Proceedings published by International Journal of Computer Applications® (IJCA)
International Conference on Computer Communication and Networks CSI- COMNET-2011

Virtualization Driven Mashup Container in Cloud Computing PaaS Model

Hitesh A Bheda
Department of Information Technology,
L.D. College of Engineering
Ahmedabad, India.

Chirag S Thaker
Department of Computer Engineering,
L.D. College of Engineering
Ahmedabad, India.

ABSTRACT
Traditional application integration technologies are performed in a rigid and slow process that usually takes a long time to build and deploy, requiring professional developers and domain experts. They are server-centric and thus do not fully utilize the computing power and storage capability of client systems. Cloud computing is a new infrastructure deployment environment that delivers on the promise of supporting on-demand services like computation, software and data access in a flexible manner by scheduling bandwidth, storage and compute resources on the fly without required end-user knowledge of physical location and system configuration that delivers the service.

This paper presents the architecture and the organization of a Mashup Container that supports the deployment and the execution of Event Driven Mashups i.e., Composite Services in which the Services interact through events rather than through the classical Call-Response paradigm, following the Platform as a Service (PaaS) model, i.e., the deployment of customer-created applications in cloud platform. In collaboration with PaaS, Virtualization provides an opportunity for extension of independent virtual resources based on available physical systems. In addition, it can provide significant benefits in data centers, such as dynamic resource configuration, live virtual machine migration. Services are deployed in virtual machines (VMs) and resource utilization can be greatly improved. This paper highlights the results of virtualization of mashup container through its supporting scalability and fault tolerance in cloud computing environment.

General Terms
CPU usage, Event Driven Mashup, Memory usage, Resource utilization, Workload.

Keywords
Cloud Computing, Mashup Container, Platform as a Service (PaaS), Virtualization.

1. INTRODUCTION
The face of the Internet is continually changing, as new services and novel applications appear and become globally noteworthy at an increasing pace. Nowadays the locus of computation is changing, with functions migrating to remote data centers via Internet based communication. Computing and communication are being blended into new ways of using networked computing systems. Next generation networks and service infrastructures should overcome the scalability, flexibility, resilience and security bottlenecks of current network and service architectures, in order to provide a large variety of services and opportunities, adoptable by business models capable of dynamic and seamless utilization of IT resources based on user-demand across a multiplicity of devices, networks, providers, service domains and social and business processes [1].

Figure-1 Evolution of Computing

Envisioning the computing utility based on the service provisioning model, where resources are readily available on demand, has led to contemporary computing paradigms that have emerged in the last decade, exploiting technological advances in networked computing environments e.g. GRID computing, peer to peer computing and more recently cloud computing [2]. Figure- 1 shows the result as Cloud Computing from Evolution process of various computing technologies. According to the NIST definition [3], “Cloud computing is a model for enabling convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. Cloud Computing is virtualized compute power and storage delivered via platform-agnostic infrastructures of abstracted hardware and software accessed over the Internet. These shared, on-demand IT resources, are created and disposed of efficiently, are dynamically scalable through a variety of programmatic interfaces and are billed variably based on measurable usage. In a traditional hosted environment, resources are allocated based on peak load requirements. In cloud computing they can be dynamically allocated.

Virtualization, in computing, is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, a storage device or network resources. Virtualization technologies promise great opportunities for reducing energy and hardware costs through server consolidation. Moreover, virtualization can optimize resource sharing among applications hosted in different virtual machines to better meet their resource needs. As a result more and more computing can be conducted in shared resource pools that act as private and public clouds.

The reminder of this paper is organized as follows: Section II presents the Cloud Computing Architecture with its Service Models. Section III formally introduces Virtualization technique and its system architecture. Section IV presents the details about Mashup container, its requirements and management in a cloud computing environment. Section V presents virtualization model characteristics for the mashup container and its proposed results in graphical manner. Section VI demonstrates whether the system should be virtualized or not. Finally, section VII concludes the paper with some future research enhancement.

2. CLOUD ARCHITECTURE

Cloud computing is a style of computing where massively scalable IT-related capabilities are provided “as a service” across the internet to multiple external customers [4]. This term effectively reflects the different facets of the Cloud Computing paradigm which can be found at different infrastructure levels. Form Figure 2, it is possible to identify three Cloud Service Models, namely IaaS, PaaS and SaaS.

2.1 Infrastructure as a Service

IaaS is what the user should opt from virtual computers, cloud storage, network infrastructure components such as firewalls and configuration services. Usage fees are calculated per CPU hour, data GB stored per hour, network bandwidth consumed, network infrastructure used per hour, value added services used, e.g., monitoring, auto-scaling etc. The most popular facebook games, Farmville [5] and Mafia Wars, has more than 230 million monthly users run more than 12000 servers on Amazon AWS. When they launch a new game, they start with a few servers and then ramp up their capacity in real time.

2.2 Platform as a Service

PaaS offerings facilitate deployment of applications without the cost and complexity of buying and managing the underlying hardware and software and provisioning hosting capabilities, providing all of the facilities required to support the complete life-cycle of building and delivering web applications and services entirely available from the Internet [6]

PaaS is a platform where software can be developed, tested and deployed. It means the entire life cycle of software can be operated on a PaaS. This service model is dedicated to application developers, testers, deployers and administrators. In some cases, like Google Apps Engine (GAE) [7], the developers may download development environment and use them locally in the developer’s infrastructure, or the developer may access tools in the provider’s infrastructure through a browser. Other well known examples of PaaS include Microsoft Azure, IBM SmartCloud, Amazon EC2 and salesforce.com.

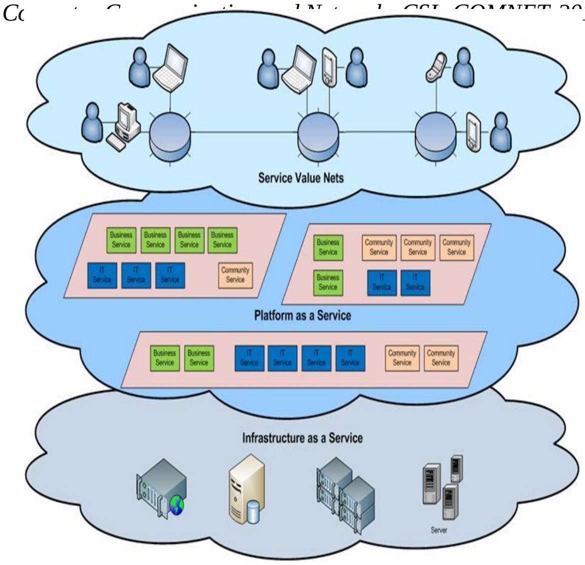

Figure-2 Cloud Architecture

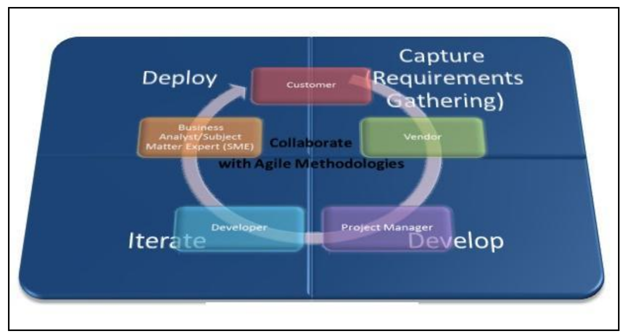

Figure-3 PaaS Framework

2.3 Software as a Service

In SaaS (Software as a Service), the consumer is free of any worries and hassles related to the service. The Service Provider has very high administrative control on the application and is responsible for update, deployment, maintenance and security. For example, Gmail [8] is a SaaS where Google is the provider and we are consumers. We have very limited administrative and user level control over it, although there is a limited range of actions, such as enabling priority inbox, signatures, undo send mail, etc, that the consumer can initiate through settings.

3. VIRTUALIZATION

Th**e** workloads of services usually vary with time, while traditional resource allocation is only done statically. Thus, execution environments are often forced to be overprovisioned based on anticipated peak demands, inevitably resulting in substantial wasted resources besides additional consumed power. The average resource utilization is typically below 15% - 20% [9]. Virtualizing a computing system’s physical resources to achieve improved sharing and utilization has been well established for decades [10]. Classic benefits of virtualization include improved utilization, manageability, and reliability of mainframe systems [11]. Using virtual infrastructure solutions enterprise IT managers can address challenges that include [12]:

• Server Consolidation and Containment – Eliminating 'server sprawl' via deployment of systems as virtual machines (VMs) that can run safely and move transparently across shared hardware, and increase server utilization rates from 5-15% to 60-80%.

• Test and Development Optimization – Rapidly provisioning test and development servers by reusing pre-configured systems, enhancing developer collaboration and standardizing development environments.

• Business Continuity – Reducing the cost and complexity of business continuity (high availability and disaster recovery solutions) by encapsulating entire systems into single files that can be replicated and restored on any target server, thus minimizing downtime.

• Enterprise Desktop – Securing unmanaged PCs, workstations and laptops without compromising end user autonomy by layering a security policy in software around desktop virtual machines.

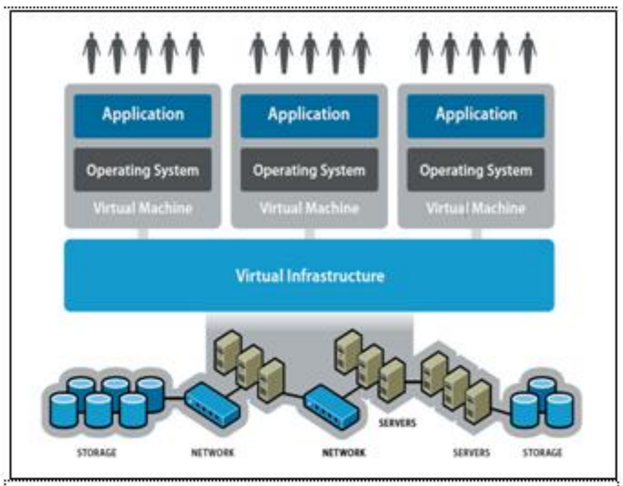

Figure- 4 Virtualization System Architecture

Figure-4 presents the Virtualization System Architecture. System architecture clearly shows that Virtualization allows multiple OSes to share a single physical interface, to maximize the utilization of computer system resources, such as I/O devices. A virtual machine is a tightly isolated software container that can run its own operating systems and applications as if it were a physical computer. A virtual machine behaves exactly like a physical computer and contains it own virtual CPU, RAM hard disk and network interface card (NIC). An operating system can't tell the difference between a virtual machine and a physical machine, nor can applications or other computers on a network. Even the virtual machine thinks it is a "real" computer. Nevertheless, a virtual machine is composed entirely of software and contains no hardware components whatsoever. As a result, virtual machines offer a number of distinct advantages over physical hardware. An additional software layer, named Virtual Machine Monitor (VMM) or hypervisor [13], is introduced to provide the illusion of Virtual Machines (VMs), on top of which each OS assumes owning resources exclusively.

There are mainly two approaches to enable virtualization: Full virtualization and para-virtualization.

3.1 Full virtualization

Full virtualization is designed to provide total abstraction of the underlying physical system and creates a complete virtual system in which the guest operating systems can execute. No modification is required in the guest OS or application; the guest OS or application is not aware of the virtualized environment so they have the capability to execute on the VM just as they would on a physical system using hardware support like Intel Virtualization Technology [14]. This approach can be advantageous because it enables complete decoupling of the software from the hardware. As a result, full virtualization can streamline the migration of applications and workloads between different physical systems. Full virtualization also helps provide complete isolation of different applications, which helps make this approach highly secure. However, in full virtualization, VM monitor must provide the VM with an image of an entire system, including virtual BIOS, virtual memory space, and virtual devices. The VM monitor also must create and maintain data structures for the virtual components, such as a shadow memory page table. These data structures must be updated for every corresponding access by the VMs. Microsoft Virtual Server and VMware ESX Server software are examples of full virtualization.

3.2 Para-virtualization

Para-virtualization presents each VM with an abstraction of the hardware that is similar but not identical to the underlying physical hardware. Para-virtualization techniques require modifications to the guest operating systems that are running on the VMs [15]. As a result, the guest operating systems are aware that they are executing on a VM—allowing for near-native performance. Para-virtualization has limitations, including several insecurities such as the guest OS cache data, unauthenticated connections, etc.

Xen [9] is an open source VMM which supports both para-virtualization and full virtualization. It runs service OS in a privileged domain and multiple guest OSes in the guest domain.

4. MASHUP CONTAINER

Application integration is a strategic approach to binding many information systems together, at both the service and information levels, supporting their ability to exchange information and leverage processes in real time. Although traditional approaches to application integration vary considerably, it is possible to create some general categories, which include information-oriented, business process integration-oriented, service-oriented and portal oriented application integration [16]. A Mashup is a Composite Service implemented as a composition of Base Services. Though mashups promise a future of lightweight application integration with lower costs and greater harmony between the business and IT, there remain many challenges to be overcome. Mashup abstraction should be able to fully utilize the computing power and storage capability of the client side as well as to avoid attack and keep control over sensitive resource for security [17]. Further, an end user-centric integration platform which implements these abstractions is a fundamental infrastructure for lightweight application integration.

The Mashup and the Cloud Computing worlds are strictly related because very often the services combined to create new Mashups follow the SaaS model and, more in general, rely on

Cloud systems. Moreover even the Mashup platforms may rely on Cloud Computing systems as already happens for IBM Mashup Center [18] and JackBe Enterprise Mashup Server [19]. Mashup can be of two types: Data Mashup and Event Driven Mashup. Data Mashups, i.e., those Mashups that combine data extracted by different sources, e.g., Yahoo Pipes! [20]. Event Driven Mashups, i.e., those Mashups in which the basic components - called Services (Svc) from now on-interact through events rather than through the classical Call-Response paradigm. Event Driven Mashups are typically entered through a graphical editing tool which is part of a platform called Service Creation Environment (SCE). The Mashup creator drags and drops blocks corresponding to Svcs and draws edges corresponding to the dependencies among them. Examples of Svcs are Monitoring Services e.g., Monitor Mail, Monitor RSS Feed, etc., Notification Services e.g., Send SMS, Make Phone Call, etc., Presence Services e.g., GTalk, Location Svc e.g., Yahoo! Fireeagle, Maps Svcs e.g., Google Maps, etc.

4.1 Mashup Container Requirements

A Mashup Container must satisfy the following requirements:

• Deployment support, for new Mashup deployment according to the PaaS approach.

• Scalability support, for the simultaneous execution of a very large number of Sessions. The number of Mashups available in the Mashup Container and the number of Mashups that are actually in execution at a given time are supposed to be high and growing as new Mashups are continuously deployed /executed in the system.

• Fault Tolerance, to ensure that even if one or more components fail, the running Sessions continue their execution.

• Low latency and high throughput, to effectively support the execution of fine-grain composite service execution. Latency is very important, e.g., when a Mashup contains short lived Telco services.

• Authentication, Authorization and Accounting (AAA) support, to allow for a seamless integration of the Mashup with the AAA modules already existing in the owner platform. In particular AAA is very important in Enterprise Mashups (i.e., applications that combine public services available on the Web with enterprise private data / functions such as information stored in a Customer Relationship Management - CRM system [21]).

• Management support, to have the complete control of the platform resources, to control Mashup activation / execution as well as to allow the platform administrator to perform appropriate management actions (e.g., enforcing Service Level Agreement rules).

4.2 Mashup Container Management

The Mashup Container is supposed to provide some mechanisms to support the management of Mashups usage. Security is probably the most important aspect. Security must be intended in the following two ways. The first is related to the protection of the system from external malicious service providers while the second is related to the management of user roles. This second aspect is very important for Enterprise Mashups in which different employees probably have different rights in accessing company data (e.g., a secretary accesses only the employees payment database while the boss will have access to all data). Another important aspect of management lies in the monitoring of the Mashups usage, Svcs usage, resource usage, etc. The Mashup Container provides a set of statistics that allow the administrator to have a clear picture of what is happening inside the platform.

The Mashup Container supports the deployment and execution of Event Driven Mashups. New composite Services are developed by means of a graphical tool and deployed in the container to be used by end users. The Mashup Container can be located "in the Cloud" and made available to the user according to the Platform as a Service model. The Mashup developers may deploy Mashups in the provider platform without taking care of low level issues as network support, memory/disk size and CPU performance. In order to achieve this objective, the system takes advantage of the functionalities provided by virtualized environments [14].

5. VIRTUALIZATION MODEL CHARACTERISTICS

To evaluate the effectiveness of mashup container in a virtualized model, workload characteristics of various resources are measured. Different resource usage e.g., processor and memory demands, can be measured at a regular interval of time for the analysis of workload characteristics [22]. The proposed virtualization model defines CPU capacity and CPU demand in units of CPU shares. A CPU share denotes one percentage of utilization of a processor with a clock rate of 1 GHz. A scale factor adjusts for the capacity between nodes with different processor speeds or architectures. For example, the nodes with 2.2 GHz CPUs assigned 220 shares. We note that the scaling factors are only approximate. The memory usage is measured in GB.

Fig. 5 and 6 summarize the memory and CPU usage for the workloads under study. Fig. 5 shows the average and maximum memory usage for each workload. Note, that we order workloads by their average memory usage for presentation purposes. Fig. 6 shows the average and maximum CPU usage of corresponding workloads. There are a few interesting observations:

• For 80% of the workloads, the memory usage is less than 2 GB. While the maximum and average memory usage are small and very close in absolute terms the peak to mean ratios are still high.

• For 10% of the workloads the memory usage is much higher, 10─70 GB; the maximum memory usage can be very large in absolute terms but the peak to mean ratios are less than 3.

• There are strong correlations: workloads with a high memory usage (both peak and average) have higher average CPU usage. Fig. 6 shows that the first 30 workloads have high memory usage and higher average CPU usage than the remaining workloads.

• Most workloads have very bursty CPU demands: while most of the time these workloads have low CPU usage (80% of the workloads use on average less than 220 CPU shares, which corresponds to one physical CPU) their maximum CPU demand is rather high (42% of the workloads have a peak usage of more than 1000 CPU shares).

• The average peak to mean ratio for CPU usage was 66.4, with some workloads having a peak to mean ratio above 1000.

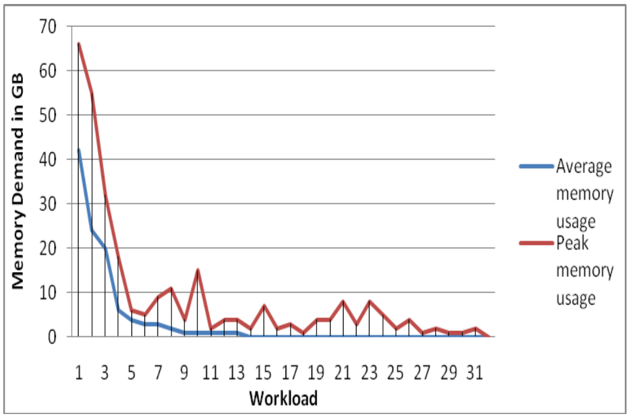

Figure-5 Memory Usage Characteristics

One of the traditional questions for any workload analysis is: how typical are the observed characteristics that are presented above? Most observations about burstiness of the CPU usage patterns were found and discussed in some other studies as well. In particular, a study presented in [23] has analyzed the CPU demands of 139 applications over a period of 5 weeks. It showed that more than half of all studied workloads are very bursty: their top 3% of CPU demand values are 2─10 times higher than the remaining CPU demands in the same workload. Furthermore, more than half of the workloads observe a mean demand less than 30% of the peak demand. These observations show the bursty nature of CPU demands for enterprise applications in different studies. Consolidating such bursty workloads onto a smaller number of more powerful servers is likely to reduce the capacity needed to support the workloads.

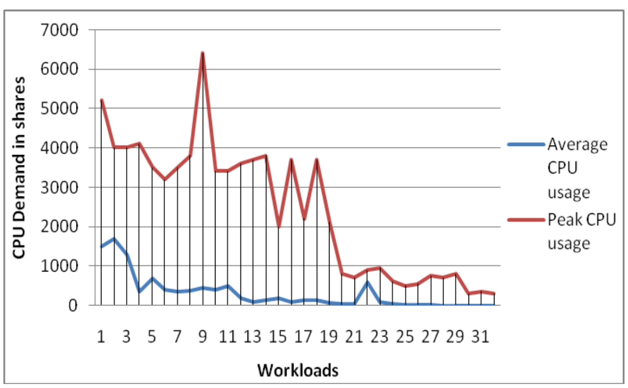

Figure-6 CPU Usage Characteristics

6. MERITS OF VIRTUALIZATION

Commercial virtualization technologies offer excellent support for managing shared resource pools. All the workloads are not use resources in the same way in a consolidated environment. It may be that some large workloads cost more to run within a consolidated environment than to run on a dedicated server. Our goal is to design an automated approach that apportions workload cost in the shared virtualized environment to identify such workloads. Other hosting alternatives can be considered for these workloads to ensure that they are "right-virtualized." The workloads can be hosted directly on dedicated physical machines or using virtualization solutions with lower or no licensing fees. For example, a workload could be less expensively deployed to a server virtualized with Hyper-V [13] or on a server running an open-source virtualization technology such as KVM or Xen [9].

Our approach takes into account the configuration of hosts and the time varying demands of workloads, i.e. resource usage traces of the application over time. The costs-per-host include the host list price, license and maintenance fees for a virtualization solution, and host power usage. The time varying demands of workloads are customer specific. We assume a three year lifetime for the hosts.

In the first phase, a desirable host configuration is chosen for the resource pool. The host has a certain capacity in terms of processing CPU cores and memory. An automated consolidation exercise packs the workloads to a small number of these hosts. The approach takes into account the aggregate time varying (multiple) resource usage of the workloads and a given capacity of the hosts. Multiple host alternatives can be considered iteratively.

In the second phase, we apportion the cost of the shared hosts in the pool among the hosted workloads. If the cost associated with a workload is greater than the cost of a smaller server that could also host the workload, then the workload is a candidate for right-virtualizing. The method can be repeated for different combinations of resource pool host and smaller server host configurations.

In the third phase, we evaluate the average resource usage in the pool to make sure that the selected host configuration for the resource pool is balanced and well utilized. For example, if host memory is often less than 50% utilized we may reduce the memory size for the hosts and repeat the exercise. It is an important exercise that helps a customer to make an informed design choice.

7. CONCLUSION

This paper introduced the virtualization technology with virtualized infrastructure architecture and effectiveness of PaaS based Event Driven Mashup container using virtualization with workload characterization by measuring CPU and memory usage in shared resource environments. Based on the graphical analysis of workload performance that impact on resource pool costs and show that these must be taken into account if the true impact of workloads on resource pool costs is to be considered. The customer can compare the design choices and then make an intelligent decision about them. These different design alternatives lead to potential cost savings of nearly 20% by "right-virtualizing" the workloads.

8. FUTURE WORK

The future work includes: planning for resources including mashup containers that are not used all the time and the relationship with pricing models; considering additional dynamism where workloads are migrated at runtime. It would be of interest to develop mappings from one architecture to another, so that our model can be extended to apply to machines with heterogeneous architectures as well.

9. REFERENCES

[1] Onur, E., Sfakianakis, E., Papagianni, C., Karagiannis, G., Kontos, T., Niemegeers, I., Chochliouros, I.P., de Groot, S.H., Sjodin, P., Hidell, M., Cinkler, T., Maliosz, M., Kaklamani, D.I., Carapinha, J., Belesioti, M., Fytros, E., Fac. of Electr. Eng., Math. & Comput. Sci., Delft Univ. of Technol., Delft, Netherlands, "Intelligent End-To-End Resource Virtualization Using Service Oriented Architecture", GLOBECOM Workshops, IEEE, 28 December 2009

[2] Buyya R, "Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility," 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 1, 2009

[3] Peter Mell and Tim Grance, "The NIST Definition of Cloud Computing", NIST Report, July 2009

[4] Thepparat T., Harnprasarnkit A., Thippayawong D., Boonjing V., Chanvarasuth P., "A Virtualization Approach to Auto-Scaling Problem", Eighth International Conference on Information Technology: New Generations (ITNG), 2011

[5] Zynga, Facebook Games, http://www.zynga.com/about/

[6] Comparing Amazon's and Google's Platform-as-a-Service (PaaS) Offerings, Enterprise Web 2.0, ZDNet.com.

[7] Google App Engine, http://code.google.com/intl/en/appengine/

[8] Gmail, http://www.techno-pulse.com/2011/06/cloud-service-models-saas-paas-iaas.html

[9] 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", In proceedings of the 19th ACM symposium on Operating Systems Principles, Bolton Landing, NY, pp. 164-177, 2003

[10] R.P. Goldberg, "Survey of Virtual Machine Research," Computer, pp. 34-45, June 1974

[11] R.J. Creasy, "The Origin of the VM/370 Time-Sharing System," IBM J. Research and Development, pp. 483-490, Sept. 1981

[12] Virtualization Overview, VMware White paper, http://www.vmware.com/

[13] M. Rosenblum, T. Garfinkel. "Virtual Machine Monitors: Current technology and future trends", Computer, 38(5), Los Alamitos, CA, IEEE Computer Society Press, pp. 39-47, 2005

[14] R. Uhlig, G. Neiger, D. Rodgers, A. L. Santoni, F.C. M. Martins, A.V. Anderson, S. M. Bennett, A. Kagi, F. H. Leung, and L. Smith, "Intel Virtualization Technology," IEEE Computer Volume 38, Issue 5, pp. 48–56, May 2005

[15] A. Whitaker, M. Shaw, and S. D. Gribble. Denali: "Lightweight Virtual Machines for Distributed and Networked Applications", Technical Report 02-02-01, University of Washington, 2002

[16] David, S, Linthicum, "Next Generation Application Integration-From Simple Information to Web Services", Addison-Wesley, 2003

[17] Wei Ye Wenhui, Hu Wen Zhao, Xin Gao Shikun, Zhang Lifu Wang, "Towards Lightweight Application Integration Based on Mashup", Sch. of Electron. Eng. & Comput. Sci., Peking Univ., Beijing, China, World Conference on Services - I, 2009

[18] IBM Mashup Center, www.ibm.com/software/info/mashupcenter/

[19] Presto Mashup Server, www.jackbe.com

[20] Yahoo Pipes!, http://pipes.yahoo.com/pipes/

[21] V. Hoyer, K. Stanoesvka-Slabeva, T. Janner, and C. Schroth, "Enterprise Mashups: Design Principles towards the Long Tail of User Needs", IEEE International Conference on Services Computing, Washington DC, USA, pages 601-602, 2008

[22] Sujesha Sudevalayam and Purushottam Kulkarni, "Affinity-aware Modeling of CPU Usage for Provisioning Virtualized Applications", IEEE 4th International Conference on Cloud Computing, 2011

[23] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper: "Workload Analysis and Demand Prediction of Enterprise Data Center Applications," Proc. of the 2007 IEEE International Symposium on Workload Characterization (IISWC), Boston, MA, USA, September 27─29, 2007