Procedia Computer Science

Volume 51, 2015, Pages 1772–1781

ICCS 2015 International Conference On Computational Science

Towards Understanding Uncertainty in Cloud Computing Resource Provisioning

Andrei Tchernykh[1*†], Uwe Schwiegelsohn[2], Vassil Alexandrov[3‡], El-ghazali Talbi[4]

[1] *Computer Science Department, CICESE Research Center, 22830, Ensenada, Mexico*
chernykh@cicese.mx
[2] *Technische Universität Dortmund, 44221 Dortmund, Germany, uwe.schwiegelshohn@udo.edu*
[3] *ICREA-Barcelona Supercomputing Centre, Spain, vassil.alexandrov@bsc.es*
[4] *LIFL, University of Lille 1, France el-ghazali.talbi@lifl.fr*

Abstract
In spite of extensive research of uncertainty issues in different fields ranging from computational biology to decision making in economics, a study of uncertainty for cloud computing systems is limited. Most of works examine uncertainty phenomena in users' perceptions of the qualities, intentions and actions of cloud providers, privacy, security and availability. But the role of uncertainty in the resource and service provisioning, programming models, etc. have not yet been adequately addressed in the scientific literature. There are numerous types of uncertainties associated with cloud computing, and one should to account for aspects of uncertainty in assessing the efficient service provisioning. In this paper, we tackle the research question: what is the role of uncertainty in cloud computing service and resource provisioning? We review main sources of uncertainty, fundamental approaches for scheduling under uncertainty such as reactive, stochastic, fuzzy, robust, etc. We also discuss potentials of these approaches for scheduling cloud computing activities under uncertainty, and address methods for mitigating job execution time uncertainty in the resource provisioning.

Keywords: Cloud computing, Uncertainty, Resource provisioning, Optimization, Scheduling, Classification

1 Introduction

The cloud computing is widely acknowledged by practitioners and researchers as a valid solution for data storage and processing in both business and scientific computing. While having many

[*] Corresponding author

[†] Part of the work was done during a research stay of Andrei Tchernykh at INRIA Lille, France and Barcelona Supercomputing Center, Spain, and partially supported by CONACYT, México, grant no. 178415

[‡] Distinguished Visiting Professor, ITESM – Monterrey Tech, Mexico

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2015
© The Authors. Published by Elsevier B.V.

doi:10.1016/j.procs.2015.05.387

advantages cloud computing still has many drawbacks, especially in the areas of security, reliability, performance of both computing and communication, to list just a few. They are strengthened by the uncertainty, which accompanies all of these shortcoming. The vast majority of the research efforts in scheduling assumes complete information about the scheduling problem and a static deterministic execution environment. However, in the cloud computing, services and resources are subject to considerable uncertainty during provisioning. We argue that the uncertainty is the main hassle of cloud computing bringing additional challenges to end-users, resource providers, and brokering. They require waiving habitual computing paradigms, adapting current computing models to this evolution, and designing novel resource management strategies to handle uncertainty in an effective way.

There is a research on cloud computing examining the uncertainty phenomena in users' perceptions of the qualities, intentions and actions of cloud providers, privacy, security, availability, etc. among other aspects of cloud computing (Trenz et al., 2013). But still, the role of uncertainty in the resource and service provisioning, provider investment and operational cost, programming models, etc. have not yet been adequately addressed in the scientific literature.

There is a variety of types and sources of uncertainty. Table 1 describes some of them and briefly explain their impact on service provisioning: dynamic elasticity, dynamic performance changing, virtualization with loosely coupling applications to the infrastructure, resource provisioning time variation, inaccuracy of application runtimes estimation, variation of processing times and data transmission, workload uncertainty, processing time constraints (deadline, due date), effective bandwidth variation, and other phenomena.

Workload in such an environment can be changed dramatically. It is difficult to estimate runtime of jobs accurately, improve prediction by historical data, prediction correction, prediction fallback, etc. (Ramírez et al., 2011). The performance can be changed due to sharing of common resources with other VM. It is impossible to get exact knowledge about the system. Parameters like an effective processor speed, number of available processors, or actual bandwidth are changing in time. Elasticity has a higher repercussion on the quality of service, but adds a new factor of uncertainty.

Uncertainty may be presented in different components of the computational and communication process. Important questions are: how particular dynamic computation and communication characteristics affect the efficiency; how these characteristics can be used to mitigate uncertainty meeting desired QoS constraints; how corresponding optimization problem can be solved in efficient way; how to deliver scalable and robust cloud behavior under uncertainties and specific constraints, such as budgets, QoS, SLA, energy costs; etc. It is important to study different stochastic, adaptive, reactive, knowledge-free, etc. resource provisioning algorithms considering effective alternatives to known deterministic optimization technologies while keeping QoS.

In most existing solutions, it is assumed that behavior of VMs and services is predictable and stable in performance. On actual cloud infrastructures, these assumptions do not hold. While most providers guarantee a certain processor speed, memory capacity, and local storage for each provisioned VM, the actual performance is subject to the underlying physical hardware as well as the usage of shared resources by other VMs assigned to the same host machine. It is also true for communication infrastructure, where actual bandwidth is very dynamic and difficult to guarantee.

A pool of virtualized, dynamically scalable computing resources, storages, software, and services of cloud computing add a new dimension to the service delivering problem. The manner in which the service provisioning can be done depends not only on the service property and resources it requires, but also users who share resources at the same time. The management of cloud infrastructure is a challenging task. Reliability, security, quality of service, and cost-efficiency are important issues in these systems. Available cloud models do not adequately capture uncertainty, inhomogeneity and dynamic performance changes inherent to non-uniform and shared infrastructures. To gain better understanding of the consequences of a cloud computing uncertainty, we study resource and service provisioning problems related with existing cloud infrastructures such as hybrid federation of public, private and community.

		Sources of uncertainty															
		Data (variety, value)	Virtualization	Jobs arrival	Migration	Energy consumption	Fault tolerance	Scalability	Cost (dynamic pricing)	Resource availability	Elasticity	Consolidation	Communication	Replication	Cloud infrastructure	Elastic provisioning	Provisioning time
Parameters	Effective performance		●	●	●	●			●	●	●	●	●	●	●	●	●
	Effective bandwidth	●		●	●	●		●	●	●	●		●	●	●	●	●
	Processing time		●	●	●	●			●	●	●	●	●	●	●	●	●
	Available memory	●		●	●	●	●	●	●	●	●	●			●	●	●
	Number of processors		●	●		●		●	●	●	●	●			●	●	●
	Available storage	●			●	●	●	●	●	●	●				●	●	●
	Data transfer time	●			●	●			●						●	●	●
	Resource capacity		●	●			●	●	●		●				●	●	●
	Network capacity	●			●		●	●	●						●	●	●

Table 1. Cloud computing parameters and main sources of their uncertainty

2 Uncertainty

In spite of extensive research of uncertainty issues in different fields in the past decades ranging from physics, computational biology to decision making in economics and social sciences, a study of uncertainty for cloud computing systems is still not available. There are numerous types of uncertainties associated with cloud computing, and one ought to account for aspects of uncertainty in assessing the efficient service provisioning. Mitigating impact of uncertainty on the performance, reliability, safety, and robustness of cloud systems is rapidly growing research topic. Uncertainty analysis should become an essential part of design of resource and service provisioning strategies.

This paper presents our understanding of how to model cloud computing with uncertainty addressing resource provisioning in hybrid private-public cloud environment, dynamic self-adaptive distributed brokering, elastic clouds, and optimization of related problems to deliver robust resource management solutions, where the main objective is not to find an absolute optimum but rather solutions that behave good and insensitive to different uncertainties. High performance objectives could leads to too risky execution policies.

Uncertainty can be viewed as the difference between the available knowledge and the complete knowledge. It can be classified in several different ways according to their nature (Tychinsky, 2006): (1) The long-term uncertainty is due to the object is poorly understood and inadvertent factors can influence its behavior; (2) Retrospective uncertainty is due to the lack of information about the behavior of the object in the past; (3) Technical uncertainty is a consequence of the impossibility of predicting the exact results of decisions; (4) Stochastic uncertainty is a result of probabilistic (stochastic) nature of the studied processes and phenomena, where the following cases can be distinguished: there is a reliable statistical information; the situation is known to be stochastic, but the necessary statistical information to assess its probability characteristics is not available; a hypothesis on the stochastic nature requires verification; (5) Constraint uncertainty is due to partial or complete ignorance of the conditions under which the solutions have to be taken; (6) Participant uncertainty occurs in a situation of conflict of main stakeholders: cloud providers, users and administrators, where each side has own preferences, incomplete, inaccurate information about the motives and behavior of opposing sides; (7) Goal uncertainty is associated with conflicts and inability to select one goal in the decision or building multi objective optimization model. It addresses the problem of competing interests and multi-criteria choice of optimal decisions under uncertainty; (8) Condition uncertainty occurs when a failure or a complete lack of information about the conditions under which decisions are made; (9) Objective uncertainty occurs when there is no ambiguity when choosing solutions, there is more than one objective function to be optimized simultaneously, and there exists a possibly infinite number of Pareto optimal solutions.

These uncertainties can be grouped into: parameter (parametric) and system uncertainties.

Parameter uncertainties arise from the incomplete knowledge and variation of the parameters, for example, when data are inaccurate or not fully representative of the phenomenon of interest. They are generally estimated using statistical techniques and expressed probabilistically. Their analysis quantifies the effect of input random variables on model outputs. It is an integral part of reliability-based and robust design. The efficiency and accuracy of probabilistic uncertainty analysis is a trade-off issue. This type of uncertainty is not reducible since it is a property of the system itself.

System uncertainties arise from an incomplete understanding of the processes that control service provisioning, for example, when the conceptual model of the system used for service provisioning does not include all the relevant processes or relationships. It is reducible if more information is obtained. It can be modeled by probability theory, evidence theory, possibility theory, and fuzzy set.

Robust system synthesis minimizes the impact of uncertainties on the system performance. It has traditionally been performed by either a probabilistic approach or a worst case approach. Both approaches treat uncertainty as either random variables or interval variables. In reality, uncertainty can be a mixture of both. Monte Carlo simulation can be used to perform robustness assessment under an optimization framework. The probabilistic approach is considered as the most rigorous approach to uncertainty analysis and its mitigating due to its consistency with the theory of decision analysis.

3 Objective Functions

In this section, we present examples of objective functions of the scheduling in stochastic environment (Table 2) to and how to evaluate the quality of the solutions. Let job j must be processed for P_j units of time, where P_j is a random variable. Let $\mathbb{E}[\mathrm{P}_j]$ be the expected value of the processing time of job j, and p_j be a particular realization of P_j. We can assume that all random variables of processing times are stochastically independent and follow discrete probability distributions, and w.l.o.g. that P_j is integral value and that all release dates r_j and deadlines d_j are integral.

Expected total weighted completion time	$\mathbb{E}\left[\sum_{j\in J}\left(w_j\cdot \mathrm{C}_j\right)\right]=\sum_{j\in J}\left(w_j\cdot\mathbb{E}[\mathrm{C}_j]\right)$
Expected mean turnaround time	$\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[C_j-r_j\right]$
Expected mean waiting time	$\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\left[C_j-P_j-r_j\right]$
Expected mean bounded slowdown	$\frac{1}{n}\sum_{j=1}^{n}\frac{\mathbb{E}[C_j-r_j]}{\max\{10,\mathbb{E}[P_j]}$
Expected total weighted tardiness	$\sum_{j=1}^{n}\mathbb{E}\left[\mathrm{w}_j\cdot\max(P_j-d_j,0)\right]$
Expected makespan	$\mathbb{E}[C_{max}]=max\left(\mathbb{E}\left[C_j\right]\right)$

Table 2. Examples of the objective functions in stochastic environment

Following (Megow et al., 2005), a stochastic policy Π is a ρ-approximation (ρ-competitive), for $\rho\geq 1$, if for all problem instances I, $\mathbb{E}[\Pi(\mathrm{I})]\leq\rho\mathbb{E}[OPT(\mathrm{I})]$, where $\mathbb{E}[\Pi(\mathrm{I})]$ and $\mathbb{E}[OPT(\mathrm{I})]$ denote expected metric values obtained by Π and an optimal offline policy, respectively. The solution of a stochastic scheduling problem is not a schedule, but a so-called scheduling policy that makes scheduling decisions at time points t without information about the future, e.g. actual p_j of the jobs that have not yet been completed by time t. Let C_j and w_j be the completion time and weight (importance, priority) of job j, respectively. The goal is to minimize the objective function in expectation. These functions can be grouped into: regular objective functions, which are non-decreasing in job completion time such as total weighted completion time, total weighted number of tardy jobs, maximum lateness, and so on, and non-regular objective functions, such as expected earliness/tardiness, completion time variance, and general costs (Cai et al., 2014). These metrics are commonly used to express the objectives of different stakeholders (end-users, resource providers, and administrators). Robustness guarantees degradation (within the bounds of the tolerable variation in the system feature) despite fluctuations of the the environment and system parameters (Ali et al., 2003). It is also can be measured by standard deviation, differential entropy, etc. (Canon et al., 2010).

4 Related Work

4.1 Programming Uncertainty

Uncertainty understanding has to lead to discoveries in how to design cloud applications in efficient way. Most of cloud applications require availability of communication resources for information exchange between tasks, with databases or end users. However, providers might not know the quantity of data that can be managed, or quantity of computation required by tasks. For example, every time when a user requires a status of a bank account, it could take different time for its delivery.

Only few approaches take communication requirements into consideration and often in a highly abstract manner. Moreover, if applications are to utilize the Cloud, their execution environment is not known at development time - the number of available machines, their location, their capabilities, the network topology, and effective communication bandwidth cannot be known ahead. In general, an execution environment differs for each program/service invocation. To deal with this dynamics, either programmers must explicitly write adaptive programs or cloud software must deal with the uncertainty.

The user adaptive solutions are based on enormous programming effort. For an effective utilization of the Cloud, the programs must be decoupled from the execution environment. Programs should be developed for uniform and predictable virtual services, thus, simplifying their development. Cloud application model has to allow high level representation of computing and communication based on the nature of the problem, and independent of the executing environment. Mapping computation on machines, balancing the loads among different machines, removing unavailable machines from a computation, mapping communication tasks and balancing the communication loads among different links have transparently be provided by the runtime system.

(Kliazovich et al. 2013) propose new model CA-DAG of cloud computing applications by introducing communication awareness, which overcomes shortcomings of existing approaches and allows to mitigate uncertainty in more efficient way. It is based on a Directed Acyclic Graph, which along with computing vertices has separate vertices to represent communications. Such a representation allows making separate resource allocation decisions, assigning processors to handle computing jobs and network resources for information transmissions. The proposed communication-aware model creates space for optimization of many existing solutions to resource allocation as well as developing completely new scheduling schemes of improved efficiency. The program is represented by a DAG $G = (V, E, \omega, \varphi)$. The set of vertices $V = \{V_c, V_{comm}\}$ is composed of two non-overlapping subsets V_c and V_{comm}. The set $V_c \subseteq V$ represents computing tasks, and the set $V_{comm} \subset V$ represents communication tasks of the program.

A computing task $v_i^c \in V_c$ is described by a pair (I, D_c) with the number of instructions I (amount of work) that has to be executed within a specific deadline D_c. A communication task $v_i^{comm} \in V_{comm}$

is described by parameters $(\mathrm{S}, \mathrm{D}_{\mathrm{comm}})$, and defined as the amount of information S in bits that has to be successfully transmitted within a predefined deadline $\mathrm{D}_{\mathrm{comm}}$. Positive weights $\omega(\mathrm{v}_{\mathrm{i}}^{\mathrm{c}})$ and $\varphi(\mathrm{v}_{\mathrm{i}}^{\mathrm{comm}})$ represent the cost of computing at the node $\mathrm{v}_{\mathrm{i}}^{\mathrm{c}} \in \mathrm{V}_{\mathrm{c}}$, and cost of communication at the node $\mathrm{v}_{\mathrm{i}}^{\mathrm{comm}} \in \mathrm{V}_{\mathrm{comm}}$, respectively. The set of edges E consists of directed edges e_{ij} representing dependence between node $\mathrm{v}_{\mathrm{i}} \in \mathrm{V}$, and node $\mathrm{v}_{\mathrm{j}} \in \mathrm{V}$. It helps to define the execution order of tasks, which exchange no data. The main difference between communication vertices $\mathrm{V}_{\mathrm{comm}}$ and edges E is that $\mathrm{V}_{\mathrm{comm}}$ represents communication tasks occurred in the network, making them a subject to communication contention, significant delay, and link errors. The edge set E corresponds to the dependences between computing and communication tasks defining the order of their execution.

4.1 Resource Provisioning

A key dimension of scheduling policies concerns with how to map a set of tasks to a set of resources. Typically, there are two ways: static scheduling and dynamic scheduling. In the static approach, detailed information about job and processor characteristics, and network's topology characteristics are known in advance making possible to achieve a near optimal schedule for some problems. The static approach makes a schedule only when a task is ready (Rodriguez et al., 2003). Unfortunately, the performance of cloud resources is hard to predict, because these resources are not dedicated to one particular user, and, besides, there is no knowledge of network's topology. Furthermore, in general, due to the virtualization technique, it is impossible to get exact knowledge about the system. Effective characteristics are changing over the time. Therefore, providers are always searching how to improve the management of resources to ensure Quality of Service (QoS).

The shifting emphasis towards a service-oriented paradigm led to the adoption of SLAs as a very important concept. The use of SLAs is a fundamentally new approach for job scheduling. With this approach, schedulers are based on satisfying QoS constraints regardless uncertainty. The main idea is to provide different levels of service (SL), each addressing different set of customers to guarantee job delivery time depending on the SL. Based on the models in hard real-time scheduling, (Schwiegelshohn and Tchernykh, 2012) introduce a simple model for job allocation and scheduling, where each SL is described by a slack factor and a price for a processing time unit. If the provider accepts a job it is guaranteed to complete by its deadline. The authors theoretically analyze the single (SM) and parallel machine (PM) models subject to jobs with single (SSL) and multiple service levels (MSL). The analysis is based on the competitive factor, which is measured as the ratio between the income of the infrastructure provider obtained by the scheduling algorithm and the optimal income. Algorithms are based on the adaptation of the preemptive EDD (Earliest Due Date) algorithm for scheduling the jobs with deadlines.

To show the practicability and competitiveness of the algorithms, (Tchernykh et al., 2014, Lezama et al., 2013) conduct a study of their performance and derivatives using simulation. The authors take into account an important issue that is critical for practical adoption of the scheduling algorithms, the use of workloads based on real production traces of heterogeneous HPC systems.

4.2 Load Balancing

One of the possible technique to solve problems of the computing and communication imbalance associated with uncertainty is the load balancing that allows to improve resource allocation. For efficient load balancing, it is important to define: the notions of the system underload/overload; who and when initializes load balancing; number of jobs to be migrated; time slot used for migration; number of VMs chosen for migration, etc. It helps to achieve a high resources utilization and quality of service by efficient and fair allocations of computing resources.

Elastic load balancing algorithm distributes incoming traffic (VMs, requests, jobs) across multiples instances to achieve greater quality of service (González et al., 2013). It detects overloaded resources and automatically reroutes traffic to underloaded resources. If all nodes of the cloud are overloaded then it can automatically scale up its request handling capacity in response to incoming traffic. When the cloud is underloaded then it can scale down. Capacity can be increased or decreased in real time according to the computing and network resources consumed. Elasticity allows handling unpredictable workload and avoid overloading. The admissibility of resources, when only limited set of resources is chosen for a job execution (Tchernykh et al., 2010), should also be taken into account in load balancing strategies to avoid job overload and starvation. The job migration can cause a huge communication overhead. The admissible factor limits such an overhead avoiding sending jobs to farther nodes. The admissible factor takes into account static factors such as the distance; and dynamics factors e.g. actual bandwidth and the traffic on the network. These characteristics are not considered in most of recent works because they are hard to quantify and vary depending on the applications.

4.3 Adaptive Scheduling

The scheduling of jobs on multiprocessors is generally well understood and has been studied for decades. Many research results exist for different variations of this single system scheduling problem. Some of them provide theoretical insights while others give hints for the implementation of real systems. However, the adaptive scheduling problem has rarely been addressed so far. Unfortunately, it may result in inefficient resource allocation and bad power utilization (Tchernykh et al., 2009).

One of the structural reasons for the inefficiency in on-line job allocation is the occupation of large machines by jobs with small processor requirements causing highly parallel jobs to wait for their execution. To this end, (Tchernykh et al., 2008) introduce the admissible factor that parameterizes the availability of the sites for the job allocation. The main idea is to set job allocation constraints, and dynamically adapt them to cope with different workloads and Grid properties. First, the competitive factor of the adaptive on-line scheduling algorithm MLBa+PS with admissible job allocation that varies between 5 and infinity by changing the admissible factor was derived for specific workload characteristics. (Tchernykh et al., 2010) extended this result for a more general workload model with the competitive factor of 17.

(Tchernykh et al., 2012) present 3-approximation and 5-competitive algorithms named MLBa+PS and MCTa+PS for the case that all jobs fit to the smallest machine, while derive an approximation factor of 9 and a competitive factor of 11 for the general case. The authors consider a scheduling model with two stages. At the first stage, jobs are allocated to a suitable machine, while at the second stage, local scheduling is independently applied to each machine.

In a real scenario, the admissible factor can be dynamically adjusted in response to the changes in the configuration and/or the workload. To this end, the past workload and allocation results within a given time interval can be analyzed to determine an appropriate admissible factor. This time interval should be set according to the dynamics in the workload characteristics and in the configuration. One can iteratively approximate the optimal admissible factor.

4.4 Knowledge-free Approach

(Tchernykh et al., 2013) address non-preemptive scheduling problems on heterogeneous P2P grids, where resources are changing over time, and scheduling decisions are free from information of application characteristics. The authors consider a scheduling with task replications to overcome possible bad resource allocation in presence of uncertainty, and ensure good performance. They analyze energy consumption of job allocation strategies exploring the replication thresholds, and dynamic component deactivation. The main idea of the approach is to set replication thresholds, and dynamically adapt them to cope with different objective preferences, workloads, and Grid properties. The authors compare three groups of strategies: knowledge-free, speed-aware, and power-aware. First, they perform a joint analysis of two metrics considering their degradation in performance. Then, they provide two-objective optimization analysis based on the Pareto optimal set, and compare twenty algorithms in terms of Pareto dominance.

4.5 Scheduling with Uncertainty

In recent years, probability theory and statistical techniques are incorporated into the scheduling to treat uncertainties from different sources. A comprehensive survey in this area, main results and tendencies can be found in the book (Sotskov and Werner, 2014). The approaches that use stochastic and fuzzy methods, and important issues of robustness and stability of scheduling are discussed.

Uncertainty about the future is considered in two major frameworks: stochastic scheduling and online scheduling. Stochastic scheduling addresses problems in which the properties of tasks, e.g. processing times, due dates, and their arriving time are modelled as random variables, which exact values are not known until they arrived and are complete, respectively. Online scheduling is characterized by no knowledge of future jobs arriving. Decisions can be made each time when job is arrived. Only jobs that arrive before are known.

(Megow, 2005, Megow et al., 2006, and Vredeveld, 2012) consider a model for scheduling under uncertainty that combines online and stochastic scheduling. Jobs arrive over the time and there is no knowledge about future jobs. Job processing times are assumed to be stochastic. As soon as a job becomes known, the scheduler knows only the probability distribution of the processing time. The authors address stochastic online scheduling policies on a single and identical parallel machines to minimize the expected value of the weighted completion times of jobs. The authors present a constant performance ratio of 2 for preemptive online stochastic scheduling to minimize the sum of weighted completion times on identical parallel machines.

(Cai et al., 2011) prove the same bound of 2 for preemptive stochastic online scheduling problem on uniformly related machines with bounded speeds. (Cai et al., 2014) survey the main results on the problems with random processing times, due dates, machine breakdowns, considering different objective functions both regular, which are non-decreasing functions of job completion time, and non-regular such as expected weighted earliness/tardiness. The authors discuss performance and risk measurements other than expectation, variance and stochastic orders that impact on the quantity of scheduling algorithms.

Other class of scheduling problems with uncertain parameters is considered by (Kasperski et al., 2014). Parameters are represented as vectors with all possible values that parameters may have with no probability distribution. The performance is measured by Minmax and Minmax regret criteria.

Bi-objective analysis of robustness and stability of the scheduling under uncertainty is presented by (Gören et al., 2014). The authors consider total expected flow time and the total variance of job completion times as a robustness and stability measures, respectively.

As already discussed, cloud scheduling algorithms is generally split into an allocation part and a local execution part. At the first part, a suitable machine for each job is allocated using a given selection criterion. In such a scheme, prediction of job execution time and queue waiting times is important to increase resorce allocation efficiency.

Accurate job runtime prediction is a challenging problem. It is difficult to improve prediction by historical data, prediction correction, prediction fallback, etc. (Smith et al., 1998, Downey, 1997).

(Kianpisheh et al., 2012) use historical information and apply different machine learning techniques including linear and quadratic regression, decision trees, support vector machine and k-nearest neighborhood. (Smith et al., 1998) and (Ramirez et al., 2011) predict the runtimes using similarity of applications that have executed in the past. (Iverson et al., 1996) use a nonparametric regression technique, where the execution time estimate for a task is computed from past observations. (Ramírez et al., 2012) apply self-similarity and heavy-tails characteristics to create scalability models for high-performance clusters. The authors formulate resource allocation problem in presence of job runtime uncertainty, and propose novel adaptive allocation strategy named Pareto Fractal Flow Predictor (PFFP). They consider two steps for the runtime prediction. The first step models the site queuing process as an aggregation of a series of self-similar variables to predict the execution times of jobs in a queue. The second step predicts the remaining execution time of the current job in a site, using conditional probability and heavy-tails.

5 Conclusions

The uncertainty is an important issue that affects computing efficiency bringing additional challenges to scheduling problems. It requires designing novel resource management strategies to handle uncertainty in an effective way. We address areas such as resource provisioning, application execution, and communication provisioning. They have to provide the capability to dynamically allocate, manage resources in response to changing demand patterns in real-time, and dynamically adapt them to cope with different workloads and cloud properties to ensure QoS.

We review and classify cloud computing uncertainty, and discuss approaches to its mitigation. We highlight emerging trends, future directions in this field, role of uncertainty from providers, user and brokering perspectives; dynamic resource and service provisioning strategies; and programming, in presence of uncertainty. These challenges are of high complexity and keys to resource management decisions that users/resource providers are facing. Other important contributions is considering these problems in the light of their mapping to other challenges: stochastic scheduling, adaptive and knowledge free approaches, load balancing, etc.. Moreover, the challenge of defining a multi-criteria version of the problems is also discussed.

References

Cai et al. (2011) X. Cai, L. Zhang, Preemptive stochastic online scheduling on uniform machines with bounded speed ratios, 8th International Conference on Service Systems and Service Management, pp.1-4

Cai et al. (2014). X. Cai, X. Wu, L. Zhang, X. Zhou, Scheduling with Stochastic Approaches. In Sequencing and Scheduling with Inaccurate Data. Y. Sotskov, F. Werner (eds.). p. 3-45. Nova Science

Downey, A.B. (1997). Predicting Queue Times on Space-Sharing Parallel Computers. In: IPPS 1997 - 11th International Symposium on Parallel Processing, pp. 209–218

Gören et al. (2014). S. Gören, I. Sabuncuoglu: A Bi-criteria Approach to Scheduling in the Face of Uncertainty: Considering Robustness and Stability Simultaneously. In Sequencing and Scheduling with Inaccurate Data. Y. Sotskov, F. Werner (eds.). pp. 253-280. Nova Science Pub.

Iverson et al. (1996). M. Iverson, F. Ozguner, G. Follen, "Run-time statistical estimation of task execution times for heterogeneous distributed computing," Proceedings of 5th IEEE International Symposium on High Performance Distributed Computing, pp.263-270

Kasperski et al. (2014). A. Kasperski, P. Zielinski. Minmax (Regret) Scheduling Problems. In Sequencing and Scheduling with Inaccurate Data. Y. Sotskov F. Werner (eds.). pp. 159-210. Nova Science Pub.

Kianpisheh et al. (2012). S. Kianpisheh, S. Jalili, N. Charkari. Predicting Job Wait Time in Grid Environment by Applying Machine Learning Methods on Historical Information. International Journal of Grid and Distributed Computing. Vol. 5, 3

Kumar et al. (2013) R. Kumar, S. Vadhiyar. Identifying Quick Starters: Towards an Integrated Framework for Efficient Predictions of Queue Waiting Times of Batch Parallel Jobs. In Job Scheduling Strategies for Parallel Processing, LNCS Vol. 7698, pp 196-215

Megow et al. (2005). N. Megow, M. Uetz, T. Vredeveld. Models and algorithms for stochastic online scheduling, Mathematics of Operations Research 31(3): 513-525

Megow et al. (2006). N. Megow, T. Vredeveld. Approximation in Preemptive Stochastic Online Scheduling. Algorithms – ESA 2006, LNCS Vol. 4168, pp 516-527

Ramírez et al. (2011). J. Ramírez, A. Tchernykh, R. Yahyapour, U. Schwiegelshohn, A. Quezada, J. González, A. Hirales. Job Allocation Strategies with User Run Time Estimates for Online Scheduling in Hierarchical Grids. Journal of Grid Computing, 9: 95–116, Springer

Tchernykh et al. (2013). A. Tchernykh, J. Pecero, A. Barrondo, E. Schaeffer. Adaptive Energy Efficient Scheduling in Peer-to-Peer Desktop Grids. Future Generation Computer Systems. Vol. 36, pp. 209–220, Elsevier Science

Smith et al. (1998). W. Smith, I. Foster, V. Taylor, Predicting Application Run Times Using Historical Information. In: Feitelson, D.G., Rudolph, L. (eds.) IPPS-WS 1998, SPDP-WS 1998, JSSPP 1998. LNCS, Vol. 1459, pp. 122–142. Springer, Heidelberg

Sotskov and Werner (2014). Sequencing and Scheduling with Inaccurate Data. Editors: Yuri N. Sotskov and Frank Werner. Nova Science Pub, Applied Statistica Science, 442pp.

Trenz et al. (2013). M. Trenz, J. Huntgeburth, D. Veit, The role of uncertainty in cloud computing continuance: antecedents, mitigators, and consequences. Proceedings of the 21st European Conference on Information Systems, June 5-8, pp. 1-12

Vredeveld T. (2012). Stochastic online scheduling. Computer Science - Research and Development, Vol.27, 3, pp 181-187, Springer-Verlag

Ali et al. (2003). S. Ali, A. Maciejewski, H. Siegel, K. Jong-Kook, Definition of a robustness metric for resource allocation, Parallel and Distributed Processing Symposium, 2003. pp. 22-26

Canon et al. (2010). L. Canon, E. Jeannot, Evaluation and Optimization of the Robustness of DAG Schedules in Heterogeneous Environments, IEEE Transactions on Parallel and Distributed Systems, 21(4):532–546

Trenz et al. (2013). M. Trenz, J.C. Huntgeburth, D. Veit, The Role Of Uncertainty In Cloud Computing Continuance: Antecedents, Mitigators, and Consequences, ECIS, 147-147.

Tychinsky A. (2006). Innovation management of companies: Modern approaches, algorithms, experience. Taganrog: Taganrog Institute of Technology. On-line book http://www.aup.ru/books/m87/

González et al. (2013). J. González, R. Yahyapour, A. Tchernykh. Load Balancing for Parallel Computations with the Finite Element Method, Computación y Sistemas, Vol. 17, 3, pp. 299-3163

Tchernykh et al. (2010). A. Tchernykh, U. Schwiegelsohn, R. Yahyapour, N. Kuzjurin. Online Hierarchical Job Scheduling on Grids with Admissible Allocation, Journal of Scheduling, Vol. 13, 5, pp. 545—552. Springer-Verlag, Netherlands

Schwiegelshohn et al. (2012). U. Schwiegelshohn, A. Tchernykh. Online Scheduling for Cloud Computing and Different Service Levels. IPDPS 2012, IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, pp. 1067-1074, IEEE.

Kliazovich et al. (2013). D. Kliazovich, J. Pecero, A. Tchernykh, P. Bouvry, S. Khan, A. Zomaya. CA-DAG: Modeling Communication-Aware Applications for Scheduling in Cloud Computing Data Centers. IEEE 6th International Conference on Cloud Computing. p. 277– 284

Tchernykh et al (2014). A. Tchernykh, L. Lozano, U. Schwiegelshohn, P. Bouvry, J. Pecero, S. Nesmachnow. Bi-Objective Online Scheduling with Quality of Service for IaaS Clouds. The 2014 3rd IEEE International Conference on Cloud Networking. p. 307 – 312

Lezama et al. (2013). A. Lezama, A. Tchernykh, R. Yahyapour. Performance Evaluation of Infrastructure as a Service Clouds with SLA Constraints. Computación y Sistemas, Vol. 17, 3, pp. 401-411

Rodriguez et al. (2003). A. Rodriguez, A. Tchernykh, K. Ecker. Algorithms for Dynamic Scheduling of Unit Execution Time Tasks. EJOR, Elsevier Science, Vol.146, 2, p. 403-416

Tchernykh et al. (2009). A. Tchernykh, D. Trystram, C. Brizuela, I. Scherson. Idle Regulation in Non-Clairvoyant Scheduling of Parallel Jobs. Discrete Applied Mathematics 157, pp. 364–376, Elsevier Science