ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

How best Capacity Planning can be done in an IT Organization

Srinivasa Rao Kosiganti[*1], R. Hari Singh[2]
[*1]Architect, Tech Mahindra, Hyderabad, India
[2]Lecturer, SV Degree College, Ramanthapur, Hyderabad, india
srinikosi@gmail.com

Abstract

- It is very common for an IT Organization to manage system performance in a reactionary fashion, analyzing, and correcting performance problems as users report them. When problems occur, hopefully system administrators have tools necessary to quick analyze and remedy the situation. In a perfect world, administrators prepare in advance in order to avoid bottlenecks altogether, using capacity planning tools to predict in advance how servers would be configured to adequately handle future workloads.
- The goal of capacity planning is to provide satisfactory service levels to the users in a cost-effective manner. Obviously the end goal of capacity planning is a smooth and speedy experience for the users. Several factors can affect the user's experience beside capacity. It's possible to have plenty of capacity but a slow website nonetheless. Even though capacity is only one part of making the end-user experience fast, that experience is still one of the real-world metrics that we will want to measure and track in order make forecasts. This paper describes the best practices to follow for capacity planning.

Keywords: IT, Capacity Planning.

Introduction

Three steps for Capacity Planning being illustrated here:

1. Determine Service Level Requirements The first step in the capacity planning process is to categorize the work done by systems and to quantify users' expectations for how that work gets done.
2. Analyze Current Capacity Next, the current capacity of the system must be analyzed to determine how it is meeting the needs of the users.
3. Planning for the future Finally, using forecasts of future business activity, future system requirements are determined. Implementing the required changes in system configuration will ensure that sufficient capacity will be available to maintain service levels, even as circumstances change in the future.

Determine Service Level Requirements

This section can be described as follows:

a. The overall process of establishing service level requirements first demands an understanding of workloads. Workloads play an important role in establishing what kind of service levels can be suitable for the visualized workloads.
b. Next, we begin an example, showing workloads on a system running a back-end Oracle database.
c. Before setting service levels, you need to determine what unit you will use to measure the incoming work.
d. Finally, you establish service level requirements, the promised level that will be provided by the IT organization.

Workloads Explained

From a capacity planning perspective, a computer system processes workloads (which supply the demand) and delivers service to users. During the first step in the capacity planning process, these workloads must be defined and a definition of satisfactory service must be created. A workload is a logical classification of work performed on a computer system. If you consider all the work performed on your systems as pie, a workload can be thought of as some piece of that pie. Workloads can be classified by a wide variety of criteria.

Who - is doing the work (particular user or department)
What - type of work is being done(order entry, financial reporting)

ISSN: 2277-9655
Impact Factor: 1.852

How - the work is being done(online inquiries, batch database backups)

It is useful to analyze the work done on systems in terms that make sense from a business perspective, using business-relevant workload definitions.

For example, if you analyze performance based on workloads corresponding to business departments, then you can establish service level requirements for each of those departments.

Business-relevant workloads are also useful when it comes time to plan for the future. It is much easier to project future work when it is expressed in terms that make business sense.

For example, it accounts payable department on a consolidated server than it is to predict the overall increase in transactions for that server.

Determine the Unit of Work

For capacity planning purposes it is useful to associate a unit of work with a workload. This is a measurable quantity of work done, as opposed to the amount of system resources required to accomplish that work.

To understand the difference, consider measuring the work done at a fast food restaurant. When deciding on the unit of work, you might consider counting the number of customers served, the weight of the food served, the number of sandwiches served, or the money taken in for the food served. This is as opposed to the resources used to accomplish the work, i.e. the amount of French fries, raw hamburgers or pickle slices used to produce the food served to customers.

When talking about IT performance, instead of French fries, raw hamburger or pickle slices, we accomplish work using resources such as disk, I/O channels, CPUs and network connections.

Measuring the utilization of these resources is important for capacity planning, but not relevant for determining the amount of work done or the unit of work. Instead, for an online workload, the unit of work may be a transaction. For an interactive or batch workload, the unit of work may be a process.

Establish Service Levels

The next step now is to establish a service level agreement. A service level agreement is an agreement between the service provider and service consumer that defines acceptable service.

The service level agreement is often defined from the user's perspective, typically in terms of response time or throughput. Using workloads often aids in the process of developing service level agreements, because workloads can be used to measure system performance in ways that makes sense to clients/users.

In the case of our appointment scheduling application, we might establish service level requirements regarding the number of requests that should be processed within a given period of time, or we might require that each request be processed within a certain time limit.

These possibilities are analogous to a fast food restaurant requiring that a certain number of customers should be serviced per hour during the lunch rush, or that each customer should have to wait no longer than three minutes to have his or her order filled.

Ideally, service level requirements are ultimately determined by business requirements.

Frequently, however, they are based on past experience. It's better to set service level requirements to ensure that you will accomplish your business objectives, but not surprisingly people frequently resort to setting service level requirements like, "provide a response time at least as good as is currently experienced, even after we ramp up our business." As long as you know how much the business will "ramp up," this sort of service level requirement can work.

If you want to base your service level requirements on present actual service levels, then you may want to analyze your current capacity before setting your service levels.

Analyze Current Capacity

There are several steps that should be performed during the analysis of capacity measurement data.

a) First, compare the measurements of any items referenced in service level agreements with their objectives. This provides the basic indication of whether the system has adequate capacity.
b) Next, check the usage of the various resources of the system (CPU, memory, and I/O devices). This analysis identifies highly used resources that may prove problematic now or in the future.
c) Look at the resource utilization for each workload. Ascertain which workloads are the major users of each resource. This helps narrow your attention to only the workloads that are making the greatest demands on system resources.
d) Determine where each workload is spending its time by analyzing the components of response time, allowing you to determine which system

ISSN: 2277-9655
Impact Factor: 1.852

resources are responsible for the greatest portion of the response time for each workload.

Identify Components of Response Time

Next we will show how to determine what system resources are responsible for the amount of time that is required to process a unit of work. The resources that are responsible for the greatest share of the response time are indicators for where you should concentrate your efforts to optimize performance.

Determine Future Process Requirements

Systems may be satisfying service levels now, but will they be able to do that while at the same time meeting future organizational needs? Besides service level requirements, the other key input into the capacity planning process is a forecast or plan for the organization's future. Capacity planning is really just a process for determining the optimal way to satisfy business requirements such as forecasted increases in the amount of work to be done, while at the same time meeting service level requirements. Future processing requirements can come from a variety of sources. Input from management may include:

- Expected growth in the business
- Requirements for implementing new applications
- Planned acquisitions or divestitures
- IT budget limitations
- Requests for consolidation of IT resources

Additionally, future processing requirements may be identified from trends in historical measurements of incoming work such as orders or transactions.

Architecture Decisions

The Architecture is the basic layout of how all of the backend pieces- both hardware and software - are joined. Its design plays a crucial role in one's ability to plan and manage capacity. Designing the architecture can be a complex undertaking. The architecture affects nearly every part of performance, reliability, and management. Establishing good architecture almost always translates to easier effort when planning for capacity.

Providing measurement Points

Both for measurements purposes as well as for rapid response to changing conditions, you want your architecture to be designed so you can easily split it into parts that perform discrete tasks. In an ideal world, each component of the backend should have a single job to do, but it could still do multiple jobs well, if needed. At the same time, its effectiveness on each job should be easy to measure.

For example, let us look at a simple, database-driven web application just starting on the path toward world domination. To get the most bang for our buck, we have our web server and database residing on the same hardware server. This means all the moving parts share the same hardware resources.

We need to understand the following nuances from the above.

Is the disk utilization the result of the web server sending out a lot of static content from the disk, or rather, the database's queries being disk-bound?

How much of the file system cache, CPU, memory, and disk utilization is being consumed by the web server, and how much is being used for the database.

With careful research, you can make some estimates about which daemon is using which resource. In the best case, the resource demands of the different daemons don't contend with one number. For example, the web server might be bound mostly by CPU and not need much memory, whereas the database might be memory-bound without using much CPU. But even in this ideal scenario, if usage continues to grow, the resource contention will grow to warrant splitting the architecture into different hardware components.

If we are recording system and application-level statistics, you can quantify what each unit of capacity means in terms of usage. With this new architecture, you can answer a few questions that you couldn't before, such as:

Database Server

How do increases in database queries-per-second affect the following?

- Disk Utilization
- I/O Wait(Percent of time the database waits due to network or disk operations)
- RAM Usage
- CPU Usage

Web Server

How do increases in web server requests-per-second affect the following?

- Disk Utilization
- I/O Wait
- RAM Usage
- CPU Usage

Being able to answer these questions is key to establishing how (and when) you'll wiant to add more capacity to each piece.

ISSN: 2277-9655
Impact Factor: 1.852

The last piece that one miss in the discussion on architecture is what drives capacity forecasting: *resource ceilings*

The questions that can posed regarding the effects of usage on resources, point to an obvious culmination: *When will the database or web server die*?

Hardware decisions(Vertical, Horizontal, and Diagonal Scaling)

Being able to scale horizontally means having an architecture that allows for adding capacity by simply adding similarly functioning nodes to the existing infra. For instance, a second web server to share the burden of website visits.

Being able to scale vertically is the capability of adding capacity by increasing the resources internal to a server, such as CPU, memory, disk, and network.

Since the emergence of tiered and *shared-nothing* architectures, horizontal scaling has been widely recognized for its advantages over vertical scaling as it pertains to web applications.

The danger of relying solely on vertical scaling is, as you continue to upgrade components of a single computer, the cost rises dramatically. You also introduce the risk of *a single point of failure*.

Diagonal scaling is the process of vertically scaling the horizontally scaled nodes you already have in your infra. Over time, CPU power and RAM become faster, cheaper and cooler, and disk storage becomes larger and less expensive, so it can be cost effective to keep some vertical scaling as part of the capacity planning, but applied to horizontal nodes.

Applications Monitoring

We should collect application-level metrics. They are collected on a daily and cumulative basis.

Some of the metrics that can be tracked are:

- Photos uploaded (daily, cumulative)
- Photos uploaded per hour
- Average Photo Size(daily, cumulative)
- Processing time to segregate photos based on their different sizes (hourly)
- User registrations(daily, cumulative)
- Pro account signups(daily, cumulative)
- No of photos tagged(daily, cumulative)
- API traffic (API keys in use, requests made per second, per key)
- Number of unique tags (daily, cumulative)
- Number of geotagged photos(daily, cumulative)

Storage Capacity

One of the most effective storage analogies is that of a glass of water. The analogy combines a finite limit (the size of the glass) with a variable (the amount of water that can be put into and taken out of the glass at any given time). This helps you to visualize the two major factors to consider when chossing where and how to store your data:

- The maximum capacity of the storage media
- The rate at which the can be accessed

Traditionally, most web operations have been concerned with the first consideration-the size of the glass. However, most commercial storage vendors have aligned their product lines with both considerations in mind. In most cases, there are two options: large, slow, inexpensive disks(usually using ATA/SATA), and smaller, fast, expensive disks(SCSI and SAS technologies).

Caching Systems

Disks are the slowest pieces of infrastructure, which makes accessing them expensive in terms of time. Most large-scale sites alleviate the need for making these expensive operations by caching data in various locations.

In Web architectures, caches are most often used to store database results(as with Memcached) or actual files. Both approaches call for the same considerations with respect to capacity planning. They are examples of reverse proxies, which are specialized systems that cache data sent from the web server to the client(usually a web browser).

Special Use and Multiple Use Servers

In order to discern what processes are consuming which resources, you'll want to do one of the following:

- Isolate each running application and measure its resource consumption
- Hold some of the applications' resource usage constant in order to measure one at a time

Conclusion

Forecasting capacity needs is part intuition, and part math. Its also the art of slicing and dicing up your historical data, and making educated guesses about the future. Outside of those rare bursts and spikes of load on your system, the long-term view is hopefully one of

ISSN: 2277-9655
Impact Factor: 1.852

steadily increasing usage. But putting all that historical data into perspective, you can generate estimates for what you'll need to sustain the growth of your website. The key to making accurate predictions is having an *adjustable* forecasting process.

References

[1] How to do Capacity Planning - TeamQuest, Internet
[2] The art of Capacity Planning by John All spaw Internet