NEI / 25 Dan Road Canton, MA / 02021-2817 / telephone: 781 332 1000 / fax: 781 770 2000 **www.nei.com**

A Virtual Appliance Model for Deploying Applications in the Cloud

Introduction

As cloud computing and Software as a Service (SaaS) become viable options for businesses worldwide, enterprise software vendors will need a scalable platform solution for deploying cloud-ready applications. Gartner estimates that the cloud computing market will grow to $126 billion by 2012 and up to $150 billion in 2013. This growth will stress many software developers who seek a reliable platform technology to help them succeed. Appliance platforms offer software vendors a great way to deploy their applications as either physical, virtual or hybrid solutions for private, public and mixed environments. The appliance architecture offered here is easily transferable, manageable and scalable. It is based on a software stack (i.e., image) that is easily tuned for performance, secured, deployed and managed remotely.

Appliance Architecture

NEI's appliance architecture (Figure 1) centers on the software vendor's application, with which supporting applications and the Operating System (OS) are integrated, hardened and tuned for operational efficiency. This architecture accommodates critical element management software with a browser-based user interface for ease of administration and remote management. The software stack is seamlessly integrated into an "image" and ready-made for deployment.

The resulting image can either be installed onto a disk for a hardware appliance or virtualized for a software appliance. Thus, the appliance model provides the flexibility to deliver software as an on-premise hardware appliance, a software appliance for the virtual data center or cloud, or as a hybrid solution that combines on-premise storage with cloud computing. By expanding their solution set to include cloud computing, software vendors can extend a compelling value proposition to their enterprise customers by simplifying the deployment and support matrix.

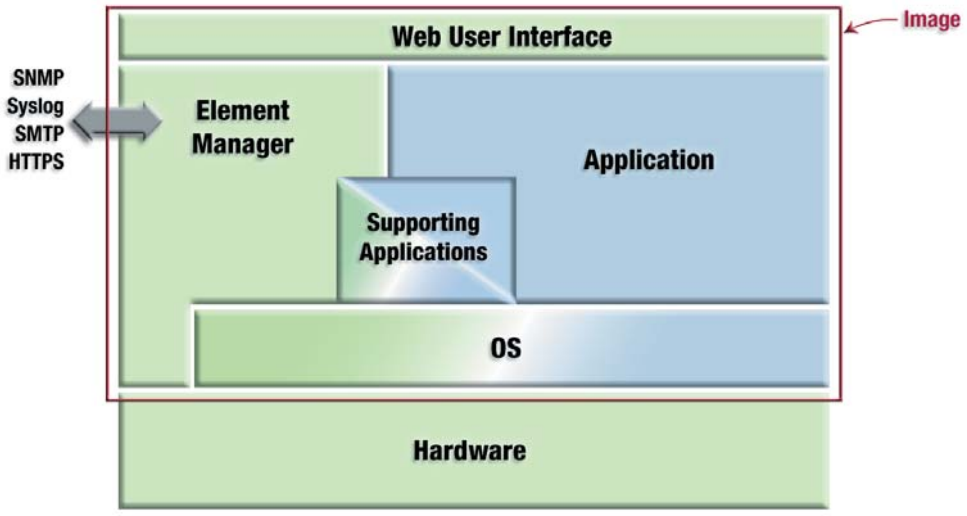

Figure 1

Simplified Appliance Architecture

Five key attributes of this architecture make it ideal for cloud-based applications. All are synonymous with those that make hardware appliances appealing:

- Purpose-built for the application, with "just enough" OS to maximize the performance and minimize the footprint
- Applications are installed and preconfigured for faster deployment times
- Hardened OS ensures greater security
- Simplified management ensures easy administration and maintenance
- Integrated life cycle management makes updates easier

Why Use a Virtual Appliance?

Virtual data centers and cloud computing models share and leverage resources, software and information delivered via the Internet to servers, computers and other connected devices. With the use of cloud computing, users can avoid the fixed capital expenditures associated with hardware, software and services by paying a provider for only the resources or services consumed.

Virtual appliances offer a natural progression – an enhanced model – for software delivery and maintenance for both application providers *and* their customers. As a result, it's an effective way for enterprise software vendors to blend and expand their physical and virtual deployments as private and public clouds become mainstream over time.

It's important to note that the principle difference between a software and hardware appliance deployment is resource sharing versus dedicated hardware. A hardware appliance uses fixed processing, memory and storage capacity – all tuned to the solution's requirements. A virtual appliance by contrast operates in an environment where processing, memory and storage are shared resources. This can introduce I/O latency and other performance-related issues if the virtual data center or cloud structure does not have timely access to necessary compute resources. Designing and building an appliance solution that is versatile enough for both physical and virtual appliances can easily be achieved by analyzing the various options and making choices based on the application and market needs.

A hybrid solution where on-site storage is combined with cloud computing and management tools can provide a best-of-breed solution.

Just Enough Operating System Design

JeOS is the abbreviation for the concept of Just Enough Operating System. JeOS is not a generic (i.e., one-size-fits-all) operating system. Rather, it refers to a customized operating system that precisely fits the needs of a particular application. The application's OS requirements can be determined manually, or by using an analytical tool.

JeOS-optimized devices contain only the pieces of an operating system (often Linux) required to support a particular application. This makes the appliance far more efficient, smaller in footprint, more secure and higher performing than one running with a full general-purpose OS.

In a virtual environment, because resources are shared, JeOS becomes even more important. Every 100MB removed from the OS means that fewer shared resources (e.g., CPU and memory) are consumed in the virtual or cloud environment. Since resource consumption drives both capacity planning and charge-back costs, even small footprint savings are amplified over time. In contrast, the trimming of OS packages in hardware decreases in value as the ratio of time to footprint savings increases. Each additional 100MB removed from a commercial distribution becomes more time-consuming with less value because the resources are dedicated.

Customize the OS

There are two distinct ways to build a customized OS. One can either start large and cut down or start small and build up. The cut-down approach takes a commercial OS and strips out all unneeded software. This can be an arduous, time-consuming task that seldom yields the smallest OS possible and the process generally overcompensates for dependencies. Conversely, the build-up approach begins with a minimum OS kernel and adds only the software required to run the application. Contact NEI to create a best-fit OS instance and learn more about our Secure OS Services SOS that make appliances more efficient, reliable and manageable.

Many Linux distributions now offer a JeOS package, and there are also several commercially available packaging tools that assist with resolving the dependencies while minimizing the footprint. The result is a hardened OS that is 65% to 75% smaller than a typical commercial distribution and contains only what is required for the application to run.

Provisioning and Management

Creating an automated startup process is critical to the operation of a virtual appliance deployed in the cloud. On Linux-based virtual machines for example, the startup process is typically driven by the init scripts. Init scripts are a good way to begin the configuration and startup process for virtual appliances, but they typically need access to external configuration data (i.e., externally stored and accessed via the virtual machine).

The simplest way to handle this situation is to start the provisioning process by bringing up the virtual machine's network interfaces and retrieving all business-critical information from a known location. The known location is typically a hardcoded address or based on a parameter passed to the virtual machine at time of boot. For example, both Amazon's EC2 and VMware provide facilities for passing-in boot time parameters. Once the virtual machine is able to contact an outside resource, it begins retrieving the information needed for proper configuration and its eventual role in the network cluster. Once all parameters are retrieved and applied, the system notifies the network of its state and begins functioning as a full node in the environment.

Likewise, maintenance may be done by making periodic requests to a designated update server. When an update is required, the system simply retrieves and applies it. There are several good system updaters available for virtual machines, with several provided via OS and appliance framework vendors. Alternatively, if full system images are preferred, a virtual machine instance should terminate gracefully and be replaced by an upgraded version.

Data Storage and Backup

Persistent storage is critical to running stateless solutions. Persistent storage may include either file systems or databases, but the key consideration for cloud computing is that the storage remain separate from these nodes. For example, in Amazon's EC2 cloud offering, stateless nodes use persistent data volumes rather than local storage for files and databases. Amazon's persistent volumes may be attached to nodes in the cloud, but the data in them remains after the node has terminated. This allows the same data to be used across many instances to minimize the risk of data loss.

As an additional safeguard, persistent storage should be periodically backed up. Most cloud providers offer robust backup solutions. Amazon, for example, tenders snapshots of persistent volumes to its S3 storage service. Designing an application to be stateless, whenever possible, allows it to better handle failures, expansion and demand more gracefully. That is, without a fixed state to inherit, a replacement resource may be spun-up more quickly to resume the failed node's functions.

Advanced Considerations in Architecting a Virtual Appliance

Before deploying an appliance to run properly for the cloud environment, the application may require minor adaptations. In the cloud computing model, compute and storage resources are best designed to scale easily. Unlike physical hardware appliances which are purpose-built for peak load and maximum storage, turnkey appliances that operate in the cloud must seamlessly balance loads and scale up and down in response to demand.

Breaking down the appliance stack into simpler, loosely coupled nodes is essential to taking advantage of virtualization's scaling capability. Hardware appliances typically separate processing and storage into individual partitions. Virtual appliances can further extend partitioning by separating these components and allowing each to be scaled independently. Each component must be accessible and served by multiple nodes simultaneously, such that as additional cloud resources become available, the application can utilize them. This approach helps produce a leaner appliance that more quickly scales to meet demand.

Another consideration in designing the application involves the many uncertainties of cloud computing. Since shared resources can put data at risk, some application developers are reluctant to store sensitive data in the cloud. A hybrid approach, where on-site (private) storage is combined with cloud computing and/or management tools,can provide a blended and highly tailored solution.

Data should be generally placed in persistent storage on a separate node or (better yet) multiple separate nodes. As a result, any failure of a single node will not cause a data loss and any node's data may be revived by other available nodes. This application architecture is akin to the concept of RAID for storage devices, where data is written in multiple separate locations in order to prevent data loss. Modular virtual appliances running in the cloud deliver RAID-like redundancy and reliability to developers and service providers alike.

The final consideration in designing appliances for the cloud is to ensure that there are reliable, automated processes for starting, stopping and resuming various operations of the application. The scaling and resiliency capabilities of the cloud may only be leveraged with application components that can come online in an automated fashion. It is always best to avoid configuration and startup processes that require human intervention or pre-configuration.

Most cloud providers have basic tools for automatically spinning-up a virtual machine, but in some cases the configuration of the virtual machine's software components are not accommodated. The startup process must therefore provide the means to automatically determine the function that the node should next perform, then configure any other software components and seamlessly insert itself into a functioning cluster of other virtual machines. Likewise, nodes must be able to extract themselves and exit from a cluster gracefully. As noted earlier, a stateless node where all data is persisted externally makes both the startup and shutdown processes easier.

Conclusion

Cloud computing is a rapidly growing and an increasingly viable option for deploying enterprise application software solutions. NEI's appliance architecture provides software vendors with the flexibility to develop their applications and deploy with physical, virtual or hybrid solutions. For more information about our hardware and virtual appliance solutions, contact NEI. We are the industry's leading provider of application platforms, appliances and deployment services for OEMs, ISVs and service providers.

About NEI

NEI is a leading provider of application platforms, appliances and services for software developers, OEMs and service providers worldwide. NEI enables customers to more effectively deploy, manage and support application platforms and appliances using its comprehensive capabilities, including solution design, integration control, global logistics, smart services, technical support and maintenance. Founded in 1997, NEI is headquartered in Canton, Massachusetts and trades on the NASDAQ exchange under the symbol NENG. For more information about NEI's products and services, visit **www.nei.com**.

Contact Us

For more information on the full range of appliance-related hardware, software and services NEI can provide to your company, please contact us by telephone +1 (781) 332-1295 or by email at **sales@nei.com**.

P/N: 595-1126-00_081810