Cloud Computing - Issues, Research and Implementations

Dr. Sanjay Arora

Director, Shri Aatm Vallabh Jain Kanya Mahavidhalaya, Sri Ganganagar, India

Abstract: ***"Cloud" computing – a relatively recent term, builds on decades of research in virtualization, distributed computing, utility computing, and more recently networking, web and software services. It implies a service oriented architecture, reduced information technology overhead for the end-user, great flexibility, reduced total cost of ownership, on-demand services and many other things. This paper discusses the concept of "cloud" computing, some of the issues it tries to address, related research topics, and a "cloud" implementation available today.***

Keywords: "cloud" computing, virtual computing lab, virtualization, utility computing, end-to-end quality of service

1. Introduction

"Cloud computing" is the next natural step in the evolution of on-demand information technology services and products. To a large extent, cloud computing will be based on virtualized resources.

Cloud computing predecessors have been around for some time now [1, 12, 15, 17, 18, 24, 29, 30, 35, 40], but the term became "popular" sometime in October 2007 when IBM and Google announced a collaboration in that domain [27, 22]. This was followed by IBM's announcement of the "Blue Cloud" effort [23]. Since then, everyone is talking about "Cloud Computing". Of course, there also is the inevitable Wikipedia entry [45]. This paper discusses the concept of "cloud" computing, some of the issues it tries to address, related research topics, and a "cloud" implementation available today. Section 2 discusses concepts and components of "cloud" computing. Section 3 describes an implementation based on Virtual Computing Laboratory (VCL) technology. VCL has been in production use at NC State University since 2004, and is a suitable vehicle for dynamic implementation of almost any current "cloud" computing solution. Section 4 discusses "cloud"-related research and engineering challenges. Section 5 summarizes and concludes the paper.

2. Cloud Computing

A key differentiating element of a successful information technology (IT) is its ability to become a true, valuable, and economical contributor to cyber infrastructure [4]. "Cloud" computing embraces cyber infrastructure, and builds upon decades of research in virtualization, distributed computing, "grid computing", utility computing, and, more recently, networking, web and software services. It implies a service oriented architecture, reduced information technology overhead for the end-user, greater flexibility, reduced total cost of ownership, on demand services and many other things.

2.1. Cyber infrastructure

"Cyber infrastructure makes applications dramatically easier to develop and deploy, thus expanding the feasible scope of applications possible within budget and organizational constraints, and shifting the scientist's and engineer's effort away from information technology development and concentrating it on scientific and engineering research. Cyber infrastructure also increases efficiency, quality, and reliability by capturing commonalities among application needs, and facilitates the efficient sharing of equipment and services."[5]

Today, almost any business or major activity uses, or relies in some form, on IT and IT services.

These services need to be enabling and appliance-like, and there must be an economy-of- scale for the total-cost-of-ownership to be better than it would be without cyber infrastructure.

Technology needs to improve end user productivity and reduce technology-driven overhead. For example, unless IT is the primary business of an organization, less than 20% of its efforts not directly connected to its primary business should have to do with IT overhead, even though 80% of its business might be conducted using electronic means.

2.2. Concepts

A powerful underlying and enabling concept is computing through service-oriented architectures (SOA) – delivery of an integrated and orchestrated suite of functions to an end-user through composition of both loosely and tightly coupled functions, or services – often network based. Related concepts are component-based system engineering, orchestration of different services through workflows, and virtualization.

2.2.1. Service-oriented Architecture

SOA is not a new concept, although it again has been receiving considerable attention in recent years [9, 25, 38]. Examples of some of the first network-based service-oriented architectures are remote procedure calls (RPC), DCOM and Object Request Brokers (ORBs) based on the CORBA specifications [32, 33]. A more recent example are the so called "Grid Computing" architectures and solutions [15, 17, 18]. In an SOA environment, end-users request an IT service (or an integrated collection of such services) at the desired functional, quality and capacity level, and receive it either at the time requested or at a specified later time. Service discovery, brokering, and reliability are important, and services are usually designed to interoperate, as are the composites made of these services. It is expected that in the next 10 years, service-based solutions will be a major vehicle for delivery of information and other IT-assisted functions at both individual and organizational levels, e.g., software applications, web-based services, personal and business "desktop" computing, high-performance computing.

2.2.2. Components

The key to a SOA framework that supports workflows is componentization of its services, an ability to support a range of couplings among workflow building blocks, fault-tolerance in its data- and process-aware service-based delivery, and an ability to audit processes, data and results, i.e., collect and use provenance information. Component-based approach is characterized by [13, 28] ***reusability*** (elements can be re-used in other workflows), ***substitutability*** (alternative implementations are easy to insert, very precisely specified interfaces are available, runtime component replacement mechanisms exist, there is ability to verify and validate substitutions, etc.), ***extensibility and scalability*** (ability to readily extend system component pool and to scale it, increase capabilities of individual components, have an extensible and scalable architecture that can automatically discover new functionalities and resources, etc.), ***customizability*** (ability to customize generic features to the needs of a particular scientific domain and problem), and ***composability*** (easy construction of more complex functional solutions using basic components, reasoning about such compositions, etc.). There are other characteristics that also are very important. Those include **reliability and availability** of the components and services, the cost of the services, **security**, total cost of ownership, economy of scale, and so on.

In the context of cloud computing we distinguish many categories of components: from differentiated and undifferentiated hardware, to general purpose and specialized software and applications, to real and virtual "images", to environments, to no-root differentiated resources, to workflow-based environments and collections of services, and so on. They are discussed later in the paper.

2.2.3. Workflows

An integrated view of service-based activities is provided by the concept of a workflow. An IT assisted workflow represents a series of structured activities and computations that arise in information-assisted problem solving. Workflows have been drawing enormous attention in the database and information systems research and development communities [16, 20]. Similarly, the scientific community has developed a number of problem solving environments, most of them as integrated solutions [19]. Scientific workflows merge advances in these two areas to automate support for sophisticated scientific problem solving [28, 42].

A workflow can be represented by a directed graph of data flows that connect loosely and tightly coupled (and often asynchronous) processing components. One such graph is shown in Figure 1. It illustrates a Kepler-based implementation of a part of a fusion simulation workflow [2, 8]. In the context of "cloud computing", the key questions should be whether the underlying infrastructure is supportive of the workflow oriented view of the world. This includes on demand and advance-reservation-based access to individual and aggregated computational and other resources, autonomics, ability to group resources from potentially different "clouds" to deliver workflow results, appropriate level of security and privacy, etc.

2.2.4. Virtualization

Virtualization is another very useful concept. It allows abstraction and isolation of lower level functionalities and underlying hardware. This enables portability of higher level functions and sharing and/or aggregation of the physical resources. The virtualization concept has been around in some form since 1960s (e.g., in IBM mainframe systems). Since then, the concept has matured considerably and it has been applied to all aspects of computing – memory, storage, processors, software, networks, as well as services that IT offers. It is the combination of the growing needs and the recent advances in the IT architectures and solutions that is now bringing the virtualization to the true commodity level. Not surprisingly, there are dozens of virtualization products, and a number of small and large companies that make them. Some examples in the operating systems and software applications space are VMware1, Xen – an open source Linux-based product developed byXenSource2, and Microsoft virtualization products3, to mention a few. Major IT players have also shown a renewed interest in the technology (e.g., IBM, Hewllet-Packard, Intel, Sun, RedHat). Classical storage players such as EMC, NetApp, IBM and Hitachi have not been standing still either.

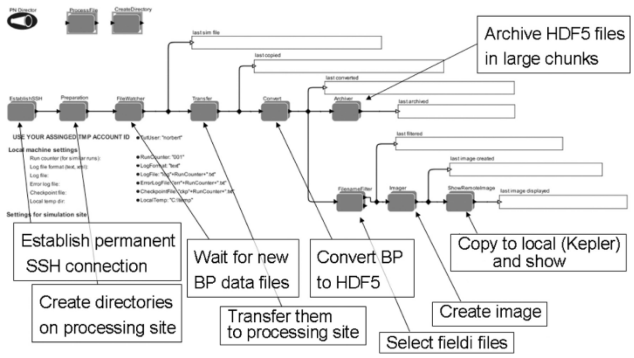

Figure 1: A Kepler-based workflow

2.3. Users

The most important Cloud entity, and the principal quality driver and constraining influence is, of course, the user. The value of a solution depends very much on the view it has of its end-user requirements and user categories.

Paper ID: SR20224104300 DOI: 10.21275/SR20224104300

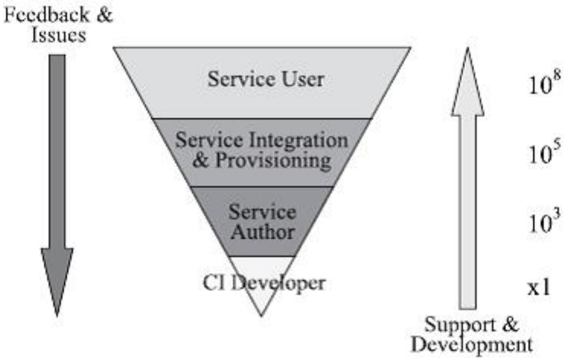

Figure 2: Cloud user hierarchy

Figure 2 illustrates four broad sets of nonexclusive user categories: System or cyber infrastructure (CI) developers; developers (authors) of different component services and underlying applications; technology and domain personnel who integrate basic services into composite services and their orchestrations (workflows) and delivers those to end-users; and, finally, users of simple and composite services. User categories also include domain specific groups, and indirect users such as stakeholders, policy makers, and so on. Functional and usability requirements derive, in most part, directly from the user profiles. Support construction and delivery of content and curricula for these users. For that, the system needs to provide support and tools for thousands of instructors, teachers, professors, and others that serve the students.

Be reliable and cost-effective to operate and maintain. The effort to maintain the system should be relatively small, although introduction of new paradigms and solutions may require a considerable start-up development effort.

2.3.1. Developers

Cyber infrastructure developers who are responsible for development and maintenance of the Cloud framework. They develop and integrate system hardware, storage, networks, interfaces, administration and management software, communications and scheduling algorithms, services authoring tools, workflow generation and resource access algorithms and software, and so on. They must be experts in specialized areas such as networks, computational hardware, storage, low level middleware, operating systems imaging, and similar. In addition to innovation and development of new "cloud" functionalities, they also are responsible for keeping the complexity of the framework away from the higher level users through judicious abstraction, layering and middleware. One of the lessons learned from, for example, "grid" computing efforts is that the complexity of the underlying infrastructure and middleware can be daunting, and, if exposed, can impact wider adoption of a solution [46].

2.3.2. Authors

Service authors are developers of individual base-line "images" and services that may be used directly, or may be integrated into more complex service aggregates and workflows by service provisioning and integration experts. In the context of the VCL technology, an **"image"** is a tangible abstraction of the software stack [6, 44]. It incorporates

a) Any base-line operating system, and if virtualization is needed for scalability, a hyper visor layer,
b) Any desired middleware or application that runs on that operating system, and
c) Any end-user access solution that is appropriate (e.g., ssh, web, RDP, VNC, etc.).

Images can be loaded on "bare-metal", or into an operating system/application virtual environment of choice. When a user has the right to create an image, that user usually starts with a "NoApp" or a base-line image (e.g.,Win XP or Linux) without any except most basic applications that come with the operating system, and extends it with his/her applications. Similarly, when an author constructs composite images (aggregates of two or more images we call environments that are loaded synchronously), the user extends service capabilities of VCL. Scalability is achieved through a combination of multi-user service hosting, application virtualization, and both time and CPU multiplexing and load balancing. Authors must be component (base-line image and applications) experts and must have good understanding of the needs of the user categories above them in the Figure 2. triangle. Some of the functionalities a cloud framework must provide for them are image creation tools, image and service management tools, service brokers, service registration and discovery tools, security tools, provenance collection tools, cloud component aggregations tools, resource mapping tools, license management tools, fault-tolerance and fail-over mechanisms, and so on [44].

2.3.3. Service Composition

Similarly, services **integration and provisioning** experts should be able to focus on creation of composite and orchestrated solutions needed for an end-user. They sample and combine existing services and images, customize them, update existing services and images, and develop new composites. They may also be the front for delivery of these new

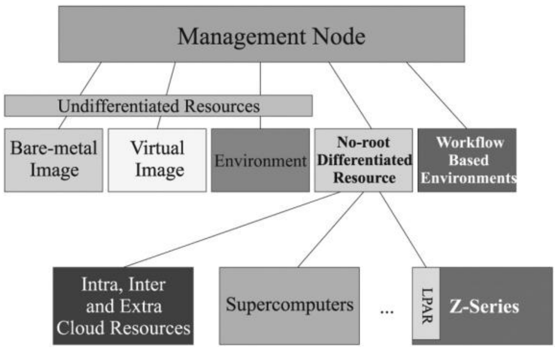

Figure 3: Some VCL cloud components

services (e.g., an instructor in an educational institution, with "images" being cloud-based in-lab virtual desktops), they may oversee the usage of the services, and may collect and manage service usage information, statistics, etc. This may require some expertise in the construction of images and services, but, for the most part, their work will focus on interfacing with end-users and on provisioning of what end-users need in their workflows.

Some of the components that an integration and provisioning expert may need are illustrated in Figure 3, based on the

VCL implementation [6, 44]. The need may range from "bare metal" loaded images, images on virtual platforms (on hyper visors), to collections of image aggregates (environments, including high performance computing clusters), images with some restrictions, and workflow-based services.

A service management node may use resources that can be reloaded at will to differentiate them with images of choice. After they have been used, these resources are returned to an undifferentiated state for re-use. In an educational context, this could be, for example, a VMW are image of 10 lab-class desktops that may be needed between 2 and 3 pm on Monday. Then after 3pm another set of images can be loaded into those resources.

2.3.4. End-users

End-users of services are the most important users. They require appropriately reliable and timely service delivery, easy-to-use interfaces, collaborative support, information about their services, etc. The distribution of services, across the network and across resources, will depend on the task complexity, desired schedules and resource constraints. Solutions should not rule out use of any network type (wire, optical, wireless) or access mode (high speed and low speed). However, VCL has set a lower bound on the end-to-end connectivity throughput, roughly at the level of DSL and cable modem speeds. At any point in time, users' work must be secure and protected from data losses and unauthorized access.

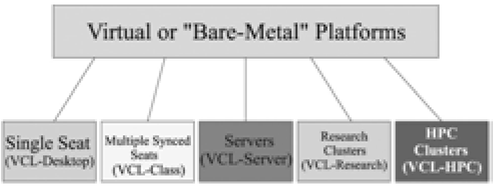

Figure 4: VCL "seats"

For example, the resource needs of educational end-users (Figure 4) may range from single seat desktops ("computer images") that may deliver any operating system and application appropriate to the educational domain, to a group of lab or classroom seats for support of synchronous or asynchronous learning or hands on sessions, one or more servers supporting different educational functions, groups of coupled servers (or environments), e.g., an Apache server, a database server, and a workflow management server all working together to support a particular class, or research clusters, and high-performance computing clusters. Figure 4 shows the current basic services (resources) delivered by VCL. The duration of resource ownership by the end-users may range from a few hours, to several weeks, a semester, or an open ended period of time.

3. An Implementation

"Virtual Computing Laboratory (VCL) – http://vcl.ncsu.edu *is an award-winning open source implementation of a secure production- level on-demand utility computing and services oriented technology for wide-area access to solutions based on virtualized resources, including computational, storage and software resources There are VCL pilots with a number of University of North Carolina campuses, North Carolina Community College System, as well as with a number of out-of-state universities – many of which are members of the IBM Virtual Computing Initiative".*

Figure 5 illustrates NC State Cloud based on VCL technology. Access to NC State Cloud reservations and management is either through a web portal, or through an API. Authentication, resource availability, image and other information are kept in a database. Resources (real and virtual) are controlled by one or more management nodes. These nodes can be within the same cloud, or among different clouds, and they allow extensive sharing of the resources provided licensing and other constraints are honored.

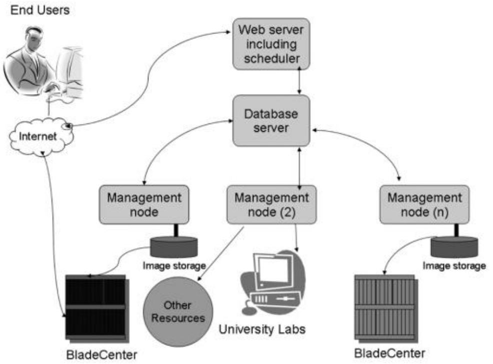

Figure 5: NC State "Cloud"

The VCL implementation has most of the characteristics and functionalities discussed so far and considered desirable in a cloud. It can also morph into many things. Functionally it has a large intersection with Amazon Elastic Cloud [1], and by loading a number of blades with Hadoop-based images [18] one can implement a Google-like map/reduce environment, or by loading an environment or group composed of Globus-based images one can construct a sub-cloud for grid-based computing, and so on. A typical NC State bare-metal blade serves about 25 student seats – a 25:1 ratio – considerably better than traditional labs at 5:1 to 10:1. Hypervisors and server-apps can increase utilization by another factor of 2 to 40, depending on the application and user profile. Our maintenance overhead is quite low – about 1 FTE in maintenance for about 1000 nodes, and with another 3 FTEs in development.

4. Research Issues

The general cloud computing approach discussed so far, as well as the specific VCL implementation of a cloud continues a number of research directions, and opens some new ones. For example, economy-of-scale and economics of **image and service construction** depends to a large extent on the ease of construction and mobility of these images, not only within a cloud, but also among different clouds. Of special interest is construction of complex environments of resources and complex control images for those resources, including workflow-oriented images. Underlying that is a considerable amount of **meta-data**, some permanently attached to an image, some dynamically attached to an image, some kept in the cloud management databases.

Cloud **provenance data**, and in general metadata management, is an open issue. The classification we use divides provenance information into

- Cloud Process provenance – dynamics of control flows and their progression, execution information, code performance tracking, etc.
- Cloud Data provenance – dynamics of data and data flows, file locations, application input/ output information, etc.
- Cloud Work flow provenance – structure, form, evolution, . . ., of the workflow itself
- System (or Environment) provenance – system information, O/S, compiler versions, loaded libraries, environment variables, etc.

Open challenges include: How to collect provenance information in a standardized and seamless way and with minimal overhead –modularized design and integrated provenance recording; How to store this information in a permanent way so that one can come back to it at anytime, – standardized schema; and How to present this information to the user in a logical manner – an intuitive user web interface: Dashboard [7]. Another research and engineering challenge is **security**. For end-users to feel comfortable with a "cloud" solution that holds their software, data and processes, there should exist considerable assurances that services are highly reliable and available, as well as secure and safe, and that privacy is protected. The question of the return-on-investment (ROI) and the total-cost-of-ownership (TCO) is complicated. Figure 6 shows **utilization** of the VCL seat oriented resources by day over the last 4 years.

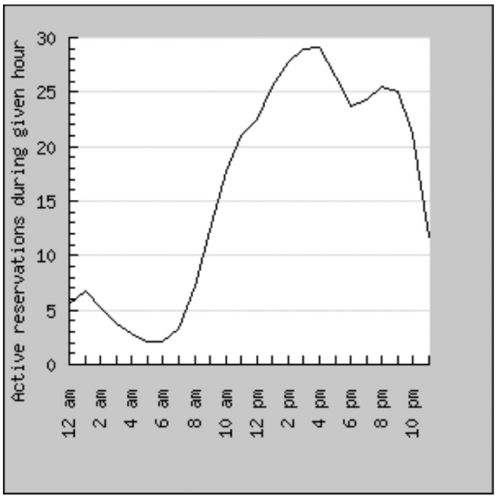

Figure 6: VCL resource utilization

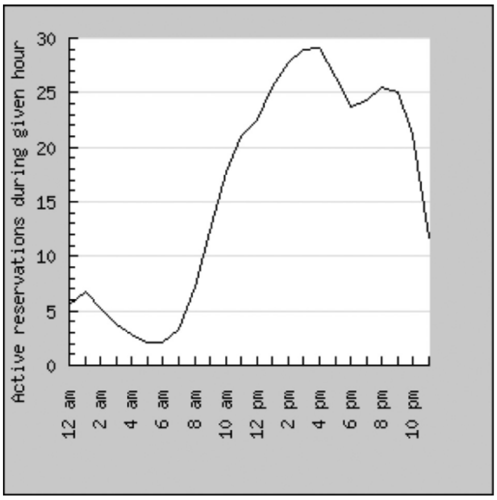

Figure 7: Daily variation in VCL single-seat utilization (averaged over four years)

5. Conclusions

"Cloud" computing builds on decades of research in virtualization, distributed computing, utility computing, and, more recently, networking, web and software services. It implies a service-oriented architecture, reduced information technology overhead for the end-user, great flexibility, reduced total cost of ownership, on demand services and many other things. This paper discusses the concept of "cloud" computing, the issues it tries to address, related research topics, and a "cloud" implementation based on VCL technology. Our experience with VCL technology is excellent and we are working on additional functionalities and features that will make it even more suitable for cloud framework construction.

6. Acknowledgements

I would like to thank my colleagues on the NC StateVCL team for their contributions, insights, and support.

References

[1] Amazon Elastic Compute Cloud (EC2): http://www.amazon.com/gp/browse.html?node=201590011, accessed Dec 2008.

[2] Ilkay Altintas, Bertram Ludaescher, Scott Klasky, Mladen A. Vouk. "Introduction to scientific workflow management and the Kepler system", Tutorial, in Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, Tampa, Florida, TUTORIAL SESSION, Article No. 205, 2006, ISBN:0-7695-2700-0, also given at Supercomputing 2007 by Altintas, Vouk, Klasky, Podhorszki, and Crawl, tutorial session S07, 11 Nov 07.

[3] Ilkay Altintas, George Chin, Daniel Crawl, Terence Critchlow, David Koop, Jeff Ligon, Bertram Ludaescher, Pierre Mouallem1, Meiyappan Nagappan, Norbert Podhorszki, Claudio Silva, Mladen Vouk, "Provenance in Kepler-based Scientific Work flow Systems", Poster # 41, at Microsoft eScience Workshop Friday Center, University of North Carolina, Chapell Hill, NC, October 13 – 15, 2007, pp. 82.

[4] D.E. ATKINS ET AL., "Revolutionizing Science and Engineering Through Cyber infrastructure: Report of the National Science Foundation Blue-ribbon Advisory Panel on Cyber infrastructure", NSF, Report of the National Science Foundation Blue ribbon Advisory Panel on Cyberinfrastructure, January 2003, http://www.nsf.gov/od/oci/reports/atkins.pdf
[5] Ditto, Appendix A (http://www.nsf.gov/od/ oci/reports/APXA.pdf).
[6] Sam Averitt, Michale Bugaev, Aaron Peeler, Henry Schaffer, Eric Sills, Sarah Stein, Josh Thompson, Mladen Vouk, "The Virtual Computing Laboratory", *Proceedings of the International Conference on Virtual Computing Initiative*, May 7-8, 2007, IBM Corp., Research Triangle Park, NC, pp. 1–16.
[7] Roselyne Barreto, Terence Critchlow, Ayla Khan, Scott Klasky, Leena Kora, Jeffrey Ligon, Pierre Mouallem, Meiyappan Nagappan, Norbert Podhorszki, Mladen Vouk, "Managing and Monitoring Scientific Workflows through Dashboards", Poster # 93, at Microsoft eScience Workshop Friday Center, University of North Carolina, Chapell Hill, NC, October 13 – 15, 2007, pp. 108.
[8] D.A. Batchelor, M. Beck, A. Becoulet, R.V. Budny, C. S. Chang, P. H.Diamond, J. Q.Dong, G. Y. Fu, A. Fukuyama, T. S. Hahm, D. E.Keyes, Y. Kishimoto, S. Klasky, L. L. Lao1, K. Li1, Z. Lin⁄1, B. Ludaescher, J. Manickam, N. Nakajima1,