Journal of Computational Science xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability

Andrei Tchernykh[a,*], Uwe Schwiegelsohn[b], El-ghazali Talbi[c], Mikhail Babenko[d]

[a] Computer Science Department, CICESE Research Center, 22830, Ensenada, Mexico
[b] Technische Universität Dortmund, 44221 Dortmund, Germany
[c] CRISTAL, University of Lille 1 and INRIA, France
[d] North-Caucasus Federal University, Stavropol, Russia

ARTICLE INFO

Article history:
Received 1 July 2016
Received in revised form 15 October 2016
Accepted 21 November 2016
Available online xxx

Keywords:
Cloud computing
Uncertainty
Resource provisioning
Optimization
Scheduling
Reliability
Privacy

ABSTRACT

An extensive research has led to a general understanding of uncertainty issues in different fields ranging from computational biology to decision making in economics. However, a study of uncertainty on large scale computing systems and cloud computing systems is limited. Most of works examine uncertainty phenomena in users' perceptions of the qualities, intentions and actions of cloud providers. In this paper, we discuss the role of uncertainty in the resource and service provisioning, privacy, etc. especially, in the presence of the risks of confidentiality, integrity, and availability. We review sources of uncertainty, and fundamental approaches for scheduling under uncertainty. We also discuss potentials of these approaches, and address methods for mitigating the risks of confidentiality, integrity, and availability associated with the loss of information, denial of access for a long time, and information leakage.

1. Introduction

Cloud technologies are widely used in the construction of IT infrastructure of business, academic, government, and people as a valid solution for data storage and processing. While having many advantages they still have many drawbacks, especially in the areas of security, reliability, performance of both computing and communication, to list just a few. The transition to big data and exascale also pose numerous unavoidable scientific and technological challenges.

In the cloud computing, services and resources are subject to considerable uncertainty during provisioning. Uncertainty may be presented in different components of the computational, communication, and storing process. It requires waiving habitual computing paradigms, adapting current computing models to this evolution, and designing novel resource management strategies to handle uncertainty in an effective way.

The management of cloud infrastructure is a challenging task. Reliability, security, Quality of Service (QoS), performance stability, 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 ones.

Extensive research examines the uncertainty phenomena in users' perceptions of the qualities, intentions and actions of cloud providers, etc. among other aspects of cloud computing (Trenz et al.) [1]. But still, the role of uncertainty in the resource and service provisioning, provider investment, operational cost, programming models, mitigating risks of confidentiality, integrity, and availability etc. have not yet been adequately addressed in the scientific literature.

In this paper, we discuss two main topics: how to provide reliability, safety and privacy of information, and how to deliver scalable and robust cloud behavior under uncertainties and specific

☆A preliminary reduced version of this article appeared in Proceedings of the SPU'2015 - Solving Problems with Uncertainties, in conjunction with The 15th International Conference on Computational Science (ICCS 2015), Reykjavík, Iceland, June 1–3, 2015. Procedia Computer Science, Elsevier, Vol. 51, Pages 1772–1781, 2015, DOI: http://dx.doi.org/10.1016/j.procs.2015.05.387.

[*] Corresponding author.

E-mail addresses: chernykh@cicese.mx (A. Tchernykh), uwe.schwiegelshohn@udo.edu (U. Schwiegelsohn), el-ghazali.talbi@univ-lille1.fr (E.-g. Talbi), mgbabenko@ncfu.ru (M. Babenko).

http://dx.doi.org/10.1016/j.jocs.2016.11.011
1877-7503/© 2016 Elsevier B.V. All rights reserved.

constraints, such as budgets, QoS, SLA (Service-Level Agreement), energy costs, availability, etc.

Reliability, safety and privacy. As more users use cloud technologies for building IT-infrastructure, reliability, safety and privacy become crucial for both providers and consumers.

Preservation of confidentiality interpreted as a limited access to information, integrity as the assurance that the information is trustworthy and accurate, and availability as a guarantee of reliable access to the information by authorized people are three most crucial components of cloud computing.

Cloud-based services can crash just like any other type of technology. For example, access to information Amazon users has been limited for a long time due to distributed denial-of-service (DDoS) attacks in 2009. In 2013, a series of cloud outages are reported for Amazon, Microsoft and Google. Technical failures and data loss due to power outages are reported by Amazon, Dropbox, Microsoft, Google, and Yandex Disk. In the first quarter of 2014, Dropbox has experienced service outages twice. Bankruptcy is imposed for cloud storage company Nirvanix in 2013.

Common methods of ensuring confidentiality are data encryption, user identity documents (IDs), passwords, and other ways of authentication through cards, retina scans, voice recognition, and fingerprints. Other options include security tokens, key fobs tokens, etc.

Integrity involves maintaining the consistency, accuracy, and trustworthiness of information, so that they are not changed and altered by unauthorized people.

Service availability depends on the robustness of the hardware, hardware repairs and maintaining a correctly functioning operating system environment, system upgrades, preventing the occurrence of bottlenecks, etc. Redundancy, failover, redundant array of independent disks (RAID) etc. can mitigate consequences when hardware failures occur.

An uncertainty in these areas demands the companies be smart providing solutions preventing losing any data in the long run even if the service is offline during the time.

Performance, scalability and robustness. The vast majority of the research efforts in scheduling assumes complete information about the scheduling problem and a static deterministic reliable execution and storage environments.

In (Tchernykh et al.) [2], we show a variety of types and sources of uncertainty: 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 (Table 1).

The performance can be changed due to sharing of common resources with other virtual machines (VMs). It is impossible to get exact knowledge about the computer system. Parameters like an effective processor speed, number of available processors, or actual bandwidth are changing over time. Elasticity has a higher repercussion on the QoS, but adds a new factor of uncertainty. It is difficult to estimate runtime of jobs accurately (Ramírez et al.) [3].

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

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

		Sources of uncertainty															
		Data (variety, value)	Virtualization	Jobs arrival	Migration	Energy consumption	Fault tolerance	Scalability	Cost (dynamic pricing)	Resource-availability	Elasticity	Consolid-ation	Communi-cation	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	●			●		●	●	●						●	●	●

Please cite this article in press as: A. Tchernykh, et al., Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability, J. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.jocs.2016.11.011

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 growing number of scientific scalable applications require resources at the exascale. This demands increased scope for optimization and uncertainty quantification. Uncertainty analysis is an important tool for tuning application parameters making them adaptive to configuration and environment changes to take advantage of the vast amount of computational resources and extreme available concurrency.

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) [4]:

(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 modelled 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.

Robustness guarantees degradation (within the bounds of the tolerable variation in the system feature) despite fluctuations of the environment and system parameters (Ali et al.) [5]. It is also can be measured by standard deviation, differential entropy, etc. (Canon et al.) [6].

Stability and consistency have became a major issue for exascale systems gathered from several millions of CPU cores running up to a billion of threads and VMs, huge intercommunication infrastructure, etc. Numerical treatment of such stochastic systems is not simple. Stochastic methods retain all of the complexity of methods for deterministic problems and have additional requirements for capturing the probabilistic features of the system.

Uncertainty quantification including uncertainty characterization, propagation, parameter estimation, model calibration, error estimation, and other probabilistic approaches aim to address these challenges. Several well-known methods can be used in exascale systems to propagate uncertainty: stochastic modeling, scenario modeling, fuzzy data sets, interval calculations, analytical uncertainty propagation, etc.

3. Related work

3.1. Towards secure cloud computing

Information security assumes defending information from unauthorized access, use, disclosure, disruption, modification, etc. Important research and industrial streams of cloud computing are to design a secure and fault tolerant multi-cloud environment, where confidentiality, integrity, and availability are not violated in the presence of the deliberate threats, accidental threats, and failures (Srisakthi and Shanthi) [7].

Confidentiality considers a set of rules and restrictions that limits access to certain types of information, and support the user data as secret, private, and not viewed even by the cloud service provider.

Integrity supports a complete or whole data structure as a fundamental concept of information security. The information stored in the cloud should not be changed by any entity other than the owner.

Availability implies that the service is available to the user all time without disruptions.

Accountability implements strategies to protect data observed by all users who process the data, irrespective of where that processing occurs.

Privacy implies the freedom from unauthorized access to the user information.

Environmental threats include natural disasters, fires, floods, technological accidents and other events that do not depend on the human. To minimize the consequences of these threats one can use error correction codes for distributed data storage, where each piece of information is stored in various cloud providers.

Deliberate threats include unauthorized access to the information, interception, falsification, forgery, hacker attacks, etc. Cloud Security Alliance states that, in recent years, the number of unauthorized accesses to the information processed and stored in the clouds is dramatically increased (Hubbard and Sutton) [8]. The simultaneous use of cryptographic protocol and error correction codes can be used to reduce this risk.

Accidental threats include user errors, carelessness, curiosity, etc. The information control and protection system based on the proactive concept can be used. The proactive concept includes the simultaneous use of weighted secret sharing scheme based on Residue Number System (RNS), encryption keys, and checksums for monitoring obtained results.

Uncertainty pervades our attempts to organize and process data. Big data paradigm adds a new complexity dimension to the problem. The huge volume of information can be safe guarded only in distributed environment. Data backup copies in geographically distributed locations, and disaster recovery plans can multiply the already high costs.

There are numerous security and reliability issues for cloud computing as it encompasses many technologies. In Section 4, we discuss an approach that minimizes the risk of the above mentioned threats and maximize productivity using RNS.

3.2. 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 to be transparently provided by the runtime system.

Kliazovich et al. [9] propose new Communication-Aware Directed Acyclic Graph (CA-DAG) model 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 $G = \left(V, E, \omega, \phi\right)$. The set of vertices $V = \left\{V_c, V_{comm}\right\}$ is composed of two non-overlapping subsets V_c and V_{comm}. The set $V_c V$ represents computing tasks, and the set $V_{comm} V$ 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 (S, D_{comm}), and defined as the amount of information S in bits that has to be successfully transmitted within a predefined deadline D_{comm}. Positive weights $\omega\left(v_i^c\right)$ and $\phi\left(v_i^{comm}\right)$ represent the cost of computing at the node $v_i^c \in V_c$, and cost of communication at the node $v_i^{comm} \in V_{comm}$, respectively. The set of edges E consists of directed edges e_{ij} representing dependence between node $v_i \in V$, and node $v_j \in V$. It helps to define the execution order of tasks, which exchange no data. The main difference between communication vertices V_{comm} and edges E is that V_{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.

3.3. 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.) [10]. 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 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 of 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 et al. [11] 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. [12], and Lezama et al. [13] 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.

3.4. 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 QoS by efficient and fair allocations of computing resources.

Elastic load balancing algorithm distributes incoming traffic (VMs, requests, jobs) across multiples instances to achieve greater QoS (González et al.) [14]. 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.) [15], 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.

3.5. Adaptive and knowledge-free approach

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.) [16].

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.) [16] 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.) [15] extended this result for a more general workload model with the competitive factor of 17.

Quezada-Pina et al. [17] 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.

Tchernykh et al. [18] address non-preemptive scheduling problems on heterogeneous Peer-to-Peer (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.

3.6. 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) [19]. 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 has arrived.

Table 2 presents examples of objective functions of the scheduling in stochastic environment and how to evaluate the quality of the solutions. These metrics are commonly used to express the objectives of different stakeholders (end-users, resource providers, and administrators).

Table 2
Examples of the objective functions in stochastic environment.

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

Let job j be processed by P_j units of time, where P_j is a random variable. Let $\left[P_j\right]$ 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.

Following (Megow et al.) [20], a stochastic policy Π is a ρ-approximation (ρ-competitive), for $\rho \geq 1$, if for all problem instances I, $[(\mathrm{I})] \leq \rho\,[OPT(\mathrm{I})]$, where $[(\mathrm{I})]$ and $[OPT(\mathrm{I})]$ 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.) [21].

Megow et al. [20], Megow et al. [22], and Vredeveld [23] 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.

Cat et al. [24] prove the same bound of 2 for preemptive stochastic online scheduling problem on uniformly related machines with bounded speeds. Cai et al. [21] 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. [25]. 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. [26]. 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.) [27], (Downey) [28].

Kianpisheh et al. [29] 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. [27] and Ramirez et al. [3] predict the runtimes using similarity of applications that have executed in the past. Iverson et al. [30] use a nonparametric regression technique, where the execution time estimate for a task is computed from past observations. Ramírez et al. [31] 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.

4. Reliability and privacy under uncertainty

In order to increase reliability and confidentiality of the data processing and storing, six basic approaches are applied: data replication, secret sharing schemes, redundant residue number system, erasure code, regenerating code, and homomorphic encryption.

4.1. Data replication

The main advantages are high reliability and possibility to process the data in distributed fashion. The drawbacks are dramatic growth in data volumes and need to protect each replication (Ghemawat et al.) [32]; (Abu-Libdeh et al.) [33].

4.2. Secret sharing schemes (SSS)

SSS such as Shamir, Blackly, etc. are methods by which a dealer distributes shares to recipients such that only authorized subsets of recipients can reconstruct the secret. They are important tools in cryptography and used in many secure protocols, e.g., general protocol for multiparty computation, Byzantine agreement, threshold cryptography, access control, attribute-based encryption, and generalized oblivious transfer. They allow to build secure distributed storage systems (Gomathisankaran et al.) [34]. However, if SSS is not homomorphic, computation over encrypted data is impossible or complex. For the construction of a fully homomorphic encryption, (Asmuth et al.) [35] and (Mignotte, M.) [36] propose schemes based on RNS. The RNS allows to build a fully homomorphic ciphers due to the properties of the parallelism of arithmetic operations, which improve performance and develop a homomorphic encryption that is asymptotically perfect and balanced, depending on the task.

Redundant Residue Number System (RRNS). In this system, original number is represented as residues with respect to a moduli set. Thus the number will be split in-to some smaller numbers which are independent and operations can be accomplished on them separately and concurrently which makes the computations simpler and much faster. Redundancy of residues allows to build reliable data processing system with multiple error detection and correction (Chessa et al.) [37], (Celesti et al.) [38]. According to RRNS property, if the number of control modules is r, then the system can detect r and correct $r-1$ errors. For error isolation and correction, projection methods are used, where the number of calculated projections grows exponentially depending on the value of r. As a result RRNS is impractical without significant optimization.

4.3. Erasure code (EC)

EC is an error correction code, which transforms a message of k symbols into a longer message with n symbols such that the original message can be recovered from a subset of the n symbols. Lin et al. [39] propose an effective implementation of erasure code with $O(L \cdot \log_2 L)$ complexity, where L is the length of the code. Since erasure code is not homomorphic, it is suitable for building reliable distributed data storage system but does not allow efficient data processing. For erasure coded systems, a common practice is to generate another encoded block instead of repairing a fault node to reconstruct the whole encoded data object. Dimakis et al. [40] show that erasure code is sub-optimal.

4.4. Regenerating code (RC)

RCs are a class of codes proposed for providing reliability of data and efficient repair (rebuild) of lost encoded fragments in distributed storage systems. They can significantly reduce the total traffic required for repairing, called repair-bandwidth, and obtain reasonable tradeoffs between storage and repair bandwidth (Dimakis et al.) [40]. For its effective implementation, pseudorandom number generator with special properties is essential (Liu et al.) [41]. However, it does not allow to perform high-performance computing, because it is not homomorphic to addition, subtraction and multiplication (Chen et al.) [42].

4.5. Homomorphic encryption (HE)

HE is an encryption that allows to carry out computations on ciphers generating an encrypted result which, when decrypted, matches the result of operations performed on the original numbers. HE systems are proposed by Rivest, Edelman and Dertuzo (Rivest et al.) [43]. They are based on an exponentiation operation and RSA function (public-key cryptosystem) with additive and multiplicative homomorphic ciphers, respectively. They are fully homomorphic, when cipher is homomorphic relatively to two arithmetic operations simultaneously. Using these ideas and approaches for the construction of a fully homomorphic encryption, Gentry [44] proposes a wide range of approaches for building modified fully homomorphic ciphers and solving underlying performance problems. An alternative approach based on the matrix polynomials for the construction of a fully homomorphic scheme is proposed by Trepacheva et al. [45]. It solves main problems of existing homomorphic ciphers, which include low performance and great information redundancy. However, in practice, encryption schemes based on polynomial matrix are not compact and cryptosecure (Rivest et al.) [43], (Gomathisankaran et al.) [34].

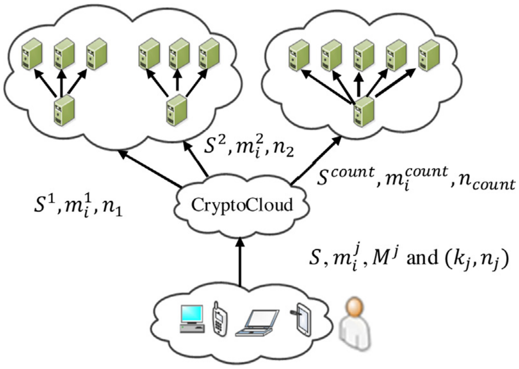

S^2, m_i^2, n_2
$S^{count}, m_i^{count}, n_{count}$
S^1, m_i^1, n_1
CryptoCloud
S, m_i^j, M^j and (k_j, n_j)

Fig. 1. Multilevel data encryption.

4.6. Clouds with reliability and safety

Six mentioned above approaches cannot ensure the security, high performance of data storage and processing in cloud systems independently.

SSS, RRNS, and fully HE based on RNS can solve this problem in a single method. To store and process data, a multilevel approach can be used (Fig. 1).

First, the user introduces a noise to the data D using formula $\mathrm{S} = \mathrm{key} \cdot \mathrm{seed} + D$, where the key is composite number equal to the product of security parameters $\mathrm{key} = \Pi_{\mathrm{i}=1}^{\mathrm{count}} \mathrm{key}_{\mathrm{i}}$ (Asmuth et al.) [35]. Then, he divides and distributes the noisy data with the parameters m_i^j, M^j, and $\left(k_j, n_j\right)$ of SSS based on RNS to cloud service providers. Where, m_i^j is i-th RNS moduli for the j-th cloud, $M^j = \Pi_{i=1}^{k_j} m_i^j$ is RNS dynamic range of the of j-th cloud, and $\left(k_j, n_j\right)$ are secret sharing parameters. S^j is data projection. In the last step, each provider independently processes data encryption for storing and processing encrypted data.

The user protects his data from the risks of violation of confidentiality, integrity, and availability by encrypting data using the HE function, and performing computations over the encrypted data (Gomathisankaran et al.) [34].

On the other hand, SSS based on RNS can detect and correct errors. To minimize data processing time, we perform load balancing among virtual machines with the use of multi-level RNS. It allows to check the correctness of the result and recover data in case of failures, loss of several data projections, and denial of access to one of the cloud providers. A necessary and sufficient condition for recovery of the data from projections is that the number of projections is greater than or equal to a threshold value (Mignotte) [36].

The adequate selection of parameters of RNS error correction codes, and cryptographic primitive of SSS allow to minimize computation time and obtain correct results even in case of ambiguous technical failure or disaster.

To decode information from RNS to original form, Chinese Remainder Theorem (CRT) is applied. The implementation of the CRT requires $O(lg^2 M)$ bit operations (Bach et al.) [46]. If size of the data M is large enough, it is feasible to use an algorithm based on recursive pairing, which allows to reduce the complexity to a linear (Wang et al.) [47]. RNS allows to construct a distributed storage system for big data, when since it maps a large number X to a tuple of small numbers $(x_1, x_2, \ldots, x_n)$, stored by different providers, where $x_1 = |X|_{m_1}, x_2 = |X|_{m_2}, \ldots, x_n = |X|_{m_n}$.

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 also discuss the role of uncertainty in privacy, review its sources, and fundamental approaches for mitigating the risks of confidentiality, integrity, and availability associated with the loss of information, denial of access for a long time, interruptions in connections, and information leakage. These approaches include mechanisms for data replication, backup copies, secret sharing, RRNS, EC, RC, homomorphic data encryption for storage and processing. To prevent data loss from such occurrences, downtime and malicious actions such as DoS attacks and network intrusions, security equipment and software such as firewalls can be used.

We show that current approaches for mitigating the risks of confidentiality, integrity, and availability will not work well in extreme scale cloud computing and in the future Exascale systems. The difficult challenge is to find new approaches, possibility radically disruptive. To this end, we discuss systems based on SSS, RRNS and HE in modular code, and distributed algorithms of the data encryption for storage and processing.

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.

Acknowledgement

Part of the work was supported by CONACYT, México, grant no.178415 and a grant of the President of the Russian Federation for the young scientists SP-1215.2016.5.

References

[1] M. Trenz, J.C. Huntgeburth, D. Veit, The role of uncertainty In cloud computing continuance: antecedents, mitigators, and consequences, Proceedings of the 21st European Conference on Information Systems (2013) 147.
[2] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, E.-G. Talbi, Towards understanding uncertainty in cloud computing resource provisioning, Proc. Comput. Sci. 51 (2015) 1772–1781.
[3] J. Ramirez, A. Tchernykh, R. Yahyapour, U. Schwiegelshohn, A. Quezada, J. Gonzalez, A. Hirales, Job allocation strategies with user run time estimates for online scheduling in hierarchical grids, J. Grid Comput. 9 (2011) 95–116.
[4] A. Tychinsky, Innovation Management of Companies: Modern Approaches, Algorithms, Experience, 2006 (Online]. Available: http://www.aup.ru/books/m87/ [accessed 29 June 2016).
[5] S. Ali, A. Maciejewski, H. Siegel, K. Jong-Kook, Definition of a robustness metric for resource allocation, Parallel and Distributed Processing Symposium (2003) 22–26.
[6] L. Canon, E. Jeannot, Evaluation and optimization of the robustness of DAG schedules in heterogeneous environments, IEEE Trans. Parallel Distrib. 21 (4) (2010) 532–546.
[7] S. Srisakthi, A.P. Shanthi, Towards the design of a secure and fault tolerant cloud storage in a multi-Cloud environment, Inf. Secur. J.: Global Perspect. 24 (4–6) (2015) 109–117.
[8] Top Threats to Cloud Computing v1.0. Cloud Security Alliance, 2010 (Online]. Available: https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf [accessed 29 June 2016).
[9] 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 (2013) 277–284.
[10] A. Rodriguez, A. Tchernykh, K. Ecker, Algorithms for dynamic scheduling of unit execution time tasks, Eur. J. Oper. Res. 146 (2) (2003) 403–416.
[11] U. Schwiegelshohn, A. Tchernykh, Online scheduling for cloud computing and different service levels, IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (2012) 1067–1074.
[12] A. Tchernykh, L. Lozano, U. Schwiegelshohn, P. Bouvry, J. Pecero, S. Nesmachnow, Bi-objective online scheduling with quality of service for IaaS clouds, 3rd IEEE International Conference on Cloud Networking (2014) 307–312.
[13] A. Lezama, A. Tchernykh, R. Yahyapour, Performance evaluation of infrastructure as a service clouds with SLA constraints, Computacion y Sistemas 17 (3) (2013) 401–411.
[14] J. Gonzalez, R. Yahyapour, A. Tchernykh, Load balancing for parallel computations with the finite element method, Computacion y Sistemas 17 (3) (2013) 299–316.
[15] A. Tchernykh, U. Schwiegelsohn, R. Yahyapour, N. Kuzjurin, Online hierarchical job scheduling on grids with admissible allocation, J. Sched. 13 (5) (2010) 545–552.
[16] A. Tchernykh, D. Trystram, C. Brizuela, I. Scherson, Idle regulation in non-clairvoyant scheduling of parallel jobs, Discrete Appl. Math 157 (2009) 364–376.
[17] A. Quezada-Pina, A. Tchernykh, J.L. González-García, A. Hirales-Carbajal, J.M. Ramírez-Alcaraz, U. Schwiegelshohn, Y. Ramin, V. Miranda-López, Adaptive parallel job scheduling with resource admissible allocation on two-level hierarchical grids, Future Gener. Comp. Syst. 28 (7) (2012) 965–976.
[18] A. Tchernykh, J. Pecero, A. Barrondo, E. Schaeffer, Adaptive energy efficient scheduling in peer-to-Peer desktop grids, Future Gener. Comp. Syst. 36 (2013) 209–220.
[19] I.N. Sotskov, F. Werner, Sequencing and scheduling with inaccurate data, in: Applied Statistica Science, Nova Science Pub, Minsk, 2014.
[20] N. Megow, M. Uetz, T. Vredeveld, Models and algorithms for stochastic online scheduling, Math Oper. Res. 31 (3) (2005) 513–525.
[21] X. Cai, X. Wu, L. Zhang, X. Zhou, Scheduling with stochastic approaches, in: Y. Sotskov, F. Werner (Eds.), Sequencing and Scheduling with Inaccurate Data, Nova Science Pub, Minsk, 2014, pp. 3–45.
[22] N. Megow, T. Vredeveld, Approximation in preemptive stochastic online scheduling, LNCS 4168 (2006) 516–527.
[23] T. Vredeveld, Stochastic online scheduling, Comput. Sci. Res. Dev. 27 (3) (2012) 181–187.
[24] X. Cat, L. Zhang, Preemptive stochastic online scheduling on uniform machines with bounded speed ratios, 8th International Conference on Service Systems and Service Management (2011) 1–4.
[25] A. Kasperski, P. Zielinski, Minmax (Regret) scheduling problems, in: Y. Sotskov, F. Werner (Eds.), Sequencing and Scheduling with Inaccurate Data, Nova Science Pub, Minsk, 2014, pp. 159–210.
[26] S. Goren, I. Sabuncuoglu, A Bi-criteria approach to scheduling in the face of uncertainty: considering robustness and stability simultaneously, in: Y. Sotskov, F. Werner (Eds.), Sequencing and Scheduling with Inaccurate Data, Nova Science Pub, Minsk, 2014, pp. 253–280.
[27] W. Smith, I. Foster, V. Taylor, Predicting application run times using historical information, LNCS 1459 (2006) 122–142.
[28] A.B. Downey, Predicting queue times on space-Sharing parallel computers, IPPS 1997–11th International Symposium on Parallel Processing (1997) 209–218.
[29] S. Kianpisheh, S. Jalili, N. Charkari, Predicting job wait time in grid environment by applying machine learning methods on historical information, Int. J. Grid Distrib. Comput. 5 (3) (2012) 11–22.
[30] M.A. Iverson, F. Ozguner, G.J. Follen, Run-time statistical estimation of task execution times for heterogeneous distributed computing, Proceedings of 5th IEEE International Symposium on High Performance Distributed Computing (1996) 263–270.
[31] R.V. Ramirez-Velarde, R.M. Rodriguez-Dagnino, From commodity computers to high-performance environments: scalability analysis using self-similarity, large deviations and heavy-tails, Concurr. Comp.-Pract. E. 22 (11) (2010) 1494–1515.
[32] S. Ghemawat, H. Gobioff, S.T. Leung, The google file system, ACM SIGOPS Op. Syst. Rev. 37 (5) (2003) 29–43.
[33] H. Abu-Libdeh, L. Princehouse, H. Weatherspoon, RACS: a case for cloud storage diversity, Proceedings of the 1st ACM Symposium on Cloud Computing (2010) 229–240.
[34] M. Gomathisankaran, A. Tyagi, K. Namuduri, HORNS: a homomorphic encryption scheme for cloud computing using residue number system, 45st Annual Conference on Information Sciences and Systems (2011) 1–5.
[35] C. Asmuth, J. Bloom, A modular approach to key safeguarding, IEEE Trans. Inform. Theory 30 (2) (1983) 208–210.
[36] M. Mignotte, How to share a secret, Workshop on Cryptography (1982) 371–375.
[37] S. Chessa, P. Maestrini, Dependable and secure data storage and retrieval in mobile, wireless networks, Proceedings. 2003 International Conference on Dependable Systems and Networks (2003) 207–216.

[38] A. Celesti, M. Fazio, M. Villari, A. Puliafito, Adding long-term availability, obfuscation, and encryption to multi-cloud storage systems, J. Netw. Comput. Appl. 59 (2016) 208–218.

[39] S.J. Lin, W.H. Chung, Y.S. Han, Novel polynomial basis and its application to reed-solomon erasure codes, An. S Fdn. Co. (2014) 316–325.

[40] A.G. Dimakis, P.B. Godfrey, Y. Wu, M.J. Wainwright, K. Ramchandran, Network coding for distributed storage systems, IEEE Trans. Inform. Theory 56 (9) (2010) 4539–4551.

[41] J. Liu, K. Huang, H. Rong, H. Wang, M. Xian, Privacy-preserving public auditing for regenerating-code-based cloud storage, IEEE Trans Inf. Foren. Sec. 10 (7) (2015) 1513–1528.

[42] H.C. Chen, P.P. Lee, Enabling data integrity protection in regenerating-coding-based cloud storage: theory and implementation, IEEE Trans. Parallel Distrib. 25 (2) (2014) 407–416.

[43] R.L. Rivest, L. Adleman, M.L. Dertouzos, On data banks and privacy homomorphisms, NATO Adv. Sci. I F-Com. 4 (11) (1978) 169–180.

[44] C. Gentry, Computing arbitrary functions of encrypted data, Commun. ACM 53 (3) (2010) 97–105.

[45] A. Trepacheva, L. Babenko, Known plaintexts attack on polynomial based homomorphic encryption, Proceedings of the 7th International Conference on Security of Information and Networks (2014) 157.

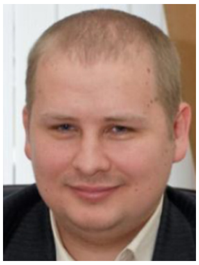
[46] E. Bach, J.O. Shallit, Algorithmic Number Theory: Efficient Algorithms, MIT press, 1996.

[47] Y. Wang, X. Song, M. Aboulhamid, A new algorithm for RNS magnitude comparison based on new Chinese Remainder Theorem II, Proceedings of the Ninth Great Lakes Symposium on VLSI (1999) 362–365.

Andrei Tchernykh received the Ph.D. degree from Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences, Russia in 1986. He is currently a full professor in Computer Science Department at CICESE Research Center, Ensenada, Baja California, Mexico, and a head of Parallel Computing Laboratory. He is a member of the National System of Researchers of Mexico (SNI), Level II. He leads a number of national and international research projects. He delivered more than 50 keynote speeches and invited lectures, served as a program committee member and general co-chair of more than 100 professional peer reviewed professional conferences. His main interests include resource optimization technique, adaptive resource provisioning, multi-objective optimization, computational intelligence, and incomplete information processing.

Uwe Schwiegelshohn received the Diploma and the Ph.D. degrees in Electrical Engineering from the TU Munich in 1984 and 1988, respectively. He was with the Computer Science department of the IBM T.J. Watson Research Center from 1988 to 1994 before becoming full Professor at TU Dortmund University where he heads the Robotics Research Lab. In 2008 he was appointed vice president of this university. Also in 2008 he became managing director of the Government sponsored D-Grid corporation to coordinate the Grid projects in Germany. His main research interest are scheduling problems and Grid computing.

El-Ghazali Talbi received the Master and Ph.D. degrees in Computer Science from the Institut National Polytechnique de Grenoble in France. He is a full Professor at the University of Lille and the head of DOLPHIN research group from both the Lille's Computer Science laboratory (LIFL, Universite Lille 1, CNRS) and INRIA Lille Nord Europe. His current research interests are in the field of multi-objective optimization, parallel algorithms, metaheuristics, combinatorial optimization, cluster and cloud computing, hybrid and cooperative optimization, and applications to logistics/transportation, bioinformatics and networks. Professor Talbi has to his credit more than 150 international publications including journal papers, book chapters and conferences proceedings



MikhailBabenko graduated from Stavropol State University (SSU) in 2007 with degree in mathematics. Received Ph.D. degree in mathematics from SSU in 2011. He works as assistant professor in Department of Applied Mathematics and Mathematical Modeling since 2012. He is an author of over 63 publications and 5 patents. His research interests include cloud computing, high-performance computing, residue number systems, neural networks, cryptography.

Please cite this article in press as: A. Tchernykh, et al., Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability, J. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.jocs.2016.11.011