Autonomic Placement of Heterogeneous Workloads

Ms. S. Umamaheswari

Abstract— An enterprise data centers consolidate the workloads on the same physical hardware in order to reduce the cost of infrastructure and electrical energy. These workloads comprise both transactional and long-running analytic computations. Such consolidation brings new performance management challenges due to the intrinsically different nature of a heterogeneous set of mixed workloads, ranging from scientific simulations to multitier transactional applications. The fact that such different workloads have different natures imposes the need for new scheduling mechanisms to manage collocated heterogeneous sets of applications, such as running a web application and a batch job on the same physical server, with differentiated performance goals. Introducing a technique that enables existing middleware to fairly manage mixed workload type on same physical hardware, and leverages virtualization control machanisms to perform online system reconfiguration.

Index Terms— **Performance management, workload management, resource management, cloud computing**

1 INTRODUCTION

Transactional application and batch jobs are widely used by many organizations to deliver services to their customers and partners. for example, in financial institution, transactional web workloads are used to trade stocks and query indices, while computationally intensive non interactive workloads are used to analyze portfolios or model stock performance. Due to intrinsic difference between these workloads, they are typically run on separate dedicated hardware, which contribute to resource underutilization method and the management complexity. Therefore, organizations demand solutions that permit such workloads to run together on the same hardware, improving resource utilization while continuing to offer performance guarantees.

Ms. S.Umamaheswari is working as Assistant Professor in SVS Engineering College, Coimbatore-642109, Email: umamect@gmail.com, Mobile no: +91 965710888

Compute clouds are commonly used by many different users that rely on the existing computing infrastructure to deploy their workloads. As the result, heterogeneous workloads run on the same physical nodes and pose extraordinary challenges problem.. First, performance goals for different workloads tend to be of different types. For interactive workloads, goals are typically defined in terms of average or percentile response time or throughput over a short time interval, while goals for non interactive workloads concern the performance(e.g., completion time)of individual jobs.Secound,due to the nature of their goals and short duration of individual request, interactive workloads lend themselves to management at short control cycle, whereas non interactive workloads typically require calculation of a schedule for an extended period of time.

In addition, different types of workload require different control mechanisms for management. Transactional workloads are managed using flow control, load balancing, and application placement. Interactive workloads are needed to scheduling and resource control.Traditionally,these have been addressed separately.Inorder to manage resource allocation to a mix of transactional and batch workloads, the system must be able to make placement decisions at short time intervals, so as to respond to changes in transactional workload intensity. While making decisions, the system must be able to look ahead in the queue of jobs and predict the future performance of all jobs. Both those started now, and those that will be started in the future. It must able to make trade-offs between the various jobs and the transactional workloads, taking into account their goals.

This paper proposes the Relative Performance Functions (RPF) and Application program Controller (APC) are designed to permit trade-offs between different workloads, and relies on common virtualization control mechanisms to manage workloads. The RPFs define application performance relative to those application goals. It can therefore be seen that equalizing the achieved relative performance between two applications result in “fairness” the application will be equally satisfied in term of relative distance from their goals. The original contribution of this paper is a scheme for modeling the performance of, and managing.non interactive long running workloads.

The rest of this paper is organized as follows: Section II describes some related works. In section III describes the System Architecture. In section IV: The proposed framework is introduced. The Experimental result and finally this paper is concluded in Section VI.

2 RELATED WORKS

The explicit management of heterogeneous workloads was previously studied in [1], in which CPU shares are manually allocated to run mixed workloads on a large multiprocessor system. This is a static approach, and does not run workloads within virtual machines (VM).

The authors of [2] focus on a utility-guided scheduling mechanism driven by data management criteria, since this is the main concern for many data-intensive HPC scientific

Applications. Have focus on CPU-bound heterogeneous environments, but have technique could be extended to observe data management criteria by expanding the semantics of RPFs. Despite the similarity between an RPF and a utility function, one difference should be pointed out. While utility functions are typically used to model user satisfaction or business value resulting from a particular level of performance, an RPF is merely a measure of relative performance distance from the goal. The authors of [3] and [4] focus on a utility-based

Systems for making placement decisions and provisioning resources: these works do not address the problem of managing heterogeneous workloads, as they are both focused on transactional workloads only. There is also previous work in the area of managing

Workloads in virtual machines.

The authors of [5] study the overhead of a dynamic allocation scheme that relies on virtualization, covering both CPU-intensive jobs and transactional workloads, but does not consider mixed environments.

Work presented in [6] and [7] focuses on the cost of VM migration, and mitigate it by minimizing migrations over time. In [8], authors propose a joint-VM sizing approach in which multiple VMs are consolidated and provisioned as an aggregate. In [9], authors propose a holistic approach to treat performance, power and cooling of IT infrastructures. Neither of these techniques provides a technology to dynamically adjust allocation based on SLA objectives in the face of resource contention.

The authors of [10] present new scheduling algorithms for the cloud, but their effort is focused only on long running jobs (specifically, simulations), and Immigrations is not used. Work on multitier transactional systems, with special effort on avoiding the damaging effects of workload burstiness. a fuzzy logic controller is implemented to make dynamic resource management decisions. This approach is not application centric it focuses on global throughput and considers only transactional applications The algorithm proposed in [11] allows applications to share physical machines, but does not change the number of instances of an application, and does not minimize placement changes, and considers a single bottleneck resource.

In [12], the authors evaluate a similar problem to that addressed in work (restricted to transactional applications), and use a simulated annealing optimization algorithm. Their strategy

Aims to maximize the overall system utility while focus on first maximizing the performance of the least performing application in the system, which increases fairness and Prevents starvation.

3 SYSTEM ARCHITECTURE

The managed system includes a set of heterogeneous server and job scheduler. Long running jobs are submitted to the system via the job scheduler, which unlike traditional schedulers, does not make job execution and placement decisions, the job scheduler only manages dependencies among jobs and performs resource matchmaking. Once Dependencies are resolved and a set of eligible nodes is determined, jobs are submitted to the application placement controller (APC). APC is the most important component of the system. It provides the decision-making logic that affects placement of both web and non interactive workloads. Its placement optimizer calculates the placement that maximizes the minimum Satisfaction across all applications.

APC relies on the knowledge of resource consumption by individual requests and jobs. The web workload profiler, introduced to obtains profiles for web requests in the form of the average number of CPU cycles consumed by requests .The job workload profiler obtains profiles for jobs in the form of the number of CPU cycles required to complete the job, the number of threads used by the job, and the maximum CPU speed at which the job may progress.

4 PROPOSED WORK

The proposed system, presents a technique that enables existing middleware to fairly manage mixed workloads long running jobs and transactional application. The technique permits collocation of the workload types on the same physical hardware, and leverages virtualization control mechanisms to perform online system reconfiguration.

Figure.1 shows the proposed RPF and APC framework. A brief explanation about its phases is followed. The scheme is composed of four main phases: The user request analysis, server allocation, The Relative performance functionality and Application program controller.

RPF AND APC Framework

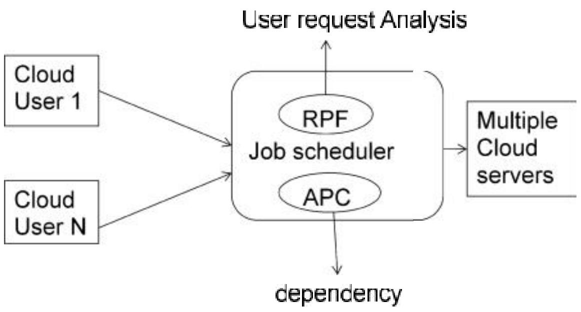

Figure. 1

Phase 1 – User Request Analysis

The users input requests are analyzed by the scheduler before the task is given to the servers. This module helps to avoid the task overloading by analyzing the nature of the users request

Phase 2 – Relative performance functionality

Generating RPFs for the long running jobs and is the Main contribution of this work. RPF allows us to better deal with heterogeneous workloads which may differ in their sensitivity to a particular resource allocation. leverage The flow controller, which comes up with RPF for each web application this RPF gives a measure of application satisfaction with particular allocation of CPU power given its current workload intensity and performance goal. The flow control technique implemented by the flow controller and request router. Each job has an associated performance goal, and when a job completes exactly on schedule, the value of the RPF is zero. Otherwise, the value increase or decreases linearly depending on the distance of completion time from the goal. Such completion time goals for long running jobs need to be mapped into CPU demand, and vice versa.

The required mapping is very hard to obtain for non interactive workloads, because the performance of a given job is not independent of CPU allocation to other jobs. After all, when not all jobs can simultaneously run in the system, the completion time of a job that is waiting in the queue for other job to complete before it may be started depends on how quickly the jobs that where started ahead of it complete.Hence,a job performance depends on the CPU allocation to other job. In the system, have implemented heuristics that allow us to estimate CPU requirements for long running jobs for a given value of RPF

Phase 3 – Application program controller

APC is the most important component of the system. It provides the decision making logic that affects placement of both web and non interactive workloads. Its placement optimizer calculates the placement that maximizes the minimum satisfaction across all application.

Phase 4 – Server Allocation

Server allocation task will take place. The server load value is identified for job allocation. To reduce the over load, the different load values are assigned to the server according to the type of the processing file. To moderately manage mixed workloads, the job scheduling algorithm is followed. In this scheduling, depends upon the nature of the request the load values are assigned dynamically.

6 CONCLUSION

The system introduces several novel features.first,it allows heterogeneous workloads to be collocated on any server machine, thus reducing the granularity of resource allocation.second,approach uses high level performance goals to drive resource allocation.Third,technique exploits a system with homogeneous, particularly non interactive, and interactive workloads by allowing more effective scheduling of jobs.

REFERENCES

[1] Sun Microsystems, "Behavior of Mixed Workloads Consolidated Using Solaris Resource Manager Software," technicalreport,http://www.sun.com/software/resourcemgr/wp-mixed/wp-mixed.pdf, May 2005.

[2] D.M. David Vengerov, L. Mastroleon, and N. Bambos, "Adaptive"Data-Aware Utility-Based Scheduling in Resource-Constrained Systems," Technical Report TR-2007-164, Sun Microsystems, Apr.2007.

[3] C.B. Lee and A.E. Snavely, "Precise and Realistic Utility Functions for User-Centric Performance Analysis of Schedulers," HPDC '07:Proc. 16th Int'l Symp. High Performance Distributed Computing,pp. 107-116, 2007.

[4] B.N. Chun and D.E. Culler, "User-Centric Performance Analysis of Market-Based Cluster Batch Schedulers," Proc. IEEE/ACM Second Int'l Symp. Cluster Computing and the Grid, p. 30, 2002.

[5] R. Urgaonkar, U. Kozat, K. Igarashi, and M. Neely, "Dynamic Resource Allocation and Power Management in Virtualized

Data Centers," Proc. IEEE Network Operations and Management Symp.(NOMS), pp. 479-486, Apr. 2010.

[6] J. Hanson, I. Whalley, M. Steinder, and J. Kephart "Multi-Aspect Hardware Management in Enterprise Server Consolidation," Proc.IEEE Network Operations and Management Symp. (NOMS), pp. 543-550, Apr. 2010.

[7] Z. Wang, X. Zhu, P. Padala, and S. Singhal, "Capacity and Performance Overhead in Dynamic Resource Allocation to Virtual Containers," Proc. IFIP/IEEE 10th Int'l Symp. Integrated Network Management (IM '07), pp. 149-158, May 2007.

[8] P. Bodı´k, R. Griffith, C. Sutton, A. Fox, M.I. Jordan, and D.A.Patterson, "Statistical Machine Learning Makes Automatic Control Practical for Internet Datacenters," Proc. Conf. Hot Topics in Cloud Computing (HotCloud '09), 2009.

[9] M. Cardosa, M.R. Korupolu, and A. Singh, "Shares and Utilities Based Power Consolidation in Virtualized Server Environments,"Proc. IFIP/IEEE Int'l Symp. Integrated Network Management (IM '09),pp. 327-334, 2009.

[10] J.-G. Park, J.-M. Kim, H. Choi, and Y.-C. Woo, "Virtual Machine Migration in Self-Managing Virtualized Server Environments,"Proc. 11th Int'l Conf. Advanced Comm. Technology (ICACT), pp. 2077-2083, 2009.

[11] E. Arzuaga and D.R. Kaeli, "Quantifying Load Imbalance on Virtualized Enterprise Servers," Proc. First Joint WOSP/SIPEW Int'l Conf. Performance Eng. (WOSP/SIPEW '10), pp. 235-242, 2010.

[12] S. Meng, L. Liu, and V. Soundararajan, "Tide: Achieving Self-Scaling in Virtualized Datacenter Management Middleware,"Proc. 11th Int'l Middleware Conf. Industrial Track (Middleware Industrial Track '10), pp. 17-22, 2010.

[13] Y. Chen, D. Gmach, C. Hyser, Z. Wang, C. Bash, C. Hoover, and S.Singhal, "Integrated Management of Application Performance,Power and Cooling in Data Centers," Proc. IEEE Network Operations and Management Symp. (NOMS), pp. 615-622, Apr. 2010.

[14] Z. Zhang, L.T.X. Phan, G. Tan, S. Jain, H. Duong, B.T. Loo, and I.Lee, "On the Feasibility of Dynamic Rescheduling on the IntelDistributed Computing Platform," Proc. 11th Int'l Middleware Conf. Industrial Track (Middleware Industrial Track '10), pp. 4-10, 2010.

[15] A. Caniff, L. Lu, N. Mi, L. Cherkasova, and E. Smirni, "Efficient Resource Allocation and Power Saving in Multi-Tiered Systems,"Proc. 19th Int'l Conf. World Wide Web (WWW '10), pp. 1069-1070,2010.

[16] H.C. Lim, S. Babu, J.S. Chase, and S.S. Parekh, "Automated Control in Cloud Computing: Challenges and Opportunities,"Proc. First Workshop Automated Control for Datacenters and Clouds (ACDC '09), pp. 13-18, 2009.

[17] A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M.Sviridenko, and A. Tantawi, "Dynamic Placement for Clustered Web Applications," Proc. 15th Int'l Conf. World Wide Web (WWW '06), May 2006.and Memory Constraints," Proc. Int'l Workshop Efficient and Experimental Algorithms,May 2005.

[18] T. Kimbrel, M. Steinder, M. Sviridenko, and A. Tantawi,"Dynamic Application Placement under Service and Memory Constraints," Proc. Int'l Workshop Efficient and Experimental Algorithms,May 2005.

[19] X. Wang, D. Lan, G. Wang, X. Fang, M. Ye, Y. Chen, and Q. Wang,"Appliance-Based Autonomic Provisioning Framework for Virtualized Outsourcing Data Center," Proc. Fourth Int'l Conf.Autonomic Computing (ICAC '07), p. 29, 2007.

[20] D. Carrera, M. Steinder, I. Whalley, J. Torres, and E. Ayguade´,"Utility-Based Placement of Dynamic Web Applications with Fairness Goals," Proc. IEEE Network Operations and Management Symp. (NOMS), 2008.