

Providing Service Using A Virtualization Infrastructure

Young R. Park, Savannah State University, USA
Srinarayan Sharma, Oakland University, USA

ABSTRACT

Services could be designed to be simple, atomic, and transparent to potential consumers who interact with providers to attain desired outcome. Providing the desired service, though, may involve a series of complex operations such as accessing multiples databases owned by multiple parties residing in multiple locations, complex queries to obtain desired outcome, transforming these queries for proper display, displaying the outcome in a format comprehensible to the consumer, et cetera. The development of service-oriented architecture (SOA) or service-oriented computing (SOC) allows organizations to exploit the services that may or may not belong to them to provide customized services to potential customers. Virtualization, viewed as the separation of a service request from the underlying physical delivery of that service, allows such services to be provided in a flexible and cost-effective manner. Virtualization is also seen as one of the green ITs which can help reduce infrastructure and maintenance cost especially for the organizations that may not be able to afford big capital investment upfront. In this paper we make a case for integrating SOA and virtualization technologies to provide desired services to the consumer in highly a flexible and cost effective manner.

Keywords: Virtualization; IT infrastructure; Service-Oriented Architecture; Service-Oriented Computing; Green IT

INTRODUCTION

Services are visible to potential consumers who interact with service providers through a series of information exchanges to produce desired outcome. From a consumer's perspective, invoking a service to get a desired outcome typically involves a simple atomic operation. But the detailed actions taken by the service providers to provide the outcome may involve a series of complex operations such as accessing multiples databases owned by multiple parties residing in multiple locations, complex queries to obtain desired outcome, transforming these queries for proper display, displaying the outcome in a format comprehensible to the consumer, et cetera. The development of service-oriented architecture (SOA) or service-oriented computing (SOC) allows organizations to exploit the services that may or may not belong to them to provide customized services to potential customers. By various estimates use of SOA is rapidly increasing. Gartner (2006) had projected that SOA would be used on more than 50% of new, mission-critical applications and business processes designed in 2007, and its use would increase to more than 80% by 2010. IDG Research's (2006) SOA survey found that almost 80% of the respondent companies planned to deploy SOA as a way to connect existing and future applications and 60% cited the need to share business components. Springboard Research (2007) predicted that by 2009 SOA integration and consultancy services would form the bulk of the SOA opportunity in Asia Pacific with around 60% market share.

Recently a complimentary technology, x86 virtualization, has come to the fore. Virtualization refers to technologies which provide a layer of abstraction of the physical characteristics of computing resources between computer hardware systems and the software systems running on them. Gartner (2008) has identified virtualization as one of the top ten strategic information technologies for 2009 and projected its rapid growth worldwide. In particular, it expects the server virtualization software market to grow at a compound annual growth rate of 28 percent during 2009-2013. IDC's projection of worldwide growth of virtualization adoption is similar (Lai and

Thibodeau, 2008). Virtualization is also seen as one of the green ITs which can help reduce infrastructure and maintenance costs (Pratt, 2007).

While SOA allows companies to provide desired services to the consumers by accessing service components from various sources owned by various entities at various locations and manipulating them, x86 virtualization allows them to provide these services in highly flexible and cost effective manner. To the best of the authors' knowledge no attempt has been made to integrate these technologies in spite of their complementarity. In this paper, we try to fill this void.

The paper is organized as follows. First we define SOA and its architecture. Then we define x86 and its architecture. Following this, we make a case for integrating SOA and x86 virtualization technologies. Finally we provide some concluding remarks.

SOA AND ITS ARCHITECTURE

OASIS' (2006) reference model defines SOA as a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired outcomes consistent with measurable preconditions and expectations. IBM (2009) depicts it as a business-centric IT architectural approach that supports integrating business as linked, repeatable business tasks, or services. SOA is also seen as an IT strategy to enable business transformation that builds IT systems out of parts. Another useful description of SOA is provided by Bell (2008) who views SOA as separating functions into distinct units called services, which developers make accessible over a network in order that users can combine and reuse them in the production of applications. The driving force behind the SOA approach is that organizations do not want to give up the functions they have already used for decades from the legacy systems. Instead of developing these already well-defined and deployed functions again from the scratch for the new platform they are transitioning to, the companies look for a way to reuse and integrate them with the new system. The value of SOA is predicated on this concept of re-use which is also a part of the green IT initiative. Reuse of internal and/or external services also accelerates the return on investment for any SOA strategy. SOA initiative is typically process-driven, standard-based and loosely-coupled but requires well-defined interfaces. With this approach companies can centralize business logic and services can be aligned with business process steps. Services then can be changed incrementally as processes change, which allows companies to have faster response to customer needs and shifting markets. Linthicum (2009) emphasizes this agility of service oriented architecture as the value proposition versus traditional architecture. Since everything is depicted as services, they can be configured and reconfigured either in composite applications, such as mashups, or binded into processes, or abstracted into applications.

Figure 1 shows a typical SOA architecture. As shown in Figure 1, services are loosely coupled and they usually have their own dedicated databases to perform the tasks. Like any other enterprise software architecture SOA infrastructure also needs to address security, scalability, performance, and others as well as the manageability of the services. WebMethods (2005) proposes seven technical capabilities -- services, a registry, messaging, services management, orchestration, analytics, and user interface -- that a company needs to consider when it creates an enterprise SOA. Please see Table 1 below for their description.

Table 1 Summary of Required Functionality of Each Capability

Capability	Description
Services	As a building block, it creates services from old and new components.
Registry	Catalog services in a central repository or a directory
Messaging	Connect services together and enable services components to communicate with each other reliably through a variety of interaction patterns.
Services Management	Govern and manage services.
Orchestration	Assemble services into a process
Analysis	Monitor process and service performance
User Interface	Deliver the user experience

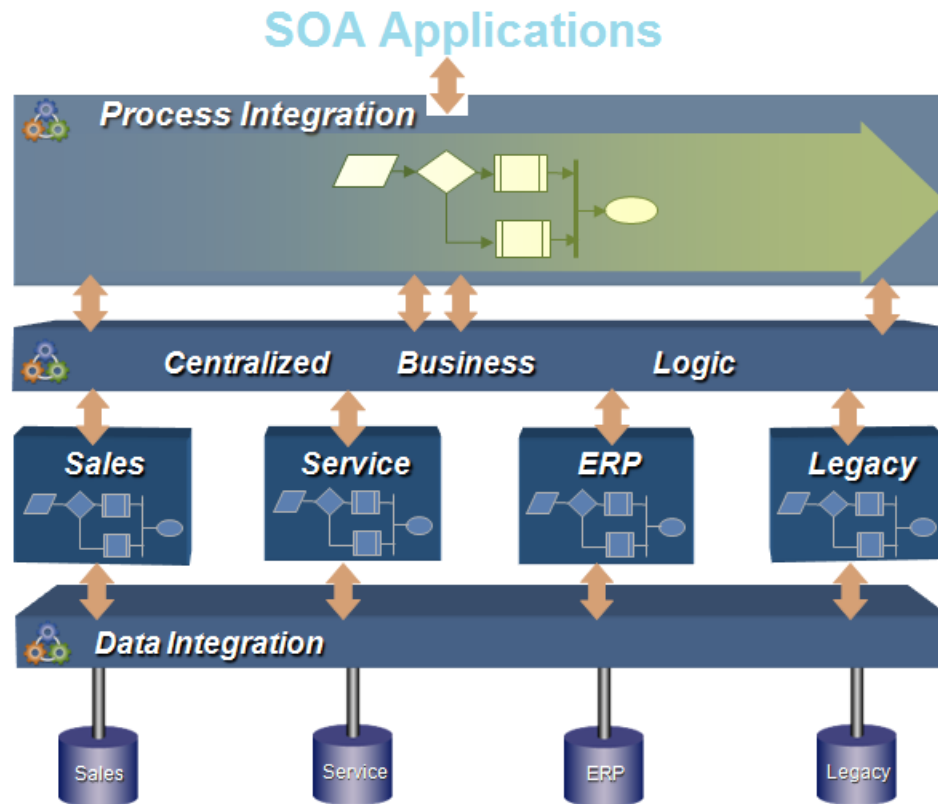


Figure 1: A Typical SOA Application Architecture (Adapted from Oracle, 2008)

VIRTUALIZATION AND ITS ARCHITECTURE

VMware (2007) considers virtualization as the separation of a service request from the underlying physical delivery of that service. Parallels (2008) views virtualization technologies as abstracting virtual servers from the underlying hardware, while Intel (2008) depicts virtualization as the abstraction of the computer hardware from the way in which it is used. Schott (2003) regards virtualization as a logical abstraction which hides the underlying resource sharing and its clusters and presents a single or arbitrary point of contact. As stated above, virtualization refers to a class of technologies which provide a layer of abstraction of the physical characteristics of computing resources between computer hardware systems (e.g., x86) and the software systems running on them (e.g., Windows Vista). Please see Figure 2. A properly designed virtualized system provides application programs an execution environment identical to its host hardware, minimally degrades performance, and has complete control of system resources (Popek and Goldberg, 1974). A virtualization layer of system software, called hypervisor, is inserted between the guest operating system and the underlying hardware on which it is implemented by software emulation of the underlying hardware platform's architecture (Crosby and Brown, 2007; VMware, 2007). Available hypervisors permit multiple operating systems instances to run concurrently within virtual machines (VMs) on a single physical machine, dynamically partitioning and sharing the available physical resources such as CPU, storage, memory, and I/O devices. In this architecture, hypervisor is responsible for running multiple operating systems' images and all associated applications to share the resources of a single hardware machine. It makes certain that hardware resources, such as CPU, memory, I/O, etc, are properly and securely partitioned among different operating systems' images and their applications. Applications running in each virtual machine are protected as if they were installed on independent physical machines (Mitchell, 2005). ESX server and Xen use hypervisor architecture. In another implementation of x86 virtualization, called hosted architecture, virtualization layer is run as an application

on top of an operating system (VMware, 2007) (Please see Figure 3 below). VMware player, ACE workstation and server use hosted architecture.

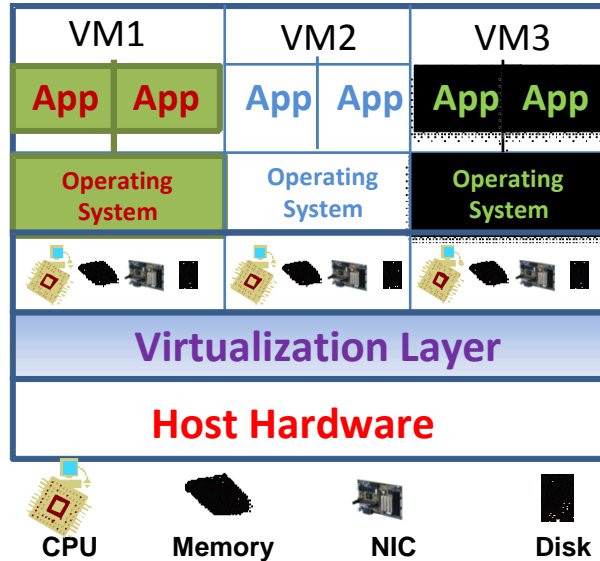


Figure 2: A Typical Virtualization Architecture (Adapted from VMware, 2007)

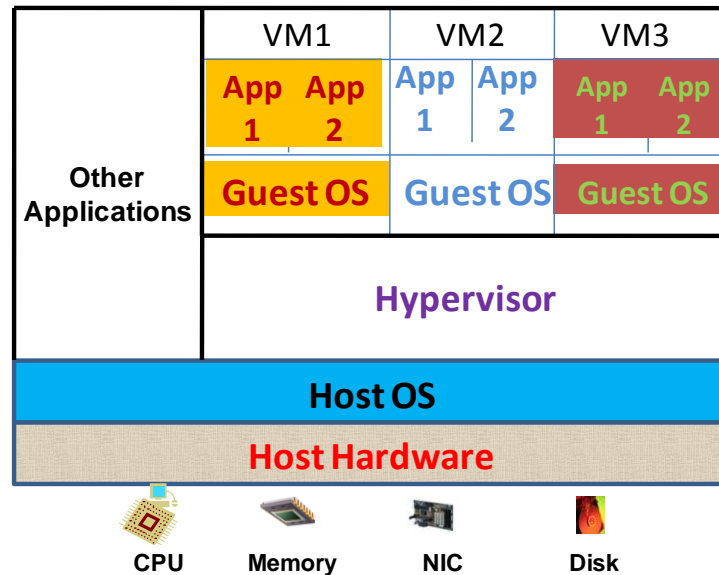


Figure 3: Hosted Architecture (Based on VMware, 2007)

Benefits of virtualization are many. Virtualization can provide user mobility, lower-cost and smaller scale infrastructure, lower carbon emission, rapid provisioning, real-time deployment of live resources, application testing, business continuity, and streamlined backups and mirroring. It can enable the creation of global virtualized infrastructure, rapid shifting of business models, scalability of business growth, smoother integration of mergers, and acquisitions, and optimal use of precious resources such as physical space, technology systems, and natural resources.

INTEGRATING SOA AND VIRTUALIZATION

As discussed above, SOA facilitates customized services to potential customers by accessing multiple databases at multiple locations owned by multiple entities. It is not uncommon that many times there are interruptions in these transactions for various reasons, such as power outage, break in the transmission lines, et cetera. On such occasions, these transactions are rolled back to a previous steady state and redone. Redoing a large volume of transactions many incur substantial cost to the service provider companies over a period of time.

Virtualization can ease this and other data access and transaction related problems by hosting a copy of the required databases each in a separate VM on a local machine in a virtualized environment. After the desired service has been provided, such instances of databases could be erased. Since virtualization also allows real-time transfer of entire platform of services, combining SOA with virtualization would also allow organizations to provide business continuity in case of natural or manmade disasters. It would also provide flexibility to move services to another virtual machine in real-time if number of transactions exponentially increases. With virtualized SOA approach, companies can easily establish a virtual infrastructure that shares computing and service resources when needed rather than having a permanent setup. This will also help them provide better services for customers with reduced infrastructure and maintenance costs.

While such an implementation of the integration of these two technologies is desirable, there may be many roadblocks in the way. If desired databases owned by multiple entities use different operating system, which invariably happens, current virtualization technologies do not allow for hosting multiple instances of different operating system on the same x86 machines. There is also concern about security and privacy of obtained data. Would the obtaining organization abide by the same standard of data security and privacy as the providing organizations? In that case, should a virtualized hosting of these databases be allowed? Moreover, some applications may not run in the virtualized environment if they were not developed for such environments. Vendors of such software applications do not provide support for virtualized implementation of their software. There may be also training cost of existing IT personnel and some upfront capital investment.

CONCLUSIONS

We have presented a case for integrating two complementary technologies, SOA and virtualization, to provide desired services to potential customers in a highly cost effective and flexible manner. We have also noted the advantages and disadvantages of such integrations. Practitioners can use the proposed idea to guide their SOA implementation to serve their customers in a better way. The paper also provides some intriguing thoughts to academicians for future research as to how multiple instances of different operating systems could be hosted on the same x86 machine without compromising security and privacy.

AUTHOR INFORMATION

Dr. Young Park is a visiting professor of CIS at Savannah State University in Savannah, Georgia. Before he joined Savannah State University, he worked as a consultant for a number of projects including IBM, Motorola, HSN, etc. His research interests include service-oriented architecture, enterprise architecture, project management, CRM, etc. He published papers in IEEE Transactions on Neural Networks, IACIS Referred Proceedings and others.

Dr. Srinarayan Sharma is an associate Professor of MIS in the School of Business Administration at Oakland University in Rochester, Michigan. His past work has involved studies of various IT innovations such as open-source software, computer-aided software engineering, data warehousing, mobile commerce, etc. His current interest lies in the application of IT to solve contemporary problems such as global warming, water scarcity, and world poverty. His past work has been published in various IT journals and conferences such as Communications of the ACM, Information Systems Journal, Information & Management, Annual Conferences of the Association of Information Systems, Annual Conferences of the Decision Sciences Institutes, et cetera.

REFERENCES

1. Bell, M. (2008), *Service-Oriented Modeling: Service Analysis, Design, and Architecture*, Hoboken, New Jersey: John Wiley & Sons.
2. Crosby, S. & Brown, D. (2007) The Virtualization reality. *ACM Queue*, Vol. 4, No. 10, December/January 2006-2007, 34-41.
3. Gartner, Inc. (2006), *Predict 2007: SOA Advances*, Nov. 2006.
4. Gartner, Inc. (2008) *Gartner Identifies the Top 10 Strategic Technologies for 2009*. 2008 PRESS RELEASES, <http://www.gartner.com/it/page.jsp?id=777212>.
5. IBM (2009), *Service Oriented Architecture – SOA*, <http://www-01.ibm.com/software/solutions/soa/>.
6. IDG Research (2006), *Poll of 500 IT & Business professionals*, March 2006.
7. Intel, Inc. (2008b) *Virtualization Usage Models*, <http://software.intel.com/en-us/articles/virtualization-usage-models>.
8. Lai, E. and Thibodeau, P. (2008), App support limitations could impede virtualization. *Computerworld*, Vol. 42, No. 2, 12.
9. Linthicum, D. (2009), *Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide*, Indianapolis, IN: Addison-Wesley Professional.
10. Mitchell, R. L. (2005), GHOSTS IN THE MACHINE, *Computerworld*, Vol. 39, No. 17, 28.
11. OASIS (2006), *Reference Model for Service Oriented Architecture 1.0*, OASIS Standard, Oct. 2006,
12. <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>, 29.
13. Oracle (2008), *Application Upgrades and Service Oriented Architecture White Paper*, Apr. 2008, <http://www.oracle.com/technologies/soa/docs/soa-application-upgrades.pdf>, 12.
14. Parallels, Inc. (2008b), *Top Ten Considerations For Choosing A Server Virtualization Technology*, http://www.parallels.com/r/pdf/wp/pvc/Parallels_Virtuozzo_Containers_WP_top_ten.pdf.
15. Popek, G.J. and Goldberg, R.P. (1974), *Formal Requirements for Virtualizable Third Generation Architectures*, *Communications of the ACM*, Vol. 17, No. 7, 412 –421.
16. Pratt, M. K. (2007) 8 Blazing Hot Technologies for '08. *Computerworld*, Dec 26, 2007, from <http://computerworld.com/action/article.do?command=viewArticleBasic&articleId=9049299>.
17. Springboard Research (2007), *SOA in Asia Pacific: Towards more Business Friendly IT*, Feb. 2007.
18. VMware, Inc. (2007), *Undersatnding Full Virtualization, Paravirtualization, and Hardware Assist*, VM Technical Resources, Nov. 11, 2007, http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf.
19. WebMethods (2005), *Service-Oriented Architecture: Revolutionizing IT Systems Development*, Oct. 2005, 9-10.