Borg

Abhishek Verma, Luis Pedrosa, Madhukar R. Korupolu, David Oppenheimer, Eric Tune, John Wilkes

Google Inc.
What is Borg?

- Search
- Gmail
- Map Reduce
- Maps
- Dremel

Borg

Datacenter resources (CPU, memory, disk, GPU, etc.)
Heterogeneous Workload

Priority

Monitoring
Production
Batch
Best effort

\{ Prod
\} Prod
\} Non-prod
Hello World!

```python
job hello_world = {
    runtime = { cell = 'ic' }
    binary = './hello_world_webserver'
    args = { port = '%port%' }
    requirements = {
        ram = 100M
        disk = 100M
        cpu = 0.1
    }
    replicas = 10000
}
```

(Example taken from John Wilkes’s presentation at EuroSys 2015)
Job
Name, Owner, Constraints, Priority
Scheduler

- Schedules from high priority to low priority, with round-robin scheduling within each priority band
- Feasibility checking
  - What machines can I run the task on?
- Scoring
  - Which machine should I run the task on?
Task
1 CPU, 1 GB mem

(All machines have 5 units of each resource)
Job
Name, Owner, Constraints, Priority

Cell

config file
borgcfg
command-line tools
web browsers

scheduler

BorgMaster

persistent store (Paxos)
link shard
read/UI shard

Borglet

Borglet

Borglet

Borglet
Scalability

- Score caching
  - Cache score of machine until task or machine changes
- Equivalence classes
  - Only score one task per equivalence class
- Relaxed randomization
  - Score a random subset of feasible machines
Scalability

- Score caching
  - Cache score of machine until task or machine changes
- Equivalence classes
  - Only score one task per equivalence class
- Relaxed randomization
  - Score a random subset of feasible machines

- Scheduling a cell’s workload down to 300s compared to 3 days
Availability

- Replication
- Admission control
  - Job resource limits are checked against user quota
- Reduction of external dependencies
  - Simple, low-level tools for deploying instances
- Cell independence
- Tasks continue to run even if Borglet and Borgmaster fail
Availability

- Replication
- Admission control
  - Job resource limits are checked against user quota
- Reduction of external dependencies
  - Simple, low-level tools for deploying instances
- Cell independence
- Tasks continue to run even if Borglet and Borgmaster fail

- 99.99% availability in practice
Cell Compaction

![Graph showing the percentage of cells against the compacted size percentage.](image)
Utilization

- Cell sharing
  - Workload segregation would result in ~20-30% increase in cell size
- Resource requests
  - Fixed-size containers/VMs would require ~30-50% more resources in the median case
- Resource reclamation
  - ~90% of cells would need ~40-50% more machines
Lessons Learned

1. Grouping mechanisms are restrictive
2. IP per machine vs IP per container
3. Allocs (or equivalent) are useful
4. Cluster management is more than scheduling
5. The master is the kernel of a distributed system