ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Cost-effective Resource Reservation Scheduling for Map-Reduce in the Cloud

Vasumathi A[1], Rupa Ezhil Arasi P[2], Dhanisha R[3], Megiba Jasmine R[4]

PG Student, Dept. of C.S.E., Muthayammal Engineering College, Rasipuram, Tamilnadu, India

Professor, Dept. of C.S.E., Muthayammal Engineering College, Rasipuram, Tamilnadu, India

PG Student, Dept. of C.S.E., Ponjesly college of Engineering, Nagerkoil, Tamilnadu, India

Assistant Professor, Dept. of C.S.E., Ponjesly college of Engineering, Nagerkoil, Tamilnadu, India

ABSTRACT: Map-Reduce Framework have evolved to provisioning increasing amount of time-sensitive and interactive data analysis. The workloads runs on large clusters, whose size and cost gives best energy efficiency, however large cluster size leads to higher cost. In this paper, we present an efficient Map-Reduce model for providing cost effective Map-Reduce services in a cloud. Compared with the existing Map-Reduce cloud services our model provides various benefits. This model is designed to provide a cost-efficient solution to handle production workloads. Existing services needs user to select the VM resources to be used for the jobs, our model automate the creation of optimal cluster configuration for user jobs by consulting profile and analyse service. To effectively creates the cluster configuration, scheduler makes the future reservation of clusters in VM pool. The existing models allows only a per-job and per-customer resource allocation for the jobs, this model provides an efficient globally resource allocation method that reduces the resource usage cost in the cloud. This model achieves significantly lower resource usage costs for the jobs. Resource management scheme contains cost-aware resource provisioning, Virtual Machine Reservation scheduling and virtual machine pool reconfiguration.

KEYWORDS: Energy Map-Reduce, Cost effectiveness, Scheduling, Reservation Technique, Resource Management

I. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand access and its pay-as-you-go cost structure have enabled hardware infrastructure service providers, platform service providers as well as software and application service providers to offer computing services on demand .This technology, combined with Big Data and Big Data analytics leverages the rapid evolution of datacenter technologies having more cost-effective, consumer-driven solutions. Map Reduce model is most popular method for big data analytics [7] and its open-source implementation called Hadoop [5][6]. Map-Reduce model allows enterprises to analyze their data without dealing with the complexity of building and man-aging large installations of Map-Reduce platforms. By using virtual machines (VMs) and storage hosted by the cloud, enterprises can simply create virtual Map-Reduce clusters to analyze their data.

In this paper, we discuss the cost-inefficiencies of the existing cloud services for Map-Reduce and propose a cost effective resource management framework that is globally optimized resource allocation to minimize the infrastructure cost in the cloud datacenter. We note that the existing cloud solutions for Map-Reduce work based on a per-job or per-customer optimization approach where the optimization and resource sharing opportunities are restricted within a single job or a single customer. In existing, dedicated Map-Reduce cloud services such as Amazon Elastic Map-Reduce [4], customers buy on-demand clusters of VMs for each job or a workflow and once the Map-Reduce job (or workflow) is submitted, the cloud provider creates VMs that execute that job and after job completion the VMs are de-provisioned. Here the resource optimization opportunity is restricted to the per-job level. Alternately, one can lease dedicated cluster resources from a generic cloud service like Amazon Elastic Compute Cloud [5] and operate Map-Reduce on them as if they were using a private Map-Reduce infrastructure. While this approach enables resource optimization at the per-customer level, we argue that in such an approach, the size of the leased dedicated clusters needs to be chosen based on the peak workload requirements of each customer and hence, the leased clusters are under-utilized for a large fraction of the time leading to higher costs.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

II. RELATED WORK

In [1] The global optimization of resources in the cloud brings the cost effective resource management to cloud provider and avoids under-utilization of virtual machine pool. [2] Map-Reduce cluster reveals that outliers takes prolong job completion include run-time contention for processor, memory and other resources, disk failures, varying bandwidth and congestion along network paths and, imbalance in task workload. This strategies overcome the prolong job completion, restarting outliers, network-aware placement of tasks leads to free up resources that can be used by subsequent.[3] This strategy demonstrates that performance evaluations using realistic workloads gives cluster operator new ways to identify workload-specific resource bottlenecks, and workload-specific choice of Map-Reduce task schedulers. [4] This application enables one can lease dedicated cluster resources from generic cloud such as Amazon Elastic Compute Cloud and operate Map-Reduce on them. [5] This application provides customers buy on-demand clusters of VMs from dedicated Map-Reduce cloud services such as Amazon Elastic Map-Reduce once job is submitted, the cloud provider creates VMs that execute that job and after job completion the VMs are de-provisioned. [6] Hadoop is an open source implementation of Map-Reduce which enables cluster configuration such as memory, VM type. [7] This strategy analysis data without dealing with the complexity of building and maintaining large installations of Map-Reduce platforms. [8] SELinux enforcing SELinux access policies in a Map-Reduce which does not lead to performance overhead. [9] This model improves Map-Reduce provisioning by analyzing and comparing resource consumption of the application at hand with a database of similar resource consumption signatures of other applications. [10] propose techniques for combining on demand provisioning of virtual resources with batch processing to increase system utilization based on either per-job or per-customer optimization.

III. PROPOSED MODEL

The profile and analyze service is used when a users job first goes from development-and-testing into production in its software life cycle for once. For forthcoming instances of the job, our model directly sends the job for scheduling. From an architectural view Fig 1, users may also choose to skip profiling and instead provide VM type, cluster size and job parameters to the cloud service similar to existing dedicated Map-Reduce cloud service models like [5]. Jobs that skip the one-time profiling and analyzing step will still benefits from the response time. However, they will fail to leverage the benefits provided by global resource optimization strategies [1]. Jobs that are already profiled are directly submitted to the resource management system.

Resource management system contains following components:

a) Secure Instant VM Allocation

In compare with existing Map-Reduce model that create VMs on demand, our model employs a secure instant VM allocation method that reduces response times for jobs. After execution of job completion, our model only destroys the Hadoop instance (all local data includes) used by the job but remains the VM to be used for other jobs. For the new job, only a quick Hadoop instance is needed which prevents having to recreate and boot up VMs. Operationally, our model creates pools of VMs of different instance types as shown in Fig. 2 and dynamically creates Hadoop clusters on them.

When time sharing a VM across jobs it is important to ensure that an untrusted Map-Reduce program is not able to gain handle the data. Our security management is based on SELinux [8]. SELinux access policies in a Map-Reduce cloud does not lead to performance overheads. SELinux requires creation of two domains, one trusted and the other one untrusted. The Hadoop framework runs in the trusted domain and the untrusted user programs runs in the untrusted domain. While the trusted domain has regular access privileges, the untrusted domain has very limited permissions and has no access to any trusted files and other system resources.

b) Job Scheduler

The job scheduler in the cloud provider makes an integral component of our system. Where existing Map-Reduce model simply provision customer-specified VMs to execute the job, Virtual Machine scheduler is faced with the challenge of scheduling jobs among available VM pools while minimizing global cloud resource usage. Therefore, carefully executing jobs in the best VM type and cluster size among the available VM pools becomes a crucial factor for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

performance. The scheduler has knowledge of the relative performance of the jobs across different cluster configurations from the predictions obtained from the profile and analyzer and uses it to obtain global resource optimization [1].

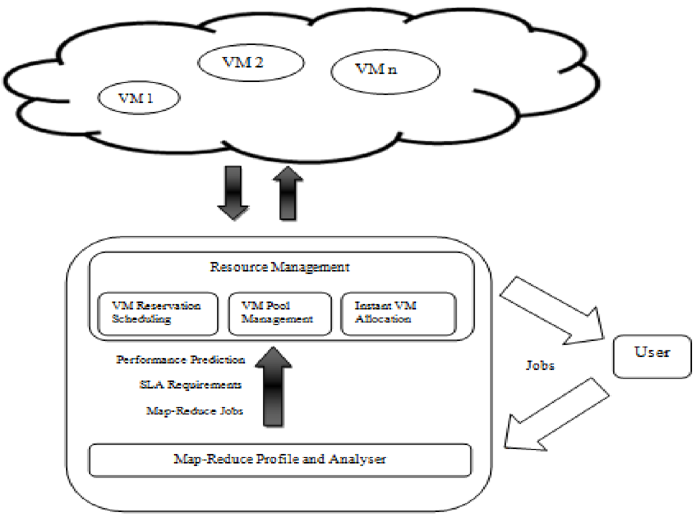


Figure 1: SYSTEM ARCHITECTURE

c) VM pool manager

The third main component of our model is the VM Pool Manager that deals with the challenge of dynamically managing the VM pools to help the job scheduler effectively obtain efficient resource allocations. If more number of jobs in the current workload requires small VM instances and the cloud infrastructure has fewer small instances, the scheduler will be forced to schedule them in other instance types leading to higher resource usage cost. The VM pool manager understands the current workload characteristics of the jobs and is responsible for online reconfiguration of VMs.

IV. PROPOSED MODEL

In this section, we present core VM resource scheduling and reservation management techniques and reconfiguration of VM pool techniques. We present VM scheduling efficiently schedule and reserve customer's jobs and present reconfiguration based VM pool technique that dynamically manages the VM instance pools by adaptively reconfiguring VMs based on current workload requirements.

a) Virtual Machine Scheduling and Reservation

The aim of the cloud provider is to minimize the infra-structure cost of the data centre by minimizing the number of servers required to handle the data centre workload. The peak workload decides the data centre infrastructure cost. The VM job scheduler schedules customer's jobs within the available VM pools to meet job deadline to minimizing the overall resource usage in the data centre thereby reducing the total infrastructure cost.

There are four major task requires for Virtual machine scheduler. The scheduler decides (a) how many VMs to use for the jobs (b) when to schedule each job in job queue, (c) which VM pool to schedule and (d) decides best Hadoop configuration settings to be used for the job by consulting the Map-Reduce profile and analyzer. Based on deadlines of the submitted jobs, the Virtual machine scheduler makes future reservations on VM pool resources. In order to maintain the most agility in dealing with continually incoming jobs and minimizing the number of reservation cancellations, our model uses a strategy of trying to create minimum number of future reservations without under-utilizing any resources. For implementing this strategy, the scheduler operates by identifying the highest priority job to schedule at any given time and creates a tentative reservation of resources for that job. The end time of job execution in reservation time window as the bound for limiting the number of reservations. This ensures that we are not unnecessarily creating a large number of reservations which may need cancellation and rescheduling after another job with more stringent deadline enters to the queue.

Assume that job J_i have higher priority over job J_j if the schedule obtained by reserving job J_i after reserving job J_j

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

becomes higher resource usage cost compared to the schedule obtained by reserving job J_j after reserving J_i. The highest priority job is chosen such that it will incur higher overall resource usage cost if the highest priority job is deferred as compared to deferring any other job. If resource usage cost of the schedule obtained by reserving job J_i after reserving job J_j and Jlist represents the job queue, then the highest priority job is chosen as the job J_i as follows

$$\sum_{J_j \mathcal{E} Jlist} Cost(J_{j\,,} J_i) - Cost(J_{i\,,} J_j)$$

For each VM pool, the scheduler selecting the highest priority job, J_{prior} in the job queue and makes a reservation of resources in VM pool for it using the cluster configuration with the lowest possible resource cost at the earliest possible time depending on the performance predictions obtained from the Map-Reduce profile and analyser [9][3]. Once the highest priority job, J_{prior} is reserved for all VM pools, the reservation time windows for the corresponding VM pools are fixed. Subsequently, the scheduler selecting the next highest priority job in the job queue that are possible within the current reservation time windows of the VM pools. The scheduler keeps on selecting the highest priority job one by one in this manner and tries to make reservations of resources in the VM pools within the reservation time window. The scheduler stops reserving either when no more jobs are schedulable in the job queue or when the reservations have filled all the resources in VM pool.

Then at each time unit, the scheduler selects the job in the job queue for reservations for the current time and schedules them on the VM pools by creating Hadoop clusters of the required sizes in the reservation. If no new jobs arrived within this one unit of time, the scheduler can simply look at the reservations made earlier and schedule the jobs that are reserved for the current time. However, if some new jobs arrived within the last one unit of time, then the scheduler needs to check if some of the newly arrived jobs have higher priority over the reserved jobs. In this case, the scheduler may require to cancel some existing reservations to reserve some newly arrived jobs that have higher priority over the ones in the reserved list.

In case, the newly arrived jobs do not have higher priority over the reservation time window deciding jobs but have higher priority over some other reserved jobs, the scheduler will not cancel the reservations. Repeat the process of reserving jobs within the reservation time windows. For example schedule obtained for 15 jobs using 40 VMs in each VM type, VM-1, VM-2 and VM-3. Here we assume that jobs 1, 2, 5, 6, 7, 8, 9, 10, 11, 13, 15 have their optimal cluster configuration using VM-1 and jobs 3, 12, and 14 are optimal with VM-2 and job 4 is optimal with VM-3. Here, the VM-aware scheduler tries its best effort to minimize the overall resource usage cost by provisioning the right jobs in the right VM types and using the minimal cluster size required to meet the deadline requirements. However, when the optimal choice of the resource is not available for some jobs, the scheduler considers the next best cluster configuration and schedules them in a cost-aware manner.

b) Reconfiguration-Based VM Management

Reconfiguration-based VM manager understands the workload characteristics of the jobs as an online process and performs online reconfiguration of the underlying VM pools to better suit the current workload. Although the VM-aware scheduler tries to effectively minimize the global resource usage by scheduling jobs based on resource usage cost, it may not be efficient if the underlying VM pools are not optimal for the current workload characteristics. For example, the VM pool allocation can be reconfigured to have more small instances by shutting down some large and extra large instances if the current workload pattern requires more small instances.

The reconfiguration manager understands the current demands for each VM instance type in terms of the average number of VMs required for each VM type in order to successfully provision the optimal cluster configuration to the jobs observed in the reconfiguration time window. At the end of the reconfiguration time window period, the algorithm decides on the reconfiguration plan by making a suitable trade off between the performance enhancement obtained after reconfiguration and the cost of the reconfiguration process.

For an example VM-aware schedule obtained for 15 jobs using 40 VMs in each VM type, VM-1, VM-2 and VM-3. Here we assume that jobs 1, 2, 5, 6, 7, 8, 9, 10, 11, 13, 15 have their optimal cluster configuration using VM-1 and jobs 3, 12, and 14 are optimal with VM-2 and job 4 is optimal with VM-3. Here, the VM-aware scheduler tries its best effort to minimize the overall resource usage cost by provisioning the right jobs in the right VM types and using the minimal

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

cluster size required to meet the deadline requirements. However, when the optimal choice of the resource is not available for some jobs, the scheduler considers the next best cluster configuration and schedules them in a cost-aware manner.

V. SIMULATION RESULTS

In order to analyze the performance and cost benefits of our model, we developed a simulator in Java that uses the profiles and performance predictions developed from the real cluster. The simulator models a cloud datacenter with servers, each having a 16 core 2.53 GHz processors with 16 GB RAM. It implements both the VM-aware scheduling with the instant VM allocation and the reconfiguration-based VM management techniques. The execution time for each job in the simulation is assumed as the predicted execution time (based on the profiles generated from the profiling cluster) and a prediction error which could be either a positive or negative error within an assumed error bound. We first present the experimental evaluation of our model by comparing with the existing techniques for various experimental conditions determined by distribution of the job deadlines. The evaluation of job deadlines on the performance of our model (Fig.5) with our techniques for different maximum deadlines with respect to number of servers requires for the cloud provider to satisfy the workload. Dedicated clusters for each customer's results in a lot of resources used based on the peak requirements of each customer and therefore the resources are under-utilized. On the other hand, per-job cluster services require lower number of servers as these resources are shared among the customers. However, our model has much lower resource requirement having up-to 80 percent reduction in terms of the number of server due to the design of global optimization capability of our model.

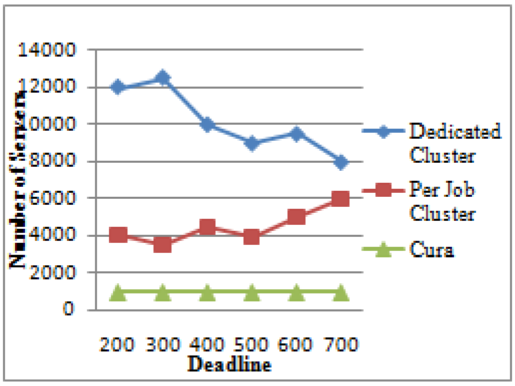
We also compare our model in terms of mean response time in (Fig 5b) model and dedicated cluster approach have lower response time (up to 65 percent). In the per-job cluster model, the VM clusters are created for each job and takes additional time for VM creation and booting process before the jobs can begin execution leading to the increased response time of the jobs. Similar to the comparison on number of servers, we see the same trend with to the per-job cost (in Fig 5.c) shows that our model can significantly reduce the per-job infrastructure cost of the jobs (up to 80 percent). Finally we compare our model in terms of utilization of resources in Fig. 5d. The per-job services spend a lot of resources in creating VMs for every job alignment. Especially with short response time jobs, the VM creation becomes a big overhead and reduces the effective utilization. The dedicated cluster approach does not create VMs for every job, however it has poor utilization because dedicated clusters are sized based on peak utilization. But our model has a high effective utilization having up-to 7x improvement compared to other techniques as our model effectively leverages global optimization and deadline-awareness to achieve better resource management.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

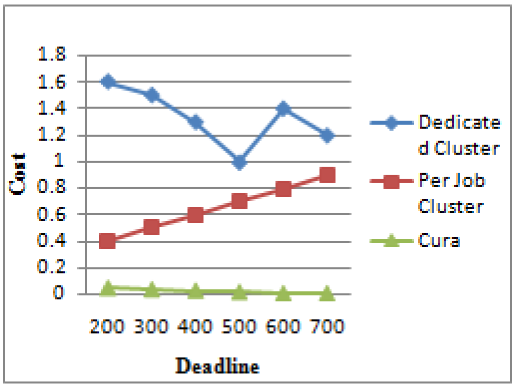
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

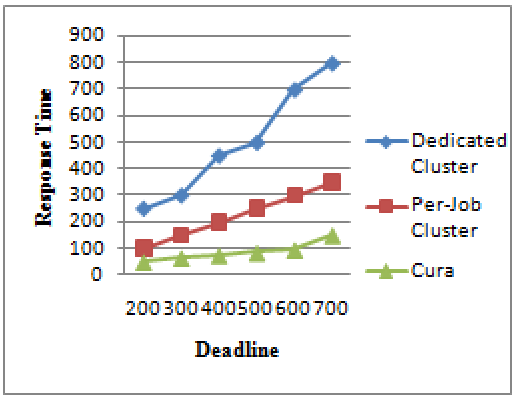
Vol. 3, Issue 11, November 2015



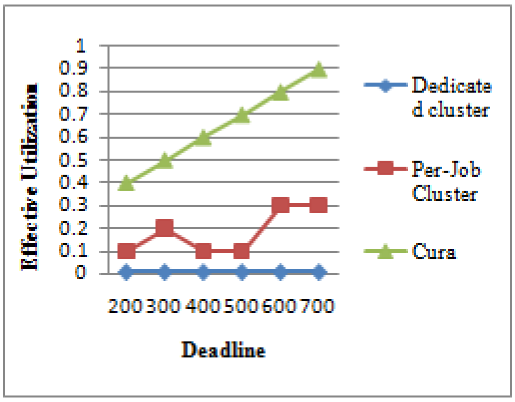
(a) Number of Servers



(c) Cost



(b) Response Time



(d) Effective Utilization

FIG 5. EFFECT OF DEADLINES

VI. CONCLUSION AND FUTURE WORK

Our model automatically creates the best cluster configuration for the jobs using MapReduce profiling and analyzer. Deadline awareness was introduced which delays the execution of certain jobs. It enables the cloud provider to optimize global resource allocation efficiently and reduces the costs. This model also uses a unique secure instant VM allocation technique that ensures fast response time. Guarantees for short interactive jobs, a significant proportion of modern MapReduce workloads. Our model resource management techniques include MapReduce profile and analyzer, cost-aware resource provisioning, VMware scheduling and online virtual machine reconfiguration. Finally reduction in infrastructure cost and decreased job response times can be obtained.

The future work is to enhance the scheduling and clustering. Thus the cost effective scheme is further improved by enhancing the allocation. And also the job used for future reservation is computed by the SLA is also consider and the process will be analyzed. Thus in future work the cost effective scheme is fully satisfied by enhancing the proposed model also believe that resource pricing in a globally optimized cloud can be quite a challenge and needs attention from both business perspective as well as from the resource management perspective.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

REFERENCES

[1] Balaji Palanisamy, Aameek Singh, and Ling Liu, "Cost-Effective Resource Provisioning For Map-Reduce in cloud," in Proc. IEEE 19th Annu. Transaction on Parallel and Distirb., 2015.
[2] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in Map-Reduce clusters using Mantri," in Proc. 9th USENIX Conference. vol. 11, pp.87-96, 2010.
[3] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating MapReduce performance using workload suites," in Proc. IEEE 19th Annu. Int. Symp. Model., Anal., Simul. Comput. Telecom-mun. Syst., 2011, pp. 390–399.
[4] Amazon Elastic Map-Reduce. [Online]. Available: http://aws. amazon.com/elasticmapreduce/, 2014
[5] Amazon Elastic Compute Cloud. [Online]. Available: http://aws. amazon.com/ec2/, 2014
[6] Hadoop. [Online]. Available: http://hadoop.apache.org, 2014.
[7] B. Igou, "User survey analysis: Cloud-computing budgets are growing and shifting; traditional IT services providers must prepare orperish," Gartner, Inc., Stamford, CT, USA, Gartner Rep. G00205813, 2010.
[8] SELinux user guide. [Online]. Available: http://selinuxproject. org/page/MainPage, 2014.
[9] H. Herodotou et al., and S. Babu, "On optimizing Map-Reduce programs/Hadoop jobs," Proc. VLDB Endowment, vol. 12, 2011.
[10] B. Sotomayor, K. Keahey, and I. Foster, "Combining batch execution and leasing using virtual machines," in Proc. 17th Int. Symp. High Perform. Distributed Computing, 2007, pp. 87–96.