Cloud Computing for Science and the Heritage from VENUS-C

Fabrizio Gagliardi

Microsoft Research, Av des Morgines, 12
CH-1213 Petit-Lancy
Geneva, Switzerland

Keywords: Cloud Computing, European Union FP7 projects, use cases

1. Introduction

Cloud Computing has become a very popular and successful platform for commercial computing. In contrast to Grid computing which was developed by computer scientists essentially for the benefit of the scientific community, Cloud Computing came with a strong and commercial business model which has gradually extended to scientific applications.

The talk will discuss and contrast the two models and give some practical examples of how cloud computing can help a broad scientific community (sometimes referred as the *long tail* of the computational scientific community distribution) in doing better science with a most cost effective business model. Microsoft Research joined forces with other European industrial and scientific partners to propose to the EU a demonstration of Cloud Computing for science in Europe. The project started in June 2010 and ended in May 2012 with considerably successful results, demonstrated through a validation study. The approach taken and the results will be presented. The talk will conclude with an outlook to some follow on activity.

2. Cloud Computing

Amazon decided to sell over the Internet the excess computing power and data storage they had in their huge data centres. This was made possible by the advances of Computer Science in virtualisation. By virtualising computer resources and data storage it is now possible for everyone to create their own computing infrastructure hosted by a major IT provider. This is called Infrastructure as a service (IAAS) and it is exposed as web services over the Internet. The virtualisation can go much further and extend all the way to the application and provide end users over the Internet access through a Software as a Service (SAAS) model. In principle, this could also have been implemented with the earlier distributed grid computing model. However, the key difference here is that rather that aggregating a large number of heterogeneous clusters, running different operative systems and managed by different authorities, in Cloud Computing massive data centres, which are all homogeneous and operated by the same authority, can offer an economy of scale of unprecedented dimensions and huge scalability. New programming models derived from web search technology such as Map/Reduce have been successfully applied to scientific problems.

3. Computing for the long tail

The VENUS-C [1] project was designed to demonstrate how Cloud computing could help the long tail of computational science and at the same time provide public funding agencies with a new and more cost effective way to provide the projects they support with virtual computing resources rather than physical hardware. Such hardware is both costly to

operate and hard to maintain over time. It is also becoming possible to offer more effective ways of preserving data generated by projects by virtualising the computing infrastructure. Furthermore, by developing easy tools for computational science such as the Generic Worker [1] and COMPSs [2], VENUS-C provided new communities of scientists traditionally excluded by High Performance Computing, because of the complexity of access or cost, with frameworks that enable them to do better science.

4. VENUS-C approach

VENUS-C has been a user-centric project which started from an analysis of a set of representative scientific user communities. Several studies [3][4][5][6][7] have concluded that there is a lack of user-friendly programming models, standardization, sharing capabilities and other specific features in current cloud computing offerings, which lead to the design of a integrative software architecture and the adaptation and deployment of a set of components that could fulfill such requirements, both those from literature and from the analysis of VENUS-C communities. After the implementation and deployment, user communities validated those components using their adapted applications in production.

4.1. VENUS-C User community

VENUS-C covered a total of 27 pilot applications, targeting 11 scientific disciplines from 10 European countries. Figure 1 shows the distribution among communities, where the highest represented community is “Molecular, Cellular and Genetic Biology”. The different pilots adopted and showcased the use popular tools in the scientific community in the cloud: BLAST, Jorca, R, Matlab, Energy+, Autodock, MetaPiga, QSAR, SPM, gCube, GAP and twitter among others, covering an existing user community of above 5.000 users.

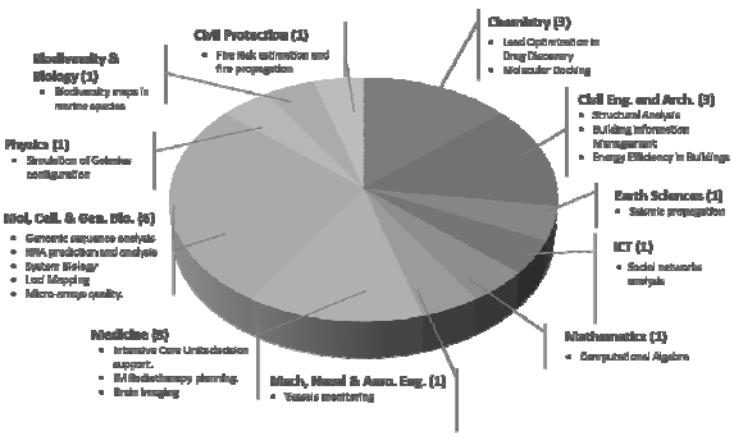

Fig. 1. User Communities in VENUS-C.

5. VENUS-C architecture

The VENUS-C Architecture is structured around five technical areas: ‘Data Management’, ‘Programming Models’, ‘Application Security’, ‘Monitoring, Accounting and Billing’ and ‘Traffic Redundancy Elimination’. Data management and programming

models are the core capabilities for scientific computation: *handling data* and *processing code*.

Security, accounting/billing and traffic redundancy elimination, cover non-functional requirements which are orthogonal to the core of scientific applications. The overall system has to be secure, provide accounting and billing information, and transfer data in and out of the cloud as efficiently as possible.

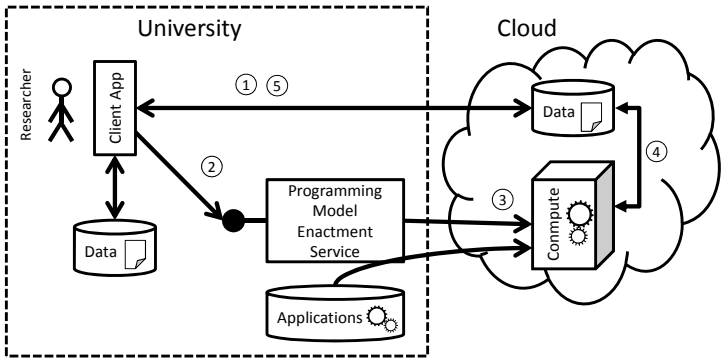

Fig. 2. The basic use case considered for the VENUS-C components: a researcher working in a university that offloads scientific computations to the cloud. The main parts of this figure are the scientific application and the Programming Model Enactment Service (PMES).

A scientific application consists of two parts:

- **The client application** (UI frontend), Uploads scientific data into the cloud (Step 1), Submits jobs to the PMES (Step 2), Retrieves the results of the jobs and visualizes them (Step 5).
- **The algorithmic parts** (compute intensive scientific algorithms) are stored in a storage service accessible from the compute instances in the cloud, are executed in the cloud (Step 4).

The Programming Model Enactment Service (PMES) exposes OGF BES[1]/JSDL[2]-compliant web service interface, to which client applications can submit jobs (Step 2).

The VENUS-C project has created reference implementations of two programming models:

- The '**Generic Worker**' pattern for batch processing.
- The Barcelona Supercomputing Center's **COMPSs** (COMP Superscalar) for batch processing and task level parallelisation of existing sequential applications.

Key differences between the Generic Worker and COMP Superscalar:

- Generic Worker nodes pull work items in a data-driven fashion, so that the programming model enactment service merely queues new work items.
- The COMPSs programming model enactment service pushes work items to the available worker nodes in an orchestrated fashion.

[1] http://www.ogf.org/documents/GFD.108.pdf
[2] http://www.gridforum.org/documents/GFD.136.pdf

6. Conclusions and future work

The programming models are a major contribution of the VENUS-C project to the scientific community. In conjunction with the data access mechanisms, these programming models provide researchers with a suitable abstraction for scientific computing on top of plain virtual machines. The VENUS-C Platform has been designed addressing the scenario requirements, carefully defining which requirements needed to be addressed as part of the VENUS-C infrastructure.

VENUS-C user community has carried out an in-depth evaluation of the components developed in the VENUS-C project. Through the use of more than 1.5 Million of core hours, users have tested the different versions of the components available for execution and data access. The overall impression of the components (at versions released in April) was good (from 3.92 to 4.37, in a scale of 5), with very good evaluation marks for the fulfillment of user requirements, ease the adaptation of applications and avoid vendor lock-in, expressed as interoperability, completeness and learning curve.

The evaluation of the benefits perceived have revealed an increase of performance with respect to conventional approaches, improved business opportunities (this is what the users perceived best) among other benefits observed, such as the increased volume for problem sizes and the simpler adaptation process.

Through this evaluation, the project contributed to understand how public cloud infrastructures (i.e. MS-Azure) are useful and practical for scientific research, and how the use of VENUS-C components improves user experience. Open-source, private infrastructures have also been tested, with similar conclusions regarding user experience. These clearly denote that adapting the applications in VENUS-C is worthwhile, since the learning curve in using the components is low, and the availability of an infrastructure is guaranteed by the use of public, commercial platforms. However, since adaptations can rely on a totally open-source stack that can be installed at user-premises, vendor lock-in is then minimised.

A new initiative to follow up from VENUS-C, named C4S (Cloud for Science) is now being launched by Microsoft Research.

References

1. VENUS-C, Virtual Multidisciplinary Environments Using Clouds, `www.venus-c.eu` (2012)
2. Lezzi, D., et al. Enabling e-Science Applications on the Cloud with COMPSs. Euro-Par 2011: Parallel Processing Workshops 7155, 25-34 (2012).
3. Cloud Computing Use Case Discussion Group, "Cloud Computing Use Cases white paper", Version 4.0, 2 July 2010.
4. Distributed Management Task Force. "Use Cases and Interactions for Managing Clouds - A White Paper from the Open Cloud Standards Incubator. s.l.", DMTF, June 2010. DSP-IS0103.
5. Peter H. Deussen, Klaus-Peter Eckert, Linda Strick, Dorota Witaszek, "Cloud Concepts for the Public Sector in Germany – Use Cases", Fraunhofer-Institute for Open Communication Systems FOKUS, 2011.
6. Magellan Project, "Magellan Final Report", December 2011.
7. Dawn M. Leaf, "Cloud Computing Use Cases", `http://www.nist.gov/itl/cloud/use-cases.cfm`