International Journal of Innovative Research in Engineering Science and Technology

APRIL 2018 ISSN 2320 –981X

Improve Work-flow Scheduling Technique for Novel Particle Swarm Optimization in Cloud Manufacturing Environment

[1] Aslam Gowthar S, [2]VijayaKumar M

[1]Department of Computer Science and Engineering,

Nandha College of Technology,

Erode-638052, India.

aslamgowthar786@gmail.com

[2]Department of Computer Science and Engineering,

Nandha College of Technology,

Erode-638052, India.

tovijayakumar@gmail.com

ABSTRACT

Cloud computing is the latest distributed computing paradigm and Workflows have been frequently used to model large-scale scientific problems in areas such as bioinformatics, astronomy, and physics. Such scientific workflows have ever-growing data and computing requirements and therefore demand a high-performance computing environment in order to be executed in a reasonable amount of time. The workflows are commonly modeled as a set of tasks interconnected via data or computing dependencies. Cloud computing is the latest distributed computing paradigm and it offers tremendous opportunities to solve large-scale scientific problems. Presents various challenges need to be addressed in order to be efficiently utilized for workflow applications. The existing system works fail to meet the user's quality of service (QoS) requirements or to incorporate some basic principles of cloud computing such as the elasticity and heterogeneity of the computing resources. In addition to that this project proposes a resource provisioning and scheduling strategy for scientific workflows on

infrastructure as a service and platform as services clouds. Presents an algorithm based on the particle swarm optimization (PSO), which aims to minimize the overall workflow execution cost while meeting deadline constraints.

INTRODUCTION

The characteristic of preceding works developed for group of grids and resources is their center of attention on meeting application deadlines of the workflow though ignoring the cost of the exploit infrastructure. Even suited for such policies developed, environments for clouds are obliged to consider the usage-per-use model of the infrastructure in order to avoid prohibitive and preventable costs.

Our proposed work is based on the meta-heuristic optimization technique, particle swarm optimization (PSO). PSO is based on a swarm of particles moving through hole and converse with each other in order to finding an optimal search direction. PSO have been better totaling performance than other evolutionary algorithms and fewer parameters to tune, which makes it easier to implement. Many problems in different areas has been successfully addressed by adapting PSO to specific domains; for instance this technique has been used to solve problems in areas such as reactive voltage control, pattern recognition and data mining among others. In this paper, proposed system develops a static cost-minimization.

The main contributions of this paper are:

- To define the problem of scheduling prioritized workflow ensembles under budget and deadline constraints.
- To analyze and develop several dynamic and static algorithms for task scheduling and resource provisioning that rely on workflow structure information (critical shortest paths and workflow levels) and estimates of task runtimes in multi cloud providers

EXISTING SYSTEM

The existing system develops a static cost-minimization, deadline-constrained heuristic for scheduling a scientific workflow application in a cloud environment. The approach considers fundamental features of IaaS providers such as the dynamic provisioning and heterogeneity of unlimited computing resources. Achieve the both scheduling and resource provisioning are merged and modeled as an optimization problem.PSO is then used to solve such problem and produce a schedule defining not only the task to resource mapping, but also the number of nodes to be assigned.

DRAWBACKS

- Adaptable only in situations where same initial set of resource availability.
- Suitable for single cloud service provider environment only.
- Data transfer cost is not considered between different cloud data centers.

PROPOSED SYSTEM

The proposed system contains all the existing system implementation. In addition, it extends the resource model to consider the data transfer cost between data centers so that nodes can be deployed on different regions. The algorithm include a task is assigned to a node with sufficient memory to execute it will be included in the algorithm. It assigns different options for the selection of the initial resource pool.

ADVANTAGES

- Adaptable in situation where the multiple set of initial resource are availability.
- Suitable for multiple cloud service provider environments.
- Data transfer cost is reduced between different cloud data centers.

SYSTEM ARCHITECTURE

5.1 Cloud Provider Module

This module is used to add the cloud provider details to the database table. The cloud provider id and the cloud provider name are added to the table. All the record details can be viewed using the Grid View control in a form. The 'Cloud Providers' table is used to store the records. The resource details must include which cloud provider id it belongs to.

5.2 Resources Module

This module is used to add the resource details to the database table. The resource id and the resource name and cloud provider id are added to the table. All the record details can be viewed using the Grid View control in a form. The 'Resources' table is used to store the records. The cloud provider ids are fetched from the 'Cloud Providers' table and any one id is selected as resource type for this record.

5.3 Process Module

This module is used to add the process details to the database table. The process id and the process name are added to the table. All the record details can be viewed using the Grid View control in a form. The 'Processes' table is used to store the records. The details regarding which process use which resource is added later. The task details contain which process it belongs to.

5.4 Assign Process/Resource Module

This module is used to add the process/resource details to the database table. The process resource id (used as primary key), process id and the resource id are added to the table. All the record

details can be viewed using the Grid View control in a form. The 'Process Resource' table is used to store the records. The details regarding which process use which resource is assigned here.

5.5 Tasks Module

This module is used to add the task details to the database table. The task id and the task name and process id, resource id and time taken in that resource are added to the table. All the record details can be viewed using the Grid View control in a form. The 'Task' table is used to store the records. The process ids are fetched from the 'Processes' table and any one id is selected as process id for this record. The relationship between process id and task id is one-to-many relationship.

5.6 Execution Time Matrix Generation Module

This module generates the execution time matrix in which number of resources is taken as columns and tasks are taken as rows and the time the tasks taken to complete in those resources are stored as values.

5.7 Transfer Time Matrix Generation Module

This module generates the transfer time matrix in which number of taken are taken as columns and rows (square matrix is prepared) and the time a task transfers the data to other task is stored as values. So, the diagonal elements are always zero since same task has no data transfer operation.

5.8 Schedule Generation Module

Initially, the set of resources to lease R and the set of tasks to resource mappings M are empty and the total execution cost TEC and time TET are set to zero. After this, the algorithm estimates the execution time of each workflow task on every resource r_i $R_{initial}$. This is expressed as a matrix in which the rows represent the tasks, the columns represent the resources and the entry $ExeTime_{i,j}$ represent

the time it takes to run task ti on resource r_j. This time is calculated using Equation (1). The next step is the calculation of the data transfer time matrix. Such matrix is represented as a weighted adjacency matrix of the workflow DAG where the entry $TransferTime_{i,j}$ contains the time it takes to transfer the output data of task t_i to task t_j. This value is taken from database and is zero whenever ij or there is no directed edge connecting t_i and t_j. An example of these matrices is shown in Figure a) and b) below.

$$exeTime = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array} \begin{array}{c} \begin{array}{ccc} r_1 & r_2 & r_3 \end{array} \\ \begin{bmatrix} 2 & 1 & 4 \\ 4 & 3 & 6 \\ 10 & 6 & 15 \\ 7 & 4 & 12 \\ 8 & 4 & 10 \\ 3 & 2 & 7 \\ 12 & 7 & 18 \\ 9 & 5 & 20 \\ 13 & 8 & 19 \end{bmatrix} \end{array}$$

(a)

Fig 5.1: Matrix Representation of Execution Time

$$transferTime = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 \end{array} \\ \begin{bmatrix} 0 & 9 & 9 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(b)

Fig 5.2: Matrix Representation of Transfer Time

CONCLUSION

The dissertation presented the SEMO (Superior Element Multitude Optimization) algorithm which is used to predict the least time computation in the cloud provider area. In addition, the dissertation compared the time evaluation work between one dynamic resource flows to another process flow of dynamic resource in the cloud environment. In addition, it extends resource model to consider the data centers between data transfer cost so that nodes can be deployed on different regions. Extending the algorithm to include heuristics that ensure a task is assigned to a node with sufficient memory to execute it will be included in the algorithm. Also, it assigns the different options for selection of initial resource pool. In addition, data transfer cost between data center are also calculated as to minimize the cost of execution in multi cloud services provider environments. The future development for this dissertation is designed as web service and it can be integrated in many web sites. Also, it will be implemented in PaaS and SaaS service model. In addition, the future enhancement will plan to analyses performance of cost estimation based on multi-processor work in the cloud.

REFERENCES

[1] R. Buyya, J. Broberg, and A. M. Goscinski, Eds., Cloud Computing: Principles and Paradigms, vol. 87, Hoboken, NJ, USA: Wiley, 2010.

[2] Y. Fukuyama and Y. Nakanishi, “A particle swarm optimization for reactive power and voltage control considering voltage stability,” in Proc. 11th IEEE Int. Conf. Intell. Syst. Appl. Power Syst., 1999, pp. 117–121.

[3] L. Golubchik and J. Lui, “Bounding of Performance Measures for Threshold-baded Systems: Theory and Application to Dynamic Resource Management in Video-on-Demand Servers” , IEEE Transactions of Computers, 51(4), pp 353-372, April 2002.

[4] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," Future Generation Computation System., vol. 29, no. 3, pp. 682–692, 2012.

[5] Maria Alejandra Rodriguez and Rajkumar Buyya, "Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds", ieee transactions on cloud computing, vol. 2, no. 2, April June 2014

List of Figures