

Optimizing Load balancing algorithms with MIPS rating factor of cloud computing system

Pooja¹, Virender Kadyan²

¹M.Tech Scholar, CSE Department, RNCET Panipat, Haryana

²Assistant Professor, CSE Department, RNCET Panipat, Haryana

Abstract— Cloud computing is an infrastructure that involves the integrated and collaborative use of computers, networks, databases and scientific instruments owned and managed by multiple organizations. A major contribution is to utilize the dynamism of virtualized Cloud resources in various workflow management operations. Several algorithms are proposed throughout the dissertation, each focusing on a different aspect of the larger problem, from monitoring individual services, to placing a new service workflow in the Cloud, to dynamically reallocating resources across different services to satisfy demands and reduce costs. The goal is to add an end-to-end solution to the Cloud provider's offerings to workflow owners so that the latter can host their workflows in the Cloud smoothly without worrying about managing the underlying Cloud resources themselves. We show through experimental results, from both real world cluster trace logs and synthetic data, that the proposed approaches can perform various management tasks for service workflows efficiently.

Keywords— ACO, Cloud Computing, MIPS rating.

I. INTRODUCTION

Cloud computing is all the rage. "It's become the phrase du jour," says Gartner senior analyst Ben Pring, echoing many of his peers. The problem is that (as with Web 2.0) everyone seems to have a different definition. As a metaphor for the Internet, "the Cloud" is a familiar cliché, but when combined with "computing," the meaning gets bigger and fuzzier. Some analysts and vendors define Cloud computing narrowly as an updated version of utility computing: basically virtual servers available over the Internet. Others go very broad, arguing anything you consume outside the firewall is "in the Cloud," including conventional outsourcing.

Cloud computing comes into focus only when you think about what IT always needs: a way to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities.

Cloud computing is at an early stage, with a motley crew of providers large and small delivering a slew of Cloud-based services, from full-blown applications to storage services to spam filtering. Yes, utility-style infrastructure providers are part of the mix, but so are SaaS (software as a service) providers such as Salesforce.com. Today, for the most part, IT must plug into Cloud-based services individually, but Cloud computing aggregators and integrators are already emerging.

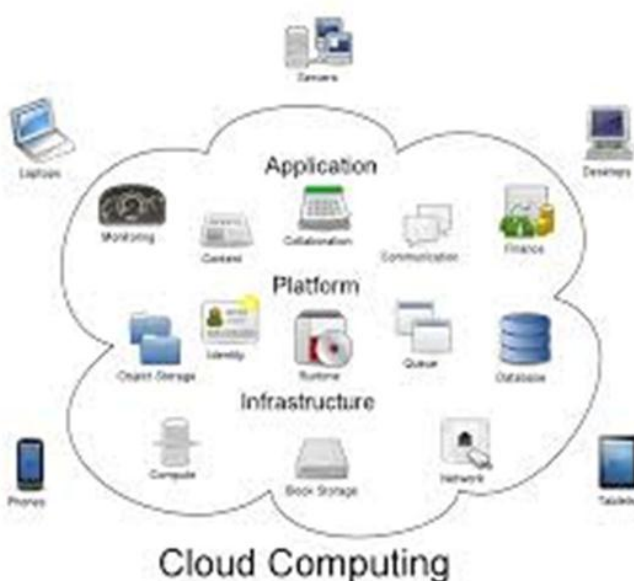


Figure 1



Figure 2

InfoWorld talked to dozens of vendors, analysts, and IT customers to tease out the various components of Cloud computing. Based on those discussions, here's a rough breakdown of what Cloud computing is all about:

1. SaaS

This type of Cloud computing delivers a single application through the browser to thousands of customers using a multitenant architecture. On the customer side, it means no upfront investment in servers or software licensing; on the provider side, with just one app to maintain, costs are low compared to conventional hosting. Salesforce.com is by far the best-known example among enterprise applications, but SaaS is also common for HR apps and has even worked its way up the food chain to ERP, with players such as Workday. And who could have predicted the sudden rise of SaaS "desktop" applications, such as Google Apps and Zoho Office?

2. Utility computing

The idea is not new, but this form of Cloud computing is getting new life from Amazon.com, Sun, IBM, and others who now offer storage and virtual servers that IT can access on demand. Early enterprise adopters mainly use utility computing for supplemental, non-mission-critical needs, but one day, they may replace parts of the datacenter. Other providers offer solutions that help IT create virtual datacenters from commodity servers, such as 3Tera's AppLogic and Cohesive Flexible Technologies' Elastic Server on Demand. Liquid Computing's LiquidQ offers similar capabilities, enabling IT to stitch together memory, I/O, storage, and computational capacity as a virtualized resource pool available over the network.

3. Web services in the Cloud

Closely related to SaaS, Web service providers offer APIs that enable developers to exploit functionality over the Internet, rather than delivering full-blown applications. They range from providers offering discrete business services -- such as Strike Iron and Xignite -- to the full range of APIs offered by Google Maps, ADP payroll processing, the U.S. Postal Service, Bloomberg, and even conventional credit card processing services.

4. Platform as a service

Another SaaS variation, this form of Cloud computing delivers development environments as a service. You build your own applications that run on the provider's infrastructure and are delivered to your users via the Internet from the provider's servers. Like Legos, these services are constrained by the vendor's design and capabilities, so you don't get complete freedom, but you do get predictability and pre-integration. Prime examples include Salesforce.com's Force.com, Coghead and the new Google App Engine. For extremely lightweight development, Cloud-based mashup platforms abound, such as Yahoo Pipes or Dapper.net.

5. MSP (managed service providers)

One of the oldest forms of Cloud computing, a managed service is basically an application exposed to IT rather than to end-users, such as a virus scanning service for e-mail or an application monitoring service (which Mercury, among others, provides). Managed security services delivered by SecureWorks, IBM, and Verizon fall into this category, as do such Cloud-based anti-spam services as Postini, recently acquired by Google. Other

offerings include desktop management services, such as those offered by CenterBeam or Everdream.

6. Service commerce platforms

A hybrid of SaaS and MSP, this Cloud computing service offers a service hub that users interact with. They're most common in trading environments, such as expense management systems that allow users to order travel or secretarial services from a common platform that then coordinates the service delivery and pricing within the specifications set by the user. Think of it as an automated service bureau. Well-known examples include Rearden Commerce and Ariba.

7. Internet integration

The integration of Cloud-based services is in its early days. OpSource, which mainly concerns itself with serving SaaS providers, recently introduced the OpSource Services Bus, which employs in-the-Cloud integration technology from a little startup called Boomi. SaaS provider Workday recently acquired another player in this space, CapeClear, an ESB (enterprise service bus) provider that was edging toward b-to-b integration. Way ahead of its time, Grand Central -- which wanted to be a universal "bus in the Cloud" to connect SaaS providers and provide integrated solutions to customers -- flamed out in 2005.

Today, with such Cloud-based interconnection seldom in evidence, Cloud computing might be more accurately described as "sky computing," with many isolated Clouds of services which IT customers must plug into individually. On the other hand, as virtualization and SOA permeate the enterprise, the idea of loosely coupled services running on an agile, scalable infrastructure should eventually make every enterprise a node in the Cloud. It's a long-running trend with a far-out horizon. But among big metatrends, Cloud computing is the hardest one to argue with in the long term.

II. RESOURCE MANAGEMENT ISSUE

Resource management is a core function required of any man-made system. It affects the three basic criteria for system evaluation: performance, functionality and cost. Inefficient resource management has a direct negative effect on performance and cost. It can also indirectly affect system functionality. Some functions the system provides might become too expensive or ineffective due to poor performance.

A Cloud computing infrastructure is a complex system with a large number of shared resources. These are subject to unpredictable requests and can be affected by external events beyond your control. Cloud resource management requires complex policies and decisions for multi-objective optimization. It is extremely challenging because of the complexity of the system, which makes it impossible to have accurate global state information. It is also subject to incessant and unpredictable interactions with the environment.

The strategies for Cloud resource management associated with the three Cloud delivery models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS), differ from one another. In all cases, the Cloud services providers are faced with large, fluctuating loads that challenge the claim of Cloud elasticity. In some cases, when they can predict a spike can be predicted, they can provision resources in

advance. For example, seasonal Web services may be subject to spikes.

For an unplanned spike, the situation is slightly more complicated. You can use Auto Scaling for unplanned spike loads, provided there's a pool of resources you can release or allocate on demand and a monitoring system that lets you decide in real time to reallocate resources. Auto Scaling is supported by PaaS services such as Google App Engine. Auto Scaling for IaaS is complicated due to the lack of standards.

In the Cloud, where changes are frequent and unpredictable, centralized control is unlikely to provide continuous service and performance guarantees. Indeed, centralized control can't provide adequate solutions to the host of Cloud management policies you have to enforce.

Autonomic policies are of great interest due to the scale of the system, the large number of service requests, the large user population and the unpredictability of the load. The ratio of the mean to the peak resource needs can be large.

Policies and mechanisms

A policy typically refers to the principal guiding decisions, whereas mechanisms represent the means to implement policies. Separating policies from mechanisms is a guiding principle in computer science. Butler Lampson and Per Brinch Hansen offer solid arguments for this separation in the context of OS design. You can loosely group Cloud resource management policies into five classes:

The explicit goal of an admission control policy is to prevent the system from accepting workloads in violation of high-level system policies. For example, a system may not accept an additional workload that would prevent it from completing work already in progress or contracted. Limiting the workload requires some knowledge of the global system state. In a dynamic system, this information is often obsolete at best.

Capacity allocation means allocating resources for individual instances. An instance is a service activation. Locating resources that are subject to multiple global optimization constraints requires you to search a large space when the state of individual systems is changing so rapidly.

You can perform load balancing and energy optimization locally, but global load-balancing and energy-optimization policies encounter the same difficulties as the ones already discussed. Load balancing and energy optimization are correlated and affect the cost of providing the services.

The common meaning of the term load balancing is that of evenly distributing the load to a set of servers. For example, consider the case of four identical servers, A, B, C and D. Their relative loads are 80 percent, 60 percent, 40 percent and 20 percent, respectively, of their capacity. Perfect load balancing would result in all servers working with the same load—50 percent of each server's capacity.

In Cloud computing, a critical goal is minimizing the cost of providing the service. In particular, this also means minimizing energy consumption. This leads to a different meaning of the term load balancing. Instead of having the load evenly distributed among all servers, we want to concentrate it and use the smallest number of servers while switching the others to standby mode, a state in which a server uses less energy. In our example, the load from D will migrate to A and the load from C

will migrate to B. Thus, A and B will be loaded at full capacity, whereas C and D will be switched to standby mode.

Quality of service is that aspect of resource management that's probably the most difficult to address and, at the same time, possibly the most critical to the future of Cloud computing. Resource management strategies often jointly target performance and power consumption.

Dynamic voltage and frequency scaling (DVFS) techniques such as Intel SpeedStep and AMD PowerNow lower the voltage and the frequency to decrease power consumption. Motivated initially by the need to save power for mobile devices, these techniques have migrated to virtually all processors, including those used in high-performance servers. As a result of lower voltages and frequencies, the processor performance decreases. However, it does so at a substantially slower rate than the energy consumption.

Virtually all optimal or near-optimal mechanisms to address the five policy classes don't scale up. They typically target a single aspect of resource management, such as admission control, but ignore energy conservation. Many require complex computations that can't be done effectively in the time available to respond. Performance models are complex, analytical solutions are intractable, and the monitoring systems used to gather state information for these models can be too intrusive and unable to provide accurate data.

Therefore, many techniques are concentrated on system performance in terms of throughput and time in system. They rarely include energy tradeoffs or QoS guarantees. Some techniques are based on unrealistic assumptions. For example, capacity allocation is viewed as an optimization problem, but under the assumption that servers are protected from overload.

Control the Cloud

Allocation techniques in computer Clouds must be based on a disciplined approach, rather than ad hoc methods. The four basic mechanisms for implementing resource management policies are:

- Control theory: Control theory uses feedback to guarantee system stability and predict transient behavior, but it can only predict local behavior.
- Machine learning: A major advantage of machine-learning techniques is that they don't need a performance model of the system. You could apply this technique to coordinating several autonomic system managers.
- Utility-based: Utility-based approaches require a performance model and a mechanism to correlate user-level performance with cost.
- Market-oriented/economic mechanisms: Such mechanisms don't require a system model, such as combining auctions for bundles of resources.

A distinction should be made between interactive and non-interactive workloads. The management techniques for interactive workloads (Web services, for example) involve flow control and dynamic application placement, whereas those for non-interactive workloads are focused on scheduling.

A fair amount of work reported in the literature is devoted to resource management of interactive workloads—some to non-interactive and only a few to heterogeneous workloads, a combination of the two. Planning ahead for how you are going

to manage these will help ensure a smooth transition to working with the Cloud

III. EXISTING APPROACHES

Qiang et al. (2009) using feedback control theory, we present VM-based architecture for adaptive management of virtualized resources in Cloud computing and model an adaptive controller that dynamically adjusts multiple virtualized resources utilization to achieve application Service Level Objective (SLO) in Cloud computing. Compared with Xen, KVM is chosen as a virtual machine monitor (VMM) to implement the architecture. Evaluation of the proposed controller model showed that the model could allocate resources reasonably in response to the dynamically changing resource requirements of different applications which execute on different VMs in the virtual resource pool to achieve applications SLOs.

Younge et al. (2010) presented a new framework is presented that provides efficient green enhancements within a scalable Cloud computing architecture. Using power-aware scheduling techniques, variable resource management, live migration, and a minimal virtual machine design, overall system efficiency will be vastly improved in a data center based Cloud with minimal performance overhead..

Zhang et al. (2012) present an adaptive power management framework in the Cloud to achieve autonomic resource configuration. We propose a software and lightweight approach to accurately estimate the power usage of virtual machines and Cloud servers. It explores hypervisor-observable performance metrics to build the power usage model. To configure Cloud resources, we consider both the system power usage and the SLA requirements, and leverage learning techniques to achieve autonomic resource allocation and optimal power efficiency. We implement a prototype of the proposed power management system and test it on a Cloud testbed. Experimental results show the high accuracy (over 90%) of our power usage estimation mechanism and our resource configuration approach achieves the lowest energy usage among the compared four approaches

Datta et al. (2012) ease and simplified the web services rendering it user friendly, stretchable, affordable and adaptable with the growing demand and complexity of developing web services and based on the analysis of Human Resource Management and information system requirements for numerous enterprises. A CloudHR Management web services would provide a technologically viable solution to the IT world and other enterprises relating to Human Resource Management. A Cloud HR Management is an open-source HR Information System that covers Personal Information Management, Employee Self Services, Benefits, Leave and Salary Information Management.

Kaewpuang et al. (2013) propose a framework for resource allocation to the mobile applications, and revenue management and cooperation formation among service providers. For resource allocation to the mobile applications, we formulate and solve optimization models to obtain the optimal number of application instances that can be supported to maximize the revenue of the service providers while meeting their source requirements of the mobile applications. For sharing the revenue generated from the resource pool (i.e., revenue management)

among the cooperative mobile Cloud service providers in a coalition, we apply the concepts of core and Shapley value from cooperative game theory as a solution. Based on the revenue shares, the mobile Cloud service providers can decide whether to cooperate and share their sources in the resource pool or not. Also, the provider can optimize the decision on the amount of resources to contribute to the resource pool.

Pengbo et al. (2014) aim to design the network as the integration of the mobile access part and the Cloud computing part, utilizing the inherent heterogeneity to meet the diverse quality of service (QoS) requirements of tenants. Furthermore, we propose a novel cross-network radio and Cloud resource management scheme for HMC networks, which is QoS-aware, with the objective of maximizing the tenant revenue while satisfying the QoS requirements. The proposed scheme is formulated as a restless bandits problem, whose *indexability* feature guarantees the low complexity with scalable and distributed characteristics. Extensive simulation results are presented to demonstrate the significant performance improvement of the proposed scheme compared to the existing ones.

Zhao et al. (2014) propose a reference service framework for integrating scientific workflow management systems into various Cloud platforms, which consists of eight major components, including Cloud workflow management service, Cloud resource manager, etc., and 6 interfaces between them. We also present a reference framework for the implementation of Cloud Resource Manager, which is responsible for the provisioning and management of virtual resources in the Cloud. We discuss our implementation of the framework by integrating the Swift scientific workflow management system with the OpenNebula and Eucalyptus Cloud platforms, and demonstrate the capability of the solution using a NASA MODIS image processing workflow and a production deployment on the Science@Guoshi network with support for the Montage image mosaic workflow..

IV. ANT COLO ALGORITHM

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.

This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, [1][2] the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to protein folding or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to produce near-optimal solutions to the travelling salesman problem. They have an

advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

The first ACO algorithm was called the Ant system [8] and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities. At each stage, the ant chooses to move from one city to another according to some rules:

- 1) It must visit each city exactly once;
- 2) A distant city has less chance of being chosen (the visibility);
- 3) The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that that edge will be chosen;
- 4) Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short;
- 5) After each iteration, trails of pheromones evaporate

With an ACO algorithm, the shortest path in a graph, between two points A and B, is built from a combination of several paths. It is not easy to give a precise definition of what algorithm is or is not an ant colony, because the definition may vary according to the authors and uses. Broadly speaking, ant colony algorithms are regarded as populated metaheuristics with each solution represented by an ant moving in the search space. Ants mark the best solutions and take account of previous markings to optimize their search. They can be seen as probabilistic multi-agent algorithms using a probability distribution to make the transition between each iteration. In their versions for combinatorial problems, they use an iterative construction of solutions. According to some authors, the thing which distinguishes ACO algorithms from other relatives (such as algorithms to estimate the distribution or particle swarm optimization) is precisely their constructive aspect. In combinatorial problems, it is possible that the best solution eventually be found, even though no ant would prove effective. Thus, in the example of the Travelling salesman problem, it is not necessary that an ant actually travels the shortest route: the shortest route can be built from the strongest segments of the best solutions. However, this definition can be problematic in the case of problems in real variables, where no structure of 'neighbours' exists. The collective behaviour of social insects remains a source of inspiration for researchers. The wide variety of algorithms (for optimization or not) seeking self-organization in biological systems has led to the concept of "swarm intelligence", which is a very general framework in which ant colony algorithms fit.

V. RESULTS & CONCLUSION

In experimental work the execution time and cost incurred by proposed algorithm and existing random resource selection algorithm to execute tasks is compared against varying number of resources and tasks.

TIME_SHARED and SPACE_SHARED resource allocation policies are used to perform the experiments. In all twelve experiments are carried out:

Experiment 1 to Experiment 4 are performed using TIME_SHARED allocation policy with varying number of tasks and resources and execution time and cost is compare.

Experiment 5 to Experiment 8 are performed using SPACE_SHARED allocation policy with varying number of tasks and resource and execution time and cost is compared.

Experiment: 1

The Total Execution Time of Heuristic Resource Scheduling Algorithm (HRSA) is compared with Random Resource Scheduling Algorithm (RRSA) with the following parameters.

Resource Allocation Policy=TIME_SHARED

Number of Resources =25

Number of Tasks = 10 to 50

TABLE 1

Sr. No.	Num ber of Task s	Execution Time using HRSA(Sec.)	Execution Time using RRSA(Sec.)	Average Improve ment %
1	10	5181	8910	71.97
2	20	18182	29597	62.78
3	30	38747	67431	74.02
4	40	66593	116923	75.57
5	50	101353	180192	84.82

Average Improvement in Total Execution Time is = 72.42 %. Figure 4.1 shows that as the number of tasks increases the difference between execution time taken by two algorithms increases.

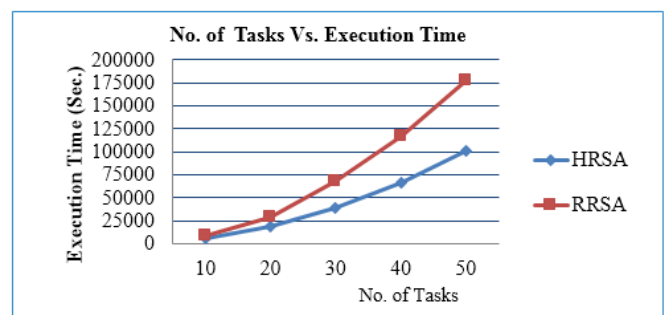


Figure 3 Number of Tasks Vs. Execution Time in TIME_SHARED Allocation.

VI. REFERENCES

- [1]. I. Foster, and C. Kesselman, "Globus: a metacomputing infrastructure toolkit", International Journal of High Performance Computing Applications, vol. 11, no. 2, pp. 115-128, 1997.

- [III]. I. Foster, and C. Kesselman(editors), The Cloud 2: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, USA, 2003.
- [III]. I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Cloud: enabling scalable virtual organizations", International Journal of High Performance Computing Applications, vol. 15, no. 3, pp. 200-222, 2001.
- [IV]. X.S. He, X.H. Sun, and G.V. Laszewski, "QoS guided min-min heuristic for Cloud task scheduling", Journal of Computer Science & Technology, vol.18, no. 4, pp. 442-451, 2003.
- [V]. V.D. Martino, and M. Mililotti, "Scheduling in a Cloud computing environment using genetic algorithms", Proceedings of the 16th International Symposium on Parallel and Distributed Processing, pp. 235-239, 2002.
- [VI]. Qiang Li; Qinfen Hao; Limin Xiao; Zhoujun Li, "Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control," Information Science and Engineering (ICISE), 2009 1st International Conference on , vol., no., pp.99,102, 26-28 Dec. 2009
- [VII]. Younge, AJ.; von Laszewski, G.; Lizhe Wang; Lopez-Alarcon, S.; Carithers, W., "Efficient resource management for Cloud computing environments," Green Computing Conference, 2010 International , vol., no., pp.357,364, 15-18 Aug. 2010
- [VIII]. Ziming Zhang; Qiang Guan; Song Fu, "An adaptive power management framework for autonomic resource configuration in Cloud computing infrastructures," Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International , vol., no., pp.51,60, 1-3 Dec. 2012
- [IX]. Datta, A; Islam, M.R.; Mukherjee, AK.; Kandar, D., "Cloud computing: A solution to Human Resource Management system," Radar, Communication and Computing (ICRCC), 2012 International Conference on , vol., no., pp.176,179, 21-22 Dec. 2012
- [X]. Kaewpuang, R.; Niyato, D.; Ping Wang; Hossain, E., "A Framework for Cooperative Resource Management in Mobile Cloud Computing," Selected Areas in Communications, IEEE Journal on , vol.31, no.12, pp.2685,2700, December 2013
- [XI]. Si, Pengbo; Zhang, Qian; Yu, F.Richard; Zhang, Yanhua, "QoS-aware dynamic resource management in heterogeneous mobile Cloud computing networks," Communications, China , vol.11, no.5, pp.144,159, May 2014
- [XII]. Zhao, Y.; Li, Y.; Raicu, I; Lu, S.; Lin, C.; Zhang, Y.; Tian, W.; Xue, R., "A Service Framework for Scientific Workflow Management in the Cloud," Services Computing, IEEE Transactions on , vol.PP, no.99, pp.1,1 doi: 10.1109/TSC.2014.2341235
- [XIII]. Z.H. Xu, X.D. Hou, and J.Z. Sun, "Ant algorithm-based task scheduling in Cloud computing", Proceedings of 2003 IEEE Canadian Conference on Electrical and Computer Engineering, vol. 2, no. 3, pp. 1107-1110, 2003.
- [XIV]. [14]. R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in Cloud computing", Concurrency and Computation: Practice and Experience, vol. 14, no. 2, pp.1507-1542, 2002.
- [XV]. R. Buyya, D. Abramson, and S. Venugopal, "The Cloud Economy", Proceedings of the IEEE, 3, vol. 93, no. 3, pp. 698-714, 2005.
- [XVI]. O.O. Sonmez, and A. Gursoy, "A novel economic-based scheduling heuristic for computational Clouds", International Journal of High Performance Computing Applications, vol.21, no. 1, pp. 21-29, 2007.
- [XVII]. R. Buyya, and Manzur Murshed, "CloudSim: A Toolkit for the Modeling, and Simulation of Distributed Resource Management, and Scheduling for Cloud Computing", Concurrency and Computation: Practice and Experience, vol. 14, no. 1, pp. 1175-1220, 2002.
- [XVIII]. Hao, "Implement of Computational Cloud Application Scheduling Simulation with CloudSim Toolkit", Journal of Jilin Normal University (Nature Science Edition), vol. 3, no. 1, pp. 63-64, 2003.
- [XIX]. Wang yue, and Tao Hongjiu, "Application in TSP Based on Ant Colony Optimization", Journal of Wuhan University of Technology (Information and Managing Engineering Edition), vol. 28, no. 11, pp. 24-26, 2006.
- [XX]. Marco Dorigo, and Luca Maria Gambardella, "Ant colonies for the traveling salesman problem", BioSystems, vol. 43, no. 2, pp. 73-81, 1997.
- [XXI]. Chen Ye, "Ant Colony System for Continuous Function Optimization", Journal of Sichuan University (Engineering Science Edition), vol. 36, no. 4, pp. 117-120, 2004.
- [XXII]. O. Ibarra and C. Kim, "Heuristic algorithms for scheduling independent tasks on nonidentical processors", Journal of the ACM (JACM), vol. 24, no. 2, pp. 280-289, 1977