Resource Management Parts I-III

Edi Shmueli, Spring 2015
RM part I: Reminder

- Environment where many users submit many jobs to a central scheduler
  - Scheduler needs to decide which job to execute next (the “scheduling problem”)
- Many ways to schedule the jobs
  - FCFS, Random, Priority, etc.
- We focus on scheduling method called “Proportional share”
  - Common scheduling technique in many environments
- Organizations are hierarchical
  - VO==BU, Projects, Sub-Projects, Users, etc.
- Each entity buys equipment
  - Equipment reaches a shared pool (data center)
- Goal is to enforce the premised shares between the entities

• Data structure called **target tree** representing the organization hierarchy
  - Defines for entity (VO, Project, User) its share of the resources (what it “should get”)
• **Measured (usage) tree** repressing what each entity is currently getting
  - Or got over time (history...)
• Diff between the trees allows to pick the most “needy” leaf
  - Scheduler picks a job from that leaf and repeats the process
RM part I: Challenges

1. Calculating resource use
2. Ensuring fast ramp-up
3. Considering past usage (history)
Challenge #1: Calculating resource usage

• Jobs are submitted by the users (leafs in the tree)
  • Resources are counted bottom-up (accumulated from the leafs towards the root)
• The challenge is what to count? (מה המספר bulundים מייצג?)
  • Jobs currently running by each user?
  • Cores used by the jobs? Memory?
  • Jobs slots?
  • Bottleneck resource (DRF)?
  • Etc.

מה המטריקה לייצוג צריכת משאבים?

Challenges in Modern Data Centers Management, Spring 2015
Challenge #2: Ensuring fast ramp-up

• Algorithm works fine as long as there is constant stream of jobs
  • Each user/project constantly submits new jobs (leafs always have waiting jobs)
• What if a user/project/VO does not have jobs to send?
  • Node “disappears” and its share gets distributed among the other nodes...
• Then the node suddenly becomes active (users return, new jobs submitted)
  • It can wait long time to get its share (jobs from other nodes use the resources...)

Challenges in Modern Data Centers Management, Spring 2015
Challenge #3: Considering history

• **After** we decided on what the numbers represent (challenge #1)
  • Lets assume we count cores...
• Should the # represent **current usage** or also account for the past (history)?
  • Is “8” = **currently used** cores by jobs? (no history)
  • Or, “8” = cores used by jobs over the last 24 hours? (with history)
• **Pros & cons**
  • No history = very easy to explain to the users
  • History = Good for handling peaks
Today’s lecture – RM part II

• After we (proportional share) chose the next job to execute (job scheduling)
• Question now is where to execute the job (resource matching)
  • Tremendous impact on data center utilization ($$....)

1. Job Scheduling
   Select next job(s) to execute

(2) Flat list (queue)
   These are the job(s) to execute

3. Resource Matching
   Select where to execute the job(s)