

Cloud computing with Linux

Cloud computing platforms and applications

Skill Level: Introductory

M. Tim Jones (mtj@mtjones.com)

Consultant Engineer
Emulex Corp.

10 Sep 2008

Updated 10 Feb 2009

Cloud computing and storage convert physical resources (like processors and storage) into scalable and shareable resources over the Internet (computing and storage "as a service"). Although not a new concept, virtualization makes this much more scalable and efficient through the sharing of physical systems through server virtualization. Cloud computing gives users access to massive computing and storage resources without their having to know where those resources are or how they're configured. As you might expect, Linux® plays a huge role. Discover cloud computing, and learn why there's a penguin behind that silver lining. [*And see the new [Resources](#) links to the latest developerWorks content on cloud computing.* -Ed]

You can't read a technical Web site these days without some mention of so-called **cloud computing**. Cloud computing is really nothing more than the provisioning of computing resources (computers and storage) as a service. Along with that comes the ability to dynamically scale the service to additional computers and storage in a simple and transparent way. All this is similar to the ideas behind *utility computing*, in which computing resources were viewed as a metered service, as is the case for more traditional utilities (such as electricity or water). What's different is not the goal behind these ideas but the existing technologies that have come together to make them a reality.

One of the most important ideas behind cloud computing is scalability, and the key technology that makes that possible is virtualization. Virtualization allows better use

of a server by aggregating multiple operating systems and applications on a single shared computer. Virtualization also permits online migration so that if a server becomes overloaded, an instance of an operating system (and its applications) can be migrated to a new, less cluttered server.

Read more by Tim on developerWorks

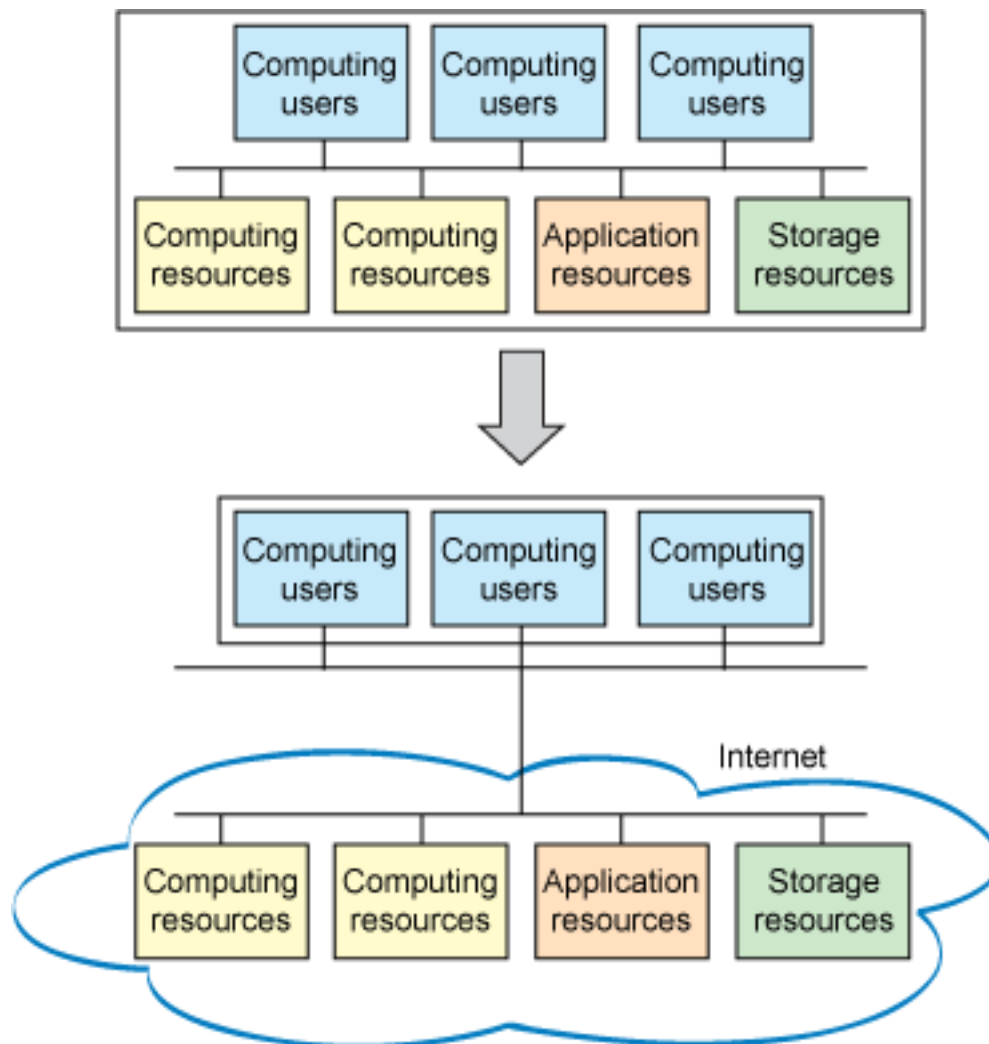
- [Tim's *Anatomy of...* articles](#)
- [All of Tim's articles on developerWorks](#)

From an external view, cloud computing is simply the migration of computing and storage outside an enterprise and into the cloud. The user defines the resource requirements (such as computing and wide area network, or WAN, bandwidth needs), and the cloud provider virtually assembles these components within its infrastructure, as shown in Figure 1.

Perspectives on cloud computing

Cloud computing implements the idea of utility computing, which was first suggested by John McCarthy in 1961, where computing is viewed as a public utility. Cloud computing can also be compared to *cluster computing*, which views a group of linked computers as a single virtual computer for high-performance computing (HPC), or *grid computing*, where the linked computers tend to be geographically distributed to solve a common problem. Time-sharing systems were offered in the 1960s IBM, General Electric, and other companies.

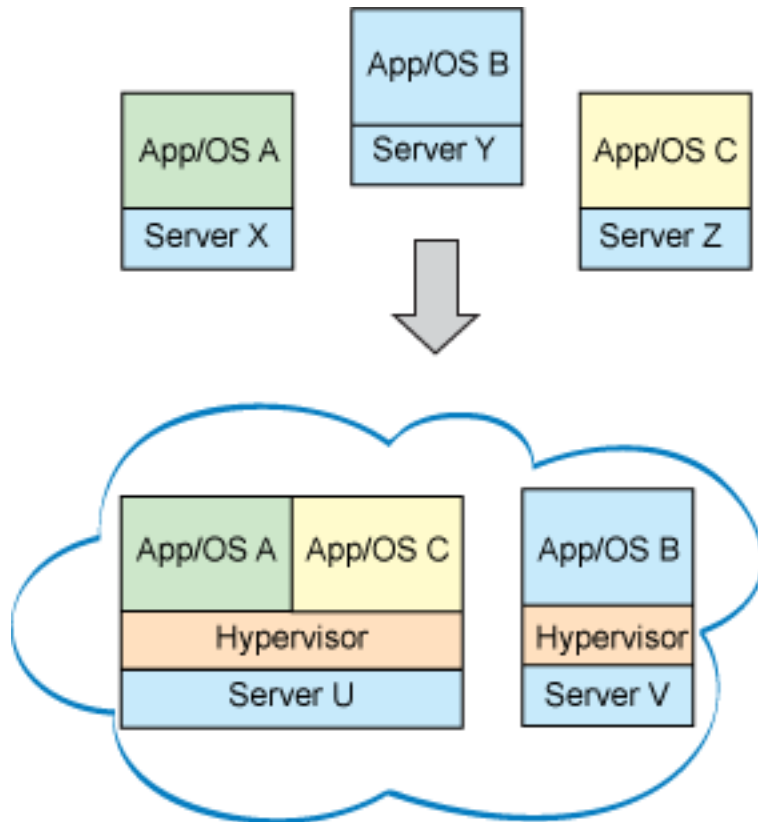
Figure 1. Cloud computing migrates resources within the Internet



But why would you willingly relinquish control over your resources and allow them to virtually exist in the cloud? There are many reasons, but two that I believe are most important are cost and scalability. The goal of cloud computing is to make these resources less expensive than what you can provide for and manage yourself. Along with this reduction in cost comes greater flexibility and scaling. A cloud computing provider can easily scale your virtual environment for greater bandwidth or computing resources with the provider's virtual infrastructure.

The green advantage to cloud computing is the ability to virtualize and share resources among different applications for better server utilization. Figure 2 shows an example. Here, three independent platforms existed for different applications, each running on its own server. In the cloud, servers can be shared (virtualized) for operating systems and applications to better use the servers, resulting in fewer servers. Fewer servers means less required space (minimizing the data center footprint) and less power for cooling (minimizing the carbon footprint).

Figure 2. Virtualization and resource use

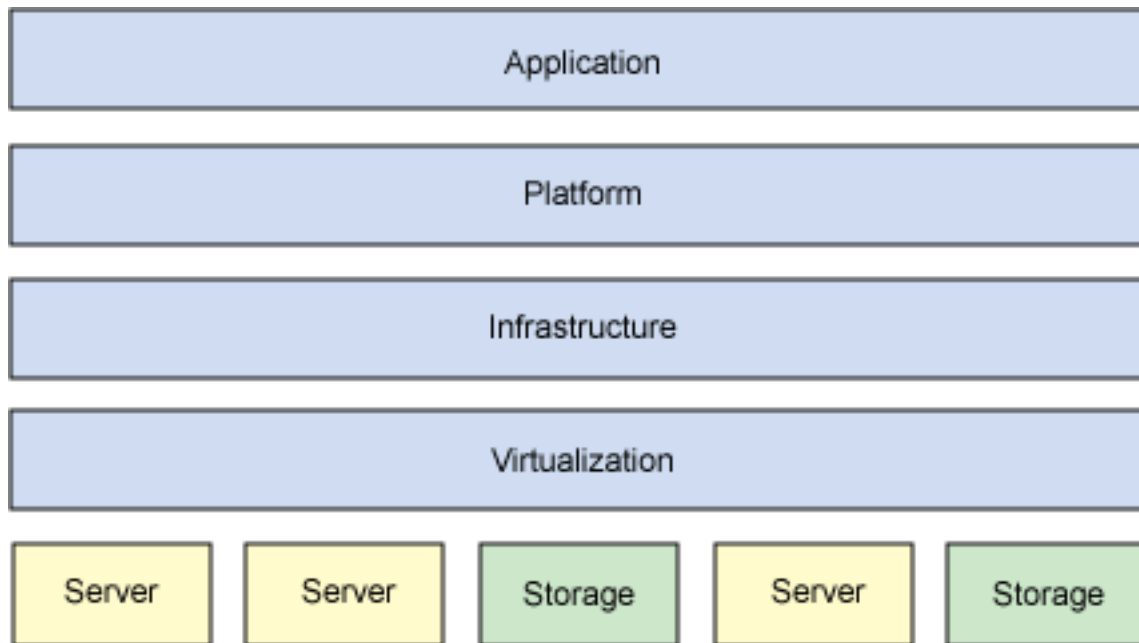


But there are trade-offs, and cloud computing is not without its warts. This article explores some of these issues later. But now, let's dig deeper into cloud computing to explore what it's all about.

Anatomy of cloud computing

As you peer inside the cloud, you find that it's actually not just a single service but a collection of services, as shown in Figure 3. These layers define the level of service provided.

Figure 3. The layers of cloud computing



Let's start at the lowest level of service provided, which is the infrastructure (*Infrastructure-as-a-Service*, or IaaS). IaaS is the leasing of infrastructure (computing resources and storage) as a service. This means not only virtualized computers with guaranteed processing power but reserved bandwidth for storage and Internet access. In essence, it's the capability of leasing a computer or data center with specific quality-of-service constraints that has the ability to execute an arbitrary operating system and software.

The value of cloud computing

Besides reducing the management cost associated with cloud computing resources, there are other advantages. For example, when you separate yourself from your resources by the Internet, it doesn't really matter where those resources reside. They could be, for example, in a climate that offers ambient (natural) cooling and therefore minimizes energy usage.

Moving up the stack, the next level of service is the platform (*Platform-as-a-Service*, or PaaS). PaaS is similar to IaaS but includes operating systems and required services that focus on a particular application. For example, a PaaS in addition to virtualized servers and storage provides a particular operating system and application set (typically, as a virtual machine, or VM, file, such as VMware's .vmdk format) along with access to necessary services such as a MySQL database or other, specialized local resources. In other words, PaaS is IaaS with a custom software stack for the given application.

Finally, at the top of [Figure 3](#) is the simplest service that can be provided: the application. This layer is called *Software-as-a-Service* (SaaS), and it is the model of deploying software from a centralized system to run on a local computer (or remotely

from the cloud). As a metered service, SaaS allows you to lease an application and pay only for the time used.

That's the 30,000-foot view of cloud computing. This view ignores some of the other aspects of the cloud, such as *data-Storage-as-a-Service* (dSaaS), which provides storage as a metered service in which the consumer is billed based on used capacity (the amount of storage used) and utilization (bandwidth requirements for the storage). Cloud services have also emerged, which provide internal mechanisms for interoperability as well as external application program interfaces (APIs), such as Web services.

The cloud computing landscape

In recent months, there's been an explosion of investment into cloud computing and related infrastructure. This massive investment indicates that there is demand for virtualization of resources inside the cloud. The past year has seen many new services, some of which are shown in Figure 4.

Figure 4. Cloud computing layers with offerings

SaaS	Software-as-a-Service	Google Apps, Microsoft "Software+Services"
PaaS	Platform-as-a-Service	IBM IT Factory, Google AppEngine, Force.com
IaaS	Infrastructure-as-a-Service	Amazon EC2, IBM Blue Cloud, Sun Grid
dSaaS	data-Storage-as-a-Service	Nirvanix SDN, Amazon S3, Cleversafe dsNet

This is by no means an exhaustive list of offerings, as it changes quite frequently. However, it does provide an overview of some of the offerings and how they are differentiated. Links to some of the offerings are included in [Resources](#) later in this article.

Linux and open source in the cloud

Let's now explore how Linux and the open source community contribute to the world of cloud computing. As you might have guessed, Linux and open source technologies play a huge role.

Software-as-a-Service

SaaS is the ability to access software over the Internet as a service. An early

approach to SaaS was the Application Service Provider (ASP). ASPs provide subscriptions to software that is hosted or delivered over the Internet. The ASP delivers the software and charges fees based on its use. In this way, you don't purchase the software but simply lease it on an as-needed basis.

Example SaaS

An interesting example of traditional versus SaaS applications is the application life cycle management tool from SoftwarePlanner.com. This company offers their tool using the traditional model, where customers host the application suite within their enterprise, or as SaaS, where customers host the application suite and make it available over the Internet.

Another perspective on SaaS is the use of software over the Internet that executes remotely. This software can be in the form of services used by a local application (defined as *Web services*) or a remote application observed through a Web browser. One example of a remote application service is Google Apps, which provides several enterprise applications through a standard Web browser. Remotely executing applications commonly rely on an application server to expose needed services. An *application server* is a software framework that exposes APIs for software services (such as transaction management or database access). Examples include Red Hat JBoss Application Server, Apache Geronimo, and IBM® WebSphere® Application Server. Many other application servers exist, and an extensive list is included in [Resources](#).

Another recent example of SaaS is Google's Chrome browser. The browser is an ideal environment as a new desktop through which applications can be delivered (either locally or remotely) in addition to the traditional Web browsing experience. (For more information, see [Resources](#).)

Platform-as-a-Service

PaaS can be described as an entire virtualized platform that includes one or more servers (virtualized over the set of physical servers), operating systems, and specific applications (such as Apache and MySQL for Web-based applications). In some cases, these platforms can be predefined and selected; in others, you can provide a VM image that contains all the necessary user-specific applications.

One interesting example of a PaaS is Google App Engine. App Engine is a service that allows you to deploy your Web applications on Google's very scalable architecture. App Engine provides you with a sandbox for your Python application that can be referenced over the Internet (and additional languages will be supported in the future). App Engine provides Python APIs for persistently storing and managing data (using the Google Query Language, or GQL) in addition to support for authenticating users, manipulating images, and sending e-mail. The sandbox in which the Web application runs restricts access to the underlying operating system.

Although App Engine limits the functionality available to your application, it supports the construction of useful Web services. Check out [Resources](#) for more information.

Note: Deploying applications in App Engine is free within certain bandwidth and storage constraints. To build production Web sites with App Engine, usage fees are assessed.

Another example of a PaaS is 10gen, which is both a cloud platform and a downloadable open source package for creating your own private cloud. A software stack similar to App Engine, 10gen provides similar functionality to App Engine—with certain differences. With 10gen, you can develop applications in Python as well as the JavaScript and Ruby programming languages. The platform also uses the sandbox concept to isolate applications and provide a reliable environment over a large number of computers (built, of course, on Linux) using their own application server.

Infrastructure-as-a-Service

IaaS is the delivery of computer infrastructure as a service. This layer differs from PaaS in that the virtual hardware is provided without a software stack. Instead, the consumer provides a VM image that is invoked on one or more virtualized servers. IaaS is the rawest form of computing as a service (outside of access to the physical infrastructure). The most well-known commercial IaaS provider is Amazon Elastic Compute Cloud (EC2). In EC2, you can specify a particular VM (operating system and application set), and then deploy your applications on it or provide your own VM image to execute on the servers. You're then billed simply for compute time, storage, and network bandwidth.

The Eucalyptus project (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) is an open source implementation of Amazon EC2 that is interface-compatible with the commercial service. Like EC2, Eucalyptus relies on Linux with Xen for operating system virtualization. Eucalyptus was developed at the University of California, Santa Barbara, for the purpose of cloud computing research. You can download it from the university's Web site (see [Resources](#)), or you can experiment with it via the Eucalyptus Public Cloud with certain restrictions.

Another EC2 style of IaaS is the Enomalism cloud computing platform. Enomalism is an open source project that provides a cloud computing framework with functionality similar to EC2. Enomalism is based on Linux, with support for both Xen and the Kernel Virtual Machine (KVM). But unlike other pure IaaS solutions, Enomalism provides a software stack based on the TurboGears Web application framework and Python.

Other cloud developments

In addition to the developments already discussed, several other Linux-based open source packages are useful in cloud environments. Hadoop is an open source Java™ software framework similar to PaaS but focused on manipulating large data sets over a set of networked servers (inspired by Google MapReduce, which enables parallel processing of large data sets). As such, it finds use in Web search and advertising applications—in particular, at Yahoo! Hadoop also provides several sub-projects, mimicking Google applications. For example, HBase provides Google BigTable database-like functionality, and the Hadoop Distributed File System (HDFS) provides similar functionality to Google File System (GFS).

Issues and challenges

The issues of cloud computing are clear—with privacy and security being two of the most important. Privacy can be combated with encryption, but due diligence is required when selecting a cloud computing service. Even e-Commerce was viewed in a skeptical light when the Web started to grow. Worldwide, trillions of dollars-worth of e-Commerce transactions occur annually, so cloud computing will benefit from all the technologies (such as Secure Sockets Layer, or SSL) that make the Web safe today.

Going further

The cloud computing rush has just begun, and so has the open source development on Linux that will drive it. Given the massive investment being made in cloud computing, it's clear that a shift is occurring back to centralized data centers. It will be interesting to see the new technologies and architectures that are around the corner.

Resources

Learn

- Learn more about [Cloud computing with Amazon Web Services](#) in a five-part series (developerWorks, July-February 2008).
- "[Automating Linux cloud installations](#)" (developerWorks, November 2008) shows how to automate the installation of SUSE Linux on a Power system or System p LPAR.
- Read more about [IBM's "Blue Cloud" centers](#) now opening around the world. In June 2008, IBM opened two new centers in Beijing, China, and Johannesburg, South Africa. IBM will open at least 13 cloud computing data centers to support data protection through disparate geographies.
- Wikipedia provides a great set of introductions to [cloud computing](#) and its related technologies:
 - [IaaS](#)
 - [PaaS](#)
 - [SaaS](#)
 - [Utility computing](#)
 - [Grid computing](#)
- In "[Virtual Linux: An overview of virtualization methods, architectures and implementations](#)" (developerWorks, December 2006), learn more about the various types of virtualization. Cloud computing relies on virtualization for optimal use of server-available resources. With virtualization, servers can be used to host multiple operating systems and application sets.
- See a detailed discussion of [Web services architectures and services](#) by the World Wide Web Consortium (W3C).
- Wikipedia gives a [great comparison of application servers](#) that includes both open source and proprietary solutions. You'll find standard Java 2 Platform, Enterprise Edition, application servers and even functional programming-based application servers such as the Haskell-oriented HApps.
- Michael Sheehan looks at the buzz factor of [cloud computing compared to grid computing](#) using [Google trends](#).
- In the [developerWorks Linux zone](#), find more resources for Linux developers, and scan our [most popular articles and tutorials](#).
- See all [Linux tips](#) and [Linux tutorials](#) on developerWorks.

- Stay current with [developerWorks technical events and Webcasts](#).

Get products and technologies

- [JBoss Application Server](#), [Geronimo](#), and [WebSphere Application Server](#) are some of the most popular application servers providing a local and Web services model for SaaS.
- Get an intro to [Google App Engine](#) and [10gen software stack](#). They exemplify PaaS solutions, which provide a virtualized operating system with software stack for the user application.
- The most notable IaaS solution is [Amazon EC2](#), but open source solutions, such as [Eucalyptus](#) and [Enomalism](#), also exist. IaaS provides a virtualized hardware infrastructure ready for VM execution.
- [Hadoop](#) is a software stack that allows you to process large amounts of data in a scalable and efficient way. It provides a programming-based platform along with distributed file system and applications.
- With [IBM trial software](#), available for download directly from developerWorks, build your next development project on Linux.

Discuss

- Get involved in the [developerWorks community](#) through blogs, forums, podcasts, and spaces.

About the author

M. Tim Jones

M. Tim Jones is an embedded firmware architect and the author of *Artificial Intelligence: A Systems Approach*, *GNU/Linux Application Programming* (now in its second edition), *AI Application Programming* (in its second edition), and *BSD Sockets Programming from a Multilanguage Perspective*. His engineering background ranges from the development of kernels for geosynchronous spacecraft to embedded systems architecture and networking protocols development. Tim is a Consultant Engineer for Emulex Corp. in Longmont, Colorado.

Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other

IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. See the current list of [IBM trademarks](#).

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.