White Paper
Intel Information Technology
Computer Manufacturing
Server Virtualization

Implementing Virtualization in a Global Business-Computing Environment

Intel IT planned, engineered, and has begun deploying a virtualized business-computing production environment at several data centers, a rollout that will continue through 2008. Our initiative has already confirmed anticipated virtualization benefits such as faster, more automated deployment. We are initially consolidating older servers running applications that are not mission-critical; we see opportunities to achieve 16:1 consolidation ratios using two-socket virtualization hosts based on Quad-Core Intel® Xeon® processors.

Bill Sunderland and Stephen Anderson, Intel Corporation

April 2008

IT@Intel

Executive Summary

Intel IT planned, engineered, and has begun implementing a virtualized business-computing production environment. Our multi-year initiative has already confirmed virtualization benefits including server consolidation ratios between 10:1 and 16:1, faster deployment, and faster recovery.

We see opportunities to achieve 16:1 consolidation ratios using two-socket virtualization hosts based on Quad-Core Intel® Xeon® processors.

Server virtualization and consolidation promises to help organizations reduce costs, increase agility, and reduce energy consumption. However, implementing virtualization is complex and can appear daunting.

To implement virtualization, we:

- Performed analysis that showed even low consolidation ratios resulted in a positive return on investment (ROI).
- Analyzed current resource utilization, available platforms, and risk factors to determine our optimum consolidation ratios.
- Designed a virtualized architecture that includes hosts, storage, backup and restore (BAR), networking, and management.
- Planned and executed required engineering tasks and business processes.

We are initially focusing on consolidating older servers running applications that are not mission-critical. We see opportunities to achieve 16:1 consolidation ratios using two-socket virtualization hosts based on Quad-Core Intel® Xeon® processors.

We have already deployed several hundred virtual machines (VMs) and expect that our worldwide business-computing production environment will grow to about 1,200 VMs at several data centers worldwide by the end of 2008.

Contents

Business Challenge

Server virtualization and consolidation promises significant benefits by helping organizations realize cost savings in hardware, software, and administration; achieve greater agility; and reduce energy consumption.

However, implementing server virtualization can appear daunting. Migrating from a traditional server environment to a virtualized environment is complex, and evaluating ROI can be challenging. Virtualization technology is still relatively new, and few organizations have implemented large-scale production environments that can serve as examples.

In 2005, Intel IT began an extensive multi-year program to analyze the benefits of virtualization within Intel's business-computing environment, develop a virtualization architecture, and implement a production environment.

Virtualizing our business-computing environment presented significant challenges. Intel's massive worldwide environment includes approximately 7,000 to 8,000 physical business-computing servers running Microsoft Windows* and a wide variety of applications. Virtualizing and consolidating these physical servers could potentially deliver very large cost savings and other benefits; however, virtualizing an environment of this size is a large and complex undertaking.

We began our program with a detailed ROI analysis, followed by an assessment of consolidation opportunities and server platforms. We analyzed our current infrastructure to understand how we would make the transition to virtualization, and we mapped out our new virtualization architecture to take advantage of the opportunities we identified. Finally, we scoped out the engineering activities, formed an engineering team, and executed our plan, ultimately creating a virtualized production environment.

Solution

We began by conducting a detailed ROI analysis, examining each aspect of our environment that would be affected by virtualization. For each of these aspects, we analyzed costs in our current environment and compared this with the corresponding costs in a virtualized environment, using the assumption that we consolidated multiple physical servers into VMs on virtualization host servers.

Examining Return on Investment

Our analysis was based on consolidating about 4,800 older business-computing servers. When calculating ROI, our finance analysts used an extremely conservative consolidation ratio of 4:1, although we expected to achieve much higher consolidation ratios in practice. Our calculations included appropriate depreciation and amortization for each relevant category.

Our analysis covered many areas, including:

- **Server capital costs.** We assessed the cost of the average physical server used in our non-virtualized environment. We compared this with the cost of two-socket and four–socket server platforms that we could potentially use as virtualization hosts.

- Data center utility costs. We focused on the cost of electricity for IT equipment power and cooling. We regarded our existing utility infrastructure as a "sunk" cost; therefore, we did not include gains due to being able to use this existing infrastructure to support additional computing capacity in a virtualized environment.
- VM hypervisor license and support. We based our analysis on the cost of a four-year license for the VM hypervisor, including support. We also included the cost of centralized VM and host server-management software provided by the VM hypervisor supplier.
- LAN. We conducted a server infrastructure resource utilization assessment, which indicated that our typical physical servers use three network ports. We calculated the cost of switch capacity, based on the number of switch ports used by each server and the total cost of the switch. We also included the cost of Ethernet cables. Our analysis showed that we would use significantly fewer network switch ports in a virtualized environment, thereby reducing cost.
- Storage area network. Our analysis showed that many of our physical servers have two Fibre Channel (FC) connections to a storage area network (SAN). For comparison, we found that each host in a virtualized solution would also use only two FC ports, even though it would support multiple virtualized server workloads. We included the cost of FC cables in our calculations.
- Engineering and support headcount. We did not assume any headcount reductions. We assumed that virtualization would deliver savings due to automation efficiency, but that those savings would be offset by the additional engineering complexity and support needs in the virtualized environment. This complexity is due to configuration and management of the virtualization infrastructure. It includes overall resource management as well as network and storage configuration. We assumed that this complexity would result in a need for additional training, knowledge, and support.
- Automation and productivity gains. Our ROI calculations did not include productivity gains due to virtualization. However, we did expect a range of productivity benefits, such as the ability to provision systems in minutes rather than days or months, perform hardware maintenance and patch upgrades without downtime, automatically load balance workloads, recover more quickly from hardware failures, and use rapid rollback when issues occur. We expected that each support person would be able to manage a larger number of physical servers due to virtualization and automation.

In addition, we expected that virtualization would result in more efficient resource utilization, such as higher memory and CPU utilization, due to consolidation. It can be difficult, if not impossible, to consolidate dissimilar applications onto a single physical server without using virtualization, whereas in a virtualized environment, it should be simple because the environment isolates VMs from one another.

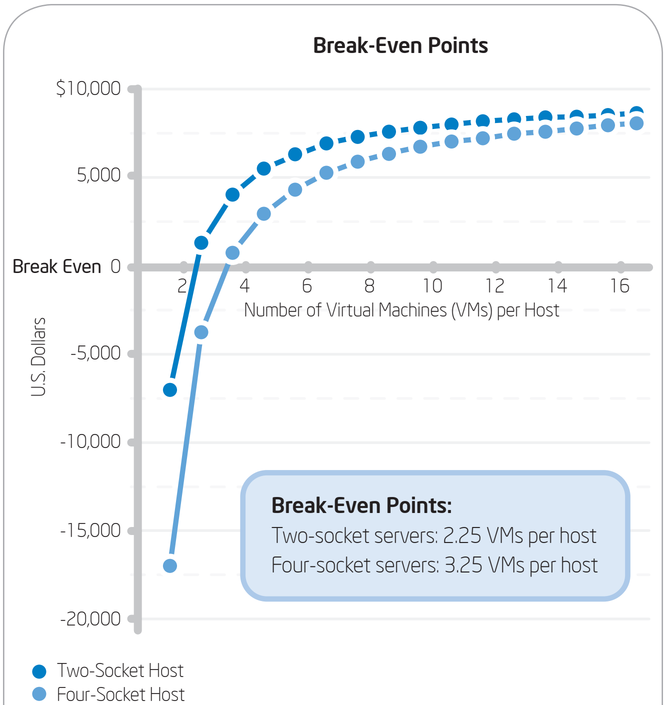
ROI Analysis Results

Our analysis showed that virtualizing our business-computing environment would generate substantial estimated savings of USD 17.6 million to 27.7 million over five years. Reduced server capital costs were the largest contributor, followed by network switch and storage savings.

Break-Even Point

To help analyze virtualization ROI, we examined and updated a previous ROI study that detailed ROI at different consolidation ratios, which had been performed by Intel's manufacturing-computing group. We updated the study to include two-socket and four-socket servers based on the latest Quad-Core Intel Xeon processors.

The analysis indicated that a migration from our current physical server environment to a virtualized environment would result in a positive ROI even at very low consolidation levels. We found that the



Assumptions include:

- Based on comparison of two-socket and four-socket servers with Quad-Core Intel® Xeon® processors
- Support headcount at .0007647 support staff per server
- Annual cost per head USD 100,000
- Electricity cost USD 0.07 per kilowatt hour
- Rack and storage costs USD 100 per server
- ROI calculations include server depreciation and cost of virtual machine and host software, licenses, and support

Figure 1. Virtualization return on investment (ROI) break-even point.

break-even point using two-socket servers as virtualization hosts would be at approximately 2.25 VMs per host, as shown in Figure 1. With four-socket hosts, we found that the break-even point occurred at 3.25 VMs per host.

Host Platform Selection and Consolidation Ratios

We needed to make two important, related decisions—selecting the right host platform and determining desired consolidation ratios—to help ensure our virtualization program delivered the most benefits. To make these decisions, we analyzed resource utilization in our current non-virtualized environment, considered different consolidation scenarios, and examined our platform options.

Resource Utilization in the Current Environment

In late 2005 and early 2006, we collected extensive server resource utilization data from our current non-virtualized environment. This enabled us to estimate the VM host resources we would need when we migrated each physical server into a virtualized environment.

To do this, we used performance-monitoring tools to collect real-time utilization data for all IT enterprise business-computing servers at five major data centers. We monitored CPU, memory, network I/O, and disk I/O, and collected utilization snapshots for each server every ten minutes over 60 days. We then compiled and analyzed the data to determine the maximum, minimum, and average utilization for each resource.[1]

This analysis demonstrated that our current environment was woefully underutilized. Average CPU utilization was 12 percent, and 75 percent of surveyed systems consumed less than 1 GB of memory—even though most had 2 GB or more of memory installed.

Consolidation Ratios

Based on our performance data analysis, we saw significant opportunities for virtualization and consolidation. The vast majority of our existing servers were underutilized and many were four or more years old. We could easily consolidate these older servers into VMs on newer, more powerful servers. When consolidating older servers, migration to VMs can actually result in performance improvements.

[1] "Memory Sizing for Server Virtualization" Intel IT, July 2007.

Using our performance data, we created real-world consolidation scenarios in which we combined workloads with dissimilar resource requirements onto a single host in order to enable the most efficient use of host resources. Using this approach, we determined that our average maximum possible consolidation ratio would be about 16:1 on two-socket servers based on Quad-Core Intel Xeon processors and equipped with 16 GB of memory.

Virtualization Host Platform

When selecting a host server platform, we first looked at the possibility of reusing existing servers and increasing their resource utilization through virtualization.

However, we quickly concluded that reuse was a bad idea. New machines offered much greater performance than our existing systems. This would enable us to reach higher consolidation ratios, reducing overall cost. Part of this cost difference was due to virtualization software licenses; we would have needed a larger number of these licenses on older, less-powerful systems.

New systems were relatively inexpensive, considering the substantial performance improvements that they could deliver. Selecting new systems also enabled us to standardize on a single virtualization-hosting platform to simplify engineering, deployment, and support. Live migration also requires that hosts meet CPU compatibility requirements, particularly for the support of 64-bit VMs.

We then needed to choose possible server platforms, which had to be capable of supporting different consolidation ratios. Alternatives included two-socket, four-socket, and eight-socket servers as well as blades.

To select the right platform, we had to ask ourselves what was important to us. A key consideration, besides cost and performance, was risk. The more VMs a host supports, the greater the risk if the host experiences failure. Virtualization technology is still relatively new, and we decided that until the technology has had more time to mature, we wanted to limit consolidation ratios to a maximum of about 20:1 in order to reduce risk.

Our alternative platforms included two-socket and four-socket servers. Both currently accommodate quad-core processors, and we determined that four-socket servers offered roughly twice the performance of two-socket servers at roughly twice the cost. Intel IT testing has found that four-socket servers also offer other advantages such as more predictable scalability and are better suited for memory-intensive applications.[2]

However, two-socket servers were less expensive than the larger servers, and we had already determined that average maximum consolidation ratios with these servers would be about 16:1—within the 20:1 limit determined by acceptable risk.

Two-socket servers were also preferable to even larger eight-socket servers and blades for similar reasons. Blades could have supported very large numbers of VMs within a single chassis, but this would have increased risk and potentially introduced other issues such as the need to upgrade network bandwidth to 10 GB.

As a result, we selected two-socket servers based on quad-core processors for our initial virtualization initiatives. Our first virtualization initiatives focused on the easiest consolidation opportunities, virtualizing applications with relatively low resource requirements. We see opportunities to achieve 16:1 consolidation ratios for these applications.

In the future, we could use higher-end four-socket servers to achieve higher consolidation levels once we are comfortable that the technology is mature and the risk is acceptable, or for applications that have high resource needs.

We decided to dovetail virtualization implementation into our normal server refresh cycle. This minimizes

[2] "Comparing Two- and Four-Socket Platforms for Server Virtualization" Intel IT, March 2008.

disruption and provides better capacity planning control, helping us size the virtualized environment to handle expected workloads.

Virtualization Architecture

Designing our virtualization architecture was a complex undertaking. We first needed to understand our current pre-virtualization environment in order to be able to map it to a virtualized architecture. We began by assessing the infrastructure technologies currently in use within Intel IT, including storage and SAN, network, BAR, monitoring and alerting, capacity management, remote management, automation, and operating systems.

Once we completed our assessment, we considered what our ideal virtualization infrastructure would look like. We thought about it like this: If we were building a completely new environment based on virtualization, what would we use?

Using this approach, we designed an architecture that merged our current and ideal virtualization architectures into a realistic solution that took into account cost and implementation effort. This meant that in some areas, such as BAR, we had to make compromises because it was not cost-effective to replace our existing infrastructure.

Virtualization Host

We based our virtualization host architecture design on our selected two-socket virtualization host platform. The design, including storage and networking, is shown in Figure 2.

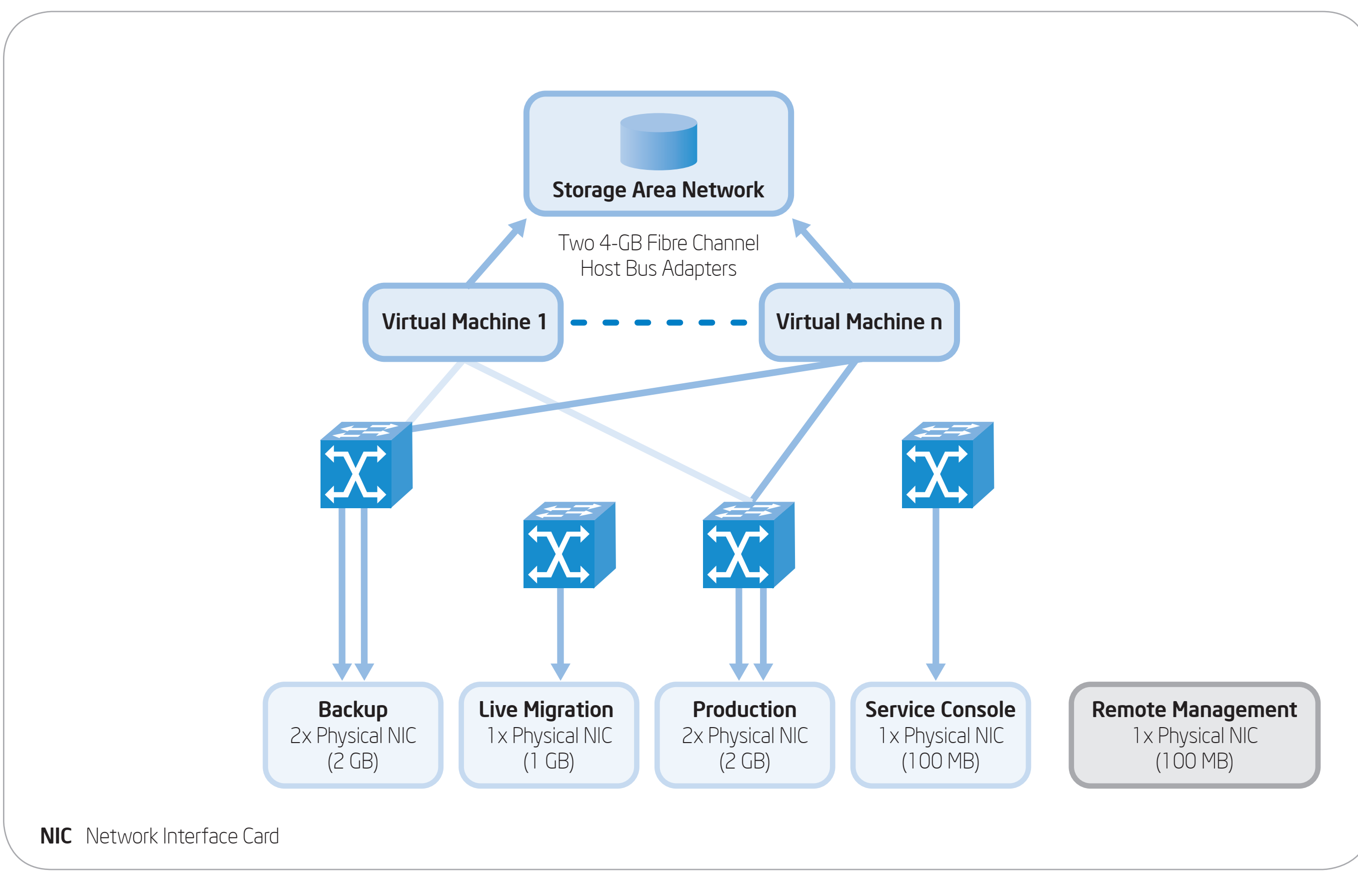


Figure 2. Virtualization host architecture, showing network and storage.

Network

Our network architecture consists of seven network ports for each host:

- One 1-GB port dedicated to the VM live migration software. We considered sharing ports for BAR and live migration, but we were concerned that when backups were intensively using bandwidth, live migrations could fail because they were not able to duplicate CPU and memory states on the source and destination hosts due to a lack of network bandwidth. This could become a significant issue because we use live migration to achieve dynamic resource balancing, which automatically balances the VM load among hosts.
- One 100-MB port for the service console. Performance testing and subsequent production use have shown that 100 MB is adequate for console network traffic in normal use, and it costs less than higher-speed ports. However, we found that a 1-GB port is advisable when conducting physical-to-virtual (P2V) conversions, because migrations use the host's service console to migrate data into the virtual environment.
- One 100-MB port for remote management. This was adequate in our physical server environment, and we have found that it suffices in the virtualized environment as well.
- Two 1-GB ports bonded together for production network traffic. We needed high bandwidth because virtualization consolidates multiple server workloads onto a single host. Performance testing showed that 2 GB is adequate to support 16:1 consolidation ratios. Our hosts can accommodate a total of 10 network interface cards (NICs), which allows us to increase this bandwidth if necessary by bonding an additional 1-GB port.
- Two 1-GB ports bonded together for backup traffic. Intel's BAR implementation requires all the combined SAN data from the VMs on the virtualization host to flow through the host's BAR network. This requires 2 GB of bandwidth in order to enable us to meet our BAR service level agreements (SLAs), which require full backups on weekends and differential backups during weekdays.

Storage Architecture

VM live migration calls for a SAN approach, particularly when using multiple host systems. The data used by each VM's applications must reside on the SAN, accessible by all hosts, in order to allow live migration to occur. We store most data and software on the SAN, with the exception of virtualization guest VM OSs, which reside on the server's local storage.

We request SAN volumes of approximately 356 GB, configured as RAID 5, to maximize platter use. We optimize performance by using as many drives as possible within each RAID group.

Analysis of performance data has shown that disk I/O has never been an issue in any of our environments, even when running numerous VMs on a single host, each with disk-intensive applications.

Back Up and Restore

Backups are performed from within the individual VMs, with the frequency and strategy determined by the owner of the VM and by our operations BAR team.

Our approach to BAR illustrates the tradeoffs we had to make due to our existing environment. It was not cost-effective to replace our existing BAR physical infrastructure. Therefore, we assessed the performance of this existing BAR infrastructure and determined the maximum amount of data we could backup and restore, given that the combined data from the VMs on each host has to pass from the SAN through the host's network before being offloaded to tape. We found that the maximum was 4 TB per host before we encountered issues with BAR SLAs.

Virtual Machine and Host Management

Our centralized VM and host server-management software, provided by our VM hypervisor vendor, performs a wide range of roles including VM and host configuration, provisioning, monitoring, migration, and resource management in our virtualized environment. Therefore, the design and implementation of our VM and host-management architecture could significantly affect the efficiency of our entire virtualized environment.

Our design is shown in Figure 3. We took into account cost, availability, scalability, and location. We determined that we did not need high availability because we could afford up to 24 hours of downtime with minimal impact. This meant that we could reduce cost by eliminating the need for clustering.

We run our centralized VM and host management software on a physical non-virtualized server, and we run the virtualization host software license server separately as a VM within our virtualized environment. This means that our license server keeps working even if the management software is unavailable, allowing us to continue deploying VMs and hosts.

We also offloaded the database used by the management software to another physical server. Performance tests showed that this helped both the database and the management software servers run more efficiently, so that the system performed better during periods of heavy concurrent administrative use such as when conducting multiple simultaneous deployments.

Other important decisions were how many management servers we should have and where they should reside. Each server was capable of managing 200 hosts and a total of 2,000 VMs. We originally expected to deploy only one server within each major region, but we subsequently decided to deploy one server per major data center because of the anticipated growth of these data centers and of virtualization overall. This decision also meant that we did not have to be concerned about the potential impact of WAN bandwidth between data centers within each region.

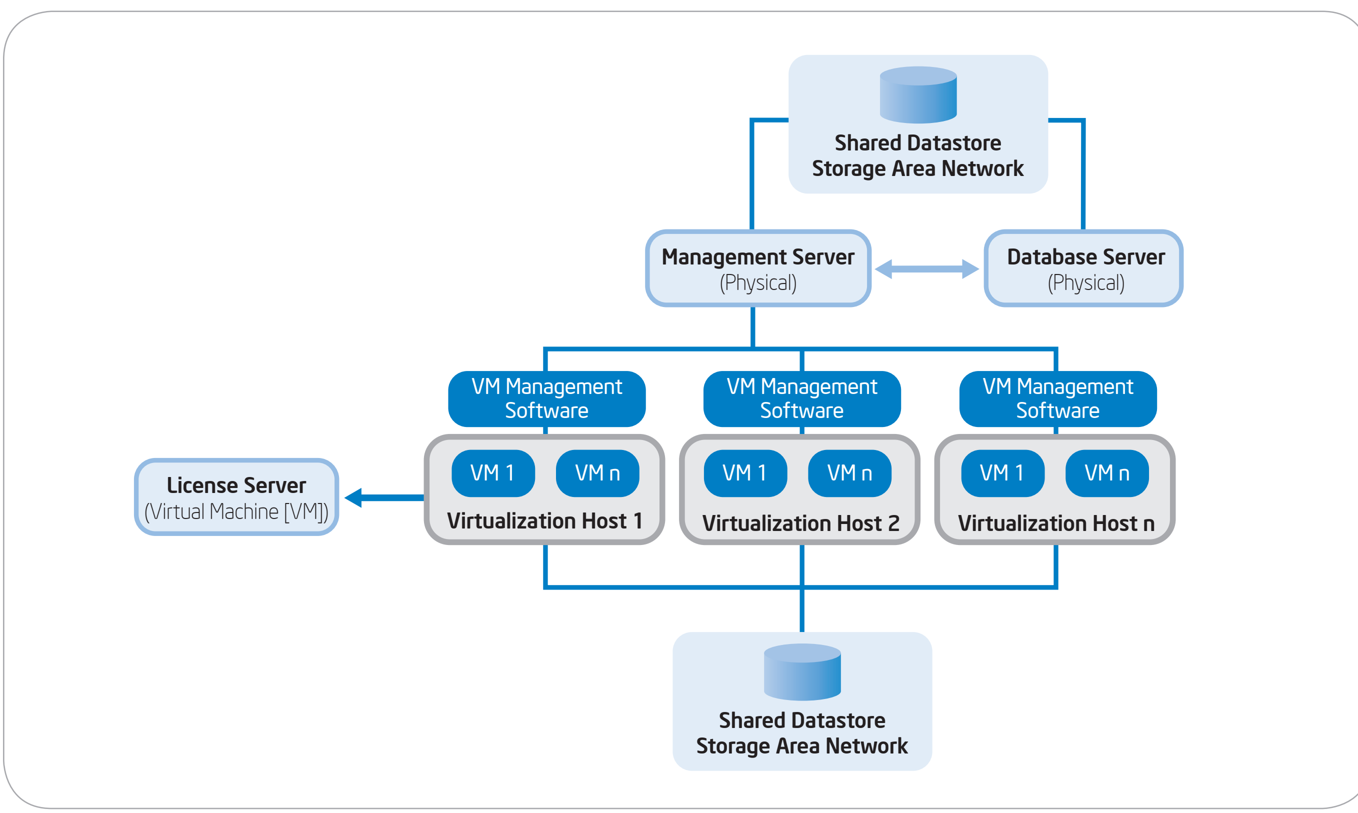


Figure 3. Virtual machine and host management architecture.

Resource Monitoring and Management

We use a range of other tools to monitor and manage resources within our environment. We monitor host resource utilization to determine whether it remains within acceptable limits:

- A daily average of less than 70 percent host CPU utilization
- Less than 5 percent swap utilization
- Less than 65 percent production network virtual switch utilization

While we consider a brief peak to be acceptable, sustained excessive resource use is not. In practice, dynamic resource balancing limits the extent to which this occurs, though it considers only CPU and memory utilization, not disk and network I/O, when deciding how to balance VMs among hosts.

We use internally developed capacity, performance, and management agents to track CPU, memory, disk I/O, and network I/O performance for each VM.

For asset management, we created scripts that run on the management server every five minutes and update our internal asset management database with information about VMs that are newly provisioned, have migrated to new hosts, or have been removed.

Physical to Virtual Conversion

We defined a clear process and architecture for the P2V conversion that migrates each physical server into the virtualized environment, as shown in Figure 4.

We decided that we would use a manual conversion process to accommodate the large number of different physical server configurations in our data centers. We refined this process as we completed more conversions, and we tested and included batch conversion capabilities. These batch conversions allowed us to convert a larger number of systems, but they also required additional planning and customer communication.

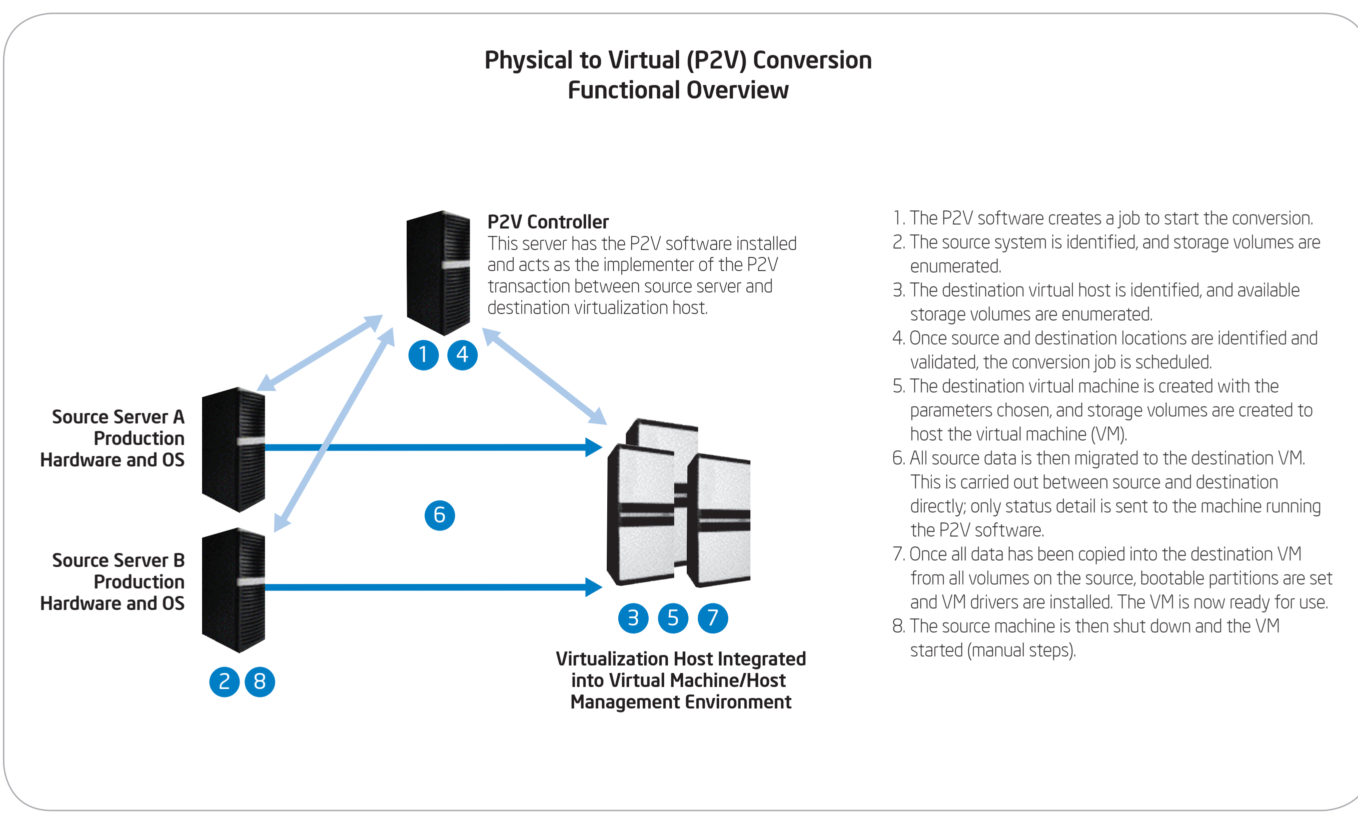


Figure 4. Physical to virtual conversion.

We found that once we had successfully completed a conversion, we needed to allocate resources to validate the results of the conversion activity as well as application functionality and performance.

We also found that we could successfully conduct multiple simultaneous conversions into virtual environments when multiple hosts were present. However, only one conversion should be attempted at any one time per host. We found that attempting multiple simultaneous conversions onto the same host significantly lengthened conversion time and could result in a stalled or failed conversion due to saturation of the service console port with network traffic.

Building the Environment

Once we had designed the architecture of our virtualized environment, our next steps were to scope and execute the engineering and business process activities needed to implement our environment.

Engineering

Our engineering activities included building and testing each part of our virtualized environment. This included running proof of concept (PoC) tests to validate aspects of our approach. These PoC efforts involved testing possible virtual host architecture against a mixture of workloads to determine individual VM responsiveness. The results of these PoCs helped establish maximum consolidation ratios and tested specific use cases for Intel business groups.

Virtualization Host and Management Server

We engineered, tested, and delivered build documentation for our selected two-socket server platform based on Quad-Core Intel Xeon processors. We did the same for the servers running our VM and host-management software and associated database. Through performance testing, we determined that these management servers required 4 GB of memory, dual 1-GB Ethernet ports, and two 72-GB disk drives; the database servers required similar disk capacity but needed 12 GB of memory.

Dynamic Resource Balancing

We engineered and documented the dynamic resource balancing capability, and tested basic functionality to help ensure this feature worked as desired.

- We established default VM settings for our initial deployment. We set a default of normal CPU priority, with unlimited ability to expand resources if necessary. This helps ensure that no VM is starved of resources and provides the ability to take advantage of the resources that are available.
- We restricted use of settings that allow VMs to reserve production environment resources. We allowed use of these settings only on an exception basis. This is because setting artificially high resource limits greatly reduces the number of VMs in a resource pool. When we do use reservations, we leave headroom because reservations that fully commit the capacity of a cluster or host can prevent live migrations.
- We generally avoided specifying rules that define which VMs should or should not share a host. This allows the dynamic resource balancing software maximum flexibility in determining where to place VMs. Rules may be necessary in some cases, however. For example, to help ensure availability of directory services, it may be advisable to specify that two VMs running directory services software do not share the same host.
- For our pilot implementations, we found it best to use a partially automated mode in which the dynamic resource balancing software initially places the VM on a host that has the required resources, but does not subsequently automatically move that VM between machines. Later, once we have gained experience and a better understand the VM's characteristics, we use a fully automated approach in which the resource balancing software migrates VMs automatically.

- We closely monitored clusters during early implementation stages to determine the optimal settings. We found that clusters should include hosts with similar CPU, memory, and storage to allow the dynamic resource balancing software more flexibility in placing VMs on different hosts.

Time Synchronization

Because VMs share time on a physical host, a VM cannot exactly duplicate the timing of a physical machine. The VM's clock could be out of synch with the host clock, causing problems for the application running on the VM.

We solved this problem by running Network Time Protocol (NTP) on the virtualization software host console to help ensure that the host is synchronized with the network. For each VM, we disabled the OS time server synchronization service and used the VM supplier's software to synchronize the VM with the host. With this setup, our hosts receive the correct time from the network and VMs receive the correct time from the host.

Provisioning

We determined which guest OSs the VMs would support, and we created automated provisioning solutions. We automated installations of 32-bit and 64-bit versions of Microsoft Windows for VMs and the hypervisor supplier's Linux*-based host OS. We accomplished this by creating images and scripts and by using a third-party deployment tool.

Inventory

Intel has security and business-related requirements for effective asset management. However, asset management can be especially challenging in virtualized environments. Dynamic resource balancing automatically moves VMs between hosts, making it difficult to track the location of each VM without manually extracting it from the VM management software.

To overcome this problem, we engineered an SQL script to automatically track creation, deletion, and physical movement of VMs. This accesses the management software database to extract the names of VMs and the physical machines hosting them. The asset database initiates this update every 15 minutes, and the update generally completes within 5 seconds.

Patching

We engineered, tested, and documented patching processes. We use fully automated patching for VMs running on Microsoft Windows and for our management server and associated database. Currently, we perform manual patching of hosts.

Enterprise Management and Monitoring

Table 1. Enterprise Management and Monitoring Solutions

Manageability Segment	Solution
Discover	Permit virtual machine (VM)/host management software to discover all virtualization hosts and VMs. Include monitoring and management software agents in all provisioned VM and host OSs.
Provision	Select a single software deployment tool to bare-metal provision host OSs and VMs.
Monitor	All application products must incorporate their fault event messages with the monitoring utility used throughout Intel IT. Use VM/host management software to track heartbeat and performance of the host OS.
Respond	Use hypervisor vendor's dynamic resource balancing software for VM load balancing. Use hypervisor vendor's high availability feature for automated host failover.
Integrate	Engineer scripting solution to track VMs and hosts in asset management database.

Table 2. Support Model for Our Virtualized Environment

Role	Responsibilities/Guidelines
First-level incident management	Includes support for applications, hardware, information security, and virtual machine (VM) patching; routing to infrastructure support groups
Second-level incident management	Hardware support; incident management; server provisioning, removal, repair, and basic troubleshooting; collecting metrics for statistical analysis
Third-level incident management; ownership of the production environment	Incident escalation; trend analysis; problem management; change management; capacity planning based on customer forecasts
Customer engagement for application landing; third-level incident management for application-related issues	Manage customer relationships for application landing and sustaining support; handle application-related escalations
Fourth-level escalations	Provide technical guidance on the platform strategy; design and define the platform; engage vendor

We defined solutions for capacity and performance management, and for monitoring, as shown in Table 1. This included establishing which management tools we would use for each function.

Other Areas

We also engineered, tested, and documented other aspects of our environment as defined in our virtualization architecture, including:

- High availability; basic testing confirmed that failover of VMs occurred as expected
- Backup and restore
- Network configuration
- Migration
- Storage; we used performance testing to verify that each host had adequate FC bandwidth to support the required number of VMs

Business Process

Implementing a virtualized environment involves significant changes to business processes as well as technology. We identified and put in place the processes necessary to implement and support our environment.

Change Frequency

We determined the maximum acceptable frequency for changes to our architected solution. New virtualization technologies are emerging rapidly—but too much change, too often, can make it difficult and cumbersome to maintain a stable environment and hinders our ability to roll out new VMs. Our goal is to allow one major change annually, plus minor revisions during the year.

Requirements

We gathered business group requirements that defined each group's expected usage of the virtualized environment. We consolidated these requirements inputs into a single product requirements document that we used to prioritize and disposition our engineering activities. With such a complex transition to a new environment, we found that it was important to restrict our initial implementation to core foundation elements and not to allow "scope creep."

Support

We defined support models for engineering, escalating responses to problems, and sustaining the environment. This included defining virtualization management roles and an associated model for delegating administration. Our operations group is responsible for the overall approach to defining the detailed assignments.

We plan to use custom-defined roles within our VM management software to map operational duties to access privileges. These custom roles will be defined and managed as a directory group. Our operations group will own the administrator role, and each management server will have the same custom-defined roles.

Pilots

We implemented a pilot project to validate each of our engineering efforts. Typically four weeks long, these pilots demonstrated that the solution would work in a production environment. The goal was to identify and fix minor problems before large-scale deployment. We found that it was important to specify success criteria and metrics that showed whether the expected performance was achieved in practice.

Training

We identified the personnel needed to support the pilots and subsequent deployments. We also identified their training requirements and made sure that they were trained before implementation.

Decision Makers

We determined the review groups, quality assurance groups, stakeholders, and decision makers who would affect the progress and direction of the virtualization effort. Identifying the technical and business decision makers from the outset helped ensure fluid project management. When we encountered challenges or needed decisions in order to proceed, we already knew the right people to talk to.

Budget

Analyzing the engineering and business process requirements enabled us to determine and discuss the budget required and gain approval from the outset.

Security

We conducted a formal risk assessment and, where necessary, defined processes or engineering solutions to mitigate each risk. To perform the assessment, we:

1. Identified each risk. For each risk, we rated the likelihood of occurrence on a scale of 1 to 5, and then the potential impact in terms of cost or downtime on a scale of 1 to 5. We multiplied the likelihood and impact ratings to obtain a total risk score.
2. We identified any controls currently in place and rated their effectiveness.
3. Based on this, we determined which risks required mitigation, identified engineering or process solutions, scored the effectiveness of these mitigation strategies, and calculated a final residual mitigated risk.

Mitigation strategies included:

- Mapping current roles and permissions to virtualization functions.
- Integrating our corporate directory for host root account management.
- Setting procedures for containment in the event that a virus infects a host or VM.
- Using automated patching for VM management software, VMs, and host wherever possible.
- Limiting service connectivity using default deny.

We log and retain all system messages to meet Intel IT requirements. Each virtualization host retains its VM software console messages. In addition, a central log server stores all messages generated on the virtualization hosts.

We installed software to restrict the systems that can connect to services on the VM software console. Our architecture also calls for a bastion host to restrict access to the VM software from other parts of Intel's business-computing environment.

For guest VMs, we provide the same controls that are enforced on OSs and applications running on physical systems in our current environment, with no changes to the compliance checks and vulnerability scan procedures that are currently performed.

We also implemented controls to enforce network security. For example, virtualization host software cannot span multiple secure zones, and virtual network devices are not allowed to participate in any Layer 2 or Layer 3 routing, tunneling, or network infrastructure protocols. Encrypted communications are required for remote administration of all physical and virtual devices on virtualization hosts.

Consolidation and Deployment

Before virtualizing each existing application, we collected performance data and ascertained the application's resource needs. We also did this for applications that were still in development. We defined a process for conducting these assessments to help ensure our data was consistent.

Virtual Machine Sprawl

Many organizations fear that virtualization will result in an unexpected proliferation of VMs, often known as VM sprawl. To avoid this, we defined a process for deploying VMs. A hosting services team within our operations group is responsible for handling all service requests. This group is responsible for landing applications and determining whether a new VM or new hardware capacity is needed. This centralized model also helps ensure that we make optimum use of resources. Another key to controlling VM sprawl is that we charge each business group for each of their VM deployments on a monthly basis.

Key Lessons

We learned many valuable lessons while architecting and deploying our virtualized environment.

Virtual Machine Management Servers

Because we rely heavily on our management servers for routine duties, dynamic resource balancing, asset management, and performance trending, we initially considered a high-availability clustered server design. However, we found the complexity of this solution caused problems. We also realized that downtime of up to 24 hours had minimal impact on our environment. We therefore decided not to use clustering. This decision also reduced cost.

We originally planned to have our VM software license server run on the management server. However, this meant that when the management software was down, we could not add VMs or hosts to our environment, which delayed deployments. Also, VM licenses expired if the management server was down for too long. Because of these potential issues, we decided to run our management software on a physical non-virtualized server and to run the license server as a VM within our virtualized environment.

Our original design called for the database used by the management software to run on the same server as the management software. However, we found that with this arrangement, the system slowed to a crawl when we were deploying multiple VMs concurrently. Through extensive performance analysis, we found that this was because the database and management software were contending for the same resources. We solved this problem by offloading the database to its own physical server.

Physical to Virtual Conversions

We discovered several factors that can make it difficult to migrate legacy systems into a virtualized environment.

These included customized OSs on legacy physical servers. We found that significant operating system customization can affect the behavior of P2V tools and may even prevent a successful migration. We realized that whenever we release an OS for internal use, we need to test it in a virtual environment as well as on a physical server. We do this by using a P2V tool to migrate it from a test physical server into a VM. We found that this helps ensure that migrations occur with few problems and that the resulting virtual machines are stable, supportable, and perform as expected.

We also took steps to help ensure the network infrastructure of physical systems is ready for P2V conversion. We make sure that all NICs in the systems to be converted are set to full duplex, regardless of their supported network speed. We found that P2V conversion takes up to four times longer when attempted on a legacy system without optimal network settings.

Application Response Times

In a virtualized environment shared by many applications, some applications inevitably have to wait for the resources they need to execute. As a result, applications that require real-time response or are sensitive to delays may not perform well in a virtualized environment. We found that it is also a good idea to turn off graphical screen savers because they consume unnecessary resources.

Virtual Machine Live Migration

Our live migration software requires that all VMs in a cluster use the same subnet. It is possible to use multiple network switches, but one must make sure they enable rapid spanning tree. If they do not, synchronization of the memory and CPU states between source and destination hosts may never occur due to network latency issues. We also learned to avoid dedicating NICs to VMs, because live migrations would not work properly with this physical constraint. We found that live migration and dynamic resource balancing required network bandwidth of 1 GB. Live migration also requires that hosts have identical or compatible CPU architectures, particularly for support of 64-bit VMs.

Security and Administration

We found that it is advisable to use superuser, or sudo, privileges for VM host administration to prevent dispersion of the root password.

Intel's security policy calls for periodic password changes. To avoid frequent manual changes to each hypervisor password, we integrated virtualization account management with our corporate directory. This allowed us to create administrator group objects that can make a single password change that takes effect across multiple host systems.

We use file integrity checking on Linux-based VMs because we have found this to be more effective than antivirus software. We use a host intrusion prevention system (HIPS) for our VM host firewall. We also protect our VM hosts by limiting the systems they will respond to. For example, they respond to backup requests only from the backup server.

We use a server to log certain activities and retain the information as required for compliance with the Sarbanes-Oxley Act.

Storage Area Network

To enable live migration, all hosts in a cluster need to have visibility into all shared disk units (logical unit numbers, or LUNs). We used different disk access management systems on these LUNs, depending on whether they were used to store OSs and data with modest I/O requirements or for larger databases.

Host assignment to SAN frame ports is on a round-robin basis to avoid starving hosts of data access. With multiple VM's, each host tends to generate more storage activity than non-virtualized hosts.

Application Landing Criteria

We developed criteria for identifying which of our applications are suitable for virtualization and which are not. We also identified factors that help us determine which applications should share the same VM host and which should be kept separate and deployed onto different hosts. Each organization is different, so these criteria may vary between companies.

At Intel, we are initially focusing our business-computing virtualization efforts on the easiest opportunities: business applications that are not mission-critical. With these applications, the business impact of downtime is minimal. We have thousands of older servers running these applications, and these systems are ripe for consolidation.

By running performance analysis tools on physical production systems, we were able to determine the resource needs of these applications. Using this information, we could efficiently use host resources and maximize performance by deploying a mixture of VMs with differing resource needs onto a single host.

For example, a CPU-intensive application could share the same host as a memory-intensive application, a network-intensive application, and a disk I/O-intensive application. Because each consumes different resources, the impact on overall system performance is minimized.

Dynamic resource balancing is also a consideration. At Intel, we use fully automated dynamic resource balancing so that VMs are automatically allocated to hosts. Because all VMs in a cluster are under the control of this software, we do not need to manually assess memory and CPU availability when determining which host an application should land on. This approach also allows automatic rebalancing of resources in production, because dynamic resource balancing will automatically migrate VMs between hosts to balance utilization of host CPU and memory resources. One limitation of automated dynamic resource balancing is that it does not take into account network and disk I/O. It can also be set to manual mode, in which case the software will notify an administrator of recommended landing hosts for new VMs and also recommend resource balances, but manual intervention will be needed to allow these migrations to occur.

We also consider SLAs for each application type. Some applications may require higher service levels than others, and therefore may need to be deployed onto a separate cluster of hosts that is supported at the right level. For example, a finance application may have higher service level needs than a print server.

BAR requirements may differ. The finance system will have more stringent BAR needs than the print server. Depending upon the backup architecture, it may make sense to have the two on the same host because they can share backup network bandwidth. The finance system is backed up more frequently than the print server, and therefore the two applications can efficiently share the same infrastructure.

Deployment

We began successfully deploying our business-computing virtualized environment in 2005 and are expanding it over time. As of early 2008, our environment consisted of a total of about 190 VMs: 75 VMs used for production applications and approximately 115 VMs used for pre-production purposes such as validating applications before moving them into the production environment. We have installed VM host capacity that will enable us to expand our environment to at least 600 VMs at three major sites in the first quarter of 2008. We plan to increase this to approximately 1,200 VMs and at least one additional site by the end of 2008.

To date, virtualization has eliminated the need for about 190 physical servers. Our initial deployments have used conservative consolidation ratios of about 10:1 on two-socket hosts with Quad-Core Intel Xeon processors and 16 GB of memory, and we are analyzing memory usage to confirm that we can increase this ratio to our goal of approximately 16:1.

We have confirmed that several anticipated virtualization benefits are achievable in practice. We have used automation to rapidly provision hosts and VMs, including OS, in 10 minutes. This is much faster than provisioning traditional physical servers, which took several hours even if we already had the servers in-house and could take months if we had to order new systems. We have been able to migrate all live VMs off a virtualized host in order to carry out routine host maintenance without experiencing application downtime. Dynamic resource balancing is also working as we expected: new VMs successfully deploy to the least-utilized VM host.

We have found that VMs are indeed isolated as if they were located on separate physical servers, and that they do not significantly interfere with the resource requirements of other VMs.

We have found that we can restore VMs significantly more quickly and easily than physical servers, because we can easily re-create the VMs, because the data is readily available on the SAN, and because we have multiple VM hosts available at any time.

We are adding higher availability to our environment with a feature that allows automated migration of VMs in the event of a host hardware failure.

We recently added a novel virtualization solution to support our labs and developers. Users who want

Conclusion

Through extensive analysis, planning, and testing, we have shown that we can design and engineer a virtualized business-computing architecture and deploy it in a production environment. Virtualization is already delivering the benefits that we anticipated, including faster and more automated deployment and faster recovery times. We expect to achieve increasing gains as our environment grows and matures. By sharing our experiences, we hope to help other IT organizations realize similar benefits.

Authors

Bill Sunderland is a virtualization program manager with Intel IT.

Stephen Anderson is a systems engineer with Intel IT.

Acronyms

BAR	backup and restore
FC	Fibre Channel
HIPS	host intrusion prevention system
LUN	logical unit number
NIC	network interface card
NTP	Network Time Protocol
P2V	physical to virtual
ROI	return on investment
SAN	storage area network
VM	virtual machine

www.intel.com/IT

This paper is for informational purposes only. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

Intel, the Intel logo, Intel. Leap ahead. and Intel. Leap ahead. logo, and Xeon are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2008, Intel Corporation. All rights reserved.

Printed in USA
0408/IPKA/KC/PDF

♻ Please Recycle
319732-001US