International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud

Rajashekhar[1], Joythi Patil[2]

[1]M.Tech, Department of Computer Science &Engineering,
Poojya Doddappa Appa College of Engineering & Technology, Gulbarga, Karnataka, India.

[2]Associate Professor, Department of Computer Science & Engineering
Poojya Doddappa Appa College of Engineering & Technology, Gulbarga, Karnataka, India.

Abstract: ***Data centers consume tremendous amounts of energy in terms of power distribution and cooling. Dynamic capacity provisioning is a promising approach for reducing energy consumption by dynamically adjusting the number of active machines to match resource demands. However, despite extensive studies of the problem, existing solutions have not fully considered the heterogeneity of both workload and machine hardware found in production environments. In particular, production data centers often comprise heterogeneous machines with different capacities and energy consumption characteristics. Meanwhile, the production cloud workloads typically consist of diverse applications with different priorities, performance and resource requirements. Failure to consider the heterogeneity of both machines and workloads will lead to both sub-optimal energy-savings and long scheduling delays, due to incompatibility between workload requirements and the resources offered by the provisioned machines. To address this limitation, the Harmony, a Heterogeneity-Aware dynamic capacity provisioning scheme for cloud data centers is introduced. Specifically, use the K-means clustering algorithm to divide workload into distinct task classes with similar characteristics in terms of resource and performance requirements. A technique that dynamically adjusting the number of machines to minimize total energy consumption and scheduling delay. Simulations using traces from a Google's compute cluster demonstrate Harmony (Heterogeneity-Aware Resource Monitoring and management system) can reduce energy by 28 percent compared to heterogeneity-oblivious solutions.***

Keywords: Cloud computing, workload characterization, energy management.

1. Introduction

Data centers have recently gained significant popularity as a cost-effective platform for hosting large-scale service applications. While large data centers enjoy economies of scale by amortizing long-term capital investments over a large number of machines, they also incur tremendous energy costs in terms of power distribution and cooling. For instance, it has been reported that energy-related costs account for approximately 12 percent of overall data center expenditures. For large companies like Google, a 3 percent reduction in energy cost can translate to over a million dollars in cost savings. At the same time, governmental agencies continue to implement and regulations to promote energy-efficient computing. As a result, reducing energy consumption has become a primary concern for today's data center operators. In recent years, there has been extensive research on improving data center energy efficiency. One promising technique that has received significant attention is dynamic capacity provisioning (DCP). The goal of this technique is to dynamically adjust the number of active machines in a data center in order to reduce energy consumption while meeting the service level objectives (SLOs) of workloads. In the context of workload scheduling in data centers, a metric of particular importance is scheduling delay, which is the time a request waits in the scheduling queue before it is scheduled on a machine. Task scheduling delay is a primary concern in data center environments for several reasons:(1)A user may need to immediately scale up an application to accommodate a surge in demand and hence requires the resource request to be satisfied as (2) Even for lower-priority requests (e.g., background applications), long scheduling delay can to starvation, which can significantly hurt the performance of these applications. In practice, however, there is often a tradeoff between energy savings and scheduling delay. Even though turning off a large number of machines can achieve high energy savings, at the same time, it reduces service capacity and hence leads to high scheduling delay. Finally, the heterogeneity-aware DCP scheme should also take into account the reconfiguration costs associated with switching on and off individual machines. This is because frequently turning on and off a machine can cause the "wear-and-tear" effect that reduces the machine lifetime. Despite the fact that a large number of DCP schemes have been proposed in the literature in recent years, a key challenge that often has been overlooked or considered difficult to address is heterogeneity, which is prevalent in production cloud data centers.

The general architecture of a cloud computing system is shown in Figure 1.1. Specifically, the physical resources in each data center are organized in racks of physical machines. The racks are connected through data center networks, which offer high bandwidth, low latency connections between physical machines. In order to reduce capital investments, cloud providers often build their data centers with large quantities of commodity machines and switches, as opposed to expensive high-end equipment. However, as commodity equipment may become outdated over-time, it is necessary to upgrade data centers with new equipment once a few years. As a consequence, modern cloud data centers often consist of multiple generations of physical machines and switches with heterogeneous processing, storage and networking capacities.

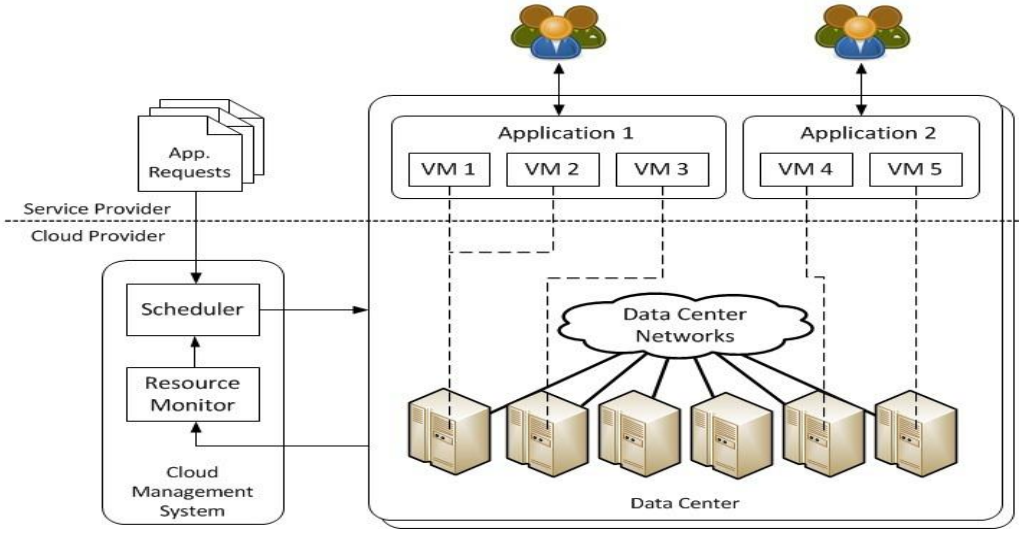

Figure 1.1: Cloud Computing System Architecture

2. Organization

This paper is organized as follows, section 1 discusses the introduction, and section 3 describes related work. Section 4 details the system design and implementation. Section 5, presents the performance evaluations of our system design. Finally, section 6 presents some concluding remark.

3. Related Work

"Effective Straggler Mitigation: Attack of the Clones," G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica[1], has introduced the clusters at Facebook and Microsoft Bing [1], even after applying state-of-the-art straggler mitigation techniques, these latency sensitive jobs have stragglers that are on average 8 times slower than the median task in that job [1].The main challenge of cloning is, however, that extra clones can cause contention for intermediate data. Use a technique, delay assignment, which efficiently avoids such contention. Evaluation of our system, Dolly, using production workloads shows that the small jobs speedup by 34% to 46% after state-of-the-art mitigation techniques have been applied, using just 5% extra resources for cloning.[1]**"Managing Server Energy and Operational Costs in Hosting Centers," Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam [2],** has proposed the thousands of dense servers within a relatively small real-estate in order to host the applications/services of different customers who may have been assured by a service-level agreement (SLA) [2]. That presents formalism to this problem, and proposes three new online solution strategies based on steady state queuing analysis, feedback control theory, and a hybrid mechanism borrowing ideas from these two[2]. The currently prototyping of this framework is on a server cluster [2]. Also refining our solution strategies further, and evaluating them with a wider-spectrum of workloads. **"Analysis and Lessons from a Publicly Available Google Cluster Trace," Y. Chen et al. [3],** has analyzed a large scale production workload trace recently made publicly available by Google. Offer a statistical profile of the data, with several interesting discoveries regarding job arrival patterns, CPU and memory consumptions, task durations, and others. Further performing k-means clustering to identify common groups of jobs, with several methodological departures and different findings compared with prior work on similar data [3].To facilitate such a repository while addressing trade secret and user privacy concerns, future work should develop a toolkit with anonymizers, data format converters, and standard algorithms for detailed statistical analysis. **"Dominant Resource Fairness: Fair Allocation of Multiple Resource Types," A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica [4],**has implemented DRF in the Mesos cluster resource manager, and show that it leads to better throughput and fairness than the slot-based fair sharing schemes in current cluster schedulers [4].They have evaluated DRF by implementing it in the Mesos resource manager, and shown that it can lead to better overall performance than the slot-based fair schedulers that are commonly in use today [4].**"Validating Heuristics for Virtual Machines Consolidation," S. Lee, R. Panigrahy, V. Prabhakaran, V. Ramasubrahmanian, K. Talwar, L. Uyeda, and U. Wieder**. The author examines two fundamental issues pertaining to virtual machines (VM) consolidation. Current virtualization management tools, both commercial and academic, enable multiple virtual machines to be consolidated into few servers so that other servers can be turned off, saving power. Yet, more sophisticated, dimension-aware heuristics provide additional benefits, especially for mixed workloads with negatively correlated dependence on resources [5].**"Dynamic Right-Sizing for Power-Proportional Data Centers," M. Lin, A. Wierman, L. Andrew, and E. Thereska [6],** has proposed the power consumption imposes a significant cost for data centers implementing cloud services, yet much of that power is used to maintain excess service capacity during periods of predictably low load. Thus, even if a data center is currently performing valley filling, it can still achieve significant cost savings via dynamic right-sizing [6]. **"Towards Characterizing Cloud Backend Workloads: Insights from Google Compute Clusters," A.K. Mishra, J.L. Hellerstein, W. Cirne, and C.R. Das [7],** has proposed the advent of cloud computing promises highly available, efficient, and flexible computing services for applications such as web search, email, voice over IP, and web search alerts.Also address characterization of the task arrival process, and extend our task classification to consider job

constraints (e.g., co-locating two tasks in the same machine) [7].**"Cost Model for Planning, Development and Operation of a Data Center," C.D. Patel and A.J. Shah [8],** has described the emergence of the compute utility and growth in data center based computer services necessitates an examination of costs associated with housing and powering the computer, networking and storage equipment. **"Cutting the Electric Bill for Internet-Scale Systems," A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs [9],** has proposed energy expenses are becoming an increasingly important fraction of data center operating costs. At the same time, the energy expense per unit of computation can vary significantly between two different locations [9]. **"Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," C. Reiss, A. Tumanov, G. Ganger, R. Katz, and M. Kozuch [10],** has described the, Cloud Computing, 2012. To better understand the challenges in developing effective cloud based resource schedulers; first analyze the publicly available trace data from a sizable multi-purpose cluster. The most notable workload characteristic is heterogeneity: in resource types (e.g., cores: RAM per machine) and their usage (e.g., duration and resources needed) [10].

4. Methodology

The main objective of proposed work is, to reduce the energy consumption in terms of power distribution and cooling by dynamically adjusting the number of active machines to match resource demands. Present system does not considered the heterogeneity of both workload and machine hardware found in production environments. To address this limitation, the Harmony, a Heterogeneity-Aware dynamic capacity provisioning scheme for cloud data centers is introduced. Specifically, use the K-means clustering algorithm to divide workload into distinct task classes with similar characteristics in terms of resource and performance requirements

4.1 Proposed System

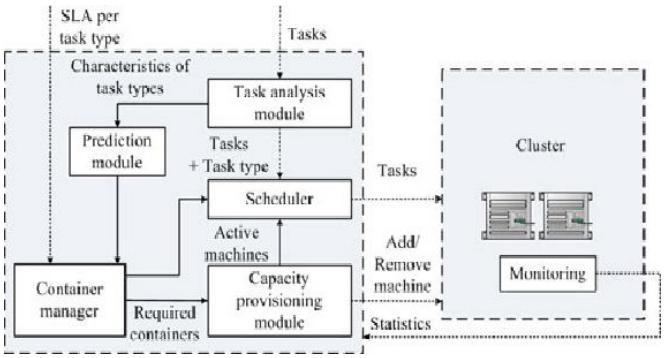

Figure 4.1: System Architecture of proposed system.

4.2 Software Requirement Specification

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Nonfunctional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, qualitystandards, or design constraints).

System requirements specification. A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms what must be delivered or accomplished to provide value.
- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization.

The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

4.3 Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economic feasibility for certain.

4.4 Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. The management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

4.5 Technical Feasibility

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

4.6 K-means Algorithm

K-Means algorithm (Pang-Ning et al., 2006) follows the partitioned or nonhierarchical clustering approach (Jain and Dubes, 1988). It involves partitioning the given data set into specific number groups called Clusters (Klosgen and Zytkow, 1996). Each cluster is associated with a enter point called centroid. Each point is assigned to a cluster with the closest centroid. The main drawback of K-Means is the number of clusters must be known in advance, which is defined by K.
Step1:Select K points as the initial centroids.
Step2: Repeat.
Step3: Form K Clusters by assigning all points to closest centroid.
Step4: Recompute the centroid of each cluster.
Step5: Until the centroids don't change.
The Initial centroids will be chosen randomly. The centroid is nothing but the mean of the points in the cluster. Euclidean distance is used to measure the closeness. K-Means generates different clusters in different runs (Murat et al., 2011).

4.7 Flow Diagrams

Use case diagram
A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

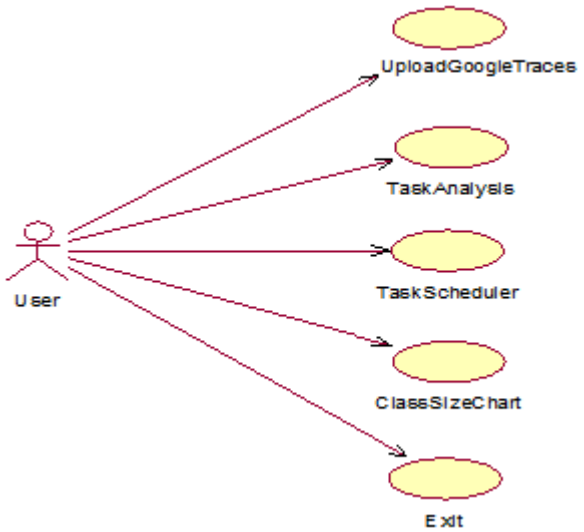

Figure 4.2: Use case diagram

Component Diagram
In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systemsComponents are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components

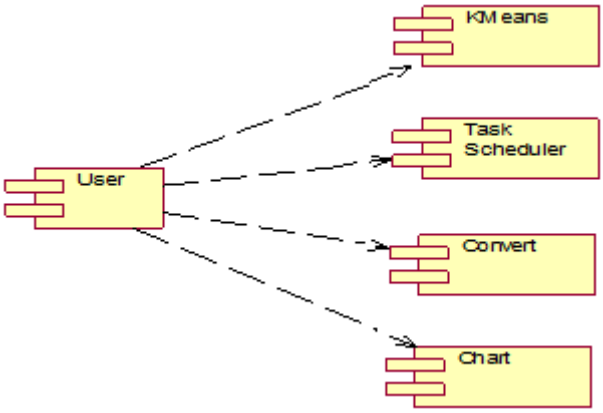

Figure 4.3: Component Diagram.

5. Results

For each priority group, we varied the value of k and evaluatedthe quality of the resulting clusters produced by the K-meansalgorithm. The best value of k for each priority group is selected as the one for which no significant benefit can be achieved by increasing the value of k. The results after the first step of our characterization for each priority group are shown in Figures respectively. These diagrams show the clustering algorithm captures the differences in task sizes and identifies cup-intensive tasks and memory-intensive tasks. Furthermore, the standard deviation is much less than the mean value for both CPU and memory, which confirms the accuracy of the characterization. In fig 5.1, showsHome Page of Dynamic Heterogeneity Aware Resource Provisioning in the Cloud showing four modules

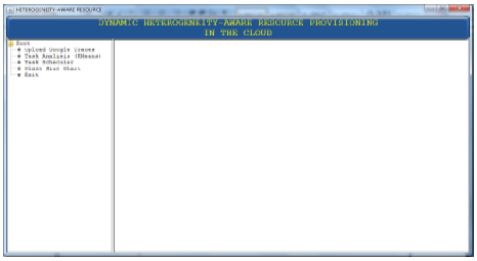

Figure 5.1: Home Page of Dynamic Heterogeneity

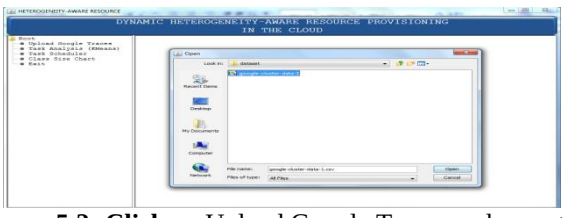

Figure 5.2: Click on Upload Google Traces and open the file (google-cluster-data-1) existed in the "dataset" folder

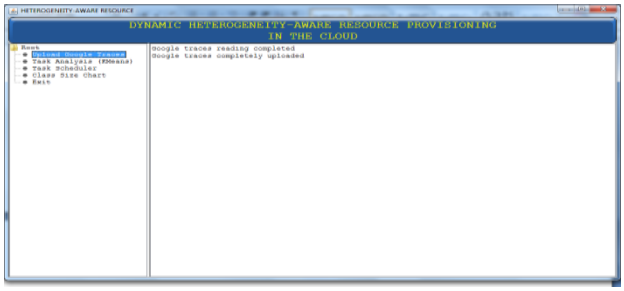

Figure 5.3: After uploading the dataset (google-cluster-data-1) existed in the "dataset" folder

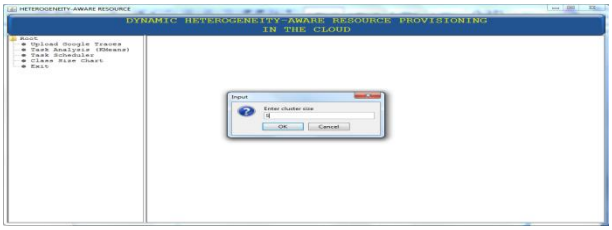

Figure 5.4: Click on"Task Analysis" and enter the number of clusters you want to be formed from the dataset in cluster size field:

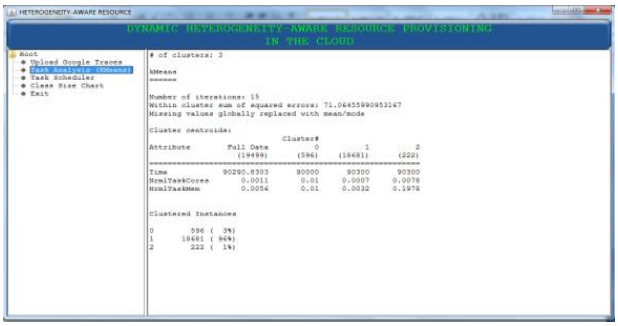

Figure 5.5: After click on OK button it will display the cluster details as shown in above result.

Before performing the "Task Scheduler" Operation, give the three attribute values from the file "google-cluster-data-1" as an input to the "test.arff". Those attributes are Time, NrmlTaskCores, and NrmlTaskMem.

Open "test.arff" file among the project files using any editor. By using editplus. Then the content of the test.arff will be displayed like the following. Select any row of values of three attributes (Time, NrmlTaskCores, and NrmlTaskMem) from "google-cluster-data-1". For example here we are copying the 50th row values (90000, 0.015625, and 0.0016384). These three copied values need to be pasted by replacing the values in the last row of the "test.arff" :

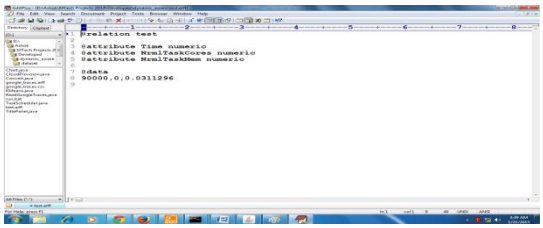

Figure 5.6: Before replacing three attribute values (Time, NrmlTaskCores, and NrmlTaskMem.)

After updating three attribute values (Time, NrmlTaskCores, and NrmlTaskMem.) click on save button to save the updated valves

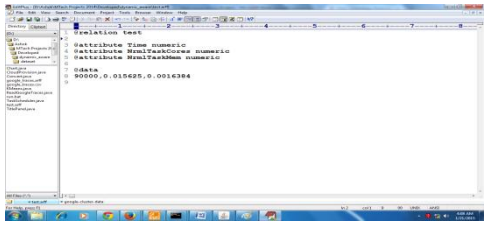

Figure 5.7: After replacing three attribute values (Time, NrmlTaskCores, and NrmlTaskMem.)

Updating the values in the test.arff nothing but you are giving the input request.
Click on "Task Scheduler"
The following Screen Depicts that input request given by the user is belong to Cluster 0.

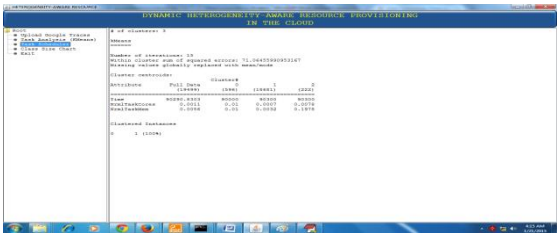

Figure 5.8: Input requests given by the user is belonging to Cluster 0.

Click on the "Class Size Chart":
It will display the number of records each cluster contains in the form of graph as follows

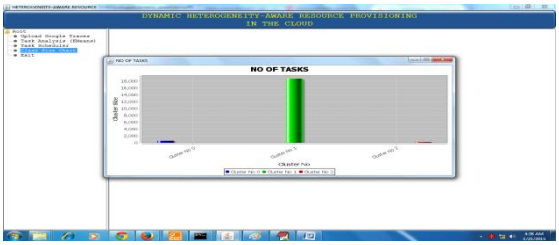

Figure 5.9: Showing graph between cluster numbers versus cluster size

6. Conclusion and Future Work

Dynamic capacity provisioning has become a promising solution for reducing energy consumption in data centers in recent years. However, existing work on this topic has not addressed a key challenge, which is the heterogeneity of workloads and physical machines. In this paper, we first provide a characterization of both workload and machine heterogeneity found in one of Google's production computes clusters. Then we present Harmony, a heterogeneity-aware framework that dynamically adjusts the number of machines to strike a balance between energy savings and scheduling delay, while considering the reconfiguration cost. Through experiments using Google workload traces, we found Harmony yields large energy savings while significantly improving task scheduling delay.

References

[1] G. Ananthanarayanan, A. Ghodsi, S. Shenker, and I. Stoica, "Effective Straggler Mitigation: Attack of the Clones," Proc. 10th USENIX Conf. Networked Systems Design and Implementation (NSDI), 2013.
[2] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing Server Energy and Operational Costs in Hosting Centers," ACM SIGMETRICS Performance Evaluation Rev., vol. 33, pp. 303-314, 2005.
[3] Y. Chen et al., "Analysis and Lessons from a Publicly Available Google Cluster Trace," Technical Report UCB/EECS-2010-95, 2010.
[4] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant Resource Fairness: Fair Allocation of Multiple Resource Types," Proc. Eighth USENIX Conf. Networked Systems Design and Implementation (USENIX NSDI), 2011
[5] S. Lee, R. Panigrahy, V. Prabhakaran, V. Ramasubrahmanian, K. Talwar, L. Uyeda, and U. Wieder, "Validating Heuristics for Virtual Machines Consolidation," Microsoft Research, MSR-TR-2011-9, 2011.
[6] M. Lin, A. Wierman, L. Andrew, and E. Thereska, "Dynamic Right-Sizing for Power-Proportional Data Centers," Proc. IEEE INFOCOM, 2011.
[7] A.K. Mishra, J.L. Hellerstein, W. Cirne, and C.R. Das, "Towards Characterizing Cloud Backend Workloads: Insights from Google Compute Clusters," ACM SIGMETRICS Performance Evaluation Rev., vol. 37, pp. 34-41, Mar. 2010.
[8] C.D. Patel and A.J. Shah1, "Cost Model for Planning, Development and Operation of a Data Center," Technical Report HPL- 2005-107(R.1), HP Laboratories Palo Alto, 2005
[9] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the Electric Bill for Internet-Scale Systems," Proc. ACM SIGCOMM, 2009.
[10] C. Reiss, A. Tumanov, G. Ganger, R. Katz, and M. Kozuch, "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis," Proc. ACM Symp. Cloud Computing, 2012.