Capacity Planning

- Capacity planning for cloud computing sounds like an oxymoron. 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.
- 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. Many people equate capacity planning with system optimization but they are not the same.
- System optimization aims to get more production from the system components you have. 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.
- If system optimization occurs during capacity planning, that is all to the good; but capacity planning efforts focus on meeting demand. If that means the capacity planner must accept the inherent inefficiencies in any system.

Capacity Planning

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.
- 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 (http://httpd.apache.org/), the Web server based on the work of the Apache Software Foundation
- MySQL (http://www.mysql.com), the database server developed by the Swedish company MySQL AB, owned by Oracle Corporation through its acquisition of Sun Microsystems.
- PHP (http://www.php.net/), the Hypertext Preprocessor scripting language developed by The PHP Group.

Defining Baseline and Metrics

- Four technologies are open source products, although the distributions used may vary from cloud to cloud and from machine instance to machine instance.
- On Amazon Web Services, machine instances are offered for both Red Hat Linux and for Ubuntu.
- LAMP stacks based on Red Hat Linux are more common than the other distributions, but SUSE Linux and Debian GNU/Linux are also common.
- Variants of LAMP are available that use other operating systems such as the Macintosh, OpenBSD (OpAMP), Solaris (SAMP), and Windows (WAMP).
- Although many other common applications are in use, LAMP is good to use as an example because it offers a system with two applications (APACHE and MySQL) that can be combined or run separately on servers.
- LAMP is one of the major categories of Amazon Machine Instances that you can create on the Amazon Web Service.

Baseline Measurements

- 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.

Two important overall workload metrics in this LAMP system:

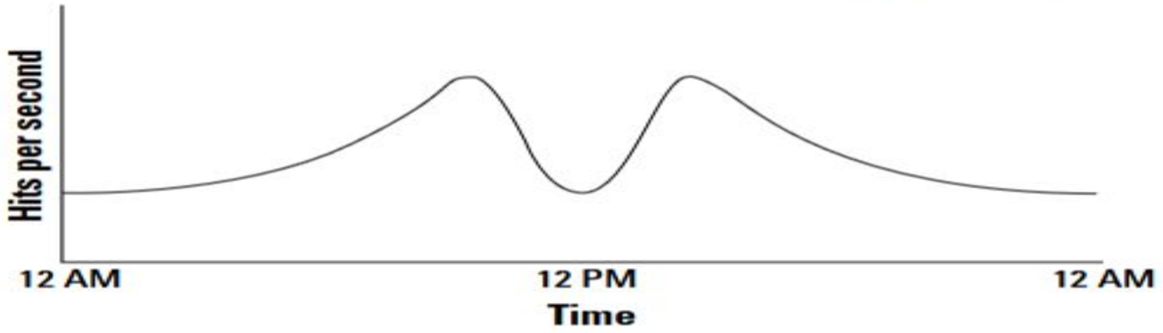
- **Page views** or hits on the Web site, as measured in hits per second.
- **Transactions** completed on the database server, as measured by transactions per second or perhaps by queries per second.
- The historical record for the Web server page views over a hypothetical day, week, and year are graphed.
- These graphs are created by summing the data from the different servers contained in the performance Web logs of the site or measuring throughput from the system based on an intervening service.
- A typical daily log (shown on top) shows two spikes in demand, one that is centered around 10 AM EST when users on the east coast of the United States use the site heavily and another at around 1 PM EST.
- When users on the west coast of the United States use the site heavily. The three-hour time difference is also the difference between Eastern Standard Time and Pacific Standard Time.

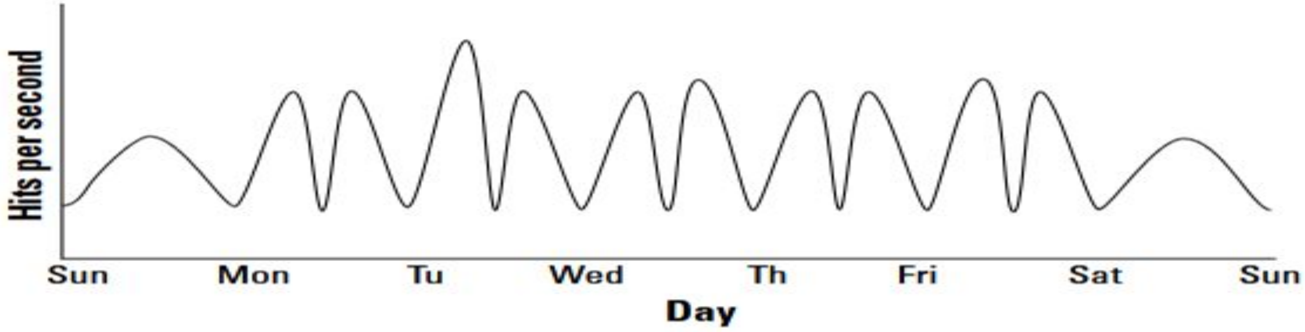
Baseline Measurements

- These daily performance spikes occur on workdays, but the spikes aren’t always of equal demand on weekends. The demand rises through the middle of the day and then ebbs later.
- Some days show a performance spike, as shown on the Tuesday morning section of the weekly graph. A goal of capacity planning is to correlate these performance spikes and dips with particular events and causations.
- Also, at some point your traffic patterns may change, so you definitely want to evaluate these statistics on an ongoing basis.
- The yearly graph, at the bottom of Figure 6.1, shows that the daily averages and the daily peaks rise steadily over the year and, in fact, double from January 1st at the start of the year to December 31st at year end.
- Knowing this, a capacity planner would need to plan to serve twice the traffic while balancing the demands of peak versus average loads. However, it may not mean that twice the resources need to be deployed.
- The amount of resources to be deployed depends upon the characterization of the Web servers involved, their potential utilization rates, and other factors.

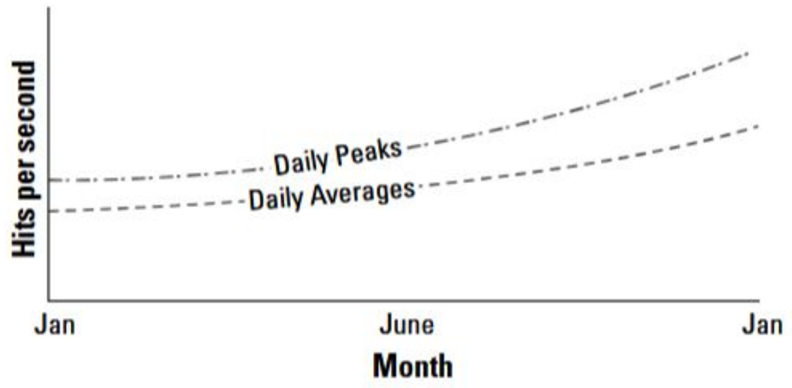
Baseline Measurements

A Web servers' workload measured on a day, a week, and over the course of a year





Baseline Measurements



A number of important characteristics are determined by these baseline studies:

- WT the total workload for the system per unit time To obtain WT you need to integrate the area under the curve for the time period of interest.

- WAVG, the average workload over multiple units of time To obtain WAVG you need to sum various WTs and divide by the number of unit times involved. You may also want to draw a curve that represents the mean work done.

Baseline Measurements

- WMAX the highest amount of work recorded by the system. This is the highest recorded system utilization.
- WTOT the total amount of work done by the system, which is determined by the sum of W T (ΣWT).
- A similar set of graphs would be collected to characterize the database servers, with the workload for those servers measured in transactions per second.
- As part of the capacity planning exercise, the workload for the Web servers would be correlated with the workload of the database servers to determine patterns of usage.
- The goal of a capacity planning exercise is to accommodate spikes in demand as well as the overall growth of demand over time.
- The growth in demand over time is the most important consideration because it represents the ability of a business to grow.
- A spike in demand may or may not be important enough to an activity to attempt to capture the full demand that the spike represents.

System Metrics

- Determined what amounts to application-level statistics for your site. These measurements are an aggregate of one or more Web servers in your infrastructure.
- Capacity planning must measure system-level statistics, determining what each system is capable of, and how resources of a system affect system-level performance.
- In some instances, the capacity planner may have some influence on system architecture, and the impact of system architecture on application- and system-level statistics may be examined and altered.

A machine instance is primarily defined by four essential resources:

- CPU
- Memory (RAM)
- Disk
- Network connectivity

System Metrics

- Each of these resources can be measured by tools that are operating-system-specific, but for which tools that are their counterparts exist for all operating systems.
- Indeed, many of these tools were written specifically to build the operating systems themselves, but that's another story.
- In Linux/UNIX, you might use the sar command to display the level of CPU activity. sar is installed in Linux as part of the sysstat package.
- In Windows, a similar measurement may be made using the Task Manager, the data from which can be dumped to a performance log and/or graphed.
- Some tools give you a historical record of performance, which is particularly useful in capacity planning.
- For example, the popular Linux performance measurement tool RRDTool(the Round Robin Database tool; http://oss.oetiker.ch/rrdtool/) is valuable in this regard.

System Metrics

- RRDTool is a utility that can capture time-dependent performance data from resources such as a CPU load, network utilization (bandwidth), and so on and store the data in a circular buffer.

- It is commonly used in performance analysis work. A time interval in RRDTool is called a step, with the value in a step referred to as a primary data point (PDP). When a function is applied to a data point.

- The function is referred to as a Consolidation Function (CF), and the value obtained is a Consolidated Data Point (CDP). An interval is stored in the Round Robin Archive (RRA), and when that interval is filled, it is replaced by new step data.

- RRDTool includes a graphical front end for displaying performance results visually. RRDTool is widely used for a number of different purposes.

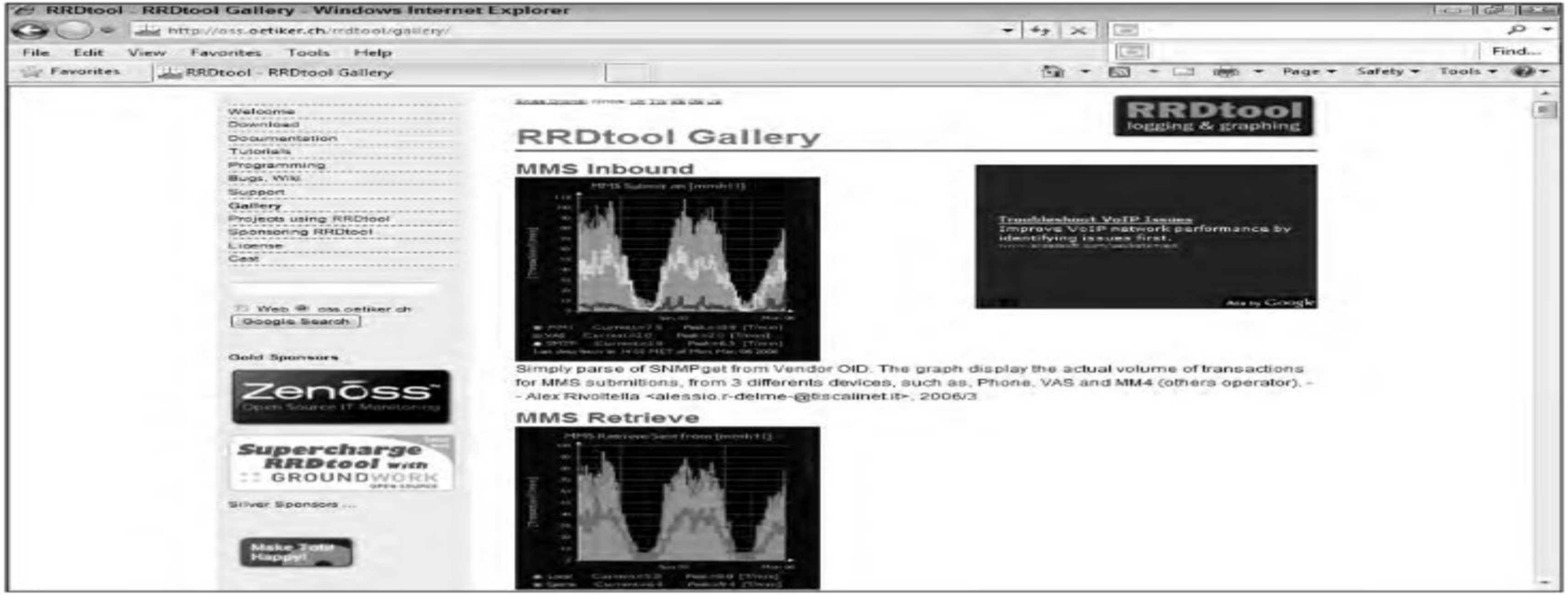
- Gallery of RRDTool graphs found at http://oss.oetiker.ch/rrdtool/gallery/.

RV College of Engineering®

System Metrics

Go, change the world®

RRDTool lets you create historical graphs of a wide variety of performance data. Some sam
in the gallery at http://oss.oetiker.ch/rrdtool/gallery/.



RRDtool Gallery

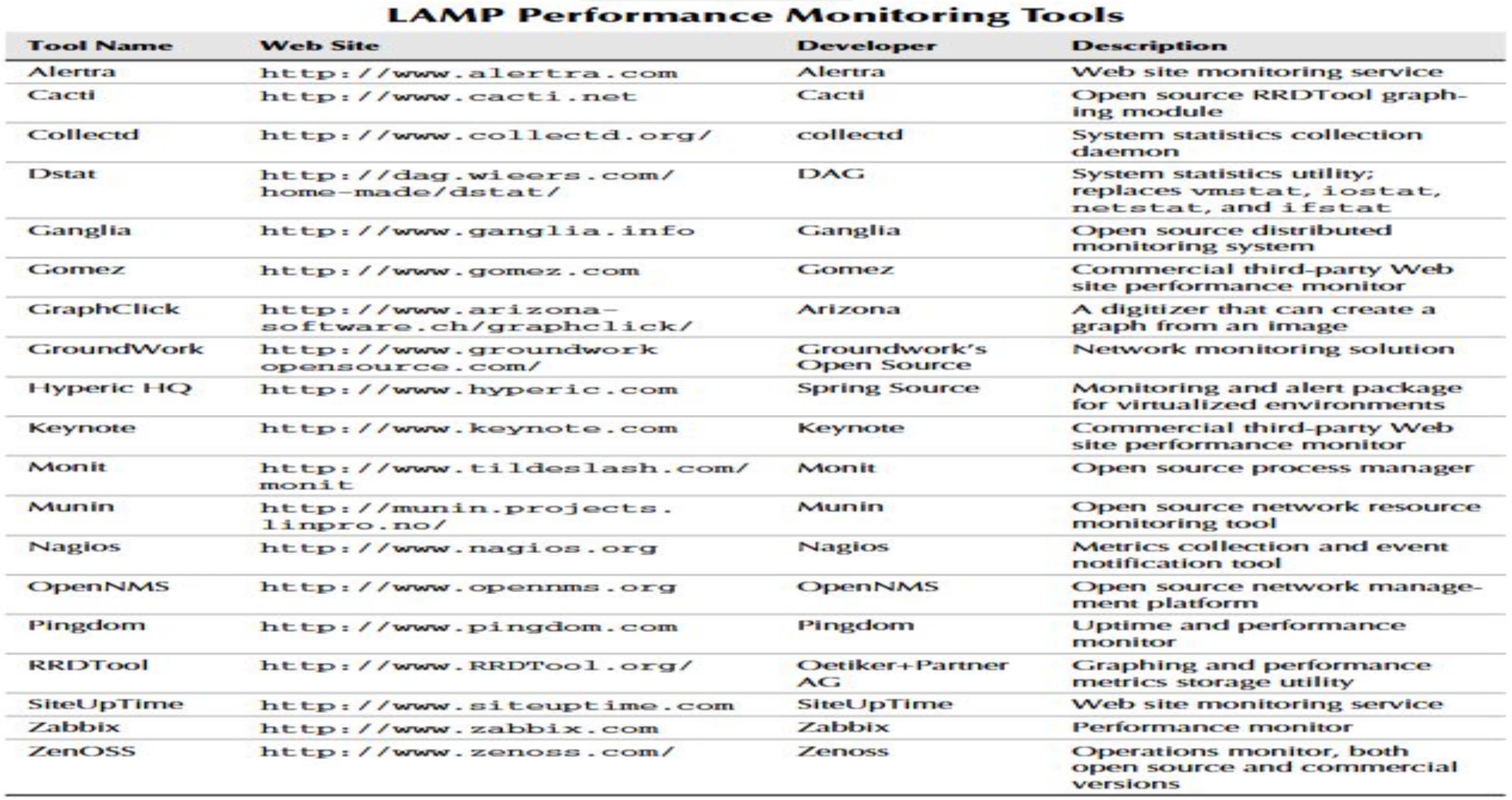
MMS Inbound

Simply parse of SNMPget from Vendor OID. The graph display the actual volume of transactions for MMS submitions, from 3 differents devices, such as, Phone, VAS and MM4 (others operator). -- Alex Rivoltella <alessio.r-delme-@tiscalinet.it>, 2006/3

MMS Retrieve

lists some LAMP performance testing tools.

System Metrics



LAMP Performance Monitoring Tools

Tool Name	Web Site	Developer	Description
Alertra	http://www.alertra.com	Alertra	Web site monitoring service
Cacti	http://www.cacti.net	Cacti	Open source RRDTool graphing module
Collectd	http://www.collectd.org/	collectd	System statistics collection daemon
Dstat	http://dag.wieers.com/home-made/dstat/	DAG	System statistics utility; replaces vmstat, iostat, netstat, and ifstat
Ganglia	http://www.ganglia.info	Ganglia	Open source distributed monitoring system
Gomez	http://www.gomez.com	Gomez	Commercial third-party Web site performance monitor
GraphClick	http://www.arizona-software.ch/graphclick/	Arizona	A digitizer that can create a graph from an image
GroundWork	http://www.groundworkopensource.com/	Groundwork's Open Source	Network monitoring solution
Hyperic HQ	http://www.hyperic.com	Spring Source	Monitoring and alert package for virtualized environments
Keynote	http://www.keynote.com	Keynote	Commercial third-party Web site performance monitor
Monit	http://www.tildeslash.com/monit	Monit	Open source process manager
Munin	http://munin.projects.linpro.no/	Munin	Open source network resource monitoring tool
Nagios	http://www.nagios.org	Nagios	Metrics collection and event notification tool
OpenNMS	http://www.opennms.org	OpenNMS	Open source network management platform
Pingdom	http://www.pingdom.com	Pingdom	Uptime and performance monitor
RRDTool	http://www.RRDTool.org/	Oetiker+Partner AG	Graphing and performance metrics storage utility
SiteUpTime	http://www.siteuptime.com	SiteUpTime	Web site monitoring service
Zabbix	http://www.zabbix.com	Zabbix	Performance monitor
ZenOSS	http://www.zenoss.com/	Zenoss	Operations monitor, both open source and commercial versions

Load Testing

- Examining your server under load for system metrics isn't going to give you enough information to do meaningful capacity planning.
- You need to know what happens to a system when the load increases. Load testing seeks to answer the following questions:
- What is the maximum load that my current system can support?
- Which resource(s) represents the bottleneck in the current system that limits the system's performance?
- This parameter is referred to as the resource ceiling. Depending upon a server's configuration, any resource can have a bottleneck removed, and the resource ceiling then passes onto another resource.
- Can I alter the configuration of my server in order to increase capacity?
- How does this server's performance relate to your other servers that might have different characteristics?

Load Testing

- If you have one production system running in the cloud, then overloading that system until it breaks isn't going to make you popular with your colleagues.
- However, cloud computing offers virtual solutions. One possibility is to create a clone of your single system and then use a tool to feed that clone a recorded set of transactions that represent your application workload.
- Two examples of applications that can replay requests to Web servers are
- HTTPerf (http:// hpl.hp.com/research/linux/httperf/) and Siege (http://www.joedog.org/ JoeDog/Siege),
- These tools run the requests from a single client, which can be a resource limitation of its own.
- You can run Autobench (http://www.xenoclast.org/auto bench/) to run HTTPerf from multiple clients against your Web server, which is a better test.

Load Testing

Consider these load generation tools as well:

- HP LodeRunner (https://h10078.www1.hp.com/cda/hpms/display/main/
- hpms_content.jsp?zn=bto&cp=1-11-126-17^8_4000_100__)
- IBM Rational Performance Tester (http://www-01.ibm.com/software/ awdtools/tester/performance/)
- JMeter (http://jakarta.apache.org/jmeter)
- OpenSTA (http://opensta.org/)
- Micro Focus (Borland) SilkPerfomer (http://www.borland.com/us/products/ silk/silkperformer/index.html)
- Load testing software is a large product category with software and hardware products.
- Load testing useful in testing the performance of not only Web servers, but also application servers, database servers, network throughput, load balancing servers, and applications that rely on client-side processing.

RV College of Engineering®

Load Testing

Go, change the world®

- When you have multiple virtual servers that are part of a server farm and have a load balancer in front of them, you can use your load balancer to test your servers' resource ceilings.
- This technique has the dual advantages of allowing you to slowly increment traffic on a server and to use real requests while doing so.
- Most load balancers allow you to weight the requests going to a specific server with the goal of serving more requests to more powerful systems and fewer requests to less powerful systems.
- Sometimes the load balancer does this optimization automatically, and other times you can exert manual control over the weighting.
- Whatever method you use to load a server for performance testing, you must pick a method that is truly representative of real events, requests, and operations. The best approach is to incrementally alter the load on a server with the current workload.
- When you make assumptions such as using a load balancer to serve traffic based on a condition that summarizes the traffic pattern instead of using the traffic itself, you can get yourself into trouble.

Load Testing

- Problems with load balancers have led to some spectacular system failures because those devices occupy a central controlling site in any infrastructure.

- For example, if you assume that traffic can be routed based on the number of connections in use per server and your traffic places a highly variable load based on individual requests, then your loading measurements can lead to dramatic failures when you attempt to alter your infrastructure to accommodate additional loads.

Resource ceilings :

- Whatever performance measurement tool you use, the goal is to create a set of resource utilization curves similar to the ones shown in Figure 6.3 for individual server types in your infrastructure.To do this, you must examine the server at different load levels and measure utilization rates.

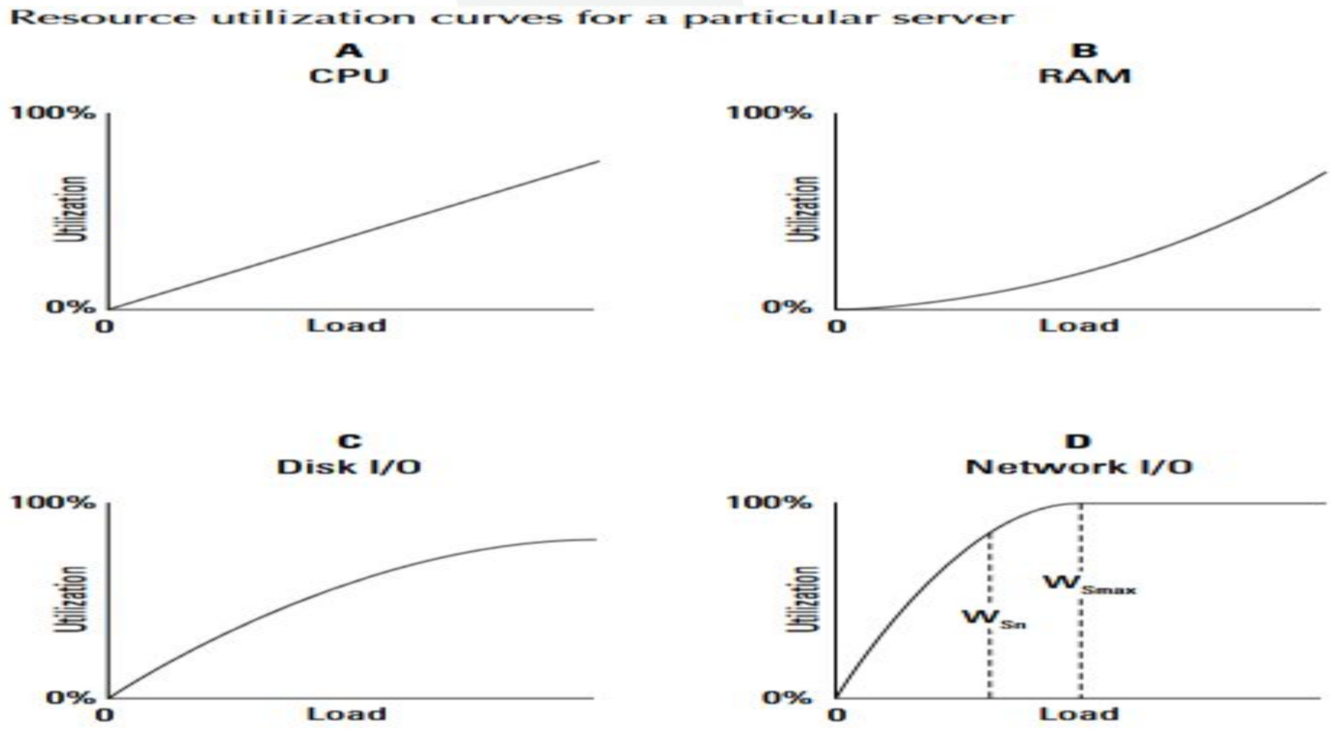
- The figure indicate that over a certain load for a particular server, the CPU (A), RAM (B), and Disk I/O (C) utilization rates rise but do not reach their resource ceiling.

- The Network I/O (D) reaches its maximum 100-percent utilization at about 50 percent of the tested load, and this factor is the current system resource ceiling.

Load Testing

- Network I/O is often a bottleneck in Web servers, and this is why, architecturally, Web sites prefer to scale out using many low-powered servers instead of scaling up with fewer but more powerful servers.
- Adding more (multi-homing) or faster network connections can be another solution to improving Web servers' performance.
- Unless you alter the performance of the server profiled in Figure 6.3 to improve it, you are looking at a maximum value for that server's workload of Wsmax as shown in the dashed line at the 50-percent load point in graph D.
- The server is overloaded and the system begins to fail. Some amount of failure may be tolerable in the short term, provided that the system can recover and not too many Web hits are lost, but this is a situation that you really want to minimize.
- Consider WSn to be the server's red line, the point at which the system should be generating alerts or initiating scripts to increase capacity.

Load Testing



Load Testing

- The parameter you are most interested in is likely to be the overall system capacity, which is the value of WT . WT is the sum over all the Web servers in your infrastructure: WT = Σ(WSnP W SnV).
- In this equation, W SnP represents the workload of your physical server(s) and WSnV is the workload of the virtual servers (cloud-based server instances) of your infrastructure.
- The amount of overhead you allow yourself should be dictated by the amount of time you require to react to the challenge and to your tolerance for risk.
- A capacity planner would define a value WT such that there is sufficient overhead remaining in the system to react to demand that is defined by a number greater than WMAX by bringing more resources on-line.
- For storage resources that tend to change slowly, a planner might set the red line level to be 85 percent of consumption of storage; for a Web server, that utilization percentage may be different. This setting would give you a 15-percent safety factor where to draw the red line.
- When you load test a system, you are applying a certain amount of overhead to the system from the load testing too a feature that is often called the “observer effect.

Load Testing

- Many load testers work by installing lightweight agents on the systems to be tested. Those agents themselves impact the performance you see; generally though, their impact is limited to a few percent.
- Additionally, in order to measure performance, you may be forced to turn on various performance counters. A performance counter is also an agent of sorts;
- The counter is running a routine that measures some aspect of performance such as queue length, file I/O rate, numbers of READs and WRITEs, and so forth.
- While these complications are second order effects, it's always good to keep this in mind and not over-interpret performance results.
- Before leaving the topic of resource ceilings, let's consider a slightly more complicated case that you might encounter in MySQL database systems. Database servers are known to exhibit resource ceilings for either their file caches or their Disk I/O performance.
- To build high-performance applications, many developers replicate their master MySQL database and create a number of slave MySQL databases. All READ operations are performed on the slave MySQL databases, and all WRITE operations are performed on the master MySQL database.

Load Testing

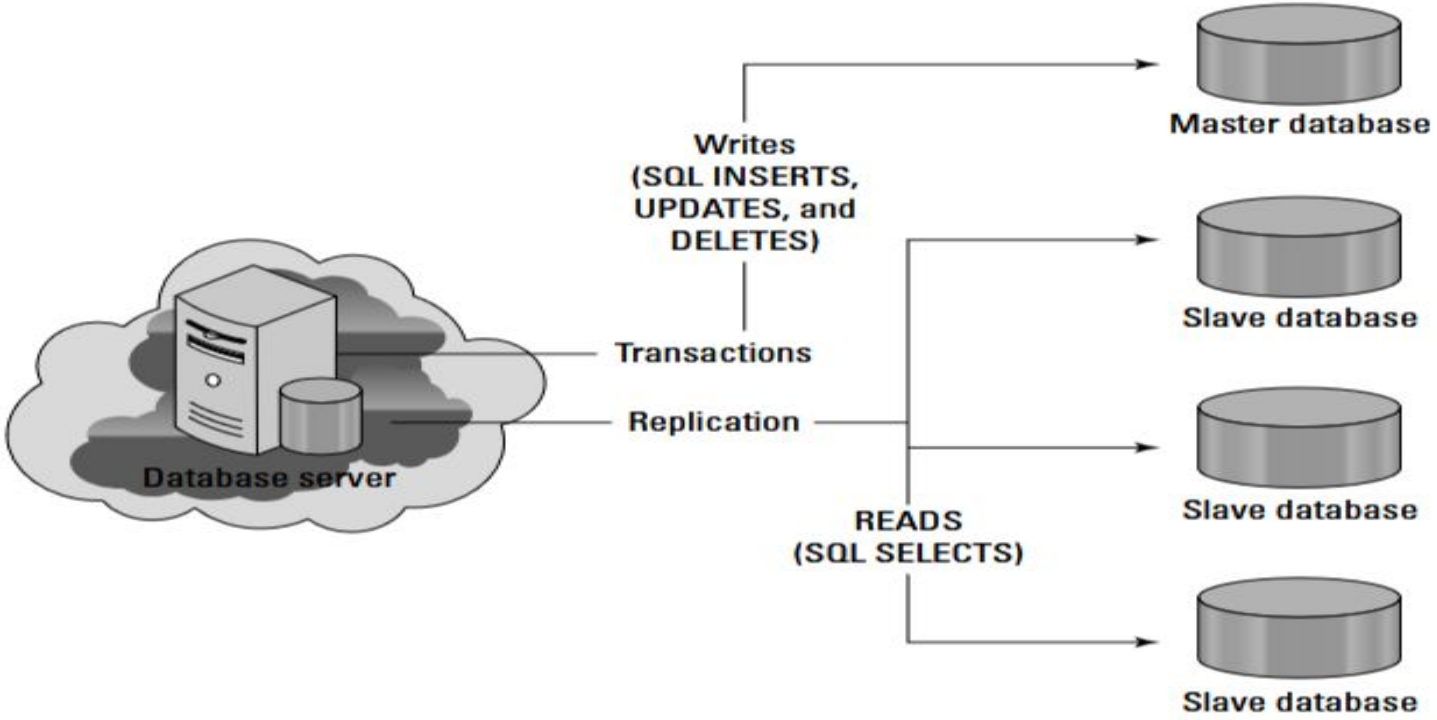
- A master/slave database system has two competing processes and the same server that are sharing a common resource: READs and WRITEs to the master/slave databases, and replication traffic between the master database and all the slave databases.
- The ratio of these transactions determines the WSn for the system, and the WSn you derive is highly dependent upon the system architecture.
- These types of servers reach failure when the amount of WRITE traffic in the form of INSERT, UPDATE, and DELETE operations to the master database overtakes the ability of the system to replicate data to the slave databases that are servicing SELECT (query/READ) operations.
- The more slave databases there are, the more actively the master database is being modified and the lower the WS for the master database is.
- This system may support no more than 25-35 percent transactional workload as part of its overall capacity, depending upon the nature of the database application you are running.

Load Testing

- Increase the working capacity of a database server that has a Disk I/O resource ceiling by using more powerful disk arrays and improving the interconnection or network connection used to connect the server to its disks.
- Disk I/O is particularly sensitive to the number of spindles in use, so having more disks equals greater performance. Keep in mind that your ability to alter the performance of disk assets in a virtual or cloud-based database server is generally limited.
- In the first case, a single server has a single application with a single resource ceiling. In the second case, you have a single server that has a single application with two competing processes that establish a resource ceiling.
- What do you do in a situation where your server runs the full LAMP stack and you have two or more applications running processes each of which has its own resource ceiling?
- In this instance, you must isolate each application and process and evaluate their characteristics separately while trying to hold the other applications' resource usage at a constant level.
- Do this by examining identical servers running the application individually or by creating a performance console that evaluates multiple factors all at the same time. Remember, real-world data and performance is always preferred over simulation

Load Testing

Resource contention in a database server



Load Testing

- As your infrastructure grows, it becomes more valuable to create a performance console that lets you evaluate your KPIs (Key Performance Indicators) graphically at an instant.
- Many third-party tools offer performance consoles, and some tools allow you to capture their state so you can restore it at a later point.
- An example of a performance analysis tool that lets you save its state is the Microsoft Management Console (MMC).
- In the Amazon Web Service, the statistics monitoring tool is called Amazon CloudWatch, and you can create a performance monitoring console from the statistics it collects.