



Innovative Applications of O.R.

Cost-based decision-making in middleware virtualization environments

Kaushik Dutta*, Debra VanderMeer

College of Business, Florida International University, Miami, FL, United States

ARTICLE INFO

Article history:

Received 24 April 2009

Accepted 2 October 2010

Available online 29 October 2010

Keywords:

Computing science

Virtualization

Resource assignment

System design

ABSTRACT

Middleware virtualization refers to the process of running applications on a set of resources (e.g., databases, application servers, other transactional service resources) such that the resource-to-application binding can be changed dynamically on the basis of applications' resource requirements. Although virtualization is a rapidly growing area, little formal academic or industrial research provides guidelines for cost-optimal allocation strategies. In this work, we study this problem formally. We identify the problem and describe why existing schemes cannot be applied directly. We then formulate a mathematical model describing the business costs of virtualization. We develop runtime models of virtualization decision-making paradigms. We describe the cost implications of various runtime models and consider the cost effects of different managerial decisions and business factors, such as budget changes and changes in demand. Our results yield useful insights for managers in making virtualization decisions.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Virtualization is a growing part of the overall information technology market. The 451 Group predicts that the virtualization market overall will grow from \$2.2B in 2008 to nearly \$11B in 2013 (Kusnetzky, 2009). A particular type of virtualization, *middleware virtualization*, also known as *Platform as a Service*, allows an application to run on a set of middleware resources such that the resource-to-application binding can be changed dynamically on the basis of each application's resource requirements. In other words, middleware resources can be deployed and un-deployed to support an application's workload needs as demand rises and falls. Here, the term *middleware* refers to application middleware software platforms (e.g., WebSphere (IBM Inc., 2006b), IIS (Microsoft Inc., 2009), WebLogic (BEA Systems Inc., 2004), others) that provide generic application services (e.g., database connection management, thread pool management, naming and directory services, and other application support services), as well as other application support systems, e.g., database servers or transaction servers. Applications written to run on application middleware platforms need only implement their specific business logic to take advantage of the generic services available from the middleware, i.e., they need not re-implement the same generic functions themselves.

In a middleware virtualization scenario, *middleware stacks* consisting of middleware software and the operating system and

hardware resources supporting it (e.g., as depicted in Fig. 1), can be provisioned as needed to support any application written for the platform (i.e., Java EE platforms can support Java-based applications written to the Java EE standard, while IIS can support .NET-based applications). Indeed, by installing multiple middleware frameworks on each middleware stack instance, it is possible for a middleware stack to support any application simply by starting the appropriate middleware software and deploying the application.

Middleware virtualization differs significantly from server virtualization, which allows multiple guest operating systems to run on a single host machine, accessing a common set of hardware resources (with the attendant additional delays associated with the extra layer of indirection imposed by the guest operating system). In contrast, middleware virtualization technologies allow multiple applications to share a pool of middleware stacks. This enables data center managers to dynamically allocate and deallocate application resources without interrupting the runtime processing of an application.

Typically, several application clusters supported by middleware stacks reside within a data center, where each application cluster supports a single application. Instead of permanently sizing application clusters for peak loads, managers should be able to reallocate resources such as application servers, database servers and storage servers in response to the current demand for each application. Because different applications are unlikely to experience peak demand simultaneously, managers can save money by reducing the total number of units deployed and moving idle resource units from cluster to cluster as demand dictates. If total demand exceeds the available resources, applications can be prioritized to ensure that critical systems do not starve. Then, if increased

* Corresponding author. Tel.: +1 305 348 3302.

E-mail addresses: kaushik.dutta@fiu.edu (K. Dutta), debra.vandermeer@fiu.edu (D. VanderMeer).



Fig. 1. Middleware virtualization resource stack.

demand persists, administrators can boost their capacity by adding new resource units to the existing infrastructure pool.

Consider the case of a major credit card company (the name has been withheld in order to honor confidentiality agreements) in New York that has several applications running in multiple data centers. The merchant credit (MC) and credit card bill check-payment (CCB) applications are two important applications in the company's data centers. These applications use various resources like database servers, remote web services, security services, application servers and storage services. A number of instances of each of these resources are located in multiple data centers located in four geographically separated cities in the United States. At any given time the number of applications running in these data centers ranges between 90 and 120. Over the course of a day, the MC application reaches its peak load between 8 a.m. and 12 noon, when merchant activity is heavy. The CCB application reaches peak loads between 4 p.m. and 8 p.m., when large numbers of customers check and pay credit card bills. During peak loads, both applications combined use approximately 85% of available physical resources. During non-peak periods, these applications use on average 20–30% of the resources. Clearly, there is often a significant under-utilization of resources. This scenario is typical of IT data centers.

In order to better utilize resources, what is needed is a mechanism to allow the MC and CCB applications to temporarily expand within a common set of resources during their peak loads. Such a mechanism would increase the average utilization of resources and enable the company to run a larger set of applications on the same set of resources. *Middleware virtualization technologies present the possibility of achieving this goal.* In this context, the application set must be mapped to a set of specific middleware stack resources. Here, the following question arises: *on what basis should we decide which application to assign to which middleware stack?* For many enterprises, the main motivation for deploying middleware virtualization technology is to manage costs. Thus, the problem of resource-to-application allocation needs to be tackled in both a cost-effective and QoS-friendly manner. *The key problem businesses face in using virtualization technology is how the virtualized resources can be utilized to promote cost and business priorities* (Business Wire, 2007). There is a gap between technical know-how and the achievement of business goals that has not been addressed in detail in the literature – while managers would like to ensure minimum-cost resource-to-application allocations, they currently have no way to determine the cost implications of allocations due to the complexity of the decision.

This complexity is based on a number of factors, starting with the difficulty of optimizing over a wide variety of applications and resources. This problem is exacerbated by the complexity introduced by multiple geographically separate data centers, where each location has a different cost profile, based on a variety of factors. We cite two examples of cost-differentiating factors, the costs of powering data centers and the cost of human IT resources, below.

Power usage is a significant cost differentiator – data centers are notorious power consumers, both to run the servers as well as to cool the server rooms. In fact, recent research indicates that the cost of powering and cooling a data center actually exceeds the cost of the IT equipment the data center houses (Belady, 2007). Further, power costs differ substantially across regional areas – in January 2010, the average cost per kilowatt-hour for commercial use was \$0.15 in the state of New York, but only \$0.074 in Oregon and \$0.063 in North Dakota (US Energy Information Administration, 2010). Such disparities in power costs have led many data center operators, e.g., Google and Microsoft, to consider locations with low-cost power sources, e.g., based on hydrodynamic (Scheier, 2007) or geothermal (Hancock, 2009) sources.

The cost of human IT resources also varies significantly on a regional basis. This is clearly demonstrated by the fact that most major IT salary surveys report average on a regional basis – ComputerWorld reports average salaries on a multi-state regional basis (ComputerWorld, 2009), while salary.com provides wage estimates by metropolitan regional area (Salary.com, 2010). For example, based on salary.com data, a systems administrator in New York costs 14% more than one in Portland, Oregon.

These difficulties are compounded by the problem of varied incentives – application users want fast response times regardless of cost, application owners want the fast response times without spending too much money, and data center managers seek to reduce the cost of running all applications within their service agreements, regardless of ownership. These competing incentives raise the question – whose incentives should be paramount in the resource-to-application mapping decision?

These incentives can play out in a variety of managerial scenarios, with slightly differing implications based on how application owners are charged for their applications. Essentially, application owners are responsible for the final cost, whereas the resource managers are responsible for managing costs.

In the first scenario, in-house IT staff make allocation decisions, and IT operations costs are not tied back to application resource usage. IT staff are incentivized to reduce costs across the board to minimize budget requirements for upper management. Application owners want maximum performance for their applications, without regard to cost.

In the second scenario, in-house IT staff make allocation decisions, but with a charge-back policy (McKinnon and Kallman, 1987) in place to tie application usage costs back to the application owners. IT staff is still incentivized to reduce costs across the board, but application owners are now incentivized to minimize cost of owned applications, regardless of cost or performance impacts on other applications.

In the third and final scenario, applications are hosted in out-sourced data centers (e.g., perhaps Amazon's Elastic Compute Cloud (Amazon Web Services, 2010)), where application owners are charged based on actual usage (Stone and Vance, 2010). Data center managers have an incentive to maximize profit, while application owners have an incentive to minimize their own costs.

In all three scenarios, application users will tolerate little in the way of delays, regardless of application workloads. However, these stakeholders have little or no control over the main parameters that drive the allocation decision – the cost of resources and allocation events, application budget limits, and demand for applications – and no way of understanding the interactions between these parameters without assistance. In this work, we take a first step toward helping stakeholders understand implications of virtualization, and develop a set of insights to help them understand what they can expect as the values of these parameters change.

In such complex scenarios, it is virtually impossible to make resource allocation decisions without a formal framework, thus motivating this research. We attempt to address this gap, providing

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات