Nexus

A common substrate for cluster computing

Benjamin Hindman  Andy Konwinski  Matei Zaharia  Ion Stoica
Problem

Rapid innovation in cluster computing frameworks

No single framework optimal for all applications

Want to run multiple frameworks in a single cluster
Solution

Nexus is a resource manager over which frameworks like Hadoop can be written

» Nexus multiplexes resources between frameworks
» Frameworks control job execution
Implications

Users can pick best framework for each app

Specialized frameworks, not one-size-fits-all
What if I only want to use Hadoop?

Nexus is a better way to manage Hadoop

Hadoop master is *complex*, hard to scale and make robust

Nexus is simple, only handles fair sharing: easier to scale and harden

Multiple Hadoop instances/versions at same time
Outline

Beyond MapReduce and Dryad

Nexus Architecture

Results

Philosophy
Beyond MapReduce & Dryad
1. Iterative Jobs

Many machine learning jobs are of the form

» parameter $p = \text{random value}$

» while not converged: $p = f(p, \text{dataset})$

Each iteration can be expressed as MapReduce, but requires reloading data set each time
2. Nested Parallelism

Recursion (quicksort), maps within maps

Difficult in MapReduce/Dryad, possible with NESL model
3. Irregular Parallelism

Sometimes, we don’t know computation graph
   » Branch-and-bound search
   » Exploring moves in chess
   » Ray tracing

Difficult to hack into MapReduce/Dryad, easy
with work-stealing programming model (Cilk)
4. Existing Parallel Apps

Parallel build (distcc)
Parallel unit test (Selenium Grid)
Web servers (!)
Nexus Architecture
Hadoop

App

Hadoop master

Hadoop slave

Hadoop slave

Hadoop slave
Nexus

App

Hadoop master

Nexus master

Nexus slave

Hadoop executor

task

Nexus slave

Hadoop executor

task

Nexus slave

Hadoop executor

task
Nexus

App 1

Hadoop master

Nexus master

Nexus slave

Hadoop executor

task

App 2

Dryad master

Nexus slave

Dryad executor

task

Nexus slave

Hadoop executor

task

Dryad executor

task
Scheduler API

- Framework scheduler
- Nexus master

Actions:
- slot_offer(slot)
- accept_offer(task)
- reject_offer()
- status(task, status)
Executor API

start_task(task)

kill_task(task)

status(task, status)

Nexus slave

Framework executor
Results
Implementation Stats

**Simple:** 2000 lines of C++

**Scalable:** 500 slaves on EC2

**Fast:** schedules 800 tasks/s versus 100 tasks/s for Hadoop
Scalability: # of Nodes

![Graph showing Scalability: # of Nodes]
Scalability: File Size

![Graph showing the relationship between file size and number of tasks]
LR Job Comparison

<table>
<thead>
<tr>
<th>Number of Iterations</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Hadoop: 92, Nexus: 38</td>
</tr>
<tr>
<td>10</td>
<td>Hadoop: 510, Nexus: 130</td>
</tr>
<tr>
<td>20</td>
<td>Hadoop: 1000, Nexus: 227</td>
</tr>
</tbody>
</table>
Philosophy
Microkernel
   » Make reliable component as simple as possible

Exokernel
   » Give maximal control to frameworks above you

IP model
   » Narrow waist over which diverse frameworks can run
Questions