Cost Effective Self-Tuning Cloud Application With Multiple Cores

S.Vykuntarao
PG Scholar
Department of CSE
JNTUA,Ananthapuramu
simmavykunta@gmail.com

K.Madhavi
Associate Professor
Department of CSE
JNTUA,Ananthapuram
kasamadhavi.cse@jtua.ac.in

ABSTRACT:Cloud computing provides large pool of shared computing resources. This new computing paradigm enables large concurrent applications that need access to computing resources on demand. The partitions of the workloads of these concurrent applications are executed by the compute nodes in the cloud. Historically, the application designers target their applications for specific hardware and environments. However, such applications need to have different design with cloud computing which enables different platforms. Self-tuning is essential for concurrent applications in cloud platforms. It is very challenging problem to design such self-tuning applications that can split workload and achieve high cost efficiency in cloud. Many researchers contributed towards self-tuning cloud applications. Of late Rajan and Thain proposed a methodology that enables adaptive self-tuning split-map-merge applications to have cost-effective processing. the optimization will reduce system overhead. However, it could be improved further with the division of workload with the intention of using multiple cores for execution. The proposed system focuses on building application that is capable of self-modelling and self-tuning besides having ability to divide workload in order to utilize multiple cores for execution. A new algorithm is proposed in order to estimate the need for workload divisions and the number of cores for the execution. The proposed application will be more cost-efficient and a prototype is built to demonstrate the cost effective.

Keywords:Cloud computing, scientific applications, Map Reduce, resource provisioning, data partitioning

INTRODUCTION:

Applications that are data-intensive need to divide workload in order to have better execution of jobs. Such applications can be run with proper access to infrastructure and provision for dynamic resource allocation at runtime. The environment to achieve this was there earlier in the form of proprietary or private setups. With the emergence of cloud computing, now it is possible to use publicly available computing resources to run such applications. Especially, this paper throws light on the elastic applications whose workload differs from time to time. When the workload is different, it needs resources differently. Allocating resources using resource reservation approach can cause wastage of resources. At the same time when resources are not provisioned, jobs cannot be completed in the given deadline. Many researchers investigated with problem as found in [1], [8], 11], and [14]. In the literature, it is found that there are many issues with resource allocations for data-intensive applications. The dynamic nature of elastic applications is major issue. Allocation of resources in the presence of workload heterogeneity and multiple cores availability is the main focus of this paper. The following are the contributions of this paper.

1. A framework is proposed to have dynamic resource applications in the presence of data-intensive elastic applications that exhibit workload heterogeneity. The framework is based on the Cloudsim which is widely used for cloud simulations.
2. We proposed an algorithm named Adaptive Workload Division based Resource Allocation (AWDRA) which has an iterative approach to examine workloads and divide them into partitions. Then it analyzes resource availability and resource estimation to make strategic decisions in the presence of multiple cores.
3. A prototype application is built using Cloudsim API and Java programming language with Graphical User Interface (GUI) to demonstrate proof of the concept.

The paper has been organized as follows. Section 2 reviews literature on the resource provisioning of data-intensive applications in the cloud environments. Section 3 presents the proposed framework and the underlying algorithm for resource allocation. Section 4 presents experimental results and prototype implementation. Section 5 concludes the paper and provides possible future scope of the research.

RELATED WORK:

This section provides review of literature on resource provisioning in data intensive applications that run in distributed environments. Dynamic provisioning of resources to multi-tier applications is studied in [1]. With respect to e-Commerce systems, provisioning of servers based on the need for e-commerce application is focused in [2]. Service Level Agreements (SLAs) is considered in [3] for dynamic provisioning of cloud databases in consumer-centric approach. With respect to multi-tenant systems, elastic resource scaling is the study made in [4]. Dynamic provisioning with an analytical model based on the regression analysis is made in [5]. It was to improve performance of multi-tier applications. In the utility based computing, where commodity commuters are used in cloud computing, virtualized resources are provisioned with adaptive controlling [6]. With respect to scientific workflows, resource provisioning options are explored in [7]. Scientific applications that run in distributed environments are explored in [8] to have high performance computing with market based resource allocation. The tradeoffs between cost and wait time with client side provision is studied in [9] for elastic clouds. Resource provision to enable high performance cloud computing in the presence of scientific applications is focused on [10]. In cloud computing environment, flexible provisioning of required resources with optimization is performed in [11]. Adaptive resource provisioning with constraints on budget is made in [12]. Resource provisioning with the help of

workflow partitioning is made in [13] for better performance. Parallel partitioning with distributed workflow execution provision is studied in [14] while optimization and re-optimization with parallel computing is explored in [15]. Similar kind of work is made in [16]. Data-intensive applications with time and cost awareness for resource allocations and optimizations are focused in [17] while the deployment of computations with orchestration is studied in [18]. The existing literature showed that there is need for adaptive nature of data-intensive applications to have resource provisioning automatically based on the size of workload. In this paper, we proposed a framework to achieve this.

PROPOSED SYSTEM:

This section provides the proposed framework that is used to have resource provisioning dynamically to elastic applications that exhibit workload heterogeneity. The framework is as shown in Figure 1. The framework is based on the programming model described here. The programming model is Map Reduce model that is split-map-merge. With respect to elastic applications this kind of programming is used and the concept is based on the Equations given below.

Workload:$(N) \rightarrow O$ (1)

Split(N,k):$N\{s_1,s_2 \ldots\ldots s_k\}$ (2)

Map(k) : $f(s_i) \rightarrow O_i$ for $i = 1,2,..k$ (3)

Merge(k) : $\{O_1,O_2,\ldots\ldots\ldots\ldots O_k\} \rightarrow O$ (4)

The Map Reduce programming model has map and reduce phases. The Map phase takes care of processing given key-value pairs and produce intermediate output. Then the reduce phase works on the outcomes of map phase. However, in this paper, the focus is on resource provisioning to elastic applications in the presence of workload heterogeneity and multiple cores availability. The Map Reduce model with essential changes is shown inFigure.

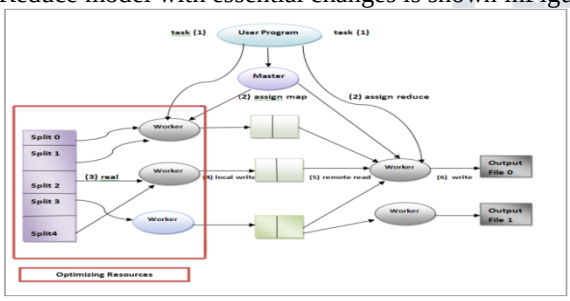

Figure 1: Shows Map Reduce where underlying framework needs optimization in resource allocation

As shown in Figure 1, it is evident that the data-intensive applications need to have distributed programming approach such as Map Reduce. It also needs to take care of resource allocation dynamically for elastic applications. The workload of the applications need different amount of resources. This is crucial as the jobs in the workload to be completed are to be processed as per the deadline expectations. We proposed an algorithm to have dynamic resource allocation with adaptive workload division in the presence of workload heterogeneity and availability of multiple cores.

Adaptive Workload Division based Resource Allocation Algorithm

An algorithm named adaptive workload division based resource allocation is proposed and implemented. This algorithm exploits the workload of elastic applications in order to have better resource allocations in the presence of multiple cores. It is adaptive in nature as it continuously monitors workloads and based on the workload size, partitioning and determination of resource availability and estimation of the resources needed besides allocation of resources optimally. Thus the resource allocation in the case of elastic applications is optimized and made cost effective.

Algorithm 1: Adaptive workload division based resource allocation

The algorithm takes workload as input and partitions it. Then it estimates the resources needed to complete the given job. If the resources are available sufficiently, it submits job to resources. If not the resources are requested or the existing partitions are subjected to merge in order to have proper resource allocations. The process is iterative in nature and adapts to the runtime workload dynamics of elastic applications.

Algorithm: Adaptive Workload Division based Resource Allocation Algorithm
Input : Work Load of Elastic Application
Output : Resource Allocation
Process:
Adoptive Resource Allocation
1.Partition the input workload
2.Determine allocation of resources for job
3.if resources are available
4.Submit the job to resources
5.Else if resources are not available
6.Find and allocate the resources
7. Calculate measurements
8.if partition size is big to continue step1
9.Merge the partitions
10.Return Allocation

IMPLEMENTATION AND RESULTS

The proposed system is implemented with Cloudsim framework. Cloudsim is the special simulator to develop applications that simulate cloud applications and cloud based research contributions. It has provision for simulating data centres, host machines, virtual machines and various algorithms that can be used for scheduling and load balancing. In this paper we evaluated the proposed system suing Cloudsim to know how it is cost effective in presence of heterogeneous workloads and the availability of multiple cores with respect to resource allocation.

Cloudsim Architecture

Cloudsim framework has three layers. Each layer has its own features and functionalities. The bottom most layer is the Cloudsim core simulation engine. The middle layer is the layer which provides cloud resources, cloud services, VM services, and user interface structures and so on as part of Cloudsim.The top most layer is the place where user applications run. User code runs in this layer.

As presented in Figure 2, the Cloudsim architecture provides various components that are reused in the applications. User applications make use of Cloudsim API in order to demonstrate proof of the concept related to resource provisioning which is adaptive nature in the presence of heterogeneity or workloads and availability of multi-cores.

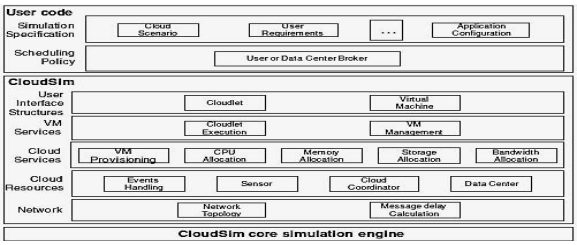

Figure 2: Overview of Cloudsim architecture

Though Cloudsim has no provision for Graphical User Interface (GUI), we built GUI for making the application intuitive. Once dataset related to workload is provided to the system, it performance its functionality on the given workload in order to optimize resource allocation. It employs the proposed algorithm in order to achieve this. The GUI application that helps in loading datasets and analyzing how it works with resource allocation in cost-effective fashion is built as prototype application.

Once the workload is provided to the application, it will analyze it with different requirements of jobs and then the algorithm makes use of the details to have adaptive workload division, resource analysis or estimation and then resource allocation.

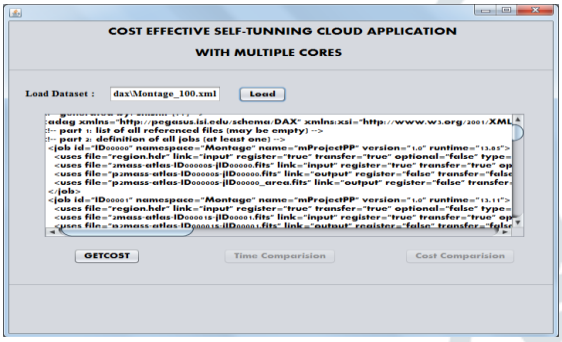

Figure 3: Prototype showing workload

The algorithm simulates the proposed system using Cloudsim API and returns the results of the experiments. Figure 3 shows experimental results.

RESULTS:

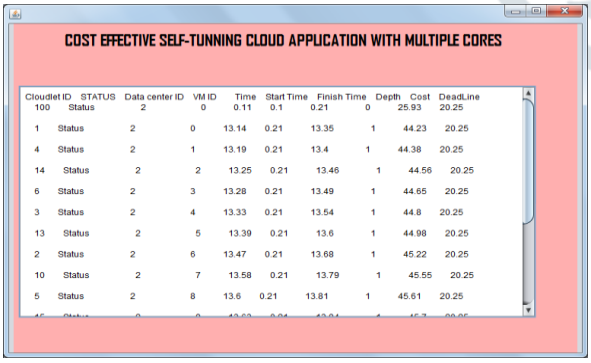

Figure 4: An excerpt from the results showing cost-effective resource allocation

The jobs are completed as per the configurations provided in the workload file. However, the proposed algorithm has its impact on the functioning of the simulator in division of workload and provides resource analysis besides resource allocation. The results show different cloud sets used in the execution. It also provides details of the IDs of data centres used, IDs of virtual machines, time, start time, end time and the cost. The cost and time analysis are made and the results are presented in Figure 5 and Figure 6.

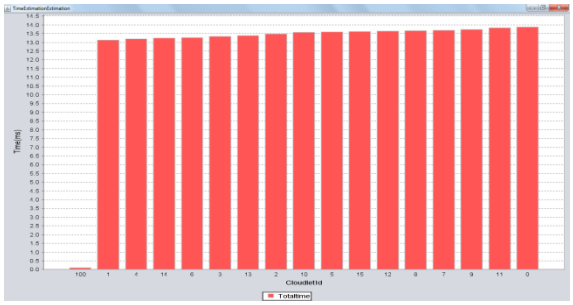

Figure 5: Time estimation details for different jobs (cloudlets)

Each cloudlet is nothing but a job which needs resource allocation and the time to complete the job. The horizontal axis presents cloudlet id while the vertical axis is representing the time in milliseconds. The results revealed that different jobs involved in the execution simultaneously and they took time according to the workload. In the same fashion, cost of the jobs is estimated and shown in Figure 6.

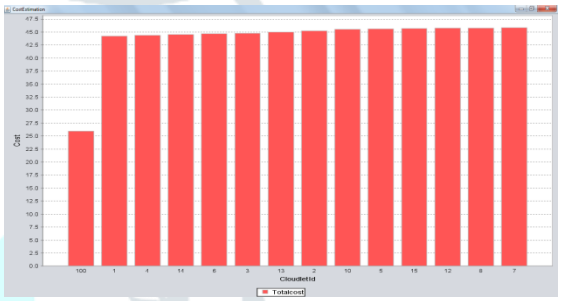

Figure 6: Cost analysis

As presented in Figure 6, it is evident that the proposed system is able to analyze cost and make well informed decisions to have resource allocation optimized. The results show the cloud let ids in the horizontal axis and the vertical axis shows cost estimated. Based on that the algorithm makes resource provisioning and ensure that the proposed system is cost effective in presence of workload heterogeneity and multiple cores. The research carried out in this paper showed that workload analysis, job allocation, resource allocation, utility of cloud data centres, usage of virtual machines with different capacities can be analyzed to have complete understanding of the dynamics of the solution provided

CONCLUSIONS AND FUTURE WORK

In this paper, elastic applications has been studied and the need for dynamic resource allocation in presence of workload heterogeneity and multiple cores. It is understood that resource allocation has to be made based on the jobs and the workload they carry. An algorithm has been proposed by name adaptive workload division based resource allocation. It has an iterative process in which workload is analyzed and partitioned. Then the job of partitioning is adaptive in nature based on the resource availability and resource estimation to complete given job. The resource availability and the estimation of resources needed by the current job helps the algorithm to make well informed decisions on the provisioning of resources and complete jobs as expected. Besides the algorithm has cost estimation and the resources are provisioned in cost-effective manner. The solution is based on Cloudsim framework. A prototype application is built to demonstrate proof of the concept. The experimental results revealed the usefulness of the application. In future we intend to explore resource allocation with respect to big data streaming applications.

REFERENCES:

[1] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood, "Agile dynamic provisioning of multi-tier internet applications," ACM Trans. Auton. Adapt. Syst., vol. 3, no. 1, pp. 1–39, Mar. 2008.

[2] D. Villela, P. Pradhan, and D. Rubenstein, "Provisioning servers in the application tier for e-commerce systems," ACM Transactions on Internet Technology, vol. 7, no. 1, Feb. 2007.

[3] S. Sakr and A. Liu, "Sla-based and consumer-centric dynamic provisioning for cloud databases," in Proceedings of the IEEE Fifth International Conference on Cloud Computing, 2012, pp. 360–367.

[4] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "Cloudscale: Elastic resource scaling for multi-tenant cloud systems," in Proceedings of the 2Nd ACM Symposium on Cloud Computing, ser. SOCC '11. New York, NY, USA: ACM, 2011, pp. 5:1–5:14.

[5] Q. Zhang, L. Cherkasova, and E. Smirni, "A regression-based analytic model for dynamic resource provisioning of multi-tier applications," in Proceedings of the Fourth International Conference on Autonomic Computing, 2007, pp. 27–.

[6] P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem, "Adaptive control of virtualized resources in utility computing environments," ACM SIGOPS Operating Systems Review, vol. 41, no. 3, pp. 289–302, 2007.

[7] G. Juve and E. Deelman, "Resource Provisioning Options for Large-Scale Scientific Workflows," in 2008 IEEE Fourth International Conference on eScience. IEEE, Dec. 2008, pp. 608–613.

[8] T. Sandholm, J. A. Ortiz, J. Odeberg, and K. Lai, "Market-Based Resource Allocation using Price Prediction in a High Performance Computing Grid for Scientific Applications," in 2006 15th IEEE International Conference on High Performance Distributed Computing. IEEE, 2006, pp. 132–143.

[9] S. Genaud and J. Gossa, "Cost-Wait Trade-Offs in Client-Side Resource Provisioning with Elastic Clouds," in IEEE 4th International Conference on Cloud Computing, Jul. 2011, pp. 1–8.

[10] C. Vecchiola, S. Pandey, and R. Buyya, "High-Performance Cloud Computing: A View of Scientific Applications," in 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks. IEEE, Dec. 2009, pp. 4–16.

[11] T. A. Henzinger, A. V. Singh, V. Singh, T. Wies, and D. Zufferey, "FlexPRICE: Flexible Provisioning of Resources in a Cloud Environment," in 2010 IEEE 3rd International Conference on Cloud Computing. IEEE, Jul. 2010, pp. 83–90.

[12] Q. Zhu and G. Agrawal, "Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments," IEEE Transactions on Services Computing, 2012.

[13] W. Chen and E. Deelman, "Integration of Workflow Partitioning and Resource Provisioning," in 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012, pp. 764–768.

[14] M. K. Hedayat, W. Cai, S. J. Turner, and S. Shahand, "Distributed Execution of Workflow Using Parallel Partitioning," in 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications. IEEE, 2009, pp. 106–112.

[15] S. Agarwal, S. Kandula, N. Bruno, M.-C. Wu, I. Stoica, and J. Zhou, "Re-optimizing data-parallel computing," in Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, Berkeley, CA, USA, 2012, pp. 21–21.

[16] Q. Ke, V. Prabhakaran, Y. Xie, Y. Yu, J. Wu, and J. Yang, "Optimizing data partitioning for data-parallel computing," in Proceedings of the 13th USENIX conference on Hot topics in operating systems, ser. HotOS'13. Berkeley, CA, USA: USENIX Association, 2011, p. 13.

[17] T. Bicer, D. Chiu, and G. Agrawal, "Time and Cost Sensitive DataIntensive Computing on Hybrid Clouds," in Cluster, Cloud and Grid Computing, 12th IEEE/ACM International Symposium on, May 2012, pp. 636–643.

[18] A. Wieder, P. Bhatotia, A. Post, and R. Rodrigues, "Orchestrating the deployment of computations in the cloud with conductor," in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012, p. 27.