International Journal of Computer Engineering and Applications, Special Edition
www.ijcea.com ISSN 2321-3469

Special Issue

Characteristics Based Scale-Out vs. Scale-Up for Green Cloud Computing

Ranjan Kumar [1], G. Sahoo [2]

[1]*Department of Computer Science & Engineering, Cambridge Institute of Technology, Ranchi, India*

[2]*Department of Computer Science & Engineering, Birla Institute of Technology, Mesra, Ranchi, India*

ABSTRACT:

The cloud uses major proportion of the energy utilization. Through this paper we are proposing scale-out and scale-in hybrid methodologies and approaches via which it would be possible to attain an energy conservation consumed by the cluster. We are using an integrated mechanism in between the servers and the jobs, consisting of combination of optimization approaches. Beginning from the scale-out/scale-in, scale-up/scale-down optimization in the cloud cluster to load balancing & solving different job arrival patterns. We have designed this approach without degrading the quality of service of the datacenters/clusters.

Keywords: Scale-Out, Scale-In, Scale-Up, Scale-Down, Sway, Cloud Computing,

[1] INTRODUCTION

With increasing demand for Cloud Computing and its services and its applications in other fields to share resources and minimize the cost of hardware for many enterprises the only dominant problem comes out to be energy consumption for these giant clusters and stations. The similar power saving approaches are applicable for datacenters and high performance computing laboratories. In this paper our major emphasis would be on, using approach of Scale-out/scale-in and scale-up/scale-down methodology to find out an optimal resource using a mechanism to serve the jobs following certain distribution patterns. The other auxiliary optimization tools which we would require to the prior optimization and classification of the server's states like idle, over-loaded and under-loaded. For the load balancing among the under-loaded and over-loaded servers.

The Organization of this paper is as follows: Related Work is discussed in section 2. Different workloads are discussed in Section 3. The proposed algorithm is discussed in section 4 and section 5 gives the conclusion of our work.

[2] RELATED WORK

Kai Hwang, Yue Shi and Xiaoying Bai [1] [2] discussed about higher efficiency that promotes the productivity and also says that the converse may not necessarily hold. They also

discussed about QoS that may vary user to user. Different users may set their own satisfaction threshold for the QoS they can accept. Their major contribution lies in the performance metrics and evaluation methodology. They said that pure scale-out or pure scale-up are more cost effective with higher efficiency and productivity. The scale-up and mixed scaling have higher performance in using smaller clusters with a few powerful machine instances. They tested five cloud benchmarks on Amazon IaaS EC2 cloud: namely YCSB, CloudSuite, HiBench, BenchClouds, and TPC-W. Raja Appuswamy, Christos Gkantsidis, Dushyanth Narayanan, Orion Hodson and Antony Rowstron [3] discussed about scale-out/ scale-in and Hadoop MapReduce. They presented a series of transparent optimizations that allow Hadoop to deliver good scale-up performance and evaluated their claims against a diverse set of Hadoop jobs. Their results have implications for the way Hadoop and analytics clusters in general are provisioned, with scale-up servers being a better option for many jobs whether in a private cluster or in the cloud.Michael et al. [6] discussed about the problem of scale-up vs scale-out for an interactive application. They found that scale-out has a better performance per price ratio than scale-up. They also found that running scale-out in a box gives better performance than using multi-threading.

[3] WORKLOAD

(a) Scale-Out: The Scale-Out technique uses homogeneous clusters, it allows adding more machine instances of the same type agreed by the quota based on the service-level agreement (SLA). If there are machine instances from q1 to q2, then q2 must be greater than q1 (q2>q1).

(b) Scale-In: The Scale-In technique also uses homogeneous clusters, it allows subtracting machine instances of the same type based on quota agreed by service-level agreement (SLA). In this case, q1 must be greater than q2 (q1>q2). The scale-out and scale-in technique are often called as horizontal scaling. The below figure 1 shows the scale-out & scale-in technique.

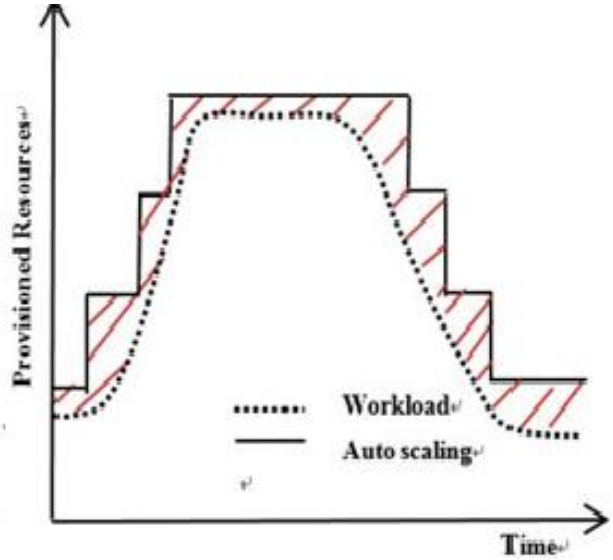

Figure: 1. Scale-Out & Scale-In

(c) Scale-Up: This technique is used in terms of memory, Processor or storage. When scaling the cloud from small to large nodes. If there are instance from v1 to v2. To scale between these instances. v2 must be more powerful than v1.

(d) Scale-Down: When scaling between these instances, v2 is not more powerful than v1. The scale-up and scale-down technique often called as vertical scaling. The below figure 2 shows scale-up & scale-down technique.

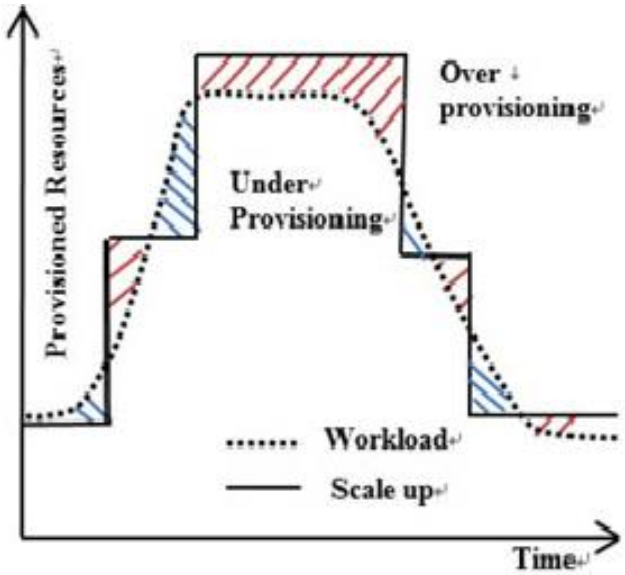

Figure: 2. Scale-Up & Scale-Down

(e) Mixed- Scaling: It allows scale-up or scale-down the instances and also adjust the instance quantity by scale-out or scale-in resources at the same time. It will give better result with heterogeneous cluster. The below figure 3 shows the mixed-scaling technique.

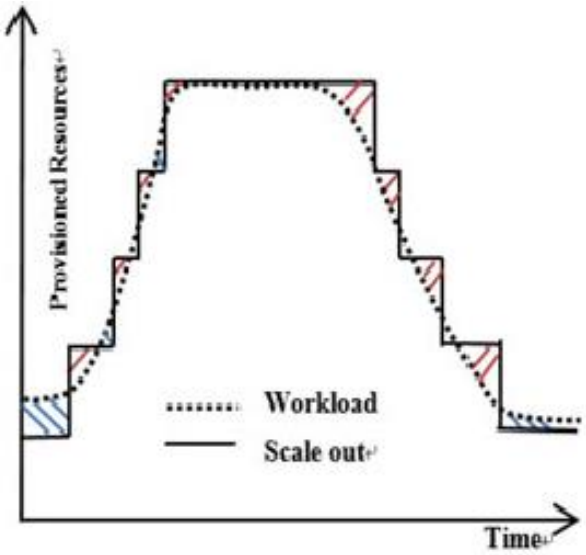

Figure: 3. Mixed Scaling

The various factors that we could be looking forward as parameter to our research are:

(I) Basic Performance Metrics:
Speed (S): Number of millions of operations per second (Mops).
Speedup (Su): Speed gain of using multiple nodes.

Efficiency (Ef): Percentage of peak performance achieved.
Utilization (U): Busy resources (CPU, memory, storage).
Scalability (Sc): Scaling ability to upgrade performance.
Sway (Sy): Scale to differentiate large and small jobs.

(II) Cloud Capabilities:
Latency (L): System response time or access latency.
Bandwidth (B): This is data transfer rate or I/O rate.
Elasticity (EI): The ability for cloud resources to scale up/down or scale in/out to match with workload variation.
Software (Sw): Software portability, API and SDK tooling.
Big Data Analytics (An): The ability to uncover hidden information or predict trends in big data.

(III) Productivity Measures:
Quality of Service (QoS): Satisfaction on user services.
System availability (A): The system up time per year.
Service costs (Co): User renting costs and provider cost.
Power Demand (W): Cloud power consumption (MWatt).
SLA/Security (L): Compliance of SLA, security, etc.
Productivity (P): QoS satisfied performance per unit cost.

[4] PROPOSED ALGORITHM

There are job schedulers which schedules different jobs to different web services. Sway differentiates between large and small jobs.
The different steps of Algorithm are:

Step 1: Randomly select a Job Scheduler
Step 2: Job Scheduler Schedules job to different web services.
    While Job is not schedules to web services
     Repeat steps 2.
Step 3: Sway checks for small and large jobs
Step 4: if
        Job is small
     Then
        Scale-out with small number of nodes
     Else
        Scale-up with large number of nodes
Step 5: Return to the Job Scheduler.
Step 6: After completion kill the job.
Step 7: End

[5] CONCLUSION

In this paper, we have proposed a method for green cloud computing. In which job scheduler schedules jobs to different web services and sway differentiate jobs into larger and

smaller and also decides whether the job go for scale-out , scale-in or hybrid technique. We have emphasis on web servers and sway which will more efficient for green cloud computing.

REFERENCES

[1] Kai Hwang, Yue Shi and Xiaoying Bai "Scale-Out vs. Scale-Up Techniques for Cloud Performance and Productivity" IEEE 6th International Conference on Cloud Computing Technology and Science, Pp- 763-768, 2014.

[2] Kai Hwang, Xiaoying Bai and Yue Shi, "Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 1, Pp. 130-143, Jan 2016.

[3] Raja Appuswamy, Christos Gkantsidis, Dushyanth Narayanan, Orion Hodson and Antony Rowstron, " Scale-Up vs Scale-Out for Hadoop: Time to rethink?" ACM,Santa Clara, Oct, 2013.

[4] Kai Hwang and Z. Xu "Scalable Parallel Computing, McGraw-Hill, San Francisco, 1998.

[5] Iosup, A., Ostermann, S., Yigitbasi, M., Prodan, R., Fahringer, T., and Epema, D. "Performance Analysis of Cloud Computing Services for Many- Tasks Scientific Computing", IEEE Transaction on Parallel and Distributed System, 2011.

[6] Michael, M., et al, "Scale-up x Scale-out : A Case Study using Nutch/Luene", IEEE Int'l Parallel and Distriduted Processing Symp. (IPDPS), March 26, 2007.

[7] Y. Chen, S. Alspaugh, and R. H. Katz. "Interactive Analytical Processing in Big Data Systems: A Cross-Industry Study of MapReduce Workloads", PVLDB 5.12, Pp. 1802-1813, 2012.

[8] R. M. Yoo, A. Romano, and C. Kozyrakis. "Phoenix Rebirth: Scalable MapReduce on a Large-Scale Shared-Memory System", IEEE International Symposium on Workload Characterization (IISWC). 2009.

[9] X. Bai, Y. Wang, G. Dai, W. T. Tsai, and Y. Chen, " A framework for collaborative verification and validation of web services", Component-Based Software Engineering, Springer, 2007.

[10] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites", in Proc. IEEE Int. Symp Model Anal. Simulation Comput. Telecom Syst. Pp. 390-399, 2011.

[11] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing", in Proceedings of the International Conference on Cloud Computing, Springer, 2010.