Announcements

– Don’t forget to fill out the evaluation form

– Assignment 3 is due Monday at 11:59pm
  – If you are submitting the hard copy of the individual problems, bring them to class the next day

– Assignment 4 will be posted this weekend, and it will be due on Monday April 3rd

– Exam 2 is on Friday April 7th

Scheduling

– When do we need scheduling?
  – Provision our resources
  – Multiple "entities" asking to use a set/pool of resources

– Scheduling:
  – Within a single machine
  – Jobs on a cluster of machines
  – Tasks within a job in a cluster of machines

Scheduling within a machine

– Jobs arrive to the system at any time
– They get added to a queue
– They get scheduled according to the policy used within the OS

– Policies that we discussed:
  – FIFO
  – SJF
  – RR
Scheduling in Hadoop

Let’s take a closer look at what needs to be scheduled in Hadoop.

Most important metrics to be maintained:
- Ensure fairness
- Maximize utilization
- Minimize response time
- Maximize throughput

In YARN

- YARN = Yet Another Resource Negotiator
  - It’s the cluster management implemented within Hadoop
- YARN has many schedulers implemented within it
- System admins have the option to change how scheduling is done by,
  - Changing the settings of YARN
  - Implementing their own schedulers

Scheduling in YARN

- We’ll discuss:
  - Hadoop Capacity Scheduler
  - Hadoop Fair Scheduler
  - Dominant Resource Fairness Scheduler
Hadoop capacity scheduler (HCS)
- Multiple queues are used
- Each queue gets some share of the capacity
  - Limits can be set for these shares
- Jobs arriving at the system get assigned to a queue according to their priority
- Within the same queue, FIFO/FCFS is usually used

More on HCS
- Preemption is not allowed
- A queue can get more than its capacity if:
  - You can have a hierarchy of queues
- Question?
  - Does the scheduling depend on what the resources the job needs?

Hadoop Fair Scheduler (HFS)
- Fairness is not guaranteed in HCS
- How can we ensure fairness?
  - What is fairness?

HFS - continued
- Divides the cluster resources into pools for every user
- Typically resources are divided equally over the pools
- Within the pool, any scheduling policy can be used
  - FIFO
  - Fair share scheduling
- What if a pool doesn't have its fair-share of resources?
  - When can that happen?
  - What does it do?
Example

HFS – within a pool

- Is FIFO the fairest of them all? Is it the most optimal?

- What about SJF?

Comparison

Dominant Resource Fairness
What is DRF?

- The above policies achieve fairness for a single resource
  - Example: Memory
- In a typical distributed system, jobs demand more than one resource
  - Example:
    - In Hadoop, jobs need CPU and Memory
    - In the cloud, jobs need CPU, Memory, and Disk
- How can you achieve fairness across multiple resources?

Heterogeneous demands

Example

- A system with total capacity 9 CPUs and 18 GB of Memory
- Two jobs come in:
  - A with the demand vector <1 CPU, 4 GB RAM>
  - B with the demand vector <3 CPU, 1 GB RAM>
- Goal:
  - Maximize allocations x and y for A and B
  - With fairness and capacity constraints considered

Online version of DRF

- In a real system jobs come and go
- How can we compute all of this online?
- Algorithm:
  - When a job comes in, compute its dominant share (ds)
  - Repeat until there is no more resources available:
    - Pick the job with the minimum ds
    - Allow it to run a new task