Reprinted from the

Proceedings of the Linux Symposium

July 23rd–26th, 2008
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*

C. Craig Ross, *Linux Symposium*

Review Committee

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*

Dirk Hohndel, *Intel*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat, Inc.*
Matthew Wilson, *rPath*
C. Craig Ross, *Linux Symposium*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
Gurhan Ozen, *Red Hat, Inc.*
Eugene Teo, *Red Hat, Inc.*
Kyle McMartin, *Red Hat, Inc.*
Jake Edge, *LWN.net*
Robyn Bergeron
Dave Boutcher, *IBM*
Mats Wichmann, *Intel*

Cloud Computing: Coming out of the fog

Gerrit Huizenga
IBM Corporation
gh@us.ibm.com

Abstract

Cloud computing is a term that has been around for a while but has been storming into the mainstream lexicon again, although with a lot of confusion about what Cloud Computing really means. Many vendors are jumping into Cloud like solutions and are labeling every new activity as somehow related to cloud computing. This paper explores some aspects of what makes a cloud, and distinguishes cloud computing from provisioning, utility computing, application service providers, grids, and many other buzzwords, primarily by focusing on the technology components that make up a cloud.

Further, this paper will briefly explore the factors that have made cloud computing a popular topic, including the current availability of Linux®, the advances in some of the virtualization technologies, and some interesting evolution in terms of provisioning and virtual appliances, and, of course, some of the current providers of technologies that are debatably clouds today. This paper also makes a projection as to the rise of a new era of Internet scale computing which will be enabled by the cloud and identify some of the technologies that will need to further evolve to make cloud computing as ubiquitous as the Internet.

1 Clouds everywhere! Or is it just fog?

You've seen the buzz by now. Cloud computing is *it*. The next big thing. The future of computing. There will be only five clouds in the future. Computing power is on the verge of being as pervasive as the network bandwidth on the Internet. But, wait a minute. What exactly *is* a cloud, you might ask? And, the answers are a bit more foggy.

Some of the first answers point to Google. There! That's a cloud! Look at Amazon's EC2—the Elastic Compute Cloud—it has the word cloud in its name, it must be a Cloud, right? Microsoft must have a cloud, right? Or, another favorite response: We'll know it when we see it.

Probably the best way to start to understand what cloud computing might actually mean is to look at some of the existing technologies that are *not* cloud computing. There are quite a few of these, and all have some distinct properties of their own. As it turns out, cloud computing encompasses a number of the properties from these technologies which is probably why they are so commonly confused.

Some of the alternate technologies covered here include utility computing, grid computing, cluster computing, and distributed computing. We'll also look at Application Service Providers (ASPs), Software as a Service (SaaS), and Hardware as a Service (Haas). Later, this paper will also briefly show the relationship between cloud computing and Software Oriented Architecture (SOA).

After examining the technologies that exist distinctly from clouds, this paper examines the challenges in the industry are driving a push towards cloud computing, and how cloud computing evolves from a number of existing technologies. Cloud computing expands beyond these current properties, usually by drawing on the best techniques present in those pre-existing technologies. Clouds are often seen from a user perspective, but creating, managing, and ultimately implementing a number of the advanced capabilities of clouds is a key topic discussed later in this paper.

Finally, we'll provide some insights on the evolution of clouds, where we *really* are with respect to clouds in the industry, and when cloud computing will be as prevalent as the Internet.

2 Technologies that are *not* cloud computing

Everyone always wants to start with the question: So just what *is* cloud computing? And, I prefer to start that answer with an analysis of what common buzz words and technologies are *not* cloud computing. And, by means of that path to the answer, I will assert that cloud computing is actually something new—not just a new name for an old technology. There are plenty of old technologies in this space, and a few of the more common technologies that are confused with cloud computing are utility computing and grid computing. Starting with those, we will begin with a short description of what they are, and in each case, show how each relates to cloud computing. Once those related technologies have been put into context, we'll dive into what cloud computing actually is.

2.1 Utility Computing

So far, most of the industry buzz around cloud computing has actually been based on providing computational capability to end users, usually with the potential for charging for the use of that compute power. Basically, with utility computing "somebody else" owns the computers or compute resources, manages them, and sells you access to capacity. This form of computing draws its name directly from the public utilities which handle the creation/management of the resources that you use every day, such as water, electricity, natural gas, etc. As an end user, you don't have to worry about managing those resources nor do you generally worry about access to those resources other than some standardized means of tapping into that resource. Then, based on the resources that you use, you get a bill.

Utility computing eliminates the need for you to acquire and manage your own compute resources, eliminates the start up costs in acquiring capital, configuring machines, performing the basic systems management and systems administration, and allows you to focus your efforts on simply running your application. Of course, in the utility computing model, *someone* still has to buy that hardware, manage that hardware, provide access, security, authentication and, in some cases, even the basic applications that allow you to run your application. However, the term *utility computing* has no particular implication as to what systems are provided, what interfaces are available, or how applications are developed, provisioned, or otherwise utilized within a utility computing service. So, in essense, utility computing is really focused on a means of using hardware managed by someone else—be it another company, or perhaps a division of your own company, government, educational or research institution.

Utility computing is often viewed as a type of use case provided by several other types of computing, such as Grid Computing or Cloud Computing, although effectively anyone with a computer could provide access to their resources, potentially with cost recovery for the time and services used, and that would effectively be utility computing. Utility computing in general does not define the means of access to the compute resources, the existence or need for any APIs, or even what types of workloads would be supported by the underlying compute resources.

Utility computing could also refer to access to storage and the related cost recovery charged by those service providers.

2.2 Grid Computing

Grid computing is an idea which started with a vision of making compute resources sharable and broadly accessible, ultimately providing a form of utility computing as described above. Grid development, though, has historically evolved from a deep computing research focus which has been heavily backed by access to large compute clusters. Most of the interesting grid-related projects are those doing deep research with cluster aware programs, developed with a willingness to use a grid-aware library such as the Globus Toolkit™ which facilities compute and data intensive applications which typically require high levels of inter-machine communication. These toolkits and applications provide the ability to locate services within the grid, provide for communication between processes, simplify the ability to partition a workload into a grid-aware application, and provide the application with secure access to data. The Globus Toolkit has recently evolved from a library approach to a more Web-centric approach based on the Web Services Definition Language (WSDL) and the Simple Object Access Protocol (SOAP) to encapsulate interprocess communication into Web-based protocols.

Most grids today provide a utility computing model, including charge-back capabilities. These grids provide access to single machines or to entire clusters of machines in most cases. Further, many of these grids assume that each machine in the cluster is running similar or identical software packages. The packaging mechansims that are typically used in Grid envrionment help to ensure that all machines are running the same software packages, have access to similar/identical versions of the base libraries, and provide the same sets of grid middleware whenever possible. They all typically allow the installation of additional packages as well, but the primary goal is on the build-up of a base stack which is as close to identical between machines as possible.

Grid computing has also recently evolved its use cases to the point of supporting commercial datacenter workloads, although it is much more difficult to find examples of commercial data centers running Grids for general purpose workloads today.

While Grids operate as providers of utility computing, they can also be used as a means of organizing a local, regional or corporate data center into a high-powered computing engine for research. It is a model which is highly targeted to provide simplification of management and processing for compute- or data-intensive workloads. However, its design strongly favors the scientific computing community and is only just evolving, at least conceptually, into a more general-purposed paradigm.

When we look at cloud computing, we'll see that it bears many similarities to grid computing, and, in fact, borrows many concepts from grid computing. In the future, it is also quite likely that cloud computing and grid computing will have a high degree of convergence because both models share many of the same core principles and have a very similar vision of the ultimate state of the computing world. Ian Foster, often viewed as the father of grid computing, also writes occasionally about clouds in his blog.[1]

2.3 Cluster computing

Cluster computing is again somewhat similar to Grid computing, and, in many ways, can most easily be viewed as a subset of Grid computing. Cluster computing is typically based on a set of machines with shared access to storage, all operating as part of a single workload. Workloads are often designed to be cluster aware and are often designed around a shared-nothing workload which limits the need for a coherent locking model, or a shared-everything model, usually based on some form of a distributed locking model. Clusters are typically used for many of the same workloads that are targeted towards Grids, although many clustered installations use a more limited form of inter-process communication, fewer libraries, and, quite often, more custom programming. Classic models for dividing a problem up into discrete components, such as the calculation of a Mandlebrot set or projects like the SETI (Search for Extraterrestrial Intelligence) project can easily divide a large working set into a large number of discrete problems.

Clusters differ from grids in a few visible ways. First, clusters are very homogenous with respect to the hardware and software installed. Most of the entries in the Supercomputer Top 500 list[2] tend to be large numbers of identical machines effectively operating as one massive multiprocessor computer. Where Grid computing tries to keep the software stack close to identical, large scale clusters or supercomputers tend to have an *identical* stack such that every machine is an identical cog in the great supercomputer. Supercomputers are typically viewed as a single aggregate machine with their throughput measured as the sum of the potential throughput in each machine.

Also, each cluster is typically designed to run one "job" at a time, typically a long running job whose goal is to address some problem so large that it would be nearly impossible to tackle in any other conventional sense. This usually means that the number of actual workloads which are available to be run on any given cluster is a very small list of applications over any particular period of time.

Cluster computing, however, brings some very interesting lessons of its own to Cloud Computing. One of the first of those stems from the fact that today's supercomputer is tomorrow's mid range computer. Clusters have started by using very high processor counts and system counts as the way to get to their very high compute limits. And with these large numbers of systems (as many as 128,000!), the tools to install and manage those systems become very important. Simply installing the operating systems and applications can become a bottleneck without excellent tools. And, managing those systems in

[1] `http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html`

[2] `http://www.top500.org/`

the face of potential failures, firmware updates, network reconfigurations, and so on, is critical to keeping those expensive systems operational. One of the key properties for simplifying the management of those systems tends to rely on their relative homogeneity, which we'll come back to briefly when discussing cloud computing.

2.4 Distributed Computing

Again, a subset of Grid Computing, Distributed Computing is worth an independent mention in this space, primarily because it is the form of computing which assumes that multiple workloads operate across many machines, working together to solve a particular problem or implement a business workflow. In the most common cases, distributed computing can be a set of coordinating compute based workloads, distributed within an intranet or across the Internet, each contributing something to the end result. Distributed computing has less dependence on similarity of operating systems, platforms, or internal data representations that Cluster computing does. Instead, the real focus is on providing either a method for communication—either very tight coupling in the case of something like grid, or potentially very light coupling in the typical SOA deployments. However, the end result is that the compute activities for a single operation are distributed across a wide range of hardware and software, where, at the extreme, those couplings must be well-managed; creation of the various workloads must be coordinated; communication points and protocols must be well-defined and managed at some higher level. In some cases, distributed computing relies on a single "parent" process which coordinates all of the components of the workflow. In other cases, the components exist as services which reside at well-known locations. In many business situations, those services or other workload components reside at locations which must be identified in a configuration file or by a similar mechanism and then managed throughout the lifecycle of that portion of the business process.

Just like with Grid and Cluster Computing, Distributed computing has several lessons which are relevent to the evolution of Cloud computing. These include the realization that in most environments today, the workload is actually highly distributed, both within a corporate, research, education or government intranet but also more globally within the Internet. As such, many of these interoperational connections need to be configured and managed. The connections between applications need to be maintained, the network configurations need to be maintained, and, on within the networking space, security between applications must also be maintained. These problems are often addressed today by the systems administrator or by the author of the software. We'll look at how another approach in the context of cloud computing can also help with managing these connections and the relevent security issues.

2.5 Provisioning

The most common solutions that I see today being labelled as Cloud Computing are those with a set of applications and the ability to provision, or deploy, those applications within a utility computing framework. And, to be clear, I believe that provisioning and utility computing are aspects present around cloud computing, but a simple provisioning capability on top of utility computing is not cloud computing. Further, most of the provisioning solutions I see in use today are capable of deploying a single application or a single set of applications onto a single machine. While that may be a start towards cloud computing, it is honestly a very small start.

Provisioning capabilities come in many forms. Those used today in Cluster computing typically specialize in deploying the same operating system and applications to thousands of machine nearly simultaneously. In the recent past, applications like Tivoli's Provisioning Manager® (TPM) provides the ability to deploy and configure operating systems and complex applications to bare metal or, recently, to any of several hypervisors. Provisioning solutions within utility computing environments today, such as Amazon.com®'s Elastic Compute Cloud (EC2) tend to provision single images, usually wrapped as a virtual appliance, to one of a few pre-set virtual platform configurations.

Provisioning can include anything from deploying an application onto an already running operating system, up through deploying a set of complex virtual appliances onto a newly configured set of virtual machines, complete with virtual networks and configured access to storage. There are a number of software applications to aid in provisioning, including 3tera's AppLogic which allows graphical display of the connections between virtual appliances that are to be provisioned.

Provisioning, including complex, multi-tier virtual appliance provisioning is a key component of cloud computing. However, one of the larger gaps in provisioning related to cloud computing is the definition and standardization of the virtual appliances as well as tools to help create those virtual appliances. Further for true adoption and deployment, the ability to manage a catalog of virtual appliances will be key.

2.6 Application Service Providers

A growing trend over the past 10–15 years has been the growth of the application service providers (ASPs). They range in nature from more conventional business applications, like hosted version of SAP, to the more current excursion of Google into Google Docs. Common questions on cloud computing still include references to ASPs, which typically just host one or more applications which allow users to use those applications without having to install or manage those applications themselves. However, there is very little flexibility on what applications a given provider makes available, and no true elasticity of the underlying resources that would be possible from true cloud computing. As we will see shortly, one of the major benefits of cloud computing is a level of elasticity in the use of compute resources and an underlying dynamic infrastructure.

2.7 Software as a Service

The new and updated term for ASP is Software as a Service (SaaS), although as the internet and the capabilities for software hosting have matured, there have been some subtle evolution of the term. Historically, ASPs provided physically isolated systems for different customers, sometimes even going as far as to provide physically isolated networks or VPNs for their larger customers. Today, many SaaS environments provide isolation based strictly on successful authentication. This in some ways is a reflection of the relative maturity of security isolation solutions more than perhaps anything else. Another shift has been a transition towards native Web interfaces on top of today's software applications. Historically, most provided their own UI (or GUI). Today, most front ends are very Web-friendly, with SOAP or REST based interfaces.

Ultimately, though, ASPs, SaaS, and Utility Computing are examples of services that can be provided by a Cloud computing environment. And, the biggest difference between these models as provided today and their deployment on a cloud computing-based environment is that the underlying management of the platforms and services will be be transparently managed and will allow these configurations to be deployed by more than just a few providers with highly sophisticated support staffs.

2.8 Hardware as a Service

Since we can offer software as a service, why not offer hardware (or platform) as a service (HaaS, PaaS)? Today a number of utility computing providers are effectively doing that. Amazon's Elastic Compute Cloud (EC2) or IBM's Research Compute Cloud (RCC) are effectively offering direct access to hardware to their respective customers. In both of these modern cases, the interesting difference is that the hardware being offered up is typically virtualized. This allows some additional functionality and flexibility on the back end because it is now easier to over-provision resources which helps improve overall utilization of the hardware resources. And, this improved utilization enables an improved thermal footprint in the data center, meaning that there is reduced power consumption for compute power and most likely reduced cooling needs as well.

Ideally, that flexibility provides yet another lesson which is applied in cloud computing. Specifically, that separation of the physical resources from their virtualized counterparts allows for some additional management benefits within the data center which implements a cloud computing. And, the proof points with current providers validates that the technologies have advances to the point where we can separate the physical resources from the virtual resources. VMware's VMotion product provides some hints as to what might be possible in that space if correctly harnessed under cloud computing.

Further, recent advances in Linux which allow the dynamic addition of memory or processors to a physical machine, and the ability to pass those capabilities on to an instance of Linux running in a virtual machine, enable a level of elasticity in the hardware which will ultimately enable the most efficient use of resources in cloud computing.

2.9 Software Oriented Architecture (SOA)

Rather than extol the virtues of SOA, we'll suffice to say that SOA is sufficiently prevalent within Enterprises today that any solution which claims to be Cloud Computing must clearly enable SOA as a model for application deployment. Clouds should be effectively deployment platforms which enable advanced models such as SOA. And, based on our earlier look into provisioning and our quick looks at Software and Hardware as a Service, it should be clear that the ability to deploy components of software applications as virtual appliances would be an extremely powerful building block for creating a cloud of applications and services.

3 If that's not Cloud Computing, What *is*?

Thus far we have avoided any real discussion of what Cloud Computing is, although we've dropped a few hints along the way. But why has the buzz on cloud computing started now? What has made this topic jump so quickly to prominence? To understand that, let's look at a few of the pressures on data centers, compute power, and the challenge of creating and deploying new applications quickly.

3.1 First, how did we get here?

The rise to prominence of Cloud Computing stems from several sources. One is clearly related to the marketing hype engines that are always looking for something new. Directly related to that, Google's CEO, Eric Schmidt, Ph.D., has been widely quoted on his definition, specifically:

> It starts with the premise that the data services and architecture should be on servers. We call it cloud computing—they should be in a *cloud* somewhere. And that if you have the right kind of browser or the right kind of access, it doesn't matter whether you have a PC or a Mac or a mobile phone or a BlackBerry or what have you—or new devices still to be developed—you can get access to the cloud. . .

This is a powerful vision and it is possible to see some of the same goals as utility Computing, grid computing, Application Service Provider(s) (ASPs), and Software as a Service (SaaS) embodied in this thinking, as well as much more. For instance, the ubiquitous access to software and applications from hand held devices is contained within this vision as well. The definition provides a powerful vision as well as something of a marketing and hype alignment vehicle, without getting into any of the pesky little details that go with that vision.

However, even that short vision doesn't go into *why* cloud computing might be useful. And to do that, we have to look to the Enterprise Data Center as well as into the pressures that are inhibiting innovation from small companies and individuals.

3.2 The IT Crisis

We have been approaching a crisis in Information Technology for quite a while now, and that impending crisis has been forcing a lot of innovation into methods of avoiding that crisis. What crisis, you ask? Okay, it may not be a real *crisis*, per se. But, throughout the IT industry, the cost of managing hardware, operating systems, and applications has been on the rise for a long time. In fact, that cost has gone up to the point that some analyst figures suggest that IT management costs in the data center range from 25 to 45 percent of the total IT budget.[3] That means that money which could otherwise be targeted towards specific new development in support of a company's value-add is instead going directly towards maintenance of just the server and software environment they already have. That directly limits a company's ability to invest in innovation or increased capacity in a way that impacts a company's bottom line. What kind of solution would substantially reduce those IT management costs?

Or, viewed from another direction, the cost of supplying existing servers with electricity and cooling data centers has escalated to the point that energy costs are now exceeding the costs of the actual hardware in many data centers. And, those power costs are often going to support capacity that is only needed in peak situations or sometimes to support failover capacity. As a result, data centers are spending a lot of money on equipment and power which is not directly contributing to the company's revenue most of the time. This waste capacity flies directly in the face of the *Green* movement as well—data centers that are only 10 to 20 percent utilized are still consuming precious raw materials and contributing to pollution and global warming without much value-add to organizations. What if there were

[3] I even saw a recent Microsoft presentation on Hyper-V indicating that number was as high as 70%!

a paradigm shift which substantially reduced the waste computing cycles, or enabled better sharing of existing computing cycles?

And, viewed from a third perspective, the time to develop and deploy new applications has been rising, especially as space, access to power, and complexity of the development environment has increased. Many innovators today have to request new hardware from their IT department; that hardware needs to be ordered, and, when it arrives, it has to have a place to be installed that has sufficient space, power, and cooling. Once installed, the OS must be installed, key applications must be identified and installed, and, oh, make sure that all the applications you've selected are inter-operable, and then, at last, you are ready to begin development. That cycle in many companies, both small and large, often approaches 3 or in some cases, even 6 months from "idea" to "ready to develop." What if that development cycle could start within hours, either based on a commercial provider's offering of compute cycles, or even your own enterprise's existing compute resources?

Cloud computing provides attractive answers to all of these scenarios. Now if only we could figure out the definition of cloud computing that provides all those answers!

4 Cloud Computing: A Vision

In many ways, the Internet provides an ideal model for Cloud Computing. The Internet provides bandwidth to everyone and happens to hide nearly all of the details of the underlying mechanism of the hardware providing that access to network bandwidth. From a user perspective, compute power should ideally be as ubiquitous for the end user as network bandwidth is today. In fact, some people have suggested that there is a Cloud with a capital *C*, just as there is an Internet with a capital *I*.

A simple way of stating this would be that "Cloud Computing provides ubiquitous access to compute resources for any user, anywhere."

Okay, so that's a pretty simple vision statement, but what does it say for people who want to do more than access a web service (we can already do that) or instantiate a pre-wrapped appliance on a utility computing service provider's platform? What is the development model? How do I get access to a machine to do proprietary development? How to I handle the set up and installation, configuration, and management of the unique development environment that I need for my development? How do we handle licensing for those software applications that aren't part of the open source ecosystem—or even those that are part of the open source ecosystem but still have maintainence fees and licenses for operation?

As an example, suppose I wanted to run Red Hat Enterprise Linux with Oracle and Rational ClearCase® installed? What if I want to implement a 3-tier database, middleware, web server environment? Our vision is a little bit lacking on the finer details of how cloud computing is actually deployed and made available.

Further, suppose that I want to be a cloud computing provider or perhaps even share some of my existing compute capacity with other people—maybe even charging some minor fee for access to my unused cycles? I guess the vision leaves out some of those details as well. In fact, given the vision and the hype, it is really unclear as to whether or not I could even be a cloud computing provider. Clearly, we need some more definition about what clouds actually are, how one creates them, maintains, them, and what a cloud actually can provide today.

4.1 Vision: Meet Reality

It is time to seperate our cloud into two distinct points of view. One point of view will focus on what services and capabilities a cloud *provides*. The other point of view will look at what technologies are present within a cloud. These two views will allow us to distinguish a user of a cloud from a maintainer of a cloud. And, buried in this analysis is the assumption that there will be, for the foreseeable future, more than just one cloud in the sky. Specifically, much like our earlier analogy of the Internet, the cloud will be the composite view of all of those individual clouds which will initially spring up in isolation. Following the existance of many clouds, there is the hope that someday, much like the view of Grid computing, all of the providers will be loosely connected, again like the Internet, to make a single, ubiquitous view of a single, well-connected cloud.

4.2 What services does the cloud provide?

This is the easy question, the question that is the most visible to end users. Specifically, the cloud provides a set of services in the form of utility computing, or even grid computing to end users. Those services could, in theory, be physical hardware or virtual hardware. They could be operating system instances or virtual appliances. They could be operating systems instances with a catalog of software which can be easily installed on them, or which can receive custom software written by and provided by the end user. They could provide for simple Internet connectivity or perhaps they even provide some level of access to virtual private networks (VPNs) or virtual LANs (vLANs) so that the end user could deploy multiple servers cooperating with some level of security protection to help in isolating proprietary data.

These clouds could provide access via the Internet—but they could also be wired directly into private intranets, either physically or virtually, enabling the applications running on the cloud to have access to data or applications residing in an end-user's internal, private network. The Cloud provides an adjustable number of resources, be they physical machines or software appliances, where the user can adjust the number of machines running their workload based on demand.

Of course, as we expand the level of definition of clouds, the astute observer will question whether these clouds provide enough security or reliability to satisfy all users. For instance, would two Wall Street trading companies both put their private applications and data on the same cloud? Do we have strong enough security isolation in place today in our operating systems, hypervisors, virtual LAN technologies, virtualized storage access to enable true and safe isolation between competitors?

What about the latency of access? If this cloud is "just out there" somewhere, how long does it take to get data between any set of applications in the cloud or between the cloud and other internal machines? What is the bandwidth between your machine and the cloud's environment—will I get the bandwidth and latency that I need for *my* application from the cloud today? Does it really have the ability to provide the services I need with the security, performance, bandwidth, latency, and availability that I need from my provider of ubiquitous computing?

Of course, the answer to that last question is a bit "It Depends." For some workloads today, clouds as a flexible, elastic provider of utility computing will do just fine. The same is true for Grids today, and that is why they are heavily used by some workloads, most commonly scientific workloads. Clouds may provide a bit more flexibility in terms of the workload supported today, but there is a long way for clouds to evolve before they are ready to support the needs of all consumers of the cloud computing resources.

Of course, the answers to some of those security, availability, performance, latency, and bandwidth questions might change if an enterprise could effectively build its own, in-house cloud. With direct access to their local SAN, with access to some of the business services that may not be hosted in the cloud, such as their print services, their LDAP services, their nearby connections to desktops, etc., some of these problems that aren't resolved globally may be addressed in a more localized implementation of clouds. We'll come back to that problem more in a little bit.

Finally, a well designed cloud based environment can enable a variety of scenarios for the end users of the cloud, including test and development configurations; the ability to deploy SaaS; the ability to deploy HaaS, aka Platform as a Service (PaaS); the ability to deploy SOA components; the ability to deploy virtual worlds or gaming environments on demand; and many more common workloads.

4.3 What does it take to *build* a cloud?

This question gets a little harder to answer, and most of the common cloud implementations today are managed by top-notch IT staff, explicitly hiding the details of what goes into making a cloud so that the consumers don't have to deal with it at all. But if someone wants to build their own cloud, they'll have to have a firmer grasp of exactly what goes into a cloud and what components they will need to build or assemble, along with some idea of what the cost for managing that infrastructure is going to be.

For simplification, I'm going to suggest a rough blueprint for what components go into a Cloud. It is definitely possible to vary from that blueprint, and to optimize within the blueprint and potentially still be a cloud computing environment. However, this blueprint should enable you to decide what components you may need to have on hand to build a cloud or to evolve your own computer center into a cloud configuration.

4.3.1 Virtualization

At the very core of cloud computing, I'm going to start with what could be a contentious choice but I'll spend more time justifying that choice later in the paper. That first choice is that any good cloud in this point in time should be built on top of a virtualized platform and that all resources in the environment should be virtualized. This includes not just the platform, but storage and networking as well. While it is possible to get the appearance of having a cloud without virtualization, I'm going to go out on a limb and suggest that non-virtualized solutions have limitations in the flexibility that will in time become a hallmark of cloud computing. This means that the base platform in the case of, say, an Intel®or AMD®-based processor should be virtualized by something like VMware®'s ESX®, Microsoft®'s Windows Server 2008®Hyper-V®, some version of Xen™, or any similar hypervisor. For non-Intel/AMD based machines, such as IBM®'s POWER®family of processors, PowerVM®[4] would be an appropriate choice and the IBM mainframe provides virtualization in several forms, such as z/VM®or z/OS®.

Today, most cloud-like deployments are based on a single underlying class of platform, although most enterprises are made up of highly heterogenous environments. In an ideal world, the cloud will include all of those platforms as the basis for cloud computing, and the greater vision clearly postulates that support for heterogenous platforms over time. But for our initial blueprinting activity, we'll start with the simplifying assumption that all of the machines are of relatively the same type, and, more importantly, can all run the same hypervisor.

4.3.2 Virtual Appliances

Next, we need a repository of applications to deploy in our cloud. And, since our cloud need not be restricted to the applications that someone else has created, we will need tools to somehow package those applications for deployment within our cloud. For that, I'd recommend that we start with virtual appliances, which are essentially a packaged version of software and an operating system, ready to run on a hypervisor. A number of companies have started down this path, including VMware and their appliance marketplace[5] or companies like rPath™[6] and their rBuilder™ tool[7] and their appliances.[8]

Since these are virtual appliances, they need to be built for a particular type of hypervisor, be it a Windows/VMware image or a Linux/Xen image. This is where a simpler environment makes it easier to build your own prototype—the more hypervisors that your virtual appliance needs to support, the more complex it is to create a cloud environment. Constructing these virtual images is one of the more challenging aspects, although luckily there is a lot of experience in this space now, with more emerging all the time. Most cloud-like environments today are building multiple versions of virtual appliances from the same sources, such as a Windows/VMware and Linux/Xen at the same time, which generally ensure that any of the appliances built at that time are as close to identical as is reasonably possible.

Another aspect to building these appliances is to understand what format they will be created in. A glance through the rBuilder appliances mentioned earlier shows that today it is possible to build in at least a dozen formats, and that just covers Intel/AMD platforms! If you wanted to build for other processor types or hypervisor technologies, that number will go up from there. Luckily the market will likely resolve this issue one way or the other before long—either a few key virtualization technologies will emerge, or the tools will evolve to support builds for multiple environments at build time.

The VMware and Xen communities are looking into using the Open Virtualization Format as a wrapper format for virtual appliances and that format is being broadly standardized by the Distributed Management Task Force (DMTF).[9] There is a proposed project to create open source tools for managing OVF files that will hopefully be under way by the time this paper is published.[10]

[4] http://en.wikipedia.org/wiki/Advanced_Power_Virtualization

[5] `http://www.vmware.com/appliances`

[6] `http://www.rpath.com`

[7] `http://wiki.rpath.com/wiki/rBuilder`

[8] `http://wiki.rpath.com/wiki/Virtual_Appliances`

[9] `http://www.dmtf.org/newsroom/pr/view?item_key=3b542cbc5e6fc9ede97b9336c29f4c342c02c4e9`

[10] `http://code.google.com/p/open-ovf/`

4.3.3 Provisioning

Once you have built a virtual appliance, you also need to have some understanding of how to provision that virtual appliance. Provisioning in this context means that the appliance will need to be able to be installed on your virtualization platform. Luckily, the OVF format described above has a place within the file which allows the builder to store some basic information about how to set up, configure, and deploy the virtual appliance. This might include things like configuring virtual LANs, configuring access to local or shared storage and any other configuration related to the virtual environment (such as how the domU[11] is configured in Xen, or how to boot the virtual machine).

Provisioning can also be expanded beyond the basic deployment of a single virtual appliance containing a single application stack to deploying either multiple copies of the same virtual appliance or more complex sets of virtual appliances. As an example, it would be possible to provision a three-tier application, such as a database appliance, a middleware appliance, and a web front end appliance. This allows for appliances to be created as building blocks and deployed in sets based on a specific need. This allows for a level of customization without the need to build a large number of highly specific virtual appliances. Also, it allows for some elasticity in the number of appliances deployed—for instance, in the database, middleware, and web front end appliances—it would be easier to deploy additional web front-end appliances as the workload increases, or additional middleware appliances depending on the type of workload.

OVF also has the ability to store multiple virtual appliances in the same wrapper, including all of the instructions to deploy the full set of appliances. To date, there is no solid provisioning tool which generally decodes that information, although the open source OVF tool will help extract that information soon. There are also some tools such as TPM which could take that input and convert it into a provisioning flow. There are limited tools today which provision virtual machines, although one interesting one comes from 3tera's™[12] AppLogic.[13] AppLogic allows for the provisioning of complex workloads onto a physical or virtual machine environment. To date, there are very limited open source projects in this area, though, and it is an interesting are for additional development work.

[11] domU is Xen's name for a guest operating system.

[12] `http://www.3tera.com`

[13] `http://download2.3tera.net/demo/applogic20demo.html`

4.3.4 Virtual Appliances Catalog

Once we have a set of virtual appliances, we need a place to put them. While a small set of appliances can be kept on a laptop or other location, ideally we would like to create a repository of these applications which can be provisioned by end users on request. Also, we would like for our end users to be able to store their own applications in the virtual appliance catalog. This repository of appliances could be something as simple as a directory with virtual appliances in it, sorted by name. Or, it could be a complex hierarchy of images built for a variety of virtualization environments. Again, for simplicity, I'm going to recommend the flat directory, possibly with a simple web-based front end to view those images. Ideally, that web front would allow end users to select one or more images to deploy to the set of machines in your cloud. These machines would all have a hypervisor installed on them already, and your provisioning software would scatter the virtual appliances intelligently amongst your physical resources.

With some additional intelligence in the deployment sofware, your virtual appliance catalog could contain images that worked on multiple hypervisors or multiple machine configurations. In particular, the OVF format contains information that identifies the environments to which that virtual appliance can be deployed. That would allow you to deploy virtual appliances to a variety of hardware, provided that you had either built each virtual appliance for multiple platforms.

Now, for simplification, I proposed the virtual appliance catalog as something containing just virtual appliances. However, it would be possible to also have that catalog contain base virtual appliances, perhaps a distribution trimmed down to just what is necessary for supporting a software stack, and a set of applications which could be streamed to the pre-built, pre-defined software appliances. This configuration would provide a bit more flexibility in terms of building blocks and perhaps reduce the number of specialized virtual images slightly while increasing the flexibility of those environments. That may be as simple as today's ability to use a package manager or install tool on a running image. Or it might be something more like a pre-installed application built as a union filesystem image, which could simply be mounted on top of an existing appliance. The latter would allow more rapid provisioning of images and ideally leave less configuration to the end user.

4.4 Cloud computing is really just that simple?

Is that all there is? Some hardware, virtual appliances, a catalog, and an ability to provision? From a user perspective, yes, that is one view and it is sufficient to provide a basic cloud configuration. And, this is the basis for Amazon.com's Elastic Compute Cloud (EC2)[14] and IBM's internal Research Compute Cloud (RCC). This is also a solution which is being rolled out to a number of other sites by an IBM team.[15][16]

However, there is another view of cloud computing as described by Google and IBM.[17][18] In this view, the focus is on a shift in the programming paradigm to solve more problems using a *huge* number of computers, a la Google's infrastructure. Again, this starts with some hardware, in this case, lot of it; one or more applications, although not necessarily in the form of appliances this time; an ability to provision a parallel style of workload; and, more uniquely, a variant of Google's internal MapReduce[19] algorithm, potentially based on the open source Hadoop[20] code. This code enables the workload to be divided quickly among hundreds or thousands—or even tens of thousands of machines—enabling some forms of complex, data intensive processing to be smashed into thousands of very small workloads which can return results in a fraction of the time.

In some ways, this model is a variation on distributed computing, which allows workloads to be partitioned into smaller workloads and distributed to a large number of machines. But in this variation, the work is partitioned among a set of machines where multiple machines may compute overlapping results. A "reduce" step winnows out the duplicates, providing a single set of results to the end user. Many believe that this workload will be one of many which take advantage of the forthcoming cloud environments.

4.5 So why all the hype if that's all there is?

So far, we've covered the basis of what make up a localized cloud environment. This definition relies on technologies that mostly exist today and integrates them in a way that makes the deployment of some workloads substantially simpler. But this simplified view is only a subset of the grand vision that many believe is the real direction for cloud computing. In particular the grand vision implies substantially more capability and sharing between the relatively smaller clouds proposed here. However, there are a number of technology gaps between this relatively modest proposal and the grand vision. We'll look at a couple of those gaps here.

4.5.1 Managing thousands of machines

One of the short term challenges for the most basic clouds, as well as a challenge to expanding towards the grand vision of cloud computing comes down to the relatively simple issue of just how to manage all of the computers in a data center. While many view cloud computing by its usage model of effectively providing utility computing, relying on just a few providers with excellent systems administrators to provide all of that capacity is not a very scalable model. And, while the predictions of only five computers in the world have run from the 1950's or 1960's[21] to the current day,[22] the reality is that, by at least some estimates, there are over 25 million servers in the market as of 2005[23] and even if there were a widespread conversion overnight to use one of five mega-datacenters, the time (not to mention the cost) for conversion would be overwhelming. And there is a psychological factor at work as well: most companies aren't ready to trust their core intellectual property

[14] `http://www.amazon.com/ec2`

[15] `http://www-03.ibm.com/press/us/en/pressrelease/23426.wss`

[16] `http://www-03.ibm.com/press/us/en/pressrelease/23710.wss`

[17] `http://www.ibm.com/ibm/ideasfromibm/us/google/index.shtml`

[18] `http://www.google.com/intl/en/press/pressrel/20071008_ibm_univ.html`

[19] `http://labs.google.com/papers/mapreduce.html`

[20] `http://hadoop.apache.org/core/`

[21] `http://en.wikipedia.org/wiki/Thomas_J._Watson#Famous_misquote`

[22] `http://www.guardian.co.uk/technology/2008/feb/21/computing.supercomputers`

[23] `http://www.itjungle.com/tlb/tlb030607-story04.html`

in the form of data or applications to a third party to host and manage.

So, for the near term at least, the proliferation of clouds will evolve, in my own estimation, from existing data centers into a more global cloud computing IT vision. This migration is akin to the growth of the Internet from its rather humble beginnings at the end of the 1970's through the end of the 1980's, where data centers—mostly in the form of university computing centers and some DARPA[24] sponsored sites—were among the first to connect computers locally in a data center with high speed networks, providing the benefits of well connected intranets long before the Internet was available. In fact, the evolution started by creating slow speed connections (based on store and forward technologies like UUCP[25] or BITNET[26]) which provided loose connectivity between well connected, locally managed compute resources. Another parallel between the expected evolution of cloud computing and the Internet can be seen in the handling of security concerns. The first UUCP capabilities were used as a simple mechanism for authenticated users to move files around or to exchange data via email. Later, when ARPANET[27] and its successori, NSFNET[28], allowed more direct access, a number of secured services such as remote shell (rsh), remote copy (rcp), or unsecured services such as finger and fingerd, whois, date and time servers, and basic domain name services, the Internet allowed select services to be offered, typically at no charge to arbitrary users. The evolution of TCP and IP helped bring the Internet to the point that it was merely a conduit for any time of data that any user might want to publish or consume.

Today any number of providers enable end users to consume compute resources along the lines of specific services, from SaaS providers or even services such as Wikis or customizable home pages. However, the visible emergence of Amazon.com's EC2 and some forms of hosting providers starts to show how shared resources can be made available, potentially with charge back. The percentage of compute resources currently available through such providers is still extremely small—some projections suggest only a few thousand machines are available through Amazon.com, for example. And, one of the key reasons for this, in my estimation, is that the management software has not evolved to efficiently and effectively managing large groups of machines to enable sites to develop cloud computing internally, and the sheer quantity of machines that would be needed to support the entire compute capacity of the world—or even some large percentage of that capacity.

While the full gamut of management complexities are too extreme to dive into in detail here, a few of the highlights include such simple things as hardware management, operating system management, network and storage connectivity management, application management, security management, availability management and energy management, to name a few. Often, each of these are managed independently for each machine and each workload in a data center. Or, in some of the best practices, machines are grouped for ease of management, applications are grouped for simplicity, security policies are centralized for consistency, and often emerging capabilities such as energy management are added as an afterthought. Without substantially improved practices for enterprise and data center management, cloud computing as a grand vision will remain as just a dream.

4.5.2 Provisioning Challenges

Earlier we mentioned provisioning as one of the key components of cloud computing. And, honestly, provisioning is really in its infancy, despite a few companies and projects which have started to address the problems. And, many of those companies and projects have focused on relatively specialized solutions which will not grow up to Internet-scale solutions. I firmly believe that the emergence of cloud computing as a buzz word and as a vision is strongly propelled by the predominance and rising eminence of open source software, including Linux and a variety of key open source components. However, until there is a world-class open source project for generalized, complex provisioning, clouds will evolve from within enterprises and from within a set of localized utility computing providers using only proprietary provisioning technologies.

Today, 3tera's AppLogic or IBM's TPM provide some highly evolved mechanisms for provisioning and deployment. However, as licensed applications, they will be focused on deployments within enterprises and larger

[24] US Defense Advanced Research Projects agency, `http://www.darpa.mil/`

[25] `http://en.wikipedia.org/wiki/UUCP`

[26] `http://en.wikipedia.org/wiki/BITNET`

[27] `http://en.wikipedia.org/wiki/ARPANET`

[28] `http://www.nsf.gov/about/history/nsf0050/internet/launch.htm`

data centers. These and similar products will aid in the evolution of clouds within the data center over the next several years, but the evolution of a project like SystemImager,[29] the new xCat[30] or improvements to RPM (the RPM Package Manager) or APT (the Advanced Packaging Tool) may help increase the ability to provision virtual machines or deploy virtual appliances. However, provisioning includes the ability to configure networking, perhaps as VLANs or VPNs, and includes the set up and creation of storage. And, being able to deploy those for any possible user-defined workload is still too complex for wide-spread adoption. And, generally, they don't deal with the security impliciations for proper isolation of workloads that are needed to make sure that I, as a user, am unable to access your proprietary data or applications.

4.5.3 Security Challenges

As alluded to above, security is another of the major inhibitors to true cloud computing today. While some workloads encourage sharing of data, such as Wikis, the actual installation and management of the software application typically needs to be restricted to the administrator of that software. But with utility providers' compute resources being completely accessed through the Internet, any applications typically have Internet access as well. Setting up and configuring VPNs or VLANs requires custom administration by the creator of the virtual appliance or via custom systems management by the person deploying the virtual appliance. Today, enterprises and even small business owners employ as much physical security and firewalling technology as possible to protect their business from crackers.[31] Setting up equivalent security at a remote internet site is not a well-practiced art today. It may be a several years until best practices emerge and those practices have withstood the test of time.

In addition to the networking component of security, an earlier postulate was that virtual appliances would be instantiated on hypervisors or virtualized platforms. However, today the leading hypervisors do not approach the level of security and isolation between guests that is present in physical hardware isolation. While there is work going on in both VMware and Xen, for instance, to improve the security isolation between guests, the ability for a guest to "escape" to the hypervisor and from there have access to other guests is a concern that is likely to prevent competitors from sharing the same physical hardware. That level of isolation will increase over the next few years, making increasing levels of multi-tenency—the ability for diverse guests to share the same hardware—more secure over time.

4.5.4 Other Challenges

While there are a number of other challenges to address, such as how to handle licensing and cost recovery in a virtualized environment, or improving the management of virtual appliances, or making the virtualized environment more dyanmic through the capability of live guest migration, those issues are likely to get worked out as utility computing becomes more common, as organizations built their own internal clouds, and virtual appliances become more common.

5 Conclusions and Outlook

Cloud computing clearly has a lot of hype now, and the vision as represented here can be very compelling. The ubiquitous access to computing resources, much as the Internet has provided us with ubiquitous connectivity, is a powerful vision for our future. Clearly, most of the technology to achieve this vision exists in some form today, which is what makes the vision so catchy in the press today. However, as shown here, we can implement subsets of cloud computing today and we can begin to migrate data centers towards a model which will allow the free flowing access of compute resources within a cloud in the next few years. In the meantime, there are a number of areas that need some additional focus to make this vision a reality.

Those challenges include dramatic simplification in the management of physical compute resources, improvements in virtualization and the management of virtual environments, the creation of a pervasive and accessible set of tools to deploy virtual appliances and workloads within a cloud of virtualized resources, improvements in provisioning of networks and storage, and continued work on improving the security provided by hypervisors

[29] http://wiki.systemimager.org/index.php/Main_Page

[30] http://www.xcat.org/

[31] http://en.wikipedia.org/wiki/Hacker_(computer_security)

at a minimum. As those are evolving, there is the ability to improve workload management, energy management and availability management. Improvements in all of these areas should also improve the efficiency and utilization of computers resources. Resources should ultimately be more effectively shared, and additional resources would be available on demand for those workloads which dynamically need access to more hardware than they they would otherwise have at hand.

The era of cloud computing is just beginning.

Legal Statement