Lecture 9:

• Borg (cont)

Announcements:

• R-1 due
• HW-1 out
• R-2 out

Cluster Job Scheduling Overview

Borg Review (from last time):

Long Running Services (never shut down) and batch jobs

Jobs consist of one or more tasks – each job’s tasks run the same program

Each job runs in a single cell – one cell per cluster

Jobs and tasks can have resource constraints

Jobs assigned priorities

Jobs can be sequenced (e.g., for applications / services)
Borg Architecture Basics

Borg’s main functions:

• figure out where (machine) in a cell to run tasks
• allocate the resources for tasks
• install programs and dependencies for running tasks
• monitor task health
• restart tasks on failure

Borgmaster: centralized controller

• each cell has a replicated borgmaster (5 replicas)
• handles client requests (e.g., create/lookup job)
• each replica stores state information of the cell (in distributed database)
• cell has a leader borgmaster that handles all operations
• if leader fails, another leader is chosen (elected)
• when a job is submitted, records info in DB and adds to pending queue
**Scheduler**: asynchronously scans pending queue ...

- determines how to assign tasks to machines
- for larger cells, scheduler is run in parallel
- scheduling algorithm has 2 parts: feasibility checking and scoring

**Feasibility Checking**: find machines with enough resources for tasks

- includes lower priority tasks that could be evicted

**Scheduler (cont)**

**Scoring**: ranking of feasible machines

- tries to minimize number and priority of preempted tasks
- considers machines with copies of needed dependencies
- “task packing” quality: mix of high and low priority per machine
- more details in paper (including “best” vs “worst” fit)
Borg Architecture Basics

**Borglets**: local “worker” agent running on each machine
- starts and stops tasks, restarts failed tasks
- manages resources, performs logging
- reports state back to Borgmaster

**More on Borgmaster**: performance scalability / helping communication cost
- polls Borglets every few seconds to retrieve state and send requests
- also delegates communication to Borgmaster replicas (the “link shard”)
- state information is aggregated (as diffs) and sent to leader

Additional Notes:

**Task evictions**
- applications on Borg expected to be designed to handle evictions
- via replication, persistent state stored in distributed file system
- taking (occasional) checkpoints
- Borg automatically reschedules evicted tasks

99.99% availability in practice (according to 2015 paper)
- various techniques used ... for example:
  - spreading tasks around machines, racks, power domains
  - keeping each cell independent (for “correlated operator” errors)
Summary – Things to Know

• The basic idea being the common system “pattern”

• The basic role of each of the three main Borg components (Borgmaster, Scheduler, Borglets)

• The basic interaction between the three main Borg components

• Generally how Borg relates to the common system “pattern” ideas