Lesson 4
Capacity Planning

- Reading list
  - Cloud Computing bible, B.Sosinsky, John Wiley & Sons, 2010 (for this lesson chapter 6)
  - Cloud Computing: principles and paradigms, R.Buyya, 2011 (for this lesson pp:166-191)

Capacity planning (CP) definition

- Match demands to available resources
- Examines what system are in place
- Measure their performances
- Determines patterns to predict demands
- Resources are provisioned and allocated to meet demands

Why we need Capacity Planning?

- According to the definition of cloud, there is contradiction in this concept !
- We said that resources are ubiquitous, infinite and unlimited !
- Then why we need the capacity planning ?
- The answer is : in big companies such as Google the above claim is sort of true (despite saying that, they also need planning), however in small private clouds it's not the case
- In such a environments, there is need of capacity planning, also :
  - To make it more profitable
  - Have less queues as possible
  - maximize efficiency

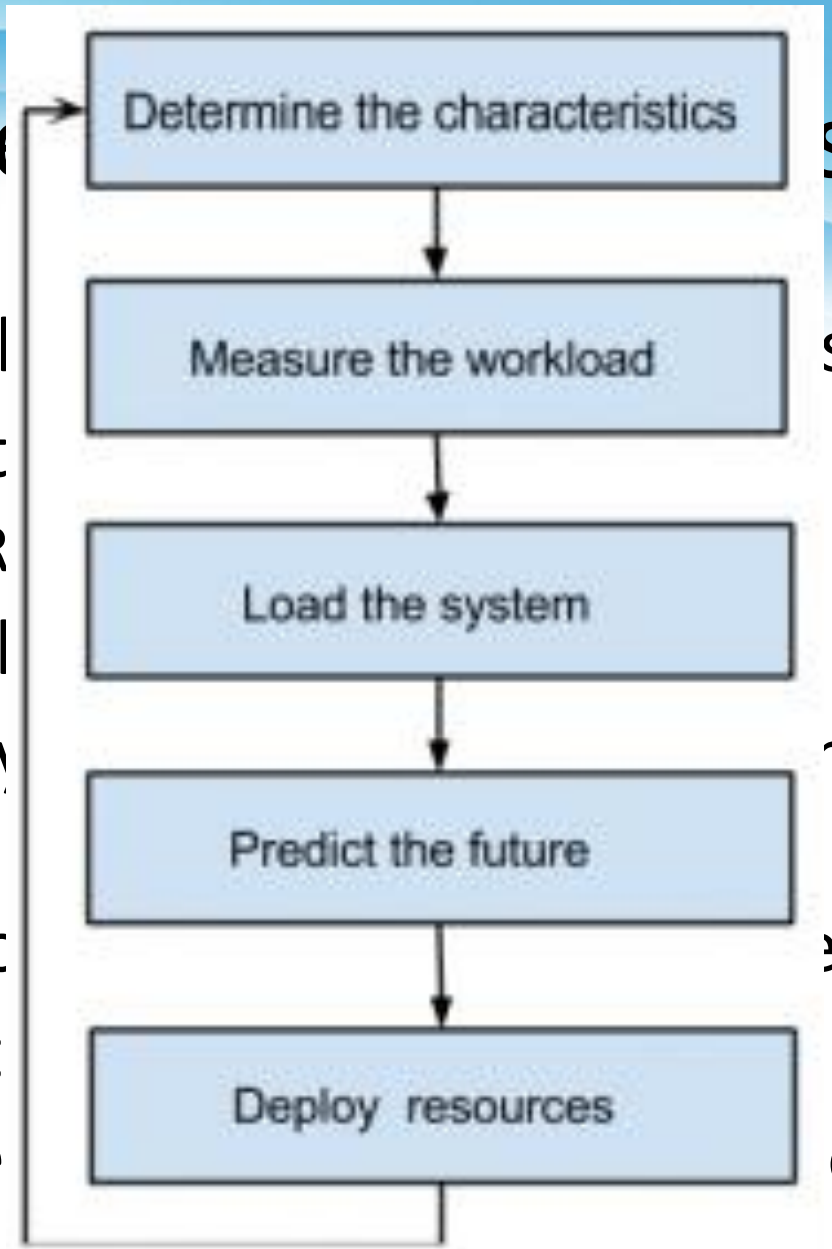
Different between **capacity** and **performance**

- **Capacity** : is about how much work a system can do.
- **Performance** : is the rate of getting work done

- Therefore Capacity planning are not equal to system optimization (performance tuning)

Ste ss

1. what are tl stem?
2. What are t
   CPU, R
3. Measure tl
4. Until the sy new workloads
5. What factc entual failure
6. Predict fut
7. Iterate the one!



how to measure capacity

- First, remember that capacity needs to be measured
- Quantity of the resources available
- Quantity of the request
- The time (start, duration) is important
- Let’s continue with an example (LAMP solution stack)

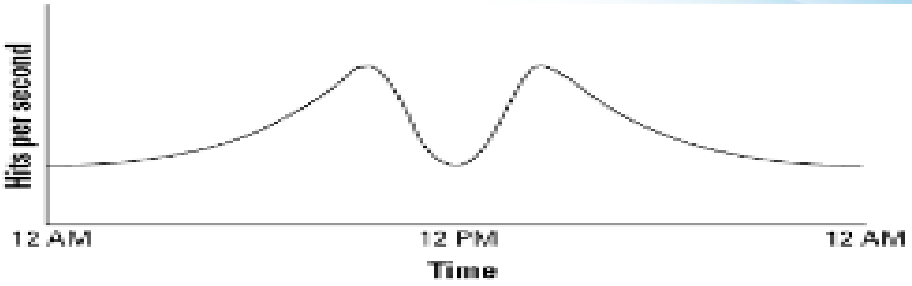
LAMP

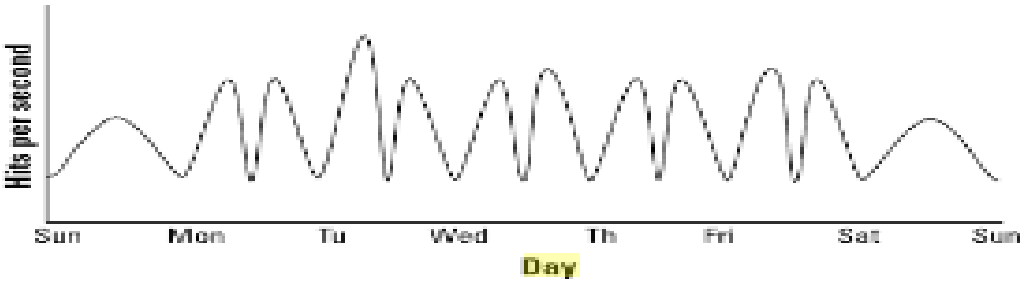
- It is a solution stack
- Many developers use it to create CC applications and web sites
- LAMP stands for :
  - Linux, the OS (RedHat, Ubuntu, SUSE, Debian, …)
    - Could be another OSs (Mac, OpenBSD, Solaries, WAMP)
  - Apache HTTP Server
  - MySQL
  - PHP

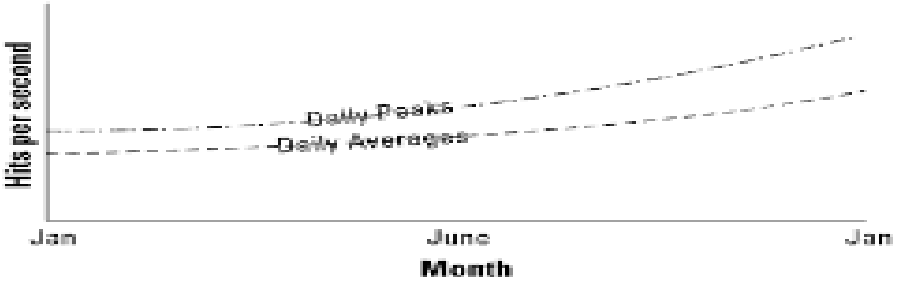
Baseline measurements of the example

- Scenario: A website based on Apache using MySQL to process database. CP has to metrics to calculate workloads:
- Page views / hits (on website per second)
- Transactions (on data server, per queries/S)
- Time (Hour, day, month, year)

Web server workload







Application level statics

- WT : Total workload for the system per unit time (integrate the area under the curve)
- WAVG: the average workload over multiple units of time (sum various WTs divided by number of unit times)
- WMAX: the highest amount of work recorded by the system
- WTOT: the total number of work done by the system ($\sum$WT)

System level statics

- To measure resources (CPU, Memory/RAM, Disk and network connectivity) you need tools , which are provided by the Oss
- Display the CPU's level activity
  - In Linux/Unix → sar command
  - In Windows → Task manager
  - In Mac → Command + Spacebar
- Round Robin Database Tool(RRDTool) : a popular Linux performance measurement tool → shows CPU load, Network/bandswith

Ma

- A cloud computing platform for managing heterogeneous distributed data center infrastructures.
  The **OpenNebula** platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service.

is

Types of job scheduling

- Advance Reservation
- Virtual Advance reservation for queues (VARQ)
- Planning-based
- Future reservation

Advance scheduling

- **Problems**
  - Limited /forced by the job abstraction → the user doesn't have direct access to the resources, only allowed to submit jobs to the system
  - There is no mechanism to directly log in to reserved resources (as it is available in VMs)
  - Lead to utilization problem → need to leave resources before a reservation can begin
- Solutions
  - Preempting running job
  - Checkpointing
  - Job migration

Preemptive scheduling

- It is based primarily on parameters specified at the **queue** level
  - Establish hierarchy of queues
  - Other factors determine which jobs from a queue should be preempted.
- There are three ways to establish which queues should be preempted:
  - **Based on queue priority**—the PREEMPTION parameter defines a queue as preemptive or preemptable and preemption is based on queue priority, where jobs from higher-priority queues can preempt jobs from lower-priority queues
  - **Based on a preferred order**—the PREEMPTION parameter defines queues that can preempt other queues, in a preferred order
  - **Explicitly, by specific queues**—the PREEMPTION parameter defines queues that can be preempted, and by which queues

checkpoint

- Checkpointing is taking a snapshot of the current state of a program in such a way that the program can be restarted from that state at a later time.
- Preempted job's entire state is saved to disk, allowing it to resume its work from the last checkpoint
- Systems that support job checkpointing allow a job to save off its current state and either terminate or continue running.
- A checkpointed job may be restarted at any time and resume execution from its most recent checkpoint.

Job migration

- Allowing checkpointed jobs to restart on other available resources
- Not waiting for the preempting job or reservation has completed
- Could be used before the start of advance reservation
- However, make some restriction → cuz burden the user with having to modify their applications to make them checkpointing

Virtual Advance reservation for queues (VARQ)

- Similar to the advance reservation , plus
  - First predicting the time job would spend waiting in the queue
  - Then submitting a job, based on wait time prediction
  - The chance it will be running at the start of the reservation is maximized

Planning based

- Job requests are immediately planned by making a reservation, instead of waiting in a queue
- Each time a new request is received, the entire schedule is re-evaluated to optimize resource usage
- E.g. a request for an advance reservation can be accepted without using preemption, since the jobs that were originally assigned to those resources can be assigned to different resources

Leasing Model

- As we mentioned earlier, Haizea is a lease manger
- Lease definition: a negotiated agreement between a resource provider and a resource consumer
- Three components become important
  - Hardware resource required
    - By the consumer → CPU, memory ,…
  - Software environment required
  - Availability period
- As we have already discussed the first tow component, the focus is on the last one

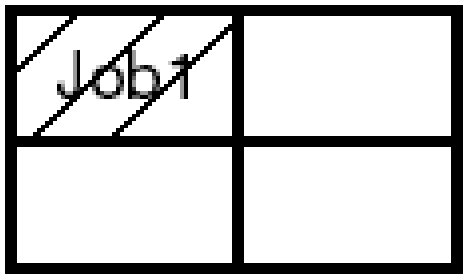
Availability period

- Start time : may be unspecified or specified
  - Unspecified → a best effort lease : where resources are provisioned as soon as possible, and requests are placed on a queue if necessary
  - Specified → Advance reservation lease : where the resources must be available at a specific time
- Maximum duration: total amount of time resource will be available
- Lease can be preemptable : can be paused without disrupting the computation that takes place inside the lease

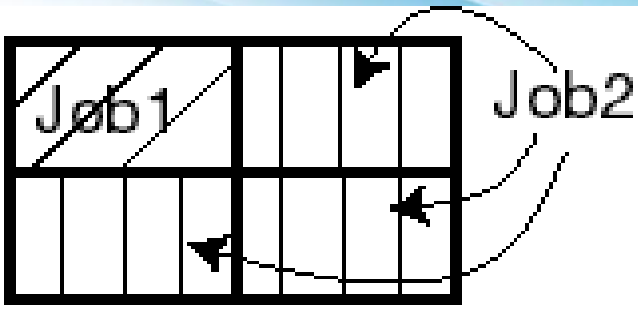
Elements of managing capacity

- W : number if physical nodes capable of running VMs
- MB(megabytes) : memory and local disk storage
- B (MB/per second) : bandwidth
- N: number of VMs
- Tuple (p,m,d,b) [tuple = a data structure consisting of multiple parts. Set of data consisting a record]
  - P : number of CPUs
  - m: memory in MB
  - d: disk space in MB
  - b: network bandwidth in MB/sec
- A disk image I with a size(I) MB → must be transferred form the repository to a node before the VM can start

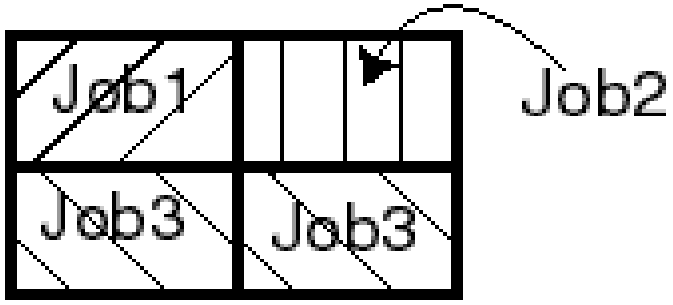
Leasing scheduling



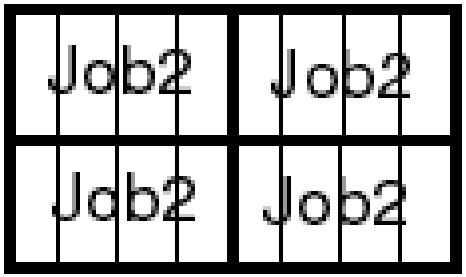
(a) Job1 started at 8:00 am.
Will finish at 10:00 am.



(b) Job2, submitted but can't start since it needs 4 processors. Remaining 3 reserved by Job2.



(c) At 8:30 am Job3 submitted. Job3 backfills Job2.



(d) At 10:00 am, Job2 starts.

References

- Cloud Computing bible, Barrie Sosinsky, Wiley publication, 2011
- Cloud Computing: principles, Rajkumar Buyya, Elsevier , 2013
- The Performance Impact of Advance Reservation Meta-scheduling , Quinn Snell, Mark Clement, David Jackson, and Chad Gregory, Brigham Young University, Provo, Utah 84602
- Preemption Management, access date 30/11/2016, http://docs.adaptivecomputing.com/mwm/archive/6-0/8.4preemption.php#checkpoint