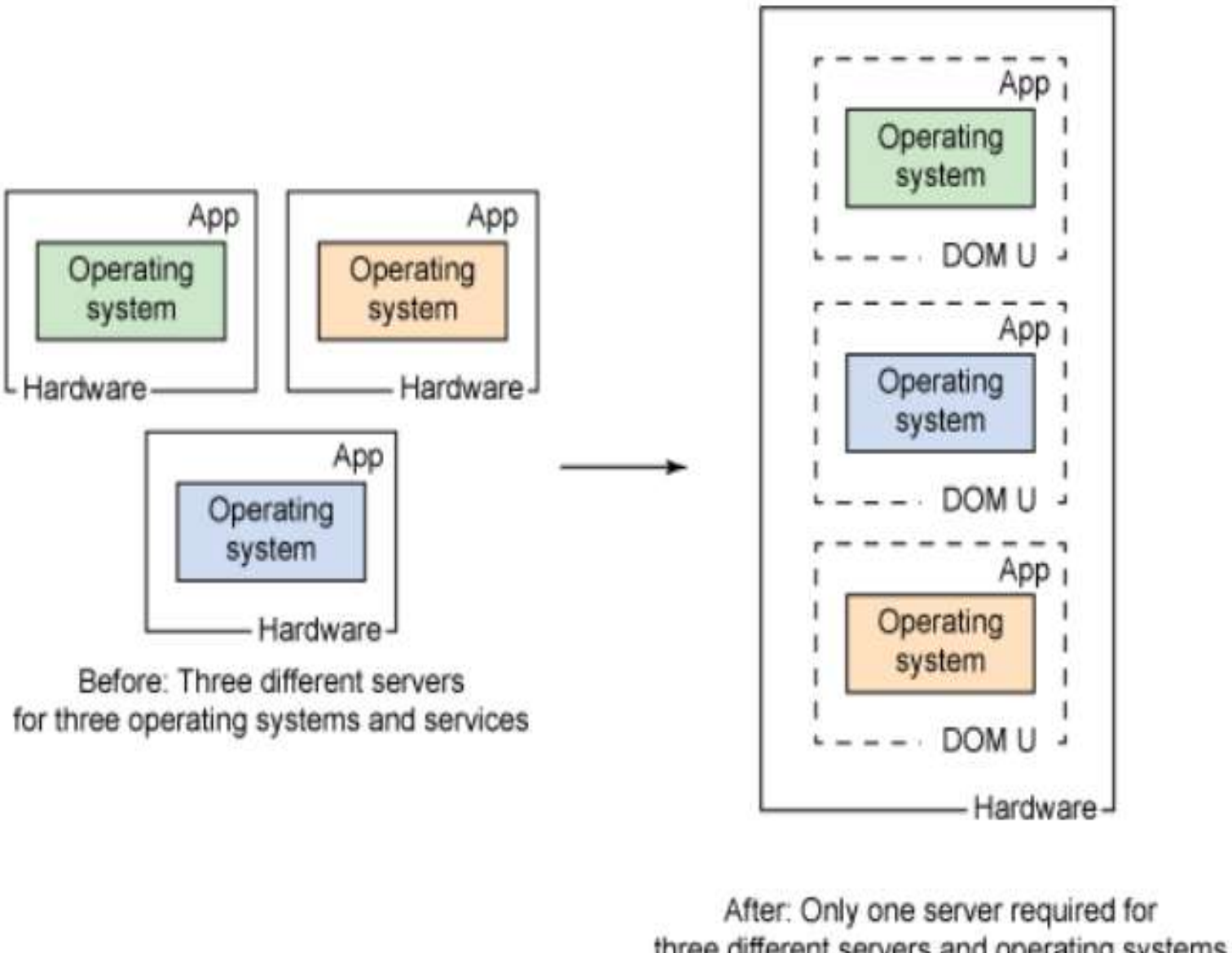
UNIT – III Cloud computing technologies and Virtualization

Virtualization

- When you use cloud computing, you are accessing pooled resources using a technique called virtualization.
- Virtualization assigns a logical name for a physical resource and then provides a pointer to that physical resource when a request is made.
- Virtualization provides a means to manage resources efficiently because the mapping of virtual resources to physical .


App
Operating system
Hardware
App
Operating system
Hardware
App
Operating system
Hardware
Before: Three different servers for three operating systems and services
App
Operating system
DOM U
App
Operating system
DOM U
App
Operating system
DOM U
Hardware
After: Only one server required for three different servers and operating systems

Types of virtualization

- These are among the different types of virtualization that are characteristic of cloud computing:
- Access: A client can request access to a cloud service from any location.
- Application: A cloud has multiple application instances and directs requests to an instance based on conditions.
- CPU: Computers can be partitioned into a set of virtual machines with each machine being assigned a workload. Alternatively, systems can be virtualized through load-balancing technologies.
- Storage: Data is stored across storage devices and often replicated for redundancy.

- To enable these characteristics, resources must be highly configurable and flexible. You can define the features in software and hardware that enable this flexibility as conforming to one or more of the following mobility patterns:
- P2V: Physical to Virtual
- V2V: Virtual to Virtual
- V2P: Virtual to Physical
- P2P: Physical to Physical
- D2C: Datacenter to Cloud

- C2C: Cloud to Cloud
- C2D: Cloud to Datacenter
- D2D: Datacenter to Datacenter

Load Balancing and Virtualization

- The technology used to distribute service requests to resources is referred to as load balancing.
- Load balancing can be implemented in hardware, as is the case with F5's BigIP servers, or in software, such as the Apache mod_proxy_balancer extension, the Pound load balancer and reverse proxy software, and the Squid proxy and cache daemon.
- Load balancing is an optimization technique; it can be used to increase utilization and throughput, lower latency, reduce response time, and avoid system overload.

- The following network resources can be load balanced:
  - Network interfaces and services such as DNS, FTP, and HTTP
  - Connections through intelligent switches
  - Processing through computer system assignment
  - Storage resources
  - Access to application instances
- Without load balancing, cloud computing would very difficult to manage.
- Load balancing provides the necessary redundancy to make an intrinsically unreliable system reliable through managed redirection

Understanding Hypervisors

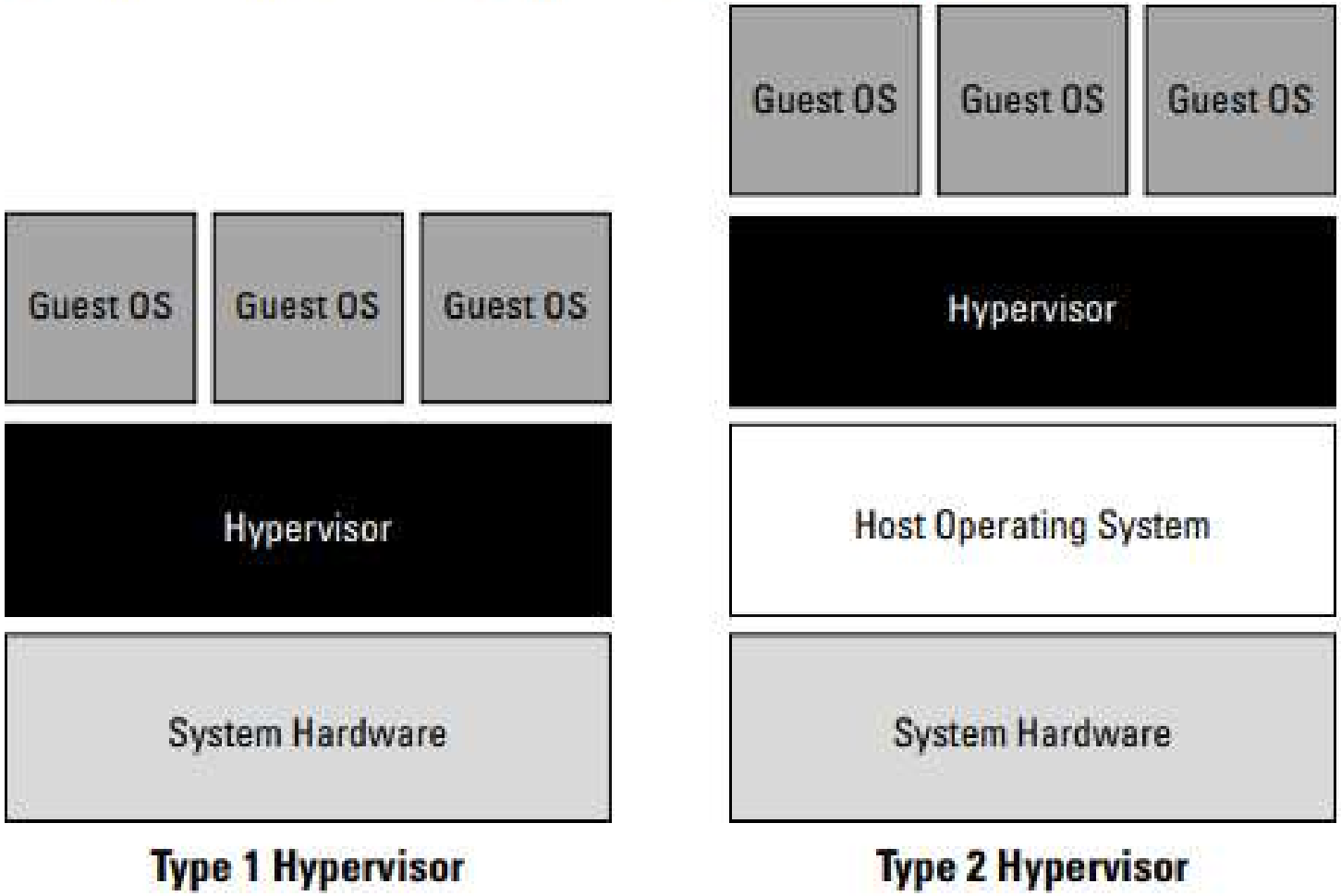
- Given a computer system with a certain set of resources, you can set aside portions of those resources to create a virtual machine.
- From the standpoint of applications or users, a virtual machine has all the attributes and characteristics of a physical system but is strictly software that emulates a physical machine.
- A system virtual machine (or a hardware virtual machine) has its own address space in memory, its own processor resource allocation, and its own device I/O using its own virtual device drivers.
- Some virtual machines are designed to run only a single application or process and are referred to as process virtual machines.

Virtual machine types

- A low-level program is required to provide system resource access to virtual machines, and this program is referred to as the hypervisor or Virtual Machine Monitor (VMM).
- Type 1
- Type2

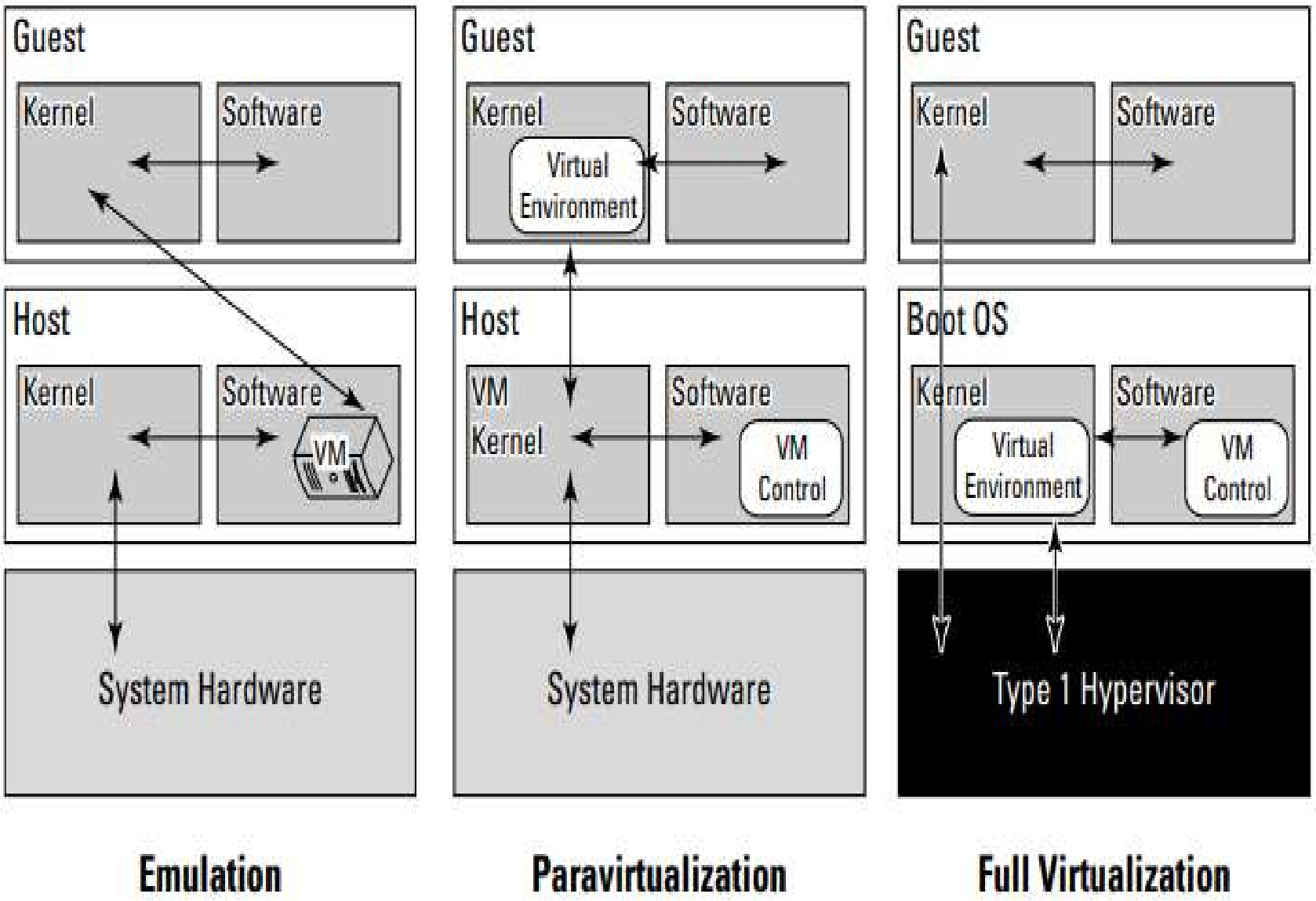
- A hypervisor running on bare metal is a Type 1 VM or native VM. Examples of Type 1 Virtual Machine Monitors are LynxSecure, RTS Hypervisor, Oracle VM, Sun xVM Server, VirtualLogix VLX, VMware ESX and ESXi, and Wind River VxWorks, among others.
- The operating system loaded into a virtual machine is referred to as the guest operating system, and there is no constraint on running the same guest on multiple VMs on a physical system. Type 1 VMs have no host operating system because they are installed on a bare system.

- Some hypervisors are installed over an operating system and are referred to as Type 2 or hosted VM.
- Examples of Type 2 Virtual Machine Monitors are Containers, KVM, Microsoft Hyper V, Parallels Desktop for Mac, Wind River Simics, VMWare Fusion, Virtual Server 2005 R2, Xen, Windows Virtual PC, and VMware Workstation 6.0


Guest OS
Guest OS
Guest OS
Hypervisor
System Hardware
Type 1 Hypervisor
Guest OS
Guest OS
Guest OS
Hypervisor
Host Operating System
System Hardware
Type 2 Hypervisor

Emulation, paravirtualization, and full virtualization types

- Emulation :In emulation, the virtual machine simulates hardware, so it can be independent of the underlying system hardware. A guest operating system using emulation does not need to be modified in any way.
- Paravirtualization : Paravirtualization requires that the host operating system provide a virtual machine interface for the guest operating system and that the guest access hardware through that host VM. An operating system running as a guest on a paravirtualization system must be ported to work with the host interface.
- Full virtualization: In full virtualization scheme, the VM is installed as a Type 1 Hypervisor directly onto the hardware. All operating systems in full virtualization communicate directly with the VM hypervisor, so guest operating systems do not require any modification.
- Guest operating systems in full virtualization systems are generally faster than other virtualization schemes.


Guest
Kernel
Software
Host
Kernel
Software
VM
System Hardware
Emulation
Guest
Kernel
Virtual Environment
Software
Host
VM Kernel
Software
VM Control
System Hardware
Paravirtualization
Guest
Kernel
Software
Boot OS
Kernel
Virtual Environment
Software
VM Control
Type 1 Hypervisor
Full Virtualization

Porting Applications

- Cloud computing applications have the ability to run on virtual systems and for these systems to be moved as needed to respond to demand.
- Developers who write software to run in the cloud will undoubtedly want the ability to port their applications from one cloud vendor to another, but that is a much more difficult proposition.

- Portability means that you can move an application from one host environment to another, including cloud to cloud such as from Amazon Web Services to Microsoft Azure.
- The work needed to complete the porting of an application from one platform to another depends upon the specific circumstances.
- Containers are one technology meant to make such porting easier, by encapsulating the application and operating systems into a bundle that can be run on a platform that supports that container standard like Docker or Kubernetes.

- The cloud computing portability and interoperability categories to consider are :
  - Data Portability
  - Application Portability
  - Platform Portability

Data Portability

- Data portability enables re-use of data components across different applications.
- Suppose that an enterprise uses a SaaS product for Customer Relations Management (CRM), for example, and the commercial terms for use of that product become unattractive compared with other SaaS products or with use of an in-house CRM solution. The customer data held by the SaaS product may be crucial to the enterprise's operation. How easy will it be to move that data to another CRM solution?
- In many cases, it will be very difficult. The structure of the data is often designed to fit a particular form of application processing, and a significant transformation is needed to produce data that can be handled by a different product.
- This is no different from the difficulty of moving data between different products in a traditional environment. But, in a traditional environment, the customer is more often able to do nothing; to stay with an old version of a product, for example, rather than upgrading to a newer, more expensive one. With SaaS, the vendor can more easily force the customer to pay more or lose the service altogether.

Application Portability

- Application portability enables the re-use of application components across cloud PaaS services and traditional computing platforms.
- Suppose that an enterprise has an application built on a particular cloud PaaS service and, for cost, performance, or other reasons, wishes to move it to another PaaS service or to in-house systems. How easy will this be?
- If the application uses features that are specific to the platform, or if the platform interface is non-standard, then it will not be easy.
- Application portability requires a standard interface exposed by the supporting platform.

- A particular application portability issue that arises with cloud computing is portability between development and operational environments.
- Cloud PaaS is particularly attractive for development environments from a financial perspective, because it avoids the need for investment in expensive systems that will be unused once the development is complete.
- But, where a different environment is to be used at run time – either on in-house systems or on different cloud services – it is essential that the applications can be moved unchanged between the two environments.
- Cloud computing is bringing development and operations closer together, and indeed increasingly leading to the two being integrated as *devops*.
- This can only work if the same environment is used for development and operation, or if there is application portability between development and operation environments.

Platform Portability

- There are two kinds of platform portability:
- Re-use of platform components across cloud IaaS services and non-cloud infrastructure – *platform source portability*
- Re-use of bundles containing applications and data with their supporting platforms – *machine image portability*
- The UNIX operating system provides an example of platform source portability. It is mostly written in the C programming language, and can be implemented on different hardware by re-compiling it and re-writing a few small hardware-dependent sections that are not coded in C.
- Some other operating systems can be ported in a similar way. This is the traditional approach to platform portability. It enables applications portability because applications that use the standard operating system interface can similarly be re-compiled and run on systems that have different hardware.

- Machine image portability gives enterprises and application vendors a new way of achieving applications portability, by bundling the application with its platform and porting the resulting bundle.
- It requires a standard program representation that can be deployed in different IaaS use environments.

The Simple Cloud API

- If you build an application on a platform such as Microsoft Azure, porting that application to Amazon Web Services or GoogleApps may be difficult, if not impossible.
- In an effort to create an interoperability standard, Zend Technologies has started an open source initiative to create a common application program interface that will allow applications to be portable. The initiative is called the Simple API for Cloud Application Services , and the effort has drawn interest from several major cloud computing companies.
- Among the founding supporters are IBM, Microsoft, Nivanix, Rackspace, and GoGrid.
- Simple Cloud API has as its goal a set of common interfaces for:
- **File Storage Services:** Currently Amazon S3, Windows Azure Blob Storage, Nirvanix, and Local storage is supported by the Storage API. There are plans to extend this API to Rackspace Cloud Files and GoGrid Cloud Storage.
- **Document Storage Services:** Amazon SimpleDB and Windows Azure Table Storage are currently supported. Local document storage is planned.
- **Simple Queue Services:** Amazon SQS, Windows Azure Queue Storage, and Local queue services are supported.

Capacity Planning

- Why bother doing capacity planning for a resource that is both ubiquitous and limitless?
- The reality of cloud computing is rather different than the ideal might suggest; cloud computing is neither ubiquitous, nor is it limitless.
- Often, performance can be highly variable, and you pay for what you use.
- Capacity planning for a cloud computing system offers you many enhanced capabilities and some new challenges over a purely physical system.
- A capacity planner seeks to meet the future demands on a system by providing the additional capacity to fulfill those demands.
- Capacity planning measures the maximum amount of work that can be done using the current technology and then adds resources to do more work as needed.

- Capacity planning is an iterative process with the following steps:
- 1. Determine the characteristics of the present system.
- 2. Measure the workload for the different resources in the system: CPU, RAM, disk, network, and so forth.
- 3. Load the system until it is overloaded, determine when it breaks, and specify what is required to maintain acceptable performance. Knowing when systems fail under load and what factor(s) is responsible for the failure is the critical step in capacity planning.
- 4. Predict the future based on historical trends and other factors.
- 5. Deploy or tear down resources to meet your predictions. 6. Iterate Steps 1 through 5 repeatedly.

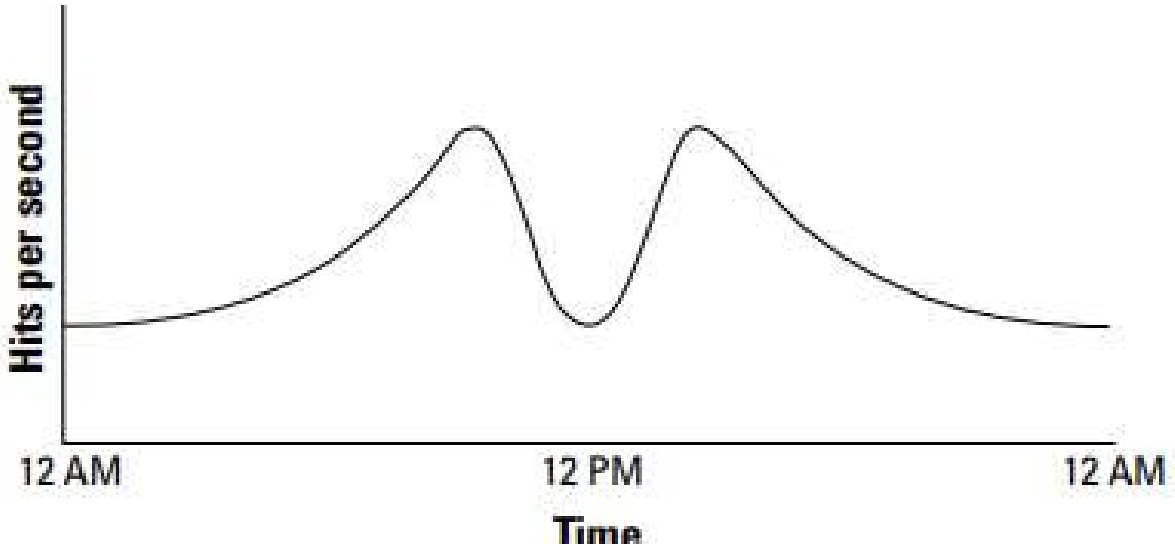
Defining Baseline and Metrics

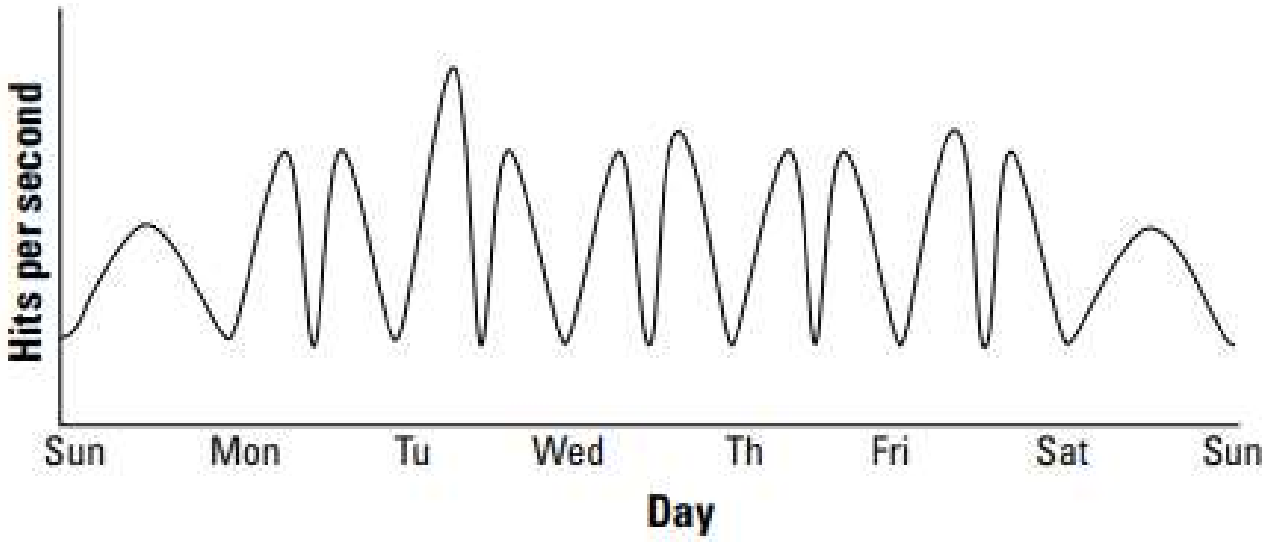
- The first item of business is to determine the current system capacity or workload as a measurable quantity over time. Because many developers create cloud-based applications and Web sites based on a LAMP solution stack
- LAMP stands for:
  - Linux, the operating system
  - Apache HTTP Server the Web server based on the work of the Apache Software Foundation
  - MySQL the database server developed by the Swedish company MySQL AB, owned by Oracle Corporation through its acquisition of Sun Microsystems
  - PHP the Hypertext Preprocessor scripting language developed by The PHP Group

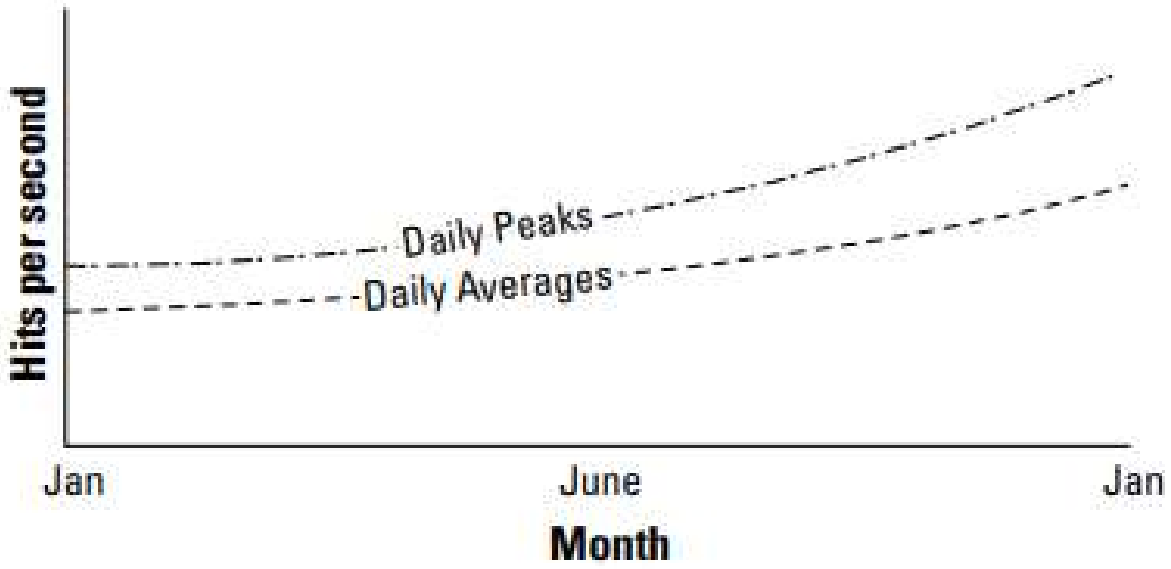
Baseline measurements

- Let's assume that a capacity planner is working with a system that has a Web site based on APACHE, and let's assume the site is processing database transactions using MySQL.
- There are two important overall workload metrics in this LAMP system:
- Page views or hits on the Web site, as measured in hits per second.
- Transactions completed on the database server, as measured by transactions per second or perhaps by queries per second

- historical record for the Web server page views over a hypothetical day, week, and year are graphed




Hits per second
Sun
Mon
Tu
Wed
Th
Fri
Sat
Sun
Day


Hits per second
Daily Peaks
Daily Averages
Jan
June
Jan
Month

- **W_T**, the total workload for the system per unit time. To obtain W_T, you need to integrate the area under the curve for the time period of interest.
- **W_{AVG}**, the average workload over multiple units of time To obtain W_{AVG}, you need to sum various W_T 's and divide by the number of unit times involved.
- **W_{MAX}**, the highest amount of work recorded by the system This is the highest recorded system utilization.
- **W_{TOT}**, the total amount of work done by the system, which is determined by the sum of W_T (ΣW_T)