Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.39 – 44

SURVEY ARTICLE

A SURVEY ON WORKFLOW SCHEDULING IN CLOUD USING ANT COLONY OPTIMIZATION

[1]J.Elayaraja, [2]S.Dhanasekar

[1]PG Scholar, CSE, Info Institute of Engineering, Coimbatore
[2]Assistant Professor, CSE, Info Institute of Engineering, Coimbatore
[1] jergem@gmail.com, [2] dhanasekar_sethupathi@gmail.com

Abstract- Schedulers for cloud computing determine on which processing resource jobs of a workflow should SSbe allocated. In hybrid clouds, jobs can be allocated on either a private cloud or a public cloud on a pay per use Sbasis. The capacity of the communication channels connecting these two types of resources impacts the makespan and the cost of workflow execution. Our new approach introduces Ant Colony Optimization for the scheduling problem in hybrid clouds presenting the main Heuristics such as cost, makesapn, number of cores (multicore), and available bandwidth to be considered when scheduling workflows. Ant Colony Optimization is one of the best optimization techniques in scheduling workflows using heuristics.

Keywords*: *Ant Colony Optimization (ACO); grid computing; workflow scheduling

I.INTRODUCTION

Cloud computing has attracted an increasing number of users because it offers computational capabilities as services on a pay-per-use basis Cloud providers offer computing and storage resources, and platforms for software development and execution, as well as software interfaces accessible throughout the network. Three models of cloud services are commonly available: infrastructure as a service (IaaS), platform as a service (PaaS), or software as a service (SaaS). In SaaS, the clients use applications but cannot control the host environment. Google Apps and Salesforce.com are examples of this model. In PaaS, the platform is typically an application framework, and clients use a hosting environment for their applications. Examples of PaaS are the Google App Engine and Amazon Web Services. In IaaS, the clients use computing resources such as processing power and storage, and they can also control the environment and the deployment of applications. Amazon Elastic Compute Cloud (EC2), Globus Nimbus Toolkit, and Eucalyptus are good examples of this service model.

The types of cloud computing are private, public, community, and hybrid cloud. Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally. It will require the organization to reevaluate decisions about existing resources. Public cloud provides applications, storage, and other resources available to the general public by a service provider. These services are free or offered on a pay-per-use model. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and offer access only via Internet (direct connectivity is not offered). It shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally.

The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized. Hybrid cloud is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. By utilizing "hybrid cloud" architecture, companies and individuals are able to obtain degrees of fault tolerance combined with locally immediate usability without dependency on internet connectivity. Hybrid cloud architecture requires both on-premises resources and off-site (remote) server-based cloud infrastructures.

In summary, clients can use/run applications from a SaaS cloud; both develop and run their applications on a development platform provided by a PaaS cloud, or extend their computational capacity by leasing computing resources from an IaaS cloud. Moreover, clients can execute most applications using their own computing infrastructure (private cloud), and yet lease service from a cloud provider (public cloud) on demand. The operation of such hybrid cloud involves two fundamental questions

- What resources should be leased?
- Which tasks should be executed on the leased resources?

These answers are provided by a scheduler, a fundamental component of distributed computing systems including clouds and grids. These questions are answered considering the capacities of the communication links connecting the available resources. Slow communication channels increase delays, thus increasing the execution time (makespan) of applications, with bounds typically negotiated in service level agreements. Understanding the impact of network delays and costs on scheduling decisions is thus fundamental for cloud service provisioning. In line with that, this paper provides a brief survey of scheduling algorithms for hybrid clouds and the impact of communication networks on scheduling decisions.

The objective of the proposed ACO algorithm is to find a feasible schedule analyzing the heuristics defined for cloud resources. To accomplish this, a scheduling algorithm is to be developed that maximizes the performance of computation in cloud in terms of the clients preferences.

II. LITERATURE SURVEY

The Hybrid Cloud Optimized Cost (HCOC) algorithm [1] schedules workflows in hybrid clouds by first attempting costless local scheduling using HEFT. If the local scheduling cannot meet the deadline, the algorithm selects jobs for scheduling in resources from the public cloud. When selecting resources from the public cloud, the HCOC algorithm considers the relation between the number of parallel jobs being scheduled and the number of cores of each resource as well as deadlines, performance, and cost. As with the MDP algorithm, the objective is to minimize the financial cost, obeying the deadlines stipulated by the user in a single-level SLA contract.

In [2], the workflow scheduling problem was formulated as an integer linear program that considers the leasing of reserved and on-demand resources from multiple IaaS providers according to a two-level SLA. The scheduler can run in either a SaaS or PaaS cloud provider, and receive workflow execution requests with dead-lines from its clients (first SLA level), but it can also lease resources from multiple IaaS providers (second SLA level).

The Heterogeneous Earliest Finish Time (HEFT) [3] scheduling algorithm was designed for heterogeneous computing systems. Since it was developed before the advent of cloud computing and utility grids, it does not consider monetary costs. Its objective is to minimize the workflow makespan.

The deadline-driven cost-minimization algorithm [4] or Deadline-Markov Decision Process (MDP) breaks the DAG into partitions, assigning a maximum finishing time for each partition according to the deadline set by the user. Based on this time, each partition is scheduled for that resource, which will result in the lowest cost and earliest estimated finishing time. This algorithm works with on-demand resource reservation.

Abrishami et al. [5] presented the Partial Critical Paths (PCP) algorithm, which schedules the workflow in a backward fashion. Constraints are added to the scheduling process when such scheduling of jobs in a partial critical path fails so that the algorithm will be restarted. This algorithm presents the same characteristics as does MDP, although it involves greater time complexity, since a relatively large number of rescheduling can be demanded during the execution of the algorithm.

The self-adaptive global search optimization technique called particle swarm optimization (PSO) is utilized to schedule workflows in the algorithm proposed in [6]. It was developed to work in clouds with single-level SLAs and on-demand resource leasing. It considers neither multi core resources nor workflow deadlines, but focuses solely on monetary cost minimization.

A. Scheduling In Clouds

Applications and services can be decomposed into sets of smaller components, called jobs. For example, an application that processes a large image can decompose this image into smaller ones for parallel processing by distinct jobs. The logical sequence of the jobs of an application is called a workflow, which is commonly represented by a directed acyclic graph (DAG). The nodes of a DAG represent the jobs of a workflow, while arcs represent their data dependencies. A job can be executed only after the data on which it depends has been produced and sent to the resource where it will be executed. Such applications can be found in a variety of fields, such as physics (astronomy, thermodynamics), bioinformatics (DNA sequencing, proteomics), chemistry (protein dynamics), and computer science (computer vision, image processing).

A hybrid IaaS cloud composed of the resources of the private cloud as well as those of one or more public IaaS clouds. A hybrid cloud scheduler must decide which resources should be leased from the public clouds to guarantee the execution of the workflow within the specified maximum execution time (deadline). After the submission of workflow by a user, a broker runs the scheduling algorithm to start the decision making process. Besides deciding which resources will be used, the scheduler also determines which part of the workflow will run in each cloud provider. One challenging issue in hybrid clouds is how interfaces can be provided to interact automatically with different existing public clouds so that the broker can gather information about resources, and the workflow can be executed and monitored in a variety of public cloud infrastructures. Some projects, such as the JClouds (www.jclouds.org), try to solve this problem by providing portable abstractions to several existing cloud providers. Another challenge involves the consideration of security requirements of the applications, which can reduce the pool of potential hosts for scheduling jobs.

The scheduling problem involved is known to be NP-complete in general, including the scheduling of workflows in heterogeneous computer systems discussed in this paper. Scheduling algorithms often utilize heuristics and optimization techniques to try to obtain a near optimal schedule. The input of the scheduling algorithm must include the DAG that represents the workflow of jobs and their dependencies, as well as information about the target system, including the processing capacity of each resource and the available capacities of the network links.

This information is obtained from a resource information repository in the private cloud. Moreover, scheduling algorithms for clouds are usually cost-aware, so that the information about the cost per time unit of usage of each resource must be available. Providing with this information, the scheduling algorithm is capable of estimating the workflow makespan and its execution costs.

B. Schedulers For Clouds

Cloud computing evolved from grid computing, service oriented computing, and virtualization paradigms. This means that scheduling algorithms developed for these types of systems can also be used in clouds. Scheduling algorithms can be distinguished by their main characteristics, such as:

• **Target system:** The system for which the scheduling algorithm was developed, which can be a heterogeneous system, a grid, or a cloud computing system.

• **Optimization criterion:** Makespan and cost are the main metrics specified by cloud user and considered by schedulers in the decision making process.

• **Multicore awareness:** Computer systems can have multiple cores, which should be considered by scheduling algorithms in resource selection.

• **On-demand resources:** Resources can be leased either on-demand or for long terms. The on-demand leasing of resources is treated by the scheduling algorithm as a "single expense" during the execution of the workflow.

• **Reserved resources:** The algorithm should consider the use of a resource reserved for a long term.

• **Levels in a service level agreement (SLA):** The scheduling algorithm should consider that SLAs can be organized hierarchically. SLAs with a single level allow clients and providers to interact directly to negotiate resource capacities and prices.

Heuristics information: Heuristics information is some problem based values to guide the search direction. Based on the problem of workflow we design heuristics here. The main heuristics are

- Cost
- Makespan
- Bandwidth
- Multicore

C. Ant Colony Optimization

Ant Colony Optimization (ACO) is a metaheuristic combinatorial optimization technique that mimics the foraging behavior of Ants. As shown in the following figure, ACO starts with initialization of parameters and ants along with permissible range. Each ant and its permissible range are processed to construct path. The constructed path is explored to form different combination of possible discrete values for the decision variable and the objective function is evaluated. Check whether the optimum solution is reached or not. If yes stop the process otherwise the best path is chosen from the evaluated values and the pheromone updating is carried out on the best path to form new set of permissible ranges for the next iteration.

While implementing ACO for any scheduling algorithm, the following steps are to be addressed.

- Initialization of Pheromone
- Initialization of Heuristic information
- Random generation of Ants
- Mapping of Ant with path
- Evaluation of objective function
- Pheromone updating

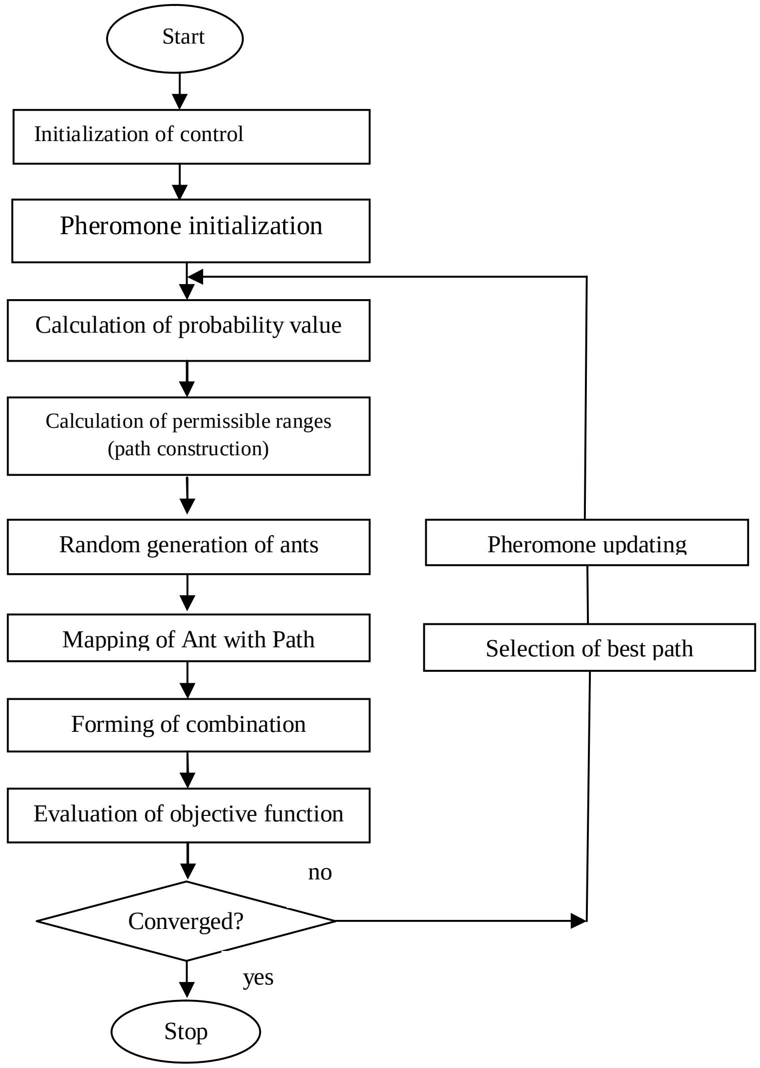

Fig. 1 workflow diagram using ACO

III.CONCLUSION

ACO optimizes workflow scheduling in hybrid cloud using heuristics information such as cost, makespan, bandwidth and multicore. To prove this technique to be effective it can be compared with other optimization techniques such as particle swarm optimization etc. The future research may be continued on multi objective problem in hybrid cloud.

REFERENCES

[1] *L. F. Bittencourt and E. R. M. Madeira, "Scheduling Service Workflows for Cost optimization in Hybrid Clouds," J. Internet Svcs. and Apps.*

[2] *T. A. L. Genez, L. F. Bittencourt, and E. R. M. Madeira,"Workflow Scheduling for SaaS/PaaS Cloud Providers Considering Two SLA Levels," IEEE/IFIP NOMS , Apr.2012.*

[3] *H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-Effective and Low-Complexity Task Scheduling for Het-Computing," IEEE Trans. Parallel and Distrib. Sys., vol. 13, no. 3, 2002, pp. 260–74.Conf. e-Science and Grid Computing, July 2005,pp. 140–47.*

[5] *S. Abrishami, M. Naghibzadeh, and D. Epema, "Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths," 11th IEEE/ACM GRID, Oct. 2010, pp. 81–88.*

[6] *S. Pandey et al., "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," 24th IEEE AINA, Apr. 2010, pp. 400–07.*

[7] *H. XiaoShan and S. XiaoHe, "QoS guided min-min heuristic for grid task scheduling," J. Comput. Sci. Technol., vol. 18, no. 4, pp. 442–451, 2003.M. M. Lopez, E. Heymann, and M. A. Senar, "Analysis of dynamic heuristics*

[8] *M. Maheswaran et al., "Dynamic mapping of a class of independent tasks onto heterogeneous computing systems," J. Parallel Distrib. Comput., vol. 59, pp. 107–131, 1999.*

[9] *L.Wang, H. J. Siegel, V. P. Roychowdhury, and A. A. Maciejewski, "Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach," J.Parallel Distrib. Comput., vol. 47, pp. 8–22, 1997.*

[10] *J.-K. Kim, et al., "Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment," J. Parallel Distrib.*

[11] S. Zheng,W. Shu, and L.Gao, "Task scheduling using parallel genetic simulated annealing algorithm," in *Proc. IEEE Int. Conf. Service Operations Logist. (SOLI'06)*, pp. 46–50.

