Fuxi: a Fault Tolerant Resource Management and Job Scheduling System at Internet Scale

Zhuo Zhang*, Chao Li*, Yangyu Tao*, Renyu Yang†*, Hong Tang*, Jie Xu§
Alibaba Cloud Computing Inc. * Beihang University † University of Leeds §
Agenda

• Motivation
• System Introduction
• Contribution
  – Incremental resource management protocol
  – User transparent failure recovery
  – Faulty node detection and multilevel blacklist mechanism
• Evaluation
• Conclusion and Future Work
Motivation

• **Big data era**
  – 2.5 exabytes of data are generated everyday
  – The speed of data generation doubles every 40 months
  – Billions of transactions on Taobao everyday, must be processed in 6 hours

• **Scalability challenges**
  – $10^3 \sim 10^4$ of machines
  – $10^3 \sim 10^4$ of concurrent jobs
  – $10^5 \sim 10^6$ of concurrent workers
  – $10^4 \sim 10^5$ OPS for resource request/assign
  – Other factors: Multi-dimensional resource, quota, priority, locality, etc.

• **Fault tolerance challenges**
  – Failures become normal at large scale
  – Partial failures worse than machine down
  – Master failures
Apsara: A Brief History

09/10/2009
Aliyun.com Inc established

08/27/2010
Apsara became the platform of four applications:
Search, Mail, Image Storage, AliFinance

07/28/2011
Aliyun.com went online, releasing 1st cloud service: ECS

08/15/2013
5000-node Aspara cluster went into production

10/24/2013
3rd Aliyun dev conference held in Hangzhou. ~5000 developers attended the conference
Apsara Cloud Platform

- Resource scheduling and allocation
- Quota control
- Preemption
- Process life cycle control
- Process isolation

DAG Job
Long running service
...

• Map, Mail, Search

• Cloud Market

• Other Cloud Services

ECS/SLB
OSS
OTS
RDS
ODPS

Resource Scheduling and Allocation

• Quota control
• Preemption
• Process life cycle control
• Process isolation

Linux Clusters

Application Scheduling

Resource Management

Distributed Coordination

Distributed File System

Remote Procedure Call

ACE

ECS/SLB
OSS
OTS
RDS
ODPS

Cloud Market

Other Cloud Services
Fuxi Workflow

- **FuxiMaster**
- **Cluster Node**
  - FuxiAgent
  - App Master
  - Worker
- **Client**

**Processes**
- Job scheduling
- Resource request and response
- Node management and status collection
- Job submission
- Process Execution Plan
Support Different Application Models

• AppMaster lib
  – Resource request/return
  – Process execution
  – Heartbeat

• Build-in models
  – DAG job (similar to Dryad and Tez)
  – Long running services
    • Mail service
    • Search service

• Can support more models
  – spark, storm, etc.
Contributions

• Scalability
  – Incremental resource management protocol

• Fault tolerance
  – User transparent failure recovery
  – Faulty node detection and multilevel blacklist mechanism
Incremental Resource Management Protocol

FuxiMaster

FuxiAgent1

FuxiAgent2

FuxiAgent3

AppMaster1

ScheduleUnit: {1cpu, 2gb}

Apply: {M1*2, C*10}

Max: 10

Assign: M1: +3, M2: +3, M3: +2

Return: M3: -1

AppMaster2

ScheduleUnit: {2cpu, 5gb}

Apply: {M1*2, C*10}

Max: 10

Assign: M1: +2, M2: -1, M3: +2

Return: M2: -2
Resource Scheduling

- Event driven real-time resource allocation
- Locality tree based scheduling
- Multi-layer schedule order

New Submission
Resource Release
Resource Preemption
Resource Change
Quota Change

Available Resource Pool
Waiting Resource Requests Queue
Fulfilled Resource Request Queue

Cluster
- App1: P1,14
- APP2: P1,9
- APP3: P2,5
- APP4: P3,16

Rack1
- App1: P1,9
- App3: P2,1
- App4: P3,8

Rack2
- App1: P1,4
- App2: P1,4
- App3: P2,3
- App4: P3,8

M1
- App1: P1,4
- App2: P1,3
- App3: P2,6

M2
- App1: P1,4

M3
- App1: P1,3
- App3: P2,6

M4
- App3: P2,1
- App4: P3,3
User Transparent Failure Recovery

• Full stack failover support
  – FuxiMaster
  – JobMaster
  – FuxiAgent

• Scalability consideration
  – Hard state
  – Soft state
FuxiMaster Failover

- **Hard state**
  - Application configuration
- **Soft state**
  - Machine list
  - Resource request
  - Resource schedule results
Faulty Node Detection and Multilevel Blacklist

- Cluster level: rule based
  - Hardware & OS: Disk statistics, machine load, network I/O, etc.
  - Apsara platform: OPS, state, heartbeat, etc.
  - Scoring system
  - Plugin based

- Job level
Faulty Node Detection and Multilevel Blacklist

• Cluster level
• Job level: history based
  – One instance failed on one machine
  – Multiple instances failed on one same machine
  – Running slow on one machine
  – Read slow from one machine
Fuxi in Production

- In production since 2009
- Manage hundreds of thousands of nodes
- 5000 nodes in one cluster
- Job type at a glance

<table>
<thead>
<tr>
<th></th>
<th>avg</th>
<th>max</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Number</td>
<td>228/task</td>
<td>99,937/task</td>
<td>42,266,899</td>
</tr>
<tr>
<td>Worker Number</td>
<td>89.92/task</td>
<td>4,636/task</td>
<td>16,295,167</td>
</tr>
<tr>
<td>Task Number</td>
<td>2.0/job</td>
<td>150/job</td>
<td>185,444</td>
</tr>
</tbody>
</table>
Evaluation: 5000 nodes with 1000 concurrent jobs

FuxiMaster scheduling time

Planned memory usage
<table>
<thead>
<tr>
<th>Provenance</th>
<th>Configuration</th>
<th>GraySort Indi Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuxi (2013)</td>
<td>5000 nodes (2 2.20GHz 6cores Xeon E5-2430, 96 GB memory, 12x2TB disks)</td>
<td>100TB in 2538 seconds (2.364TB/min)</td>
</tr>
<tr>
<td>Yahoo! Inc. (2012)</td>
<td>2100 nodes (2 2.3Ghz hexcore Xeon E5-2630, 64 GB memory, 12x3TB disks)</td>
<td>102.5 TB in 4,328 seconds (1.42TB/min)</td>
</tr>
<tr>
<td>UCSD (2011)</td>
<td>52 nodes (2 Quadcore processors, 24 GB memory, 16x500GB disks) Cisco Nexus 5096 switch</td>
<td>100 TB in 6,395 seconds (0.938TB/min)</td>
</tr>
<tr>
<td>UCSD&amp;VUT (2010)</td>
<td>47 nodes (2 Quadcore processors, 24 GB memory, 16x500GB disks) Cisco Nexus 5020 switch</td>
<td>100 TB in 10,318 seconds (0.582TB/min)</td>
</tr>
<tr>
<td>KIT (2009)</td>
<td>195 nodes x (2 Quadcore processors, 16 GB memory, 4x250GB disks) 288-port InniBand 4xDDR switch</td>
<td>100 TB in 10,628 seconds (0.564TB/min)</td>
</tr>
</tbody>
</table>
Evaluation: Performance of Fault Handling

- 5% failure: 15.7% performance degradation
- 10% failure: 19.6% performance degradation
- 0 job fail

<table>
<thead>
<tr>
<th>Injected Fault Type</th>
<th>Node Number</th>
<th>Node Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node Down</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Partial Worker Failure</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Slow Machine</td>
<td>11</td>
<td>23</td>
</tr>
<tr>
<td>Total Number (ratio)</td>
<td>15 (5% failure)</td>
<td>30 (10% failure)</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

• Conclusion
  – Scalability
    • Incremental communication/schedule instead of full run
  – Fault tolerance
    • Full stack failover support
    • Do checkpoint only for hard state
    • Detect faulty node by rules and history

• Future work
  – Improve real resource utilization
  – More resource isolation
  – Interactive jobs
Thanks!
Recently Alibaba’s Chinese online commerce sites drove over US $5.7 billion dollars worth of sales in a single day. Learn what it takes to run these sites and how the Alibaba Group has impacted the Chinese economy. This event is specifically geared towards researchers, engineers and IT professionals who are interested in learning about the growth of ecommerce and technology.

Join us for this FREE event as we share learning’s and surge ahead in 2014!

Date: September 4th, 2014

Location: Alibaba Xixi Campus

Agenda:
- Tour of Xixi Campus
- Presentations from Alibaba
- Q&A Session

Notes:
- Please register at the Alibaba booth in the Exhibition area.
- Due to limited space, we may not be able to accommodate everyone. Sorry!
- This event will be primarily conducted in Mandarin.