EGEE

AN EGEE COMPARATIVE STUDY: GRIDS AND CLOUDS- EVOLUTION OR REVOLUTION?

Document identifier:	EGEE-Grid-Cloud-v1.1.doc
Date:	**30/05/2008**
Activity:	**NA1**
Lead Partner:	**CERN**
Document status:	**Final**
Document link:	https://edms.cern.ch/document/925013/

Copyright notice:

Copyright © Members of the EGEE-II Collaboration, 2006.

See www.eu-egee.org for details on the copyright holders.

EGEE-II ("Enabling Grids for E-sciencE-II") is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 6th Framework Programme. EGEE-II began in April 2006 and will run for 2 years.

For more information on EGEE-II, its partners and contributors please see www.eu-egee.org

You are permitted to copy and distribute, for non-profit purposes, verbatim copies of this document containing this copyright notice. This includes the right to copy this document in whole or in part, but without modification, into other documents if you attach the following reference to the copied elements: "Copyright © Members of the EGEE-II Collaboration 2006. See www.eu-egee.org for details".

Using this document in a way and/or for purposes not foreseen in the paragraph above, requires the prior written permission of the copyright holders.

The information contained in this document represents the views of the copyright holders as of the date such views are published.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE MEMBERS OF THE EGEE-II COLLABORATION, INCLUDING THE COPYRIGHT HOLDERS, OR THE EUROPEAN COMMISSION BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks: EGEE and gLite are registered trademarks held by CERN on behalf of the EGEE collaboration. All rights reserved"

Delivery Slip

	Name	Partner/Activity	Date	Signature
From	Marc-Elian Bégin	CERN		
Reviewed by	Bob Jones, James Casey, Erwin Laure, Francois Grey, Cal Loomis, Rolf Kubli	CERN CERN CERN CERN CNRS EDS		
Approved by	Project Management Board			

Table of Contents

1 EXECUTIVE SUMMARY

This report compares grid and cloud computing services, taking a practical look at implementations of both: namely the Enabling Grids for E-sciencE (EGEE) project for grid and the Amazon Web Service (AWS) for cloud. Taking performance, scale, ease of use, costs, functionality and other aspects into consideration, the report looks at the overall opportunity that converging cloud and grid services can bring to users.

Cloud computing is picking up momentum as the next "big thing" in computing. In contrast grid has passed the peak of the new technology hype and is now in production usage. Due to the commonality between the technologies underlying grid and cloud, a question might be "will cloud computing make grid obsolete?" Yet, as this report explains, a better question should be "how can users benefit from the developments around cloud computing to extend and simplify their grid utilisation."

The EGEE grid infrastructure federates existing computing resources spanning many sites, countries and continents. It is an infrastructure of independently administered heterogeneous resources with distributed multi-science user communities, and is mostly publicly funded including an important contribution from the European Commission. EGEE offers high-level services that allow its user communities to collaborate and contribute resources to common scientific challenges.

The Amazon commercial cloud offering includes two flagship services: Elastic Computing Cloud (EC2) and Simple Storage Service (S3). EC2 relies on hardware virtualization. Using a simple interface, the user provides as input a virtual image (stored on S3) as well as input data and the resources are allocated as required. S3 is a simple service for storing and accessing data on the Amazon cloud, using technologies such as REST, HTTP, SOAP and BitTorrent.

Cloud and grid do have a lot in common, but there are differences. One important difference being that grids are typically used for job execution (i.e. limited duration execution of a programme, often as part of a larger set of jobs, consuming or producing all together a significant amount of data). While clouds support a job usage pattern, they are more often used to support long-serving services. Users are gaining confidence in the cloud services and are now outsourcing production services and part of their IT infrastructure to cloud providers such as Amazon. Grids provide higher-level services that are not covered by clouds; services enabling complex distributed scientific collaborations (i.e. virtual organisations) in order to share computing, data and ultimately scientific discoveries. As cloud computing evolves, it is likely that higher-level services will appear.

Technologies such as REST, HTTP, hardware virtualisation and BitTorrent could improve existing accesses to grid resources.

The potential benefits of simplicity that cloud technologies offer grids may help to better serve its current users, attract new user communities, accelerate grid adoption and importantly reduce operations costs.

The integration of grid and cloud would be eased if open source software implementations were made available to encourage standardisation. The interfaces provided by Amazon with S3 and EC2 could be valid starting points for such an open source implementation and standardisation.

While it is expected that the technologies underlying grid and cloud are likely to converge and offer gateways between different implementations, differences will persist between commercial services and publicly managed resources at the usage and access level due to the influence of national policy and legislation.

It is important that new developments are not a distraction from ensuring that current grid users can continue to rely on a production e-Infrastructure for their daily work. Therefore, a roadmap should be defined to include cloud technology in current e-Infrastructures in an incremental and harmonious fashion.

2 INTRODUCTION

2.1 OBJECTIVES AND CHALLENGES

This report compares grid and cloud computing services, taking a practical look at concrete implementations of both. In this report, we go beyond the simplistic, and probably sterile, comparison of computing and storage costs from commercial cloud offerings versus a publicly funded e-infrastructure such as EGEE. Whilst taking cost into consideration, throughout this comparison, we try to identify opportunities for the grid to adapt and take advantages of cloud, both from a resource provisioning and technological point-of-view. Finally, we propose convergence routes and opportunities for new investigation and development work to improve the e-Infrastructure European and international researchers require to produce world-class science.

In this report, we ask more questions than provide answers, as this is one of the goals of this study: to identify those aspects that need to be considered for grids from the latest development in clouds.

In this study, we try to go beyond the cloud hype and compare cloud and grid for what we understand they are today. We briefly remind the reader what the objectives of grid and cloud are, using concrete implementations of both in order for this report to be practical as opposed to a theoretical dissertation.

Once we have better understood the approach and technological choices made by grid and cloud systems in production, we identify convergence paths, as well as concrete steps to take, to realise this convergence between grid and cloud.

During this analysis we take a fresh look at grid and cloud usage patterns, the technologies used and the justification for standards, in order to identify where cross-fertilisation can take place. The cost associated to both approaches is considered, but is not the only theme of this study since issues of ownership and access policy will have a significant impact on how grid and cloud services are used.

As cloud computing picks-up momentum, it is only normal that people ask the obvious question: “is cloud not making grid obsolete”? While there is enough commonality between cloud and grid to invite this question, we need to remember the objectives of grid in a public research context, versus the commercial objectives of cloud computing providers. However, this question is valid and deserves an answer, especially when grids are still being co-funded by taxpayers’ money. In parallel, since the grid has passed the peak of the new technology hype that it created a few years ago, many sectors are looking at cloud computing as the next ‘big thing’ in computing.

In a GridToday editorial [12] in April 2008, Derrick Harris asked the question: “*is Cloud Computing Actually for Real?*” to which he answers: “*The simple answer, according to Gartner [talking about Gartner’s Symposium ITXpo in Las Vegas 2008[1]], is ‘Yes.’ Among the myriad statistics thrown out at the conference was Gartner’s prediction that by 2012, 80 percent of Fortune 1000 companies will pay for some cloud computing service, and 30 percent of them will pay for cloud computing infrastructure. Pretty impressive if it comes true*”. This indicates that although cloud computing is in a hype phase it actually delivers services and generate real business.

An important challenge in this work is to see through the hype that cloud computing has generated. This is one of the reasons we have chosen to focus on a current and publicly accessible implementation of cloud: *Amazon Web Service* (AWS), with a specific focus on the EC2 and S3 services. Where appropriate, we also endeavour to keep the discussion general, so that similar reasoning can still be applied to different commercial offerings of cloud computing, such as similar offers in the works by Google and IBM.

---

[1] http://www.gartner.com/it/sym/2008/spg10/spg10.jsp

2.2 DEFINITIONS

Two definitions are required for this analysis: *grid* and *cloud*. To avoid a debate on these definitions, we will simply refer to well-known and documented implementations: the 'EGEE grid' and the 'Amazon cloud'. These definitions are expanded below to provide an overview of the architecture of these two implementations.

2.3 GRID

Unless explicitly stated in the text, when using the term *grid* in this report, we refer to the EGEE grid infrastructure[2]. In this context, the highlights of such a grid are:

- Federated yet separately administered resources, spanning multiple sites, countries and continents;
- Heterogeneous resources (e.g. hardware architectures, operating systems, storage back-ends, network setups);
- Distributed, multiple research user communities (including users accessing resources from varied administration domains) grouped in Virtual Organisations (VO);
- Mostly publicly funded (both resources and engineering, but not necessarily from the same funding source), at local, national and international levels;
- Range of data models, ranging from massive data sources, hard to replicate (e.g. medical data only accessible at hospital premises), to transient datasets composed of varied file sizes.

For a more abstract definition, the EGEE's implementation of grid is compatible with Ian Foster's original definition of grid, as documented in [1].

The EGEE grid is powered by open source software, under the banner of gLite[3]. gLite is a distribution composed of contributions from various groups and projects in Europe and the USA including the Globus Toolkit[4], Condor[5] via the Virtual Data Toolkit[6] and LCG tools[7]. gLite also includes development funded directly by the EGEE project, often in collaboration with other grid-related projects.

This middleware provides the user with high level services for scheduling and running computational jobs, accessing and moving data, and obtaining information on the grid infrastructure as well as grid applications, all embedded into a consistent security framework. Security services encompass the Authentication, Authorization, and Auditing services which enable the identification of entities (i.e. users, systems and services), allow or deny access to services and resources, and provide information for post-mortem analysis of security related events. It also provides functionality for data confidentiality and a dynamic connectivity service, i.e. a means for a site to control network access patterns of applications and grid services utilising its resources. Information and Monitoring Services provide a mechanism to publish and consume information and use it for monitoring purposes. The information and monitoring system can be used directly to publish, for example, information

---

[2] http://www.eu-egee.org

[3] http://www.glite.org

[4] http://www.globus.org/

[5] http://www.cs.wisc.edu/condor/

[6] http://vdt.cs.wisc.edu/

[7] http://cern.ch/lcg

concerning the resources on the grid. More specialized services, such as the Job Monitoring Service and Network Performance Monitoring service, can be built on top.

The Computing Element (CE) provides the virtualization of a computing resource (typically a batch queue of a cluster but also supercomputers or even single workstations). It provides information about the underlying resource and offers a common interface to submit and manage jobs on the resource.

The Workload Management System (WMS) is a grid level meta-scheduler that schedules jobs on the available CEs according to user preferences and several policies. It also keeps track of the jobs it manages in a consistent way via the logging and bookkeeping service.

The Storage Element (SE) provides the virtualization of a storage resource (which can range from simple disk servers to complex hierarchical tape storage systems) much as the CE does for computational resources. The three main services that relate to data and file access are: Storage Element, File & Replica Catalog Services and Data Management.

The development of these software contributions and the operations of the infrastructure on which they are deployed are primarily funded by national (European member state governments) and international public bodies and organisations, notably the European Commission.

Most resources are provided by research institutes, performing research in one or more disciplines. Currently, over 250 institutions in 50 countries have connected elements of their IT infrastructure (ranging from CERN[8] which contributes approximately 15% of EGEE's total resources to individual university faculties with less than 10 PCs) to the EGEE grid and thus providing computing and storage (including networking).

The EGEE project, now entering its 3rd phase, has been funded in two years cycles under a competitive-call scheme, which makes it difficult to build a permanent or long-term strategy for an infrastructure that has become mission-critical for research communities such as High Energy Physics (HEP) and Life Sciences. To address the sustainability of infrastructures such as EGEE, a more permanent arrangement is been sought via the European Grid Initiative (EGI) Design Study project[9], which would bring public national and international stakeholders together with a longer term commitment of support for a pan-European grid infrastructure.

A more detailed description of EGEE is provided in [18].

2.4 CLOUD

Wikipedia[10] provides the following definition for Cloud computing:

> *"**Cloud computing** gained prominence in 2007 as a term used to describe computing that is made generally available on a publicly available IP basis (i.e. the Internet) -- "in the cloud". The term derives from the fact that most technology architecture diagrams depict the Internet or IP availability by using a drawing of a cloud. The compute resources being accessed are typically owned and operated by a third-party on a consolidated basis in [Data Center] locations. Consumers of the "cloud" are **concerned with services it can perform rather than the underlying technologies used** to achieve the requested function."*

Cloud computing, as many authors have already pointed-out, is not new. As Robertson *et* al stated in [2] "*The* cloud paradigm *is not a new concept; it has been developed and promoted already by*

---

[8] http://www.cern.ch

[9] http://web.eu-egi.org/

[10] http://en.wikipedia.org/wiki/Cloud_computing

companies like HP and IBM with names like utility *or* ubiquitous computing. *Other keywords used were* on-demand-computing, pay-as-you-go IT services *or* grid-computing".

In the context of this report, we refer to Amazon's current commercial cloud offering called the *Amazon Web Services* (AWS), with a particular focus on its flagship services: S3 and EC2.

2.4.1 Computing: EC2

EC2 (Elastic Computing Cloud) is the computing service of Amazon. EC2 allows users to request the instantiation of virtual images. The EC2 service therefore relies entirely on hardware virtualisation, as opposed to a *job* running on worker nodes, as is the most common approach in grid and batch systems. Virtual images are stored in the S3 service. Users can request the instantiation of existing public images, or can craft their own. EC2 requires virtual images to be in a specific proprietary format: Amazon Machine Image (AMI)[11]. While this format is not publicly documented, tools already exist to port and convert existing images to the AMI format[12] [13]. AMI is based on the Xen[14] technology, but also adds a few other items such as file system and other system information. Other software providers, such as rPath, developers of rBuilder (a software product able to build virtual appliances), also supports the AMI format. While recipes exist to build AMI bundles from Xen images, standardising the AMI format would improve portability of applications between different clouds and grids; though this is not a priority for Amazon at the moment.

Since the launch of EC2 and S3, Amazon has released new services, such as SQS (Simple Queue Service) and SimpleDB. Amazon is also testing a new billing service called DevPay, which will allow users to leverage its secure billing infrastructure as a service. Lately, Amazon has also released "Elastic IPs" (Static IPs for Dynamic Cloud Computing)[15] allowing users to assign static IPs to dynamic resources deployed using EC2, as well as "Multiple Locations" which allows users to request EC2 instances to be geographically distributed. These last services address the need for EC2 IP addresses in a static range for applications like email service hosting, as well as providing building blocks for building more resilient services in that case the operations of an AWS data centre are compromised.

The EC2 relies heavily on hardware virtualisation. When a user requests an EC2 resource, the user provides a reference to a virtual image, stored in the S3 service. The image can be custom (public or private), or can simply be one of the standard images provided for free by Amazon. The EC2 service then boots the image and returns a handle to the virtual instance. Amazon proposes different types of hosts with different performance profiles. Once the machine has booted, the user has full access to the machine, including logging into it as root.

Amazon has also introduced its own performance unit for computing called *EC2 Compute Unit*[16]. Each type comes with a specific usage price (for more details, see section 3.2).

---

[11] http://www.amazon.com/gp/browse.html?node=201590011

[12] http://www.enomalism.com/features/amazon-ec2-migration

[13] http://www.rpath.com/corp/amazon.html

[14] http://www.cl.cam.ac.uk/research/srg/netos/xen/

[15] http://www.businesswire.com/portal/site/google/?ndmViewId=news%5fview&newsId=20080327005155&newsLang=en

[16] http://www.amazon.com/b/ref=sc_fe_c_0_201590011_2?ie=UTF8&node=370375011&no=201590011&me=A36L942TSJ2AJA

With an EC2 request, the user can also provide user input data, such that the EC2 instance can be parameterised. This is critical for a job pattern, where a large number of resources will be requested, from an identical virtual image. This can be compared to a small input sandbox on the grid. Another standard pattern on AWS is to use the Amazon Simple Queue Service[17] (SQS) service to load job instructions in the queue and let each job in a set to extract its instruction from the queue.

Amazon provides different ways of requesting EC2 instances. The simplest way is to issue an HTTP GET or POST request. A SOAP interface is also available, as well as Java based command-line tools and Java APIs that can be directly integrated into a Java application. This means that standard operating system installations are able to request EC2 instances, with no or little software to install. A Firefox plugin[18] is also available for managing, launching and monitoring EC2 instances.

At this point in time, EC2 does not support Windows directly. While third party solutions exist[19], it is unclear how exploitable these solutions are for production usage.

2.4.2 Storage: S3

The Simple Storage Service (S3) is a service for storing and accessing data on the Amazon cloud. From a user's point-of-view, S3 is independent from the other Amazon services.

The structure of the S3 storage is composed of *buckets* (a.k.a. 'containers') and *objects* (a.k.a. 'files'). Buckets can contain buckets and objects. Metadata can be associated with objects, in the form or key/value pairs.

Several interfaces including SOAP and REST are available. REST (Representational State Transfer) is a term coined by Roy Fielding in his Ph.D. dissertation [4] to describe an architecture style of networked systems. In the summer of 2007, the book *RESTful Web Services*, by Leonard Richardson and Sam Ruby [5] proposed a functional approach and reference architecture to implement the REST concepts proposed by Fielding. The proposed framework is coined *Resource Oriented Architecture* (ROA). In their book, the Amazon S3 service is used as a case study. The main idea of ROA is that if the architecture of a system can be defined such that entities in the system can be represented as resources addressable using distinct URLs, then the proposed RESTful pattern of ROA applies. This means that the resources can be remotely manipulated using the HTTP(S) protocol actions (e.g. GET, PUT, POST and DELETE). The benefit of this approach is that an arbitrary Remote Procedure Call (RPC) access pattern is not required, which simplifies the definition of the interface to these resources. In the case of Amazon S3, Amazon has defined a ROA access pattern to its data. *Buckets* and *Objects* are addressable via distinct URLs, which means that to retrieve a bucket or object, an HTTP(S) GET is sufficient, where calling DELETE on the same URL will delete the bucket or the object, etc. Options can be provided with the request but the bulk of the access pattern is defined by the simple fact of exposing the resources as addressable URL and using the HTTP(S) protocol to manipulate these resources, which is the promise offered by ROA using REST.

Continuing on the REST access to S3, security is provided by the standard HTTP(S) protocol. HTTP(S) supports encryption, authorisation, both via username/password and digital certificates.

These technological choices significantly simplify access to data resources making it more attractive to application developers and tool providers.

---

[17] http://www.amazon.com/Simple-Queue-Service-home-page/b/ref=sc_fe_l_2?ie=UTF8&node=13584001&no=3440661&me=A36L942TSJ2AJA

[18] http://developer.amazonwebservices.com/connect/entry.jspa?externalID=609

[19] http://fabrice.bellard.free.fr/qemu/

The usage of HTTP as the underlying protocol means that little, if any, tooling is required to access the S3 resources. Since most operating systems now come with HTTP connectivity capability, the entry point to access such resources is very low.

Again this choice probably explains why simple, yet very convenient, tools such as Firefox plugin[20] are now available for managing, launching and monitoring S3 buckets and objects.

Finally, another interesting interface that S3 provides is BitTorrent, a popular file-sharing protocol that enables efficient cooperative data distribution. Appending “?torrent” to a standard S3 URL will return the required information for any BitTorrent client to handle the object as a torrent[21].

Here follows an extract from Amazon’s S3 documentation: “*These charges will appear on your S3 bill and usage reports in the same way. The difference is that if a lot of clients are requesting the same object simultaneously via BitTorrent, then the amount of data S3 must serve to satisfy those clients will be lower than with client/server delivery. This is because the BitTorrent clients are simultaneously uploading and downloading amongst themselves. The data transfer savings achieved from use of BitTorrent can vary widely depending on how popular your object is. Less popular objects require heavier use of the “seeder” to serve clients, and thus the difference between BitTorrent distribution costs and client/server distribution costs may be small for such objects. In particular, if only one client is ever downloading a particular object at a time, the cost of BitTorrent delivery will be the same as direct download.*” For more details on the S3 cost and metering model, refer to section 3.2.

Again, while the usage of BitTorrent in a grid context needs to be better understood, what is interesting here is that in the scenario where more than one client requests the same file, and assuming that the clients are outside the cloud, clients become sources of data outside the cloud. This in turn means that these transfers would not be metered by the cloud. However, the transfers will still take place over the network, with associated costs by network service providers. This scenario also applies if the clients are inside the cloud and the data outside, since data transfers inside the cloud are not metered.

In terms of limits, each file stored under S3 is limited to 5 GB. Further, a single user is limited to 100 top buckets, but there are no known limits to the number of buckets and objects a bucket can contain.

---

[20] http://www.rjonna.com/ext/s3fox.php

[21] http://noisemore.wordpress.com/2006/03/14/amazon-s3-has-bittorrent-support/

3 COMPARISON

3.1 OVERVIEW

In the way it is offered by Amazon, cloud technology is a disruptive technology, which in part explains the attention it has received in the media. However, considering Figure 1 below, we only know the user interface side of the cloud from reviewing the Amazon Web Services (AWS), while we have little information on the resource interface and how Amazon is providing and managing the resources available via the cloud user interfaces. This is important since, if we believe that the interfaces offered by AWS are interesting and worth considering as the basic constructs for grid, then we need to understand better how to expose resources to provide these user-level interfaces.

AWS has two sets of interfaces, as shown in Figure 1, the user interface and the resources interface. This last interface is hidden and Amazon has been careful not to disclose how they operate their data centres and implement the user interfaces, execute the user requests, maintain their accounting, etc. While users have inferred some basic strategies that Amazon seem to have adopted (see [8]), the AWS back-end is still not well known.

The EGEE grid, which federates separately administered resources, must not only expose a user interface but also a resource interface to permit providers to connect their resources.

It is reasonable to assume that on the resource side, exposing grid or cloud resources is performed using similar techniques and strategies. For example, a queuing mechanism is required in both cases whether the data centre is to dispatch a grid job via its batch system, or requested to instantiate a new virtual machine.

On the data side, S3 seem to favour a synchronous access pattern, which would indicate that the data is stored on disk instead of tape. However, the ROA proposed in [5] also proposes an asynchronous access pattern, where the user issues a POST command to trigger asynchronous file retrieval. The newly created file request resource can then be used as an asynchronous file handle. With this, the user can monitor the file request and/or get a torrent descriptor to that file. This is to say that while further investigation is required, the REST access pattern to data could potentially work for asynchronous data access to data stored on slower media, like tape.

Considering these two interface levels, we need to explore what could be the potential impact, opportunities and disruptions (positive and negative) on current grid technology, middleware, operations and cost.

In the medium term, the greatest potential benefit of cloud, as proposed by Amazon, is probably not the service itself, but its interfaces and usage patterns. We should also consider the lessons learned from its very existence, as discussed earlier with its technological choices and focus on simplicity.

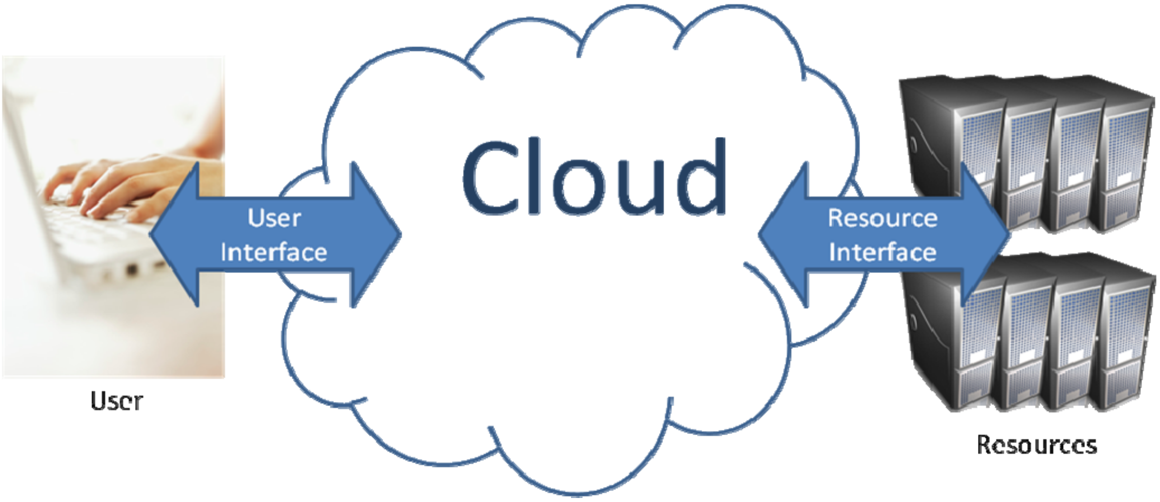

Figure 1: Cloud interfaces high-level diagram

The Amazon S3, like any other computing system, is not fail proof as shown by an important system failure that occurred on 15 February 2008[22]. While e-Infrastructures such as EGEE are not guaranteed to be fail proof either, their distributed nature provides resilience that a centralised cloud might not be able to provide easily. Although it is apparent that EC2 and S3 are not located on a single site, the extent of their distribution is not known. To ensure availability and avoid being locked into a single supplier it would be advisable to use multiple clouds from different suppliers but this requires the existence of standard interfaces or migration tools which do not yet exist.

Clouds and grids do have a lot in common, but also have several differences. Bearing in mind the proposed definitions of cloud and grid provided in this report, one important difference between the two is that grids are typically used for job execution (i.e. limited duration execution of a programme, often part of a larger set of jobs, consuming or producing all together a significant amount of data). Although grid user patterns are not limited to this scenario, it largely dominates the usage of the EGEE grid. Most grid computing resources, provided in the form of a computing element, are organised behind a batch system, which lend itself well to the job pattern.

While clouds support a job usage pattern, they seem more often used to support long-serving services. Users are gaining confidence in the cloud services and are now outsourcing production services and part of their IT infrastructure to cloud providers such as Amazon..

3.2 COST

EGEE's funding model is such that the hardware resources (i.e. computing, storage, networking – and the supporting infrastructure such as space, power and cooling) and operational staff is not covered by the European Commission funding for the project but rather by the participating institutes directly. The European Commission funding covers less than one third of the project's total costs but is a necessary and important incentive to encourage the participants to integrate their resources into the shared infrastructure and adopt agreed operational procedures.

An accounting model exists for tracking the consumption of computing resources and, to a lesser extent, storage occupied by users data in EGEE. This accounting model is used by the major user communities such as HEP to verify that resource contributors are honouring the pledges they made as

---

[22] http://developer.amazonwebservices.com/connect/message.jspa?messageID=79882#79882

part of the planning for the LHC experiments[23]. Given the collaborative nature of the scientific challenges the infrastructure is addressing, billing has not been seen as a priority for most of the user communities. If billing were required, some form of credit scheme would be required to compensate resource providers because many of them are not allowed by their funding agencies to generate income from their IT facilities.

Many IT managers have argued that IT infrastructures were too sensitive to be outsourced. This is perhaps about to change, as Mark Williams puts it, in his review of Nicholas Carr's book *The Big Switch: Rewiring the World, from Edison to Google* [3]: "*IT is a cost center, after all, not so dissimilar from janitorial and cafeteria services, both of which have long been outsourced at most enterprises. Security concerns won't necessarily prevent companies from wholesale outsourcing of data services: businesses have long outsourced payroll and customer data to trusted providers. Much will depend on the specific company, of course, but it's unlikely that smaller enterprises will resist the economic logic of utility computing. Bigger corporations will simply take longer to make the shift.*"

Amazon will bill users for computing resources usage with a minimum of one hour of usage. While this might be reasonable for long running jobs, it stops being cost efficient when dealing with a large set of small jobs, as is the case for many biomedical grid applications.

This situation can be compared with the early days of mobile phones, where service providers would charge users per minute of usage. Recognising that a significant proportion of mobile phone conversations are short (less than a minute), this represented a significant overcharge. This was eventually solved when providers started to charge their customers per second of call time.

In the case of grid applications running on EC2, unless this issue is addressed (although there is no indication of this on the EC2 developers' mailing list), overlaying a 'pilot-job' or 'job agent' pattern might be required to better utilise cloud resources, a pattern[24] well-known to grid users.

---

[23] http://cern.ch/LCG/planning/planning.html#res

[24] http://wiki.egee-see.org/index.php/Pilot_jobs_using_centralized_storage

EC2 Cost Model

The EC2 resources are charged per instance (running virtual images), per hour. Different performance *instance types* are available, each with an associated price[25]. Both computing and data movement are charged, except for data movement between EC2 and S3.

Amazon is using its own EC2 Compute Unit, which means that we need to translate it into a better known performance unit is order to perform meaningful comparisons (see section 3.3).

S3 Cost Model

The S3 cost model[26] includes transient and permanent data. The S3 pricing policy means that not only will Amazon charge for storing data on the cloud, they will also charge for moving the data in and out of the cloud.

An important exception though is that the data movement between S3[27] and EC2 is free. In section 3.3 we report published experimental results on network performance between S3 and EC2.

The costs comparison between the EGEE grid infrastructure and the Amazon cloud is not a simple comparison, since both infrastructures offer significantly different services.

In terms of ball-park figures, the recorded work-load provided by the EGEE grid to its user communities for the whole of 2007[28] would cost in excess of 59 million dollars if performed with EC2 and S3 at current advertised prices.

In the case of EGEE and the LHC experiments, considering that the bulk of the data will come from a few sources, e.g. LHC Experiments, we could consider sending the data via standard mail (i.e. 'Truck FTP') to the nearest Amazon data centre for a direct injection into the cloud. This procedure is recommended by Amazon, but the cost is not clear (i.e. higher or lower than standard network transfer). This could potentially save (part of) the data transfer costs that Amazon would charge grid users and would be subject to negotiation with Amazon.

In a cost study [2] (not yet published) Ian Bird, Tony Cass, Bernd Panzer-Steindel and Les Robertson propose a costs comparison between different solutions for providing the LHC community with supplementary storage and computing resources in view of the increased demand that the LHC Upgrade Programme will bring in the next few years. One of the solutions investigated is the '*Computing Cloud*' model. Their study shows that the cost of providing 40 MSI2000[29] of computing would be 92 MCHF if EC2 were to be used, compared with 4.4 MCHF if a new computing centre was to be commissioned at CERN.

---

[25] http://www.amazon.com/b/ref=sc_fe_l_2?ie=UTF8&node=201590011&no=3440661&me=A36L942TSJ2AJA. Note: the figures reported are as of March 2008

[26] http://www.amazon.com/S3-AWS-home-page-Money/b/ref=sc_fe_l_2?ie=UTF8&node=16427261&no=3440661&me=A36L942TSJ2AJA. Note: the figures reported are as of March 2008.

[27] Note that data used from EC2 from the new European S3 service will be charged as if the data was outside the cloud.

[28] Slide 10: http://indico.cern.ch/contributionDisplay.py?contribId=172&sessionId=0&confId=22351

[29] Million of SPECINT 2000.

The calculation proposed by Robertson *et al* strongly relies on the translation of EC2 Compute Unit to SPECINT2000. Amazon states that "*One EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. This is also the equivalent to an early-2006 1.7 GHz Xeon*"[30]. The problem here is that Intel and AMD are now delivering chips clocked at a much higher rate, and doing simple linear scaling of the chip clock is not accurate. Using a SPECINT2000 value[31] for the Dell Precision WorkStation 530 (1.7 GHz Xeon, with 1 core, 1 chip, 1 core/chip) of ~600 (not taking into account that the tested Dell machine had only 256 MB of RAM, instead of over 1 GB for EC2), we find that the cost of EC2 using the *large instance* (which correspond to 4 EC2 Compute Unit) would be around 57 MCHF, taking an hourly rate for a large instance of 0.40$.

On the other hand, the value calculated in [2] for the custom computing centre does not include manpower, both for commissioning and maintaining the system. Considering that CERN employs several dozens of system administrators and engineers to operate the current computing centre, the real cost of a custom centre would be more than the proposed 4.4 MCHF. We should also add licensing costs for large scale commercial software deployment from Oracle[32] and Platform[33]. While this extra cost would arguably not be significant for CERN since it already has a large computing centre and this extension would largely be operated by the same teams, it should be taken into account for other institutes contemplating building a custom computing centre from scratch. Adding to this the significant risk of commissioning a complex system for the first time, the price difference between outsourcing to a cloud service such as AWS compared to building and maintaining its own data centre is probably less obvious.

Finally, to have a more complete cost estimate in the case of AWS, we also need to consider the cost related to data (see costs listed above).

An important advantage of commercial cloud services however, and this is also acknowledged in [2] is the ability for resource managers, with or without access to a local data centre, to outsource peaks of activity to cloud services, using a pay-as-you-go policy without long-term commitment.

3.3 PERFORMANCE

A common question from people deploying and using distributed and data intensive systems looking at cloud computing is *performance*, including bandwidth or I/O. In this last case, few empirical measurements are available, with the exception of [6]. The results of the simple series of tests reported in this web posting are summarised below:

Table 1: EC2, S3 bandwidth performance summary

Test type	Transfer (MB/sec)	Remarks
EC2 -> EC2	75.0	Using curl on 1-2 GB files, without SSL
S3 -> EC2	49.8	Using 8 x curl on 1 GB files, with SSL
	51.5	Using 8 x curl on 1 GB files, without SSL
EC2 -> S3	53.8	Using 12 x curl on 1 GB files, with SSL

---

[30] http://www.amazon.com/Instances-EC2-AWS/b/ref=sc_fe_c_0_201590011_2?ie=UTF8&node=370375011&no=201590011&me=A36L942TSJ2AJA

[31] http://www.spec.org/cpu/results/cint2000.html

[32] http://www.oracle.com

[33] http://www.platform.com/

Notes:

- All EC2 instances were 'large instances'. Incidentally, they are the same instances used in the cost comparison reported in section 3.2.
- S3 was the USA-based service (an S3 European service is now available).

The conclusions from [6] regarding the EC2 -> EC2 transfers are that "*basically we're getting a full gigabit between the instances*". And in summary: "*The bottom line from these experiments is that Amazon is providing very high throughput around EC2 and S3. Results were readily reproducible (except for the problem described with the non-SSL uploads) and definitely support high bandwidth high volume operation. Clearly if you put together a custom cluster in your own datacenter you can wire things up with more bandwidth, but for a general purpose system, this is a ton of bandwidth all around.*"

These tests would have to be repeated at production scales before these conclusions can be confirmed. These results are measurements and not part of the Amazon Service Level Agreements, therefore not guaranteed over time. Lastly, since there is little public information on the way Amazon deploys the requested resources, it is not clear if the performance of EC2 resources can be influenced by other data transfer intensive instances (e.g. another high-throughput application happening to be hosted on the same rack, sharing the same network connection).

3.4 SCALE

The Elastic Computing Cloud (EC2) is an interesting choice of words. *Elastic* would indicate, metaphorically, that the cloud can grow, without having to worry about quota or size limitation. Amazon monitors the resources such that the cloud remains elastic by injecting new resources as the demand grows. As long as the time delay between the request of a resource and the delivery of that resource (computing or data) is reasonably small (something difficult to quantify), the cloud remains elastic and users are not forced to build special functionality to take delays and queues into account. While this might be possible for a commercial service, with direct costs being the modulating factor, we need to see if this concept can be exported to other implementations of cloud.

In the grid world, the *elasticity* of the service is provided on two levels: by increasing the number worker nodes, either physically or via configuration of the batch system, at a site or adding new sites. The EGEE grid is continuously monitored but adding new resources requires human intervention (either at the level of the batch system of an existing site or by enabling the VO at more sites).

As documented in [6] and [8], the network performance of EC2 and S3 is reasonable and compares well to the performance that can be expected from large data centres like CERN's. However, while CERN has designed its computing and storage infrastructure to scale such that it can deliver and sustain 70 GByte/sec of data between the disk farm and the computing farm, it is unclear if AWS could offer the same level of service. Interestingly, both CERN and AWS have opted for splitting their storage and compute farms, using a fast interconnect between them. However, Amazon have not disclosed large scale usage measurements which would allow us to conclude that the AWS can scale to the level of what large publicly funded centres can offer.

Performing a test on AWS to verify the data rates performance on a large scale would be an expensive exercise since it would require to instantiate thousands of machines, push into S3 a significant amount of data, and then transfer the order of petabytes of data between S3 and EC2.

For a while, Amazon did not offer a Service Level Agreement (SLA). Here is what Werner Vogels, Amazon's CTO, had to say on the issue in April 2007, as extracted from an interview at the London

QCon[34] (see [10] for details): "*A barrier to adoption of Amazon's web services is the absence of any SLA (Service Level Agreement), making some businesses reluctant to entrust data or critical services to Amazon. "They are absolutely correct," says Vogels, with disarming frankness. "You have to understand that this is a nascent business. So we have to figure out on our side how to give these guarantees. It doesn't make sense to guarantee things, and then not be able to meet those guarantees. It is better to explain to people that there are no guarantees at the moment, except high level statements that it is fast and reliable, instead of lying to them.*""

Amazon subsequently released an SLA for S3[35] in October 2007. This SLA has several small print statements but shows the commitment of Amazon to provide 99.9% of service availability to S3 (i.e. a few hours of down time per year).

In parallel with this careful SLA policy, Amazon has recently released two services to help developers build more reliable applications deployed on AWS: *Multiple Locations* and *Elastic IP Addresses*. Both services are associated to EC2. The first allows users to specify different geographical regions in which the EC2 instances will be deployed. If used to build primary and redundant services in different regions, it would in principal avoid a local EC2 service stoppage to bring down the deployed service. The second one associates static IP addresses to user accounts, such that IP addresses can be assigned to specific EC2 instances. Possibly combined with the Multiple Location feature, this service would allow developers to maintain the same IP address while the real EC2 instances are being rerouted between geographically distant instances for better overall service resilience.

These mechanisms are interesting since they provide building blocks for building more reliable applications 'by design' instead of relying on SLA. While one should not completely replace the other, the availability of both maximises the chances of building reliable and resilient services on the cloud.

3.5 EASE OF USE

A key to the success of cloud is its simplicity as confirmed by blog and mailing list postings on EC2 and S3. This was made obvious to the author of this report with his first attempt to instantiate an EC2 resource. From starting to read the online 'Getting Started'[36] documentation to logging to his private instance using PUTTY on his laptop, he only needed 45 minutes. This included creating an account at Amazon and requesting access keys for S3 and EC2. For simple data browsing, the plugins mentioned in section 2.4.1 and 2.4.2 also provide access to storage and computing within minutes. While these tools are very simple and limited in functionality, they do have the merit of showing that is actually easy to start using the system.

The (re)emergence of HTTP(S), REST, ROA, BitTorrent and hardware virtualisation mix together form a new cocktail that define the AWS cloud. The simplicity comes from the choice of standard technology already available in most operating systems. This means that the complexity is kept server-side, and makes the entry point into the cloud very low which has been a long-standing yet unachieved goal for grids. If these interfaces were to confirm themselves as interesting for a very large user base, they may change the standardisation landscape, in terms of requirements and reference implementations.

3.6 SERVICE MAPPING

---

[34] http://qcon.infoq.com/

[35] http://www.amazon.com/gp/browse.html?node=379654011

[36] http://developer.amazonwebservices.com/connect/entry.jspa?externalID=992&categoryID=87

Ease of use comes at a cost: 'The cost of simplicity'. We mean by this that the basic constructs that EC2 and S3 services offer do not currently meet all the requirements of grid users and do not replace high-level services provided by gLite, such as the File Transfer Service[37] (FTS), the Workload Management System[38] (WMS) or grid catalogues such as ARDA Metadata Catalogue[39] (AMGA), LCG File Catalog[40] (LFC) or GANGA[41].

The EGEE grid is massively distributed, with over 250 sites inter-connected around the world. However, is it clear that a single submit endpoint is required by all? As the grid evolves, will we see the trend of even more sites being added to the infrastructure, or will we see a trend towards a consolidation of sites, with the end results being fewer but much larger data centres providing multi-disciplinary resources to grid users?

If this last scenario turns out to dominate the grid landscape, then would a simpler meta-scheduler such as GridWay[42] or GANGA not provide the right submission abstraction, instead of solutions more complex and expensive to operate?

A similar question could be raised regarding data and file catalogues. Would HTTP(S) and BitTorrent not be able to address a significant part of grid user needs for data? With the ability to decorate 'objects' in S3 with metadata, would this not provide the right hooks to build fast indexing metadata catalogues?

Looking at existing grid services, and considering the new, simpler interfaces proposed by low level services like EC2 and S3, we can ask ourselves what the impact will be on the existing grid services? Will we see the simplification trend continue higher up the stack? Will current grid services grab the opportunity and evolve or will a new generation of high-level services appear and replace the current ones?

This section suggests several areas of investigation. The point is that the latest advancements in cloud computing should be used as an opportunity to improve the grid.

A roadmap should be defined such that the path to convergence between grid and cloud is agreed and managed, in order to ensure that current grid users can continue to rely on the grid for their daily work during the transition, while benefiting from cloud breakthroughs in the medium term.

3.7 COLLABORATION AND VIRTUAL ORGANISATION (VO)

An important aspect of grid is the platform it offers for collaborative and distributed work. While one might argue that, with the lobbying required for getting access to resources, it is still too heavy to set up a Virtual Organisation (VO) in EGEE, it is a fundamental requirement for most of the work that takes place on the grid, since it forms the trust foundation between the stakeholders of scientific grid collaborations.

If a site was to either expose its resources using cloud interfaces, or a research centre was to outsource its data and computing needs to a cloud provider, how would the access be controlled and the VO use-case implemented?

---

[37] http://www.gridpp.ac.uk/wiki/EGEE_File_Transfer_Service

[38] http://glite.web.cern.ch/glite/wms/

[39] http://amga.web.cern.ch/amga/

[40] http://www.gridpp.ac.uk/wiki/LCG_File_Catalog

[41] http://ganga.web.cern.ch/ganga/

[42] http://www.gridway.org

Would it be possible or desirable to provide a VOMS[43]-like service able to issue signed certificates such that a user or a service could present the right credentials to cloud resources? Or should a totally different mapping mechanism be defined? Would a token-based security model proposed by grid middleware such as Unicore[44] be more appropriate?

The S3 Access Control List (ACL) feature contains hooks that could probably be exploited to implement a VO access policy. For example, the owner of an S3 bucket or object can issue time limited tokens that can provide specific users with access to buckets or objects. Whether this feature would provide adequate primitives to implement VO policy is outside the scope of this study, and further work would be needed to answer this question.

Another fundamental aspect to large grids such as EGEE is that they provide a mechanism where individual data centres (small or large) can contribute their resources, without losing control of these resources. For example, resources ‘connected’ to the grid at a given time can be ‘disconnected’ at a later date for local work.

Funding agencies are still funding local and national resources and people associated with them, while the outsource model to cloud providers removes this aspect of being able to ‘touch the resources’. Another example is that countries already funding local data centres might be reluctant to outsource computing and storage to a commercial provider that does not operate in their country.

3.8 APPLICATION SCENARIOS

A large and rich set of grid applications exists as demonstrated at this year’s EGEE User Forum [19][45]. While a thorough review of all applications is outside the scope of this report, we briefly discuss two typical application scenarios that are commonly deployed on the EGEE grid: High Energy Physics (HEP) and Life Sciences applications. We conclude this sub-section with a few remarks on new opportunities given by hardware virtualisation to reduce operations costs regarding application software deployment.

3.8.1 High Energy Physics Application

The highlights in several grid HEP applications, as in the four LHC experiments at CERN, are the following:

- Large amount of data to be processed (~ 15 PB/year);
- Large files (>1 GB each);
- Defined topology for processing (organised in tiers, with CERN being tier-0);
- Single source of original raw data (the four LHC experiments at CERN).

---

[43] http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html

[44] http://www.unicore.eu/

[45] http://egee-uf3.healthgrid.org/

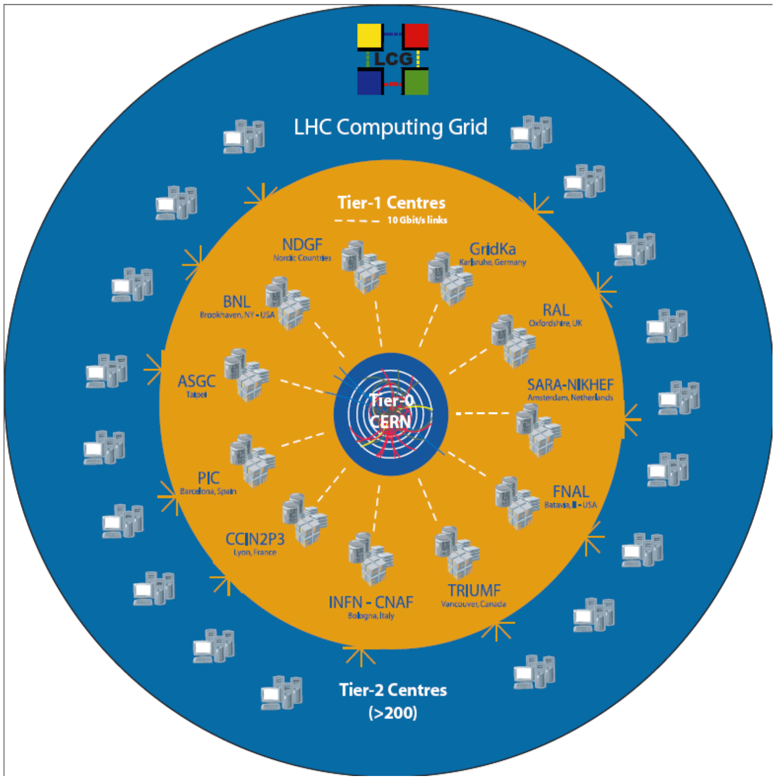

Figure 2: High-level deployment of LCG grid resources[46]

Large grid users such as the High Energy Physics community have very data intensive requirements. CERN will transfer a large amount of data to each of the eleven tier-1 centres (see Figure 2), where it might reach up to 15 PB of data per year for some tier-1. If a tier-1 were to outsource its activity to S3 for storage, the following would be the approximate costs (using the costs reported in section 3.2):

16,200,000 $ per year (1,350,000 $ per month) for permanent storage (at 0.18$ GB/month assuming a constant linear data injection over the year);

1,500,000 $ for uploading the data (at 0.10$ per GB), however it is unclear what would be the cost if the data was sent by mail or dedicated links

If we were to assume that all the data processing was performed within the cloud, since S3 to EC2 data transfers are free of charge, the only cost left from S3 would be the data that would have to be extracted out of S3 in order for other data centres to perform further processing on the data produced and stored by the tier-1 on the cloud. However, if other tiers were to also use S3, then the costs of storage could potentially be shared, reducing substantially the costs reported in the above calculation.

We do not have at hand the amount of computing that a typical tier-1 would have to perform on the data. This cost would have to be added according to the EC2 price model reported in section 3.2.

If the follow-on processing were to also be performed within the cloud, by for example a tier-2 also using AWS, then the access to S3 data by their EC2 machines would be free.

---

[46] Figure extracted with permission from http://it-comm-team.web.cern.ch/IT-Comm-Team/repository/ITposters/LCG-tiers.pdf

As briefly discussed in section 2.4.2, if BitTorrent were to be used, where the BitTorrent clients and the 'seeder' (i.e. original data source) were on different sides of the cloud border and accessing the same files, then S3 would not charge the data transfer since it would not cross the cloud border. This benefit would apply in both setups where the seeder is inside the cloud and the clients requesting the data, or vice versa.

An investigation of the viability of S3 for DZero, one of two physics experiments at Fermilab's Tevatron accelerator, has been performed [20] which concluded that S3 is a performant and reliable services but currently lacks support for fine-grained delegation, essential for large scale collaborative work.

3.8.2 Life Science Application

Another interesting scenario that takes place on the EGEE grid comes from the medical field, where source data is provided by a medical institution (e.g. hospital). By the very nature of the medical raw data, privacy issues often make it impossible to move the data outside the restricted private network of the institution. To access the data, jobs have to be sent to the institution, extracting a slice of the data, perhaps performing local data processing. If the data slice is authorised to leave the restricted network into the grid, then further processing can take place. This is a scenario where it is unlikely that these types of institutions will, at least for a significant period until strong guarantees are available by resource providers, outsource their computing and data needs to commercial cloud providers. In this case, all the data slices that are authorised to leave the medical institutions will have to go through the cloud border, with an associated cost. Further, it is likely that the data will not be authorised to stay on the cloud, which means that the slices will have to be extracted from the medical network every time it is required.

These are only two typical scenarios currently used on the grid. While theoretically it seems possible to use a cloud to perform the processing, beyond the costs, other considerations have to be taken into account, including the scaling issues already mentioned in section 3.4, policy issues (both political and legal) as well as the long term persistence requirements for some of the data.

How are costs likely to evolve in the future? One could imagine negotiating discounts with cloud providers for such large-scale usage scenarios. Similarly, the costs for operating EGEE will remain stable for the next two years during which it is expected that the workload it supports will more than double hence significantly reducing the its estimated cost per CPU/hour and GB/month for the user.

3.8.3 Application Software Deployment

Finally, since cloud relies, for computing, on virtual machines as the scheduling unit, it could be used to simplify the way grid user applications are deployed to the grid. For example, grid user applications could simply 'bake' their application on a virtual image and send this image with their job or instruct the computing service to instantiate this image for each job they require to run. There is already work starting in this direction from the Physics Department at CERN[47] and at Fermilab. For users, having full control of the computing environment they require is very powerful, flexible and convenient. The virtual machines run under strict isolation from the fabric thanks to the hypervisor[48] (i.e. virtual machine execution layer completely separating the physical host from the virtual machines), which gives full control to the resource providers on resource utilisation. This means that users can safely be given privileged (super-user or root) access to the computing resources.

---

[47] http://indico.cern.ch/conferenceDisplay.py?confId=28823&view=standard&showDate=15-April-2008&showSession=2&detailLevel=contribution

[48] http://linuxvirtualization.com/pages/glossary/

Using virtual images to specify the application environment for grid users could have a significant impact on simplifying the relationship between users and resource providers. For example it could remove the need for grid resource providers to give privileged shared file system access on which to install application software, from which jobs can access and setup the required computing environment. Another benefit would be to reduce, if not remove completely, any pre-configuration required for any Virtual Organisation (VO) in order for a given resource provider to execute jobs from that VO.

Lastly, using hardware virtualisation, resource providers do not have to worry about installing user specified operating systems and system level software on their computing farm, since these are fully defined in the target virtual image, as long as the underlying architecture is compatible (e.g. x86, x86-64, Sparc, Power-PC), for the hypervisor to run. This would increase the number of platforms grids such as EGEE can support and potentially reduce the porting effort required for its middleware.

3.9 MONITORING

As with all large scale distributed systems such as grid and cloud, keeping track of system activities, usage record and performance monitoring are critical for smooth operations.

Whether one uses grid or cloud, the same challenge exists and must be solved. To date, Amazon has provided little visibility to their monitoring system. While efforts to provide measurements of the AWS performance are taking place (see [8]), they are done at the user level and not using Amazon monitoring tools.

Perhaps this is an area of work where projects like EGEE with its extensive experience in operations of grid systems could contribute. But here as well, as discussed in section 3.11 on standards, a standard on distributed resource monitoring would help in reaching the goal of a uniform monitoring approach to grid and cloud resources.

If we move towards a grid that includes *cloud sites* the issue of uniform monitoring then extends to cloud. This means that the mechanism used to monitor grid resources must scale but also work in a cloud context where perhaps less control even is provided by cloud providers. On the other hand, since cloud computing resources such as EC2 uses hardware virtualisation, perhaps this is an opportunity to better instrument the resource at the hypervisor level.

3.10 INTEROPERABILITY

As illustrated in Figure 1, clouds offer two main set of interfaces: the user interface and the resource interface. If grid projects such as EGEE, expose cloud user interfaces to grid users, as discussed in section 3.11 below, there will be the need to standardise these interfaces.

Further, in order for grid projects to have freedom of choice between cloud implementations and providers, and/or been able to use and mix different clouds in their grid, interoperability between clouds and services built on top of them is essential.

The same interoperability issue exists for the cloud resource interface illustrated in Figure 1. In order for grid projects to simplify their operations, enable simple swap or addition of resources and allow seamless migration or utilisation of different resources (e.g. data centres), interoperability at the resource interface level will also be required.

While this latter requirement is probably not a real issue for cloud providers, since they fully control their fabric and administration procedures, it is not the case for grid projects such as EGEE, which aggregates existing resources. In EGEE, resources are provided by independent data centres contributing resources using a variety of different funding sources, working on varied scientific domains and from different cultures. For EGEE, operations is the most expensive activity, and if clouds are to play a role, it is therefore important to bear in mind the necessity for providing interoperable cloud resources, in order to allow for streamlining operation, and associated costs, without jeopardising the independence of the resource providers.

As mentioned throughout this report, simplicity is an important tenet responsible for the interest of cloud computing. This also applies to interoperability. It is generally quicker to standardise and ensure interoperability using simple interfaces based on existing implementations. The REST interface to storage proposed by S3 provides an interesting choice for interoperability, since it relies on already largely standardised technology (e.g. HTTP(S), URL) and is widely used, therefore well understood. BitTorrent is also not new and well used, but perhaps less so in the grid world.

Having more independent services, as is the case for example with EC2 and S3, also simplifies the interoperability and standardisation work, since they can be performed in parallel. It is more likely that contributions will be made and new high-level services developed if the underlying interfaces are clear and simple. This also contributes to interoperability since the more interfaces and implementations are used, the more we understand these interfaces and implementations, with corresponding documentation and ability to improve and build confidence in them.

Performance also has to remain on the interoperability radar since we need to be able to easily compare and characterise the performance of different resource providers in order to offer harmonious and consistent services to grid users. This effort also points back to the ability for grid operators to efficiently and reliably monitor the activities and behaviour, of the different resource providers, including cloud providers, as briefly mentioned in section 3.8.

3.11 THE ROLE OF STANDARDS

The lack of standards has been reported on many occasions as a key limiting factor in wider adoption of grid technologies. One contributing element to the slow emergency of standards is the very definition of *grid* in communities such as the Open Grid Forum (OGF - responsible for leading standardisation in the grid world). This was highlighted when the Global Grid Forum (GGF)[49] and the Enterprise Grid Alliance (EGA), in 2006, merged into the Open Grid Forum (OGF)[50]. During GGF17 and GGF18, where it was clear that GGF and EGA were going to come together, the types of grid that the enterprise world (members of EGA) and of a more academic focus (GGF) represented were different. EGA people tended to refer to the high-level goals of grid in terms of 'dynamic commissioning of services and data centres', as opposed to a more 'job based' approach.

We can see this difference when comparing grid (à la EGEE) and cloud (à la Amazon). While the batch system and large data store are apparent aspects of grids, in accordance with the GGF vision, clouds better fulfil the EGA vision of providing dynamic access to computing at the level of a server and a basic access to storage. Lately, Amazon has also released a limited beta service for providing simple data base functionality called SimpleDB[51].

This difference might have a knock-on effect on the need for standards. During OGF22, a BoF was organised to look into the impact of cloud on OGF and explore the opportunity and need for OGF to consider it in its standardisation efforts. The proceedings of the BoF are summarised in [7]. Reading these proceedings, it is clear that OGF recognises the importance for cloud development in general, including AWS in particular.

---

[49] http://en.wikipedia.org/wiki/Global_Grid_Forum

[50] http://www.gridforum.org/

[51] http://www.amazon.com/SimpleDB-AWS-Service-Pricing/b/ref=sc_fe_l_2?ie=UTF8&node=342335011&no=3440661&me=A36L942TSJ2AJA

Such convergence could be simplified if grid providers were to start considering cloud technologies, including the simple cloud user interfaces. Meanwhile, these interfaces will need to be standardised, in order to ensure interoperability for both users and grid providers, as different cloud providers are used and migrations take place. Taking the AWS interfaces as an example, the type of interfaces that might become candidates for standardisation could be:

- REST access to storage;
- Virtual Image formats;
- Instantiation API (perhaps based on REST);
- Metering interfaces (including monitoring).

On the other hand, existing technology that might become widely used for grids such as BitTorrent will not require the same attention from communities like OGF, since already standardised. Virtual machine format might go down the same route, where Xen[52] might provide the foundation for the standardised format, as is already the case for EC2. Further, since Microsoft has contributed[53] to Xen, it might be a topic where de-facto standards take place and most providers rally behind them.

A grid world where cloud plays a larger role may require re-visiting the role of existing standards such as SRM, JSDL and OGSA-BES.

3.12 COMPARATIVE TABLE

The comparison of grid and cloud computing covered in this section is summarised in the table below.

Features	*Grid (EGEE/gLite)*	*Cloud (AWS: EC2 + S3)*
Licensing and Policy		
Client-side APIs and libraries	Open source (Apache 2.0)	Open source
Resource-side middleware	Open source (Apache 2.0)	Proprietary/Closed
Commercial offering	Available via EGEE Business Associate companies	Yes
SLA	Local (between the EGEE project and the resource providers)	Global (between Amazon and users)
Usage and users		
Ease of use	Heavy	Light
Ease of deployment	Heavy	Unknown
Architecture	Well documented	Unknown
Shared access to data and code for an agreed set of users (Virtual Organisation)	Yes	No (only ACL primitives, further work required to assess if VO can be implemented on top of S3)
Number of users	~10000	>>1000 (real number not known)
Number of sites	>250	Several but not clear how many
Virtual image support for	Local policy, if provided	Yes

---

[52] http://www.cl.cam.ac.uk/research/srg/netos/xen/

[53] http://www.informationweek.com/news/software/operatingsystems/showArticle.jhtml;jsessionid=2NCKQ2233KNJMQSNDLRSKH0CJUNN2JVN?articleID=190500358&_requestid=568868

Features	*Grid (EGEE/gLite)*	*Cloud (AWS: EC2 + S3)*
user applications		
Data sources	Federated data sources (e.g. files or external databases/repositories)	Hosted by S3 service
Support for 'Pilot job' usage pattern	Yes	Yes
High-level services (Grid services)		
Data File Transfer Service	Yes	No
Data metadata	Yes (rich schema via metadata catalogue)	Yes (simple key/value pairs directly on S3 objects)
Workload management system	Yes	No
Metering	Yes (via batch system)	Yes
Protocols, standards and technologies		
REST/HTTP(S) interface to storage	No	Yes
REST/HTTP(S) interface to computing	No	Yes
BitTorrent interface to storage	No	Yes
SOAP interface to storage	No	Yes
SOAP interface to computing	Yes (CREAM CE)	Yes
SRM storage interface	Yes	No
Native support for virtualisation	No	Yes
Based on standards	Partially	No (only underlying technologies)
Resource policy		
Resource pool	Federated institutional resources	Access to commercial resources
Queue prioritisation	Yes (via negotiation with resource owner)	No
Local system administration policy	Yes	No
Ability to add custom resources	Yes	No
User platform support	RedHat compatible distributions (32/64 bit)	All (only requires standard libraries, such as HTTP(S) client)
Resource platform support	RedHat compatible distributions (32/64 bit)[54]	Linux, Windows (via QEMU), OpenSolaris (limited beta) –(32/64) bit
Geographic distribution	International (50 countries)	Mainly USA based, with an S3 site in Europe
Support for hardware virtualisation	Local policy	Yes
Performance		

---

[54] http://www.grid.ie/porting/

Features	*Grid (EGEE/gLite)*	*Cloud (AWS: EC2 + S3)*
Scalable	Yes (>20 PB of storage and 55 000 CPUs)	Yes (as far as we know)
Storage to compute bandwidth	~70 GB/sec (measured at CERN between storage and compute farms)	~10 Gbit/sec (measured between S3 and single EC2 instance)
File size limit	?	5 GB
Compute scale	55 000 CPUs	Unknown (but probably large)
Storage capacity	>20 PB	Unknown (but probably large)
Cost of data transfer between resources	Paid by data centres from national/international sources (e.g. GEANT network and extensions)	Pay-as-you-go for data in/out to S3. Free between S3 and EC2
Funding		
Funding model	Publicly funded (negotiation with resource owner for resource access)	Pay-as-you-go (with credit card)
Funding cycle	Project based with national funding agencies and European Commission (confirmed until 2010). Plans for long term funding via EGI.	Commercial operation
Features		
Secured service	Yes	Yes
Support for job execution	Yes	Yes
Support for service virtualisation (long lasting virtual image execution)	No	Yes
SSH access to resources	Local policy	Yes
Root access to resources	No (possible if local policy supports Virtual Machine)	Yes (via Virtual Machine)

4 FUTURE EXPLORATION PATHS

Opportunities for technical convergence and service interoperation of commercial cloud computing services with grid infrastructures should be considered by grid projects such as EGEE. Areas of potential benefit could be in learning from simpler user interfaces to better accelerate grid adoption by existing and new scientific communities, as well as potentially lighten the operational costs of providing resources to the grid via standard resource interfaces. In order for these potential benefits to materialise in the grid world, current grid projects such as EGEE might want to explore new paths that cloud now makes possible, in order to pave the way to a new generation of grids and a convergence between grid and cloud.

In this section, we propose a few exploration paths to stimulate further discussion within grid projects, such as EGEE, for the convergence of grid and cloud.

For example, if data centres, large or small, were to expose their resources following an S3 and/or EC2 like interface, it might make sense to recommend that an open source implementation be developed. While this implementation might fit well as an extension to the existing gLite middleware, the opportunity to start from simpler design and technological choices should not be ignored. This open source implementation could also be used to support standardisation effort by bodies such as OGF.

A problem for both cloud and grid resource providers is the need to guarantee to their users that persistent data is keep persistent. As reported in [9] "*Large government and academic institutions began grappling with the problem of data loss years ago, with little substantive progress to date. Experts in the field agree that if a solution isn't worked out soon, we could end up leaving behind a blank spot in history. "Quite a bit of this period could conceivably be lost," says Jeff Rothenberg, a computer scientist with the Rand Corp. who has studied digital preservation*".

While data persistence is not within the scope of this study the freedom of choice and availability of alternatives from publicly funded and commercial offerings might decrease the probability for a single point service failure to bring down unique copies of important data. As companies, as well as projects, come and go, active management of data is required to ensure that data is protected against digital erosion. We therefore cannot only reason in terms of economical arguments but must also take into account strategic motives to grid and cloud, such as data preservation.

5 RECOMMENTATIONS

Based on our current understanding of cloud computing and the state of the EGEE infrastructure, we believe that a convergence path must be defined and followed to enable grids, such as EGEE, to converge with cloud services.

In this section we propose a set of recommendations that define concrete steps in establishing this path.

1. Promote and/or support the development of an open source cloud middleware distribution, based on interfaces similar to those promoted by current commercial offerings;
2. Promote the standardisation of the cloud, pushing the above mentioned implementation as a reference;
3. Identify a convergence path between cloud services such as EC2 and S3 and the current EGEE security model based on VOMS;
4. Virtualise all key grid services (e.g. information system, metadata catalogues, security service) with the goal of being able to deploy these on EC2-like resources;
5. Promote/lobby the need for experiments (i.e. LHC/HEP, Life science) and other grid users to virtualise their application, with the goal of being able to deploy them on EC2-like resources;
6. As a follow-on to point 5, promote/lobby the need for all service dependencies that grid user applications have to also be virtualised;
7. Launch/support a feasibility study to verify that monitoring of cloud jobs can be performed at the hypervisor level, such that monitoring is independent from the virtualised applications;
8. Upgrade current metadata catalogues to support http(s) endpoints and S3-like metadata;
9. Explore feasibility of running BitTorrent on grid sites.

Point 1 is required to give simple and free access to cloud computing middleware to institutes and organisations wanting to contribute resources to the grid via cloud. Assuming that the middleware is simpler to deploy and operate, significant cost savings should be made by institutes and organisation running the middleware, compared to current solutions. Further, assuming that all application software will be distributed using virtual images, no special operations should be required for any given user communities.

Point 2 will help ensure that in the medium term, all cloud services will be interoperable and interchangeable. This is important to avoid vendor locking and ensure that the grid remains homogeneous in terms of grid development environment.

A key difference between grid and cloud is the ability for grid to support collaborative work. A corner stone to this is the grid concept of the Virtual Organisation (VO). In order to support the VO model using cloud security primitives, we need to bridge services such as VOMS and the cloud. This is what point 3 will provide.

In order to improve flexibility, reliability and resilience of the grid key services, if not all services, services should be virtualised, such that they can be moved and replicated easily. Once they are virtualised, which is what point 4 proposes, their deployment will be facilitated. For such an approach to work, solutions such as rBuilder from rPath should be explored to ensure that the virtual image are maintained with the necessary security updates and general support of underlying operating systems alongside the service and application software.

A significant operations cost and hurdle in grid adoption is the need for custom installation of application software on the different data centres used to run jobs from corresponding user communities. Point 5 gets rid of the necessity to perform any custom installation. This however means that the application software must be virtualised, including all services that these applications require to run successfully, as proposed in point 6.

The grid, as it is currently defined, is composed of a large number of data centres contributing resources. The grid operations team must be able to monitor the health and performance of the different sites, including grid services they host and the jobs they are currently processing. Currently, sites will be instructed to equip their resources with monitoring probes, such that operations metrics can be recorded and aggregated for operations monitoring purposes. Using non-virtualised resources, as is mostly the case at the moment with grids, the probes are pre-installed on the resources and run along side with the jobs and grid services. If grid jobs and related services are virtualised, for the reasons stated above, then there is the issue of monitoring the execution of the job in its virtualised environment. If grid application developers have to 'bake' themselves software monitoring probes in their application virtual images, it would defeat the purpose of separating the application from the underlying resource. In order to keep the applications completely separate from the monitoring, a solution could be to perform all required monitoring at the hypervisor level – e.g. the virtualisation execution layer on the host resource. This means that the resource owners would have to deploy specially configured hypervisors, but at least this layer would be independent from the user application it is mandated to monitor. This is what is proposed in point 7.

In order for grid users to be able to use S3-like storage solutions in conjunction with metadata catalogues, these have to be upgraded. This is why point 8 proposes that metadata catalogues must be able to support http(s) and indicate that the endpoint is a REST resources (following the ROA principles), as well as understanding key/value pairs as currently featured by S3.

Finally, in order for the grid to leverage BitTorrent for data transfers, as proposed in point 9, the deployment of such technology on grid sites should be investigated. This will not only allow data sources to be exposed as torrents, as in S3, but also jobs to embed a BitTorrent client to access data sources as torrents.

6 CONCLUSION

Cloud computing, even though not new, is currently getting traction, especially in the form of the Amazon Web Services (AWS) commercial offering, with the EC2 and S3 services.

While grid, as defined and provided by EGEE, has a larger scope, the availability of the AWS with its technological choices and simple interfaces is relevant to the grid world.

In this study, we have reviewed the state of the art in cloud computing focusing on the AWS, and related this development to the current state of grid and more specifically the EGEE grid.

The current situation with respect to standards was also briefly reviewed, in order to propose new areas where effort could be usefully invested.

Finally, exploration paths and a number of recommendations were proposed to investigate the opportunity for the EGEE grid to accelerate its adoption using advancements in cloud.

The question "what is the usage pattern that will emerge in the coming years?" remains unanswered and will have to be carefully tracked. A grid composed of a loose collection of a large number of resource providers would probably shape usage patterns differently from a grid where few very large data centres provide most of the resources. Further, "will the majority of the data source be injected in the grid or cloud?", or will it remain outside, as is the case for large medical data sets heavily protected by medical institutes such as hospital and government registries. In a commercial cloud context where data movement is charged to the user, the location of the data and the computing will impact research costs.

None of the resources contributed to the EGEE grid come from commercial offerings, such as Amazon. However, this may change as resource providers consider outsourcing their computing need to commercial providers. Comparing costs between custom construction of large custom data centres like CERN's and the Amazon retail price list, a gap still exists in favour of large custom data centres. This comparison might not, however, be so clear cut in the case of institutes with perhaps less in-house expertise in running large computing fabric.

"Is commercial cloud offering a real competitor for EGEE resource providers?" Since they offer significantly different services, the answer is "no". However, if we ask the question differently, such as "is commercial cloud offering a real competitor to grid resource providers?", then the answer is probably "maybe". However, the role of federating the resources and providing high-level services is still required for the grid to fulfil, with or without cloud being used to provide resources. As cloud services expand, it is likely that high-services will be available as well.

The ability for grid users and resource providers to offload peak activity to commercial cloud computing should be possible. This might relax the need for some institutes to provide large capacity for sporadic usage, with a corresponding low average usage, in favour of outsourcing to cloud providers.

Technologies such as REST, HTTP, hardware virtualisation and BitTorrent could displace existing accesses to grid resources. EGEE has an opportunity to take a role to ensure that the next generation e-Infrastructure is as inclusive as possible, federating both resources from academic organisations as well as commercial providers to ensure it is as pervasive, accessible, performing and cost effective as possible.

We also discussed the benefits of simplicity that cloud technologies propose and the opportunities they offer for grids to better serve its current users and hopefully attract new user communities and accelerate grid adoption. This simplification could also have an impact in lowering the operations cost of large infrastructures, with virtualisation providing a way to reduce the burden on resource providers to setup and maintain complex user environments.

This path would be eased if open source implementations of cloud computing were made available as well as standards put in place to guide user and resource accesses. The choices made by Amazon with S3 and EC2 could be valid starting points for both of these activities.

It is important that new developments are not a distraction from ensuring that current grid users can continue to rely on a production e-Infrastructure for their daily work. Therefore, a roadmap should be defined to include cloud technology in current e-Infrastructures in an incremental and harmonious fashion.

7 REFERENCES

The following is the list of references used in the text:

[1] What is the Grid? A three point checklist, Ian Foster, Grid Today, 22 July 2002, http://www.gridtoday.com/02/0722/100136.html

[2] Summary of the Plan for ensuring the infrastructure needed to meet the Computing Needs for Physics at CERN in the Next Decade, Bird, Cass, Panzer-Steindel, Robertson, CERN/IT, contact authors for a copy.

[3] Technology Review, The Digital Utility, Mark Williams, MIT, http://www.technologyreview.com/Infotech/20227

[4] Architectural Styles and the Design of Network-based Software Architectures, Fielding, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[5] RESTful Web Services, Leonard Richardson, Sam Ruby, O'Reilly, May 2007

[6] Network performance within Amazon EC2 and to Amazon S3, http://blog.rightscale.com/2007/10/28/partners/

[7] Cloud Systems BOF, OGF22, Cambridge, MA, USA, http://www.ogf.org/OGF22/materials/1055/Cloud+BoF+Report+Editedv3.pdf

[8] Commodity grid computing with Amazon’s S3 and EC2, Simson L. Garfinkel, Commodity Grid Computing, February 2007**,** http://www.usenix.org/publications/login/2007-02/openpdfs/garfinkel.pdf

[9] The Digital Ice Age, Brad Reagan, Popular Mechanics, December 2006, http://www.popularmechanics.com/technology/industry/4201645.html

[10] Inside Amazon's web services, Interview with Amazon CTO Werner Vogels, http://www.regdeveloper.co.uk/2007/04/15/amazon_vogels_webservices/

[11] Stock exchange for ‘Grid’ computing?, ICT Results, February 2008, http://cordis.europa.eu/ictresults/index.cfm/section/news/tpl/article/BrowsingType/Features/ID/89550

[12] Clouds are Looming -- and We Love It, Derrick Harris, GRIDToday, April 2008, http://www.gridtoday.com/grid/2284608.html

The following references were used as background information for the elaboration of this report:

[13] The Virtual Evolution of Managed Services, Derrick Harris, GRIDToday, March 2008, http://www.gridtoday.com/grid/2247724.html

[14] Why ‘Grid’ doesn’t sell, GRIDToday, March 2008, http://www.gridtoday.com/grid/2248928.html

[15] There's Grid in them thar Clouds, Ian Foster’s Blog, http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html

[16] Urgent Computing: Exploring Supercomputing’s New Role, Pete Beckman, CTWatch Quarterly, http://www.ctwatch.org/quarterly/articles/2008/03/urgent-computing-exploring-supercomputings-new-role/

[17] Life or Death Decision-making: The Medical Case for Large-scale, On-demand Grid Computing, Steven Manos, Stefan Zasada, Peter V. Coveney, CTWatch Quarterly, March 2008, http://www.ctwatch.org/quarterly/articles/2008/03/life-or-death-decision-making-the-medical-case-for-large-scale-on-demand-grid-computing/

[18] The EGEE-II Project: Evolution Towards a Permanent European Grid Initiative, High Performance Computing and Grids in Action
Edited by Lucio Grandinetti, IOS Press, ISBN 978-1-58603-839-7

[19] 3rd EGEE User Forum Book of Abstracts, Clermont-Ferrand, France, ISBN 978-92-9083-310-9, http://doc.cern.ch//archive/electronic/egee/pub/egee-pub-2008-001.pdf

[20] Amazon S3 for Science Grids: a Viable Solution?, Mayur Palankar and Adriana Iamnitchi (University of South Florida), Matei Ripeanu (University of British Columbia), Simson Garfinkel (Harvard). *DADC'08*, June 24, 2008, Boston, Massachusetts, USA, Copyright 2008 ACM 978-1-60558-154-5/08/06; http://www.csee.usf.edu/~anda/papers/dadc108-palankar.pdf