© 2015 IJSRST | Volume 1 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Engineering and Technology

Approaches for Efficient Workflow Planning and Execution of Scientific Application with Cloud Resources to Maximize Throughput within a Limited Time Limit

Bakul Panchal
Department of Computer Engineering L. D. College of Engineering, Ahmedabad, Gujarat, India

ABSTRACT

In Cloud Environment, Resources available to the client on demand with pay per usage. Higher throughput with minimal execution time can reduce the budget cost for client as well as can never violate Service Level Agreement (SLA). For scientific application like weather forecasting maximum calculation within a limited time scope must be requirement. Prediction of incoming future work and its parallel execution with unlimited cloud resources can fulfil our requirement. Various techniques should be developed for this strategy. Usage Pattern [1] scheme use knowledge base for predicting future jobs by pattern matching which help to reduce start-up overhead. Bag of Task [2] scheme provide concept of simultaneous execution of scientific tasks with variable resources on available cloud. Resources can be increased or decreased on demand which can reduce the overall cloud rent cost. Our approach suggests a technique to combine usage pattern with bag of task which can provide efficient result with higher throughput in minimum time. Predict the future work, estimate execution time, divide jobs in small tasks and execute them parallel with on-demand variable resources can provide good results.

Keywords : Cloud Computing, Bag of Task, Usage Pattern, eScience Application on Cloud, SLA, Higher Throughput.

I. INTRODUCTION

Cloud computing is a technology which provides internet based on demand pay per usage of various services like Infrastructure as a service, Software as a service and Platform as a service. Cloud computing create an illusion of an infinite resources by creating an instance of resource which can be helpful for very heavy large computing application like eScience application - weather forecasting. This type of application contains list of different task that may be independently executed. By executing them parallel we can reduce the execution time. Science Lab requires very large computation which cannot be done using their limited resources. They are using Grid infrastructure for large computation. The problem with grid infrastructure is that it works on batch processing system and maintains queue for various tasks. Due to this a task which require little execution time also take longer processing time due to long queue in Grid environment. Cloud provide on-demand resource instance which reduce the queue processing time and fast response time. Proper scheduling should be required to utilize the available cloud resource to minimize the rent cost of cloud services. Various approaches are designed for efficient usage of cloud resources to reduce the cost and get fast response time. In Greedy approach, as many cloud resources as possible are acquired at the beginning of an application execution. This approach will provide faster response time but resources are not efficiently utilized. Some resources are idle for most of the time during total application execution which increase the cloud rent cost because heavy eScience application take so many hours to finish application execution. Another approach is online scheduling in which scheduler make decision when resources should be required and for that moment resource should be acquired on demand base which can reduce the cloud rent cost. Cloud does not have queue problem so using this approach we can also get faster output within deadline. Bag of Task approach use Work Queue [3]with Replication using online scheduling technique. Here replicas of task are available to grid infrastructure and provide parallel execution but do not duplicate execution of same task. Another approach for eScience application is predication of future task. Prediction can be done by the scheduler using heuristic approach. Prediction algorithm using historical data are available to scheduler to predict the execution time of current eScience application [4]. In cloud environment historical data of an application also depends on factor of non-uniformity of available hardware resources. The goal of this various approaches to achieve low cost execution that minimizes total turnaround time and cloud rent cost within a deadline. Proper workflow with available cloud resource using discussed approaches can solve it.

II. METHODS AND MATERIAL

We identify relationships between Cloud Computing, Service-Oriented Computing (SOC) and Grid Computing.

2.1 Bag of Task Approach

In this approach independent tasks of an application are ready for execution. Initially scheduler starts parallel execution of task on available cloud resources. At the end of time period t, scheduler revaluates its decision by calculating throughput and estimating task behaviour for next turn. Cloudburst scheduler revaluates its initial decision throughout the application execution time. Once the heuristic information for tasks is available scheduler calculate the number of cloud instances required to complete application. Using this approach scheduler can acquire the cloud resources whenever they required and reduce the application execution cost.

Here the approach of Work Queue with Replica of tasks for available grid and cloud resources is used for task execution. It avoids running the same task once it is finished. When the application is finished, scheduler has complete information about tc (task completion time) and nc (number of resources) application acquired for execution. Now using the equation $p(t) = u(t) – c(nc(t))$ profit can be calculated where $u(t)$ is a utility function for an application. Online scheduler uses the feedback mechanism. It uses the heuristic information of executed task and makes decision to acquire resources for next available tasks.

Online scheduler use three different type of heuristic: (i) Conservative (ii) Derivative and (iii) Predictive. Conservative heuristic believes that grid has reached to a maximum required resources and it is fair enough for future task execution and cannot be change. Derivative Heuristic use difference of throughput value of last two turns ΔS=St-St-1. At first startup schedulers has no information about St and St-1. As the first turn it consider as warm up turn and calculate the throughput of available grid and cloud resources. Using predictive approach scheduler can also predict initial throughput from user defined resources and from available historical data of same type of application.

2.2 Usage Pattern Approach

Pattern matching with Knowledgebase in Prediction algorithm can predict task execution time [5], job startup time [6], queue waiting time [7] and resource requirement [8]. Prediction algorithm to find task characteristics using pattern matching with available historical data can work fine on Grid Infrastructure but with cloud it also add factor of different heterogeneous hardware resources execution capability. This approach includes (i) Meta scheduler which draws from AppleS [9] and (ii) GRADS [10] to maintain and use historical data for pattern matching of future activity. Using this approach we can predict future jobs by extracting user pattern for historical data. It reduces high startup overhead for a heavy and time critical application like eScience application. Knowledge-based should be created from zero or pre populated job information and updated after successful completion of an application. Knowledge-based heuristic should be very useful to predict the future job for similar type of application or job. Prediction algorithm classify in two models: (i) Statistical technique and (ii) Artificial Intelligence knowledge base technique. AppleS include good scheduler with application prediction using statistical technique. Ganapathi et also [11] propose good statistical model to predict the resource requirement and execution time of application on Cloud Environment. This model use Map Reduce [12] technique for tasks by using pre execution features an post execution performance metrics. In Artificial Intelligence technique Knowledge base heuristic should be created and maintain by the system itself by learning from past job execution experience. It includes instance based learning, genetic algorithms, case based learning together with heuristic for searching an optimization.

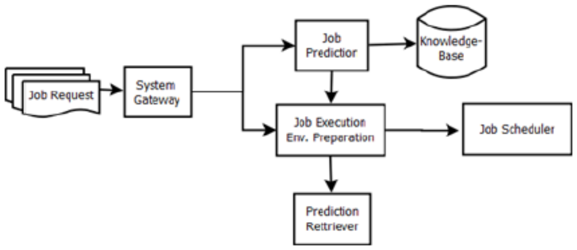

Figure 1: System Architecture

As per given figure1 System work as a middle ware or Meta scheduler. When Job request arrives, Job execution environment prepare environment and job predictor predict for next future job by extracting user patterns information in knowledgebase for same type of application. If next job is predicted then relevant required cloud resources should be prepared. If job is not predicted System prepare the environment and execute job with startup overhead and try to find reason why system is not able to predict job. During this execution cycle system maintains its knowledge base using self-learning approach.

2.3 Proposed Approach

We are using the combine approach of Usage pattern with Bag of Task execution on cloud environment. System structure use a heuristic knowledge base for various jobs - tasks of an application. Self-learning system always updates this knowledge base. Using this algorithm we can predict the future job and find out required execution time and cloud resources. So whenever cloud resources should be required they are rented which reduce the cost. Bag of Task provide an environment to execute available task in parallel with available cloud resources and efficiently utilize it. Conservative, Derivative and Predictive technique is used to predict the job as discussed in Bag of Task Scheduling. System work as Meta scheduler and provide efficient resource allocation and utilization for heavy application like eScience application. This reduces the total cost of execution of an application.

III. CONCLUSION

Using heuristic usage pattern with bag of task scheduling technique predict the future job, cloud resource requirement and task execution time. Advance knowledge of these characteristics can help to efficiently create an execution environment using efficient resource utilization. Resource can be rented only whenever they require which reduce the overall execution cost. Generally, this approach works fine for eScience application because it requires high computation power. Most of eScience application has similar type of characteristics so prediction using available Knowledgebase produces good accuracy. These reduce cost and provide fast response time within a deadline. Overall Service Level Agreement should be maintained because of fast response with good performance on cloud resources.

V. REFERENCES

[1] D. P. da Silva, W. Cirne, and F. V. Brasileiro. Trading cycles for information: Using replication to schedule bag-of -tasks applications on computational grids. In Euro-Par, pages 169–180, 2003.

[2] M. Armbrust et al., "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech.

[3] W. Smith, I. Foster, and V. Taylor, "Predicting application run times using historical information," in Job Scheduling Strategies for Parallel Processing. Springer, p. 122.

[4] H. Li, D. Groep, J. Templon, and L. Wolters, "Predicting job start times on clusters," in ccgrid. IEEE, 2004, pp. 301–308.

[5] D. Nurmi, J. Brevik, and R. Wolski, "QBETS: Queue bounds estimation from time series," in Job Scheduling Strategies for Parallel Processing. Springer, pp. 76–101.

[6] A. A. Julian, J. Bunn, R. Cavanaugh, F. V. Lingen, M. A. Mehmood, H. Newman, C. Steenberg, and I. Willers, "Predicting the resource requirements of a job submission arshadali," in In Proceedings of the Conference on Computing in High Energy and Nuclear Physics (CHEP 2004, 2004, p. 273.

[7] F. Berman et al., "Adaptive computing on the grid using apples," IEEE Transactions on Parallel and Distributed Systems, vol. 14, no. 4, pp. 369–382, 2003.

[8] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy, C. Kesselman, J. Mellor-Crumme et al.,

[9] "The GrADS project: Software support for high-level grid application development," International Journal of High Performance Computing Applications, vol. 15, no. 4, p. 327, 2001.

[10] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, "Statistics-Driven Workload Modeling for the Cloud," Technical Report

[11] UCB/EECS-2009-160, EECS Department, University of California, Berkeley, Tech. Rep., 2009.