



Building a Dynamic Infrastructure with IBM Power Systems: A Closer Look at Private Cloud TCO

Scott A. Bain
Fehmina Merchant
Bob Minns
John J Thomas
IBM SWG Competitive Project Office
February, 2010

Table of Contents

Table of Contents	2
Executive Summary	3
A Virtuous Circle to Reduce IT Costs	4
Take Cost Out Through Virtualization	5
Labor and the Server Provisioning Lifecycle	8
Standardization Helps Lower Labor Costs	12
Automation Can Help Lower Labor Costs Even Further!	15
Putting It All Together	19
Summary	21

Executive Summary

Many companies are finding their need for greater business agility being frustrated by an increasingly costly and rigid IT infrastructure. The culprits are many. Maintenance of the current environment accounts for over 70% of the IT budget, leaving less than 30% available for new projects. Annual operational costs (power, cooling, and labor) of distributed systems and networking exceed their acquisition cost by 2-3X and continue to climb. Utilization rates of these commodity servers hover around 5-15% on average, leading to excess capacity going to waste. Time to provision new servers can be as long as six months, hampering lines-of-business efforts to quickly respond to competitive threats or new opportunities. As a result, LOB units are beginning to go outside the datacenter to public cloud providers like Amazon in hopes of lowering their costs and improving their responsiveness. To avoid disintermediation, IT needs to re-invent the datacenter by moving towards a more dynamic infrastructure. One that takes out cost through the use of virtualization to improve utilization levels with a commensurate reduction in power consumption. One that embraces a private cloud model that uses standardized workloads and service automation to dynamically provision IT services in minutes/hours rather than months (and at lower cost) via self-service portals. Customers can build such an environment using IBM's Power Systems servers coupled with Tivoli service management software.

This paper examines the Total Cost of Ownership (TCO) for a dynamic infrastructure built around private cloud services and compares it to public cloud alternatives as well as conventional one-application-per-distributed server models. The results show that private cloud implementations built around new Power 7 based servers can be up to 90% less expensive than public cloud options over a three year period and around 80% less than a distributed stand-alone server approach.

A Virtuous Circle to Reduce IT Costs

Figure 1 depicts a three-pronged approach to how customers can reduce their overall IT costs through the implementation of a dynamic infrastructure built on virtualization, standardization, and automation.

A Virtuous Circle To Reduce I/T Costs

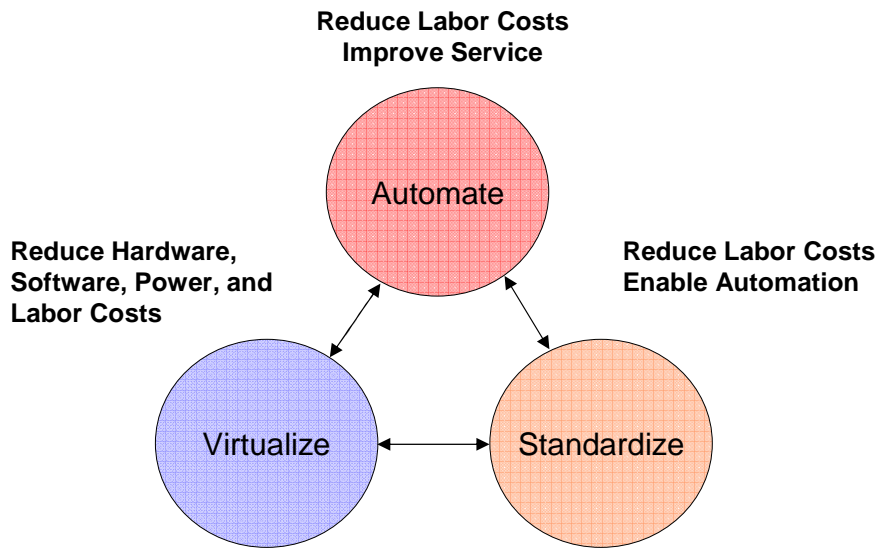


Figure 1

Although enterprise usage of these three approaches is expected to rise to 50% by 2012, adoption thus far has been limited to around 12%¹. One of the reasons for the tepid adoption rate so far has been an inability to quantify the impact these capabilities have on reducing IT costs. Customers want a better understanding of the savings they can anticipate through the application of these technologies before committing resources to their implementation. To that end, the rest of this paper takes a look at each approach in more detail and provides some guidelines on how customers can go about estimating cost savings for their own company.

¹ Internal IBM Cloud study 2009

Take Cost Out Through Virtualization

A recent IBM internal study of its nearly 4000 distributed servers showed annual operational costs attributed to each server to be over \$34,000, with almost 90% due to software maintenance and systems administration. It stands to reason that reducing the number of physical servers to fewer, larger, more capable machines can serve to greatly reduce these costs. Indeed, the virtues of virtualization to accomplish this have been well-publicized. What has proven to be more elusive, however, is the quantification of these benefits. How many workloads can actually be consolidated onto a given platform while maintaining acceptable service level agreements? Which platform gives you the greatest economy of scale, producing the lowest cost per virtual machine image/workload?

To answer this question, the CPO evaluated three different alternatives for running 75 Linux workloads as shown in Figure 2 below:

Dynamic Infrastructure - Compare Options For Deploying Heavy Workloads

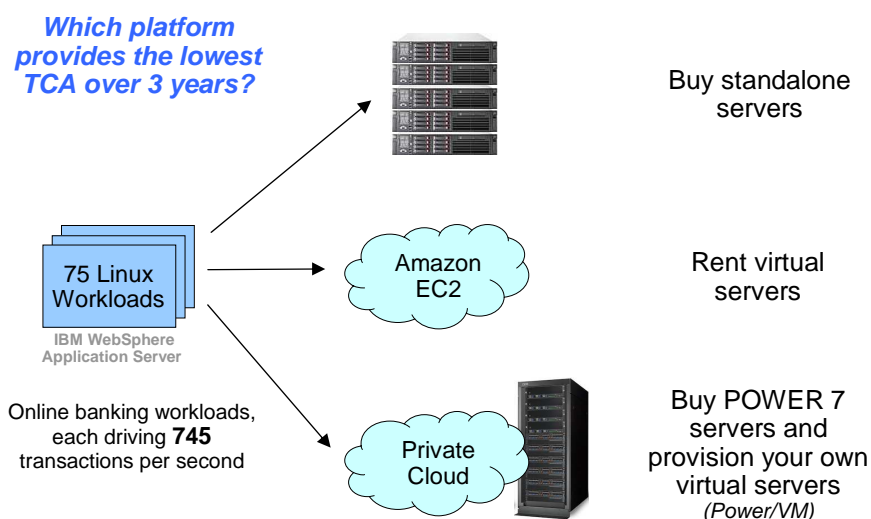


Figure 2

The workload in question was an online banking application built using IBM WebSphere Application Server (WAS) and requiring an average throughput of 745 transactions per second. We first ran this workload on an 8-core Intel Nehalem server (2.93 GHz), which resulted in an average utilization of 12%. A VM image of the online banking application was then created to see how many images could be placed on virtualized x86 and Power Systems servers. Multiple running instances of this VM image were added incrementally to the servers until it could no longer handle any additional throughput. Because

of the CPU utilization levels, we were unable to run more than one workload on an x86 system with a hypervisor. For the Power Systems servers, we found that it would take four Power 750 servers (32 cores each) and two Power 770 servers (64 cores each) to handle the 75 VM images.

However, these consolidation ratios assume “flat-out” operation. In practice, not all servers/images are used all the time. Experience from customers and public cloud providers have shown typical usage patterns of 14 hours per day (59%). This means in theory, we should be able to reduce our hardware requirements by 1.7X in order to support our 75 workloads. Applying this factor to our Power Systems configurations means we only need two Power 750 servers and one Power 770 server as shown in Figure 3.

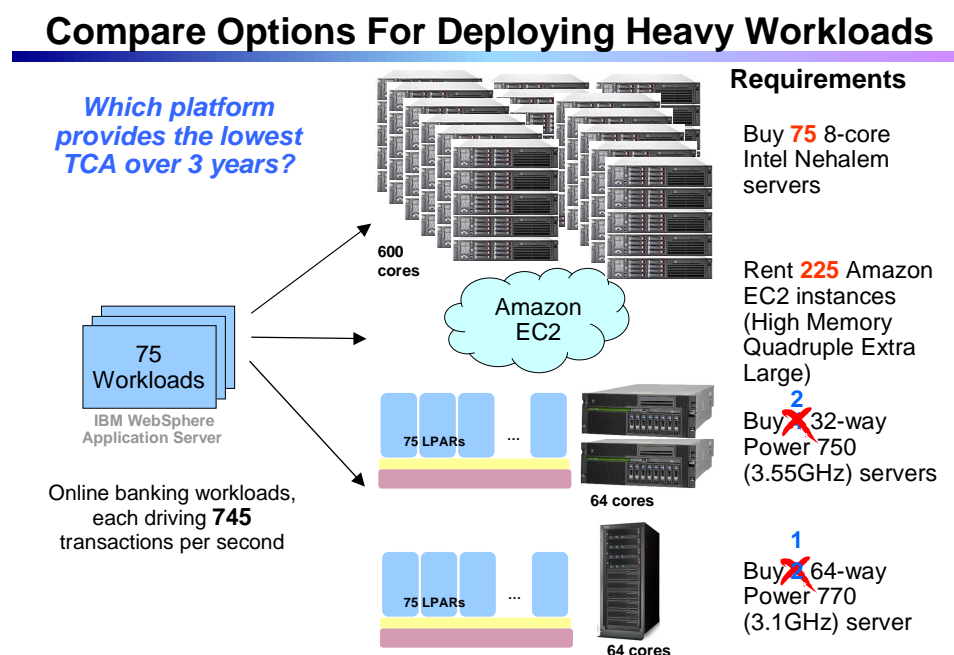


Figure 3

In selecting the appropriate Amazon EC2 instance size required to handle our workload, we found that we would need to cluster three High Memory Quadruple Extra Large instances in order to match the performance we achieved with a single 8-core Intel Nehalem server. As a result, this drove the total number of paid EC2 instances needed up to 225 (75 x 3).

When you look at the four options from a Total Cost of Acquisition (TCA) perspective, the Power Systems servers are the lowest cost alternative, 78% lower than the stand-alone servers and less than one-tenth the cost of the public cloud option (Figure 4):

Hardware And Software Costs Per Image for Linux Workloads (3 Yr TCO)

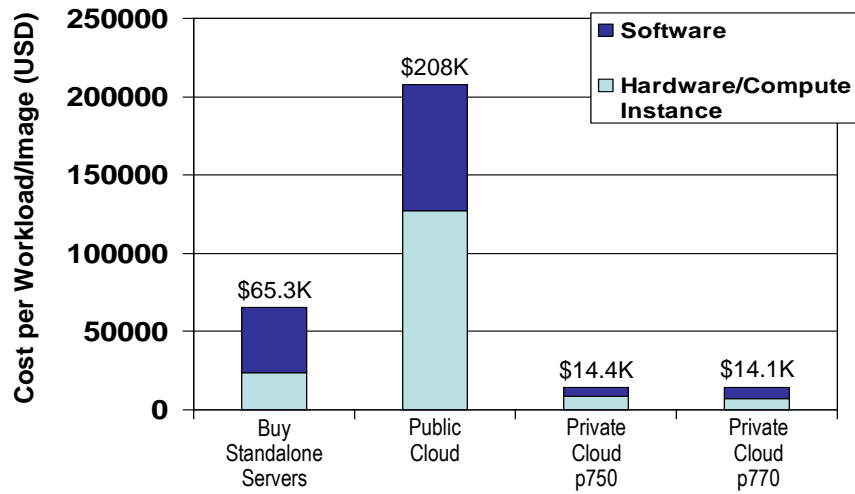



Figure 4

Labor and the Server Provisioning Lifecycle

Now that we have a handle on the impact of virtualization on hardware and software costs, what about the effects of labor? Any discussion of labor needs to start with a process that describes the tasks associated with the acquisition, deployment and retirement of servers. Servers are first planned and acquired, then they are handed over to administrators to configure, set up and deploy. The operating systems software is installed, Hypervisors are configured, virtual servers configured, security profiles for users established, and the server is tested and deployed into production. Monthly maintenance continues including routine patches and fixes, and upgrades. The servers are ultimately cleansed and retired from service.

Figure 5 below depicts this provisioning lifecycle approach. It includes some procurement functions, set up and deployment functions, maintenance, troubleshooting and ultimate tear down. The labor categories included setup and tear down costs as well as the ongoing monthly maintenance and troubleshooting costs for physical servers and software virtual images.

Server Provisioning Lifecycle: Labor Components

 focus of labor model

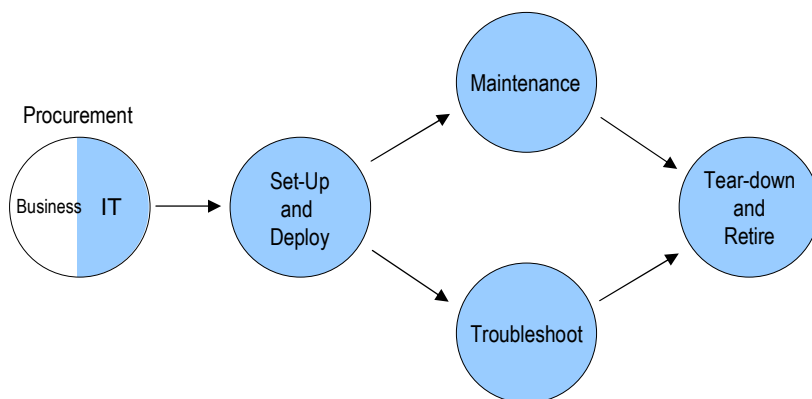


Figure 5

To quantify the impact of labor, we developed a labor model for servers (Figure 6). The formula represents the total labor hours ascribed to the management of a server environment as comprised of the hours spent managing a physical server over its lifetime plus the hours spent managing the software images over their lifetime. Total hardware server labor hours (H) include the set up and deployment hours representing one-time events such as sizing and configuring workloads, and testing of a physical

computing element. They also include hours for scrubbing of servers, decommissioning, maintenance and troubleshooting for physical servers over the analysis period. Total software labor hours (S) include both the initial installation labor associated with the software stack or virtual images on the physical server along with ongoing maintenance and troubleshooting over the assessment period. These tasks include periodic patching and upgrades, associated testing functions, analysis of errors, debugging, fixes, testing and reboots.

Labor Model For Servers

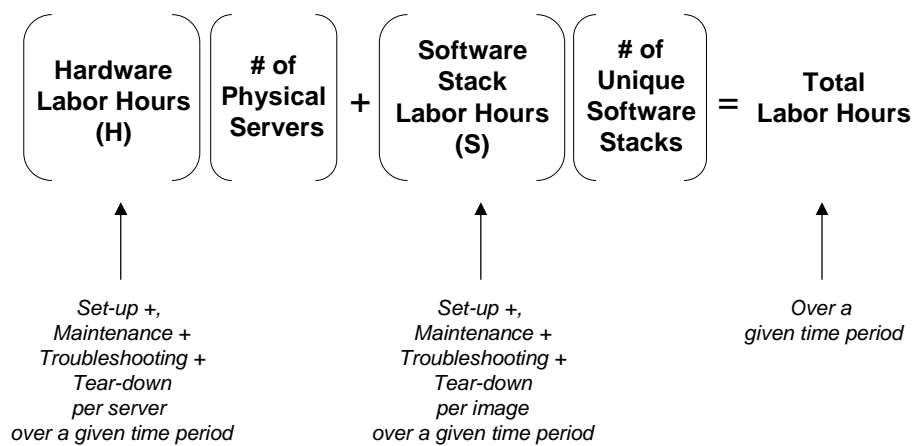


Figure 6

Solving this equation for a stand-alone x86 environment gives us a picture of how much labor was required *before* virtualization. Similarly, solving the equation for the Power Systems-based environment gives us insight into the total hours needed *after* virtualization. Fortunately, we have data from customer case studies that can help us evaluate both equations as shown in Figure 7:

Using Customer Data to Derive Average Number of Servers per Administrator

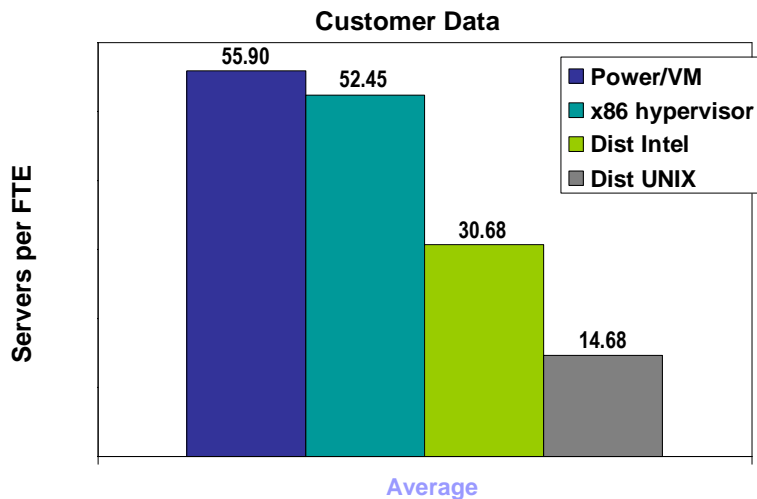


Figure 7

For the stand-alone x86 server case, this works out to be 30.7 servers/FTE, while the virtualized Power Systems server case turns out to be 55.9 servers per administrator.

We then wanted to calculate the portion of *FTE labor needed to manage a server*.

Calculating the FTEs per server for stand-alone and virtualized x86-based servers

- **Stand-alone x86** data shows 30.7 servers managed per FTE,
 $1/30.7 = .0326$ FTE's needed per server
- **Virtualized Power Systems** data shows 55.9 virtual servers managed per FTE,
 $1/55.9 = .0179$ FTE's needed per server

Next, we wrote equations to represent the total FTE hours required to manage our 75 Linux workloads over 3 years for both stand-alone and virtualized Power Systems platforms.

We assumed 6,240 hours or 52 weeks per year, 8-hour days for 3 years.

FTE hours needed to manage 75 workloads over 3 years:

Multiply **FTEs needed per server * total hours over 3 yrs. * number of software images**

- $.0326 * 6,240 * 75 = 15,259$ hours needed for all stand-alone x86 servers
- $.0179 * 6,240 * 75 = 8,372$ hours needed for all virtualized Power Systems servers

On balance, this shows a virtualized Power Systems-based environment requires 45% less total labor hours to manage 75 Linux workloads over 3 years than the stand-alone x86 scenario. But what percentage of that time can be attributed to managing the hardware (H) vs. managing the software images (S)? From our earlier analysis, it took 75 8-core Intel Nehalem servers to handle 75 Linux workloads. For the virtualized Power Systems case, we found that you could handle 75 workloads on a single 64-core Power 770 server.

Thus, we are left with the following equations:

$$(1) \text{ Stand-alone x86} \qquad 75H_i + 75S = 15,259$$

$$(2) \text{ Virtualized Power Systems} \qquad 1H_p + 75S = 8,372$$

While the amount of time to install software on either a Power-based or x86 server is about the same, our own hands-on usage of a Power-based server showed that it took roughly twice the amount of hours to administer as a stand-alone x86 platform. Thus, substituting $H_p = 2H_i$ allows us to solve the equations for their respective H and S values.

Subtracting equation 2 from equation 1 to solve for H_i , H_p , and S:

$$H_i = 94 \text{ hours}$$

$$H_p = 188 \text{ hours}$$

$$S = 109 \text{ hours}$$

Therefore, over a 3-year planning horizon, the total hardware labor (H_i) to manage one x86 server is 94 hours while the Power Systems server labor hours (H_p) requires twice that (188 hours). The cost to manage a single software image (S) is 109 hours.

Standardization Helps Lower Labor Costs

Servers need a full load of software to run a workload. This includes not only an operating system, middleware and the application itself, but also things like patches and configuration specifications. We refer to all of this software as a “software stack”. Without controls, the variety of software stacks tends to proliferate, driving up labor costs. For example, many stacks will have different levels of software installed, along with different patches and product selections. The standardization of these software stacks, however, can reduce labor costs. Uniformity reduces the number of unique stacks to manage and allows for greater re-use. We refer to the concept of re-using a standard software stack as “cloning”. The question is, how can we quantify the material impact standardization has on reducing labor costs?

To estimate this, we applied a cloning factor to our original equation as shown below in Figure 8:

Use of Standardized Stacks Can Drive Down the Labor Hours for Software Images

$$\left(\text{Hardware Labor Hours} \right) \left(\# \text{ of Physical Servers} \right) + \left(\text{Software Stack Labor Hours} \right) \left(\frac{\# \text{ of Software Images}}{\text{Clone Factor } C} \right) = \text{Total Labor Hours}$$

This is the number of unique stacks

**Where C = average number of copies
deployed for each unique software stack
(from 1 to 75 in our example)**

Figure 8

Solving this equation for the virtualized Power 770 environment discussed earlier in the paper yields the following:

$$1H_p + 75(S/C) = \text{total labor hours}$$

Since we already know H_p and S from our previous calculations, we can substitute those values, resulting in the following:

$$1(188) + 75(109)/C = \text{total labor hours}$$

Expressing the formula this way allows us to play some “what if” games with the clone factor (C) to gauge the impact of standardization on total labor hours. For example, applying a clone factor of five would mean that out of 75 servers there are 75/5 or 15 unique images deployed, of which the rest are duplicates of the original five unique templates. This reduces the overall labor hours from the original virtualized Power Systems case of 8,372 to 1,823, a reduction of 78%!

The graph below in Figure 7 shows the labor savings to be had as you adjust the clone factor “C” between no clones (1) and 75 clones (75).

Benefit Of Cloning Factor On Software Labor Costs In A Virtualized Environment

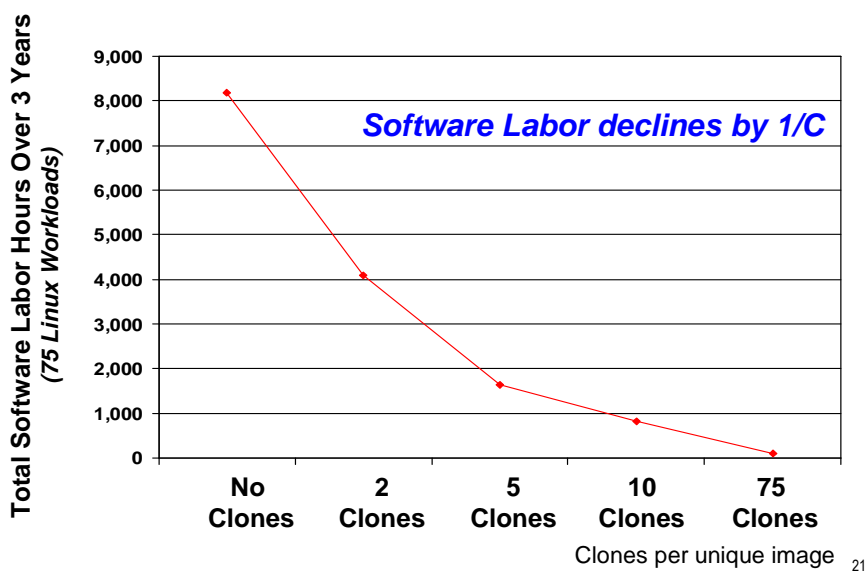


Figure 9

As you can see from the curve, total software labor hours decline by roughly the inverse of the cloning factor. Based on this revised labor model that takes into account the use of clones, we can make the following observations as shown in Figure 10:

Effects of Virtualization and Standardization On Labor Costs

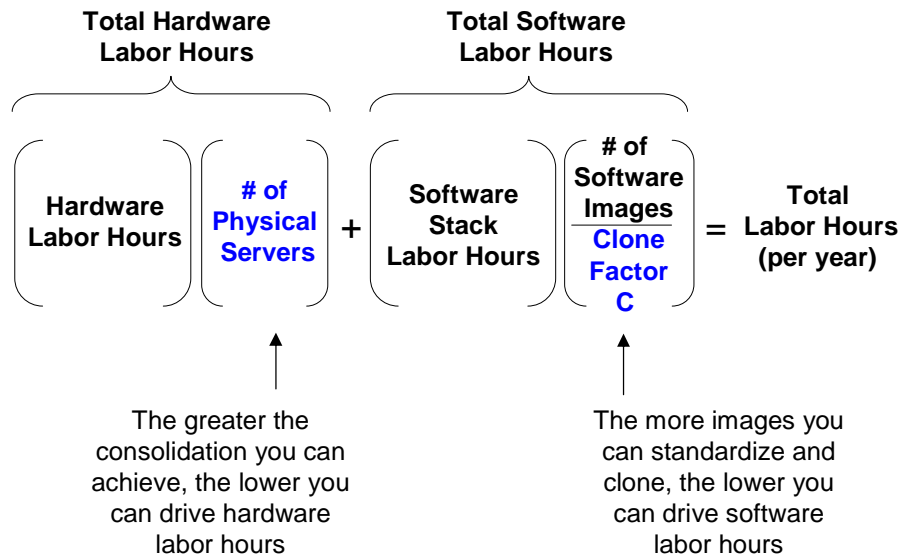


Figure 10

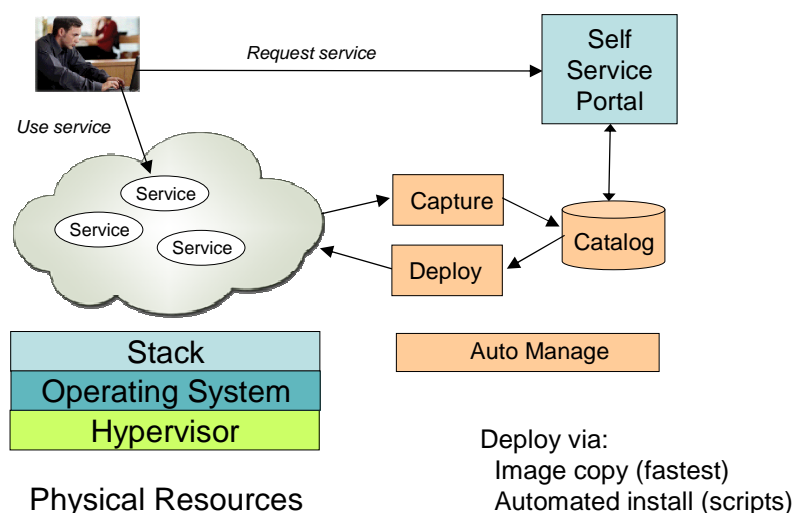
One of the levers in reducing labor costs is to reduce the number of physical servers you have to manage. Put another way, **the more workloads you can consolidate on a given platform, the more you can lower your labor costs**. This makes larger, more scalable systems like the IBM Power Systems family an ideal virtualization and consolidation platform for implementing private clouds.

Another lever is the degree to which you can use workload standardization and cloning in your environment. Simply stated, **the higher the clone factor, the greater the reduction in labor costs associated with deploying and maintaining software virtual images**.

Automation Can Help Lower Labor Costs Even Further!

While virtualization and standardization can go a long way in reducing overall labor costs, the task of deploying a software stack as a VM image onto a virtualized server has historically been a highly labor-intensive task. For instance, one has to first deploy and configure the OS along with all requisite patches. After that, the administrator has to install and configure the application server and all its constituent components (e.g. HTTP server, etc.) as well as patches and other fixes. For applications requiring a database, that becomes yet another piece of middleware that needs to be installed and configured. Then there is the application itself. Collectively, deploying and testing a complete application manually can require days or weeks to accomplish depending upon its overall complexity. In a private cloud environment, this kind of turnaround is untenable. The use of automation promises to reduce the labor required dramatically. Figure 11 depicts such an environment with a self-service portal that enables users to request IT services on demand and have the request fulfilled in minutes/hours versus days/weeks/months.

Automated Self Provisioning Further Reduces Labor Costs And Speeds Up Delivery



51

Figure 11

In this environment, services are initially defined/created and stored in a service catalog. Requesters can then browse the catalog to find and select the desired service. After submitting the request, it gets routed for approval and then fulfilled by the underlying infrastructure. The software needed as part of the overall service is typically deployed in one of two ways: image copy (the fastest) or via automated

install using scripts. When the service is no longer needed, the affected resources are freed up so that they can be claimed by other subsequent requests. In order for all of this to work seamlessly and transparently to the user, there needs to be automated management software that undergirds each step in the process.

IBM offers Tivoli Service Automation Manager (TSAM) to manage this cloud services lifecycle and deliver request-driven provisioning for a private cloud environment. It leverages Tivoli Service Request Manager (TSRM) to provide a self-service UI for users to search against the catalog and select the desired service. It also utilizes Tivoli Provisioning Manager (TPM) to provision hardware and software resources according to best practices to satisfy the service request (Figure 12).

Example: IBM Tivoli Service Automation Manager (TSAM) Delivers Fast Self-Service Provisioning

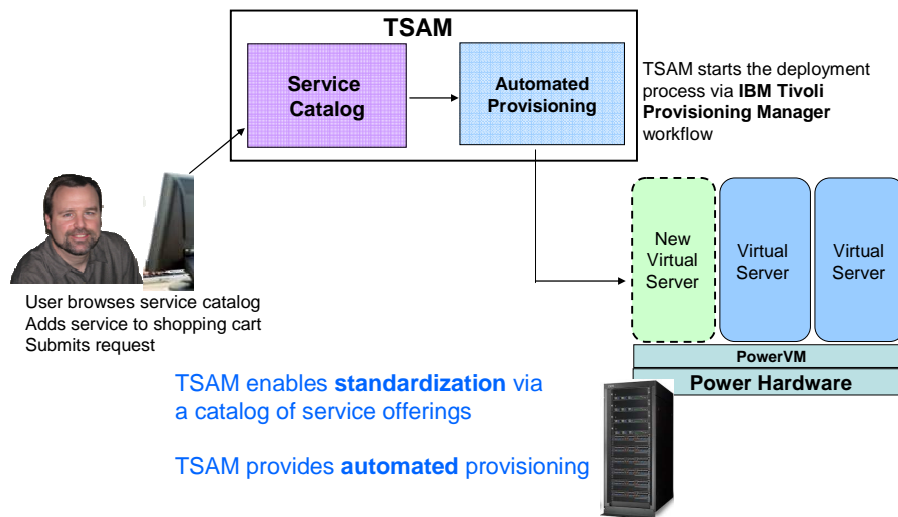


Figure 12

To help assess the extent to which the use of TSAM can reduce labor hours, we conducted a hands-on study as shown on Figure 13 below:

Deployment Study On The Labor Benefits Of Self-Service Provisioning and Automated Install

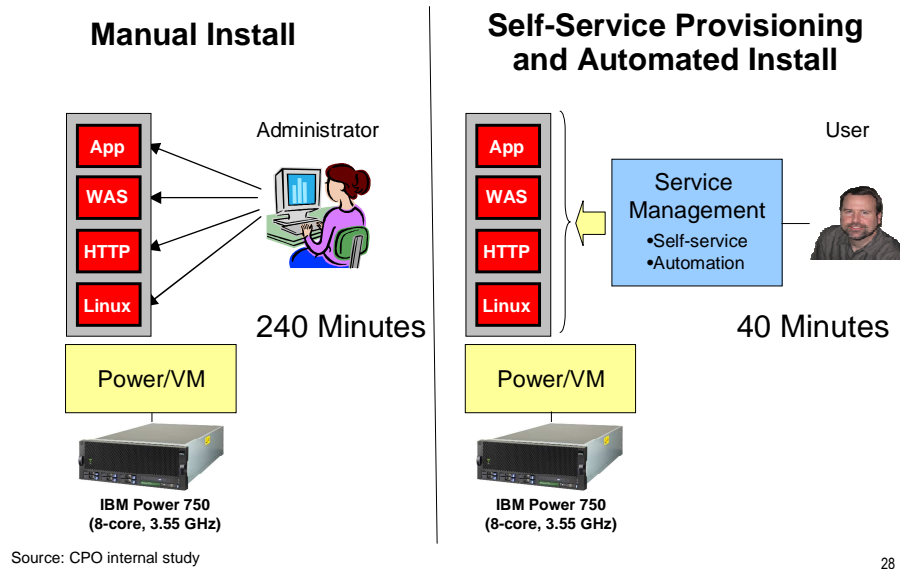
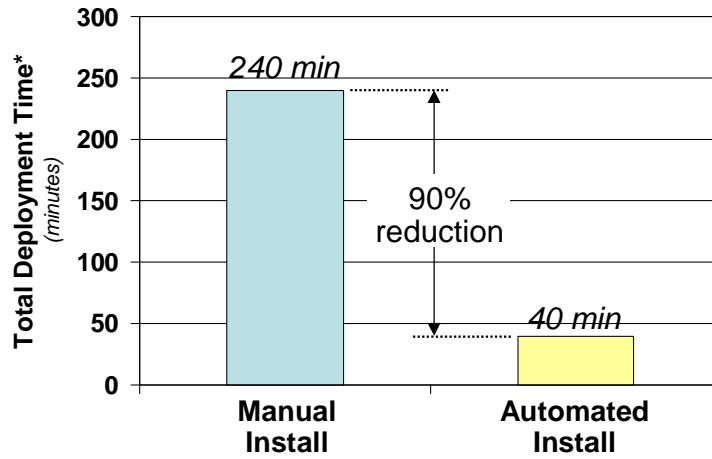


Figure 13

This study tracked the time it took to deploy and instantiate a WebSphere-based application on a virtual server using Power/VM. We captured metrics for doing this manually as well as using TSAM. The results from this study show that the use of automation via TSAM can reduce software image labor hours by as much as 90%! (Figure 14):

Benefit Of Automated, Self Provisioning On Labor Costs



Applying this labor savings ratio reduces Software Labor (\$) from 109 to 11 for each VM image!

* Excluding network transmission time

Figure 13

Putting It All Together

As our analysis shows, there are significant labor savings to be had through the use of virtualization, standardization, and automation. For our example of 75 Linux workloads over three years, virtualization by itself yields a 45% reduction while standardization alone reduces labor hours up to 78% with just a modest clone factor (C=5). Using Tivoli Service Automation Manager for automation in conjunction with Power/VM hypervisor on a Power Systems server yields a reduction of 90%. Taken collectively, companies can reduce their labor costs by up to 97% compared to a traditional stand-alone x86 environment and manual deployment methods (Figure 15):

Total Hardware and Software Labor Hours for 75 Linux Workloads Over 3 Years

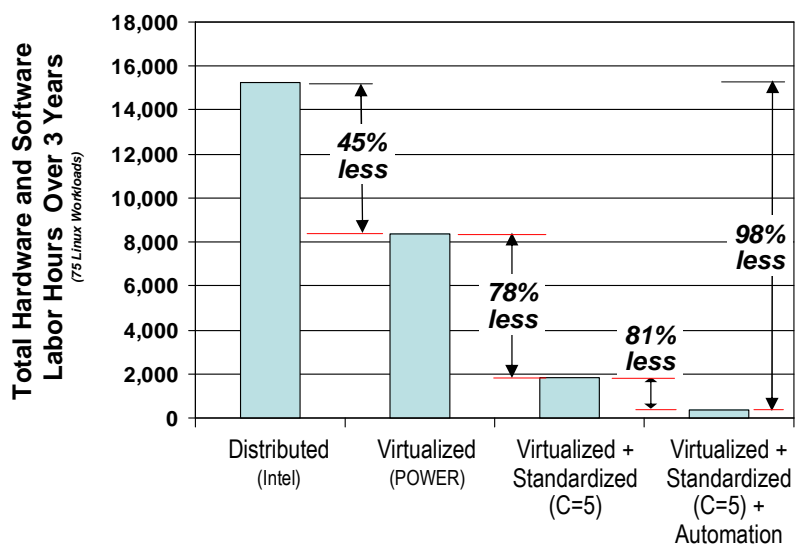


Figure 15

Now that we have been able to quantify the labor savings through the use of virtualization, standardization, and automation, we need to combine these with our earlier hardware and software numbers in order to show a complete cost picture. As shown in Figure 16, you see that the Power Systems 750 and 770 server options come in at the lowest cost per image over three years for our 75 workloads at \$23.4K and \$24K, respectively. This works out to be a savings of almost 80% compared to the stand-alone x86 server alternative and over 90% for the public cloud option.

Let's Put It All Together In Our Example- Cost Per Image for Linux Workloads (3 Yr TCO)

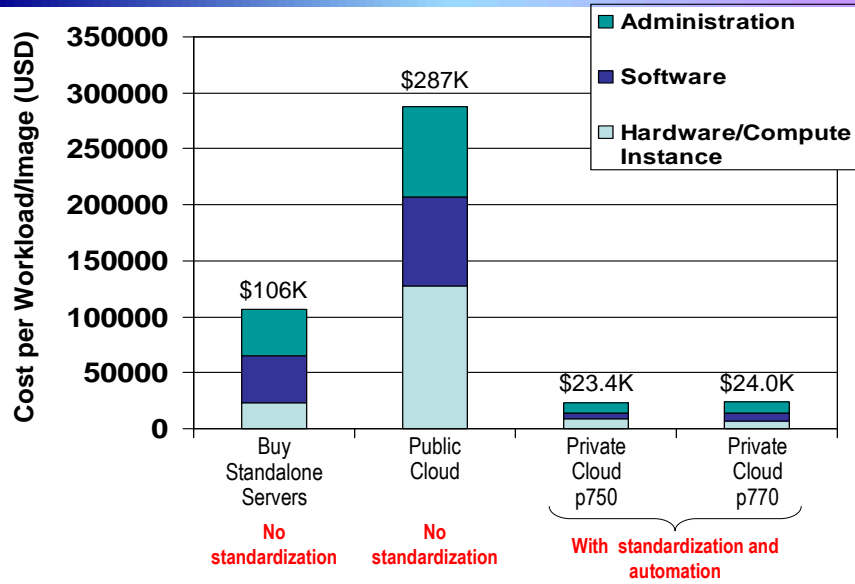


Figure 16

Summary

Escalating business requirements will continue to drive companies toward datacenter transformation. This includes pursuing ways to take costs out of their existing infrastructure through the use of virtualization, standardization, and automation. The labor model described in this paper can be used to estimate potential savings for a number of different deployment scenarios and technology choices. In our example, we chose to highlight the advantages of using IBM Power Systems servers in conjunction with Tivoli service management software as a means to deliver a cost-effective private cloud environment. Some of the benefits that can be expected include:

- **Private clouds built on IBM Power Systems servers and Tivoli service management software can be up to 80-90% less expensive on a cost/image basis than stand-alone x86 servers or public cloud alternatives**
- **The greater the consolidation you can achieve, the lower you can reduce total physical server labor hours**
- **The more images you can standardize and clone, the lower you can reduce software image labor hours**
- **The use of Tivoli Service Automation Manager can reduce labor hours for a unique software image by up to 90% compared to manual deployment on an IBM Power Systems server**

© Copyright IBM Corporation 2010

IBM Corporation
Software Group
Route 100
Somers, NY10589
USA

Produced in the United States

February 2010

All Rights Reserved

IBM, the IBM logo, DB2 and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Amazon EC2 is a registered trademark of Amazon Web Services in the United States and/or other countries.

Intel and Xeon are registered trademarks of the Intel Corporation in the United States and other countries.

Other company, product or service names may be trademarks or service marks of others.

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in these materials may change at any time at IBM’s sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.