Fact Sheet

Beyond the Clouds, How Should Next Utility Computing Infrastructures Be Designed?

The DISCOVERY Initiative

DIstributed and COoperative management of Virtual Environments autonomouslY

Date of preparation: February 6, 2014
Version: 2014-02-06 11:46

Proposal Abstract:

The DISCOVERY initiative aims at exploring a new way of operating Utility Computing (UC) resources.

To accommodate the ever-increasing demand for Utility Computing (UC) resources, while taking into account both energy and economical issues, the current trend consists in building larger and larger data centers in a few strategic locations. Although such an approach enables to cope with the actual demand while continuing to operate UC resources through centralized software system, it is far from delivering sustainable and efficient UC infrastructures. We claim that a disruptive change in UC infrastructures is required: UC resources should be managed differently, considering locality as a primary concern. To this aim, **we propose to leverage any facilities available through the Internet in order to deliver widely distributed UC platforms that can better match the geographical dispersal of users as well as the unending demand. Critical to the emergence of such locality-based UC (LUC) platforms is the availability of appropriate operating mechanisms. We advocate the implementation of a unified system driving the use of resources at an unprecedented scale by turning a complex and diverse infrastructure into a collection of abstracted computing facilities that is both easy to operate and reliable.**

By deploying and using such a *LUC Operating System* on backbones, our ultimate vision is to make possible to host/operate a large part of the Internet by its internal structure itself: A scalable and nearly infinite set of resources delivered by any computing facilities forming the Internet, starting from the larger hubs operated by ISPs, government and academic institutions to any idle resources that may be provided by end-users. Unlike previous researches on distributed operating systems, we propose to consider virtual machines (VMs) instead of processes as the basic element. System virtualization offers several capabilities that increase the flexibility of resources management, allowing to investigate novel decentralized schemes that will enable to achieve the LUC infrastructure we target.

Consortium:

Coordinator - Adrien Lèbre (Mines de Nantes, Inria Rennes Bretagne Atlantique, adrien.lebre@inria.fr)

Inria(Asap, Ascola, Avalon, Myriads - France), Orange R&D (France), Renater (France), BSC (Spain), CRS4(Italy), PSNC (Poland), GRNET (Greece)

Web site: `http://beyondtheclouds.github.io`

Contributors: Marin Bertier, Thierry Coupaye, Frédéric Desprez, Adrien Lebre, Anne-Cécile Orgerie, Jonathan Rouzaud-Cornabas, Cédric Tedeschi.

1 Context

1.1 Current Trends.

The success of Cloud Computing has driven the advent of Utility Computing (UC). However, Cloud Computing is a victim of its own success: In order to answer the escalating demand for computing resources, Cloud Computing providers must build data centers (DCs) of ever-increasing size. As a consequence, besides facing the well-known issues of large-scale platforms management, large-scale DCs have now to deal with energy considerations that limit the number of physical resources that one location can host.

Instead of investigating alternative solutions that could tackle the aforementioned concerns, the current trend consists in deploying larger and larger DCs in few strategic locations presenting energy advantages. For example, Western North Carolina, USA, is an attractive area due to its abundant capacity of coal and nuclear power brought about the departure of the textile and furniture industry [GC13]. More recently, several proposals suggested building next generation DCs close to the polar circle in order to leverage free cooling techniques, considering that cooling is accounting for a big part of the electricity consumption [GHMP08].

1.2 Inherent Limitations of Large-scale DCs

Although building large scale DCs enables to cope with the actual demand, it is far from delivering sustainable and efficient UC infrastructures. In addition to requiring the construction and the deployment of a complete network infrastructure to reach each DC, it exacerbates the inherent limitations of the Cloud Computing model:

- The externalization of private applications/data often faces legal issues that restrain companies from outsourcing them on external infrastructures, especially when located in other countries.
- The overhead implied by the unavoidable use of the Internet to reach distant platforms is wasteful and costly in several situations: Deploying a broadcasting service of local events or an online service to order pizzas at the edge of the polar circle, for instance, leads to important overheads since most of the users are *a priori* located in the neighborhood of the event/the pizzeria.
- The connectivity to the application/data cannot be ensured by centralized dedicated centers, especially if they are located in a similar geographical zone. The only way to ensure disaster recovery is to leverage distinct sites.[1]

The two first points could be partially tackled by hybrid or federated Cloud solutions [AFG^{+}10], that aim at extending the resources available on one Cloud with those of another one; however, the third one requires a disruptive change in the way UC resources are managed.

Another issue is that, according to some projections of a recent IEEE report [Gro12], the network traffic continues to double roughly every year. Consequently, bringing the IT services closer to the end-users is becoming crucial to limit the energy impact of these exchanges and to save the bandwidth of some links. Similarly, this notion of locality is critical for the adoption of the UC model by applications that need to deal with a large amount of data as getting them in and out actual UC infrastructures may significantly impact the global performance [Fos11].

The concept of micro/nano DCs at the edge of the backbone [GHMP08] may be seen as a complementary solution to hybrid platforms in order to reduce the overhead of network exchanges. However, operating multiple small DCs breaks somehow the idea of mutualization in terms of physical resources and administration simplicity, making this approach questionable.

1.3 Ubiquitous and Oversized Network Backbones

One way to partially solve the mutualization concern enlightened by the defenders of large-scale DCs is to directly deploy the concept of micro/nano DCs upon the Internet backbone. People are (and will be) more and more surrounded by computing resources, especially those in charge of interconnecting all IT equipments. Even though these small and medium-sized facilities include resources that are barely used [And03, BAM10], they can hardly

[1] "Amazon outages – lessons learned", http://gigaom.com/cloud/amazon-outages-lessons-learned/ (valid on Nov 2013, the 30$^{\text{th}}$).

be removed (*e.g.* routers). Considering this important aspect, we claim that a new generation of UC platforms can be delivered by leveraging existing network centers, starting from the core nodes of the backbone to the different network access points in charge of interconnecting public and private institutions. By such a mean, network and UC providers would be able to mutualize resources that are mandatory to operate network/data centers while delivering widely distributed UC platforms able to better match the geographical dispersal of users. Figure 1 allows to better capture the advantages of such a proposal. It shows a snapshot of the network weather map of RENATER[2], the backbone dedicated to universities and research institutes in France. It reveals several important points:

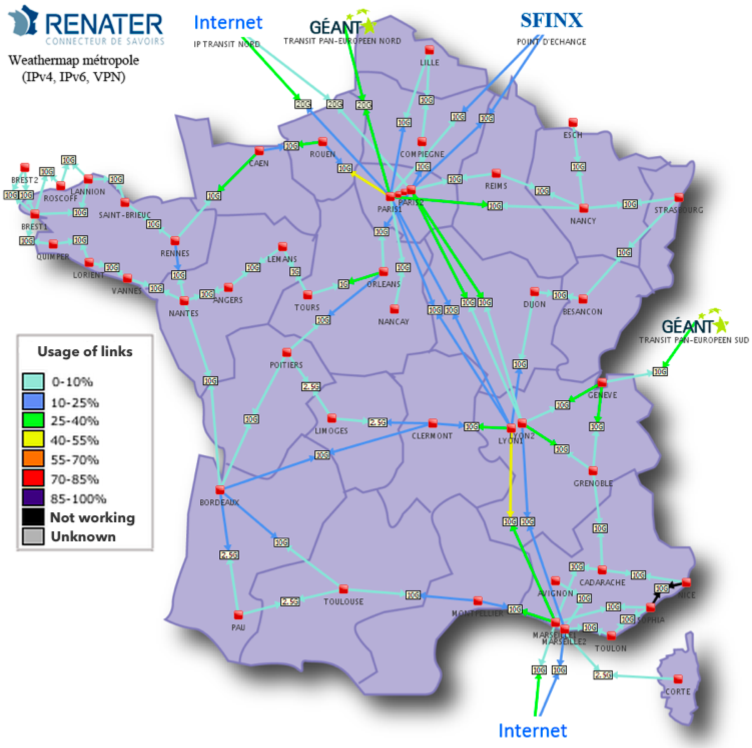

Figure 1: The RENATER Weather Map on May 2013, the 27th, around 4PM. Each red square corresponds to a particular point of presence (PoP) of the network. The map is available in real-time at: http://www.renater.fr/raccourci

- As mentioned before, most of the resources are underutilized (only two links are used between 45% and 55%, a few between 25% and 40%, and the majority below the threshold of 25%).
- The backbone was deployed and is renewed to match the demand: The density of points of presence (PoP, *i.e.* a small or medium-sized network center) as well as the bandwidth of each link are more important on the edge of large cities such as Paris, Lyon or Marseille.
- The backbone was designed to avoid disconnections, since 95% of the PoPs can be reached by at least two distinct routes.

1.4 Locality-based Utility Computing

Instead of building and deploying dedicated facilities, we claim that next UC infrastructures should be tightly coupled with any facilities available through the Internet, starting from the core routers of the backbone, the different network access points and any small and medium-size computing infrastructures that may be provisioned by public and private institutions. Although it involves radical changes in the way physical and virtual resources are managed, locating and operating computing and data on facilities close to the end-users is the only way to deliver highly efficient and sustainable UC services.

[2] http://www.renater.fr

From the physical point of view, network backbones provide appropriate infrastructures, *i.e.*, reliable and efficient enough to operate UC resources spread across the different PoPs. Ideally, UC resources would be able to directly take advantage of computation cycles available on network active devices, *i.e.* those in charge of routing packets. However, leveraging network resources to make external computations may lead to important security concerns. Hence, we propose to extend each PoP with a number of servers dedicated to hosting virtual machines (VMs). Because it is natural to assume that the network traffic and UC demands are proportional, larger network centers will be completed by more UC resources than the smaller ones. Moreover, by deploying UC services on relevant PoPs, a LUC infrastructure will be able to natively confine network exchanges to a minimal scope, minimizing altogether the energy footprint of the network, the impact on latency and the congestion phenomena that may occur on critical paths (for instance Paris and Marseille on RENATER).

From the software point of view, the main challenge is to design a comprehensive distributed system in charge of turning a complex and diverse network of resources into a collection of abstracted computing facilities that is both reliable and easy to operate.

2 The DISCOVERY Objectives

The main target of this initiative is the definition of a complete distributed system in charge of turning a complex and diverse network of resources into a collection of abstracted computing facilities that is both easy to operate and reliable.

2.1 Overview

By designing an advanced system that offers the possibly to operate in a unified manner a large number of UC resources spread throughout distinct sites, ISPs as well as academic and private institutions in charge of operating a network backbone will be able to build an extreme-scale LUC infrastructure with a limited additional cost. Instead of redeploying a complete installation, they will be able to leverage IT resources and specific devices such as computer room air conditioning units, inverters or redundant power supplies that are already present on each hub of their backbone.

The DISCOVERY initiative aims at making significant advances towards such a LUC operating system with the ultimate goal of allowing end-users to launch virtual environments (VEs), i.e. users' working environments, throughout a DISCOVERY infrastructure as simply as they are used to launch processes on a local machine, i.e. without the burden of dealing with resources availability or location.

In addition to considering *locality* as a primary concern, the novelty of the LUC OS proposal is to consider the VM as the basic object it manipulates. Unlike existing research on distributed operating systems designed around the process concept, a LUC OS will manipulate VMs throughout a federation of widely distributed physical machines. Virtualization technologies abstract out hardware heterogeneity, and allow transparent deployment, preemption, and migration of virtual environments (VEs), *i.e.* a set of interconnected VMs. By dramatically increasing the flexibility of resource management, virtualization allows to leverage state-of-the-art results from other distributed systems areas such as autonomous and decentralized systems.

2.2 Scientific/Technical Challenges

Similarly to traditional operating systems (OSes), a LUC OS will be composed of a significant number of mechanisms. Trying to identify all of them and establishing how they interact is an on-going work that will be finalized during the first months of the project. However, in order to reach the goal of delivering a unified system in charge of operating a complex and diverse infrastructure into a LUC platform, we have already identified that the following objectives should be considered when designing a LUC OS:

- Scalability: a LUC OS must be able to manage hundreds of thousands of virtual machines (VMs) running on thousands of geographically distributed computing resources, including small and medium-sized computing facilities as well as any idle resource that their owner would make available. These resources might be highly volatile, especially if the LUC infrastructure allows to include resources hosted by end-users.
- Reactivity: To deal with the infrastructure's dynamicity, a LUC OS should swiftly handle events that require performing particular operations, either on virtual or on physical resources, with the objective

of maximizing the system utilization while meeting QoS expectations of VEs. Reconfiguring VEs over distributed resources, sometimes spread across wide area networks, or moving VMs, while preserving their active connections, are examples of operations that should be performed as fast as possible.

- Resiliency: In addition to the inherent dynamicity of the infrastructure, failures and faults should be considered as the norm rather than the exception at such a scale. The goal is therefore to transparently leverage the underlying infrastructure redundancy to (i) allow the LUC OS to continue working despite node failures and network disconnections and (ii) to provide snapshotting as well as high availability mechanisms for VEs.
- Sustainability: Although the LUC approach natively reduces the energy footprint of UC services by minimizing the network impact, it is important to go one step further by considering energy aspects at each level of a LUC OS system and propose advanced mechanisms in charge of making an optimal usage of each source of energy. To achieve such an objective, data related to the energy consumption of the VEs and the computing resources as well as the environmental conditions (computer room air conditioning unit, localization of the site, etc.) should be taken into account by the system.
- Security and Privacy: Similarly to resiliency, security affects the LUC OS itself and the VEs running on it. For the LUC OS security, the goals are to (i) avoid attacks on the P2P layers, (ii) create trust relationships between different locations, (iii) include security decision and enforcement points in the LUC OS and (iv) make them collaborate to provide a secured infrastructure. For the VEs security, we need to provide users with a way to express their security requirements. The LUC OS security decision and enforcement points will collaborate to enforce these requirements.

In addition to the aforementioned objectives, targeting a distributed system where VM is the elementary granularity requires to deal with important issues regarding the management of the VM images. Managing VM images in a distributed way across a WAN is a real challenge that will require to adapt state-of-the-art techniques such as replication and deduplication. Also, several mechanisms of a LUC OS must take into account VM images' location, for instance to allocate the right resources to a VE or to request VM images prefetching to improve deployment performance or VM relocations.

Amongst the numerous scientific and technical challenges that should be addressed, the lack of a global view of the system introduces a lot of complexity. In order to tackle it while addressing the above-mentioned challenges, we claim that internal mechanisms of a LUC OS should be based on decentralized mechanisms specifically designed for it. These techniques should provide mechanisms which are fully decentralized and autonomous, so to allow self-adapting control and monitoring of complex large-scale systems. Simple locality-based actions by each of the entities composing the system can lead to the global emergence of complex and sophisticated behaviors, such as the self-optimization of resource allocation, or the creation of decentralized directories. These techniques are starting to be used in well-known large systems. As an example, the Amazon website relies on its Dynamo service [DHJ+07], based on fully-decentralized mechanisms, to create massive scale distributed indexes and recover from data inconsistencies. Facebook's Cassandra massive scale structured store [LM10] also leverages P2P techniques for its core operation. In a LUC OS, decentralized and self-organizing overlays will enable to reflect the current state of both virtual and physical resources, their characteristics and availabilities. Such information is mandatory to build higher mechanisms ensuring the correct execution of VEs throughout the whole infrastructure.

2.3 Background

Several generations of UC infrastructures have been proposed and still co-exist [FK11]. However, neither Desktop, nor Grid, nor Cloud Computing platforms provide a satisfying UC model. Contrary to the current trend that promotes large offshore centralized DCs as the UC platform of choice, we claim that the only way to achieve sustainable and highly efficient UC services is to target a new infrastructure that better matches the Internet structure. Because it aims at gathering an unprecedented amount of widely distributed computing resources into a single platform providing UC services close to the end-users, a LUC infrastructure is fundamentally different from existing ones. Keeping in mind the aforementioned objectives, recycling UC resource management solutions developed in the past is doomed to failure.

As previously mentioned, our vision significantly differs from hybrid Cloud Computing solutions. Although these research activities address important concerns related to the use of federated cloud platforms such as the standardization of the interfaces for supporting cooperation and resource sharing over Cloud federations, their propositions are incremental improvements of the existing UC models. From our point of view, hybrid and cloud federation investigations are comparable in some ways to previous works that have been done for Grids and where, the purpose of the Grid middleware is to interact with each resource management system composing the Grid [BRC10, RBL+09, ZYT+12]. By taking into account network issues in addition to traditional computing and storage concerns in Cloud Computing systems, the European SAIL project[3] is probably the one which targets the biggest advances in comparison with previous works on Grid systems. Concretely, this project investigates new network technologies in order to provide end-users of hybrid/federated Clouds with the possibility to configure and virtually operate the network backbone that interconnects the different sites they use [MSS+12]. More recently, the *Fog Computing* concept has been proposed as a promising solution to applications and services that cannot be put into the cloud due to locality issues (mainly latency and mobility concerns) [BMZA12]. Although it might look similar to our vision as they propose to extend the Cloud Computing paradigm to the edge of the network, *Fog Computing* does not target a unified system but rather proposes to add a third party layer (*i.e.* the *Fog*) between cloud vendors and end-users. In our vision, UC resources (*i.e.* Cloud Computing ones) should be repacked in the different network hub of backbones and operated through a unified system, *i.e.* the LUC OS. As far as we know, the only system that investigated whether a widely distributed infrastructure can be operated by a single system, was the XtreemOS Project [Mor07]. Although this project was sharing some of the goals of the LUC OS, it did not investigate how the geographical distribution of resources can be leveraged to deliver more efficient and sustainable UC infrastructures.

To sum up, we argue for the design and the implementation of a distributed OS, manipulating VEs instead of processes and considering locality as a primary concern. Referred as a LUC Operating System, such a system will include most of the mechanisms that are common to actual UC management systems [clo, nim, opea, opeb, Low11, MVML12]. However, each of them will have to be rethought in order to leverage P2P algorithms. While largely unexplored for building operating systems, P2P/decentralized-based techniques have the potential to achieve the scalability required for LUC systems. Using all these technologies for establishing the foundation mechanisms of massive-scale distributed operating systems will be a major breakthrough from current static, centralized or hierarchical management solutions.

3 Preliminary Work Program

3.1 A Mutli-Agent Peer to Peer System

The DISCOVERY system will rely on a multi-agent peer-to-peer system deployed on each physical resource composing the LUC infrastructure. Agents are autonomous entities that collaborate to efficiently use the LUC resources. In our context, efficiency means that a good trade-off is found between satisfying user's expectations, ensuring reliability, reactiveness as well as availability of the services while limiting the energy consumption of the system and providing scalability. We propose thus to leverage P2P techniques, that allow self-* properties, such as self-adaptation and self-repairing of overlays. To reduce the management complexity as well as the design and the implementation of the different mechanisms that are mandatory, we strongly support to use micro-kernel concepts. Such an approach should enable to design and implement services at higher level while leveraging P2P mechanisms at the lower ones. Furthermore, to address the different objectives and reduce the management complexity, we also underline that self-* properties should be present at every level of the system. We think that relying on a multi-agent peer-to-peer system is the best solution to cope with the scale as well as the network disconnections that may create temporary partitions in a LUC platform.

In DISCOVERY, each agent has two purposes: (i) maintaining knowledge based on the LUC platform composition (ii) ensuring the correct execution of the VEs. Concretely, the knowledge base will consist of overlays that will be used for the autonomous management of the VEs life cycle. This includes the configuration, deployment and monitoring of VEs as well as the dynamic allocation or relocation of VMs to adapt to changes in VEs requirements and physical resources availability.

[3] `http://www.sail-project.eu`

3.2 Work Packages

Designing and implementing the DISCOVERY system will require a clear organization of the work as there are strong dependencies between the different mechanisms of the system. To avoid any fruitless development and to ensure the maximal usability of the work achieved within the project, we propose to build the system by successive steps following a bottom-up approach driven by five scientific/technical work packages:

- Mechanisms related to physical resource localization and monitoring,
- Mechanisms related to VEs management,
- Mechanisms related to the VM images management,
- Mechanisms related to reliability,
- Mechanisms related to security and privacy.

To prevent any deviation and limit the problems that might occur if a specific element of the work plan should fail to deliver its promises, we will leverage an additional work package dedicated to the design of the architecture, its integration and the API that will be exposed to the applications so that they can benefit from the LUC infrastructure.

The purpose of this organization is to group similar concerns together. First, it clearly states the contact point for each expertise area, in order to improve interactions within the project. Second, it will allow to address each WP in an independent way once the basic architecture of the whole system and the different APIs will have been defined. However, due to the strong dependencies between WPs and in order to limit the problems that might occur if a specific element of the work plan should fail to deliver its promises, the design of the DISCOVERY system will not be a linear process. Instead, it will include several iterations between WP1 and the other ones, to evolve from a basic solution to a complete one. Figure 2 shows this organization.

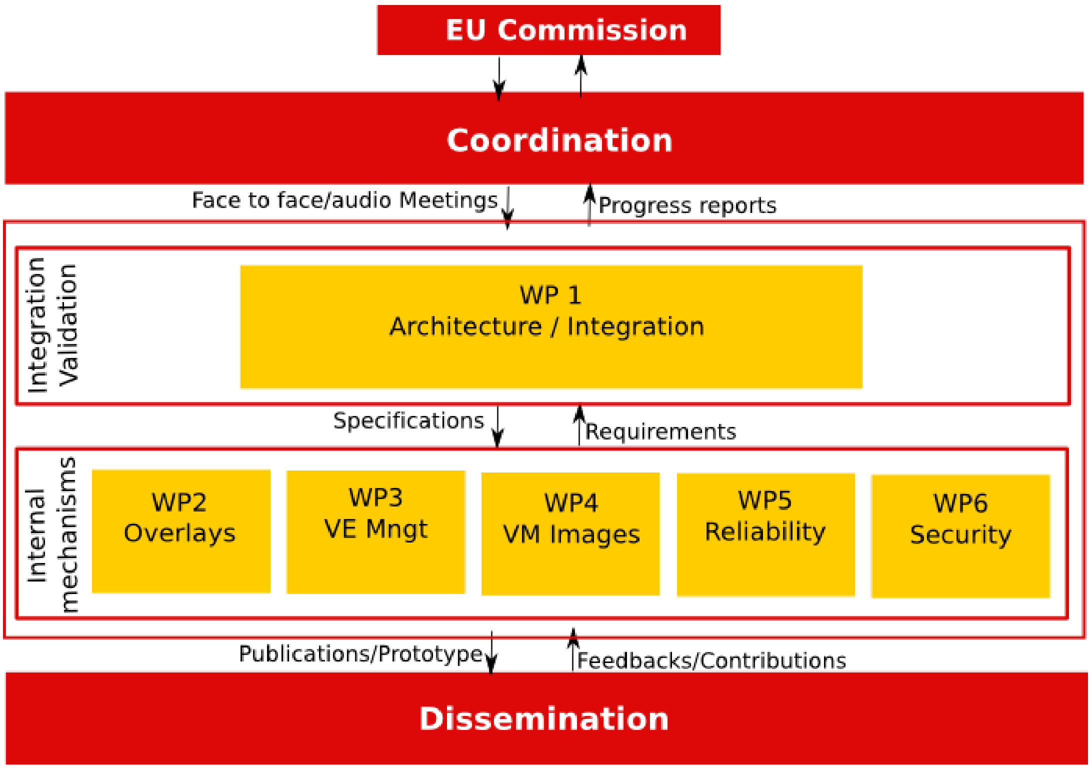

Figure 2: Work Package Organization

3.3 Consortium as a Whole

Currently, the DISCOVERY consortium is composed of 3 research group, Inria (France), the storage research group from BSC (Spain),and the CRS4 in Sardinia. Three European NRENs and one telco complete the consortium: RENATER (French NREN), PSNC (Polish NREN), GRNET (Greek NREN) and Orange Labs (France). This geographic pluralism coupled with the privileged relationships the participants in the project enjoy with the different communities guarantee a wide dissemination of the DISCOVERY results.

From the scientific point of view, the consortium gathers the expertise of well-recognized research groups dealing for several years with Cluster/Grid/Cloud/P2P Computing challenges. By integrating important and well-established network operators, we point out that the consortium has the complete knowledge necessary to reach the objectives of DISCOVERY project. This association of strong research groups and major network partners assures prime scientific results to the DISCOVERY project while securing a wide possible exploitation impact.

Finally, it is worth nothing that most of the members are used to collaborate directly around formal or informal collaborations. From the industrial point of view, Inria and Orange Labs have been collaborating for several years around different projects such as ANR Selfware, ANR SelfXL and more recently the OpenCloudware FSN project. Regarding RENATER, several members of the consortium are used to interact with the main technical leaders through the Grid'5000 project. At the European level, ASCOLA and BSC have been taking part the European Marie Curie Initial Training Network SCALUS, SCALing by means of Ubiquitous Storage since 2009. In (Lèbre, et al. March 2011), ASCOLA with CRS4 conducted a prospective study that sketch out the premises of the current DISCOVERY proposal. All these fruitful collaborations are excellent indicators for the successfully achievement of the DISCOVERY project.

References

[AFG+10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A View of Cloud Computing. *Commun. ACM*, 53(4):50–58, April 2010.

[And03] Odlyzko Andrew. Data networks are lightly utilized, and will stay that way. *Review of Network Economics. 2(3)*, 2003.

[BAM10] Theophilus Benson, Aditya Akella, and David A. Maltz. Network traffic characteristics of data centers in the wild. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC'10, pages 267–280, New York, NY, USA, 2010. ACM.

[BMZA12] 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*, MCC '12, pages 13–16, New York, NY, USA, 2012. ACM.

[BRC10] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. In *Proceedings of the 10th international conference on Algorithms and Architectures for Parallel Processing - Volume Part I*, ICA3PP'10, pages 13–31, Berlin, Heidelberg, 2010. Springer-Verlag.

[clo] CloudStack, Open Source Cloud Computing.

[DHJ+07] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: Amazon's Highly Available Key-Value Store. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, SOSP '07, pages 205–220. ACM, 2007.

[FK11] I. Foster and C. Kesselman. *Advances in Parallel Computing - Volume 20: High Performance Computing: From Grids and Clouds to Exascale*, chapter The History of the Grid. IOS Press, 2011.

[Fos11] I. Foster. Globus Online: Accelerating and Democratizing Science Through Cloud-Based Services. *Internet Computing*, 2011.

[GC13] Jodie Van Horn Gary Cook. How dirty is your data ? Greenpeace International Report, 2013.

[GHMP08] Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. The Cost of a Cloud: Research Problems in Data Center Networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, December 2008.

[Gro12] IEEE 802.3 Ethernet Working Group. IEEE 802.3TM Industry Connections Ethernet Bandwidth Assessment, July 2012.

[LM10] Avinash Lakshman and Prashant Malik. Cassandra: A Decentralized Structured Storage System. *SIGOPS Oper. Syst. Rev.*, 44(2):35–40, April 2010.

[Low11] Scott Lowe. *Mastering VMware vSphere 5*. Wiley Publishing Inc., Oct. 2011.

[Mor07] Christine Morin. XtreemOS: A Grid Operating System Making your Computer Ready for Participating in Virtual Organizations. In *Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, ISORC '07, pages 393–402. IEEE Computer Society, 2007.

[MSS+12] Paul Murray, Azime Sefidcon, Rebecca Steinert, Volker Fusenig, and Jorge Carapinha. Cloud Networking: An Infrastructure Service Architecture for the Wide Area, July 2012.

[MVML12] R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente. IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures. *Computer*, 45(12):65–72, 2012 2012.

[nim] Nimbus is Cloud Computing for Science.

[opea] Open Source Data Center Virtualization.

[opeb] The Open Source, Open Standards Cloud.

[RBL+09] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Cáceres, M. Ben-Yehuda, W. Emmerich, and F. Galán. The Reservoir Model and Architecture for Open Federated Cloud Computing. *IBM J. Res. Dev.*, 53(4):535–545, July 2009.

[ZYT+12] Han Zhao, Ze Yu, Shivam Tiwari, Xing Mao, Kyungyong Lee, David Wolinsky, Xiaolin Li, and Renato Figueiredo. CloudBay: Enabling an Online Resource Market Place for Open Clouds. In *Proceedings of the 2012 IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, UCC '12, pages 135–142. IEEE Computer Society, 2012.