Indian Journal of Science and Technology, *Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102058, October 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Survey and Analysis on Task Scheduling in Cloud Environment

P. Akilandeswari* and H. Srimathi

Computer Science and Engineering, SRM University, Chennai - 203203, Tamil Nadu, India;
akilandeswari.p@ktr.srmuniv.ac.in, srimathi.h@ktr.srmuniv.ac.in

Abstract

Objectives: In cloud environment many scheduling algorithms are available to achieve the quality of service, but the scalability of task execution increases, scheduling becomes more complex. So there is a need for better scheduling. This paper deals with the survey of dynamic scheduling, different classification and scheduling algorithms currently used in cloud providers. **Analysis**: This paper focuses on a comparative study of static and dynamic task scheduling currently used and various researchers work on dynamic task scheduling for many applications. **Findings:** Scheduling methods/techniques used by various cloud service providers, advantages and limitations of each method are discussed. Scheduling algorithms in cloud setup in each paper is analyzed. **Application:** Analyzing task scheduling in cloud environment, uniform standard can be applied for all cloud providers. Online scheduling and dynamic scheduling can be further improved

Keywords: Cloud Computing, Cloud Setup, Static and Dynamic Scheduling, Task Scheduling Algorithms, Various Metrics

1. Introduction

Cloud computing is utility based environment as pay per use model achieved by Parallel, Distributed and Cluster computing accessed through the Internet. A key advantage of cloud computing is on-demand self-service, scalability and elasticity. In on-demand self-service, the cloud user can request, deploy their own software, customize and pay for their own services. Scalability is achieved through virtualization. Being elastic in nature, cloud service gives the infinite computing resources (CPU, Memory and Storage).

The rapid growth of cloud environment applications is executed in parallel, to achieve minimum execution time. Tasks are assigned to machines (matching) and execution order of the tasks referred as scheduling. Scheduling is one of the ways to achieve the quality of service in the cloud environment. To improve the quality of service, any scheduling algorithm or scheduling method can be implemented. In cloud environment, many cloud consumers wants the different quality of service requirements. The different consumer requirements are efficient scheduling, traffic control, dynamic resource provisioning, admission control etc. Some cloud consumer needs job completion on the same day with specified time, the cost is not a constraint. Another consumer wants job completion after some days with reduced cost, another consumer may need image resolution or audio quality must be the best. Sometimes single consumer may have many requests. For all these kind of requirements in the cloud, task scheduling is the challenging one. The existing task scheduling techniques offered by major cloud providers are good and efficient manner. But there is no uniform standard for cloud providers. There is need of analyzing various scheduling algorithms to improve the quality of service.

2. Classification of Scheduling

Generally, scheduling in cloud computing is classified into different categories. The first category is based on Task. Based on task scheduling is divided into static scheduling and dynamic scheduling. In static scheduling task arrives simultaneously at the processor and the tasks are submitted on the available resources, scheduling decisions are

**Author for correspondence*

taken before tasks are submitted. The processing time is updated after task completion; this kind of task-based scheduling is mostly applied for the periodic task. In the case of dynamic scheduling number of task, machine location and resource allocation is not fixed. Arrival times of the tasks are not known before submission. Further dynamic scheduling is classified into two types either batch mode or online mode. In batch mode tasks are queued, collected into a set and scheduled after fixed period of time. In online mode task is scheduled when they arrive in the system.

The second category is based on various metrics used in cloud and classified into a batch system, interactive system and real-time system. In batch system turnaround time and throughput can be calculated. The response time and fairness can be calculated using an interactive system and in real-time system deadline is measured. The third category is performance and market-based. In the performance-based focus on optimal execution time not considering cost, certain policies are considered for mapping the task and execution. In the market-based purely consider cost as the factor. Backtrack algorithm, Genetic Algorithm are based on this market-based scheduling algorithm. In static scheduling, all traditional scheduling algorithms can be implemented FCFS, round robin, min-min and max-min. In dynamic scheduling, all heuristic scheduling algorithms can be implemented in Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing, Ant Colony Optimization and dynamic list scheduling.

Dynamic tasks are mostly represented by directed acyclic graph; various metrics used to calculate the quality of service in the cloud are makespan, early completion time, total execution time, cost, trust, fairness, deadline, response time, reliability and availability. Table 1 shows current scheduling techniques, advantages and limitations by various cloud service providers.

In[1] designed a job schedule that minimizes job completion time on the set of map reduce jobs. The proposed framework applies classic Johnson algorithm which uses a new framework called the balanced pool of resources. Framework efficiently utilizes characteristics and properties of map reduce jobs in a given workload for constructing the optimized job schedule which improves makespan 50% in Hadoop cluster. Performance and accuracy are compared with Johnson optimal schedule and worst schedule. In future instead of finding job completion time for each job, smart job ordering is addressed then that will be the new solution for the Hadoop cluster.

In[2] proposed job scheduling for homogeneous cluster computing. It dealt with virtual machine technology to share fractional node resources in a precise and controlled manner for the various job. Mark Stillwell focus on batch scheduling, which limits overall resource utilization since some nodes were ideal. Next problem was focused on user-centric metrics response time and fairness. Fractional scheduling addressed the scheduling problem based on these metrics. The key idea for the online non-clairvoyant version of scheduling algorithm could afford improvement in performance for HPC workloads. Comparison with batch scheduling, dynamic fractional scheduling improved the performance. Still the algorithm was found unsuitable for underutilization.

In[3] proposes rolling horizon architecture for real-time aperiodic independent task scheduling and energy saves. Author focus on virtualization technique applied for task through rolling horizon. This rolling horizon architecture sorts the incoming task in order by its deadline. The real-time controller checks whether a task in the rolling-horizon can be finished before its deadline, if not the VM controller add VM to finish the task. Energy-aware scheduling also handled through virtual machine by calculating start time and execution time. If task can be allocated, then it selects the VM yielding minimal energy consumption to execute the task, otherwise, the algorithm rejects task. The energy-aware scheduling is tested in simulator CloudSim but since this is real time task if it is applied in real cloud environment will improve energy efficiency.

In[4] evaluates trust service oriented task workflow scheduling algorithm to find the optimal solution for execution time with deadline constraint. The scheduling algorithm is designed for cloud service selection algorithms to form optimum workflow application. To meet user constraints the scheduling algorithm combines direct trust and recommendation includes QoS metrics time, cost. The trust oriented workflow scheduling allocates task suitable for cloud service, with fuzzy and multi-objective functions. The multi-objective functions are time, cost and trust. Workflow is represented as DAG with a dependent task. Minimum completion time is calculated by length of the critical path. Trust is calculated by weight corresponds to the similarity between the active user and each of the other users. Calculations are compared with Minimum Critical Path and greedy cost model. The trust workflow scheduling has statically predetermined

Table 1. Scheduling algorithms/methods by various cloud providers

Cloud providers	Scheduling Algorithm	Advantages	Limitations
Hadoop	Fair scheduling	Simple, low overhead, useful for periodically running jobs	Job priority. High priority gets more chance, low priority gets less chance
Facebook	Fair scheduling	Achieve high data locality	Job priority
Yahoo	Capacity scheduler	Uses cluster sharing, capacity scheduler partition the resources into pools in the form of multilevel queue, achieve maximum resource utilization and throughput	Only abstraction is queues which are setup by the administrator that reflects cost of the shared cluster, if short job comes after long one it will have to wait until the long one finish
Google app engine, Heroku	CRON (command Run scheduler on notice) this is time-based job scheduler	Cron is most suitable for scheduling repetitive tasks	Cron is not suited for complex, event-driven tasks, the type of jobs that automate tasks across the enterprise.
Open nebula	Rank matchmaker scheduling algorithm	Matchmaker is to prioritize cloud resources more suitable for the Virtual machine.	Based on highest rank capacity, data stores are allocated, over committing capacity will result in negative values
Eucalyptus	Uses cluster controller to a specific node currently uses greedy and round robin scheduler	Simple, easy to implement	Only Near optimal solution is achieved, reconsidering the previous solution is difficult
Salesforce.com	APEX jobs	Tasks are executed in a regular manner, it reduces more complexity from a programmer's point of view.	Execution delayed on the basis of service availability. At a time only, 100 apex jobs are allowed.
Windows Azure	Web role, worker role	High-performance, in-memory, dedicated caching service Table storage reduces cost	Speed
Amazon EC2	XEN, Genetic Algorithm	Xen-thin provisioning, live migration, support for enterprise storage (ISCSI arrays, FC arrays), CPU overcommit, good Windows support	Genetic Algorithm need artificial intelligence and fuzzy logic

schedule but in a dynamic runtime environment, some adaptive method is needed to get optimum result.

In[5], proposes a market-oriented hierarchical scheduling strategy in cloud workflow systems. The author explains about service level scheduling and task-level scheduling where service-level scheduling deals the task to service assignment. In task-level scheduling deals with the task to VM assignment in the local data center to minimize total execution cost. The scheduling strategies presented random scheduling along with meta-heuristic scheduling algorithm. The author adopts meta-heuristic algorithm such as Genetic Algorithms, Ant Colony Optimization and Particle Swarm Optimization. For each algorithm QoS metrics, makespan, cost and CPU time compared, results show Ant Colony Optimization is better than the other. In this paper different kinds of a workflow are applied, QoS constraint investigated for dynamic optimization.

In[6] brought out a framework capable of adjusting system workload in incremental steps for overloaded real time system. And each task has been assigned a critical value with utilization function evaluated in overloaded condition till optimal criteria are met. Also, a periodic task is scheduled and performance is achieved near optimal results.

In[7] developed a static cost minimization based on deadline constraint in the scientific workflow. Particle Swarm Optimization was used to solve resource provisioning and scheduling. The analysis was done for gravitational waves, bioinformatics and seismograms on cost and makespan metrics. The Table 2 gives various scheduling algorithms implemented by the researchers.

Table 2. Comparison of scheduling algorithms

Scheduling algorithms	Metrics	Methodology	Cloud setup
Parallel Bee Colony Optimization Particle Swarm Optimization (PBCOPSO)-Hybrid technique[8]	makespan and resource utilization	Algorithms were framed with initialization, forward pass and backward pass steps. Forward and backward passes are performed till a stopping criterion is met. Evaluate and Update individual and global best fitness functions and each task is assigned, average time is computed, proposed algorithm gives optimal solution in terms of makespan and resource utilization	CloudSim
Constraint Based Decentralized Algorithm[9]	makespan	This paper focus on efficiency of task scheduling on constraint based decentralized model. it allocates the task to the unoccupied nodes that follows makespan minimization policy to reduce task completion time.	Tree based network overlay
list scheduling and Dynamic min–min scheduling[10]	Task execution time	With preemptable tasks and Adjust the resource allocation dynamically based on the updated information of the actual task executions.	1024 Node cluster, with 4 Intel IA-64 1.4 ghz Itanium processors, 8 gb memory, and 185 GB disk space per node; 1152 node cluster, with 8 AMD Opteron 2.4 ghz processors, 16 GB memory, and 185 GB disk space per node; 2048 processors bluegene/L system with 512 MB memory, 80 GB memory.
Genetic Algorithm[11]	Performance	GA scheduling function every task scheduling cycle, evaluates the quality of each task schedule with user satisfaction	12 vm, 2000 task generated
Dynamic multitask scheduling[12]	Makespan	Ordinal optimization using rough models and fast simulation is introduced to obtain suboptimal solutions in a shorter timeframe	128 VM on amazonec2
An adaptive virtualized GPU resource Scheduling algorithm[13]	Performance	Feedback control that diminishes the impact of the runtime uncertainties on the system performance	GPU virtualization Framework
Rule-based scheduling algorithms- hyper-heuristic scheduling algorithm[14]	Makespan	Uses two detection operators to automatically finds when to change the low-level heuristic algorithm and a perturbation operator to fine-tune the solutions obtained by each low-level algorithm.	CloudSim (a simulator) and Hadoop (a real system)
Benefit-fairness algorithm based on new Berger's model[15]	Cost and fairness, efficiency	Model states about task allocation based on fairness priority and calculated user expectation value of resources till convergence	CloudSim
Particle Swarm Optimizatio[16]	Total execution Cost and the total execution time	PSO does the schedule, resource mapping, time to be leased for task scheduling	CloudSim
Online scheduling algorithm[17]	Cost-delay parameter and energy-fairness parameter	Scheduling batch jobs among multiple geographically distributed data centers For arbitrarily random job arrivals in queue	Four data centers, each one consisting of up to 200 blade servers arranged in four racks.

Rolling-horizon scheduling architecture for real-time task[18]	Dead line	The real-time controller finds whether a task in the rolling-horizon, that can be finished before its deadline. If it is not finished, the real-time controller informs the VM controller, and then the VM controller adds VM to finish the task within Timing constraint.	CloudSim, Additional settings Each host is modeled to have one CPU core and the CPU performance is 1,000 MIPS, 1,500 MIPS or 2,000 MIPS.
Gain scheduling[19]	Response time and throughput	Using fuzzy logic controller gain scheduling is achieved	XEN based cloud test bed
Earliest Deadline First for scheduling[20]	Deadline	This incremental algorithm is that its runtime overhead and The quality of the solutions are parameters that can be controlled online.	Dynamic environment set up: 100 independent simulations. The INCA server .For each simulation, 5,000 tasks are generated dynamically.
Round-robin scheduler [21]	Response time	Paper focuses primarily on designing and evaluating adaptive schemes that exploit on-line measurement and take decisions with low computational Overhead for fast on-line decision making	The three hosts (with 2.8 ghz, 2.4 ghz, and 3.0 ghz, respectively, dual-core CPU, are used for task execution, while a separate host (2.8 ghz dual-core CPU) supports the controller.

3. References

1. Verma A, Cherkasova L, Roy HC. Orchestrating an ensemble of map reduce jobs for minimizing their makespan. IEEE Transactions on Dependable and Secure Computing. 2013; 10(5):314–27.
2. Stillwell M, Vivien F, Casanova H. Dynamic fractional resource scheduling versus batch scheduling. IEEE Transactions on Parallel and Distributed Systems. 2012 Mar; 23(3):521–9.
3. Zhu X, Yang LT, Chen H, Wang J, Yin S, Liu X. Real-time tasks oriented energy-aware scheduling in virtualized clouds. IEEE Transactions on Cloud Computing. 2014 Apr-Jun; 2(2):168–80.
4. Tan W, Sun Y, Li LX, Lu G, Wang T. A trust service-oriented scheduling model for workflow applications in cloud computing. IEEE Systems Journal. 2014 Sep; 8(3):868–78.
5. Wu Z, Liu X, Ni Z, Yuan D, Yang Y. A market-oriented hierarchical scheduling strategy in cloud workflow systems. Journal of Supercomputing. 2013; 16:256–93.
6. Mejıa-Alvarez P, Melhem R, Mosse D, Aydin H. An incremental server for scheduling overloaded real-time systems. IEEE Transactions on Computers. 2003; 52(10):1347–67.
7. Rodriguez MA, Buyya R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing. 2014; 2(2):222–35.
8. Priyadarsini RJ, Arockiam L. PBCOPSO: A Parallel Optimization Algorithm for Task Scheduling in cloud environment. Indian Journal of Science and Technology. 2015; 8(16):1–5.
9. Lavanya M, Janani TN, Sushmita S, Sunandha M, Vaithiyanathan V, Saravanan S. A constraint-based decentralized task routing approach to large-scale scheduling in cloud environment. Indian Journal of Science and Technology. 2015; 8(35):1–4.
10. Li J, Qiu M, Ming Z, quan G, Qin X, Gu Z. Online optimization for scheduling preemptable tasks on IaaS cloud systems. Journal Parallel and Distributed Computing. 2012; (72):667–8.
11. Jang SH, Kim TY, Kim JK, Lee JS. The study of Genetic Algorithm-based task scheduling for cloud computing. International Journal of Control and Automation. 2012; 5(4):1–6.
12. Zhang F, Cao J, Tan W, Khan SU, Li K, Zomaya AY. Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud. IEEE Transaction on Emerging Topics in Computing. 2014; 2(3):338–51.
13. Zhang C, Yao J, Qi Z, Yu M, Guan H. vGASA: Adaptive scheduling algorithm of virtualized GPU resource in cloud gaming. IEEE Transactions on Parallel and Distributed Systems. 2014; 25(11):3036–45.
14. Tsai CW, Huang WC, Chiang MH, Chiang MC, Yang CS. A hyper-heuristic scheduling algorithm for cloud. IEEE Transactions on Cloud Computing. 2014; 2(2):236–50.
15. Sun H, Chen SP, Jin C, Guo K. Research and simulation of task scheduling algorithm in cloud computing. Telkomnika. 2013; 11(11):6664–72.

16. Rodriguez MA, Buyya R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing. 2014; 2(2):222–35.
17. Polverini M, Cianfrani A, Ren S, Vasilakos AV. Thermal-aware scheduling of batch jobs in geographically distributed data centers. IEEE Transactions on Cloud Computing. 2014; 2(1):71–84.
18. Zhu X, Yang LT, Chen H, Wang J, Yin S, Liu X. Real-time tasks oriented energy-aware scheduling in virtualized clouds. IEEE Transactions on Cloud Computing. 2014; 2(2):168–80.
19. Rao J, Wei Y, Gong J, Xu CZ. QoS guarantees and service differentiation for dynamic cloud applications. IEEE Transactions on Network and Service Management. 2013; 10(1):43–55.
20. Mejia-Alvarez P, Melhem R, Mosse D, Aydin H. An incremental server for scheduling overloaded real-time systems. IEEE Transactions on Computers. 2003; 52(10):1347–61.
21. Wang L, Gelenbe E. Adaptive dispatching of tasks in the cloud. IEEE Transactions on Cloud Computing. 2015; PP(99):1. DOI: 10.1109/TCC.2015.2474406.