Anatomy of an open source cloud

Building blocks for Infrastructure as a Service

M. Tim Jones
Independent author

05 June 2012
(First published 09 March 2010)

Cloud computing is no longer a technology on the cusp of breaking out but a valuable and important technology that is fundamentally changing the way we use and develop on-demand applications. As you would expect, Linux® and open source provide the foundation for the cloud (for public and private infrastructures). Explore the anatomy of the cloud, its architecture, and the open source technologies used to build these dynamic and scalable computing and storage platforms.
05 Jun 2012 - *The author added basic updates to the entire article, including information about OpenStack, nested virtualization and its importance for the cloud), plus three related Resources.*

Develop skills on this topic

This content is part of a progressive knowledge path for advancing your skills. See Cloud computing: Introduction to Infrastructure as a Service

The use of the cloud as an abstraction is quite common for the distributed system that is the Internet, but the past few years have seen this abstraction expanded to incorporate highly virtualized and scalable infrastructures that are easily provisioned as a service (locally, remotely, or a combination of local and remote resources). This article forgoes an in-depth definition of cloud architecture and its benefits, and instead refers to worthwhile reading in the Resources section.

Cloud computing anatomy

This article begins with an exploration of the core abstractions of cloud architectures (from Infrastructure as a Service [IaaS]), then moves beyond the building blocks to the more highly integrated solutions.

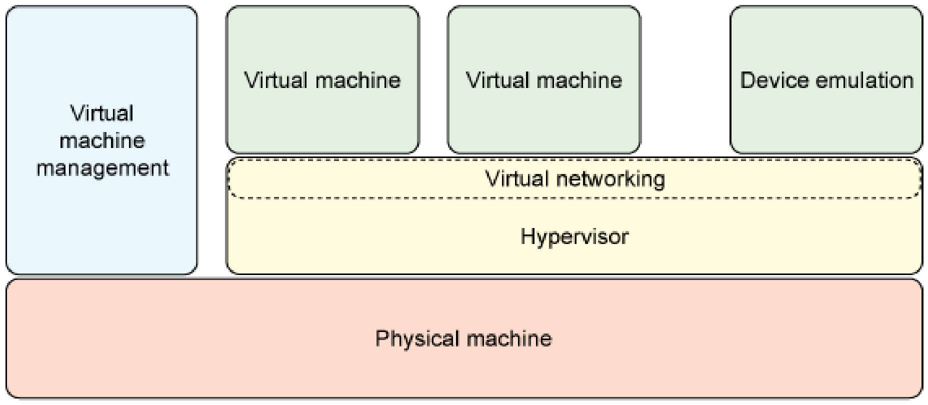
Frequently used acronyms

- **API:** Application programming interface
- **I/O:** Input/output

- **SLA:** Service-level agreement
- **UI:** User interface

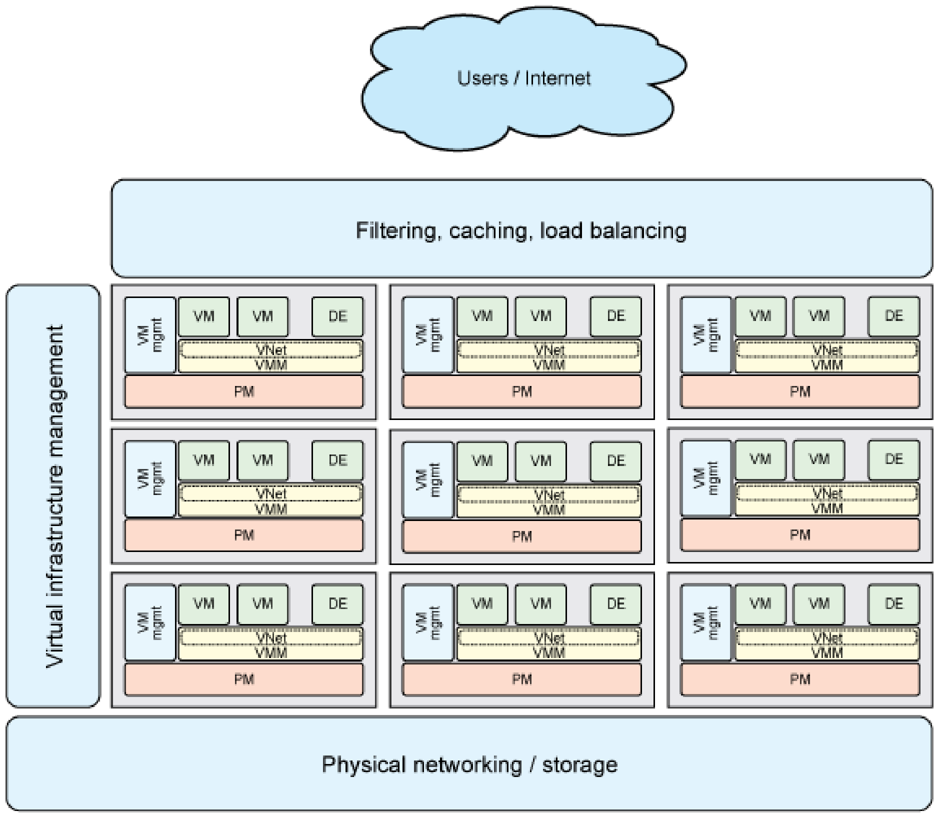
Although not a requirement, virtualization provides unique benefits for building dynamically scalable architectures. In addition to resource sharing and scalability, virtualization introduces the ability to migrate virtual machines (VMs) between physical servers for the purposes of load balancing. Figure 1 shows that the virtualization component is provided by a layer of software called a *hypervisor* (sometimes called a *virtual machine monitor* [VMM]). This layer provides the ability to execute multiple operating systems (and their applications) simultaneously on a single physical machine. Each operating system views a logical machine that is mapped by the hypervisor to the physical machine. On the hypervisor is an object called a *virtual machine* that encapsulates the operating system, applications, and configuration. Optionally, device emulation can be provided in the hypervisor or as a VM. Finally, given the new dynamic nature of virtualization and the new capabilities it provides, new management schemes are needed. This management is best done in layers, considering local management at the server, as well as higher-level infrastructure management, providing the overall orchestration of the virtual environment.

Figure 1. Core elements of a node in the cloud



If you take those nodes from Figure 1 and multiply them on a physical network with shared storage, orchestrating management over the entire infrastructure, then provide front-end load balancing of incoming connections (whether in a private or a public setting) with caching and filtering, you have a virtual infrastructure called a *cloud.* This new construction is shown in Figure 2. Dormant servers can be powered down until needed for additional compute capacity (providing better power efficiency), with VMs balanced (even dynamically) across the nodes depending upon their collective loads.

Figure 2. Cloud computing infrastructure


Users / Internet
Filtering, caching, load balancing
Virtual infrastructure management
VM mgmt
VM
VM
DE
VNet
VMM
PM
Physical networking / storage

With the basic architecture of a cloud defined, let's now explore where open source is being applied to build out a dynamic cloud infrastructure.

Core open source technologies

The Linux landscape is seeing a wave of development focused on virtualized infrastructures for virtualization, management, and larger-scale integration of cloud software packages. Let's start with a view of open source at the individual node level, then step up to the infrastructure to see what's happening there.

Hypervisors

The base of the cloud at the node level is the hypervisor. Although virtualization is not a requirement, it provides undisputed capabilities for scalable and power-efficient architectures. There exist a number of open source virtualization solutions, but two key solutions are those that transform the Linux operating system into a hypervisor: the Linux Kernel Virtual Machine (KVM) and Lguest. KVM is the official hypervisor solution, being deployed into production environments. Lguest is a Linux-focused solution that runs only Linux VMs, but is integrated into the kernel and finding wider use. The Xen hypervisor is also widely used within public and private IaaS solutions due to its performance advantages.

Outside of converting Linux to a hypervisor, there are other solutions that take a guest VM-focused approach. User-Mode Linux (UML) is another approach that modifies a guest Linux kernel to run on top of another Linux operating system (without hypervisor extensions). Because many users want to run an unmodified kernel, full virtualization solutions (such as KVM) are preferred.

This UML approach is also popular, but requires virtualized hardware (such as a console, virtual disk, and networking).

Device emulation

The hypervisor provides the means to share the CPU with multiple operating systems (CPU virtualization), but to provide full virtualization, the entire environment must be virtualized for the VMs. Machine—or platform—emulation can be performed in a number of ways, but a popular open source package that supports a number of hypervisors is called *QEMU.* QEMU is a complete emulator and hypervisor. But KVM makes use of QEMU for device emulation as a separate process in the user space (see Figure 1). One interesting feature of QEMU is that because it provides disk emulation (through the QCOW format), QEMU provides other advanced features such as snapshots and live VM migration.

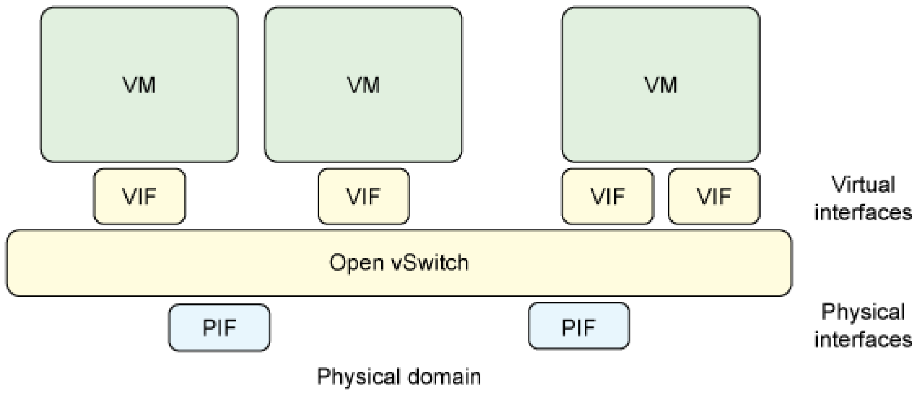
KVM, since kernel 2.6.25, uses virtio as a means of optimizing I/O virtualization performance. It does this by introducing paravirtualized drivers into the hypervisor with hooks from the guest to bring performance to near-native levels. This works only when the operating system can be modified for this purpose, but finds use in Linux guest on Linux hypervisor scenarios.

Today, virtio and QEMU work together so emulated device transactions can be optimized between the Linux guest and QEMU emulator in the user space.

Virtual networking

As VMs consolidate onto physical servers, the networking needs of the platform intensify. But rather than force all of the VM's networking to the physical layer of the platform, local communication could instead be virtualized itself. To optimize network communication among VMs, there is the introduction of the *virtual switch.* The vSwitch behaves like a physical switch, but is virtualized into the platform (see Figure 3). In this figure, virtualized interfaces (VIFs) associated with the VMs communicate through the virtual switch to the physical interfaces (PIFs).

Figure 3. High-level view of Open vSwitch with virtual and physical interfaces



Open source is addressing this problem as well, with one very interesting solution called the *Open vSwitch.* In addition to providing a virtual switch for virtual environments, the vSwitch can also integrate across physical platforms and provide enterprise-level features like virtual local area networks (VLANs), priority-based Quality of Service (QoS), trunking, and support for hardware acceleration (such as single-root I/O virtualization [IOV] network adapters). The Open vSwitch is currently available for 2.6.15 kernels and supports the range of Linux-based virtualization solutions (Xen, KVM, VirtualBox) and management standards (Remote Switched Port Analyzer [RSPAN], NetFlow, etc.).

VM tools and technologies

As VMs are an aggregation of operating system, root file system, and configuration, the space is ripe for tool development. But to realize the full potential of VMs and tools, there must be a portable way to assemble them. The current approach, called the *Open Virtualization Format* (OVF) is a VM construction that is flexible, efficient, and portable. OVF wraps a virtual disk image in an XML wrapper that defines the configuration of the VM, including networking configuration, processor and memory requirements, and a variety of extensible metadata to further define the image and its platform needs. The key capability provided by OVF is the portability to distribute VMs in a hypervisor-agnostic manner.

A number of utilities exist to manage VM images (VMIs) as well as convert them to and from other formats. The `ovftool` from VMware is a useful tool that you can use for VMI conversion (for example, to convert from the VMware Virtual Disk Development Kit [VMDK] format into OVF). This tool and others are useful once you have a VMI, but what if you have a physical server you'd like to convert into a VMI? You can employ a useful tool called *Clonezilla* for this purpose. Although it was originally developed as a disk-cloning tool for disaster recovery, you can use it to convert a physical server instance into a VM for easy deployment into a virtualized infrastructure. Numerous other tools exist (such as utilities built upon libvirt) or are in development for conversion and management as the OVF format gains adoption.

Local management

This article explores management from two perspectives. This section discusses platform management; a later section expands to infrastructure management at the higher level.

Red Hat introduced the libvirt library as an API for managing platform virtualization (hypervisors and VMs). What makes libvirt interesting is that it supports a number of hypervisor solutions (KVM and Xen being two) and provides API bindings for a number of languages (such as C, Python, and Ruby). It provides the "last mile" of management, interfacing directly with the platform hypervisor and extending APIs out to larger infrastructure-management solutions. With libvirt, it's simple to start and stop VMs, and it provides APIs for more advanced operations, such as migration of VMs between platforms. Using libvirt, it's also possible to use its shell (built on top of libvirt), called `virsh`.

Infrastructure open source technologies

Now that you've seen some of the open source solutions at the virtualized node level, look at some other open source applications that support this infrastructure. This article explores three categories. The first two are infrastructure-level technologies that complement the solutions previously discussed. The third category consists of integrated solutions that bring all of the pieces together for simpler deployment.

I/O technologies

Building a scalable and balanced web architecture depends upon the ability to balance web traffic across the servers that implement the back-end functionality. A number of load-balancing solutions exist, but recently, Yahoo! open sourced a solution called *Traffic Server.* Traffic Server is interesting, because it encapsulates a large number of capabilities in one package for cloud infrastructures, including session management, authentication, filtering, load balancing, and routing. Yahoo! initially acquired this product from Inktomi, but has now extended and introduced the product into open source.

Infrastructure management

Larger-scale infrastructure management (managing many hypervisors and even more VMs) can be accomplished in a number of ways. Two of the more common solutions are each built from the same platform (libvirt). The oVirt package is an open VM management tool that scales from a small number of VMs to thousands of VMs running on hundreds of hosts. The oVirt package, developed by Red Hat, is a web-based management console that, in addition to traditional management, supports the automation of clustering and load balancing. The oVirt tool is written in the Python language. VirtManager, also based on libvirt and developed by Red Hat, is an application with a GTK+ UI (instead of being web-based like oVirt). VirtManager presents a much more graphically rich display (for live performance and resource utilization) and includes a VNC client viewer for a full graphical console to remote VMs.

And Puppet is another open source package designed for data center infrastructure (a cloud). Although not designed solely for virtualized infrastructures, it simplifies the management of large infrastructures by abstracting the details of the peer operating system. It does this through the use of the Puppet language. Puppet is ideal for automating administrative tasks over large numbers of servers and is widely used today.

Integrated IaaS solutions

The following open source packages take a more holistic approach by integrating all of the necessary functionality into a single package (including virtualization, management, interfaces, and security). When added to a network of servers and storage, these packages produce flexible cloud computing and storage infrastructures (IaaS). For details about these platforms, see Resources.

Eucalyptus

One of the most popular open source packages for building cloud computing infrastructures is *Eucalyptus* (for *Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems*). What makes it unique is that its interface is compatible with Amazon Elastic Compute Cloud (Amazon EC2—Amazon's cloud computing interface). Additionally, Eucalyptus includes Walrus, which is a cloud storage application compatible with Amazon Simple Storage Service (Amazon S3—Amazon's cloud storage interface).

Eucalyptus supports KVM/Linux and Xen for hypervisors and includes the Rocks cluster distribution for cluster management.

OpenNebula

OpenNebula is another interesting open source application (under the Apache license) developed at the Universidad Complutense de Madrid. In addition to supporting private cloud construction, OpenNebula supports the idea of hybrid clouds. A hybrid cloud permits combining a private cloud infrastructure with a public cloud infrastructure (such as Amazon) to enable even higher degrees of scaling.

OpenNebula supports Xen, KVM/Linux, and VMware and relies on elements like libvirt for management and introspection.

Nimbus

Nimbus is another IaaS solution focused on scientific computing. With Nimbus, you can lease remote resources (such as those provided by Amazon EC2) and manage them locally (configure, deploy VMs, monitor, etc.). Nimbus morphed from the Workspace Service project (part of Globus.org). Being dependent on Amazon EC2, Nimbus supports Xen and KVM/Linux.

Xen Cloud Platform

Citrix has integrated Xen into an IaaS platform, using Xen as the hypervisor while incorporating other open source capabilities such as the Open vSwitch. An interesting advantage to the Xen solution is the focus on standards-based management (including OVF, Distributed Management Task Force [DTMF], the Common Information Model [CIM], and Virtualization Management

Initiative [VMAN]) from the project Kensho. The Xen management stack supports SLA guarantees, along with detailed metrics for charge-back.

OpenQRM

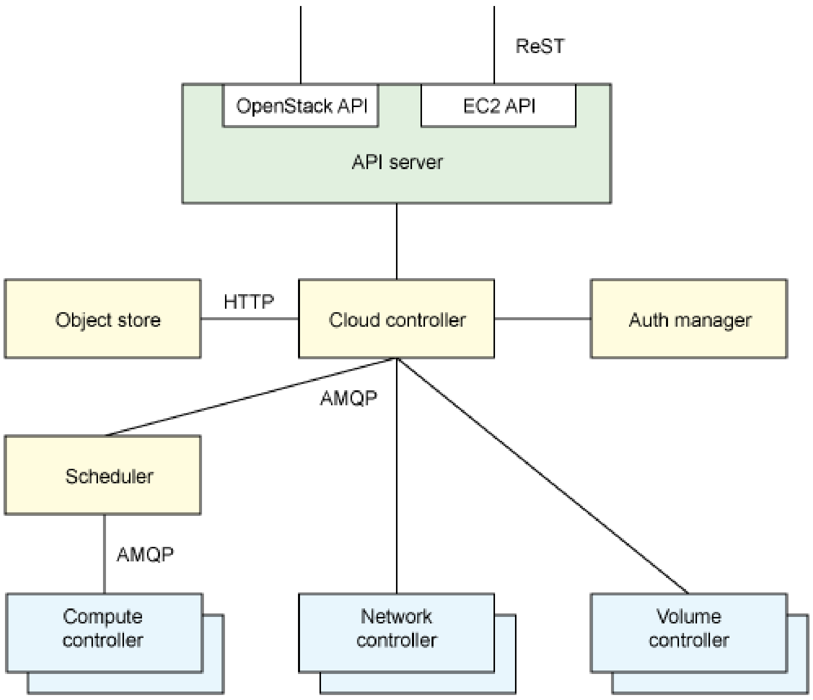
Our penultimate solution is OpenQRM, which is categorized as a data center management platform. OpenQRM provides a single console to manage an entire virtualized data center that is architecturally pluggable to permit integration of third-party tools. OpenQRM integrates support for high availability (through redundancy) and supports a variety of hypervisors, including KVM/Linux, Xen, VMware, and Linux VServer.

OpenStack

Today, the leading IaaS solution is called OpenStack. OpenStack was released in July 2010, and has quickly become the standard open-source IaaS solution. OpenStack is a combination of two cloud initiatives from RackSpace Hosting (Cloud Files) and NASA's Nebula platform. OpenStack is being developed in the Python language, and is under active development under the Apache license.

Figure 4 shows OpenStack as a modular architecture for managing a set of servers for compute and storage. OpenStack is defined by three major components, consisting of Nova (representing the compute side), Swift (representing object storage), and Glance (implementing the image service). OpenStack supports a wide range of hypervisors, including KVM, Linux Containers (LXC), QEMU, UML, Xen, and XenServer. OpenStack is now driving open standards for cloud infrastructures, and as a result is growing rapidly in use.

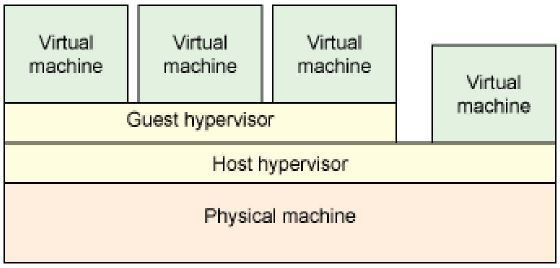
Figure 4. The OpenStack modular architecture



Nested hypervisors

One of the most interesting advancements occurring within virtualization and open source is the ability to host a hypervisor on top of another hypervisor (using a concept called *nesting*). Recall from Figure 1 that the typical usage model is the hypervisor as an abstraction of the physical platform to guest virtual machines. With nesting, the *host* hypervisor provides the base level abstraction for either a virtual machine (as the typical case) in addition to a *guest* hypervisor (see Figure 4).

Figure 5. Nesting hypervisors as a useful abstraction



While nesting might seem to be a odd usage model, consider its application to cloud computing. Clouds commonly require virtual machines to conform to their hypervisor of choice. But if that hypervisor supported nesting, then the user could provide not only their virtual machines, but also the guest hypervisor of their choice. This also provides an extra abstraction for the user, adding portability to clouds (the ability to easily switch between cloud providers).

You can find support for nested virtualization today in the KVM hypervisor. Check out Resources to find out how to demonstrate this on an AMD server.

Going further

Volumes could be written about the leadership role that open source is playing in the cloud and virtualization domain, but this article provided a short introduction to some of the popular and visible solutions available today. Whether you're looking to build a cloud based on your own requirements from individual pieces or simply want a cohesive solution that works out of the box, open source has you covered. See Resources for more details on the solutions discussed here and others.

Resources

Learn

- OpenStack is the latest IaaS solution, released into open source from NASA and Rackspace. OpenStack implements a cloud operating system for compute and storage, as well as providing an image service for virtual machines.
- "The Turtles Project: Design and Implementation of Nested Virtualization," a paper authored by IBM Research Haifa and IBM Linux Technology Center, details the Turtles project for nested virtualization and introduces the idea of nesting a guest hypervisor on a host hypervisor to create another useful level of abstraction.
- Nested Virtualization with KVM and AMD provides a useful demonstration of nesting using KVM and an AMD server. It's certainly work in progress, but a useful technology that's advancing.
- Michael Galpin provides a great introduction to cloud computing, its benefits and challenges, and a review of some of the basic and specialized platforms available in "Realities of open source cloud computing, Part 1: Not all clouds are equal."
- Learn more about Linux as a hypervisor in "Anatomy of a Linux hypervisor" and learn more about virtual appliances and the use of OVF in "Virtual appliances and the Open Virtualization Format."
- Kernel Virtual Machine was the first Linux-based hypervisor; read more about its architecture in "Discover the Linux Kernel Virtual Machine." Lguest entered the kernel shortly thereafter and took a more lightweight approach (5,000 additional lines of source code). User-mode Linux was discussed as another alternative, which instead of a hypervisor is a modified guest running on a standard kernel. And Xen is a popular hypervisor used in some of the largest cloud infrastructures (Amazon's Elastic Compute Cloud, for example); it supports full virtualization and para-virtualization.
- Read more about QEMU in "System emulation with QEMU." Additional resources for device emulation and virtual I/O include "Linux virtualization and PCI passthrough" and "Virtio: An I/O virtualization framework for Linux."
- The Open vSwitch can be used by all of the major hypervisors (KVM, Xen, VirtualBox); learn more about its design and motivation in the Hot Topics in Networks paper "Extending Networking into the Virtualization Layer (PDF)."
- Yahoo! open sourced its Traffic Server software under the Apache Software Foundation. Yahoo! uses this software internally, and it provides the front end for cloud computing infrastructures. Traffic Server aggregates load balancing, session management, authentication, configuration management, and routing for large-scale Web 2.0 infrastructures. Learn more in the company's announcement and the Traffic Server wiki.
- For detailed instructions on the creation of a KVM-based VM, check out "Create a KVM-based virtual server."
- VM tools are growing, but a good start is available today. "Migrate to a virtual Linux environment with Clonezilla" explores Clonezilla, which can be used to build a VM from a physical instance. There's also ovftool from VMware, which is useful for VM conversion.
- This article explored a few different virtual infrastructure management applications, two of which were built on the libvirt virtualization API. Read more about libvirt in "Anatomy of the

libvirt virtualization library." To learn more about the higher-level management solutions, check out oVirt, Virtual Machine Manager (virt-manager), and Puppet.
- To listen to interesting interviews and discussions for software developers, check out developerWorks podcasts.
- Stay current with developerWorks' Technical events and webcasts.
- Follow developerWorks on Twitter.
- Check out upcoming conferences, trade shows, webcasts, and other Events around the world that are of interest to IBM open source developers.
- Visit the developerWorks Open source zone for extensive how-to information, tools, and project updates to help you develop with open source technologies and use them with IBM's products, as well as our most popular articles and tutorials.
- The My developerWorks community is an example of a successful general community that covers a wide variety of topics.
- Watch and learn about IBM and open source technologies and product functions with the no-cost developerWorks On demand demos.

Get products and technologies

- The development of open source IaaS is exploding today with a large number of very useful collections. Examples include Eucalyptus, OpenNebula, Nimbus, Xen Cloud Platform (with Project Kensho), and openQRM.
- Innovate your next open source development project with IBM trial software, available for download or on DVD.
- Download IBM product evaluation versions or explore the online trials in the IBM SOA Sandbox and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

Discuss

- Participate in developerWorks blogs and get involved in the developerWorks community.

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,AI Application Programming,* and *BSD Sockets Programming from a Multi-language Perspective.* His engineering background ranges from the development of kernels for geosynchronous spacecraft to embedded systems architecture and networking protocols development. He is a senior architect for Emulex Corp. in Longmont, Colo.