Chapter 3

Building Cloud Networks

3.1 Chapter Overview

In previous chapters we have explained what cloud computing is. In this chapter, we will describe what it takes to build a cloud network. You will learn how and why companies build these highly automated private cloud networks providing resources that can be managed from a single point. We will discuss the significant reliance of cloud computing architectures on server and storage virtualization as a layer between applications and distributed computing resources. You will learn the basics of how flexible cloud computing networks such as those modeled after public providers such as Google and Amazon are built, and how they interconnect with corporate IT private clouds designed as service-oriented architectures (SOAs). We provide an overview of how SOA is used as an intermediary step for cloud computing and the basic approach to SOA as it applies to data center design. We then describe the role and use of open source software in data centers. The use and importance of collaboration technologies in cloud computing architectures is also discussed. Last and most important, you will gain an understanding of how the engine of cloud computing will drive the future of infrastructure and operations design.

Ten years ago, no one could have predicted that the cloud (both hardware and software) would become the next big thing in the computing world. IT automation has evolved out of business needs expressed by customers to infrastructure management and administrators. There has never been a grand unified plan to automate the IT industry. Each provider, responding to the needs of individual customers, has been busily building technology solutions to handle repetitive tasks, respond to events, and produce predictable outcomes given certain conditions. All the while this evolutionary process was occurring, it was presumed that the cost of not doing it would be higher than just getting it done.[1] The solutions provided to

meet customer needs involved both hardware and software innovation and, as those solutions emerged, they gave rise to another generation of innovation, improving on the foundation before it. Thus the effects of Moore's law[2] seem to prevail even for cloud evolution.

From the military use of TCP/IP in the 1960s and 1970s to the development and emergence of the browser on the Internet in the late 1980s and early 1990s, we have witnessed growth at a rate similar to what Gordon Moore had predicted in 1965: essentially, a doubling of capability approximately every two years. We saw the emergence of network security in the mid/late 1990s (again, as a response to a need), and we saw the birth of performance and traffic optimization in the late 1990s/early 2000s, as the growth of the Internet necessitated optimization and higher-performance solutions. According to Greg Ness, the result has been "a renaissance of sorts in the network hardware industry, as enterprises installed successive foundations of specialized gear dedicated to the secure and efficient transport of an ever increasing population of packets, protocols and services."[3] Welcome to the world that has been called Infrastructure1.0 (I-1.0).

The evolution of the basic entity we call I-1.0 is precisely the niche area that made successful companies such as Cisco, F5 Networks, Juniper, and Riverbed. I-1.0 established and maintained routes of connectivity between a globally scaled user base constantly deploying increasingly powerful and ever more capable network devices. I-1.0's impact on productivity and commerce have been as important to civilization as the development of transoceanic shipping, paved roads, railway systems, electricity, and air travel. I-1.0 has created and shifted wealth and accelerated technological advancement on a huge number of fronts in countless fields of endeavor. There simply has been no historical precedent to match the impact that I-1.0 has had on our world. However, at this point in its evolution, the greatest threat to the I-1.0 world is the advent of even greater factors of change and complexity as technology continues to evolve. What once was an almost exclusive domain of firmware and hardware has now evolved to require much more intelligent and sophisticated software necessary for interfacing with, administering, configuring, and managing that hardware. By providing such sophisticated interfaces to firmware/hardware-configured devices, it marked

1. James Urquhart, http://blogs.cisco.com/datacenter/comments/the_network_the_final_frontier_for_cloud_computing, retrieved 5 Feb 09.
2. http://www.intel.com/technology/mooreslaw/index.htm, retrieved 6 Feb 09.
3. Greg Ness, http://gregness.wordpress.com/2008/10/13/clouds-networks-and-recessions, retrieved 5 Feb 09.

the beginning of the emergence of virtualization. When companies such as VMware, Microsoft, and Citrix announced plans to move their offerings into mainstream production data centers such as Exodus Communications, the turning point for I-1.0 became even more evident. The I-1.0 infrastructure world was on its way into the cold, dark halls of history.

As the chasm between I-1.0 and the increasingly sophisticated software packages widened, it became evident that the software could ultimately drive the emergence of a more dynamic and resilient network. This network became even more empowered by the addition of application-layer innovations and the integration of static infrastructure with enhanced management and connectivity intelligence. The evolving network systems had become more dynamic and created new problems that software was unprepared to contend with. This gave rise to a new area, virtualization security (VirtSec, which once again, arose out of necessity), and marked the beginning of an even greater realization that the static infrastructure built over the previous quarter of a century was not adequate for supporting dynamic systems or for avoiding the impact that malevolent actions would have on such dynamic networking paradigms. The recognition that new solutions had to be developed became apparent when the first virus hit back in the 1970s (The Creeper virus was first detected on ARPANET, the forerunner of the Internet, in the early 1970s[4]). No one realized at the time that this single problem would create an entire industry. As we have discussed, the driving force for all such technological innovation has been need. For the cloud, the biggest evolutionary jump began with managed service providers (MSPs) and their motivation to satisfy and retain customers paying monthly recurring fees.

3.2 The Evolution from the MSP Model to Cloud Computing and Software-as-a-Service

If you think about how cloud computing really evolved, it won't take long to realize that the first iteration of cloud computing can probably be traced back to the days of frame relay networks. Organizations with frame relay were essentially singular clouds that were interconnected to other frame relay-connected organizations using a carrier/provider to transport data communications between the two entities. Everyone within the frame network sharing a common Private Virtual Connection (PVC) could share

4. http://en.wikipedia.org/wiki/Computer_virus, retrieved 6 Feb 09.

their data with everyone else on the same PVC. To go outside their cloud and connect to another cloud, users had to rely on the I-1.0 infrastructure's routers and switches along the way to connect the dots between the clouds. The endpoint for this route between the clouds and the I-1.0 pathway was a demarcation point between the cloud and the provider's customer. Where the dots ended between the clouds (i.e., the endpoints) was where access was controlled by I-1.0 devices such as gateways, proxies, and firewalls on the customer's premises.

From customers' perspective, this endpoint was known as the main point of entry (MPOE) and marked their authorized pathway into their internal networking infrastructure. By having applications use specific protocols to transport data (e.g., Simple Mail Transfer Protocol [SMTP] for sending mail or File Transfer Protocol [FTP] for moving files from one location to another), applications behind the MPOE could accept or reject traffic passing over the network and allow email and file transfer to occur with little to no impedance from the network infrastructure or their administrators. Specialized applications (developed out of necessity to satisfy specific business needs) often required a client/server implementation using specific portals created through the firewall to allow their traffic protocols to proceed unhindered and often required special administrative setup before they could work properly. While some of this may still hold, that was, for the most part, how it was done "old school." Things have changed considerably since that model was considered state of the art. However state of the art it was, it was difficult to manage and expensive. Because organizations did not want to deal with the complexities of managing I-1.0 infrastructure, a cottage industry was born to do just that.

3.2.1 From Single-Purpose Architectures to Multipurpose Architectures

In the early days of MSPs, the providers would actually go onto customer sites and perform their services on customer-owned premises. Over time, these MSPs specialized in implementation of infrastructure and quickly figured out ways to build out data centers and sell those capabilities off in small chunks commonly known as monthly recurring services, in addition to the basic fees charged for ping, power, and pipe (PPP). *Ping* refers to the ability to have a live Internet connection, *power* is obvious enough, and *pipe* refers to the amount of data throughput that a customer is willing to pay for. Generally, the PPP part of the charge was built into the provider's monthly service fee in addition to their service offerings. Common services

provided by MSPs include remote network, desktop and security monitoring, incident response, patch management, and remote data backup, as well as technical support. An advantage for customers using an MSP is that by purchasing a defined set of services, MSPs bill a flat or near-fixed monthly fee, which benefits customers by having a predictable IT cost to budget for over time. Step forward to today and we find that many MSPs now provide their services remotely over the Internet rather than having to sell data center space and services or perform on-site client visits (which is time-consuming and expensive).

3.2.2 Data Center Virtualization

From the evolutionary growth of the MSP field, coupled with the leaps made in Internet and networking technology over the past 10 years, we have come to a point where infrastructure has become almost secondary to the services offered on such infrastructure. By allowing the infrastructure to be virtualized and shared across many customers, the providers have changed their business model to provide remotely managed services at lower costs, making it attractive to their customers. These X-as-a-Service models (XaaS) are continually growing and evolving, as we are currently standing at the forefront of a new era of computing service driven by a huge surge in demand by both enterprises and individuals. Software-as-a-Service (SaaS, and other [X]aaS offerings such as IaaS, MaaS, and PaaS) can be seen as a subset or segment of the cloud computing market that is growing all the time. One IDC report indicated that cloud computing spending will increase from $16 billion in 2008 to $42 billion in 2012.[5] Is there little wonder there is incentive for consumers to pursue cloud computing and SaaS?

Typically, cloud computing has been viewed as a broad array of Internet Protocol (IP) services (generally using an application called a Web browser as the main interface) in order to allow users to obtain a specific set of functional capabilities on a "pay for use" basis. Previously, obtaining such services required tremendous hardware/software investments and professional skills that were required in hosting environments such as Exodus Communications, Cable & Wireless, SAVVIS, and Digital Island. From an enterprise customer perspective, the biggest advantages of cloud computing and

5. Roger Smith, "IDC Says IT Cloud Services to Reach $42 Billion by 2012," http://www.informationweek.com/blog/main/archives/2008/10/idc_says_it_clo.html, October 2008, retrieved 6 Feb 2009.

SaaS over the traditional hosting environment are that cloud computing is an I-1.0 response to a business need to find a reasonable substitute for using expensive out-sourced data centers. Also, SaaS is a "pay as you go" model that evolved as an alternative to using classical (more expensive) software licensing solutions.

The cloud evolved from the roots of managed service provider environments and data centers and is a critical element of next-generation data centers when compared to the MSPs they evolved from. Today, customers no longer care where the data is physically stored or where servers are physically located, as they will only use and pay for them when they need them. What drives customer decision making today is lower cost, higher performance and productivity, and currency of solutions.

3.3 The Cloud Data Center

Unlike the MSP or hosting model, the cloud can offer customers the flexibility to specify the exact amount of computing power, data, or applications they need to satisfy their business requirements. Because customers don't need to invest capital to have these services, what we have today is a reliable and cost-effective alternative to what has been available in the past. Today, customers are able to connect to the cloud without installing software or buying specific hardware. A big reason for their desire to use the cloud is the availability of collaborative services. *Collaboration is the opiate of the masses in "cloud land."*

3.4 Collaboration

Collaboration is a very natural experience that humans have been engaging in for thousands of years. Up until the 1970s, most businesses embraced collaboration through a management style called "management by walking around." This was facilitated by corporate styles in which people tended to be working together in the same place. In the 1960s and 1970s the "head office/branch office" model emerged as companies grew in size. These introduced time and distance into business processes, but the productivity gap was minimized because branch offices tended to be autonomous and people could still easily connect with one another.

Since then, the workforce has become increasingly distributed. This has accelerated as globalization has taken hold. In the last 30 years, tools such as voice mail and email have tried to close the gap by facilitating

communications in real and nonreal (sometimes, even unreal) time. However, an increasing remote workforce coupled with the variable nature of a team (including contractors, suppliers, and customers) has meant that the productivity gap is also quickly growing. Distance and time slow down decision making and have the adverse effect of impeding innovation. Existing technology models are failing to keep up. Part of this failure has been introduced by the rapidly evolving workspace.

When we talk about the workspace, we talk about the wide variety of tools and systems that people need to do their jobs. It is the range of devices from mobile phones to IP phones, laptop computers, and even job-specific tools such as inventory scanners or process controllers. It is about the operating systems that power those tools. And it's about accessibility, as workspaces constantly change—from the home to the car, from the car to the office or to the factory floor, even to the hotel room.

Intelligent networks are used to unify not only the elements of the workspace, but also to unify workspaces among groups of users. People need to connect, communicate, and collaborate to ensure that everyone can be included in decision making. Only architectures that embrace the ever-changing workspace can enable collaboration, and only the network can ensure that the collaboration experience is universally available to all. The role of the network has been critical in driving productivity innovations. In fact, the network has fueled each of the IT-driven productivity shifts over the last 30 years.

While IBM, Microsoft, and Apple were making computing power available to all, it wasn't until the emergence of the IP network that people could connect easily from one machine and person to another. This network gave rise to both the Internet and to IP telephony. IP telephony dramatically changed the economics of communications, making corporate globalization financially feasible. IP telephony gave rise to unified communications and the ability to blend together many forms of communications including text, video, and voice. And while unified communications have enabled business transformation, it is collaboration that will close the productivity gap by overcoming the barriers of distance and time, speeding up business, and accelerating innovations by enabling the inclusion of people, anywhere.

Today's leading-edge collaboration portfolio solutions, FaceBook and Google, capture the best of two very different worlds, offering speed, ubiquity, and flexibility. Cloud-based solutions offer widely adopted standards used by legions of developers. It is where innovation happens rapidly and on

a large scale. Most applications are offered as subscription services, available on demand and hosted in distant data centers in "the cloud." The enterprise world offers certainty of availability, security, reliability, and manageability. The enterprise experience is all about consistency. It also carries with it the legacy of proprietary toolsets and slower innovation cycles. It is a world that, for reasons of compliance, is usually hosted on-premises under tight controls and purchased through a capital budget. A portfolio of products can be built to enable the best of two worlds, the speed and flexibility of the consumer world and the certainty of the enterprise world.

Collaboration is not just about technology. Collaboration is the platform for business, but to achieve it, customers must focus on three important areas. First, customers need to develop a corporate culture that is inclusive and fosters collaboration. Second, business processes need to be adapted and modified to relax command and control and embrace boards and councils to set business priorities and make decisions. Finally, customers need to leverage technologies that can help overcome the barriers of distance and time and changing workforces.

If collaboration is the platform for business, the network is the platform for collaboration. Unlike vendor-specific collaboration suites, the next-generation portfolio is designed to ensure that all collaboration applications operate better. Whether it is WaaS (Wide-Area Application Service) optimizing application performance, or connecting Microsoft Office Communicator to the corporate voice network, the foundation ensures the delivery of the collaborative experience by enabling people and systems to connect securely and reliably. On top of the network connections, three solutions are deployed to support and enable the collaborative experience. These solutions are unified communications that enable people to communicate, video that adds context to communications, and Web 2.0 applications that deliver an open model to unify communications capabilities with existing infrastructure and business applications.

Unified communications enable people to communicate across the intelligent network. It incorporates best-of-breed applications such as IP telephony, contact centers, conferencing, and unified messaging. Video adds context to communication so that people can communicate more clearly and more quickly. The intelligent network assures that video can be available and useful from mobile devices and at the desktop. Web 2.0 applications provide rich collaboration applications to enable the rapid development and deployment of third-party solutions that integrate

network services, communications, and video capabilities with business applications and infrastructure.

Customers should be able to choose to deploy applications depending on their business need rather than because of a technological limitation. Increasingly, customers can deploy applications on demand or on-premises. Partners also manage customer-provided equipment as well as hosted systems. With the intelligent network as the platform, customers can also choose to deploy some applications on demand, with others on-premises, and be assured that they will interoperate.

3.4.1 Why Collaboration?

Several evolutionary forces are leading companies and organizations to collaborate. The global nature of the workforce and business opportunities has created global projects with teams that are increasingly decentralized. Knowledge workers, vendors, and clients are increasingly global in nature. The global scope of business has resulted in global competition, a need for innovation, and a demand for greatly shortened development cycles on a scale unknown to previous generations. Competition is driving innovation cycles faster than ever to maximize time to market and achieve cost savings through economies of scale. This demand for a greatly reduced innovation cycle has also driven the need for industry-wide initiatives and multiparty global collaboration. Perhaps John Chambers, CEO and chairman of Cisco Systems, put it best in a 2007 blog post:

> Collaboration is the future. It is about what we can do together. And collaboration within and between firms worldwide is accelerating. It is enabled by technology and a change in behavior. Global, cross-functional teams create a virtual boundary-free workspace, collaborating across time zones to capture new opportunities created with customers and suppliers around the world. Investments in unified communications help people work together more efficiently. In particular, collaborative, information search and communications technologies fuel productivity by giving employees ready access to relevant information. Companies are flatter and more decentralized.[6]

6. John Chambers, "Ushering in a New Era of Collaboration," http://blogs.cisco.com/collaboration/2007/10, 10 Oct 2007, retrieved 8 Feb 2009.

Collaboration solutions can help you address your business imperatives. Collaboration can save you money to invest in the future by allowing you to intelligently reduce costs to fund investments for improvement and focus on profitability and capital efficiency without reducing the bottom line. It can also help you unlock employee potential by providing them a vehicle by which they can work harder, smarter, and faster, ultimately doing more with less by leveraging their collaborative network. With it you can drive true customer intimacy by allowing your customers to be involved in your decision process and truly embrace your ideas, personalize and customize your solutions to match customer needs, empower your customers to get answers quickly and easily, all without dedicating more resources. Even further, it can give you the opportunity to be much closer to key customers to ensure that they are getting the best service possible.

Collaboration gives you the ability to distance yourself from competitors because you now have a cost-effective, efficient, and timely way to make your partners an integral part of your business processes; make better use of your ecosystem to drive deeper and faster innovation and productivity; and collaborate with partners to generate a higher quality and quantity of leads. Ultimately, what all of these things point to is a transition to a borderless enterprise where your business is inclusive of your entire ecosystem, so it is no longer constrained by distance, time, or other inefficiencies of business processes. Currently there is a major inflection point that is changing the way we work, the way our employees work, the way our partners work, and the way our customers work. There is a tremendous opportunity for businesses to move with unprecedented speed and alter the economics of their market. Depending on a number of variables in the industry you're in, and how big your organization is, there are trends that are affecting businesses in any combination of the points made above.

Collaboration isn't just about being able to communicate better. It is ultimately about enabling multiple organizations and individuals working together to achieve a common goal. It depends heavily on effective communication, the wisdom of crowds, the open exchange and analysis of ideas, and the execution of those ideas. In a business context, execution means business processes, and the better you are able to collaborate on those processes, the better you will be able to generate stronger business results and break away from your competitors.

These trends are creating some pretty heavy demands on businesses and organizations. From stock prices to job uncertainty to supplier viability, the

global economic environment is raising both concerns and opportunities for businesses today. Stricken by the crisis on Wall Street, executives are doing everything they can to keep stock prices up. They are worried about keeping their people employed, happy and motivated because they cannot afford a drop in productivity, nor can they afford to lose their best people to competitors. They are thinking about new ways to create customer loyalty and customer satisfaction. They are also hungry to find ways to do more with less. How can they deliver the same or a better level of quality to their customers with potentially fewer resources, and at a lower cost?

Collaboration is also about opportunity. Businesses are looking for new and innovative ways to work with their partners and supply chains, deal with globalization, enter new markets, enhance products and services, unlock new business models. At the end of the day, whether they are in "survival mode," "opportunistic mode," or both, businesses want to act on what's happening out there—and they want to act fast in order to break away from their competitors.

So what choices do current IT departments have when it comes to enabling collaboration in their company and with their partners and customers? They want to serve the needs of their constituencies, but they typically find themselves regularly saying "no." They have a responsibility to the organization to maintain the integrity of the network, and to keep their focus on things like compliance, backup and disaster recovery strategies, security, intellectual property protection, quality of service, and scalability.

They face questions from users such as "Why am I limited to 80 MB storage on the company email system that I rely on to do business when I can get gigabytes of free email and voicemail storage from Google or Yahoo?" While Internet applications are updated on three- to six-month innovation cycles, enterprise software is updated at a much slower pace. Today it's virtually impossible to imagine what your workers might need three to five years from now. Look at how much the world has changed in the last five years. A few years ago, Google was "just a search engine," and we were not all sharing videos on YouTube, or updating our profiles on Facebook or MySpace. But you can't just have your users bringing their own solutions into the organization, because they may not meet your standards for security, compliance, and other IT requirements. As today's college students join the workforce, the disparity and the expectation for better answers grows even more pronounced.

The intent of collaboration is to enable the best of both worlds: web-speed innovation and a robust network foundation. New types of conversations are occurring in corporate board rooms and management meetings, and these conversations are no longer happening in siloed functional teams, but in a collaborative team environment where multiple functions and interests are represented. Enabling these collaborative conversations is more than systems and technology. It actually starts with your corporate culture, and it should be inclusive and encourage collaborative decision making. It's also not just about your own culture; your collaborative culture should extend externally as well as to your customers, partners, and supply chain. How do you include all these elements in your decision-making processes? Are you as transparent with them as you can be? How consistently do you interact with them to make them feel that they are a part of your culture? Once you have a collaborative culture, you will have the strong user base through which to collaboration-enable the processes in which people work.

All business processes should include collaborative capabilities so that they are not negatively impacted by the restrictions we see affecting processes today: time, distance, latency. At any point in a business process, whether internal or external, you should be able to connect with the information and/or expertise you need in order to get things done. This is especially true with customer-facing processes. As consumers, we always want to be able to talk directly to a person at any time if we have a question. Of course, this is all enabled by the tools and technology that are available to us today. Collaboration technology has evolved to a point where it is no longer just about being able to communicate more effectively; it is now at a point where you can drive real business benefits, transform the way business gets done, and, in many cases, unlock entirely new business models and/or new routes to market. As you look at the key business imperatives to focus on, it is important to consider the changes and/or investments you can make on any of these levels (short-term or long-term) to deliver the value you are looking for. Let's take a look at some examples now.

Customer Intimacy

When we talk about customer intimacy, we are really talking about making yourself available to communicate with them frequently in order to better understand their challenges, goals, and needs; ensuring that you are delivering what they need, in the way they need it; and including them in the decision-making processes. And just as there are a number of solutions that can improve the employee experience, your vendor should offer several

solutions that can do the same for the customer experience, including an increase in the frequency, timeliness, and quality of customer meetings; improvement in the sales success rate, reduced sales cycle time, improved and more frequent customer engagements that can lead to uncovering new and deeper opportunities, and increasing your level of communication uplevels and your relationship as a business partner, not just as a vendor.

Extending Your Reach to Support Customers Anywhere and at Any Time

You can extend your reach to support customers anywhere and at any time by promoting a collaborative culture through the use of collaborative technologies such as Wikis or blogs. Enabling customers to voice their questions, concerns, opinions, and ideas via simple web 2.0 tools such as Wikis or blogs gives them a voice and contributes tremendous feedback, ideas, and information to your business and "innovation engine." These collaborative technologies can also be used to promote employee participation to drive innovation and self-service and increase employee morale, which is key to productivity. In turn, this can yield higher customer satisfaction and loyalty in branch locations. It is really more about driving a collaborative culture than anything else. This culture is created by initiatives that promote participation in these tools, which are easier to implement and use than most executives believe. A Wiki can be a self-regulated setup for any operating system and can become one of the most helpful and information-rich resources in a company, even if the department does not support that particular operation system or have anything to do with the Wiki itself.

Save to Invest

Organizations are doing many things to cut costs to free up money to invest in the future through the use of collaborative technologies such as telepresence, unified communications, and IP-connected real estate. Telepresence has vastly simplified the way virtual collaboration takes place, currently offering the most realistic meeting experience and an alternative to traveling for face-to-face meetings with customers, suppliers, and staff as well as other essential partners. Most important, it yields significant reductions in travel costs, improved business productivity, and elimination of travel-induced stress. Consolidation and centralization of communications infrastructure and resources resulting from moving away from legacy communication systems to IP-based unified communications and management systems can

result in drastic reductions in PBX lease costs, maintenance costs, and management costs.

Mobility costs can be controlled by routing mobile long-distance calls over the Enterprise IP network. A unified communications solution allows users to place a call while they are on the public mobile network, but the call is originated and carried from the customer's communications manager cluster. In other words, now your customers can leverage a unified communications manager to manage mobile calls, offering the same cost-reduction benefits that Voice over IP (VoIP) did for land-line long-distance calls. Real estate, energy, and utility expenses can be cut by enabling remote and connected workforce through IP-connected real estate solutions. These collaborative technology solutions provide the ability to conduct in-person meetings without traveling, reduce sales cycles, significantly increase global travel savings, and increase productivity. Even better, many of these technologies can pay for themselves within a year because of their significant cost savings. Most important, these savings free up hard-earned company revenue to invest elsewhere as needed.

The opportunity is there to drive tremendous growth and productivity with new collaborations tools and composite applications, but it presents great challenges for IT. Collaboration is challenging, not only from an IT perspective but also from a political and a security perspective. It takes a holistic approach—not just throwing technology at the problem but rather an optimized blend of people, process, and technology. To fill this need, the service-oriented architecture was developed and SOA-based infrastructures were created to enable people to collaborate more effectively.

The service-oriented infrastructure is the foundation of an overall service-oriented architecture. An important part in this is the human interface and the impact of new technologies that arrived with Web 2.0. The benefits include the way IT systems are presented to the user. Service-oriented architectures have become an intermediate step in the evolution to cloud computing.

3.5 Service-Oriented Architectures as a Step Toward Cloud Computing

An SOA involves policies, principles, and a framework that illustrate how network services can be leveraged by enterprise applications to achieve desired business outcomes. These outcomes include enabling the business capabilities to be provided and consumed as a set of services. SOA is thus an

architectural style that encourages the creation of coupled business services. The "services" in SOA are business services. For example, updating a customer's service-level agreement is a business service, updating a record in a database is not. A service is a unit of work done by a service provider to achieve desired end results for a service consumer.

An SOA solution consists of a linked set of business services that realize an end-to-end business process. At a high level, SOA can be viewed as enabling improved management control, visibility, and metrics for business processes, allowing business process integration with a holistic view of business processes, creating a new capability to create composite solutions, exposing granular business activities as services, and allowing reuse of existing application assets. Differentiating between SOA and cloud computing can be confusing because they overlap in some areas but are fundamentally different. SOA delivers web services from applications to other programs, whereas the cloud is about delivering software services to end users and running code. Thus the cloud-versus-SOA debate is like comparing apples and oranges.[7]

A couple of areas that SOA has brought to the table have been mostly ignored in the rapid evolution to cloud computing. The first is governance. Although governance is not always implemented well in with SOA, it is a fundamental part of the architecture and has been generally ignored in cloud computing. The control and implementation of policies is a business imperative that must be met before there is general adoption of cloud computing by the enterprise. SOA is derived from an architecture and a methodology. Since cloud computing is typically driven from the view of business resources that are needed, there is a tendency to ignore the architecture. The second area that SOA brings to cloud computing is an end-to-end architectural approach.

Cloud service providers such as Amazon, TheWebService, Force.com, and others have evolved from the typically poorly designed SOA service models and have done a pretty good job in architecting and delivering their services. Another evolutionary step that cloud computing has taken from the SOA model is to architect and design services into the cloud so that it can expand and be accessed as needed. Expanding services in an SOA is typically a difficult and expensive process.

7. Rich Seeley, "Is Microsoft Dissing SOA Just to PUSH Azure Cloud Computing?," http://searchsoa.techtarget.com/news/article/0,289142,sid26_gci1337378,00.html, 31 Oct 2008, retrieved 9 Feb 09.

SOA has evolved into a crucial element of cloud computing as an approach to enable the sharing of IT infrastructures in which large pools of computer systems are linked together to provide IT services. Virtual resources and computing assets are accessed through the cloud, including not only externally hosted services but also those provided globally by companies. This provides the basis for the next generation of enterprise data centers which, like the Internet, will provide extreme scalability and fast access to networked users. This is why cloud computing can be used across an entire range of activities—a big advantage over grid computing, which distributes IT only for a specific task.

Placing information, services, and processes outside the enterprise without a clear strategy is not productive. A process, architecture, and methodology using SOA and for leveraging cloud computing is used. As part of the enterprise architecture, SOA provides the framework for using cloud computing resources. In this context, SOA provides the evolutionary step to cloud computing by creating the necessary interfaces from the IT infrastructure to the cloud outside the enterprise. Cloud computing essentially becomes an extension of SOA. Services and processes may be run inside or outside the enterprise, as required by the business. By connecting the enterprise to a web platform or cloud, businesses can take advantage of Internet-delivered resources that provide access to prebuilt processes, services, and platforms delivered as a service, when and where needed, to reduce overhead costs. We have discussed SOA as an evolutionary step because you don't move to cloud computing from SOA or replace SOA with cloud computing but rather use SOA to enable cloud computing or as a transit point to cloud computing. SOA as an enterprise architecture is the intermediate step toward cloud computing.

3.6 Basic Approach to a Data Center-Based SOA

A service-oriented architecture is essentially a collection of services. A service is, in essence, a function that is well defined, self-contained, and does not depend on the context or state of other services. Services most often reflect logical business activities. Some means of connecting services to each other is needed, so services communicate with each other, have an interface, and are message-oriented. The communication between services may involve simple data passing or may require two or more services coordinating an activity. The services generally communicate using standard protocols, which allows for broad interoperability. SOA encompasses legacy systems and processes, so

the effectiveness of existing investments is preserved. New services can be added or created without affecting existing services.

Service-oriented architectures are not new. The first service-oriented architectures are usually considered to be the Distributed Component Object Model (DCOM) or Object Request Brokers (ORBs), which were based on the Common Object Requesting Broker Architecture (CORBA) specification. The introduction of SOA provides a platform for technology and business units to meet business requirements of the modern enterprise. With SOA, your organization can use existing application systems to a greater extent and may respond faster to change requests. These benefits are attributed to several critical elements of SOA:

1. Free-standing, independent components
2. Combined by loose coupling
3. Message (XML)-based instead of API-based
4. Physical location, etc., not important

3.6.1 Planning for Capacity

It is important to create a capacity plan for an SOA architecture. To accomplish this, it is necessary to set up an initial infrastructure and establish a baseline of capacity. Just setting up the initial infrastructure can be a challenge. That should be based on known capacity requirements and vendor recommendations for software and hardware. Once the infrastructure is set up, it is necessary to establish a set of processing patterns. These patterns will be used to test capacity and should include a mix of simple, medium, and complex patterns. They need to cover typical SOA designs and should exercise all the components within the SOA infrastructure.

3.6.2 Planning for Availability

Availability planning includes performing a business impact analysis (BIA) and developing and implementing a written availability plan. The goal is to ensure that system administrators adequately understand the criticality of a system and implement appropriate safeguards to protect it. This requires proper planning and analysis at each stage of the systems development life cycle (SDLC). A BIA is the first step in the availability planning process. A BIA provides the necessary information for a administrator to fully understand and protect systems. This process should fully characterize system

requirements, processes, and interdependencies that will determine the availability requirements and priorities. Once this is done, a written availability plan is created. It should define the overall availability objectives and establish the organizational framework and responsibilities for personnel. Management should be included in the process of developing availability structure, objectives, roles, and responsibilities to support the development of a successful plan.

3.6.3 Planning for SOA Security

The foundations of SOA security are well known and are already widely used in the IT industry. SOA practitioners have come to realize they also must understand these foundations in order to provide adequate security for the systems being developed. The foundations include public key infrastructure (PKI), the common security authentication method Kerberos, XML (Extensible Markup Language) encryption, and XML digital signatures. Three main areas of concern are widely accepted as part of the SOA security arena. First, message-level security provides the ability to ensure that security requirements are met in an SOA environment, where transport-level security is inadequate because transactions are no longer point-to-point in SOA. Second, Security-as-a-Service provides the ability to implement security requirements for services. Third, declarative and policy-based security provides the ability to implement security requirements that are transparent to security administrators and can be used to quickly implement emerging new security requirements for services that implement new business functionalities.

Message-Level Security

The OASIS set of WS-Security standards addresses message-level security concerns. These standards are supported by key vendors including IBM, Microsoft, and Oracle. The standards provide a model describing how to manage and authenticate message exchanges between parties (including security context exchange) as well as establishing and deriving session keys. The standards recommend a Web service endpoint policy describing the capabilities and constraints of the security and other business policies on intermediaries and endpoints including required security tokens, supported encryption algorithms, and privacy rules. Furthermore, a federated trust model describing how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities, is described. The standards include a Web service trust model that

describes a framework for trust models that enables Web services to operate securely. There is also an authorization model describing how to manage authorization data and authorization policies. Finally, the standards include a Web service privacy model describing how to enable Web services and requesters to state subject privacy preferences and organizational privacy practice statements.

Security-as-a-Service

Security-as-a-Service can be accomplished by collecting an inventory of service security requirements throughout the enterprise architecture (EA) and specifying the set of discrete security services that will be needed for the enterprise. Next, the organization must complete the process of designing and implementing these security services as services themselves. Often, a toolkit approach can help specify the set of typical security services that may be used to provide most of the requirements and accelerate the establishment of Security-as-a-Service in an organization.

Declarative and Policy-Based Security

Implementation of declarative and policy-based security requires tools and techniques for use at the enterprise management level and at the service level. These tools and techniques should provide transparency for security administrators, policy enforcement, and policy monitoring. When policy violations are detected, alerts should be issued. Traceability of such violations, both for data and users, should be included as a critical element.

3.7 The Role of Open Source Software in Data Centers

The Open Source Initiative uses the Open Source Definition to determine whether a software license can truly be considered open source. The definition is based on the Debian Free Software Guidelines,[8] written and adapted primarily by Bruce Perens.[9] Under Perens's definition, the term *open source* broadly describes a general type of software license that makes source code available to the public without significant copyright restrictions. The principles defined say nothing about trademark or patent use and require no cooperation to ensure that any common audit or release regime applies to

8. Bruce Perens, "Debian's 'Social Contract' with the Free Software Community," http://lists.debian.org/debian-announce/debian-announce-1997/msg00017.html, retrieved 08 Feb 2009.
9. Bruce Perens, "The Open Source Definition," http://opensource.org/docs/osd, 1999, retrieved 08 Feb 2009.

any derived works. It is considered as an explicit "feature" of open source that it may put no restrictions on the use or distribution by any organization or user. It forbids this, in principle, to guarantee continued access to derived works even by the major original contributors.

Over the past decade, open source software has come of age. There has always been a demand for software that is free, reliable, and available to anyone for modification to suit individual needs. Open source distributions such as Red Hat, OpenSuSE, and BSD, coupled with open source applications such as Apache, MySQL, and scores of others have long been used to power databases, web, email, and file servers. However, something that has as much impact as the applications used in a data center has caused many implementors to hesitate to adopt open source software—until now. Recently, more than just a few users have become strong advocates that open source can and does work in the data center environment. In an online article, Robert Wiseman, chief technology officer at Sabre Holdings (a travel marketing and distribution technology company in Southlake, Texas, that uses open source software on over 5,000 servers) stated:

> It's true that with open-source products, users generally forfeit the security of professional support teams to help resolve their problems quickly. But in our environment, we almost always purchase support for our open-source products from high-quality vendors. This, of course, reduces some of the cost advantages of using open source, but the advantages are big enough that there's still plenty left over, and the security we get from a service contract lets us sleep better at night.[10]

Sabre Holdings uses an enterprise service bus for message transformation, routing, and other tasks. An enterprise service bus (ESB) refers to a software architecture construct that is typically implemented by technologies seen as a type of middleware infrastructure. ESBs are usually based on recognized standards and provide fundamental services for complex architectures via an event-driven and standards-based messaging engine (called the bus since it transforms and transports the messages across the architecture).

10. Julie Sartain, "Open-Source Software in the Data Center—There Is a Place for It, but It Won't Do Everything," http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9057879 (Computerworld, 25 Jan 2008), retrieved 08 Feb 2009.

One example of open source ESB, Apache Synapse, is an easy-to-use and lightweight ESB that offers a wide range of management, routing, and transformation capabilities. With support for HTTP, SOAP, SMTP, JMS, FTP, and file system transports, it is considered quite versatile and can be applied in a wide variety of environments. It supports standards such as WS-Addressing, Web Services Security (WSS), Web Services Reliable Messaging (WSRM), efficient binary attachments (MTOM/XOP), as well as key transformation standards such as XSLT, XPath, and XQuery. Synapse supports a number of useful functions out of the box, without programming, but it also can be extended using popular programming languages such as Java, JavaScript, Ruby, and Groovy.

Another example is a project called Open ESB, which implements an enterprise service bus runtime with sample service engines and binding components. Open ESB allows easy integration of enterprise applications and web services as loosely coupled composite applications. This allows an enterprise to seamlessly compose and recompose composite applications, realizing the benefits of a true service-oriented architecture.

Today, most users of open source agree that these products have now reached a level of maturity equal to and, in some cases, better than their commercial counterparts. Open source products have forced commercial vendors to compete on price and quality of service. Because open source code is open and transparent, developers can troubleshoot problems and learn how other developers have addressed issues. Users gain the freedom to use these products across their organizations, all over the world, without worrying about tracking client licenses.

3.8 Where Open Source Software Is Used

Perhaps because of the great flexibility of open source, which facilitates the efforts of large commercial users, cloud implementors, and vendors most of all, the successful applications of open source have evolved from within consortia. These consortia employ other means, such as trademarks, to control releases, documentation, etc., and they require specific performance guarantees from their members to assure reintegration of improvements. Accordingly, consortia do not want or need potentially conflicting clauses in their licenses. Perens's open source definition has led to a proliferation of other types of licenses that claim to be open source but would not satisfy the *share alike* provision that free software and Open Content Licenses require.

An alternative, commonly used license, the Creative Commons License, requires commercial users to acquire a separate license when the product is used for profit. This contradicts open source principles, because it discriminates against a type of use or user. However, the requirement imposed by free software to reliably redistribute derived works does not violate these principles. Accordingly, free software and consortia licenses are a variant of open source, while an Open Content License is not.

Now that we understand exactly what open source is, lets look at how some open source software is used in cloud computing.

3.8.1 Web Presence

Web presence refers to the appearance of an entity on the World Wide Web. It is said that a company has web presence if it is accessible on the WWW. A common measure of web presence tends to be the number of pages or sites an entity owns. This web presence may include web sites, social network profiles, and search engine ranking, traffic, popularity, and links. Open source software commonly used to assist in web presence includes Apache, the Zend Framework, and Jetty.

Apache

The Apache project began in 1995 as a collaborative effort between a group of webmasters who wanted to build a robust and commercial-grade implementation of the HTTP protocol and make it available to everyone free of charge. Originally conceived as a series of patches to the original NCSA httpd daemon, the project ultimately took on a life of its own, with the NCSA daemon undergoing several redesigns in order to make it more extensible and modular. The term Apache Server is derived from a play on the words A PAtCHy sErver—paying homage to Apache's birth as a continual series of patches applied to the existing Linux-based daemon httpd. Today, the Apache product is powerful enough to meet the needs of nearly any enterprise, yet it is simple enough to configure that most administrators can get it up and running in a few minutes.

To illustrate the powerful effect that open source software is having on cloud architectures today, the January 2009 survey conducted by Netcraft evaluated responses from 185,497,213 sites, reflecting an uncharacteristic monthly loss of 1.23 million sites.[11] Analysis showed that Apache's market share grew by more than 1 percentage point during the month of January

11. http://news.netcraft.com/archives/web_server_survey.html, retrieved 08 Feb 2009.

2009, extending its lead over second-ranked commercial product Microsoft IIS (which has fallen to less than a third of the market share at 32.91%). During this time, Apache gained 1.27 million sites and enjoyed a 52.26% market share. The Microsoft IIS product showed the largest loss for this period, after more than 2 million blogging sites running Microsoft-IIS expired from the survey. This is very impressive for a free, open source product that began life as a series of patches to a little-bitty Linux daemon.

Apache is truly a cloud-based and cloud-owned tool. Today, the Apache HTTP Server Project continues to be a collaborative software development effort boasting a commercial-grade, full-featured, freely available (with source code) implementation of an HTTP (web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet and the web to communicate, plan, and develop the server and its related documentation.

Jetty

Jetty is also an open source, standards-based, full-featured web server implemented entirely in Java.[12] Java implementation means that it is capable across platforms—meaning it can run on pretty much any platform that can run Java. Jetty is released under the Apache 2.0 licence and is therefore free for commercial use and distribution. It was created in 1995 and since then has benefitted from input from a vast user community and consistent development by a strong core of open source developers. Jetty aims to be as unobtrusive as possible. Built with such a strong focus on simplicity, the Jetty mantra is "simplicity not complexity." Once it is installed, Jetty configuration is accomplished by either an API or XML configuration file. Default configuration files provided with the open source download make Jetty usable right out of the box. Jetty is also highly scalable. For example, in asynchronous Web 2.0 applications using AJAX (Asynchronous JavaScript and XML), connections to the server can stay open longer than when serving up static pages. This can cause thread and memory requirements to escalate drastically. Cloud infrastructure must be able to cope with these types of load situations gracefully or risk catastrophes such as the possibility of a slow database connection bringing down an entire site because of a lack of available resources (threads). Jetty ensures performance degrades gracefully under stress, providing a higher

12. http://www.mortbay.org/jetty, retrieved 08 Feb 2009.

quality of service. Leveraging existing web specifications, Jetty can handle large user loads and long-lived sessions easily.

Zend Framework

The Zend Framework (ZF) was conceived in early 2005 and was publicly announced at the first Zend Conference.[13] ZF is an open source, object-oriented web application framework for the hyptertext preprocessor language PHP. At the time of its introduction, no other framework was widely available to the PHP community to fill the need for an industrial-strength open source web development toolset. Wanting more than a simple toolset, the designers of ZF sought to combine ease of use and rapid application development features with the simplicity and pragmatic approach to web development that is highly valued in the PHP community.

ZF is often called a component library because it has many components that can be used more or less independently. However, ZF provides an advanced Model-View-Controller (MVC) that can be used to establish basic structure for ZF applications. All components are object-oriented using PHP 5 and support "use at will," in that using these components entails only minimal interdependencies. ZF provides support for many of the major commercial and open source database systems, including MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, and Informix Dynamic Server. ZF also provides email composition and delivery features, and supports retrieval of email via mbox, Maildir, POP3, and IMAP4. It has a flexible caching subsystem with support for many types of back-end architectures (e.g., memory or file systems).

The ZF MVC implementation has become a *de facto* standard in the design of modern web applications because it leverages the fact that most web application code falls into one of three categories: presentation, business logic, or data access. MVC models this separation of categories quite well. This allows presentation code to be consolidated in one part of an application, business logic in another part of the application, and data access code in yet another. Many developers have found this well-defined separation indispensable for maintaining their code.

Let's take a quick look at what MVC really entails, starting with the Model. This is the part of a ZF application that defines basic functionality

13. Oonagh Morgan, "Zend Announces Industry-Wide PHP Collaboration Project at Its Inaugural PHP Conference," Zend Technologies, http://www.zend.com//news/zendpr.php?ozid=109, 19 Oct 2005, retrieved 8 Feb 2009.

using a set of abstractions. Data access routines and some business logic can also be defined in the Model. The View defines exactly what is presented to the user. The Controller binds the whole thing together. Usually, controllers pass data to each view, specifying how it should be rendered. Views often are used to collect data from the user. This is where standardized HTML markup can be used in MVC applications. They manipulate models, decide which view to display based on user input and other factors, then pass along the data to each view as needed.

Sometimes, there is a need to hand off control to another controller entirely. In cloud computing, having a standardized architecture that facilitates web presence is highly desirable and explains the increased use seen with open source in data centers. Now let's move from web presence influences to the data tier[14] itself.

3.8.2 Database Tier

Whether an application resides on a desktop or is virtualized in a cloud somewhere, when data is used or stored, it often requires the use of a database. A database is a structured collection of records or data that is stored in a computer system. A database relies on software known as a database management system (DBMS) to organize, store, and retrieve data. Database management systems are categorized according to the database model that they support. The model chosen often determines the type of (often structured) query language that is used to access the database. The structure is achieved by organizing the data according to a database model. The model in most common use today is the relational database model. Other models, such as the hierarchical model and the network model, use a more explicit representation of relationships, but they are not commonly used in cloud environments.

A great deal of the internal engineering of a DBMS is done independent of the data model it supports. While data storage, access, and retrieval are important, they are most often defined by standards and implemented accordingly. A DBMS implementation is often less concerned with how the data is accessed and retrieved and more concerned with managing performance, concurrency, integrity, and recovery from hardware failures. In these areas, there are large differences between almost all products. It is these differences that separate them from one another.

14. In computing usage, the word tier is synonymous with layer. As such, a tier implies something that sits on top of or between something else.

All of the products we will discuss in this section are relational database management systems (RDBMS) and implement the features of the relational model outlined above.

MySQL

MySQL is *the* preferred open source database based on usage. According to the MySQL web site,[15] it has become the world's most popular open source database. It is used by individual developers as well as by many of the world's largest companies, including Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, and Zappos.com. MySQL runs on more than 20 platforms, including Linux, Windows, OS/X, HP-UX, AIX, and Netware. Users can freely download and deploy MySQL from the official web site without cost. This product is in use in millions of small to medium-scale applications. MySQL is the preferred database in LAMP architecture (Linux/Apache/MySQL/PHP-Python-Perl). This regal position affords MySQL access to over two-thirds of the world's web database servers. MySQL is deployed with nearly every Linux distribution, and is easily installed on Windows, Mac, and Solaris platforms for both server and client use. In the cloud, MySQL is the king of the database server packages because it is proven, reliable, scalable, and free.

However, MySQL is not without some minor problems. The rapid pace of development has left some of its users faced with major upgrade tasks. Until the release of version 5.1, MySQL had to take a back seat to commercial enterprise-grade database products such as Oracle and IBM's DB2 because of a lack of clustering, partitioning, and replication features. With the 5.1 release, those hurdles were overcome. Now, spatial data, Web Services, and native XML support are what has to be overcome.

PostgreSQL

PostgreSQL is another powerful open source DBMS. According to the official web site,[16] it has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems and prides itself in standards compliance. PostgreSQL has a fully relational system catalog which itself supports multiple schemas per database.

15. http://www.mysql.com/why-mysql, retrieved 08 Feb 2009.
16. http://www.postgresql.org, retrieved 08 Feb 2009.

PostgreSQL is highly scalable, both in the magnitude of data it can manage and in the number of concurrent users it can accommodate. There are active PostgreSQL systems in production environments that manage in excess of 4 TB of data. For larger cloud implementations, PostgreSQL may be the DBMS of choice. Another important point to consider for any cloud implementation of a database tier is the security of the database. Accordingly, PostgreSQL is considered by many to be the most secure out-of-the-box configuration available for a database. PostgreSQL boasts many sophisticated features and is another good choice for cloud computing applications.

Data is used in many applications in the cloud. What specific applications—particularly open source applications—use this data? Let's find out.

3.8.3 Application Tier

A multitier architecture (or *n*-tier architecture) is a client-server architecture in which the presentation, application processing, and data management are logically separate processes. Most often, multitier architecture refers to a three-tier architecture—that is, presentation, application, and data tiers. The presentation tier is the topmost level of the application. The presentation tier displays information to the user, often via a web browser or windowed form. It communicates with other tiers by transferring input or data results to the other tiers in the architecture. The application tier is sometimes referred to as the business logic tier. It controls an application's functionality by performing detailed processing to satisfy specific requirements. Finally, the data tier consists of a database server or servers which are used to store and retrieve data. This tier keeps all data independent from the application or business logic tier and the presentation tier. Giving data its own tier greatly improves scalability and performance and allows applications to share data from a centralized repository.

Zope

In cloud computing, most back-end infrastructures rely an *n*-tier architecture, as shown in Figure 3.1. Zope is an open source application server for building content management systems, intranets, portals, and custom applications.

The Zope community consists of hundreds of companies and thousands of developers all over the world, working on building the platform itself and the resulting Zope applications. Zope can help developers quickly create dynamic web applications such as portal and intranet sites. Zope

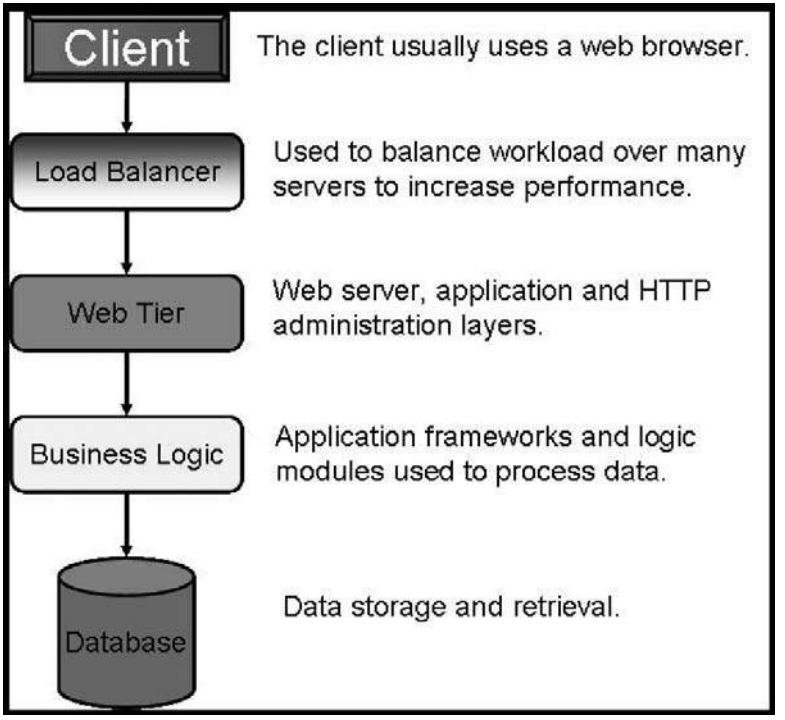

Figure 3.1 The n-tier architecture used in cloud environments.

comes with everything you need, including support for membership, search, and news. Zope provides top-notch access to databases and other legacy data. Zope is written in Python, a highly productive, object-oriented scripting language.

Zope features a transactional object database which can store not only content and custom data, but also dynamic HTML templates, scripts, a search engine, and relational database connections and code. It features a strong through-the-web development model, allowing you to update your web site from anywhere in the world. To allow for this particular feature, Zope integrates a tightly integrated security model. Built around the concept of *safe delegation of control,* Zope's security architecture also allows you to turn control over parts of a web site to other organizations or individuals.

The transactional model applies not only to Zope's object database, but, through connectors, to many other relational databases as well. This helps to ensure strong data integrity. This transaction model ensures that all data is successfully stored in connected data stores by the time a response is

returned to a web browser. According to the Zope web site,[17] numerous products (plug-in Zope components) are available for download to extend the basic set of site building tools. These products include new content objects; relational database and other external data source connectors; advanced content management tools; and full applications for e-commerce, content and document management, or bug and issue tracking. Zope includes its own HTTP, FTP, WebDAV, and XML-RPC serving capabilities, but it can also be used with Apache or other web servers. Zope users include major business entities such as Viacom, SGI, AARP, Verizon Wireless, Red Hat, NASA, and the U.S. Navy.

Zope Content Management Framework

On top of what Zope offers out of the box, there are a variety useful applications available for those who need something right away. The Content Management Framework (CMF) adds many tools and services to Zope to allow community- or organization-based content management. It comes with a workflow system and a powerful customization framework. The CMF Workflow system leverages Zope's built-in security architecture. A major feature of the CMF Workflow system is the ability for edit permissions to be taken away from an author once he or she has submitted a document for review and publishing. This ensures that what the reviewer sees won't change during or after review without the author intentionally taking control of the document.

Plone

Plone is built to leverage the CMF platform and is basically a very well designed interface that sits on top of the CMF. You can download Plone, run the installer, and in short order have a community or organizational web site (i.e., a *collab-net*) with content such as news, documentation, and events, which are supplied by members of the collab-net. The collab-net can be comprised of almost any grouping that shares common goals or interests. Because Plone is built on the CMF, it delivers the same powerful set of tools mentioned above while adding helpful content entry forms and validation.

AJAX

AJAX (Asynchronous JavaScript and XML) is a collection of interrelated standards-based web development techniques that are used to create highly

17. http://www.zope.org/WhatIsZope, retrieved 08 Feb 2009.

interactive (rich) Internet applications. The use of AJAX has led to an increase in interactive animation on web pages. AJAX web applications can retrieve data from the server asynchronously,[18] without interfering with the display or behavior of the current page. In many cases, related pages on a web site consist of much content that is common between them. Using traditional methods, that content must be reloaded with every request. With AJAX, however, a web application can request only the content that needs to be updated, thus drastically reducing bandwidth usage and load time. AJAX can reduce connections to the server, since scripts and style sheets only have to be requested once. Users may perceive the application to be faster or more responsive, even if the application has not changed on the server side.

In current use, JavaScript and XML are no longer required and the requests don't actually need to be asynchronous. The acronym AJAX has thus changed to Ajax, which does not represent use of these specific technologies. Microsoft, of course, has its version of AJAX, called ASP.NET AJAX. This is also a free framework for quickly creating efficient and interactive web applications that work across all popular browsers. ASP.NET AJAX is built into ASP.NET 3.5.

Apache Struts

Apache Struts is another open source framework for creating Java web applications. The Apache Struts Project is the open source community that creates and maintains the Apache Struts framework. The project is called "Struts" because the framework is meant to furnish the invisible underpinnings that support professional application development. Struts provides the glue that joins the various elements of the standard Java platform into a coherent whole. The goal is to leverage existing standards by producing the missing pieces to create enterprise-grade applications that are easy to maintain over time.

The Apache Struts Project offers two major versions of the Struts framework. Struts 1 is recognized as the most popular web application framework for Java. The 1.x framework is mature, well documented, and widely supported. Struts 1 is the best choice for teams that value proven solutions to common problems. Struts 2 was originally known as WebWork

18. In computer programming, an asynchronous operation is a process capable of operating independently of other processes. Conversely, a synchronous operation means that the process runs only as a result of some other process being completed or handing off the operation.

2. After working independently for several years, the WebWork and Struts communities joined forces to create Struts 2. The 2.x framework is the best choice for teams that value elegant solutions to difficult problems.

Web applications differ from conventional web sites in that web applications can create a dynamic response. Many web sites deliver only static pages. A web application can interact with databases and business logic engines to customize a response. Web applications based on JavaServer Pages sometimes commingle database code, page design code, and control flow code. In practice, unless these concerns are separated, larger applications may become difficult to maintain. One way to separate concerns in a software application is to use a Model-View-Controller architecture, as described previously. The Struts framework is designed to help developers create web applications that utilize a MVC architecture. The Struts framework provides three key components:

1. A request handler that is mapped to a standard URI.[19]
2. A response handler that transfers control to another resource to complete the response.
3. A tag library that helps developers create interactive form-based applications with server pages.

The Struts framework's architecture and tags are compliant with most common applicable standards. Struts works well with conventional REST[20] applications and with newer technologies such as SOAP (Simple Object Access Protocol) and AJAX.

3.8.4 Systems and Network Management Tier

Open source software has developed strong roots in the cloud community. Much of the cloud operates in a mission-critical space, so there is often great trepidation about whether investment in a commercial application may be a better option. However, many developers have come to realize that the "sweet spot" for open source is actually that mission-critical space. Given the high reliability and maturity of many of the better-known open

19. A Uniform Resource Identifier (URI) is a string of characters used to identify or name a resource on the Internet.
20. REpresentational State Transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web.

source solutions available, there are many reasons why implementers are starting to give open source more than a passing glance when evaluating options. Many of the commercial offerings available offer open source solutions and make their money providing enhancements to the open source, service and support, and other types of services that enhance customer adoption of their product.

Open source is not with its detractors, however. Many experts still advise caution when it comes to adopting open source solutions. They argue that users of open source software can potentially risk encountering security issues because the software internals are so widely known. Adoptors are encouraged to research into which industrial-strength software is available for their particular mission-critical environment and to compare potential open source candidates. For mission-critical environments, especially within the context of cloud computing, we see several major categories:

1. Administrative and management applications
2. Performance applications
3. Monitoring and security applications
4. Virtualization applications

In the next few paragraphs, we will discuss the salient features of each of these categories and provide examples of the types of open source and (sometimes) commercial applications used today. This will provide a good starting point for understanding what implementation of cloud infrastructure entails from the perspective of the engineering team that is tasked with putting a solution architecture together and making it work for a business.

Performance Monitoring and Management Applications

Performance monitoring is critical for businesses operating mission-critical or data-intensive IT infrastructure that provides access to users on-site, from remote office locations, and from mobile devices. Many factors can influence the performance of a network, such as the number of users accessing it, the bandwidth capacity, use of coding platforms and protocols, and attacks on its vulnerabilities. Performance monitoring tools are used by organizations to ensure that their networks and the applications delivered over them operate at the highest levels of performance

achievable. Monitoring performance is a proactive approach to mitigating risk and limiting potential damage.

The purpose of administrative and management applications is to facilitate systems personnel in administering large numbers of hosts and services running simultaneously. The monitoring software watches over hosts and services, alerting administrators when conditions are detected operating outside defined parameters. Common tasks performed by this type of software include monitoring network services (SMTP, POP3, HTTP, NNTP, PING, etc.) and host resources (processor load, disk usage, etc.). Many packages support the use of plug-ins that allow users to easily develop their own service checks or add functionality from third-party developers (this is especially common in the open source development arena). The ability to define a hierarchy of network host parent–child node relationships is desirable because it allows detection of and distinction between hosts that are down and those that are unreachable.

An important feature that nearly all such products support is contact notification whenever a service or host problem is detected and/or is resolved. Such notification usually occurs using email, a pager, or Instant Messaging (IM). When a problem occurs that needs to be resolved, some packages offer the ability to define event handlers to be executed when service or host events are detected, which then initiate proactive problem resolution. Some of the many other commonly available features of administrative and management applications include automatic log file rotation and support for implementing redundant monitoring hosts. Most packages interact with administrative staff using a web interface (which generally can be accessed remotely) for viewing current network status, notification and problem history, log file, etc.

Now that we understand the types of application used for administering cloud infrastructure, let's take a look at a couple of examples.

openQRM

openQRM is a comprehensive, flexible, open source infrastructure management solution. Supporting a highly modular architecture, openQRM focuses on automatic, rapid, appliance-based deployment. It is used for monitoring high-availability cloud computing infrastructure, especially when deployment is implementing multiple virtualization technologies. openQRM is a unified management console for IT infrastructure, and it provides a well-defined API that can be used to integrate third-party tools as plug-ins. This feature provides companies with a highly scalable system that

supports any size business with high-availability requirements running multiple operating systems on various platforms.

Key features of openQRM include complete separation of the physical devices and virtual machines from the software (or virtualized server-images/instances. With openQRM, hardware is seen as a computing resource that can be easily replaced without reconfiguring hardware. Support for different virtualization technologies such as VMware, Xen, KVM, and Linux-VServer vms can be managed transparently via openQRM. It can support P2V (physical to virtual), V2P (virtual to physical), and V2V (virtual to virtual) migration efforts. Because of this flexibility, not only can servers be migrated from physical to virtual environments or virtual to physical environments easily, they can be easily migrated from virtual environment A to virtual environment B.

The commercial product Nagios is one of the best available. Nagios also performs system, network, and application monitoring. openQRM supports fully automatic Nagios configuration to monitor all systems and services using a completely automatic configuration module that maps the entire network. When administrators deploy a new bank of servers, it is quite easy to configure them for Nagios using openQRM.

High-availability is another strong feature that OpenQRM supports. It follows a many-to-one fail-over model. This means that multiple high-availability servers can operate using a single standby system. It is a simple matter to bring up a virtual machine as a standby. In case of problems, the high-availability devices will fail-over from their physical environment to a virtual environment or from one virtual to another virtual environment. To facilitate this, ready-made server images are available from the image shelf plug-in. openQRM provides out-of-the-box images for Debian, Ubuntu, CentOS, and openSuse via a web interface. OpenQRM supports integrated storage management;there is a single place for backup/restore on the storage server, so its cloning/snapshot features can be reused at any time to create "hot backups" for servers without a service interruption.

Zenoss

Zenoss (also called Zenoss Core), another open source application, is a server and network management platform based on the Zope application server. It is released under the GNU General Public License (GPL) version 2. Zenoss Core provides a web interface that allows system administrators to monitor availability, inventory/configuration, performance, and events. It is the most popular IT management project on SourceForge.Net.[21] According

to the Zenoss web site,[22] Zenoss software has been downloaded over 750,000 times and deployed in over 18,000 organizations in over 175 countries. Zenoss Core provides nearly all of the same capabilities mentioned above for openQRM. Additionally, Zenoss offers the Zenoss Configuration Management Database (CMDB).

The CMDB houses a unified model of the entire IT environment and provides the basis for the Zenoss "model-driven" IT monitoring approach. The first commercial open source CMDB on the market, Zenoss CMDB features include modeling the entire environment—networks, servers, software, and applications. Zenoss provides inventory management, change tracking services, and availability monitoring, so users can have a real-time of view of availability throughout the IT infrastructure. This feature provides IT staff with information they need to quickly identify problems and potential outage locations. Zenoss has the ability to map IT elements and cross-platform information into a normalized data structure and to create logical and physical groupings that relate to business systems, locations, and responsible parties. Data is populated using a combination of processes such as auto-discovery, the web services API and XML import/export, and manual input. Zenoss also allows administrators to create configuration policies that specify required configuration items.

Performance monitoring with Zenoss provides high-speed collection, historical graphing, and real-time threshold analysis for any available metric in the IT environment. Event monitoring and management provides the ability to aggregate log and event information from various sources, including availability monitoring, performance monitoring, syslog sources, SNMP trap sources, and the Windows Event log. IT staff can create event processing rules through a graphical user interface (GUI) with automatic event classification and prioritization. Alerting and reporting is also an integral part of Zenoss solutions. Customized alert messages are sent via paging or emails. Zenoss also includes basic and advanced escalation rules to avoid alarm fatigue.[23] Finally, the browser-based Zenoss Dashboard is easily customized to provide personal views based on geographies, systems, business applications, or other options to suit the needs of the end users. A web portal ensures secure web-based access using role-based permissions.

21. Sourceforge.net is the largest repository for open source software.
22. http://www.zenoss.com/product/network-monitoring, retrieved 09 Feb 2009.
23. Like the little boy who cried wolf, too many alarms can desensitize one to the value of an alarm.

The personalized dashboard summarizes active events by business system and severity.

Load Balancing

With the explosive growth of the Internet and its increasingly important role in our lives, the traffic on the Internet is increasing dramatically, which has been growing at over 100% annualy. The workload on servers is increasing rapidly, so servers may easily be overloaded, especially serversfor a popular web site. There are two basic solutions to the problem of overloaded servers, One is a single-server solution, i.e., upgrade the server to a higher-performance server. However, the new server may also soon be overloaded, requiring another upgrade. Further, the upgrading process is complex and the cost is high. The second solution is a multiple-server solution, i.e., build a scalable network service system on a cluster of servers. When load increases, you can simply add one or more new servers to the cluster, and commodity servers have the highest performance/cost ratio. Therefore, it is more scalable and more cost-effective to build a server cluster system for network services.

A server is limited in how many users it can serve in a given period of time, and once it hits that limit, the only options are to replace it with a newer, faster machine, or to add another server and share the load between them. A load balancer can distribute connections among two or more servers, proportionally cutting the work each has to do. Load balancing can help with almost any kind of service, including HTTP, DNS, FTP, POP/IMAP, and SMTP. According to the online web encyclopedia Wikipedia, load balancing is

> a technique to spread work between two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, and minimize response time. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The balancing service is usually provided by a dedicated program or hardware device (such as a multilayer switch). It is commonly used to mediate internal communications in computer clusters, especially high-availability clusters.[24]

24. http://en.wikipedia.org/wiki/Load_balancing_(computing), retrieved 10 Feb 2009.

Cloud-based server farms can achieve high scalability and availability using server load balancing. This technique makes the server farm appear to clients as a single server. Load balancing distributes service requests from clients across a bank of servers and makes those servers appear as if it is only a single powerful server responding to client requests.

In most common server load balancing architectures, an incoming request is redirected to a load balancer that is transparent to the client making the request. Based on predetermined parameters, such as availability or current load, the load balancer typically uses a round-robin scheduling algorithm to determine which server should handle the request and then forwards the request on to the selected server. To make the final determination, the load balancing algorithm, the load balancer retrieves information about the candidate server's health and current workload in order to verify its ability to respond to a request.

Load balancing solutions can be divided into software-based load balancers and hardware-based load balancers. Hardware-based load balancers are specialized boxes that include Application Specific Integrated Circuits (ASICs) customized for a specific use.[25] ASICs enable high-speed forwarding of network traffic and are often used for transport-level load balancing, because hardware-based load balancers are much faster than software solutions. Software-based load balancers run on standard operating systems and standard hardware components such as desktop PCs. We will look at an open source software solution called Linux Virtual Server next.

Linux Virtual Server Load Balancer

The Linux Virtual Server is an advanced load balancing solution that can be used to build highly scalable and highly available network services such as HTTP, POP3, SMTP, FTP, media and caching, and Voice over Internet Protocol (VoIP). There are more than a few open source load balancing applications available today, but the Linux Virtual Server (LVS) continues to be one of the most popular. LVS is a simple, powerful product used for load balancing and fail-over. LVS behavior is controlled at runtime by issuing commands using a Command Line Interface (CLI). The syntax used for issuing these commands is very straightforward and simple. The LVS cluster[26] system is also known as a load balancing server cluster. It is built over a cluster of physical servers with the load balancer running on top of the

25. Gregor Roth, "Server Load Balancing Architectures, Part 1: Transport-Level Load Balancing: High Scalability and Availability for Server Farms," JavaWorld.com, 21 Oct 2008, retrieved 10 Feb 2009.

Linux operating system. The architecture of the server cluster is fully transparent to end users, and they interact with it as if they were using a single high-performance virtual server.

The physical servers and the load balancers may be interconnected by either a -speed local-area network (LAN) or by a geographically dispersed wide-area network (WAN). The load balancers dispatch requests to the different servers and make parallel services of the cluster appear as a virtual service using a single IP address. Scalability of the system is achieved by transparently adding or removing servers (often referred to as nodes) in the cluster. High availability is provided by detecting node or daemon failures and reconfiguring the system dynamically to prevent performance degradation.

The LVS employs a common three-tier architecture that consists of the load balancer, which acts as a front end to the cluster and balances (or distributes) requests from clients across a set of servers (sometimes called a bank) so that the client's perception is that all the services come from a single IP address. The second tier, the server cluster, is a group of servers running network services, such as HTTP, POP3, SMTP, FTP, or DNS. The third tier is shared storage, which provides a shared storage space for the servers so it is easy for them to use/reuse the same contents and provide the same services for each client request.

The load balancer itself is the primary entry point of server cluster systems. It can run Internet Protocol Virtual Server (IPVS), which implements transport-layer load balancing inside the Linux kernel. In network parlance, this is known as Layer-4 switching. IPVS running on a host can direct requests for Transmission Control Protocol (TCP) and User Datagram Protocol (UDP)[27] services to the physical servers, and this redirect makes it appear as if it were a virtual service running on a single IP address. When IPVS is used, all servers must provide the same services and content. The load balancer forwards a new client request to a server based on a scheduling algorithm and the current workload for each server. Regardless of which server is selected, the client should always see the same result. The number of nodes active in a server cluster can be changed according to the load that system encounters. When all servers are operating at capacity, more servers can be added to handle a larger workload. For most Internet services,

26. According to the online encyclopedia Wikipedia, "A computer cluster is a group of linked computers, working together closely so that in many respects they form a single computer." http://en.wikipedia.org/wiki/Computer_cluster, retrieved 10 Feb 2009.
27. TCP and UDP are common transport protocols defined in Request for Comment RFC 1122.

requests are not usually interrelated and can be run in parallel on multiple servers. As the number of nodes active in a server cluster increases, the performance of the entire cluster scales linearly.

Physically, the shared storage tier can be a database system, a network file system, or a distributed file system, or any combination thereof. The data that cluster nodes must update dynamically should be stored in database systems because whenever a node performs a read or write operation in a database system, the database system can guarantee the integrity and consistency of concurrent data access. Static data is stored using a network file system so data can be shared by all cluster nodes. Because scalability using a single network file system is limited, it can only support data operations from a limited number of nodes (typically four to eight).

For large-scale cluster systems, distributed file systems[28] can be used for shared storage and scaled according to system requirements. The load balancer, server cluster, and shared storage are usually interconnected by high-speed networks, such as Gigabit Ethernet, so the network will not become a bottleneck. Common methods used to construct the clusters are Domain Name System (DNS)-based load balancing and dispatcher-based load balancing. We will take a look at both methods in the following sections.

DNS-Based Load Balancing Clusters

DNS load balancing is probably the simplest method for building a network service cluster.[29] The Domain Name System is a hierarchical naming system for computers, services, or any resource used on the Internet. DNS associates various information with domain names assigned to such Internet participants. DNS translates Internet domain names (which are meaningful to humans) into the binary identifiers associated with networking equipment in order to locate and address Internet devices globally. This process is known as name resolution and is used to distribute requests to different IP addresses of cluster servers.

When a DNS request comes to the DNS server, it provides an available IP address based on scheduling strategies such as are used in a round-robin scheduler. Subsequent requests from clients using the same local caching name server are sent to the same server if they occur within the specified time-to-live (TTL) parameter set for name resolving. The original idea of

28. A Distributed File System (DFS) provides a means for administrators to create logical views of directories and files that can be used in a cluster regardless of where those files physically reside on the network.
29. http://www.linuxvirtualserver.org/why.html, retrieved 10 Feb 2009.

TTL was that it would specify a time span in seconds that, when expired, would cause the packet to be discarded. Because every router along the path from one device to another is required to subtract at least one count from the TTL field, the count is usually used to mean the number of router hops the packet is allowed to take before it must be discarded. Each router that receives a packet subtracts one from the current count in the TTL field. When the count reaches zero, the router detecting it will discard the packet and send an Internet Control Message Protocol (ICMP) message back to the originating host, letting it know that the packet was discarded.

The caching nature of clients in a hierarchical DNS system can easily lead to a dynamic load imbalance condition across cluster nodes. This makes it difficult for a node to operate efficiently at peak load capacity. Since it is impossible for the DNS system to guess the TTL value of a domain name, adjustments are often required to "tweak" the system in order for it to operate efficiently. If the TTL value is set too small, DNS traffic increases and the DNS server itself will bottleneck. If the TTL value is set too high, the dynamic load imbalance will only get worse. Even if the TTL value is set to zero, scheduling granularity is based on each host, so different user access patterns can also lead to dynamic load imbalances. This is because some people may pull lots of pages from a site, while others may just surf a few pages and leave. Load imbalance can also occur when a server node fails and the client request that was mapped to the failing IP address responds to the client, who often exacerbates the problem by clicking the reload or refresh button in the browser, sending yet another request to a dead node. Other nodes in the cluster pick up the load of the failed node, and the workload for all nodes increases accordingly.

Dispatcher-Based Load Balancing Clusters

A dispatcher performs intelligent load balancing by using server availability, capability, workload, and other user-defined criteria to determine where to send a TCP/IP request. The dispatcher component of a load balancer can distribute HTTP requests among nodes in a cluster. The dispatcher distributes the load among servers in a cluster so the services of nodes appear as a virtual service on a single IP address; end users interact as if it were a single server, without knowing anything about the back-end infrastructure. Compared to DNS-based load balancing, dispatcher load balancing can schedule requests at a much finer granularity (such as per connection) for better load balancing among servers. Failure can be masked when one or more nodes fail. Server management is becoming easy with the new tools available

today. An administrator can take put any number of nodes online or take them offline at any time without interrupting services to end users, which is exactly what is required for operating a cloud.

The Direct Routing Request Dispatching Technique

This request dispatching approach is similar to the one implemented in IBM's NetDispatcher. The virtual IP address is shared by real servers and the load balancer. The load balancer has an interface configured with the virtual IP address too, which is used to accept request packets, and it directly routes the packets to the chosen servers. All the real servers have their non-arp (address resolution protocol) alias interface configured with the virtual IP address or redirect packets destined for the virtual IP address to a local socket, so that real servers can process the packets locally. The load balancer and the real servers must have one of their interfaces physically linked by a hub or switch.

When a user accesses a virtual service provided by the server cluster, the packet destined for the virtual IP address (the IP address for the virtual server) arrives. The load balancer (Linux Director) examines the packet's destination address and port. If they are matched for a virtual service, a real server is chosen from the cluster by a scheduling algorithm, and the connection is added into the hash table which records connections. Then, the load balancer forwards the packet directly to the chosen server. When the incoming packet belongs to this connection and the chosen server can be found in the hash table, the packet is also routed directly to the server. When the server receives the forwarded packet, the server finds that the packet is for the address on its alias interface or for a local socket, so it processes the request and finally returns the result directly to the user. After a connection terminates or times out, the connection record is removed from the hash table. The load balancer simply changes the MAC address of the data frame to that of the chosen server and retransmits it on the LAN. This is why the load balancer and each server must be connected directly to one another by a single uninterrupted segment of a LAN.

Virtualization Applications

Application virtualization describes software technologies that improve portability, manageability, and compatibility of applications by encapsulating them from the underlying operating system on which they are executed.[30] A virtualized application is redirected at runtime to interface with

the virtual operating system and all related resources that are managed by it rather than an actual, physical implementation of that operating system.

Full application virtualization requires a virtualization layer.[31] The virtualization layer must be installed on a machine to intercept file and registry operations performed by a virtualized application, where it can transparently redirect those operations to a virtualized destination. The application that performs file operations never knows that it is not directly accessing a physical resource. Using this approach, applications can be made portable by redirecting their I/O tasks to a single physical file, and traditionally incompatible applications can be executed side by side.

Using application virtualization allows applications to run in nonnative environments. For example, Wine allows Linux users to run Microsoft Windows applications from the Linux platform. Virtualization also helps protect the operating system and isolate other applications from poorly written or buggy code. With application virtualization, physical resources can be shared so an implementation uses fewer resources than a separate virtual machine. Simplified operating system migrations are possible because administrators are able to maintain a standardized configuration in the underlying operating system and propagate that configuration across multiple servers in an organization, regardless of whatever applications may be used. In the next few sections, we will take a look at some of the more popular virtualization environments in use today.

VMWare

The VMware virtualization platform is built to virtualize hardware resources found on an x86-based computer (e.g., the CPU, RAM, hard disk, and network controller) to create a fully functional virtual machine that can run its own operating system and applications just like a standard computer. Each virtual machine is completely encapsulated in order to eliminate any potential conflicts. VMware virtualization works by inserting a thin layer of software directly on the computer hardware or on a host operating system. This layer is actually a monitor called a Hypervisor, and its task is to allocate hardware resources dynamically and transparently. Multiple operating systems can run concurrently on a single computer and share that computer's hardware. A virtual machine is completely compatible with all standard x86 operating systems, applications, and device drivers. It

30. http://en.wikipedia.org/wiki/Application_virtualization, retrieved 11 Feb 2009.
31. Amir Husain, "How to Build an Application Virtualization Framework," http://vdiworks.com/wp/?p=15, retrieved 11 Feb 2009.

is possible to run several operating systems and applications simultaneously on a single computer, and each operating system has access to the physical resources it needs on demand.

Readers interested in trying virtualization may consider using VMware ESXi (a free download from the official web site).[32] With ESXi, you can create virtual machines quickly and easily. A menu-driven startup and automatic configurations enable you to get virtual machines set up and running in minutes. You can even import a virtual appliance using the VMware Virtual Appliance Marketplace. For more information on VMware, the reader is encouraged to visit the official web site.

Xen

Xen is a unique open source technology[33] invented by a team led by Ian Pratt at the University of Cambridge. Xen was originally developed by the Systems Research Group at the University of Cambridge Computer Laboratory as part of the XenoServers project, funded by the UK-EPSRC. XenoServers aimed to provide a public infrastructure for global distributed computing. Xen plays a key part in that, allowing one to efficiently partition a single machine to enable multiple independent clients to run their operating systems and applications in an environment. This environment provides protection, resource isolation, and accounting. The project web page contains further information as well as pointers to papers and technical reports.[34]

Using Xen server virtualization, the Xen Hypervisor is installed directly on the host hardware and exists as a thin layer between the hardware and the operating system. This abstraction layer allows the host device to run one or more virtual servers. It isolates hardware from the operating system and its applications. Xen is licensed under the GNU General Public License (GPL2) and is available at no charge in both source and object format. According to the official web site, "Xen is, and always will be, open sourced, uniting the industry and the Xen ecosystem to speed the adoption of virtualization in the enterprise."

The Xen Hypervisor supports a wide range of guest operating systems including Windows, Linux, Solaris, and various versions of the BSD operating systems. The Xen Hypervisor has an exceptionally lean footprint. The Xen Hypervisor offers a smaller code base, greater security, and up to 10

32. http://www.vmware.com.
33. http://www.xen.org.
34. http://www.cl.cam.ac.uk/xeno, retrieved 11 Feb 2009.

times less overhead than alternative virtualization approaches. That means that it has extremely low overhead and near-native performance for guests. Xen reuses existing device drivers (both closed and open source) from Linux, making device management easy. Xen is robust to device driver failure and protects both guests and the Hypervisor from faulty or malicious drivers.

Virtual device monitors (which are also known as hypervisors) are often used on mainframes and large servers seen in data center architectures. Increasingly, they are being used by Internet service providers (ISPs) to provide virtual dedicated servers to their customers. Xen support for virtual-machine live migration from one host to another allows workload balancing and avoids system downtime. Some of the main advantages of Xen server virtualization are

- Consolidation and increased utilization
- The ability to rapidly provision and start a virtual machine
- Better ability to dynamically respond to faults by rebooting a virtual machine or moving a virtual machine to a different hardware platform
- The ability to securely separate virtual operating systems on the same platform
- The ability to support legacy software as well as new operating system instances on the same computer

Xen may also be used on personal computers configured in a dual-boot configuration (e.g., those that run Linux but also have Windows installed). Traditionally, such systems provided the user the option of either running Windows or Linux, but with Xen it is possible to start Windows and allow it to run from in a separate Window on the Linux desktop, enabling the user to run applications from both systems simultaneously.

For operating system development tasks, virtualization has a significant additional benefit—running the new system as a guest avoids any need to reboot the computer whenever a bug is encountered. This protected or insulated environment is known as a "sandbox," and such sandboxed guest systems are useful in computer security research and development. In order to study the effects of malware, viruses, and worms without compromising the host system, developers often prefer to use a sandbox. Hardware appliance vendors increasingly have begun to ship

their products preconfigured with several guest systems. This allows them to deliver complex solutions that are able to execute various software applications running on different operating systems.

Xen touts a para-virtualization technology that is widely acknowledged as the fastest and most secure virtualization software in the industry. Para-virtualization takes full advantage of the latest Intel and AMD hardware virtualization advancements and has fundamentally altered the way virtualization technology is built. Virtual servers and the Hypervisor cooperate to achieve very high performance for I/O, CPU, and memory virtualization.

According to the Xen User Manual,[35] the Xen system has multiple layers, the lowest and most privileged of which is Xen itself. Xen can host multiple guest operating systems. Each operating system is run within a secure virtual machine environment known as a domain. In order to make effective use of the available physical CPUs, such domains are scheduled by Xen. Each guest operating system is responsible for managing its own applications. This management includes scheduling each application within the time allotted by Xen to the virtual machine. The primary domain, domain 0, is created automatically when the system boots, and it has special management privileges. Domain 0 builds other domains and manages their virtual devices. Domain 0 also performs administrative tasks such as suspending, resuming, and migrating other virtual machines. Within domain 0, a process called *xend* is responsible for managing virtual machines and providing access to their consoles.

3.9 Chapter Summary

In this chapter we discussed what it takes to build a cloud network, evolution from the managed service provider model to cloud computing and SaaS and from single-purpose architectures to multipurpose architectures, the concept and design of data center virtualization, the role and importance of collaboration, service-oriented architectures as an intermediary step and the basic approach to data center-based SOAs, and the role of open source software in data centers and where and how it is used in cloud architecture. Cloud computing provides an end-to-end, unified solution that maximizes the ability to address the performance, scalability, virtualization, and collaboration requirements being driven by today's global business challenges and opportunities. It should be clear that a properly designed and

35. http://tx.downloads.xensource.com/downloads/docs/user/user.html, retrieved 11 Feb 2009.

implemented cloud infrastructure provides the benefit of substantially lowering the total cost of ownership over the traditional hosting environment though the use of virtualization and the use of open source software. Cloud infrastructure maximizes the potential for creating value through collaboration. In future chapters we will discuss the ability of cloud computing to provide a solution to current challenges in presence and identity while enhancing security and privacy. First, however, we will give you a chance to see for yourself the value and process in implementing and using cloud computing. In the next chapter, we will give guide you through a practicum on the how you can build a virtualized computing infrastructure using open source software.