# Result Analysis of Safety issues on Cloud Computing

Er. Amandeep Singh[1], Er. Rimanpal Kaur[2]
*[1]Research Scholar, M.Tech (CSE), CGC Technical Campus, Jhanjeri, Mohali, India*
*[2]A.P (CSE), CGC Technical Campus, Jhanjeri, Mohali, India*

***Abstract -*** The safety issues of cloud computing, is the branch of computer science, which is linked with the various applications or services, provided from the cloud platforms. Now-a-days a number of online applications and service are budding as the developers and researchers are kept working upon them. The mobile cloud computing applications may also be used for the purpose of various applications and how we can get secure for safety issues. The mobile or some other electronic gadgets in cloud computing is also being used as the remote processing unit for the mobile platforms. Any type of electronic gadgets must be capable of understanding the process load, data associated, total process cost, etc. and data must be secure in this device. The projected model has been premeditated as the main improvement to the progression partitioning models in the mobile cloud computing (MCC). The future model has been designed to understand the process load, data association, process dependency, and process return, process input and is capable to associate the processes in the batches. The proposed model has been designed irrespective of the specific mobile platforms. The proposed model of process partitioning and process offloading has been designed for the empowerment of processing on the low computationally powered devices by sharing their load with the cloud based platforms. The proposed model is designed to calculate the CPU time in the form of early finish time (EFT) and CPU cycles. The execution is calculated on the basis of latter parameters, whereas the communication cost is evaluated on the basis of process memory allocation, memory requirement & data size, which is further used for final decision making by comparing the communication cost with the process cost. The dynamic threshold is computed on the basis of the communication cost, EFT and CPU cycles, which plays the important role in taking the offloading decision. The experimental results have been evaluated in the form of time and cost based performance parameters. The experimental results have proven the effectiveness of the proposed model in comparison with the existing models.

***Keywords—***SaaS,HaaS,PaaS,IaaS

## I. INTRODUCTION

In the recent years, the main aim of IT industry is to lower the computational costs and achieve high productivity. This can be done by improving the utility of computing resources, reducing administrative costs and avoiding large amount of investments to provide a service. The main goal of any computing model is to make a better use of resources, put them together in order to achieve high throughput and able to tackle large computation problems and provide better results. Cloud computing is the recent technology, which was growing in popularity, which enables these functions. [16, 20] Cloud computing is coined recently and gained popularity in 2007, which can be also called as Internet based Distributed Computing. The roots of cloud computing can be traced to early stages of Grid computing and Distributed Computing. Cloud computing is a combination of utility computing (on-demand computing), Software as a service, and distributed computing. [14] Utility computing and Software as a service are two services that are provided by cloud computing, whereas distributed computing is one of the underlying technologies for implementing cloud computing. "Cloud Computing refers to both the applications delivered as a service over the internet and hardware and system software in the datacenters that provide these services."[1]. Mobile cloud computing intends to make the advantages of cloud computing for mobile users. It will also provide some additional functionality to the cloud as well. Mobile cloud computing helps to overcome the limitations of mobile devices in individual of the processing power and data storage. Mobile cloud computing helps to extend battery life by transferring the execution of reckoning severe application to the cloud. [23]. Mobile Cloud computing is regarded as the next generation computing infrastructure. Here the information is permanently stored in servers on internet and cached temporarily on clients through mobile devices. [70, 32] Although the wireless network brings us many benefits, there are many challenges that will hinder the growth of mobile computing. [21-22] With the growth of mobile industry, a substantial amount of mobile applications and facilities are existing. Now day's users are capable sharing and distributing digital media contents easily through internet. Here only authorized users who have obtained the license should access the information. [23]. Cloud computing is the recent trend that moves computing and data way from the desktop, portable PCs into large datacenters. The term "cloud" refers to the datacenters. Cloud computing takes the benefits of cyber infrastructure and is developed on the research of virtualization, distributed computing, grid computing and more recently networking, web and software services. According to R Buyya [2]. The definition of cloud computing are "A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers". [11]

## II. LITERATURE REVIEW

In this chapter, we take a closer look at our target environment, the Cloud, and the target application, Map Reduce. We also investigate related works regarding resource allocation, scheduling, and performance prediction problem.

Finally, we enumerate hardware and software infrastructure used in our work.

### A. *Cloud Computing*

The cloud computing environment refers to the hardware and systems software in the datacenters that provide computing resources as services [24]. Below is service models of cloud computing defined by National Institute of Science and Technology (NIST). Note that they used the term “consumer” instead of “user”. Software as a Service (SaaS). The capability provided to the consumer is to use the provider’s applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web- based email), or a program interface. The consumer does not manage or control the under- lying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment. Infrastructure as a Service (IaaS). The capability provided to the consumer is to 19 provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls). Throughout this thesis, we refer to IaaS by “the cloud environments” or “the Cloud” unless otherwise specified. SaaS and PaaS are also important service models of cloud computing, but we note that they are out of our focus in this thesis. The last few years have seen a dramatic growth in the availability and demand for “cloud” systems, or IaaS, best exemplified by Amazon’s EC2. Amazon’s EC2 [4]is one of the most widely used cloud services. It offers various virtual machine (VM) types with different capacities. Users rent a number of VM instances to run their applications and pay by the hour for active instances. Rackspace [35] and Joyent [56] offer similar services. Because these services are publicly available, they are often called “public” clouds. In contrast, “private” cloud refers to the infrastructure operated solely for a single organization. For example, Eycalyptus [7] is a software platform for the implementation of private cloud computing on computer clusters. It exports a user-facing interface that is compatible with Amazon’s EC2. Its resource allocation policy is modularized and extensible, and currently supports two simple policies; Greedy and Round-robin. The cloud users rent compute cycles, storage, and bandwidth with small minimum billing units (an hour or less for compute and per-MB for storage and bandwidth) and almost- instant provisioning latency (minutes or seconds). These systems contrast with traditional colocation centers (colos), where equipment leases span months and provisioning resources can take days or longer. [63]

### B. *Scheduling in a Heterogeneous Environment*

In many systems, a heterogeneous environment is preferable to one that is 23 homogeneous [17, 18]. However, it provides better performance only for particular systems and workloads [41]. Even if the workload itself is more suitable to a heterogeneous environment, the system’s scheduling algorithm should exploit heterogeneity well to benefit from it. Otherwise, it will lead to undesirable outcomes. The process of scheduling parallel tasks determines the order of task execution and the processor to which each task is assigned. Typically, an optimal schedule is achieved by minimizing the completion time of the last task. Finding the optimal schedule has long been known as an NP-complete problem in both homogeneous and heterogeneous environments [50]. Therefore, many heuristics have been proposed [64, 29, 27, 63] to find a feasible solution within a reasonable time. Some heuristics are known to have a bound on the deviation from the optimum [55]. However, most of those studies have concentrated on processing power and neglected other resources such as network bandwidth. In contrast, data-intensive computing systems, such as Hadoop [6] and Dryad [52], schedule tasks in favor of data-locality while assuming homogeneity in machines and tasks. If a machine or a task turns out to be slower than the others, it is treated as faulty and handled by speculative task re-execution.

## III. PROBLEM FORMULATION

The mobile cloud offloading is the process to send the mobile data to the cloud environments. The mobile cloud offloading techniques are applicable in many cases like cloud storage to save the storage space on mobile, offload the process computations over cloud to save the energy of cloud, etc. In the existing scheme, the authors have used the mobile data offloading scheme for process based offloading for mobile energy efficiency. The mobile devices analyze the process cost using a simple process cost calculation scheme based upon the exponential moving average algorithm. The existing process cost calculation scheme is not efficient enough because it does not calculate the computational cost according the real time mobile energy consumption prediction. The average CPU workload computation does not reflect the real energy consumption cost on the mobile devices. The proposed method will use an effective cost calculation mechanism for the mobile devices based upon the earliest finish time (EFT) which will not be based upon the pre-stored mobile resource information, which will make it acceptable for the variety of mobile devices.

## IV. PROPOSED MODEL

In the existing mobile data offloading model to offload the process data from mobile to cloud platforms utilizes the CPU time estimation method to make the offloading decision. The offloading decision is taken on the basis of a threshold value. The existing scheme does not evaluate the communication cost for process data, which may put the communication overhead more than the local processing cost, which reduces the effort for energy efficiency. Also, the existing process cost calculation scheme is not efficient enough because it does not calculate the whole process tree's computational cost according the real time mobile energy consumption prediction. The average CPU workload computation does not reflect the real energy consumption cost on the mobile devices. Also the existing scheme does not propose any solution to resolve the cost calculation conflicts, which can be produced by the different cost calculation and ranking resulted by the HEFT and Cheapest cost methods for the task batch. Rapid increment in the 67 multitasking possess a necessity of scheduling different processes in cloud computing. For scheduling process different scheduling algorithms are already proposed like DAG, HEFT, D- HEFT, EFT etc. HEFT can be also used in combination with DAG to attain lower complexity of scheduling of processes. This combination results in a process tree. Using this process tree, cost of each process can be found. Each process can be scheduled according to the found cost or the priority of the process. This efficient process Scheduling results in quicker response time, which in turns increase availability of resources to the customer. This paper describes work towards decreasing the complexity of existing budget-constrained state-of-the-art algorithms while maintaining the same budget level. The proposed method will use an effective cost calculation mechanism for the mobile devices based upon the earliest finish time (EFT) with DAG for process offloading proxy and remote execution management which will not be based upon the pre-stored mobile resource information, which will make it acceptable forth variety of mobile devices.

## V. OBJECTIVES

1. To develop the process generation model will become the first step towards the implementation of the proposed model.

2. To develop the process cost evaluation method will be implemented afterwards to evaluate the local process execution cost.

3. To develop the partitioning method the processes on the basis of cost calculated in the previous method in order to take the execution decision.

4. To finalize and obtain the results from the simulation.

5. To analyze and conclude the results obtained from the simulation.

## VI. FUTURE WORK

This research can be extensively analyzed to describe its significance towards decreasing the complexity of existing budget-constrained state-of-the-art algorithms while maintaining the same budget level. The proposed method can be improved by using an effective cost calculation mechanism for the mobile devices based upon the earliest finish time (EFT) which will not be based upon the pre-stored mobile resource information, which can make it highly acceptable for the variety of mobile devices.

## VII. FACILITIES REQUIRED FOR THE PROPOSED WORK

### A. *Minimum Hardware Requirements*

- A simple PC (With Wireless NIC embedded)
- 100 GB storage
- 2 GB RAM
- Multi Core Processor

### B. *Software Requirements*

- OS: Windows 7/8
- Simulator Option 1: MATLAB12a or above
- Simulator Option 2: Cloud sim 3.0 or above

## VIII. EXPERIMENTAL DESIGN

### A. *MATLAB*

MATLAB, short for Matrix-Laboratory, is a scientific computing environment developed by Math Works. It is mostly used to manipulate matrices, plot functions, implement algorithms, create user interfaces etc. So, it is ideal for computations that require extensive use of arrays and graphical analysis of data.

The design of MATLAB programming language is such that a powerful program can be written in a few lines of code. It can achieve a solution to complex problems in a relatively simple set of statements, as compared to the conventional general-purpose programming languages such as C++ or Java. Due to its vast area of application, it is widely accepted in science, economics and engineering research as well as industries. MATLAB help window can be accessed from 'help' in the menu bar. Help documentation can also be opened by simply pressing the F1 key. This opens up a window with all the content oriented around a wider topic. Selecting each topic further elaborates its sub-topics using examples and help for the correct syntax of related functions.

### B. *Version Used and System Specifications*

Table 8.2 MATLAB Specifications

| | |
|---|---|
| VERSION | R2014a |
| PROCESSOR | INTEL CORE 2 DUO |
| RAM | 2 GB |
| DISK SPACE | 4-5 GB |
| PLATFORM | WINDOWS 7 (64-bit) |

## IX. COMPONENTS

MATLAB has following components through which users interact with it:

**Command Window:** Command window is the place where the user can input commands to get a desired output. Non-graphical output is also displayed in this window. The '>>' operator signifies that the user can input commands, whereas if it is not visible, then the simulator is busy. Previously typed commands can be accessed using the UP arrow key. This helps the user in typing similar commands repetitively, thus saving time.

**Command History:** It displays the commands that have been executed in the present and previous sessions.

**Workspace:** Workspace displays all the variables that the user has defined, along with each variable's additional information such as calculated, minimum, maximum values and dimensions. Double-clicking on a variable name opens a new window which is its variable or matrix editor. This shows all the values associated with that variable. The extension for a workspace file is .mat.

## X. PROPOSED MODEL

In the existing mobile data offloading model to offload the process data from mobile to cloud platforms utilizes the CPU time estimation method to make the offloading decision. The offloading decision is taken on the basis of a threshold value. The existing scheme does not evaluate the communication cost for process data, which may put the communication overhead more than the local processing cost, which reduces the effort for energy efficiency. Also, the existing process cost calculation scheme is not efficient enough because it does not calculate the whole process tree's computational cost according the real time mobile energy consumption prediction. The average CPU workload computation does not reflect the real energy consumption cost on the mobile devices. Also the existing scheme does not propose any solution to resolve the cost calculation conflicts, which can be produced by the different cost calculation and ranking resulted by the HEFT and Cheapest cost methods for the task batch. Rapid increment in the multitasking possess a necessity of scheduling different processes in cloud computing. For scheduling process different scheduling algorithms are already proposed like DAG, HEFT, D-HEFT, EFT etc. HEFT can be also used in combination with DAG to attain lower complexity of scheduling of processes. This combination results in a process tree. Using this process tree, cost of each process can be found. Each process can be scheduled according to the found cost or the priority of the process. This efficient process Scheduling results in quicker response time, which in turns increase availability of resources to the customer. This paper describes work towards decreasing the complexity of existing budget-constrained state-of-the-art algorithms while maintaining the same budget level. The proposed method will use an effective cost calculation mechanism for the mobile devices based upon the earliest finish time (EFT) with DAG for process offloading proxy and remote execution management which will not be based upon the pre-stored mobile resource information, which will make it acceptable for the variety of mobile devices.

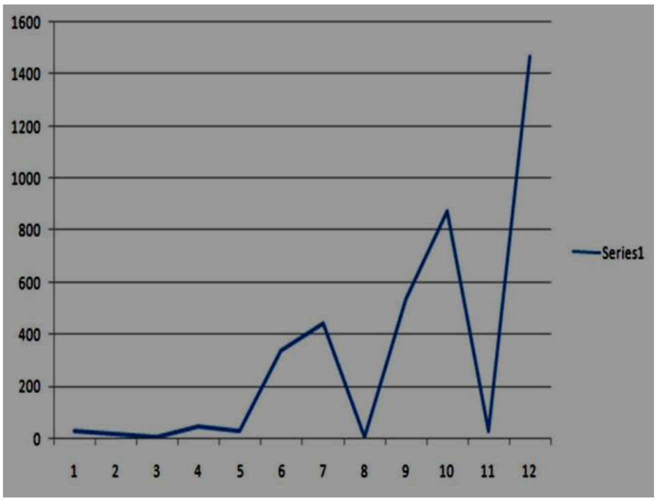
## XI. RESULT ANALYSIS

The results of the proposed model have been obtained in the form of various performance parameters. The performance evaluation has been performed on the basis of accuracy of the system to offloading the processes. The following table indicates the performance of the proposed model in terms of early finish time.

Table 11.1 The process list obtained from the application

| Index | Process Name |
|---|---|
| 1 | @bfun1 |
| 2 | @bfun2 |
| 3 | @bfun3 |
| 4 | @bfun4 |
| 5 | @bfun5 |
| 6 | 'App1' |
| 7 | 'App2' |
| 8 | 'App1' |
| 9 | 'App1' |
| 10 | 'App2' |
| 11 | 'App2' |
| 12 | 'App2' |

Fig: The Process Computation for all process CPU instruction set side.



## XII. CONCLUSION

The proposed model has been designed for the purpose of process offloading in the mobile cloud computing environment. The proposed model has been evaluated with the processes and process trees. The proposed model has been evaluated on the basis of results obtained in the form of time cost, CPU cost and Communication cost. The proposed model results have been found better than the existing models in handing the process trees and individual processes. The proposed model is highly acceptable in the real-time mobile cloud environments.

## XIII. REFERENCES

[1]. Xia, Feng, Fangwei Ding, Jie Li, Xiangjie Kong, Laurence T. Yang, and Jianhua Ma. "Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing." *Information Systems Frontiers* 16, no. 1 (2014): 95-111.

[2]. Wang, Lian, Yong Cui, Ivan Stojmenovic, Xiao Ma, and Jian Song. "Energy efficiency on location based applications in mobile cloud computing: a survey."*Computing* 96, no. 7 (2014): 569-585.

[3]. Ma, Xiao, Yong Cui, and Ivan Stojmenovic. "Energy efficiency on location based applications in mobile cloud computing: a survey." *Procedia Computer Science* 10 (2012): 577-584.

[4]. Nir, Manjinder, Ashraf Matrawy, and Marc St-Hilaire. "An energy optimizing scheduler for mobile cloud computing environments." In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pp. 404-409. IEEE, 2014.

[5]. Zhang, Weiwen, Yonggang Wen, and H-H. Chen. "Toward transcoding as a service: energy efficient offloading policy for green mobile cloud." *Network, IEEE* 28, no. 6 (2014): 67-73.

[6]. Ma, Xiaoqiang, Yuan Zhao, Lei Zhang, Haiyang Wang, and Limei Peng. "When mobile terminals meet the cloud: computation offloading as the bridge."*Network, IEEE* 27, no. (2013): 28-33.

[7]. Shiraz, Muhammad, and Abdullah Gani. "A lightweight active service migration framework for computational offloading in mobile cloud computing." *The Journal of Supercomputing* 68, no. 2 (2014): 978-995.

[8]. Flores, Huber, and Satish Srirama. "Adaptive code offloading for mobile cloud applications: Exploiting fuzzy sets and evidence-based learning." In *Proceeding of the fourth ACM workshop on Mobile cloud computing and srvices*, pp. 9-16. ACM, 2013.

[9]. Yang, Lei, Jiannong Cao, Yin Yuan, Tao Li, Andy Han, and Alvin Chan. "A framework for partitioning and execution of data stream applications in mobile cloud computing." *ACM SIGMETRICS Performance Evaluation Review* 40, no. 4 (2013): 23-32.

[10]. Shiraz, Muhammad, Ejaz Ahmed, Abdullah Gani, and Qi Han. "Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing."*The Journal of Supercomputing* 67, no. 1 (2014): 84-103.

[11]. Dileep bhandarkar on datacenter energy efficiency http://perspectives.mvdirona.com/2011/2/20/DileepBhandarkarOnDatacenterEnergyEfficiency.

[12]. The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.Self-service, prorated supercomputingfun,

[13]. http://open.blogs.nytimes.com/2007/11/01/selfservice-prorated-super-computing-fun/.Phase-change memory.

[14]. http://en.wikipedia.org/wiki/Phase-change_memory. Pharmaceuticals test-drive the cloud.

[15]. http://www.datacenterknowledge.com/archives/2009/05/26/pharmaceuticals-test-drive-thecloud/.

[16]. Solid state storage 101: An introduction to solid state storage. SNIA White Paper, January 2009.

[17]. Gelado, Isaac, John E. Stone, Javier Cabezas, Sanjay Patel, Nacho Navarro, and Wen-mei W. Hwu. "An asymmetric distributed shared memory model for heterogeneous parallel systems." In *ACM SIGARCH Computer Architecture News*, vol. 38, no. 1, pp. 347-358. ACM, 2010.

[18]. Virglio Almeida and Daniel Menasc. Cost-performance analysis of heterogeneity in superomputer architectures. In ACM/IEEE conference on Supercomputing,1990.

[19]. G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E.Harris. Reining in the outliers in mapreduce clusters using mantri. In Proceedings of the 10th Symposium on Operating System Design and Implemen- tation (OSDI '10), 2010.

[20]. Ganesh Ananthanarayanan, Srikanth Kandula, Albert Greenberg, Parveen Pa- tel, and Ion Stoica. What slows datacenter jobs and what to do about it. RADLab Retreat, Tahoe City, CA, January 2010.

[21]. Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Disk- locality in datacenter computing considered irrelevant.In Proceedings of the 13th USENIX conference on Hot topics in operating systems, HotOS'13, pages 12–12, Berkeley, CA, USA, 2011. USENIX Association.

[22]. David Andersen, Jason Franklin, Michael Kaminsky, Amar Phanishayee, Lawrence Tan, and Vijay Vasudevan. Fawn: A fast array of wimpy nodes.In 22nd ACM Symposium on Operating Systems Principles (SOSP), October2009.

[23]. Michael Armbrust and Gunho Lee.Evaluating amazon's ec2 as a research platform. http://radlab.cs.berkeley.edu/w/upload/0/0c/EC2_Performance_v2.ppt.pdf.

[24]. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H.Katz , Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud comput- ing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.URL http://www.eecs.berkeley.edu/Pubs/ TechRpts/2009/EECS-2009-28.html.

[25]. Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtu- alization. In Proceedings of the 19th ACM Symposium on Operating Systems Principles, pages 164– 177, Bolton Landing, NY, 2003.

[26]. Luiz Andr´e Barroso and Urs Holzle. The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2009.

[27]. O. Beaumont, A. Legrand, L. Marchal, and Y. Robert. Steady-state scheduling on heterogeneous clusters: why and how? In Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International, page 171, april 2004.

[28]. Peter Bod´ık, Rean Griffith, Charles Sutton, Armando Fox, Michael Jordan, and David Patterson. Statistical machine learning makes automatic control practical for internet datacenters. In HotCloud'09: Proceedings of the 2009 conference on Hot topics in cloud computing, pages 12–12, 2009.

[29]. Tracy D. Braun, Howad Jy Siegel, Noah Beck, Ladislau L. Blni, Muthucumaru Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D. Theys, Bin Yao, Debra Hensgen, and Richard F. Freund. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing, 61(6):810–837, Jun, 2001.

[30]. John S. Bucy, Jiri Schindler, Steven W. Schlosser, Gregory R. Ganger, and et al. The disksim simulation environment version 4.0 reference manual. Technical Re- port CMUPDL-08-101, Parallel Data aboratory, Carnegie Mellon University, Pittsburgh, PA, May 2008.

[31]. S´ebastien Cahon, Nordine Melab, and El-Ghazali Talbi. ParadisEO: A frame- work for the reusable design of parallel and distributed metaheuristics. Journal of Heuristics, 10(3):357–380, 2004.

[32]. Fangzhe Chang, Jennifer Ren, and Ramesh Viswanathan. Optimal resource allocation in clouds. Cloud Computing, IEEE International Conference on, pages 418–425, 2010.

[33]. Navraj Chohan, Claris Castillo, Mike preitzer, Malgorzata Steinder, Asser Tantawi, and Chandra Krintz. See spot run: Using spot instances for mapreduce workflows. In Workshop on Hot Topics in Cloud Computing (HotCloud '10),2010.

[34]. Condor Project. http://www.cs.wisc.edu/condor/.

[35]. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In Proceedings of the 6th Symposium on Operating System Design and Implementation (OSDI '04), pages 137–150, San Francisco, CA, December 2004.

[36]. Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In Symposium on Operating Systems Design and Implementation, 2004.

[37]. Mathijs den Burger, Thilo Kielmann, and Henri E. Bal. TOPOMON: A moni- toring tool for grid network topology. In Proceedings of the International Conference on Computational Science, pages 558–567, Amsterdam, Netherlands, April 2002.

[38]. A. Ganapathi, Yanpei Chen, A. Fox, R. Katz, and D. Patterson. Statistics- driven workload modeling for the cloud.In Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on, pages 87–92,march 2010.doi: 10.1109/ICDEW.2010.5452742.

[39]. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google File System. In ACM Symposium on Operating Systems Principles (SOSP), 2003.

[40]. Globus Toolkit. http://www.globus.org/.

[41]. P. Brighten Godfrey and Richard M. Karp. On the price of heterogeneity in parallel systems. In SPAA '06: Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures, pages 84–92, 2006.

[42]. David E Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[43]. Albert Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. VL2: A scalable and flexible data center network. In Proceedings of the ACM SIGCOMM 2009 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), pages 51–62, Barcelona, Spain, August 2009.

[44]. Albert Greenberg et al. VL2: A scalable and flexible data center network. In procsigcomm, pages 51–62, 2009.

[45]. Chona S. Guiang, Kent F. Milfeld, Avijit Purkayastha, and John R. Boisseau. Memory performance of dual-processor nodes: comparison of intel xeon and amd opteron memory subsystem architectures. In Proceedings for Cluster World Conference and Expo 2003, 2003.

[46]. Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. DCell: A scalable and fault-tolerant network structure for dat- acenters. In Proceedings of the ACM SIGCOMM 2008 Conference on Applica- tions, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM), pages 75–86, Seattle, WA, August 2008.

[47]. Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Chen Tian Yunfeng Shi, Yongguang Zhang, and Songwu Lu. BCube: A high performance, server-centric network architecture for modular data centers. In SIGCOMM, pages 63–74, Barcelona, Spain, August 2009.

[48]. H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. B. Cetin, and S. Babu. Starfish: A self-tuning system for big data analytics.In 5th Conference on Innovative Data Systems Research (CIDR '11), January 2011.

[49]. Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D.Joseph, Randy Katz, Scott Shenker, and Ion Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In USENIX Symposium on Network Systems Design and Implementation (NSDI, 2011.

[50]. O. H. Ibarra and C. E. Kim. Heuristic algorithms for scheduling independent tasks on nonidentical processors. Journal of the ACM, 24(2):280–289, 1977.

[51]. Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar, and Andrew Goldberg. Quincy: Fair scheduling for distributed computing clusters. In Proceedings of 22nd ACM Symposium on Operating Systems Principles (SOSP '09), Big Sky, MT, October 2009.

[52]. Michael Isard et al. Dryad: Dsitributed data-parallel programs from sequential building blocks. In ACM SIGOPS/EuroSys European Conference on Computer Systems, 2007.

[53]. Michael A. Iverson, Fusun Ozguner, and Gregory J. Follen. Run-time statistical estimation of task execution times for heterogeneous distributed computing. High-Performance Distributed Computing, International Symposium on, 0:263, 1996. ISSN 1082-8907.

[54]. Michael A. Iverson, Fu¨sun O¨ zgu¨ner, and Lee Potter. Statistical prediction of task execution times through analytic benchmarking for scheduling in a heterogeneous environment. IEEE Trans. Comput., 48(12):1374–1379, 1999.

[55]. Jeffrey M. Jaffe. An analysis of preemptive multiprocessor job scheduling. Mathematics of Operations Research, 5(3):pp. 415–421, 1980.

[56]. Joyent Smart Computing. http://www.joyent.com/.

[57]. Tim Kaldewey, Theodore M. Wong, Richard A. Golding, Anna Povzner, Scott A. Brandt, and Carlos Maltzahn. Virtualizing disk performance. In IEEE Real-Time and Embedded Technology and Applications Symposium, pages 319–330