Reinventing virtual appliances

X. Jin
R. Willenborg
Y. Zhao
C. Sun
L. He
Z. Chen
Y. Chen
Q. Wang

Today, data centers are being built with more entities on a larger scale in order to satisfy the rapid growth of services and the demand for efficient service delivery and management. These data centers are facing scalability, manageability, and consumability challenges that require breakthrough technology for design and management. This paper extends previous work that focused on using virtual appliances for service management, by describing the following areas: 1) rapid provisioning, which simplifies and accelerates the deployment and activation of virtual appliances by checking parameter dependencies and coordinating execution sequences; 2) selective presentation, which provides flexible interfaces to present meaningful information through customized metrics; 3) simplified operation, which enables different operation granularities by coordinating the operation sequences across virtual appliances; and 4) automated consolidation, which enables automated service performance management by automatically monitoring and optimizing resource allocations. In addition, we describe the design and prototype of an extendable management framework, virtualization integrator (VSI), which enables an appliance builder to easily fulfill the key features of virtual appliances and coordinate management function across components located in multiple virtual appliances. By reinventing virtual appliances with built-in management capability, data center management is radically simplified and automated. The results, improved server utilization and simplified system manipulation, demonstrate the potential for virtual appliances to be fundamental building blocks for large-scale data centers and potentially for cloud-based new enterprise data centers.

Introduction

Information technology (IT) has become a core component of modern businesses in a wide variety of companies and is increasingly leveraged to deliver their services.[1] These services, residing in data centers, are becoming more and more complex in order to provide rich and advanced functions. To support the rapid growth of services and the demand for efficient service delivery and management, data centers are being built to contain more and more entities and are growing to ever larger scales. With increasing complexity of deployment due to environment heterogeneity and service dependency, these data centers are faced with scalability, manageability, and consumability[2] challenges that require breakthrough technology for data center design and management.

Virtual appliances offer a new paradigm for simplified and automated data center management. Virtualization [1] is widely used to consolidate servers in order to improve server utilization and to reduce operational space and power expenditures [2]. It can also enable services such as co-located hosting facilities, distributed Web services, secure computing platforms, and service mobility [3]. With self-contained computing modules with

[1] *Services* denote the application functionality provided by the underlying infrastructure, middleware, and applications. Service deployment is the process of making the service ready for use.

[2] *Consumability* refers to a set of attributes that are particularly important to the client experience. By increasing the consumability of the product, we increase the value of that software to the client.

0018-8646/09/$5.00 © 2009 IBM

specific functions and limited configuration abilities, appliances are usually easier to use than similar solutions created from general-purpose hardware and software, because in the appliance the hardware and all required software come together as a single-purpose solution. Virtual appliances [4] are a set of virtual machines including optimized operating systems, prebuilt, preconfigured, ready-to-run applications, and embedded appliance-specific components. By combining the advantage of both appliances and virtualization technology, virtual appliance packages are preconfigured and virtualization-ready solutions that eliminate the need for manual configuration and relieve the management burden. Thus, virtual appliances are emerging as a groundbreaking technology to deal with the complexities of large-scale data centers.

In this paper, we first examine how virtual appliances, together with some key technology trends such as service-oriented architecture (SOA)[3] [5] and ensembles[4], can provide fundamental changes in traditional enterprise data centers and form a new architecture for large-scale data centers [6]. Virtual appliances that include self-management capabilities can be used to easily deliver and deploy services. In addition, the services provided by virtual appliances in the data centers can be conveniently and uniformly organized and managed regardless of the underlying physical hardware servers and operating systems. Therefore, these services can compose a loosely coupled, self-managed system and make the data centers very scalable.

In order to support their new role in future data centers, virtual appliances must advance to deliver additional capabilities. This paper explores the following areas: 1) rapid provisioning, which simplifies and accelerates the deployment and activation of virtual appliances by checking parameter dependencies and coordinating execution sequences; 2) selective presentation, which provides flexible interfaces to present meaningful, customized information; 3) simplified operation, which enables different operation granularities by coordinating the operation sequences within virtual appliances; and 4) automated consolidation, which enables automated service performance management by automatically monitoring and optimizing resource allocations.

Delivering virtual appliances with these new capabilities presents new challenges to the appliance developer. To meet these requirements, we designed and prototyped an extendable framework, virtualization integrator (VSI), which enables the appliance builder to easily fulfill the key features of virtual appliances. (Here, *appliance builder* refers to a role whose responsibility is to integrate applications into virtual appliance templates with some tools in the creation and assembling phase.) By embedding VSI inside the appliances, management function can be coordinated across components located in multiple virtual machines. VSI is also able to work with the management services provided by the data centers to fulfill the whole life-cycle management of the business services provided by the virtual appliances.

The remainder of this paper is organized as follows. First, we briefly discuss the requirements of large-scale data centers and propose a new architecture that leverages virtual appliances to address the challenges. We then describe the four key features of virtual appliances to support service deployment and management. Next, we discuss the capabilities of VSI to fulfill the key features of virtual appliances and give an implementation of a sample capability. Related work and the differences compared to our work are discussed, and finally, we conclude the paper.

Redesigning data centers

As data centers grow to larger scales in order to meet the demands of new services, the challenge to scalability and manageability has been growing. In addition, changing business demands with increased pressure on efficiency, revenue, and time to market mean that services need to be changed quickly in order to adapt to new applications. To support the growth and flexibility requirements, we need to rethink and redesign the architecture of data centers by utilizing successful programs and design principles from the evolution of the IT industry.

Designing principles

First, a loosely coupled architecture is recommended for the next generation of large scale, highly dynamic, distributed applications and services [7]. Compared with the tightly coupled architectures, loosely coupled architectures are much easier to maintain and reuse because the integration interfaces are developed with minimal assumptions between the interrelated parties, thus reducing the risk that the change in one party will force some variation in another party.

Next, the intelligence of the end system must support self-organization and complement the loosely coupled architecture. Compared with centralized intelligence, end-system intelligence or decentralized intelligence brings the management decision to the end system and eliminates a potential single-point bottleneck in large-scale systems. Putting the intelligence into the end systems [8] is one of the main reasons for the success of the Internet, the most

[3]SOA is a rapidly emerging new business model to deliver application functionality through a subscription model over the Internet. The customer does not take ownership of the software, but instead rents a total solution that is delivered remotely.
[4]An ensemble is a pool of like systems that is manageable as a single system. It integrates compatible networked systems, virtualization, and management functions and scales from few to many servers while having management complexity and cost like those of a single system.

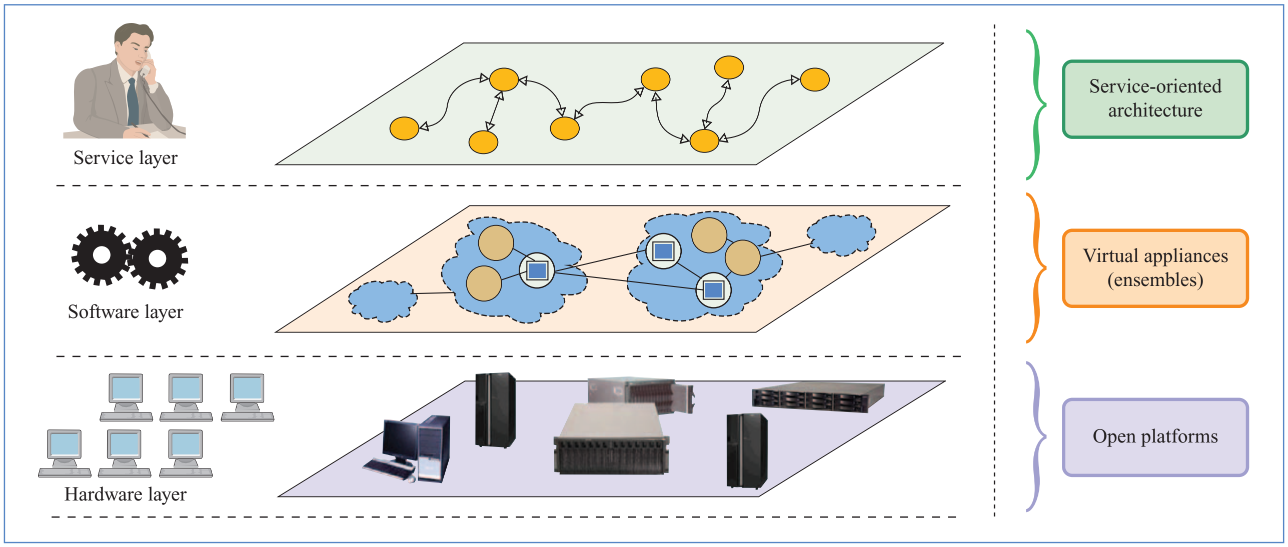

Figure 1

Simplified architecture for future data centers.

robust distributed system with massive entities running within it.

New data center architecture

A new architecture with three conceptual layers for future data centers is shown in **Figure 1**. To support the scalability and flexibility requirements of new data centers, loosely coupled architecture and decentralized intelligence are the key points for the design of each layer. The service layer utilizes SOA to provide various business services. The software layer is the underlying software systems that provide functions for supporting business services. The hardware layer provides the aggregated computing resources.

Today, SOA is the most widely adopted loosely coupled architecture in enterprises and it supports standards-based and protocol-independent distributed computing [9]. An increasing number of enterprise services are provided through SOA to support service orientation and provide business alignment, flexibility, reusability, integration, and standardization.

For the hardware layer, physical resources are being aggregated as collections, referred to as *ensembles*, to provide computing capabilities with highly autonomic management within them, thus presenting a simplified resource view to the outside world as if the ensemble were a single system [9]. Because virtualization decouples the physical platforms from the software systems that rely on those physical platforms, it is a powerful catalyst for consolidating servers into these ensembles in order to improve resource utilization and to reduce operational space and power expenditures. Compared with the traditional approach where the software and hardware layers are tightly coupled, today the ensemble-like platforms are designed to provide virtualized hardware resource interfaces for running the software systems on top of them in a loosely coupled way.

For the software layer, virtual appliances are the proposed candidates to be used as the building blocks for the software layer in future data centers. Virtual appliances can be extended to support a loosely coupled architecture by encapsulating the functional components in each service into a standard reusable software unit. The individual virtual appliances are integrated to compose higher-level solutions without dependencies on external software systems. By embedding end-system intelligence directly into the integrated virtual appliances, the dependencies on specific external management systems are removed, allowing services to be modified quickly to adapt to changing demands.

In summary, achieving a simple, loosely coupled software layer with virtual appliances reduces the complexity in large-scale data centers and supports SOA and aggregated resource layers. In the following section, we introduce our definition of virtual appliance in more detail.

Redefining virtual appliances

An IT service can be defined as several software components that provide useful functions [10]. Traditionally, a virtual appliance is defined as one minimalist virtual machine image with one or more

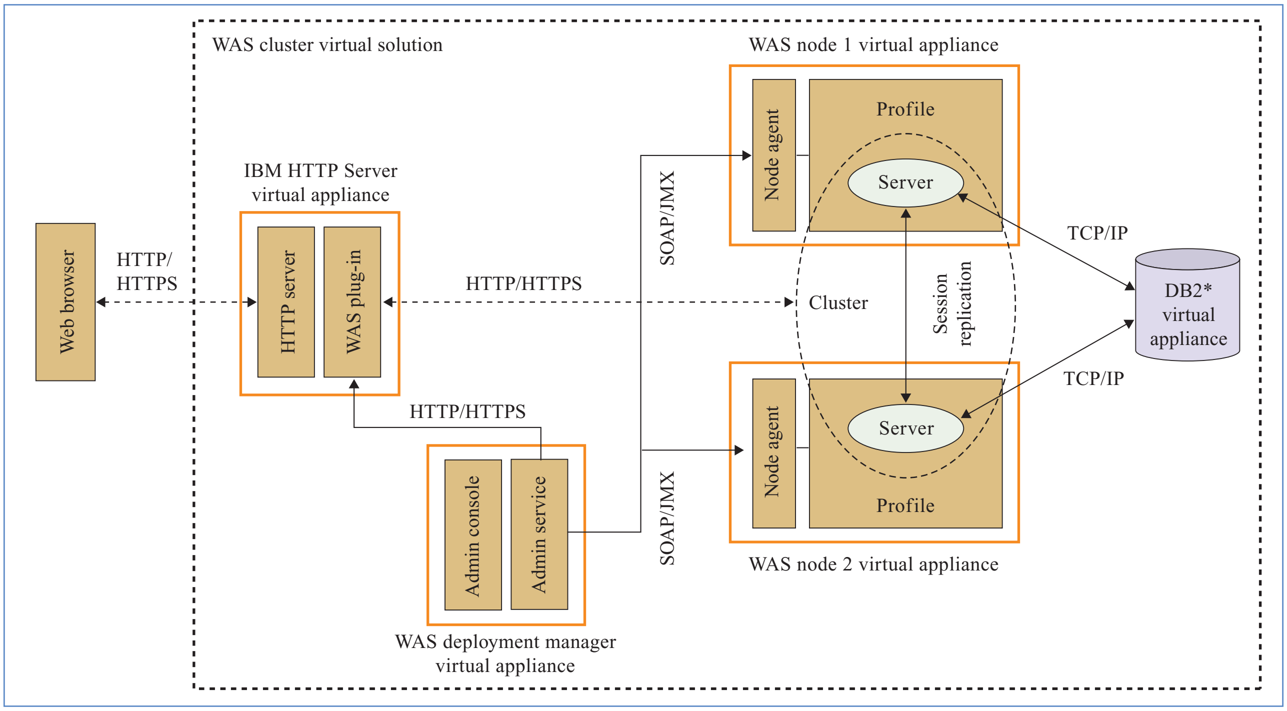

Figure 2

Topology of a WebSphere* Application Server (WAS) cluster. (SOAP: Simple Object Access Protocol; JMX**: Java** Management Extensions.)

specific applications and an optimized operating system. However, most of today's services, especially those for enterprise use, are not standalone systems. They often consist of a large number of components with each offering and require the functions of other components. Therefore, one service will likely span the capabilities of multiple virtual appliances, and these appliances need to be integrated with each other for coordinated service deployment and management. Typically, each software component is deployed on a different physical machine to provide good isolation. For services provided by virtual appliances, each software component can be mapped to one virtual appliance and does not require a dedicated physical machine.

To meet the above requirements, we extend the traditional definition of virtual appliances by embedding appliance intelligence inside virtual machines and providing integrated features inside multiple virtual machines to fulfill service management capabilities. We identified the following key features: rapid provisioning, selective presentation, simplified operation, and automated consolidation, with which services built with virtual appliances can be easily deployed, activated, operated, and monitored.

In the remainder of this section, we discuss these four features in detail. We mainly discuss two roles related to the service: 1) the independent software vendor (ISV), or enterprise application developer, who is responsible for service creation and delivery; and 2) the data center administrator, who deploys and manages the service in the data centers.

Throughout this paper, in order to illustrate the key features of virtual appliances, we use an IBM WebSphere Application Server (WAS) cluster, together with a database server and a front Web server to support J2EE** applications. As shown in **Figure 2**, the WAS cluster consists of three nodes: the management node and two managed nodes. The database server is IBM DB2, and IBM HTTP Server (IHS) is used as the Web server.[5] We also apply any required patches [11].

Rapid provisioning

As IT services become more powerful and complex, service deployment becomes more difficult and expensive. Service deployment, the process of making a service ready

[5]IBM HTTP Server, or IHS, is an enhanced Web server based on the Apache Software Foundation Apache HTTP Server. It can be remotely administered and configured using the WebSphere administrative console.

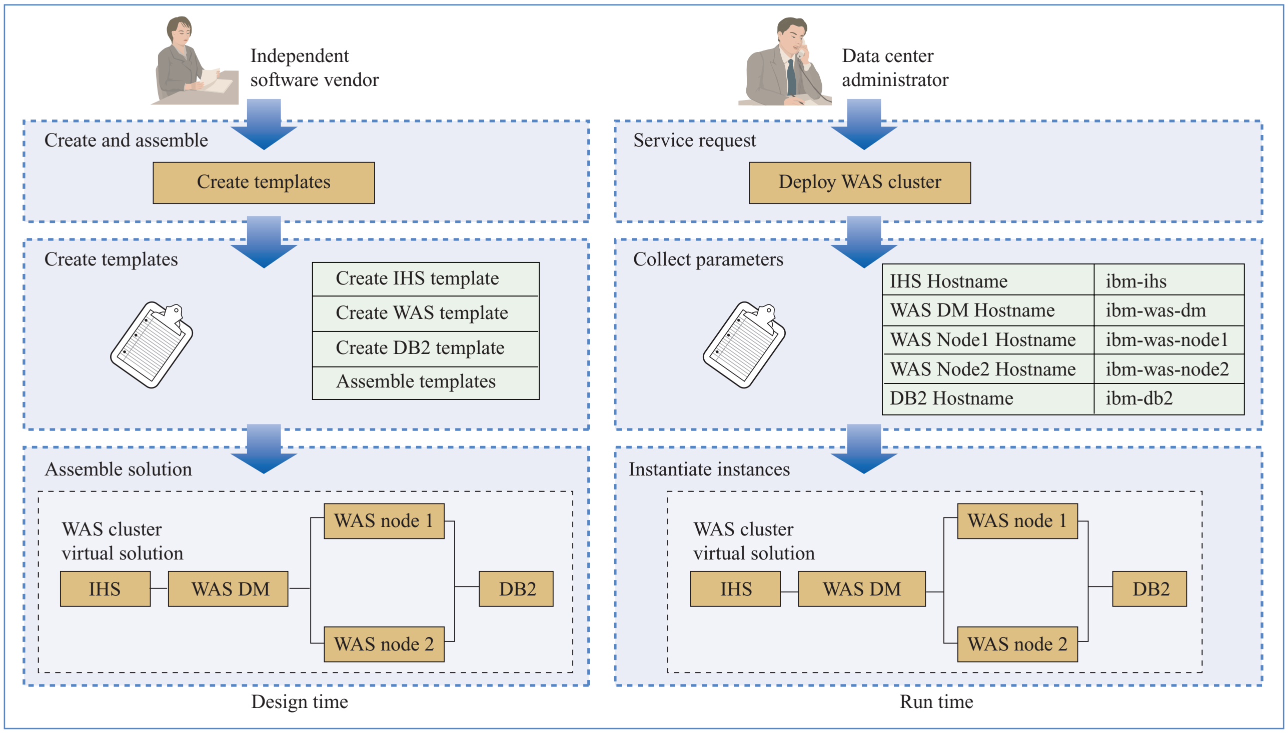

Figure 3

IBM WebSphere Application Server (WAS) rapid provisioning with virtual appliances. (IHS: IBM HTTP Server; DM: deployment manager.)

for use, often includes deploying multiple interrelated software components into heterogeneous environments. This can be a complex and labor-intensive process entangled by dependencies and execution constraints because enterprise services often include many complicated and advanced functions and are deployed over heterogeneous and distributed environments. After receiving an application, to make it available for use, the data center administrator typically must ensure hardware and software compatibility, install prerequisite software, and check configurations. Often considerable effort is spent on testing and certification in the enterprise environment. To move to a different environment, the same process must be repeated. Thus, it can be argued that a service prepared by an ISV is hardly complete. In fact, the data center administrator is involved in installation and configuration to get the service up and running.

In response to the above challenges, rapid provisioning becomes a core function of virtual appliances in order to simplify and automate service deployment and eliminate the complexities in the service installation process. The simplifications are possible because the service is delivered as a set of preconfigured, prepackaged virtual appliances without application or platform dependencies. To deploy the service with rapid provisioning, the data center administrator is only expected to have a list of machines, a link to the template images of the service, and some parameters for customization. Virtual appliances encapsulate the entire service environment, allowing the ISV to resolve execution constraints and dependencies prior to delivery. In other words, the virtual appliance applications are preinstalled and validated on the virtual machine on which they run. As a result, the data center administrator is not confronted by the challenges of configuring and validating each application in a custom environment. For post-provisioning customizations, the embedded appliance-specific components examine the software configurations in each virtual appliance and automatically activate them in a coordinated manner.

For the WAS cluster sample, the traditional deployment approach requires installing IHS, installing DB2 on two separate machines, installing and setting WAS deployment manager and managed nodes in other machines, configuring them to bridge the DB2 client in WAS and the DB2 server with its host name, username, and password, and finally starting the cluster. As shown in **Figure 3**, with virtual appliances, the ISV installs all the software into template images and designs the topology, activation sequence, and parameter dependencies into

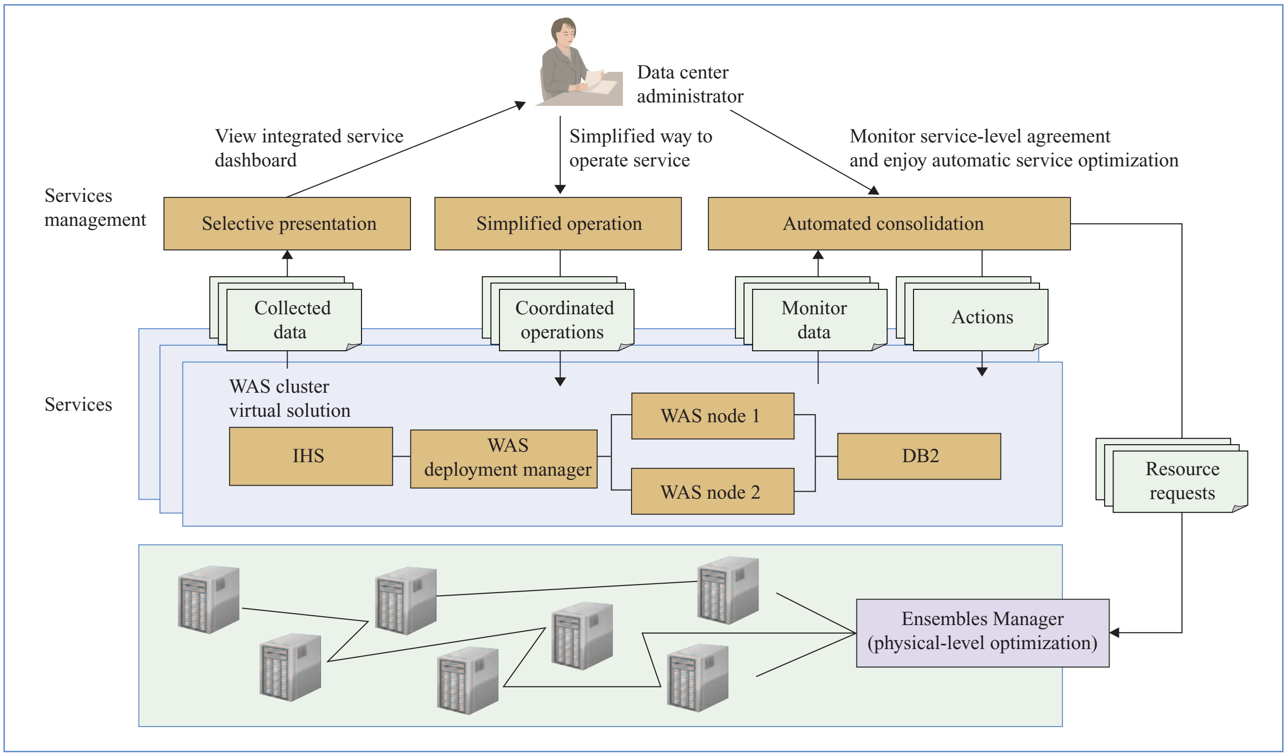

Figure 4

Selective presentation, simplified operation, and automated consolidation with virtual appliances.

the service. Rapid provisioning will then automatically activate the WAS, DB2, and IHS images and start the service on request from the data center administrator. This greatly improves service deployment efficiency.

Selective presentation

Another central feature of virtual appliances is data gathering and reporting. To enable the data center administrator to monitor services from a higher-level point of view, the data provided should be distilled so it is simple to grasp while being rich enough to help the administrator make decisions. Selective presentation facilitates this by providing the appliance data reporting capabilities to display the important and meaningful data to the administrator.

Traditionally, the data center administrator has limited application-specific knowledge and, therefore, has difficulty organizing the data for a running application. This restricts the data traditionally used to common, application-independent, dispersed information such as standard CPU (central processing unit) and memory usage. Using virtual appliances, each application is encapsulated in an ISV-created environment where the ISV can obtain a plethora of data, distill it within the appliance, and provide meaningful service-level metrics.

Our proposed virtual appliances implement the selective presentation capabilities by gathering information from sensors located in multiple virtual appliances and by processing the raw data to a higher level of meaningful and easily understood information according to predefined rules.

For example, during the selective presentation for the sample WAS cluster service shown in **Figure 4**, the response time of an http request, the throughput of WAS, and the CPU utilization of the WAS virtual appliance are collected at runtime. The information is further processed internally to get service-level identifiers such as service health, workload balance, and resource status. This distilled data provides a more meaningful view to the data center administrator interested in the service status. Another example is to use the selective presentation function to provide the total CPU usage to the data center administrator, instead of the individual CPU usage of each machine, which would not be enough information for calculating the resource expense of this service. Leveraging the capability of virtual appliances, the ISV and the data center administrator's control over data presentation is maximized for the most insightful data.

Simplified operation

In the quickly changing global business environment, a service management platform must support continuous system operations, such as adding, removing, or updating services depending on the business condition and user needs. Moreover, from the data center administrator's perspective, *service management* refers to high-level usage of IT systems, and it is not in the interest of the data center administrator to be extensively involved in the underlying detailed operation commands of his systems.

We propose simplified operation to allow the data center administrator to simply and rapidly manage services by dynamically operating resources to optimally meet business requirements. This provides the administrator with tremendous flexibility and simplicity. For example, the administrator controls the on and off states of the service by issuing one operation command through a simplified interface to the appliance-specific components. The appliance-specific components control the starting and stopping of the appliances, as well as the software inside, in order to make the service available.

The key enablement technology of simplified operation combines a series of operations into one single operation, for example, one operation to start or stop a service and one operation to change the password of one component. Therefore, the sequence dependencies between the appliances are designed into the service and resolve it in the runtime. As shown in Figure 4, to start the WAS cluster service, the starting sequence of the service appliances is predefined to first start the DB2 server, then the WAS manager, next the WAS managed nodes, and finally the IHS. The virtual appliances combine the five operations and resolve the dependencies between these operations, allowing the service to provide a simple "start WAS topology" interface. The start process automatically handles the five operations instead of requiring the data center administrator to understand the different software components and their dependencies.

Another example is to use the simplified operation to give the data center administrator the capability to manage passwords. A typical service contains applications deployed across multiple appliances, each of which requires user authentication. To reduce the cost of password management, virtual appliances are designed to monitor and ensure password consistency across software resources. For example, a DB2 server and its clients must have matching passwords for remote data access. In this case, the passwords of the server and the client must remain consistent in order to ensure proper operation. When users make password changes to one of the resources, the virtual appliances provide change guidelines, alerts, or automatic password change capability.

Automated consolidation

For the data center administrator, an important task for system management is to take the advantages of automated consolidation offered by virtualization to measure the service quality and improve the efficiency of using the underlying computing resources, such as CPU, memory, disk I/O, and network. Virtual appliances need to include these service performance management functions as well as resource management functions in order to monitor and optimize the underlying resource utilization and to request the resource allocations and adjustments from the manager of the aggregated physical resources.

We designed automated consolidation functions to allow the data center administrator to monitor key service-level agreement indicators of the service, for example, response time and throughput. Once key metrics are measured, mechanisms need to be devised to enhance the efficiency of providing these services. The automated consolidation function also includes the capabilities of allocating resources, dynamic load balancing, and live migration of a running virtual appliance from one physical host to another.

In traditional approaches, data center administrators do not have much knowledge of the service behavior and do not know how to optimize it. Using virtual appliances, the optimization problem is simplified because it is divided into service-level optimization, which is to be done by the ISV, and physical-level optimization, which is to be done by the manager of the aggregated resources. This approach decouples the variability of service behavior and its resource requirements from the complexity of physical-level optimization. In our proposed framework for automated consolidation in Figure 4, the service-level optimizer takes the monitored data (e.g., changing workload) as input and outputs the optimized resource requirements of each virtual appliance through analyzing and optimization, such as prebuilt workload modeling and prediction policies and rules. For the physical-level optimizer, the manager of the aggregated resources receives the resource requirements from the services and then optimizes the data center to use the fewest physical resources to host the services with an acceptable service quality.

The sample procedure of leveraging the automated consolidation function is shown in Figure 4. At the service level, the predefined monitor retrieves the workload and service quality from each appliance. On the basis of workloads, a predefined workload predictor estimates the approximate workload in the next phase. According to this estimated value and the service-level

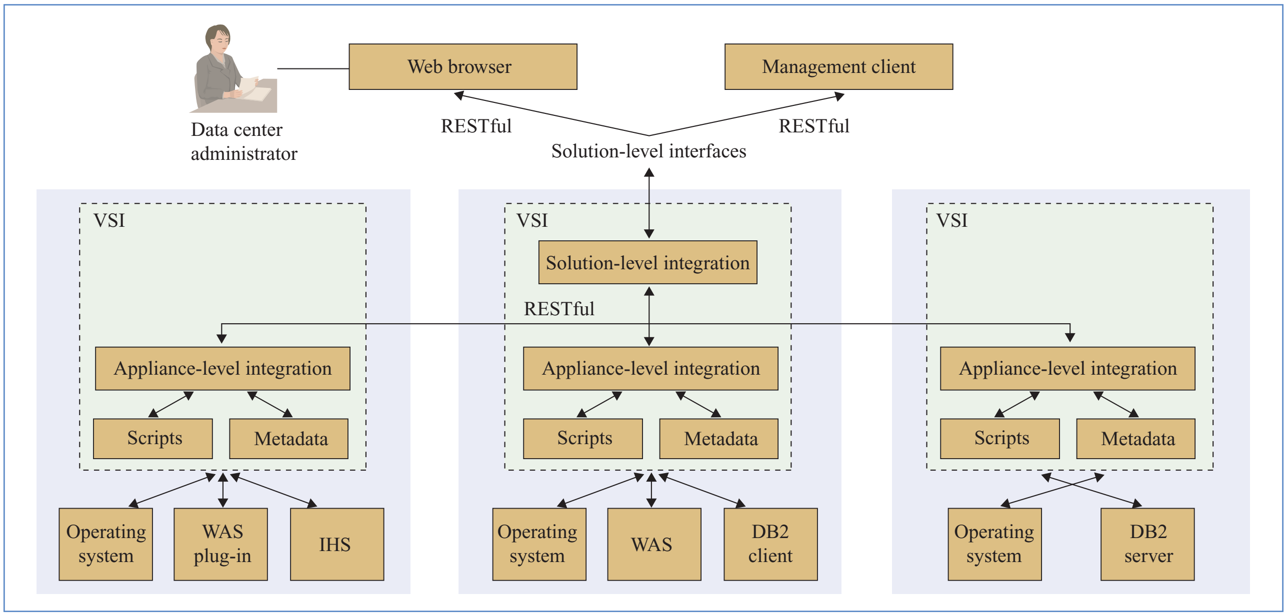

Figure 5

Virtualization integrator (VSI) architecture of a sample service. (RESTful: representational state transfer.)

agreement, a predefined service-level optimizer computes and optimizes the amount of resources required for each virtual appliance and corresponding software configuration. If the required resources need to change, the service-level optimizer will send a new resource request to the manager of the aggregated resources, which triggers physical resource optimization by allocating physical resources with minimal resource adjustment overhead.

Reinventing virtual appliances with built-in management capability

To provide end-to-end life-cycle management of IT services, virtual appliances need to be reinvented to incorporate enough management capabilities for different stages of the whole service life cycle. As a service usually consists of a set of software resources located in multiple virtual appliances, the software resources and virtual appliances need to be organized to provide higher-level manageability for the data center administrator and external management systems. We designed VSI, an inner appliance management framework into which management capabilities are easily plugged and integrated. VSI connects the resources distributed across multiple appliances to provide highly autonomic management within the virtual appliances and to provide a very simplified view to the outside higher-level management services. Virtual appliances with VSI embedded are able to be easily created, assembled, deployed, and managed, in support of the functions described in the previous section "Redefining virtual appliances."

VSI architecture

VSI is an extendable integration framework embedded inside virtual appliances. **Figure 5** shows the overall architecture of VSI, which is designed with the following style.

- *Embedded into appliances*—VSI is embedded in each virtual appliance as management software, enabling management capabilities to be easily plugged and integrated.
- *Interconnected and coordinated*—The VSIs within the scope of a virtual solution are interconnected so that multiple VSIs can coordinate to achieve the management capabilities. Among these VSIs, one of them behaves as the primary coordinator through which the VSIs exchange information and commands.
- *Integration capabilities*—Management capabilities are integrated into two levels: the appliance level and the solution level. The appliance-level capability integrates the software resources located in an appliance and exposes the appliance-level function using standard application-programming interfaces (APIs). The solution level connects multiple

appliances within the service to compose higher-level management capabilities. Each management capability is implemented as a solution-level agent embedded into the primary coordinator and as appliance-level agents embedded into every virtual appliance within the solution. The primary coordinator exposes solution-level capabilities for the administrator by using the appliance-level capabilities provided by appliance-level integration.

- *RESTful and external manageable APIs*—The management capabilities of virtual appliances are exposed by way of REST (representational state transfer) APIs, through which the external management component and user with a Web browser can access the appliance. The communication between the primary coordinator and other appliances is also through REST APIs.
- *Plugged scripts and metadata*—The ISV customizes the VSI function for a particular service by providing software-specific scripts and metadata. Pluggable scripts are provided into the VSI framework to support the software resources-specific configuration, control, and data-fetching operations in each appliance. Metadata is used to describe the virtual images, software scripts, and functional scripts of the virtual appliances. The metadata also defines the operation sequences and parameter dependencies in a single virtual appliance and across multiple virtual appliances. The metadata used in VSI is an XML file based on the Open Virtualization Format (OVF) standard plus VSI extensions.

VSI integrated capabilities

Virtual appliances embed the VSI to provide a set of management capabilities to the external world. The ISV customizes the VSI function for a particular service by providing software-specific scripts and metadata. VSI is also extendable for the ISV to add management functions, for example, problem determinations. In the following subsection, we describe our design for these capabilities by the VSI and meanwhile discuss the challenges for implementation.

Appliance creation

After the VSI is installed in the virtual machine, appliance creation is started. During this phase, the VSI components are embedded into the virtual appliances, including metadata and script. The metadata includes the necessary information for the appliance to be successfully configured and activated, such as the network customization parameters for new environments; the script provides the capability to configure and manage the software inside the appliance sothe resources required by the following phases are provided.

Solution assembly

In the next step, solution assembly, multiple virtual appliances are assembled into a virtual solution to provide a high-level service. Linkages are built across the virtual appliances. VSI provides the function for creating cross-appliance constraints, checking metadata compliance, and merging and synchronizing metadata.

Topology construction

For virtual appliances, service deployment becomes much easier. After choosing the virtual solution that contains multiple virtual machines for the service, the data center administrator only needs to place the virtual appliance templates and the corresponding configuration metadata on the targeted physical platform(s). Then, the data center administrator simply boots these virtual appliances to make the service ready for use. Since a virtual solution generally contains multiple virtual appliances and each performs certain functions of the service, the topology construction function is provided for describing inter-appliance and the intra-appliances structure of the service. VSI enables constructing the topology of a virtual solution at runtime, and the constructed topology provides one integrated service-level interface to the external systems, rather than individual appliance-level interfaces.

Service activation

Service activation instantiates software and appliances, constructing the service in a coordinated manner from the preconfigured template state to the customized state with the customers' unique parameter values. The instantiation operation of a solution spans multiple software resources located on multiple virtual appliances. Traditionally, in order to coordinate the activation sequence of all the software resources in a solution, domain expertise is required in the deployment phase. However, since VSI-embedded virtual appliances have already had the intelligence implanted by domain experts in the creation and assembly phase, in the deployment phase the data center administrator does not need to develop a thorough knowledge of the prerequisite software and their interdependencies.

Environment adaptation

In a complex enterprise environment, many services exist simultaneously. Service registration is especially important, since it helps the data center administrator to have an aggregate view of all the services. Therefore, commonly there is a service registration center in the IT environment. To support service registration, services composed of VSI-embedded virtual appliances are able to provide information for service registration. In VSI, this function is called *environment adaptation*.

Service measurement

The service measurement instrumentation component of VSI provides useful data that allows the virtual solution management node to monitor the health of the service and all registered appliances. The virtual solution management node manages and monitors all appliances in the solution by checking the information about CPU, memory, network, disk storage, and energy consumption in each appliance, and then providing selective summary information for the service.

According to the requirements of data presentation from customers, the virtual solution management node collects utilization data from each appliance and consolidates it. The consolidated data can be grouped by utilization types, such as the CPU information of all nodes in the virtual solution. In addition, the data can support VSI capabilities such as performance analysis and resource allocation. To reduce the traffic transmitted among the virtual appliances, VSI is implemented to collect the required metrics in a distributed manner from different appliances and then to compose these metrics in the management node according to the customer's requirements.

Availability management

The availability module updates the state of the service, appliances, and software resources. The state values can be queried and changed based on different management requirements. To enable the service-level and appliances-level operations, these operations are well defined as a sequence of software resource-level operations in the appliance development phase.

When a service start command is launched, it will be divided into several sub-items to start the virtual machine or start software. These sub-items are executed following a particular sequence generated by VSI based on the dependencies described in metadata. In addition, availability is transactional. If any part of the transactional chain does not update the state successfully, the state update of the whole solution fails and the whole state update process is rolled back to the beginning.

Security compliance

Security compliance is designed to track and guarantee the password consistency among software and systems, since one password needs to be consistent with another one. Suppose the password between a DB2 server and a DB2 client need to be compliant. If the user wants to change the password of the DB2 server, the VSI automatically verifies and modifies the password for both the DB2 server and the DB2 client.

Performance optimization

Performance optimization provides a framework to achieve automated performance adaptation and optimization by utilizing the monitoring, tuning, and optimizing functions that are defined and embedded into the VSI framework in the appliance creation and solution assembly phases. The monitoring function provides the capability to obtain run-time performance metrics such as throughput and response time. The ISV also needs to define the optimization policies and operations in the appliance creation and solution assembly phase. The performance optimization function executes these predefined rules automatically in the runtime.

Resource allocation

Resource allocation fulfills the physical resource management requirements for the services running on virtual appliances by communicating with the external resource manager of the aggregated resources. The external resource manger utilizes the dynamic resource adjustment and live migration capabilities of the hypervisor to perform the resource allocation and optimization operations, which are completely transparent to the data center administrator. The hypervisor, also called *virtual machine monitor* (VMM), is a virtualization platform including both hardware and software, which allows multiple independent guest operating systems to run on it concurrently. The optimization needs to consider power consumption, the impact of server live migration, and CPU and memory utilization. VSI provides a framework to plug in the scripts and tools for the service to easily talk to different external resource managers.

Capacity extension

The VSI is also extendable so that the ISV can add management functions. For example, in order to add the capability of problem determination (PD), the ISV needs to design and implement a solution-level PD agent and appliance-level PD agent following the VSI architecture style. The appliance-level PD agent may gather useful information for problem determination and the solution-level PD agent may create the analysis report based on this information.

A sample implementation: Service activation with VSI

In this section, we use service activation as a sample capability to illustrate how to leverage VSI to activate services. From the viewpoint of VSI, the service activation process is a set of coordinated activation items.

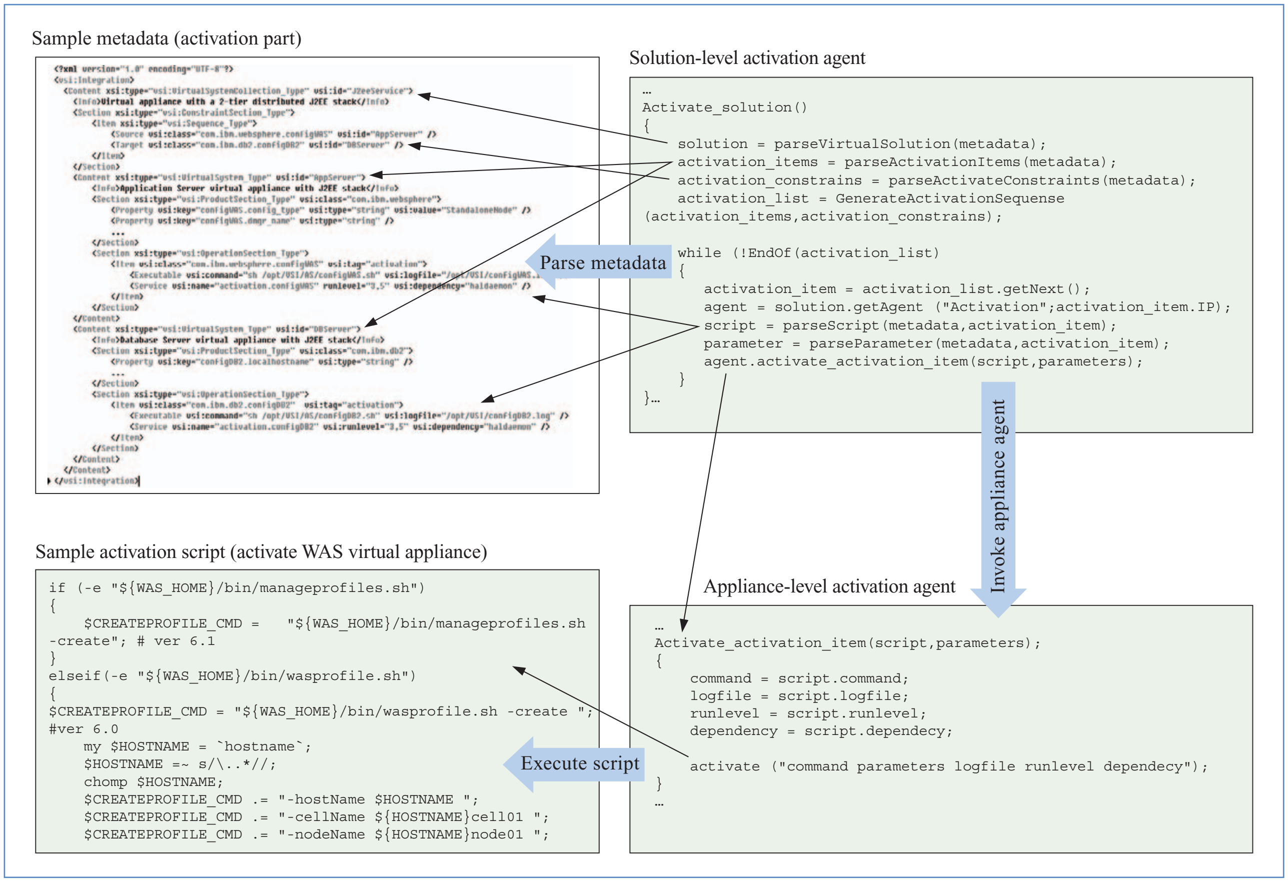

Figure 6

Virtualization integrator (VSI) implementation for service activation capability.

Each activation item, implemented as an activation script, initializes a series of settings for a software application or guest operating system with user environment-compatible parameters. The solution builder is responsible for writing these solution-specific activation scripts and defining these scripts and their constraints in metadata. In order to drive these activation scripts, we implemented a solution-level activation agent and an appliance-level activation agent in VSI. Following the architecture of VSI, the solution-level activation agent is built in the management virtual appliance only and the appliance-level activation agent is built in both the management virtual appliance and the managed virtual appliances. These activation scripts and metadata, working together with the solution-level activation agent and appliance-level activation agent, provide the service activation capability. The implementation detail is shown in **Figure 6**.

The top left of Figure 6 shows sample activation metadata in which two activation items are described. One is in the WAS virtual appliance and the other is in the DB2 virtual appliance. The metadata describes the item name, parameters and their types, script command, log file path, and dependency for each activation item. Also, it indicates that the DB2 item should be activated before the WAS item. The solution-level activation agent can expose the capability for activating the solution, which is executed automatically when the solution starts for the first time.

When the solution is going to be activated, the solution-level activation agent parses the metadata for all the activation items within the solution and their activation constraints that depict the activation sequence between these activation items. After generating the sequence for all the activation items, the solution-level activation agent parses the metadata for scripts and parameters for each activation item and invokes appliance-level activation agents to activate it.

The appliance-level activation agent exposes the capability of activating an activation item. This agent gets the script command, log file path, and activation parameters and then runs the activation script with the context. A sample activation script is shown on the bottom left of Figure 6. This activation script is responsible for creating a WAS standalone profile and a data source in connection with the backend DB2.

After implementing the virtual appliances with service activation capability, the deployment time is greatly reduced during the activation for the WAS solution. The traditional approach takes 115.5 minutes to complete 20 operations for installing, patching, and creating different software, but our approach reduces this time to 2 minutes and 10 seconds, that is, 1 minute for activating five images and 1 minute and 10 seconds for starting the service. While the time using traditional mechanisms is dependent on the specific software system, the deployment time for virtual appliances is mainly driven by the number of virtual appliances to be activated (including configuration of virtual appliances) and the time to start the software system. The complexities of service deployment are, thus, largely eliminated since most of the problems are solved in the development phase. Because of space limitations and other reasons, additional details on our implementation and field results are not provided in this paper.

Related work

Virtualization is a very active research area today. However, previous work in system management was generally based on a particular physical machine and operation system. With the increasing scale of services, and with heterogeneous environments, those specific techniques can no longer satisfy new requirements for operations on various physical machines within a data center. In meeting the need for cross-platform and cross-operating system operation, there has been recent work on the technology of virtualization and hypervisors (VMMs) [3, 12–16]. Chen and Noble [17] described how secure logging, intrusion prevention and detection, and environment migration can take advantage of virtual machines. Wlodarz [1] made a survey of virtualization technologies and listed the suitability of virtual machines for hardware replacement, testing and debugging, education and e-learning, and security systems. Barham et al. [3] present an x86 VMM, Xen**, that allows multiple commodity operating systems to share conventional hardware in a safe and resource-managed fashion, without decreasing performance or functionality. Adams and Agesen [13] compared the software VMM with a new VMM designed for the emerging hardware support and found the hardware VMM often exhibits lower performance than the pure software VMM. Further, to overcome the poor scalability and extensibility of traditional VMMs, which partition a single physical machine into multiple virtual machines, Whitaker et al. [16] proposed the Denali VMM, which uses para-virtualization to promote scalability and hardware interposition to enhance extensibility.

By combining the advantage of both appliances and virtualization technology, virtual appliances [4] are emerging as a groundbreaking technology to solve the complexities in large-scale data centers. Sapuntzakis et al. [18, 19] proposed Collective, a compute utility using virtual appliances to manage systems. Reference [18] attempts to address the complexity of system administration by making the labor of applying software updates independent of the number of computers, through leveraging of virtual appliances. It also presents the concept of virtual networks of virtual appliances and describes the prototype of the collective utility. Mastrianni et al. [20] proposes the design of IT Autopilot to use appliances to simplify the deployment and management of small and medium-size business services. The IT Autopilot integrated IT service management platform is capable of combining different tools and services to create specific customized IT service solutions. Virtual appliances are believed to simplify service deployment [11], since they include the guest operating system and preconfigured applications. However, traditional virtual appliances [21] have little or no management function and cannot deploy services automatically.

In contrast to the existing work, our work focuses on combining virtualization technology with appliances to deploy and manage services. We extend previous work [11, 22–24] toward reinvented virtual appliances with built-in management capability at both the individual appliance level and the virtual solution level. We redefine virtual appliances to be much more appliance-like by including appliance-specific components and management functions. We also look at using virtual appliances to solve the management challenges in large-scale data centers. Our solution enables the solution provider to combine virtual appliance and related management capability at the solution creation stage, thus avoiding the traditional burden of handling most operation and management challenges at runtime.

As for the relationship of virtual appliances to related applications, such as virtual machines, virtual appliances also have to incorporate an application, operating system, and virtual hardware. However, as opposed to virtual machines without application software, virtual appliances are delivered to customers as prebuilt and preconfigured solutions for simple deployment by eliminating the need for manual configuration of the virtual machines and operating systems [25]. Meanwhile, virtual appliances solve the key problems of hardware heterogeneity and operating system configuration complexity in grid computing and decouple the grid operator from the grid consumer by encapsulating all knowledge of the application within the virtual appliance [25]. In addition, with no need to change the application architecture for multi-tenancy, virtual appliances provide a direct route for traditional on-premises applications to be rapidly redeployed in a software-as-a-service (SaaS) mode. In contrast, traditional approaches to SaaS leverage shared infrastructure by forcing massive change and increased complexity on the software stack [25].

In addition, to provide a complete view of virtual appliance technology, we have also described the potential challenges of virtual appliances here. As analyzed in Reference [4], since virtual appliances remove control points around the operating system, auditing, and management, they also cause the IT department to lose some management control and thus force a vertical management approach for each virtual appliance stack. Meanwhile, a proper hypervisor must be developed and established on the physical platform to enable virtual appliances, which may become an obstacle for the establishment of a virtualization platform, and thus increase costs for the virtual appliance vendor. Also, configuration of virtual appliances often necessitates communication with existing databases, directory servers, and other enterprise components, which may result in certain complexity beyond the expectation for an appliance. In addition, as the development of virtual appliances is still in its infancy, there are no general standards for large-scale implementation. As standards and new management products emerge, IT organizational processes will also need to adapt in order to support wider uses of virtual appliances [4].

Conclusion

In this paper, we have extended existing work on using virtual appliances for service deployment, by examining the following new areas: 1) rapid provisioning, which simplifies and accelerates the deployment and activation of virtual appliances by checking parameter dependencies and coordinating execution sequences, 2) selective presentation, which provides flexible interfaces to present meaningful information through customized metrics, 3) simplified operation, which enables various operation granularities by coordinating the operation sequences across virtual appliances, and 4) automated consolidation, which enables automated service performance management by automatically monitoring and optimizing resource allocations. Virtual appliances with these features can be easily used to automatically deploy services, activate and deactivate services, monitor and dynamically allocate system resources, and manage themselves.

To deliver virtual appliances with these new capabilities, we designed and prototyped an extendable framework, VSI, that enables an ISV to easily build the key features of virtual appliances. VSI is embedded inside virtual appliances, so it connects the components located over multiple virtual machines and coordinates operations to perform various management functionalities. We explored VSI by examining the whole life cycle of a typical enterprise SOA service using virtual appliances.

Virtual appliances provide novel opportunities to enable radically simplified and automated system deployment and management in both current data centers and new enterprise data centers, possibly in a cloud computing environment. Recently, cloud computing has emerged as a novel model for virtualized and efficiently managed data centers to better utilize the underlying computing resources and provide IT-related capabilities as a service. With built-in management capability, our re-inventing of virtual appliances can also reduce the operation workload of the system administrator at cloud-based data centers to achieve improved server utilization and simplified system manipulation. For example, the service deployment for cloud computing can be facilitated by leveraging virtual appliances to automatically deploy prebuilt and preconfigured solution images for different Internet customers. We envision that virtual appliances will be among the most promising candidates for the building blocks of large-scale data centers, and expect more work to follow to further implement virtual appliances for service management and other functions.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Sun Microsystems, Inc., or Citrix Systems, Inc., in the United States, other countries, or both.

References

1. J. J. Wlodarz, "Virtualization: A Double-Edged Sword," 2007; see *http://www.citebase.org/abstract?id=oai:arXiv.org:0705.2786*.
2. T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-Box and Gray-Box Strategies for Virtual Machine Migration," *Proceedings of the ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2007, pp. 229–242.
3. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, 2003, pp. 164–177.
4. R. Willenborg, "Virtual Appliances—Panacea or Problems?" October 3, 2007; see *http://www.ibm.com/developerworks/websphere/techjournal/0710_col_willenborg/0710_col_willenborg.html*.
5. A. J. Keel, M. A. Orr, R. R. Hernandez, E. A. Patrocinio, and J. Bouchard, "From a Technology-Oriented to a Service-

Oriented Approach to IT Management," *IBM Syst. J.* **46**, No. 3, 549–564 (2007).
6. N. Bowen, "Global Technology Outlook 2008: An Overview," IBM Research, January 2008.
7. R. Van Renesse, K. P. Birman, and W. Vogels, "Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining," *ACM Trans. Comput. Syst.* **21**, No. 2, 164–206 (2003).
8. M. Blumenthal and D. D. Clark, "Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World," *ACM Trans. Internet Technol.* **1**, No. 1, 70–109 (2001).
9. M. P. Papazoglou and W. J. van den Heuvel, "Service Oriented Architectures: Approaches, Technologies and Research Issues," *The VLDB J.* **16**, No. 3, 389–415 (2007).
10. V. Talwar, D. Wenchang, and Y. Jung, "Approaches for Service Deployment," *IEEE Internet Computing* **9**, No. 2, 70–80 (2005).
11. C. Sun, L. He, Q. Wang, and R. Willenborg, "Simplifying Service Deployment with Virtual Appliances," *Proceedings of the IEEE International Conference on Services Computing* (SCC), Hawaii, 2008, pp. 265–272.
12. R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F. C. M. Martins, A. V. Anderson, S. M. Bennett, et al., "Intel Virtualization Technology," *IEEE Computer Magazine* **38**, No. 5, 48–56 (2005).
13. K. Adams and O. Agesen, "A Comparison of Software and Hardware Techniques for x86 Virtualization," *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, New York, 2006, pp. 2–13.
14. M. Rosenblum and T. Garfinkel, "Virtual Machine Monitors: Current Technology and Future Trends," *IEEE Computer Magazine* **38**, No. 5, 39–47 (2005).
15. J. E. Smith and R. Nair, "The Architecture of Virtual Machines," *IEEE Computer Magazine* **38**, No. 5, 32–38 (2005).
16. A. Whitaker, R. S. Cox, M. Shaw, and S. D. Gribble, "Rethinking the Design of Virtual Machine Monitors," *IEEE Computer Magazine* **38**, No. 5, 57–62 (2005).
17. P. M. Chen and B. D. Noble, "When Virtual Is Better than Real," *Proceedings of the Workshop on Hot Topics in Operating Systems* (HotOS), 2001, pp. 133–138.
18. C. Sapuntzakis, D. Brumley, R. Chandra, N. Zeldovich, J. Chow, M. Lam, and M. Rosenblum, "Virtual Appliances for Deploying and Maintaining Software," *Proceedings of the USENIX Conference on System Administration* (LISA), Berkeley, CA, 2003, pp. 181–194.
19. C. Sapuntzakis and M. S. Lam, "Virtual Appliances in the Collective: A Road to Hassle-Free Computing," *Proceedings of the Workshop on Hot Topics in Operating Systems*, 2003, p. 10.
20. S. Mastrianni, D. F. Bantz, K. A. Beaty, T. Chefalas, S. Jalan, G. Kar, A. Kochut, et al., "IT Autopilot: A Flexible IT Service Management and Delivery Platform for Small and Medium Business," *IBM Syst. J.* **46**, No. 3, 609–624 (2007).
21. VMware, Best Practices for Building Virtual Appliances, White Paper, 2007; see *http://www.vmware.com/files/pdf/Best_Practices_Building_Virtual_Appliances.pdf*.
22. R. Willenborg, Q. Wang, D. Gilgen, and S. Smith, "Using Virtual Image Templates to Deploy WebSphere Application Server," *IBM WebSphere Developer Technical Journal,* Volume 5, May 2007; see *http://www.ibm.com/developerworks/websphere/techjournal/0705_willenborg/0705_willenborg.html*.
23. L. He, S. Smith, R. Willenborg, and Q. Wang, "Automating Deployment and Activation of Virtual Images," *IBM WebSphere Developer Technical Journal,* Volume 8, August 2007; see *http://www.ibm.com/developerworks/websphere/techjournal/0708_he/0708_he.html*.
24. M. Yarnall, L. Berc, and Q. B. Wang, "Using VMware ESX Server with IBM WebSphere Application Server," July 2006; see *http://www-900.ibm.com/cn/crl/download/ESX_WAS_WP_24Jul06_final.pdf*.
25. VMware, Virtual Appliances: A New Paradigm for Software Delivery, White Paper, 2008; see *http://www.vmware.com/files/pdf/vam/VMware_Virtual_Appliance_Solutions_White_Paper_08Q3.pdf*.

Received December 31, 2008; accepted for publication February 25, 2009

Xing Jin *IBM China Research Laboratory, Building 19, Zhouguancun Software Park, 8 Dongbeiwang West Road, Haidian District, Beijing, P.R.C. 100193 (jinxing@cn.ibm.com).* Dr. Jin is a Research Staff Member at the IBM China Research Laboratory. He received his B.S. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 1999 and 2005, respectively. His current research interests include virtualization, appliances, and cloud computing. He is currently working on simplifying service management in large-scale data centers by leveraging virtual appliance technology.

Ruth Willenborg *IBM Software Group, WebSphere Technology Institute, 4205 South Miami Boulevard, Durham, North Carolina 27703 (rewillen@us.ibm.com).* Ms. Willenborg is a Senior Technical Staff Member in the IBM WebSphere Technology Institute and is currently working on WebSphere cloud computing and virtual appliance initiatives. Prior to this assignment, she was the manager of the WebSphere performance team responsible for WebSphere Application Server performance analysis, performance benchmarking, and performance tool development. She has more than 20 years of experience in software development at IBM. She is coauthor of *Performance Analysis for Java Web Sites* (Addison-Wesley, 2002) and numerous articles on both WebSphere performance and using WebSphere with virtualization technologies. Ms. Willenborg holds a B.S. degree in computer science from Duke University.

Yang Zhao *IBM China Research Laboratory, Building 19, Zhouguancun Software Park, 8 Dongbeiwang West Road, Haidian District, Beijing, P.R.C. 100193 (zzhaoy@cn.ibm.com).* Mr. Zhao is a Research Staff Member at the IBM China Research Laboratory working in the fields of virtual appliances and cloud computing. He received a Master of Computer Science degree from Peking University, Beijing, China, in 2007. He is one of the main architects and chief developers of virtualization integrator (VSI). His research has focused on virtual appliance and cloud environment provisioning, monitoring, and scheduling, as well as Web-based virtual appliance life-cycle management.

Changhua Sun *Department of Computer Science and Technology, Tsinghua University, Beijing, P.R.C. 100084, China (sunchanghua@tsinghua.org.cn).* Mr. Sun is a Ph.D. candidate in the Department of Computer Science and Technology at Tsinghua University. He has spent one year performing research on service deployment and virtual appliances as an intern in the Appliance and Virtualization Group, IBM China Research Laboratory. His research interests include network security, distributed systems, and virtualization. He will join China the Service-oriented Architecture Design Center at IBM China Development Laboratory after his graduation.

Le He *IBM China Research Laboratory, Building 19, Zhouguancun Software Park, 8 Dongbeiwang West Road, Haidian District, Beijing, P.R.C. 100193 (lehe@cn.ibm.com).* Mr. He is a Research Staff Member at the IBM China Research Laboratory. He received a master's degree in signal and information processing from Beijing University of Posts and Telecommunications in 2007. He has been actively working on the technologies of virtual appliances and cloud computing for more than three years. He was the primary architect and chief developer of the activation engine, which is widely adopted within IBM. He also played an important role in designing and developing a prototype for the life-cycle management of virtual appliances: virtualization integrator (VSI). He is currently working with the AIM/WebSphere team on the WebSphere cloud computing product.

Zhijia Chen *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (zj-chen05@mails.tsinghua.edu.cn)*. Mr. Chen is currently a Ph.D. candidate at Tsinghua University. He was a visiting student at the Stanford School of Engineering in the spring of 2007 and visiting scholar in the Computer Science Department of the Hong Kong University of Science and Technology in 2008. His areas of research include content distribution networks, person-to-person streaming, cloud computing.

Ying Chen *IBM China Research Laboratory, Building 19, Zhouguancun Software Park, 8 Dongbeiwang West Road, Haidian District, Beijing, P.R.C. 100193 (yingch@cn.ibm.com)*. Dr. Chen is a Research Staff Member and Senior Manager at IBM China Research Laboratory. He received a Ph.D. degree in computer science in 1999 from Southeast University, Nanjing, China. His current research interests are distributed computing, cloud computing, and IT services management.

Qingbo Wang *IBM China Research Laboratory, Building 19, Zhouguancun Software Park, 8 Dongbeiwang West Road, Haidian District, Beijing, P.R.C. 100193 (wangqbo@cn.ibm.com)*. Mr. Wang is a Research Staff Member and Manager at IBM China Research Laboratory. He received his B.S. and M.S. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 2000 and 2002, respectively. His research interests include virtualization, appliances, and cloud computing. Mr. Wang is currently researching systems management simplification by applying virtual appliances technology to enterprise cloud environments.