From grid to cloud computing service model: new business model for web services based computing

Serena Pastore
INAF – Astronomical Observatory of Padova
Vicolo Osservatorio 5 – 35122 - PADOVA
ITALY
serena.pastore@oapd.inaf.it

Abstract: - Grid and cloud computing are two models of distributed computing systems that are based on the same objective of sharing heterogeneous resources in terms of networks, compute, storage, software, platform and infrastructure, in a transparent way for the end users. They manifest similarities and was born to satisfy the same needs, even in different times, but their adoptions in profit or not-profit organizations followed different paths. In Europe grid computing emerged thanks to several European funding research projects and demonstrations, which gave rise to the creation of a European grid infrastructure (EGI), even if the core software of architecture was first developed as a US project of the Globus Alliance. Cloud computing, instead, exploded through wide commercial offerings as a dynamic and scalable computing and storage platform that more closely responds to the demands and needs of users. The model is now embraced by different organizations, since it seems much easier to adopt and use and could implement economies of scales considering investments in computing infrastructure. With the aim to demonstrate that the two models could converge rather than compete, this paper analyses the two infrastructures to evaluate their use in a typical not-for-profit environment such the educational or public research institutes, focusing on both the benefits and problems.
Keyword:.

Key-Words: - Distribute computing systems, Grid computing, Grid middleware, Cloud computing, Cloud services

1 Introduction

Grid [1] and cloud computing [2] are two different models of distributing computing systems [3] that have been adopted in the organizations with different results. Both have been developed with the common objective of sharing heterogeneous distributed resources, among which network, storage, compute, software, platform and infrastructure, users, hiding the complexity of the underlying hardware and software architectures used to implement them. Such systems were needed for the growing needs of data and applications in the Internet network that require high computational, storage, and network resources. The software includes not only computing-intensive or storage-intensive applications typical of scientific application, but also web frameworks, interactive and real-time requiring a highly variable number of requests and a considerable amount of storage for the exchanged data. Sharing derived from the need to exploit expensive existent hardware not used to its full potential and the acquisition costs of more powerful machines.

The concept of shared distributed resources is not new, but these technologies take advantage of communication networks improvement and the development of specific technologies (e.g., virtualization, web services and service oriented architectures, and the concepts of the web 2.0) [4]. The resources in an extended meaning (e.g., processing and storing capabilities, networks, software and applications, platforms and infrastructures) for these models are provided as always online, accessible services with a web user interface to different targets. Exploiting the spread of multi-services communication networks, both grid and cloud methods would provide distributed resources as services to the different category of users in a transparent way. However, the two models shows different business models and application domains, leading to a different adoption according the type of organization. The paper gives an overview of these distributed systems models that is grid and cloud computing. It provides an analysis of the state of art solutions provided as grid computing and cloud computing implementations to verify their use mainly in an educational and public research institution characterized by a great demand of computing services, but very few financial capacities in such investment. Given the existence of a European grid infrastructure (EGI) [5], to which a lot of European educational and research institutions belong, the paper evaluates this production infrastructure, comparing to similar cloud offering. It analyzes the possibility that the two solutions could converge rather than compete. Moreover, as regards cloud solutions, it consider a comparison between choosing one of the several commercial offerings against the implementation of a private cloud as in-house offerings using open source cloud software. The aim is to provide an overview of benefits and issues in the adoption of these utilities from an economical perspective.

2 Grid vs. cloud implementation

The current increase of the type and amount of information and data to be analyzed, processed, and transformed, and the services or applications accessible on the Internet, have resulted in higher requirements for information systems and their hardware and software components. Such requirements includes hardware (faster processors, larger storage, and the online availability and always accessibility of systems and data) and software performance (software and services available anywhere and anytime). To meet such demands, hardware and software computing environments have resulted in powerful, reliable, redundant and efficient computer systems (i.e., mainframes in a sever farms), aggregated in local networks and centrally managed homogeneous resources. The further application of collaboration concepts has led to distributed networked systems as a set of independent computer networks sharing according to the heterogeneous resources of different architectures, models or paradigms distributed everywhere. Moreover, the virtualization technologies allowed at extending the concept of resource to include infrastructure (Infrastructure as a Service, IaaS), platform (Platform as a service, SaaS), software (Software as a Service) rather than storage, computation or network.

A distributed computing system is a type of distributed system that shares computational resources. It is applied for running applications that require large computing capacity and/or storage (e.g., scientific applications, but also web services that require billions of requests per day with variable peaks and billions of GB to be stored). The underlying computing systems should support higher computing capability (more processors and memory to perform faster calculations) and/or greater storage capacity (mass memory to store large amounts of data) and allowing the use of Internet based services (e.g. web, mail, etc.) or applications (e.g. Facebook) by multiple users.

Grid and cloud infrastructures are implementations of distributed computing systems. They are modelled with a layered architecture (Fig. 1) that distinguishes a lower fabric layer, an upper application layer and an intermediate layer called middleware. The central layer provides all the functionalities needed for the applications to use the networked distributed resources through specific software and libraries.

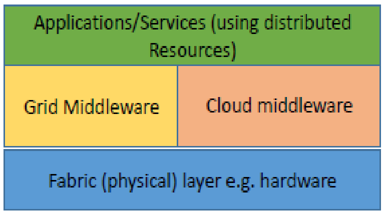
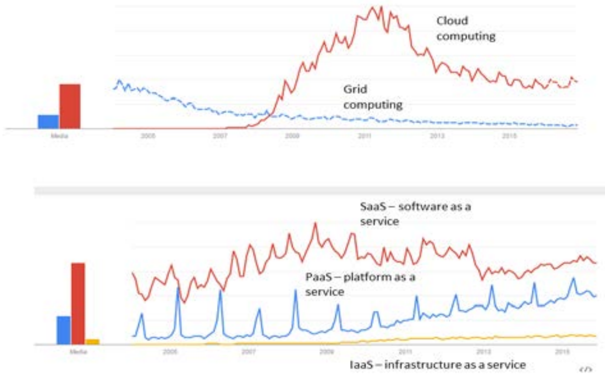


Figure 1. Layered architecture of a distributed computing system

Both models refer to a hardware and software infrastructure that aggregates heterogeneous distributed resources. However, they are based on two different business models that are so applied in different application domains. This aspect led to a different success in their adoptions by organizations. Fig. 2 shows a graph based on the data of Google trends [6] that analyses the occurrences of grid and cloud computing highlighting the diffusion and success of the two models. Grid computing was most successful around 2005 (the first middleware is dated 1998), while cloud computing was most successful around 2011. The higher curve shows the greater success of cloud computing.



ISBN: 978-1-61804-362-7

Figure 2. Grid and cloud computing keywords, and cloud different services (IaaS, PaaS and IaaS) according Google trends

The same figure shows also the success of the different cloud services offering software, platform of infrastructure as a service. Differences between grid and cloud are mainly in the business model, but this depends on their implementation. Grid computing relies on a community-based model. Users can only access grid resources, if they belong to a Virtual Organization (VO), [7] that is, an organization dynamically collecting individuals, institutions and resources. Each VO provides a set of resources to the infrastructure and the other VOs in accordance with security policies as regards authentication of users and authorization mechanisms for using the resources. A cloud has a commercial model. A user accesses a cloud services provided by several heterogeneous providers using a pay-per-use or pay-as-you-go model, that is, he pays directly for the used resources or pays according to a monthly or yearly plan.

A rough comparison between the two utilities, as analyzed for a not for profit organization like a public educational or research institute, with few funds and needs of resources for scientific computing or data-intensive applications or web applications with high but random requests.

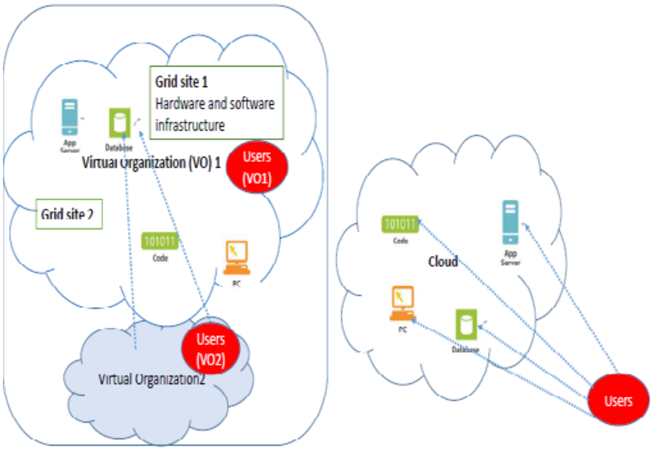


Figure 3. Comparison between grid and cloud structures

2.1 The grid model in Europe

2.1. Grid computing and its implementation: the European Grid Infrastructure

A computational grid according to the definitions given by the two progenitors of these systems (Ian Foster and Kesselman) is "a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities" [1] or "a flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources, what we refer to as Virtual Organizations" [1]. A grid is thus logically structured in VOs that collect users, organizations, shared resources for a common goal (e.g. a project or as part of the same institution) in a secure way. Since its main application in the research environment a grid is used with the concept of e-science (i.e., researchers working in computationally intensive science or using huge data sets, or complex instrumentations by using geographically distributed heterogeneous resources such as High Energy Physics or Astrophysics).

A grid system is a set of coordinated and large scale-resources belonging to different administrative domains, which, with a measurable quality of services (QoS), provides faster response times, reliability and safety. Based on standard protocols and interfaces, it ensures interoperability between different software implementations. The grid resources are implemented as a set of hardware and software (the grid middleware) that allow you to integrate, manage and provide the heterogeneous resources to VOs users. The grid middleware, as shown in the Fig.4, provides all the necessary services for the coordination and use of resources (e.g., from resource management, to monitoring and indexing) and the secure access to such resources and services.

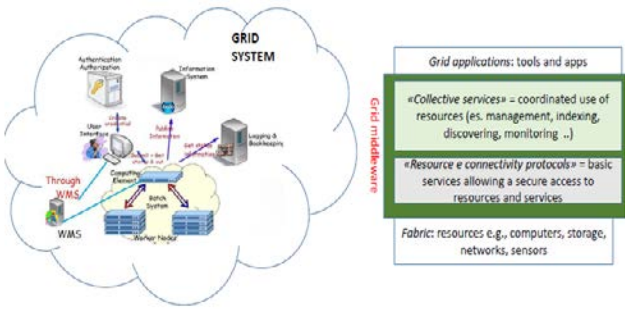


Figure 4. Grid middleware components and system

The same software provides user interfaces as command line interface (CLI) or graphical (GUI) and as a web-based interface. A site grid is generally made up of several nodes, the physical or virtual machines implementing both the resources (e.g. computing and storage) and the other software providing the middleware software [8].

ISBN: 978-1-61804-362-7

Several grid middlewares have been developed: the first was the Globus toolkit [9] (now at the GT5 version) that is also the de facto standard for grid development. It is an open source software toolkit developed by the Globus Alliance [10] (USA) as a set of services, software and libraries as technologists for sharing tools and resources securely between geographical domains for single institutions without sacrificing local autonomy. Other grid middleware have been developed in European research projects. Several countries or research communities have created, and funded, grid initiatives and developed specific middleware, even if some of them are based on the Globus Toolkit. For example, the CERN (European Organization for Nuclear Research) started a grid platform that was implemented also in the Italian research community by the INFN (National Institute for Nuclear Physic) [8] to create a computing infrastructure for data management issues related to the Large Hadron Collider (LHC) project [11] through the LHC computing software. This infrastructure evolved as a European infrastructure through several EU projects (Fig. 6) aiming at providing a common infrastructure for European scientists in different fields (e.g., high energy, astrophysics, life sciences). The first project, called DataGrid aimed at building "a computing infrastructure for intensive computing and analysis of shared large-scale databases sized from hundreds of Terabytes to Petabytes, across widely distributed scientific communities".

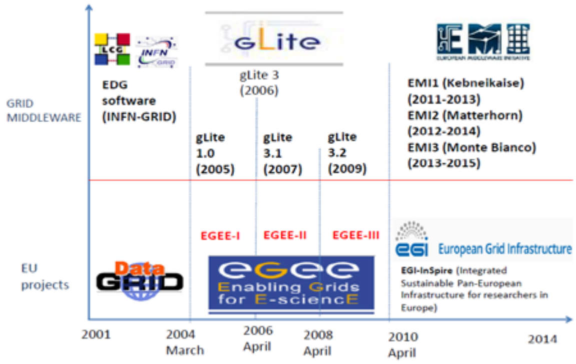


Figure 5. Evolution of grid initiative and middleware in the European framework

The infrastructure went into production in 2004 with a capacity above forecasts (about 4000 CPUs located in approximately 30 European and other locations). The next projects, called Enabling Grids for E-sciencE (EGEE I-II-III)) have further developed the infrastructure to service the various scientific communities. In 2013, the infrastructure consisted in approximately 372,00 CPU cores located at 315 resource centers in 56 countries for activities dedicated for the most part to high-energy physics [12]. Now the EGI infrastructure is "a federation of over 350 resource centers and coordinated by EGI.eu". The middleware used in this infrastructure has followed a number of changes including with regard to the name. The first middleware, called EGD (European Datagrid software), was the aggregation of different software and a customized version was developed by each organization participating in the infrastructure (e.g., the INFN with the so-called INFN-GRID middleware [8]). Then the middleware was enhanced, resulting in the gLite software [13] adopted during the EGEE projects. Next to this grid initiative, other grid initiatives were developed in the same years, and with them the grid software: the NorduGrid, community of European Nordic states, developed the ARC (Advanced Resource Connector) software [13], the Uniform Interface to Computing Resources (UNICORE) middleware [13] was developed as a German grid project. Now, within EGI the effort is to provide a Unified Middleware Distribution (UMD). Such grid middleware, called European Middleware Initiative (EMI) software, is intended to consolidate, integrate, harmonize and extend solutions to the challenges that have arisen in several applications of the three main software solutions provided by the different grid platforms. As the Fig, 5 shows, EMI takes the best of the grid solutions (e.g., ARC, gLite, UNICORE and dCache as a system able to manage distributed storage), developing a collection of consolidated software components through three main releases (EMI1, EM12 and EMI3).

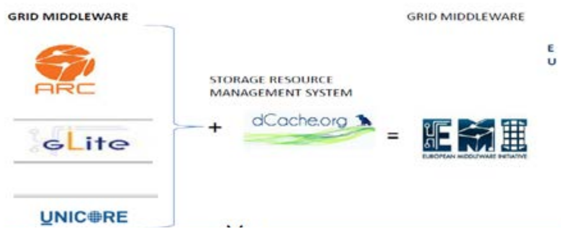


Figure 6. Actually adopted grid middleware

The grid middleware [14] is focused on four areas: computing, data, security and infrastructure. Computing and storage area refer to the resources understood in the usual sense (computing and storage) and include services related to the provision and management of such resources (the so-called computing element and storage element in grid middleware terminology with the various resource and storage management services that include scheduling and workflow). Security area includes

ISBN: 978-1-61804-362-7

services that implement the grid security model (e.g., VO membership management, authentication and authorization), while infrastructure area includes all the services that manage all the resources together (e.g., the information system and service registry, logging, bookkeeping). The benefits of this infrastructure is the secure availability of distributed resources. However, the use of the grid is not always easy and its user should have technical skills. To access the resources, the grid software (e.g. the GLite User Interface) must be installed on a physical or virtual machine to install the digital certificates in possession of the user for authentication. The user must also belong to one or more VOs to use its resources. Normally commands are given from the command line (e.g., job or application submission in a job or application that uses distributed resources), and are not really user-friendly. Some web services have been developed to give a graphical user interface to the main tasks, and attempts have been made to make the grid infrastructure more oriented towards end-users through the science gateways [15], i.e., web portals integrating the grid tools. In any case, the grid user must be tied to a research or educational environment, whereas no developed commercial grid platforms have been abandoned (e.g., Oracle grid engine).

2.2 The cloud model

Turning to the second paradigm, there are various definitions of the concept of cloud computing, but these go back to the 60s and the idea of computation organized in public-access systems (e.g., water, electricity, gas, telephony). The definition that is broadly accepted is the one given by the American Institute of Standard technology (NIST) [16]. Cloud computing is “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. The model refers to computational resources that are abstracted and provided as a service, This concept means that different service providers offer resources easily modelled as services, mainly through a web interface that hides the infrastructure details, with an approach to consumption (pay-for-use) or charging (prepaid service classes) that transforms iT as iT-as-a-service. From this concept, arises the acronym “as a service or aaS”, used to define the type of resource provided. The term “cloud” is thus intended for both the software resources (e.g., applications or SaaS, platforms or PaaS, and infrastructure or IaaS) provided as a service over the Internet and the underlying infrastructure (hardware and software system in the datacenters) that compose the cloud system. Cloud computing is a type of distributed system, consisting in a collection of interconnected and virtualized computers that, provided in a dynamic way, are presented as a single computing resource. The provision is based on service-level agreements (SLAs) established through negotiation between the service provider and user. Generally, the cloud represents this set of computer systems where resources are hosted whose structure is transparent for end-user.

Cloud computing, as Fig. 7 shows, manifests the same main features:

- It is entirely based on the Internet network with on-demand, but modificable-at-will, web access following the logic of pay-per-use;

- It provides a wide field of application targeted to different kinds of users (e.g., researchers, end users, businesses, service providers);

- It is based on economic aspects with plans for pricing and billing according to the QoS provided and the levels of service agreement (SLA) negotiation.

- It uses a pool of physical and virtual computing resources available to all users with a multi-tenant or one-to-many model.

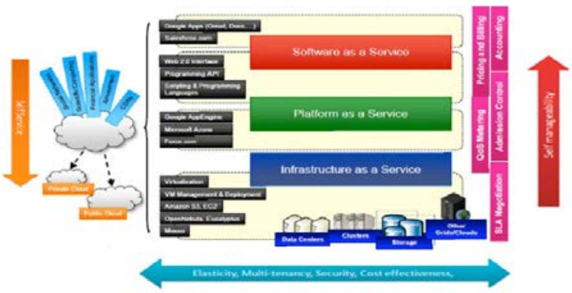


Figure 7. Cloud computing: type of services and deployment with main features

Resources are dynamically reassigned according to demand with the possibility of measurement, even if the user generally has no control or knowledge of the exact location of resources. Each user has their own slot, and customization of the resource depends on the service chosen, contrary to what happens in grid computing, where there is a community of users who can access a super computer;

- It provides flexibility and scalability so resources can be increased or released rapidly to scale the power at any time.

The software needed by a cloud depends on the distribution patterns and on the service provided to discriminate the cloud model (i.e., software, platform and infrastructure). Cloud systems could be deployed mainly in private, public, or hybrid way. The terms indicate if the services are accessed from a private network (private), from the Internet (public), or in a mixed way (hybrid). Hybrid cloud solutions are usually chose by organizations that use private cloud for core services and public cloud for other services not requiring strong security features. However, three are other kinds of distributions for study purposes, such as the community cloud.

A layered structure represents a cloud software (Fig. 8). For each layers, there are a plethora of different and often heterogeneous providers that may provide one or more cloud services as open source or commercial software.

Fig. 8. The cloud software platforms layers

Starting from the hardware layer, the cloud software platform includes the virtualization software necessary to create the hardware and software infrastructure underlying the major cloud services. Then, the cloud middleware, in analogy with the grid middleware, provides functionality for managing the entire infrastructure and therefore its services (the cloud platforms or middleware). The upper layers include the software providing main the cloud services (infrastructure IaaS, platform PaaS and software SaaS), while the last layer represent the cloud marketplace, with commercial web-based services and business apps for cloud solutions. Focusing on the central layers (the middleware and the main cloud services), there are many and varied options.

Among the open-source cloud middleware [17], most competitive seems to be Cloudstack, OpenStack, Eucalyptus and OpenNebula [18]. Each of these projects could be vendor-neutral (e.g., Cloudstack that is an Apache project) or vendor-sponsored, that is, in some way related to major commercial cloud platform management providers (e.g., Wmare, Amazon, Rackspace). In any case, each project can find its niche in the broad cloud ecosystem [19] (as happens with Linux distrinutions), and the different projects can share or integrate components with other solutions as a way to differentiate themselves in the market. As reported in the OpenNebula project [18], cloud platforms are used in two main models: for datacentre virtualization and for infrastructure provision, according a classification that is given various names by different authors. The position of the solution in the different models is the main parameter to be used in the choice. As regards the other layers, the market offers different solutions that are in some way related to the different target users of each cloud service. However, the main underlying features are that all the services are provided through the Internet network, mainly in a web-based manner, (i.e., using through whatever web browsers) thus avoiding local software installation and customization. Each cloud service that charges a fee cloud follows the pay-for-use or pay-as-you go model, usually offerings a free starting plan that gives basic functionalities according to the type of service (e.g., amount of storage for SaaS or a basic virtual machine for IaaS), providing users with the possibility to test the service before paying.

Among IaaSs providers, a market that continues to be the fastest-growing segment of the overall public cloud services market, Amazon Web services (AWS) [20] is still the leader. AWS provides a range of resources and services: EC2 (Elastic Cloud Computing) provides computing capacity in the cloud through the instances of Virtual machines (running Linux or Windows operating systems), while the storage offering uses S3 storage capacity or Elastic Block Storage. To encourage the use of such products, AWS offers a one-year free plan called AWS Free Tier that can be used to test applications in the cloud and gain experience. It requires a subscription that offers 750 hours of Amazon EC2 Linux or Windows instance usage per

ISBN: 978-1-61804-362-7

month (1GB of memory) and 5GB of S3 standard storage with 20,000 get requests and 2000 put requests. The free usage expires if the user exceeds the limits and he/she is subjected to the pay-as-you-go service rates that are calculated for each service.

In the PaaS solution, a user can make a distinction in their choice based on the programming language he/she is interested in. PaaS providers usually give platforms for a selected numbers of programming languages. For example, Windows-based programming environments and tools based on the .NET framework are provided mainly by Windows Azure [21], which, however, is not just a PaaS service, but includes different cloud services such as infrastructure and storage. Its offering also starts with a free plan for each service. Azure Website for example provides a platform to develop and deploy web applications, according four tiers (free, shared, basic and standard) that consider different hardware features related to performance (e.g., CPU cores, memory of hosted virtual machines) and a pricing plan based on a fee per hour per website instance. Other well-known providers are Google App Engine (GAE) [22] or Heroku [23]. GAE, as the provider of an entire ecosystem of applications, provides also a platform to develop and deploy web applications and services. It supports mainly Python of Java programming languages, but also Javascript and PHP web applications hosted on Google machines and thus in the cloud infrastructure. Applications can use other Google services (e.g., datastore), and so a GAE app is strictly tied to the Google environment, presenting some issues related to porting into another platform. Heroku was the first PaaS provider and is available to Facebook app developers. It was initially targeted at Ruby developers, but was extended to other languages such as PHP, Java, Python, Node.js, Clojure and Scala. Pricing is based on resources that are calculated on entities called “dynos”, units of computing power in a container where an application runs. There are different dynos configurations according to the resources available (e.g., 1x means 512MB RAM and 1xCPU share)

Finally, SaaS providers provide software as a web application and are therefore widely used. There are two main models for SaaSs: the provision of document storage in the cloud (e.g., Dropbox [24]) or the provision of online software with storage. For the second model, the web application runs on a cloud infrastructure and is accessible from various client devices through a user interface (web browser). The category of delivered software is wide and includes online productivity tools (software for word processing, spreadsheets, presentations) with Google Drive [25] for example, enterprise applications (Customer Relationship Management, CRM software) and collaboration software (email, online calendars) with Zoho online [26]. These services represent a ready-made, globally accessible and normally low-cost solutions. After an overview of the main features and solutions for the two utilities, we made a rough comparison study in order to highlight benefits and issues of using them in a research or educational environment

3 Results and analysis

The comparison, shown in Table 1, between grid and cloud computing focuses on experience in EGI production; grid and cloud providers are divided according to the kind of cloud services provided. Both are intended as a software and hardware infrastructure geographically distributed to share heterogeneous computing resources to users in a wide sense. Generally, computing resources are distributed across different sites, countries and continents in grid computing, while in cloud computing it depends on the cloud provider whether data centers are centralized in a few locations.

In a grid environment it is possible to share data in a secure way, since all security infrastructure is based on Public Key Infrastructure (PKI) and authorization mechanisms [27]. With cloud a user can access to shared resources in an easy way and with the possibility to access extra resources during peak work periods and the to release them when the resources do not need more. Grids were designed to handle large sets of jobs with short duration that produce or use large quantities of data (e.g., in physics and life science applications), while clouds best support long-term services and longer running jobs. Then as seen in Table 1, grid allows collaboration through a federated platform of distributed and collective work, and ensures ownership, since each resource provider maintains ownership of the resources shared in the grid, and transparencency, since technologies used are open-source and mainly standard.

Table 1. Grids vs. clouds

ISBN: 978-1-61804-362-7

	Grid computing	Cloud computing
Main feature	Computing resources shared with a Virtual organization	Computing resources leased to any type of user
Application field	Mainly research and educational environment	Every user on the web with IT needs
Providers	Research institute and universities	Companies and or organization of different dimension, structure and scope (commercial or open source)
Technology used	Open-source technology implemented with standards (interoperability between different grid middleware)	Mainly proprietary technology, even if there are some examples of open source cloud middleware (only providers know how their cloud manages resources and the other functionalities (e.g., security)
Benefits	Collaboration, ownership, transparency and resilience	Flexibility, reliability and easy of use
Drawbacks	Reliability, complexity, commercial	Generality, security, opacity, rigidity and provider lock-in

Finally, it offers resilience, since grids are distributed on multiple sites and this reduces the risks of working problems (e.g., if a resource fails, other resources are used as replacement). Grids are secure since the organizations work inside a higher security infrastructure and, several mechanism to use the resources. Moreover, each user must know exactly, where the relevant data are or in which machines the relevant application is running.

Cloud computing allows flexibility, since users can quickly request and release resources on demand, reliability, since the provider guarantees high service availability, and ease of use, since cloud services can be easily used also by non-expert users thanks to the wide use of a web interface.

As for drawbacks, grids can manifest problems as regards reliability, since the distributed management may result in inconsistency across individual sites, complexity, since grids are complicated to build and use and users require some level of expertise that can discourage their use. Finally, grids are generally available for non-profit work, and, without commercial revenue, the market is limited.

Clouds, since they do not offer many of the high-level services currently provided by grid technology, are generally not very secure, since users usually do not have information about where the data are or the applications are running; sensitive data could therefore end up being entrusted to external providers. There is therefore a certaing opacity of the technology in terms of its guarantee of reliability and safety. Rigidity is also a potential issue, since the cloud can be located at a single site, which increased risk of failure; there also is a risk of being locked in to services provided by a very small group of suppliers.

From this analysis it seems reasonable to choose grid if a user needs a collaborative environment to execute applications requiring higher load or a lot of data, such as scientific applications. For other goals and other types of context, the user could choose a cloud service distinguishing between SaaS, PaaS or IaaS services according to needs (e.g., software online, platform where develop or infrastructure where to run specific software or applications).

3.1 Choosing among commercial offerings

Cloud is changing the way we use the resources of computation, in storage, network, machines, software and applications, with the idea of being able to use them when needed with easy access over the web. However, each cloud service (IaaS, PaaS or SaaS) is offered by a great number of providers. The choice of one provider could be difficult for a user, whether an IT architect who chooses an IaaS service to manage information systems, a software developer who needs a PaaS solution or a generic user who wants a SaaS solution. The choice is not based on the kind of access as a client, but on the features provided and on the real cost of the solution. It is necessary to understand what the solution provides and at what cost – not easy. For some cloud services, it is necessary to discriminate among software platforms available (e.g., operating systems, programming languages platforms) and the method used to calculate the real use of resources, which, however, could be provided for hourly, monthly or annual subscription.

Offerings usually cover the various cloud services: IaaS, as the infrastructure consisting in customizable VMs with the possibility of renting hardware capacity (CPU, storage, network), and self-management of software (i.e., operating systems, applications), PaaS as underlying platform that includes application servers or database systems to deploy the cloud apps, and SaaS as software on line. The advantages of cloud solution allows to avoid hardware and software management duties (e.g., installation, configuration and maintenance), focusing only on the software and the scalability and flexibility of the solutions. Regarding the distribution patterns, in the public cloud resources are shared for the public use (i.e., Internet accessible); in private clouds, resources are shared but for the exclusive use of organizations (e.g., usually hosted in an intranet); hybrid cloud combines private cloud with public cloud, merging public and private property into a single entity. The public cloud is currently used as a complementary option for most organizations, rather than replacing the IT infrastructure. Private IT infrastructure, which, however could be organized in a private cloud, hosts the core business, while public cloud is

ISBN: 978-1-61804-362-7

used to solve specific issues or to delegate services that could be publicly accessible on the Internet.

This is the case for example in temporary applications when solving the problem of users' workload for a website or web application for a limited period of time (e.g., application like online context, clicking day, online games). However, even if considering the benefits of cloud reduced IT costs, increased flexibility and agility and increased QoS, the problem is to understand whether there is a real cost reduction in respect the other features. The great problem is sizing in clouds.

To evaluate the cost of choosing a cloud provider for the specific application or service, it is necessary to evaluate the cost of the solution. In addition, each cloud service, except perhaps SaaS, is based on a calculation of different parameters that affect the use of the resource (e.g., workload, networking, data transfer or processing). As an example, a tier includes an instance of VM with the service storage (the so-called Elastic Load balancing and a block storage in any combination of disks both solid state or magnetic), a bandwidth and a regional data transfer. However, according to the choice of on-demand or reserved instance, the provision is broken down according to utilization (light, medium and heavy) and the costs are variable. So, it is difficult to size the parameters lacking the possibility to make a direct comparison between CPU and RAM needs in the case of a service installed in a dedicated infrastructure.

This aspect is independent from cloud providers. Focusing on web-server hosting in the cloud, for example with Windows Azure, there are four level that use a different combinations of CPU core, RAM, storage, data transfer, and so on. Sizing for porting an application on cloud is only possible by testing using the free trial version of the different providers. This poses a great issue, especially in terms of time spent in doing such tests. Finally, a lot of providers offer solutions as an ecosystem of hardware and software products, implementing several cloud services in private and public distribution. This aspect, on the one hand allows a uniformity of systems and avoids incompatibility or interaction problems between the different software, but raises a problem related to vendor lock-in. The aspect is strictly related to the interoperability of different cloud solution and the possibility for the user to change cloud provider in a relatively easy way. Because of this consideration, a public organization should carefully analyses these aspects in choosing a cloud service. Moreover, considering the possibility of using the EGI infrastructure for applications, it should be possible to understand how to integrate the two paradigm for computing needs. Inside EGI projects, there is a move towards the EGI federated cloud [28] as a federation of institutional private clouds offering cloud services, especially in the IaaS perspective, to researchers in Europe and worldwide. Taking advantage of the work in grid computing, the EGI cloud aim at interoperability [29], proposing a federated, standards-based IaaS cloud platform. Thus, a comparison between cloud IaaS and EGI IaaS should be made

2.1 Choosing among commercial offerings

3.2 The EDFI federated cloud

EGI has created a task force to identify and test standard-based technologies, deploy and operate services in a federated cloud. The governance model is a tradeoff between individual ownership of the single organizations and the cost benefits of sharing resources as services. The cloud model is an IaaS since it refers to virtual machines that encapsulate the application software, whose image is instantiated on machines provided by the cloud. Every site of the EGI federated cloud exposes the same programming interfaces (API), through the Open Cloud Computing Interface (OCCI) [30]. It is one of the first standards in cloud, which developed both as a protocol and an application programming interface (API) for all kind of cloud management tasks, allowing integration, portability and interoperability. The first cloud infrastructure platform is just going into production (in the middle of 2014), with, as reported by EGI [29], an initial capacity of 2000 cores and 15TB of storage. The main benefit is that this platform is strongly based on open standard, uses some open cloud software (e.g., OpenStack, OpenNebula and Cloud stack [31]) to build the cloud middleware and thus interoperability is guaranteed. As regards security, for example, the OASIS Security Assertion Markup Language, SAML [32] is adopted as a framework for exchanging security information between partners, as well in the virtualization format (the Open Virtualization Format (OVF)) [33] for packaging and distributing virtual machines or software in general. Fig. 9 shows the cloud infrastructure platform architecture and the used standards as proposed by EGI.

ISBN: 978-1-61804-362-7

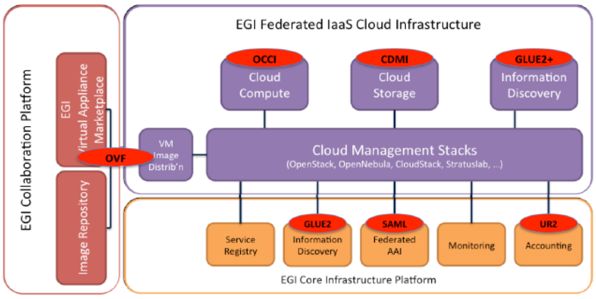


Fig. 9. EGI federated cloud platform architecture and standards (EGI.org)

The aim of the federated cloud is thus to provide cloud services as integrated into the existing production infrastructure. The implementation uses neutral vendor technologies preventing vendor lock-in. Providers organizations acting as providers should only expose the chosen interface and services to the platform in order that cloud services are accessible to users utilizing common standards of profile. In any case, this opportunity should also be evaluated and analyzed considering the actual complexity of use.].

4 Conclusion

Cloud computing seems a winning solution to the growing demand for hardware and software resources (e.g., storage, servers, networks, applications, virtual machines and services) from several organizations and users, also because of the variety of existing commercial and open source solutions. When compare with the other model of distributed computing, grid computing, it seems to prevail. The main drawbacks of grid technology, even if proven mature, include a limited application field (i.e., a research and non-profit environment with the idea of executing resource and storage intensive applications) and complexity in its build and use; these have much slowed its expansion, despite the advantages of using open source technologies to guarantee transparency and interoperability between different solutions. Even in a research environment, for many there is the possibility of access to a production grid such as EGI, with a huge amount of computing resources, grid computing is only used for limited applications, in specific collaborations and contexts. Moreover, since its infrastructure is complex, at this time it is not very clear, despite the efforts of the EGI community, what grid resources will provide, and how, in a cloud service perspective.

On the contrary, cloud computing seems to take advantage of its benefits and users seem to forget the drawbacks, probably because of the ease of use and general services. However, it is still more difficult to understand what a provider means by "work or put all on a cloud", since sometimes it is not immediately clear what type of services he is providing. Even if, as seen, there are different kinds of cloud services offered as utility to the users, other resources use the same terminology (e.g., database as a service or DBaaS). When adopting a cloud model, it is necessary to distinguish the type of service requested, the users' needs and the cost. Then, the problem remains the choice of the provider, given the great availability in the market, and the complexity of understanding the real cost of the solution adopted. In a cloud solution, resource provision is not based on trust agreements between organizations or groups implemented with a Public Key security infrastructure (PKI), but on specific requests to resource providers that require a fee that depends on the type of resource, its size and the time of use. Sizing of cloud solution is actually a great issue.

In research and education domains, grid initiatives are born with the purpose of sharing computing power between research communities (e.g., the high-energy physics community) for the needs of specific projects. The grid approach has not been as successful in other contexts, where improvement of virtualization technologies, service oriented architectures and web services paradigm have encouraged the development of a more flexible and scalable distributed computing model. The problem is related to the integration of cloud services in EGI and the real availability of resources in the service perspective. On the other hand, interoperability and standards are of great importance in grid. In this sense, the choice of standard platforms seems to highlight the choice. There are, however, some issues related to the choice of provider with cloud solutions, both with commercial and public offerings and with the solution provided by the EGI community, because of the lack of an easy way to calculate the benefits and drawbacks of each solution.

References:

[1] I. Foster, C. Kesselman, S. Tuecke, The anatomy of the Grid: enabling scalable virtual organizations, International Journal of High performance computing applications, Vol. 15, Issue 3, August 2001, pp. 200-222.

[2] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future

ISBN: 978-1-61804-362-7

generation computer systems Journal, Vol. 25, Issue 6, , , June 2009, pp. 599-616

[3] A. D. Kshemkalyani, M. Singhal, Distributed computing: principles, Algorithms, and systems, Cambridge University Press, 2011

[4] L. Wang, G.Von Laszewsky, M. Kunze, J. Tao, Cloud computing: a perspective study, Proceedings of the Grid Computing Environments (GCE) workshop, Austin, Texas, 2008

[5] T. Ferrari, European Grid Infrastructure: services of big data processing in a secure federated environment, Big data and open data Workshop, Brussels, May 7-8, 2014.

[6] G. Soskey, Google trends make search data more useful for marketers, HubSpot Inboud Hub, 2013. Online: http://blog.hubspot.com/marketing/google-trends-update-nj

[7] S. Pastore, Distributed and local security policies for grid web-based resources, Proceedings 18th International Workshop on Database and Expert Systems Applications (DEXA 2007), 5th International Workshop on Internet Communications Security (WICS'07), 3-7 September 2007, Regensburg, Germany. Editors A.M. Tjoa and R.R. Wagner, IEEE Computer Society, pp. 703-707, 2007

[8] S. Pastore, Using web services in a grid infrastructure: an early implementation of web services actors programming a grid application to access astronomical databases', LNCS –3270 Springer-Verlag Volumes, on "Grid Services Engineering and Management (GSEM'04)", 2004 pp.1-14

[9] I. Foster, Globus Toolkit version 4: software for service-oriented stystems. IFIP International Conference on Network and parallel computing, Springer-verlag LNCS 3779, pp.2-13, 2006

[10] The globus Alliance, available at url: .. http://toolkit.globus.org/alliance/about.php

[11] D. Britton and S. L. Lloyd, How to deal with petabytes of data: the LHC Grid project, Reports on Progress in Physics, Vol. 77, N. 6, 2014

[12] T. Ferrari, Status and future of the European grid infrastricture after 10 years of production activities, International Symposium on Grids and Clouds (ISCG) 2013, Taipei, 2013.

[13] The Grid: Software, middleware, hardware, CERN Accelerating science, Available: http://home.web.cern.ch/about/computing/grid-software-middleware-e-hardware

[14] P. Asadzadeh, R. Buyya, C. L. Kei, D. Nayar, and S. Venugopal, Global Grids and Software Toolkits: A study of four grid middleware technologies, Available on line: http://arxiv.org/ftp/cs/papers/0407/0407001.pdf

[15] D. Kaleva, Virtual research environments : from portals to science gateways by Robert J. Allan, Library management, Emerald Group Publishing Ltd, Bingley, England, vol. 31, no. 4-5, pp. 361-362, 2010

[16] P. Mell and T. Grance. The NIST definition of cloud computing, NIST special publication 800-145. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2011

[17] A. S. Pillai and L.S. Swasthimathi, A study on open source cloud computing platforms, International Journal of Multidisciplinary Management Studies, Vol.2, Issue 7, July 2012, pp.31-40

[18] I. M. Llorente, Eucalyptus, CloudStack, OpenStack and OpenNebula: a tale of two cloud models, OpenNebula.org, 2013 available at: http://opennebula.org/eucalyptus-cloudstack-openstack-and-opennebula-a-tale-of-two-cloud-models

[19] J. Brockmeier, It's not highlander: there can be more that one open source cloud, readwrite.com, 2012, available at:

[20] The 100 coolest cloud computing vendors of 2014, CRN staff, Jan 31, 2014. http://www.crn.com/news/cloud/240165812/the-100-coolest-cloud-computing-vendors-of-2014.htm

[21] Introducing Microsoft Azure, Available online: azure.microsoft.com/en-us/documentation/articles/fundamentals-introduction-to-azure

[22] J. Euphrosine, Getting started with Google Tasks API con Google App Engine, 2013. Available online: https://cloud.google.com/appengine/articles/python/getting_started_with_tasks_api

[23] Heroku dev center, Getting Started on Heroku, https://devcenter.heroku.com/start

[24] R. Lawler, Dropbox beefs up its pro feature set, now offering 1TB of storage for $10/month, 2014. Available online: http://techcrunch.com/2014/08/27/dropbox-pro-update

[25] D. Walter, 6 powerful Google Drive features you're probably not using, 2014. Available online: http://www.pcworld.com/article/2466790/6-powerful-google-apps-productivity-features.html

[26] W.V. Winkle, Cloud office suites: comparison of online solutions, 2014. Available on line: http://www.tomsitpro.com/articles/cloud-office-suites,2-690-4.html[27] S. Pastore, Distributed and local security policies for grid web-based resources, Proceedings 18th International

Workshop on Database and Expert Systems Applications (DEXA 2007), 5th International Workshop on Internet Communications Security (WICS'07), 3-7 September 2007, Regensburg, Germany. Editors A.M. Tjoa and R.R. Wagner, IEEE Computer Society, USA, 2007, pp. 703-707.

[28] The EGI Federated cloud, a production IaaS infrastructure for the ERA, EGI Community Forum 2014, helsinki, May 2014.

[29] M. Drescher, Interoperability is the key to freedom in the Cloud, EGI Cloud position paper, Cloudscape VI, Brussles, Belgium, Feb. 2014

[30] Open Cloud Computing Interface, OCCI website, available at url: http://occi-wg.org

[31] D. Wallom, Integrating openstack into a federation of institutional private clouds for escience, http://www.openstack.org/vote-paris/Presentation/integrating-openstack-into-a-federation-of-institutional-private-clouds-for-escie

[32] OASIS security services (SAML) TC, OASIS website, available at url: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[33] Open Virtualization Format (OVF), DMTF site, available at url: http://www.dmtf.org/standards/ovf