Research Issues for Future Cloud Infrastructures
Inria Position Paper

Frédéric Desprez
INRIA/LIG, France
Deputy Scientific Director
Frederic.Desprez@inria.fr

Adrien Lebre
INRIA/Ecole des Mines de Nantes, France
Researcher
Adrien.Lebre@inria.fr

November 7, 2016

Cloud offers now have reached a certain level of maturity and have become critical infrastructures within the Internet ecosystem. However, the convergence of the Internet of Things (IoT) and the data center worlds is favoring the evolution of current cloud computing infrastructures towards a massively distributed federation of smaller data centers placed at the edge of network backbones. Referred as Fog/Edge Computing, this paradigm shift is dictated by technological advances in the capacity and capabilities of both mobile networks and end-user devices, along with requirements for improved QoS and growing user concern and awareness of trust and privacy issues. In addition to challenges related to the fine management of resources as well as the efficient use of cloud computing platforms by applications, experts from academia and industry should address scientific and technical challenges related to this paradigm shift.

This position paper provides a list of some of them as well as some concerns and issues dealing with Cloud and Service Computing. It covers the following topics:

- Infrastructure/Application Management
- Network "softwarization" and Network as a Service
- Energy Proportionality
- Formalization of Cloud Computing solutions
- Data-Intensive scalable computing
- Experimental driven research in Cloud Computing

It has been written with contributions of F. Desprez (Inria), M. Dias de Assuncao (Inria), I. Chrisment (Telecom Nancy), J. Francois (Inria), A. Lebre (Inria), L. Lefèvre (Inria), D. Margery (Inria), P. Merle (Inria), L. Nussbaum (Université de Lorraine), C. Perez (Inria), G. Pierre (IRISA), D. Saucez (Inria), and P. Valduriez (Inria).

1 Infrastructure/Application Management

Elasticity is a major feature of Cloud Computing defined as "the degree to which a system or an application is able to adapt to workload changes by provisioning and relaxing resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible [8]. To achieve this, elasticity is based on deployment and scaling techniques (i.e, horizontal replication, vertical scaling, migration) automatically driven by dynamic optimization policies (i.e., reactive, predictive). Thus, elasticity can be summarized by the following equation: Elasticity = Scaling + Automation + Optimization. However there is no one-size-fits-all solution to make any cloud system elastic. To identify the right elasticity strategy for a given application our community should address two important challenges.

The first challenge is related to the way developers describe the software architecture of applications. Several efforts have promoted the usage of models, usually component models [4], to describe the application to be deployed on (multi-) clouds such as TOSCA[1] or CAMEL[2]. While they provide a support for hardware constraints—such as CPU, memory, OS, storage, etc.—needed to be able to deploy an application, they have limited support for application variability. In most cases, they are only able to support scalability of some components. This is not enough to handle the large variability that applications have to take in order to adapt to various situations. Also, this is not enough to express advanced (and so complex to describe) applications beyond the basic 3-tier applications. Thus a simple enough application description model that non experts can use while being able to match the user constraints should be proposed. However, more features are also needed such as to enable advanced application composability to enable the building of advanced applications made of several building blocks.

The second challenge is then to develop a framework for elasticity "à la carte". Such a framework would use the aforementioned model to support any scaling techniques and optimization policies and would automatically and dynamically select the right elasticity strategy matching the current demand.

Providing elasticity "à la carte" leads to the challenge of multi level loops of optimisation. On one hand, cloud providers execute some optimizations process to consolidate the usage of their resource that include for instance virtual machine migration. On the other hand, application runtime can have some adaptive layers that understand advanced application level features (cross cloud optimization for example). The integration of both adaption loops is an open challenge that can lead to a better utilisation of resources.

Moreover, it is noteworthy to mention that the distribution of Fog/Edge Computing infrastructures will increase the needs of such application models as well as dedicated frameworks to manage the life-cycle of applications. Fog/Edge computing is defined as a widely-distributed execution environment composed of heterogeneous resources such as one or more data centers, edge clouds at the border of the public Internet, and end-user devices (IoT devices, smartphones, tablets. . .) [2]. Depending on their specific requirements and the availability of suitable resources, fog/edge computing applications will be able to seamlessly

[1] OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA)
[2] `http://camel-dsl.org/`

deploy themselves over the most appropriate set of resources (possibly belonging to a variety of independent providers). They will also need to dynamically adapt to any modification in the user demand (because of users' mobility or user-induced workload fluctuations) and the execution environment (because of variations in resource availability, price, or ownership).

Although the ICT program has recently started to fund a few projects focusing on the design of basic infrastructures to support this vision, it is important to pursue this effort. In particular, EU Engineers and Researchers should address the following questions:

- How can an average developer **build** and **debug** Fog/Edge Computing applications as easily as he/she currently designs regular mobile applications?
- How can Fog/Edge Computing platforms efficiently **deploy** applications across heterogeneous resources belonging, possibly, to multiple independent cloud/fog/telco providers?
- How can Fog/Edge Computing platforms efficiently monitor the execution of applications to enforce specific **security regulations?**
- How can Fog/Edge Computing platforms efficiently monitor the execution of applications to **dynamically adapt the execution environment** to changing user demands and/or execution conditions?
- Can aforementioned models and frameworks deliver answers to those questions?

In addition to proposing resource management systems capable of supervising Fog/Edge Computing infrastrutures deployed throughout a network operator but also between different tenants, Engineers and Researchers should propose advanced solutions and technologies which will realistically transform Fog/Edge Computing research into usable and marketable products.

2 "Softwarization" of Networks

The innovations that drive our society all rely on networks and are made by people from the web, developers, data scientists, or even biologists that will never be network experts. To fill the gap between the Internet and its users, network "softwarization" with Network Function Virtualization (NFV) and Software Defined Networking (SDN) is thus becoming the norm and offers to conceive networks as pieces of software instead of a bunch of devices to configure.

The "softwarization" of networks is built upon the ability to virtualize network functions in order to instantiate them on-demand by the intermediate of Application Programming Interfaces (API). A network operator can offer dedicated and customized services to their customers and are now even able to offer their own network to other customers (Network as a Service) [9]. In such situations, programmers can compose network services in an elastic manner using dedicated APIs and therefore integrate the network directly into their applications, as yet another component of their system .

This technical shift is strongly coupled with an unavoidable business model change due to the openness of the Internet, which allows major Over-the-Top

(OTT) actors to lead the service sector without managing the network infrastructure. Hence, this is a differential value that operators can provide by offering their resources, in particular, their network as a service (NaaS).

The "softwarization" of the network is definitely the right approach to follow. However several research questions need to be addressed to favor its adoption.

The first question our community should answer is related to the network specifics. Networks are inherently distributed systems with potentially long propagation delay, high performance variability, malicious behavior, bugs, frequent outages . . . These properties are very specific to networks and the question of how the service creation and lifecycle management have to be operated in this context is crucial. Virtualization of network functions has led to the whole new concept of Service Function Chaining (SFC) that aims at building on the fly network services by deploying them in the Cloud. However, if we hide the notion of network to users, they will ignore the fact that they are not reliable and suffer of propagation delay and variability. To allow such level of abstraction, the network itself must thus provide resiliency to the chains. To that aim, it is necessary to develop mechanisms that can deliver a comprehensive monitoring of the resources and dynamically determine the number of instances to be deployed for each function and how to deploy them in the network in order to guarantee some level of resiliency to variability and outages. Similarly, these mechanisms should detect malicious behavior and propose countermeasures: the advantage of the Internet is to share the infrastructure between and operators. Unfortunately, part of them are malicious and try either to steal information or to block it. By proposing mechanisms that can take into account network specifics, it will be possible to ensure that services deployed in the Internet will provide some level of safety and security with reasonable performance guarantees.

The second question is related to the limit of the network virtualization. To answer such a question, there is a need to study the problem of virtualizing the network infrastructure for critical services, i.e., services requiring more than the usual best-effort offer made by the Internet. Typically, critical systems such as planes, power plants, or factories have one redundant physical network built of specialized network elements per functionality in order to guarantee a total isolation of the functions and reach high performance, safety, and security levels. This approach results in extremely complex networks that make CAPEX and OPEX to explode. The idea here, would be to virtualize these networks in order to have to rely only on one physical network infrastructure built with Commodity Off-The-Shelf (COTS) technology (i.e., x86, Ethernet) and to run the specialized networks as virtual entities transported on top of the commodity network.

3 Energy Proportionality

Electricity consumption is one of the main limiting factor for deploying large set of physical resources and equipment in Cloud datacenters.

Due to their consolidation and elasticity capabilities, Clouds are the natural candidates to dynamically adapt performances, resources availability and energy consumption to load and usage. However, in order to react quickly to possible non predicted usage and load bursts, most of Clouds are over-provisioned by providers. Redundancy, duplication approaches are applied to Cloud resources.

But, the important static consumption of IT resources, performing small amount of services or being idle, remains a reality. The combination of such resources in Clouds result to large non energy proportional infrastructures.

While understanding the energy usage of large scale systems mixing virtual instances of applications, physical IT resources and physical infrastructures remains a challenge; exploring approaches able to express and support energy proportional consumption of virtualized infrastructures is a mandatory step for designing energy efficient Clouds. This will require the design of new energy models, algorithms and frameworks through the involvement of academic and industrial contributions.

In parallel, the adoption of software defined infrastructures, protocols, tools and models (like Software Defined Networks as described in the previous Section) allows new levels of flexibility in Clouds. If well oriented, this dynamics can have potential big impact on energy performance and efficiency. Metrics and approaches must be designed to combine this flexibility with multi-objectives models in order to support trade-off between performance, energy efficiency, QoS..etc..

Moreover, with the emergence of fog and edge computing able to support latency aware services, deployment of Clouds infrastructures face new challenges in terms of energy provisioning issues. We observe several initiatives embedding renewable energy provisioning to Clouds. Such multi-disciplinary explorations must be encouraged in order to reduce the environmental impact of Cloud infrastructures.

4 A Theory of Cloud Computing

In less than a decade, cloud computing has become the most popular Internet-based computing paradigm to offer and access on-demand computational services and resources. On the one hand, many definitions of cloud computing were proposed in the literature. The most well-known and accepted definition is certainly the NIST definition of cloud computing. This cloud model is composed of five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service), three service models (Software as a Service or SaaS, Platform as a Service or PaaS, and Infrastructure as a Service or IaaS), and four deployment models (private cloud, community cloud, public cloud, and hybrid cloud). However all definitions in the literature are mostly informal as they are written in natural language and then could be interpreted differently by cloud practitioners. On the other hand, the cloud market has grown rapidly in the last decade and now encompasses hundreds of cloud offers including public clouds like Amazon Web Services, Google Cloud Platform, Microsoft Azure, Salesforce, and cloud software stacks like VMware vSphere, Apache OpenStack, Cloud Foundry, Docker, to cite a few. We could easily predict that a multitude of new cloud offers will appear in coming years. However this plethora of cloud offers raises prominent issues in terms of heterogeneity of provided services, interoperability between clouds and portability of business applications on multiple clouds. To address these issues, various standards have emerged like DMTF Open Virtualization Format (OVF), DMTF Cloud Infrastructure Management Interface (CIMI), OASIS Cloud Application Management for Platforms (CAMP), OGF Open Cloud Computing Interface

(OCCI), and OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA). Each of these cloud standards covers only a part of cloud interoperability and portability issues: OVF is a standard packaging format for virtual appliances, CIMI is a standard REST API for any IaaS, CAMP is a standard REST API for any PaaS, OCCI is a model and REST API for any kind of cloud resources, and TOSCA is a standard language to describe cloud applications and orchestrate their deployment. However, these cloud standards are far from being supported by all cloud offers. Thus, cloud computing is and will continue to be a collection of informal and ambiguous definitions, non interoperable technological offers and poorly deployed standards.

We think that there is really a lack of a theory of cloud computing. The challenge is to build such a theory. This theory would provide a formal definition of cloud computing and specify its foundations mathematically. This theory would allow us to capture and specify any functional and nonfunctional characteristics of any computational resources, including current IaaS/PaaS/SaaS ones but also future Everything as a Service (XaaS) ones. Then we would be able to reason and prove properties on individual cloud resources but also sets of interconnected cloud resources, aka cloud applications/systems. Cloud standards would be specifiable within this cloud theory. Finally, this cloud theory would provide foundations to address semantic interoperability in cloud computing.

5 Scientific Data Analysis Using Data-Intensive Scalable Computing in the Cloud

Data-intensive science [6] requires the integration of two fairly different paradigms: high-performance computing (HPC) and data-intensive scalable computing (DISC). HPC is compute-centric and focuses on high-performance of simulation applications, typically using powerful, yet expensive supercomputers. DISC [3], on the other hand, is data-centric and focuses on fault-tolerance and scalability of web and cloud applications using cost-effective clusters of commodity hardware. Examples of DISC systems include big data processing frameworks such as Hadoop or Apache Spark or NoSQL systems (see [1] which includes a survey of DISC systems). To harness parallel processing, HPC uses a low-level programming model (such as MPI or OpenMP) while DISC relies on powerful data processing operators (Map, Reduce, Filter, . . .). Data storage is also quite different: supercomputers typically rely on a shared disk infrastructure and data must be loaded in compute nodes before processing while DISC systems rely on a shared-nothing cluster (of disk-based nodes) and data partitioning.

Spurred by the growing need to analyze big scientific data, the convergence between HPC and DISC has been a recent topic of interest [5, 11]. However, simply porting the Hadoop stack on a supercomputer [7] is not cost-effective, and does not solve the scalability and fault-tolerance issues addressed by DISC. On the other hand, DISC systems have not been designed for scientific applications, which have different requirements in terms of data analysis and visualization.

A grand challenge becomes scientific data analysis using DISC in the cloud, which requires developing architectures and methods to combine simulation and data analysis. We can distinguish between three main approaches depending on where analysis is done [10]: postprocessing, in-situ and in-transit. Postprocessing analysis performs analysis after simulation, e.g. by loosely coupling a supercomputer and a DISC cluster in the cloud. This approach is the simplest but is restricted to batch analysis. In-situ analysis runs on the same compute resources as the simulation, e.g. a supercomputer, thus making it easy to perform interactive analysis. In-transit analysis offloads analysis to a separate partition of compute resources, e.g. using a single cluster in the cloud with both compute nodes and data nodes that communicate through a high-speed network. Although less intrusive than in-situ, this approach requires careful synchronization of simulation and analysis. More work is needed to study different architectures for different scientific data analysis applications using DISC in the cloud and their trade-offs.

Another important field of research is around stream processing. Data captured in applications such as operational monitoring of large infrastructures, Internet of Things, and smart cities need to be analyzed quickly. Several engines have been developed to perform distributed data stream processing in scalable and fault tolerant ways. Most engines follow a one-pass processing model where the application is designed as a directed acyclic graph of which vertices are processing elements that execute a predefined or used-specified function and edges define the communication patterns. Another model consists in discretising incoming data streams and launching periodical micro-batch executions. Under this model, data received from an input stream is buffered during a time window, and towards the end of the window the engine triggers distributed batch processing.

With the growing number of scenarios where huge amounts of data are collected by numerous devices, and for which low latency processing is required, service providers aim at exploiting resources available at the edge of the Internet. In addition to optimising the placement data processing tasks in such environments whilst minimising the use of network resources and latency, efficient methods to manage resource elasticity in these scenarios need to be designed.

6 Experimental Reproducible Research on Clouds

The business model behind commercial clouds relies on the delivery of abstracted services: computes, storages, networks, execution environments, applications. The way such services are provisioned is kept secret, with in the best case service level agreements (SLAs) to formalise the relationship between the consumer of services and the provider. The level of multi-tenancy and over-provisioning, the load induced by other users and the variability in the way a given service is provisioned are trade secrets, jealously kept. Observability and Control, as needed to understand the performance of applications or algorithms, are therefore very limited in many cloud environments.

They are nevertheless very much needed should users of such environments want to gain insights into the key factors of performance. This is important for the scientific community, so that the knowledge produced relies less on observing the behaviour on current cloud offerings that can change in nature at any moment and more on correlation between root causes and their effects. It is also very important for business users of cloud services so they can gain insights in the performance of their applications that rely more on their characteristics than on the combination of cloud provider chosen, application design and circumstances. These insights can then be used to compare offerings, predict pricing and in negotiation of SLAs better suited to their needs.

Finally, current offerings rely on highly centralised data-centers provisioned with x86_64 processors. Experimenting at scale with more decentralised approaches, other processor types such as ARM or any other highly innovative approaches is difficult because commercial providers have standardised on the highly centralised data-center model.

We therefore believe that in order to support discovery and innovation, we should invest in platforms to support experiment-driven approaches. These platforms should highly value Observability and Control, and offer to academics and businesses alike a wide range of possible configurations, from highly instrumented and controllable classical central data-centers, to highly innovative platforms. Some should be usable for short lived experiments while others could be open to experimental deployments of services available to the general public. In terms of Observability, they should provide experimenters with metrics about all levels of the infrastructure: network, storage elements, hypervisor, etc. They should also enable the experimenter to deploy components of the Cloud infrastructure with their own in order to evaluate new ideas. Additionally, those platforms should also be inter-connected into federations of testbeds, in order to support experiments involving more heterogeneous set of resources (edge clouds, end-user mobile devices, etc).

It can also be expected that some enhanced Observability and Control features will percolate to current production Cloud offerings, enabling them to provide various levels of control and predictability of performance, to adjust to the needs of more demanding applications. However, it is likely that this higher control will be provided in exchange of a higher cost. Application developers need to continue to design applications that will perform fine despite the current performance uncertainties in Cloud environments. To advance the state of research in this area, more work is needed to characterize the performance provided, and the perturbations encountered on typical Clouds. That work should lead to the design of suitable load and faults injectors, that could be use to create synthetic but realistic execution environments in order to evaluate applications.

References

[1] C. Bondiombouy and P. Valduriez. Query Processing in Multistore Systems. *Int. Journal of Cloud Computing*, 2016. to appear, `http://hal-lirmm.ccsd.cnrs.fr/lirmm-01341158`.

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[3] R. Bryant. Data-Intensive Scalable Computing for Scientific Applications. *Computing in Science & Engineering*, 13(6):pp. 25–33, 2011.

[4] Szyperski Clemens, Gruntz Dominik, and Murer Stephan. Component software: beyond object-oriented programming, 1998.

[5] A. Coutinho. Computational Science and Big Data: Where are We Now? In *XLDB Workshop*, 2014. http://xldb-rio2014.linea.gov.br/program.

[6] T. Critchlow and K. Kleese van Dam. *Data-Intensive Science*. Chapman and Hall/CRC, 2013.

[7] G. Fox, J. Qiu, S. Jha, S. Ekanayake, and S. Kamburugamuve. Big Data, Simulations and HPC Convergence. https://www.researchgate.net/publication/301231174, 2016.

[8] N.-R. Herbst, S. Kounev, and R. Reussner. Elasticity in Cloud Computing: What It Is, and What It Is Not. In *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 2013)*, San Jose, CA, June 2012.

[9] Hyojoon Kim and Nick Feamster. Improving network management with software defined networking. *IEEE Communications Magazine*, 51(2):114–119, 2013.

[10] R. Oldfield, K. Moreland, N. Fabian, and D. Rogers. Evaluation of Methods to Integrate Analysis into a Large-Scale Shock Physics Code. In *ACM Int. Conf. on Supercomputing*, pages pp. 83–92, 2014.

[11] P. Valduriez. Data-intensive HPC: opportunities and challenges. In *Big Data and Extreme-scale computing (BDEC)*, 2015. http://hal-lirmm.ccsd.cnrs.fr/lirmm-01184018.