.techno@gmail.com
knarmatha18.cool@gmail.com

Abstract: ***Cloud computing has enabled entirely new business models for high-performance computing. It is a dedicated local high-performance computer is still an option for some, but more are turning to cloud computing resources to fulfil their high-performance computing needs. With cloud computing it is possible to tailor your computing infrastructure to perform best for your particular type of workload by selecting the correct number of machines of each type. This paper presents an efficient algorithm to find the best set of computing resources to allocate to the workload. This research is applicable to users provisioning cloud computing resources and to datacenter owners making purchasing decisions about physical hardware. Studies have shown that cloud computing machines have measurable variability in their performance. Some of the causes of performance variability include small changes in architecture, location within the datacenter, and neighboring applications consuming shared network resources. The proposed algorithm models the uncertainty in the computing resources and the variability in the tasks in a many-task computing environment to find a robust number of machines of each type necessary to process the workload. In addition, reward rate, cost, failure rate, and power consumption can be optimized, as desired, to compute Pareto fronts.***

Keywords — datacenter, workload, Cloud Computing

1.INTRODUCTION

Some high-performance computing users are turning to cloud providers to complete their work due to the potential cost effectiveness and/or ease of use of cloud computing. The ability to provision hardware on-demand from a pre-defined set of different machine types, known as instance types, is very powerful. In fact, a proof of concept cluster was built by Amazon Web Services from their high performance instance types composed of over 26,000 cores with nodes connected via 10G Ethernet. This cluster ranked 101 on the Top 500 list for November 2014 [1]. The cloud has been successfully employed to process HPC jobs for actual scientific experiments [2]. Recent studies have shown that the performance of small and medium virtual clusters can compete with physical hardware. Cloud infrastructure as a service (IaaS) providers charge for the amount of time a virtual machine, known as an instance, is allocated (idle or active). This means that it is advantageous to terminate some or all instances once the workload has been processed. Leaving instances idle in the cloud is usually not cost effective. Once a new set of work needs to be processed, the decision of what instance types to start can be re-evaluated each time, considering the size and Sciences and Engineering instance types. Selecting the ideal number of instances of each instance type a user needs is challenging. The approach to provisioning computational resources given in this paper not only applies to cloud resource provisioning but also to selecting physical machines to purchase for use within HPC systems. The goal for provisioning HPC systems is to determine how to originally select or upgrade a system in such a way that maximizes the performance of the resultant system while meeting specific requirements that often include a budget constraint. The instance types available in the cloud have widely varying capabilities, by design, so that users can choose the resources that best match their workload and in doing so minimize the cost. For example, there is no need to provision high memory instance types if the workload does not require large amounts of memory. The cost for the smaller memory instance type will often be significantly less and provide nearly identical performance assuming all else is equal. Within a single IaaS provider, instance types vary in the amount of memory, number and type of CPUs, disk capacity and performance, and network performance. All of these properties of instance types affects the performance of the workload executing on the instances [5]. Due to the availability of heterogeneous resources, IaaS is inherently a heterogeneous computing system.

2.DESIGN AND IMPLEMENTATION

A. Performance Results

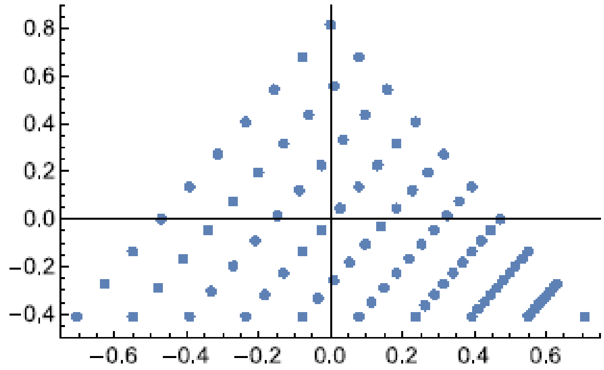
Stochastic linear programming is an extension of linear programming, where some of the coefficients in the objective and the constraints are random variables.The particular stochastic program we use is the recourse problem (RP) given in standard form as

minimize cT x + E [Q(x,£)]

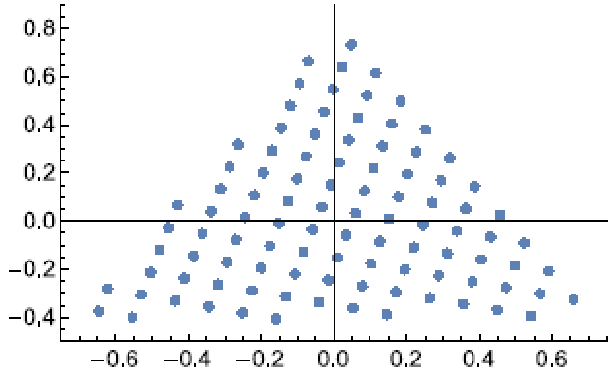
where £ is a random vector representing the uncertain parameters. For the RP in , the first stage decision variable, x is a flattened version of MB and MS. The second stage decisions, y are flattened versions of the schedule p. The coefficients that are deterministic are incorporated into c and the coefficients that are random are incorporated into q.This linear RP is similar to a linear program except for the expectation of the value function, Q(x,£), in the objective. The RP in (13) is known as a two-stage RP. The optimization problem finds the optimal x that minimizes the sum of the linear function cT x and the expectation of Q.1 The second stage optimization problem finds optimal y given a fixed realization of _. The random vector y is known as the recourse decision vector. This RP finds a robust solution for x in the sense that the objective value will on average be minimal when the optimal value of x is used. The solution x is robust to unknown values of the parameters.The vector x is often referred to as a strategy of the RP.

The primary issue with stochastic programming is accurately computing the expectation in the objective function. The SAA approach uses many samples of to compute the expectation as a sample average. Realizations of are called scenarios. The process of creating scenarios is known as scenario generation. When using SAA, generating a reasonably small set of representative scenarios is important. Let there be K scenarios. For scenario k, let the probability of occurring be given by pk. With SAA, the scenarios are generated by randomly sampling according to its distribution, thus all the samples are equally probable. The expectation operator is linear and is applied to a linear function, namely qT y(£). Each scenario (i.e., realization of the) defines a set of matrices Tk andWk, and vectors qk and hk. Each scenario also introduces a new vector of decision variables yk into the problem. The SAA is an unbiased estimator of the mean. In practice, it converges to the mean quickly in K. The DEP can have a large number of variables and constraints. For very large problems a technique called the L-method can be used to exploit the block structure of the constraint matrix to distribute the work of solving the linear program to many nodes.The problem is broken into two coupled decisions. The first is what to provision or purchase, namelyMB andMS. Then the random variables in the problem are realized and the second decision, known as the recourse decision, can be made. For this work, the random variables are the arrival rates, execution times, and power consumption, but virtually any other parameter in the model can be converted to a random variable. The recourse decision involves selecting the schedule p that is optimal for the actual arrival rates, execution times, and power consumption of the tasks.

$$
\begin{array}{lllllll}
\underset{\mathbf{x},\mathbf{y}_k}{\text{minimize}} & \mathbf{c}^T\mathbf{x} & +p_1\mathbf{q}_1^T\mathbf{y}_1 & \cdots & +p_K\mathbf{q}_K^T\mathbf{y}_K & & \\
\text{subject to:} & \mathbf{A}\mathbf{x} & & & & = \mathbf{b} \\
 & \mathbf{T}_1\mathbf{x} & +\mathbf{W}_1\mathbf{y}_1 & & & = \mathbf{h}_1 \\
 & \vdots & & \ddots & & \vdots \\
 & \mathbf{T}_K\mathbf{x} & & & +\mathbf{W}_K\mathbf{y}_K & = \mathbf{h}_K \\
 & \mathbf{x}, & \mathbf{y}_1, & \cdots & \mathbf{y}_K & \geq 0
\end{array}
$$



(a) recursive



(b) orthogonal

Multi-objective optimization is challenging because there is usually no single solution that is superior to all others. Instead, there is a set of superior feasible solutions that are referred to as the non-dominated solutions [31]. When all objectives are to be minimized, a feasible solution x1 dominates a feasible solution x2 where dth objective function. Feasible solutions that are dominated are generally of little interest because one can always find a better solution from the non dominated set. The non-dominated solutions, also known as outcomes and efficient points, compose the Pareto front. There are many techniques for solving multi-objective optimization problems. For linear optimization problems, there are two primary approaches. The first is known as Benson's algorithm that iteratively refines the Pareto front. The second is a technique that converts the multi-objective problem into a set of scalar optimization problems through a process called scalarization. There are many scalarization techniques but most are specializations of Pascoletti-Serafini scalarization such as the weighted sum algorithm.

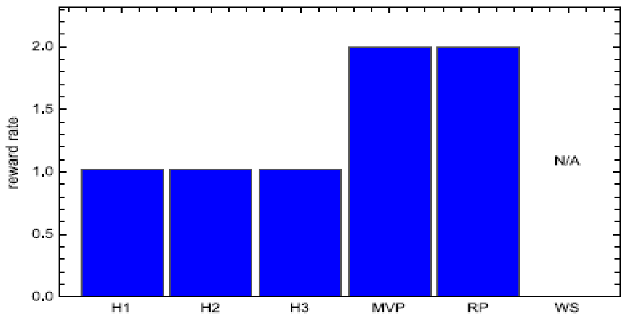
$$
\begin{array}{ll}
\forall d & f_d(x_1) \leq f_d(x_2) \\
\exists d & f_d(x_1) < f_d(x_2)
\end{array}
$$

Three very different environments are used to analyze the behavior of the proposed algorithms for resource provisioning. The heuristic-based algorithms H1, H2, and H3, and the RP that uses stochastic programming are compared. The first environment is a small example used to illustrate the behavior of the algorithms. The second environment is a larger environment. The third environment was built based on benchmark data. A complete description of the sourc code, system parameters, and simulation results are available in the

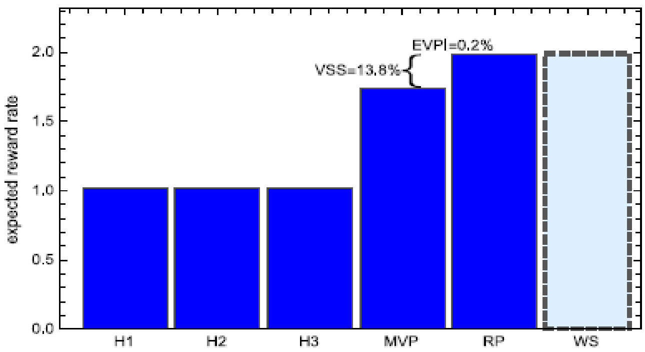
DOI: 10.18535/ijecs/v6i3.19

supplementary material to aid in reproducibility. The data is provided as CSV and JSON files with further
details available in the accompanying README.txt file.

The steady-state model and the scenario generation are written in C++. To generate the DEP in (14) the Coin-OR Stochastic Modeling interface (SMI) is used [38]. The underlying linear programming problem is solved with the Coin-OR Linear Programming (CLP) solver [39]. CLP is a high quality open-source solver written in C++. All the simulations where run on an Apple MacBook Pro Mid 2014, 2.2 GHz Intel Core i7. The solver is single threaded so timing results are for one core.To compare MVP, RP, and WS against the heuristics described in Section 7, one must be able to compute all the objectives including reward rate. The reward rate is a function of the steady-state schedule. To allow heuristics to perform as best as possible we use the optimal schedule from the steady-state model by solving a linear programming problem where the MB is fixed by the heuristic and MS = 0. When computing the expected objective values the optimal schedule is used for each scenario.



(a) objective value with mean parameters



(b) expected objective value

(a) Shows the reward rate computed with the mean of the parameters.
(b) Shows the expected reward rate over all uncertainty in the parameters

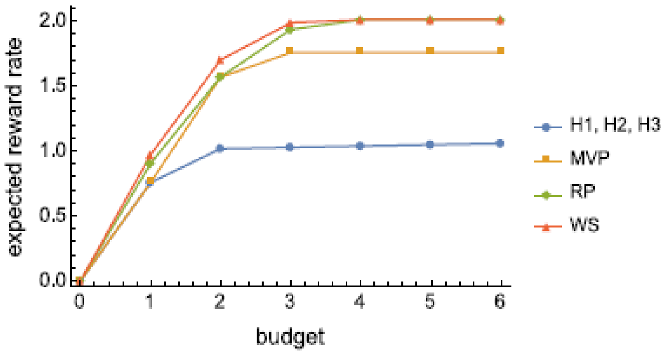


Fig. 3: Expected reward rate for different budgets for the E1 environment: The reward rate asymptotes at 2:0 with RP approaching WS. The other algorithms never reach the maximum reward rate.

3.Conclusion

Stochastic programming is a powerful tool that can be applied to make robust decisions in the midst of the inherit uncertainty in computing systems in both IaaS provider clouds and traditional environments. The linear steady-state model and representative stochastic model enables the use of an efficient two-stage stochastic program for solving the machine provisioning problem.

References

[1] I. Zinno, L. Mossucca, S. Elefante, C. D. Luca, V. Casola, O. Terzo, F. Casu, and R. Lanari, “Cloud computing for earth surface deformation analysis via spaceborne radar imaging: A case study,”IEEE Transactions on Cloud Computing vol. 4, no. 1, pp. 104–118,Jan 2016.

[2] E. Roloff, F. Birck, M. Diener, A. Carissimi, and P. O. A. Navaux,“Evaluating high performance computing on the windows azure platform,” in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on,June 2012, pp. 803–810.

[3] J. Gibson, R. Rondeau, D. Eveleigh, and Q. Tan, “Benefits and challenges of three cloud computing service models,” in Fourth International Conference on Computational Aspects of Social Networks (CASoN), 2012, pp. 198–205.

[4] W. Lloyd, S. Pallickara, O. David, J. Lyon, M. Arabi, and K. Rojas,“Performance implications of multi-tier application deployments on infrastructure-as-a-service clouds: Towards performance modeling,” Future Generation Computer Systems, vol. 29, no. 5, pp. 1254–1264, 2013.

[5] A. Iosup and D. Epema, “Grid computing workloads,”IEEE Internet Computing, vol. 15, no. 2, pp. 19–26, March 2011.

[6] 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.

[7] J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, and Z. Gu, “Online optimization for scheduling preemptable tasks on iaas cloud systems,” Journal of Parallel and Distributed Computing, vol. 72, no. 5, pp. 666–677, 2012.

[8] J. G. F. Coutinho, O. Pell, E. O’Neill, P. Sanders, J. McGlone, P. Grigoras, W. Luk, and C. Ragusa, “HARNESS project: Managing heterogeneous computing resources for a cloud platform,” in Reconfigurable Computing: Architectures, Tools, and Applications. Springer, 2014, pp. 324–329.

[9] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, “Improving mapreduce performance in heterogeneous environments,” in 8th USENIX Conference on Operating Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2008.

DOI: 10.18535/ijecs/v6i3.19

[10] Q. He, S. Zhou, B. Kobler, D. Duffy, and T. McGlynn, "Case study for running HPC applications in public clouds," in 19th ACM International Symposium on High Performance Distributed Computing, 2010, pp. 395–401.
[11] E. Jeannot, E. Saule, and D. Trystram, "Optimizing performance and reliability on heterogeneous parallel systems: Approximation algorithms and heuristics," Journal of Parallel and Distributed Computing, vol. 72, no. 2, pp. 268–280, 2012.
[12] J. Koomey, "Growth in data center electricity use 2005 to 2010," The New York Times, vol. 49, no. 3, 2011.
[13] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Nov 2011, pp. 1–12.
[14] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Cost- and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds," in International Conference for High Performance Computing, Networking, Storage and Analysis (SC), Nov 2012, pp. 1–11.
[15] K. Jansen and L. Porkolab, "Improved approximation schemes for scheduling unrelated parallel machines," Mathematics of Operations Research, vol. 26, no. 2, pp. 324–338, 2001. [Online]. Available: http://dx.doi.org/10.1287/moor.26.2.324.10559
[16] Sreeja B P , Saratha Devi G, "E-shopping community structure analysis using data clustering" International Journal of Engineering and Computer Science, vol. 6, issue 3.
[17] O. Beaumont, A. Legrand, L. Marchal, and Y. Robert, "Steadystate scheduling on heterogeneous clusters," International Journal of Foundations of Computer Science, vol. 16, no. 2, pp. 163–194, 2005.

Author Profile

Ms.Sreeja B P,
Working as an Assistant Professor in IT Department KCE from 2011,Has received her M.Tech Degree in Computatioanl Engineering and networks, B.Tech Degree in Information Technology&Her area of working includes Wireless Sensor Networks,Machine Translation,Cloud Computing.

Ms.Saratha Devi G,
Working as an Assistant professor in the department of Information Technology in Karpagam College of Engineering,Coimbatore,Tamil Nadu from 2012.She is doing her Ph.d in Information and Communication Engineering in Anna university,Chennai. She received M.E degree in Karpagamuniversity, Coimbatore. She received her B.E in Avinashilingam university, Coimbatore.

Ms.Narmatha K,
Received my bachelor's degree from SNS College of Technology, Anna University, Coimbatore, India in Computer Science and Engineering during 2011. Received my Master's degree in Embedded Systems from Kumaraguru College of Technology, Anna University, Coimbatore, during 2013. I'm currently working as Assistant Professor at the department of Information Technology in Karpagam College of Engineering, Coimbatore, India.

.