Reliability in the Utility Computing Era: Towards Reliable Fog Computing

H. Madsen
Department of Informatics and Mathematical Modelling
Technical University of Denmark
Lyngby, Denmark
hm@imm.dtu.dk

G. Albeanu
Department of Mathematics and Computer Science
Spiru Haret University
Bucharest, Romania
g.albeanu.mi@spiruharet.ro

Bernard Burtschy
Informatique et Réseaux
Telecom ParisTech
46, rue Barrault 75634 Paris Cedex 13, France
bernard.burtschy@telecom-paristech.fr

Fl. Popentiu-Vladicescu
"UNESCO Chair" Department
Academy of Sciences
Oradea, 1, University Street, Romania
popentiu@imm.dtu.dk

***Abstract*— This paper considers current paradigms in computing and outlines the most important aspects concerning their reliability. The Fog computing paradigm as a non-trivial extension of the Cloud is considered and the reliability of the networks of smart devices are discussed. Combining the reliability requirements of grid and cloud paradigms with the reliability requirements of networks of sensor and actuators it follows that designing a reliable Fog computing platform is feasible.**

Keywords*— *grid computing, cloud computing, utility computing, internet of things, fog computing, reliability

I. INTRODUCTION

Computing today is based on various smart devices, computers, networks of computers, and high performance computers who benefits of various middleware to efficiently provide fastest, dependable and secure services. Through the ages, some technological waves were considered: mainframes – each one shared by many people, personal computing (one people interacting with one device), and ubiquitous computing (one people interacting with many devices – wireless networks of sensors/smart devices, in the age of calm technology). Considering the network computing field, various networking architectures and protocols were proposed and used over time. Nowadays, the most popular is based on the TCP/IP protocol – namely "the Internet", offering many services including those called "web-based services" [8, 12]. The major objective of last decade was to develop devices and services supporting HPC on grids and clouds [1]. The fog computing [2], just begun, and makes possible not only *Internet of Things* (IoT) development but also *Ubiquitous Computing* (UC) approaches, by extending cloud computing services to include smart sensors and intelligent devices.

The next section describes the major reliability characteristics of existing computing paradigms. Developments on Fog computing reliability are presented in the third section.

978-1-4799-0944-5/13/$31.00 ©2013 IEEE

II. CURRENT COMPUTING TECHNOLOGIES

In the utility computing era of cluster computing, grid computing, cloud computing, and recently of fog computing, the services are delivered in a manner similar to traditional utilities providers, according to [1, 18].

According to Buyya et al (2009), a clusters can be described as a number of stand-alone computers which are interconnected to form a distributed system seen as a single integrated computing resource [3]. More particular, such an architecture enabling "the sharing, selection, and aggregation of geographically 'autonomous' resources dynamically at runtime depending on their availability, capability, performance, cost, and user's quality-of-service requirements" is called a grid. From grid reliability point of view, the following areas of reliability should be considered [4]: (a) reliability of hardware, software, and data grid resources that participate to execute user applications; (b) reliability of end-user nodes and applications submitted to the grid for execution; (c) reliability of the grid services (resource allocation, scheduling etc.); (d) reliability of networks supporting data transport and message exchanging. The areas (a), (c), and (d) are also common to cluster reliability analysis.

When considering cloud computing paradigm many definitions arises depending on the viewpoint: business management or inform technology professionals.

In the following we are based on the works of Buyya [3] and Youseff, Butrico & Da Silva [18]. As a holistic view, "a cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers". It is clear that cloud computing has similarities with both clusters and grids, adding support for virtualization and dynamically management of resources.

The most important characteristics to be considered when analyzing cloud systems are related to the failover mechanism and the cloud delivery model (private, public, and hybrid) used by a particular implementation. A failover mechanism consists of automatic switching to redundant or standby resources upon the failure of abnormal response.

When consider the reliability of the cloud, we need a reference model. The most general cloud model is based on five layers. The bottom layer, called HaaS – Hardware as a Service, provides to the users (usually enterprises needing huge amount of IT resources) hardware & firmware as computational resources, storage infrastructure, and networking architecture. This layer is named also as "fabric" by Foster et al. (2008) in [8]. Not only the reliability of every technical resource is important, but also aspects related to the management of the data center, the scheduling policy, and the power-consumption model. The reliability of the service is highly influenced by infrastructure including power assurance and cooling strategies.

The next layer is represented by the Cloud Software Kernel (CSK) that manages the resources provided by the HaaS layer. There are many ways to implement CSK: virtual machine monitor, operating system kernel, and clustering middleware. It should be noted that grid computing operates the resources of large number of grid computers to work in parallel to solve a single problem, while cloud computing are multi-application oriented. The Cloud Software Infrastructure (CSI) is the next layer which is responsible to provide Computational Resources (CR), Data Storage (DS), and Communication (Comm) to the higher-level layers.

When considering the computational resources in the cloud, the IaaS (Infrastructure as a Service) model is obtained. The virtualization technique makes possible this service. Para-virtualization and Hardware-assisted Virtualization are the most used virtualization technologies. The reliability of this service can be discussed considering the reliability of the computational resources, and the virtualization procedure.

DS represents a very important infrastructure resource which consists of remote disks for storage supporting access from anywhere. Such service is called, sometimes, DaaS – Data Storage as a Service. The reliability of this service is more difficult to be assured if other cloud computing aspects are more important. According to Brewer's CAP conjecture [11], it is impossible for a distributed service to provided consistency, availability, and partition-tolerance at the same time. By consistency is assured that "all copies of data in the system must be observed coherently by external users". Partition-tolerance will assure that "the system must continue to operate even if it suffers from network failures", by partitioning the system into disconnected sub-networks. It was proved that Consistency & Partitioned-tolerance can be assured using the data model Bigtable (provided by Google), HBase, and Hypertable, the last two proposals being open source and emulating approximately the first model.

Providing a high quality CaaS (Communication as a Service) requires a guaranteed quality of service (QoS) in the cloud consisting of: (1) network security assurance; (2) dynamic provisioning or dedicated bandwidth availability; (3) guaranteed networking time delay, and (4) communication encryption and network monitoring. There are some acceptable proposals like Microsoft Connected Service Framework.

The second layer of the cloud computing model is Cloud Software Environment (CSE) used by software developers, and is commonly known as PaaS (Platform as a Service). The reliability of PaaS is influenced by the reliability of basic components (Google's App Engine, Hadoop, etc.) and the reliability of other software components available to the users (Map Reduce, Yahoo's Pig, etc.). In a simplified view, for web-community, examples of platforms can be considered LAMP (Linux/Apache/MySQL/Perl, PHP or Python) and WINS (Windows Server/Internet Information Services/.NET/SQL Server).

The outer layer is the Cloud Application (CA) layer, offered as SaaS (Software as a Service). Well known examples of SaaS are Saleforce CRM and Google Apps. An important concept is multi-tenancy. As mentioned by Singla (2012), multi-tenancy appears whenever only one instance of software, running on the server, are able to serve multiple clients (tenants) [17]. This concept differs from multi-instantiation, where separate software instances are running for different clients.

Considering failure management for clusters and grids we found that this is limited (a restarting procedure is used), while cloud offers strong support for fail over and content replication. This is possible by migrating virtual machines from one node to other.

The actual stage in cloud computing is represented by Google App Engine (PaaS), Microsoft Azure (PaaS), Sun Grid (IaaS), Amazon EC2 (IaaS), Aneka [3], and some solutions offered by Europe FP7 framework.

In the following we outline the main problems related on reliability of such computing paradigms: (1) *Failures are hard to be detected*: The detection failure probability is positive when some processing node of cloud returns incorrect results, but the client are not able to observe [6]; (2) *Existing failure detection methods remain experimental*: the failover mechanism should be improved in order to increase the MTBF and to decrease the MTR (failure/recovery orientation). Both run time and administration time failover mechanisms should be implemented. Failover functionality for the Data Server is required; (3) *More research on scalable fault detection methods is required*: failure detectors should be placed on every node, and a monitoring procedure based on heartbeats sent by nodes; (4) *Recognizing the faults leading to failures is also difficult*: the faults should be classified as hardware, process, and transmission faults; (5) *Faults are hard to be detected*, mainly the Byzantine faults and the faults generating cascading effects.

The reliability can be improved by checkpointing (periodically, the state is saved, to be resumed after failure), replication (executing in parallel), or rescheduling (of failed tasks). The checkpointing approach saves the computation time for the faulty tasks, but the runtime overhead, latency, and recovery delay are increased [14]. Checkpointing can be initiated by CSK or within applications (at SaaS lavel). In order

to assure a consistent recovery, coordinated checkpointing (by synchronization) can be used. Other strategies can use uncoordinated checkpointing combined with message logging, replication of the checkpointed data, and detection of optimal checkpointing intervals, and adaptive checkpoint. Checkpointing can also be used for security assurance reason, as proposed by Park et al. (2012) in [14].

When replication strategy is used the consistency should be assured. A good strategy of consistency assuring is based on implementing state-machine replication, and partition-tolerance. Various cloud providers use Geo-replicated systems and there are various rescheduling algorithms. The algorithm proposed by Lee & Zomaya (2010) asks for the initiation of rescheduling "together with subsequent tasks on the resources that caused the delay in task completion" [12].

In order to evaluate the reliability of a layer the following elements can be used [9]:

- n – the number of users;

- f – the number of users affected by failures in time less than specified in Service Level Agreement;

- p – the promised MTTF (also specified in Service Level Agreement);

- v – the probability of agreement violation = 1-f/n.

Then, the service reliability is given by p*v.

However, in order to analyze the reliability of cloud services in a holistic manner, a deep view is necessary. As Dai, Yang, Dongara & Zhang (2009) found out in [5], the failure analyses of cloud service should consider more categories of failures, including: hardware/firmware failure, network failure, database failure, software failure, computing resource missing, data resource missing, timeout, and overflow.

It is said that an overflow failure is experienced when the request queue is full. Timeout failure, in general, appears when the waiting time is over the time established by the service level agreement. Overflow and Timeout failures belong to the class Request Stage Failures (RSF), while the rest of the above mentioned failures belong to the class of Execution Stage Failures (ESF). Based on queuing theory and Markov processes, the Cloud Service Reliability Model for the RSF class, proposed in [5] is able to provide: the probability for the overflow failure not to occur, the probability density function of waiting time, the probability of service without timeout or overflow failures.

The Cloud Service Reliability Model for ESF class is a graph-based model integrating different types of failures which appear during execution phase. A set of parameters are used: processing speed, amount of data downloaded/uploaded, the workload (Number of Instructions), amount of data exchanged between subtasks, the communication link's bandwidth, the failure rate of elements. The output can provides the execution reliability based on the minimal execution spanning tree of the graph. Finally, the reliability of the service is obtained by combining the request stage reliability and execution stage reliability.

III. TOWARDS RELIABLE FOG COMPUTING

Following [10], smart connectivity of intelligent devices supporting wireless communication with existing networks and participating to computational tasks using network resources is an important objective within IoT vision. IoT model is based on H/M/P structure, as described by [1]: H - Hardware (consisting of sensors, actuators, and embedded communication hardware); M – Middleware (providing on demand storage and data analysis tools); P – presentation (based on understanding visualization and interpretation tools). Machine-to-Machine (M2M) communication promises to be the solution to obtain an "intelligent" environment of things. Therefore, the IoT graph connects users to sensors, which are identified by a label (through URN – Uniform Resource Name). The nodes are accessible (through URL-Uniform Resource Locator). The IoT resources are controllable (through URC - Uniform Resource Characteristics). Evans (2011) describes IoT as a network of individual networks connected together with security, analytics, and management [7]. Gubbi et al. (2012) propose a Cloud Centric IoT based on the Aneka cloud computing platform [10].

When considering the reliability of IoT we have to take into consideration the failure of individual sensors, the lack of coverage from access networks in some region, the failure of the whole network, the failure of the service platform, the failure of the user's interface connected to system etc.

In the following will be addressed the fog computing paradigm, a non-trivial extension of the Cloud, as developed by CISCO Systems (see [2]). Then we discuss the associated reliability challenges. According to Bonomi et al. (2012), in [2], "Fog Computing is a highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network." The extension is motivated by: (1) the need for geographical distribution of resources rather than a centralized one as in the Cloud; (2) the need to incorporate large networks of sensors communicating, usually by wireless access, various data; (3) the requirement to support real-time communication with mobile devices; (4) the need for supporting heterogeneous devices and interoperability with different providers; (5) the requirement of on-line analytic and interplay with the Cloud.

Many applications require both fog localization and cloud centralization. In this respect, Fog computing plays an important role in at least the following fields: Smart Connected Vehicles (SCV), Smart Grids (SG), and Wireless Sensor and Actuator Networks (WSN & WSAN). The benefit of connecting a Smart Light Traffic system to the Cloud is obvious. Also, the characteristics of the Fog computing (proximity and location awareness, geo-distribution, hierarchical organization) make it the suitable platform to support both energy-constrained WSNs and WSANs and the usage of the network to optimize the business in the field of interest.

To be efficiently useful in the above fields, the Fog computing platform follows a multi-tier architecture. At the border of the network are fog collectors of data generated by smart devices (vehicles, sensors etc.). The first tier is designed

for M2M interaction and has the following functionalities: (1) using volatile memory, collects and process data from devices, and sends control commands to the actuators; (2) filters the local data and sends the rest of data to the higher tiers. The next tiers are responsible with visualization and reporting. The higher tier is represented by the Cloud.

From reliability point of view it is necessary to add at above requirements those related to the wireless network of sensors and actuators. The most important aspects resides from practical challenges, as EMI project [19] identified, like: (1) Multi-path propagation generates distortions ; (2) Industrial and hospital environments exhibit higher levels of radiated electromagnetic interference; (3) The electromagnetic interference arises from different devices and processes.

According to [13], the various reliability protocols for WSNs can belong to two general categories: packet reliability and event reliability. The most basic facts about WSNs reliability are: (a) *Packet reliability* has to ensure to the sink node the delivery of every data packet, while *event reliability* has to ensure that at least one of many packets from the sensors that detected an event is delivered; (b) *End-to-end packet reliability* requires the acknowledgement of every packet and the retransmission of each lost packet; (c) There has been proposed many protocols to transport packets [15], or send events [13]; (d) There are available methodologies for the reliability evaluation of a WSN as shown in [15]; (e) The sensors, in general, are not expensive but their readings can be affected by noise, in this case the information accuracy problem can be solved by redundancy.

The above analysis shows that building Fog computing based projects is challenging. The availability of various algorithms and methodologies dealing with reliability of the networks of smart devices, and operating under specific conditions that ask for fault tolerant techniques to assure the information accuracy collected and/or sent to the high level tiers of the platform makes possible such projects.

IV. CONCLUSIONS

Current view on computing is linked to the utility concept: pay for using a service (selected from a large offer: HaaS, CaaS, DaaS, IaaS, PaaS, SaaS etc.).

This paper presented the reliability challenges posed by current computing paradigms and extends the discussion towards reliable Fog platforms incorporating networks of smart devices communicating among them, but also with the Cloud. The most important idea consists of the feasibility of reliable Fog computing platforms for real life projects.

ACKNOWLEDGMENT

During this research, the authors were supported by their departments in the framework of research projects on future computing paradigms. Authors gratefully acknowledge many helpful comments and suggestions that they have received from anonymous reviewers.

REFERENCES

[1] Albeanu G., Popentiu-Vladicescu F., "Performability metrics for online services in education", Proceedings of eLSE, 2013, Bucharest..

[2] Bonomi F., Milito R., Zhu J. & Addepalli S. "Fog Computing and its Role in the Internet of Things", Proceedings of MCC'12, August 17, 2012, Helsinki, Finland, pp. 13-15, 2012.

[3] Buyya R., Yeo C.S., Venugopal S., Broberg J. & Brandic I., "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, 25, pp. 599-616, DOI: 10.1016/j.future.2008.12.001, 2009.

[4] Dabrowski C., "Reliability in grid computing systems", Concurrency and Computation: Practice and experience, DOI: 10.1002/cpe.1410, 2009.

[5] Dai Y.-S., Yang B., Dongarra J. & Zhang G., "Cloud Service Reliability: Modeling and Analysis", the 15th IEEE Pacific Rim International Symposium on Dependable Computing, 2009.

[6] Deng J., Huang C.-H. S., Han S. Y. & Deng H., "Fault-Tolerant and Reliable Computation in Cloud Computing", Proc. of IEEE Globecom Workshop on Web and Pervasive Security, Miami, FL, USA, pp. 1601-1605, 2010.

[7] Evans D. "The Internet of Things. How the Next Evolution of the Internet is Changing Everything", CISCO White Paper, 2011.

[8] Foster I., Zhao Y., Raicu I. & Lu S., "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop, GCE '08, 2008.

[9] Garg S.K., Versteeg S. & Buyya R., "A framework for ranking of cloud computing services", Future Generation Computer Systems, DOI:10.1016/j.future.2012.06.006, 2012.

[10] Gubbi J., Buyya R., Marusic S. & Marimuthu Palaniswami M. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions", Technical Report CLOUDS-TR-2012-2, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne.

[11] Hsu C.W., Wang C.W. & Shieh S., "Reliability and Security of Large Scale Data Storage in Cloud Computing", The Reliability Society 2010 Annual Technical Report, 2010.

[12] Lee Y. C. & Zomaya A. Y. "Rescheduling for reliable job completion with the support of clouds", Future Generation Computer Systems (2010), doi:10.1016/j.future. 2010.02.010.

[13] Mahmood M.A. & Seah W.K.HG. "Event Reliability in Wireless Sensor Networks", The Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp. 377 – 382, 2011.

[14] Park M.-W., Eom J.-H., Kim S.-H., Kim N.-U. & Chung T.-M. "A Study on Architecture of a Cyber Checkpoint Model in the Cloud Computing Environment", ITCS 2012, pp. 123-128, 2012.

[15] Pereira P.R., Grilo A., Rocha F., Nunes M.S., Casaca A., Chaudet C., Almström P. & Johansson M. "End-to-End Reliability in Wireless Sensor Networks: Survey and Research Challenges", EuroFGI Workshop on IP QoS and Traffic Control, Lisbon, Portugal, December 6-7, 2007.

[16] Silva I., Guedes L.A., Portugal P. & Vasques F. "Reliability and Availability Evaluation of Wireless Sensor Networks for Industrial Applications", Sensors 2012, 12, 806-838; doi:10.3390/s120100806.

[17] Singla B.S., "Comparison of Various Computing Technologies and Emerging Cloud Platforms", VSRD International Journal of Computer Science & Information Technology, 2(11), pp. 907-911, 2012.

[18] Youseff L., Butrico M. & Da Silva D., "Toward a Unified Ontology of Cloud Computing", Grid Computing Environments Workshop GCE '08, 2008.

[19] ***, The EMI project, "Reliable Wireless Machine-to-Machine Communications", http://www.hig.se/Ext/Sv/Organisation/Akademier/Akademin-for-teknik-och-miljo/Forskning/Elektronik/Elektronik/Reliable-Wireless-Machine-to-Machine-Communications.html, 2009-2011.