The Challenges of Application Service Hosting

Ike Nassi, Joydip Das, and Ming-Chien Shan

SAP America
3410 Hillview Avenue, Palo Alto, CA 94304
ming-chien.shan@sap.com

Abstract. In this paper, we discuss the major issues associated with the new model of software delivery – service on demand – and explain why it alters the economics of software. As this model is expected to deliver fundamental leaps in cost efficiency, operation performance, infrastructure orchestration and application control, we describe the supporting technology required to achieve these goals. We also highlight those crucial operational processes for enhancing the quality of software delivery under the service on demand model. We then briefly outline our research roadmap to develop an on demand operating environment based on the fundamental principles: standardization, repetition, and, ultimately, automation.

Keywords: Service hosting, service on demand, software as a service, service provisioning, resource virtualization.

1 Introduction

Buying and operating enterprise applications is not like buying traditional durable goods such as TVs or refrigerators – customers are not just buying a discrete product and simply plugging it in for use. Instead, they are buying into an R&D stream of maintenance and operation. Today, the total spent on enterprise IT exceeds one trillion dollars per year. However, amazingly, 75% of that is used for managing existing systems. IDC cites five major areas that cost companies millions of dollars annually: “CIOs and their departments really focus on five key aspects to the management and maintenance of their software systems: availability, performance, security, problems and change management.” For example, Wachovia Bank has been spending $700,000-$800,000 a year on consultant fees just to keep their old system up and running while, similarly, the amount of money spent by 200,000+ Oracle customers for only managing their Oracle software ranged between $40 and $80 billion per year. Maintenance and administration costs over time have been a major culprit in the high spending required to maintain the status quo. “Customers can spend up to four times the cost of their software license per year to own and manage their applications,” says Gartner Inc. Furthermore, the cost of any outage can range from $100 to $10,000 per minute of downtime, according to AMR Research.

As a consequence, customers don’t want to spend money managing their software environment and prefer someone to take over this responsibility but still provide the

L. Baresi, P. Fraternali, and G.-J. Houben (Eds.): ICWE 2007, LNCS 4607, pp. 545 – 549, 2007.

same facility to support their business operations. This led to the advent of new models that delivers ***software as a service*** (SaaS).

In the 1990s, traditional client-server architectures were adapted to the SaaS model by Application Service Providers (ASPs) with limited success. The trend has continued with off-shoring of IT operations to take advantage of cheaper globalized resources provided by Business Process Outsourcing (BPO) vendors. However, these approaches focus on delivering traditional packaged software applications as services. While economies of scale lower cost, solution implementation and operation cycles continue to be complex and time consuming and primarily suitable for large enterprises. More recently, a new breed of software platforms and applications have emerged to take advantage of internet scale architectures to provide ***software on demand***. New on-demand vendors are taking advantage of this model to drive significant adoption of software in the high-volume small and mid-size enterprise markets. In effect the ***software on demand*** model has become synonymous with the ***software as a service*** moniker.

Gartner, IDC and Fortune Magazine have all cited *on demand software* as one of the key megatrends and predicted that up to 40% of all applications will be delivered over the Internet within the next 2-3 years. As a matter of fact, people have already been using service on demand in their daily life, without being aware of it, e.g., even private individuals think nothing of using eBay and PayPal to sell goods to people half a world away, essentially making them small businesses outsourcing their IT needs to a sophisticated global vendor. This is an example of delivering enterprise applications on a needed-base via the Web for the masses.

The on demand model fundamentally alters the economics of software delivery through operational efficiency and performance, infrastructure orchestration and application control. In addition, it changes both the relationships between the software vendor and the customer, and the economics of purchasing and owning software. However, the profitable transition to such an on-demand model will require incumbent client-server application vendors to adapt and migrate to a new breed of technical solutions as well as fundamentally different operation models.

2 Basic Issues and Approaches

Service on demand addresses the cost of software delivery at several levels. First, standardized technology platforms that support internet scale architectures reduce the initial infrastructure costs. Second, a very high degree of automation with focus on availability, security and fault tolerance reduces the human cost of delivering software. Finally, applications that lend themselves to a high degree of modularization and non-disruptive change management allow seamless incremental deployments and pay-as-you go service acquisition.

The technology array used to support service on demand is classified into three categories:

1. **Resource virtualization.** This category of technologies aims to provide a single, consolidated, logical view of and easy access to all available resources in an environment. It is the process of presenting computing resources in ways that users

and applications can easily ascertain values from many types of integrated implementation rather than presenting them in a way dictated by a specific implementation, geographical location, or physical package. A meta-level of implementation views can in some sense describe the overall concept of virtualization. However, resource virtualization, specially at the hardware level alone is insufficient in driving down the cost of delivering service on-demand to create profitable margins.

2. **Service provisioning.** This category of technologies makes the right resources available to the right processes at the right time. The execution resource allocation and application management should support the adequate level of quality of service to meet the minimum SLA requirements contracted with customers. It is the end-to-end capability to automatically deploy and dynamically optimize resources, including servers, storages, network bandwidth, operating systems, middleware, applications, and third-party connections. It requires tools to manage service levels, meter system usage and performance, and billing for service usage, as well as monitor and optimize service provisioning processes.
3. **Application adaptation.** This category of technologies reshapes the existing/new software applications to be able to leverage the virtualized execution environment and internet scale architectures, achieving the maximum degree of performance and flexibility. For example, vendors like RightNow have started to build their software to be delivered as a service. As a result of this effort, the average deployment time of RightNow's call center solution from customer commitment to being live is less than 40 days, including integration and customization. Meanwhile, vendors like Oracle and SAP are both on the way to break up their application packages into modular components to facilitate service provisioning.

However, the service management and resource virtualization technologies alone cannot deliver profitability in the on demand service model. As indicated in a Gartner report, human operations were responsible for more than 50% of process operational failures. Additionally, numerous studies point to the cost of operations as the primary contributor to overall IT costs. As a result, in order to ensure a sustainable level of service quality while maintaining profit margins, we must do three things: First, we must standardize the fundamental infrastructure and building blocks. Second, we must develop specialization and repetition of the key availability, security, performance, problem, and change management processes. If a process is repeated, it can be automated. The next generation of complex software management cannot be based on people; it must be based on processes and automation, which is the key to higher quality and lower cost. Finally, the process can be optimized, meaning data from the process is used to change and improve the process. In summary, an on demand operating environment is an open standards-based, heterogeneous world, integrated, automated and freely enabled with self-managing capabilities.

In addition, we need to develop the operational processes focusing on the support of key service business operations, including configuration management, availability management, performance management, security management problem management, and change management. At SAP, based on our experience of managing thousands of software systems, a set of key management processes has been identified, including an event correlation and root cause analysis process, a disaster recovery management process, a capacity management process, a resource management process, a production assessment process, an escalation management process, an update management process and a proactive problem management process.

To evolve all of the above-mentioned capabilities and deliver more choice in tailoring services to our customers, we at SAP Research Lab are developing a prototype researching an execution and development framework supporting the business of service on demand. We will also study how to codify changes to the existing application packages facilitating the service on demand operation.

3 Additional Challenges and Directions

The basic business model for service on demand tends toward fixing functionality to put an upper limit on cost. But businesses change. In today's dynamic business environment, without the ability to innovate, companies will risk watching more nimble competitors eat into their established markets. Therefore, the business objectives for customers to move to the service on demand model is not just trying to reduce the total cost of ownership, but also more critically, to reduce the time of implementation of new composite services to support their end-to-end business operations. This is because, in the past, most vendors had constrained the ability to change the software, providing very basic offering of their services. As the service business continued to mature, there is a realization that, for larger mid-size companies, there is a need to extend and tailor the applications to the customer's business.

In addition, large global service providers always attempt to define what they called "killer applications." These killer applications were competitive services predicted to be highly utilized by a wide variety of customers. Oftentimes, these applications were time-intensive to create and somewhat costly to maintain. The on demand business ecosystem must afford creators the ability to quickly create and push out to their users, virtually eliminating risk from the large enterprise in determining what best killer application was the target, so as to provide to their customer base the most robust service.

Based on the observations mentioned above, it becomes clear that the ability of a service provider to enable the quick response to ever-changing customer demands will determine who survives in today's rapidly changing marketplace. It requires a new set of composite service development technology fabric that can adapt on demand business to these ever-changing requirements. Currently, SAP is exploring a new category of services, called ***tool on demand***, to provide a hosted development environment for composite service development, deployment, monitoring and maintenance in its hosted environment in addition to its hosted service on demand environment.

Last, as adoption of its offerings grows, SAP may also consider expanding its services to include offering its NetWeaver application server as a service. This next level of outsourcing, called ***execution environment on demand***, offers customers the flexibility to not only use the framework to tailor the software, but also use independent software vendors and custom-written applications that run on SAP technology to provide more flexibility to support their business needs.

References

1. Chou, T.: The hidden cost of Software. ASPNews (May 2003)
2. Fellenstein, C.: On Demand Computing, IBM Press (2005)
3. Murphy, B., Gent, T.: Measuring system and software reliability using an automated data collection process. Quality and reliability engineering international.
4. Oppenheimer, D., Ganapathi, A., Patterson, D.: Why do Internet services fail, and what can be done about it?. In: Proc. Of the 4th USENIX Symposium on Internet Technologies and Systems, Seattle, WA (March 2003)
5. Rowell, J.: A step-by-step guide to starting up SaaS operations. OpSource (2004)