Forget about the Clouds, Shoot for the MOON

Wu FENG  |  feng@cs.vt.edu
Dept. of Computer Science
Dept. of Electrical & Computer Engineering
Virginia Bioinformatics Institute

© September 2012, W. Feng
Motivation

• Data Deluge
  – New scientific instruments generate data rapidly
  – High-performance simulations generate a flood of data
  – Internet data sharing allows data caching and replication

• Need for Rapid Scientific Discovery

• Solution: Ubiquity of Parallel Computing

Images: Courtesy of http://images.google.com/
Traditional Parallel Computing Resources

- Government-Funded Supercomputers
  - Not easily accessible to majority of scientists
  - Long queuing time

- Institutional Clusters
  - Expensive to acquire
    - Japan K Computer: $1250M
    - DOE/Cray Jaguar: $104M
    - Microsoft Datacenter: ????
  - Expensive to own
    - Facilities: O($10M - $100M)
    - Operations: Power and cooling
    - Personnel: Experienced system administrators
The Cost of Parallel Computing

- Electrical power costs $\$$\$\$$.

The Cost of Parallel Computing

Examples: Power, Cooling, and Infrastructure $$$

• Japanese K Computer
  – Power & Cooling: 9.89 MW → $10M/year
Cloud Computing Taxonomy

- Public Clouds
  - Windows Azure
  - Amazon Web Services
- Private Dedicated Clouds
  - Cloud Era
  - Rackspace
- Private Opportunistic Clouds

Example: Our MOON Project
Solution: Cloud Computing

Public Clouds

Private Dedicated Clouds

Private Opportunistic Clouds

Example: Our MOON Project
Public Clouds

• Computing as Utility

• Commercial Clouds
  – Software as a Service
    • Gmail
  – Platform as a Service
    • Google AppEngine, Microsoft Azure
  – Infrastructure as a Service
    • Amazon EC2

• Academic Cloud
  – DOE Magellan
Cloud Computing Taxonomy

Public Clouds

Private Dedicated Clouds

Example: Our MOON Project

Private Opportunistic Clouds
Private Dedicated Clouds

• Pros
  – Currently Built on Dedicated Resources
    • Eucalyptus
    • Virtual Computing Lab
  – Better Security & Privacy
    • Behind the firewall
    • Owners have complete control of infrastructure
    • No data transfer to/from public networks

• Cons
  – Inflexible for handle load variance
  – Not that different from datacenter
    • $$$ for infrastructure, power, and cooling
Alternative Resources for Private Clouds?

• “Free” Computing Resources within Institutions: Idle Personal Computers
  – E.g. Math Emporium at VT: 550 dual-core Intel Mac
  • Collective compute power equivalent to a modest supercomputer
Challenges

- **Resource Volatility**
  - Example opportunistic environment (Entropia @ SDSC)
  - Average unavailability 0.4 and as high as 0.9
Cloud Computing Taxonomy

- Public Clouds
  - Windows Azure
  - Google Apps
  - Amazon Web Services
- Private Dedicated Clouds
  - Cloud Era
  - Rackspace
  - UNIVA
- Private Opportunistic Clouds

Example: Our MOON Project
Private Opportunistic Clouds

• Private Cloud Computing on Opportunistic Resources

• Our Approach
  – MOON: MapReduce On Opportunistic eNvironments
    • Platform as a Service
      – Reliable and efficient MapReduce service

• Minimize performance impact to desktop users
  … while
    delivering compute cycles to cloud end users
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Public Clouds</th>
<th>Private Dedicated Clouds</th>
<th>Private Opportunistic Clouds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Efficiency</td>
<td>🎈</td>
<td>😞</td>
<td>😊</td>
</tr>
<tr>
<td>Security &amp; Privacy</td>
<td>😞</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Accessibility</td>
<td>😞</td>
<td>😊</td>
<td>😊</td>
</tr>
<tr>
<td>Performance</td>
<td>🎈</td>
<td>😊</td>
<td>😞 → 🎈</td>
</tr>
</tbody>
</table>
Roadmap

- Introduction
- MOON: MapReduce On Opportunistic eNvironments
  - What is MapReduce?
  - What is an Opportunistic Environment?
  - Overview of MOON
  - Data Management
  - Task Scheduling

- Results
- Conclusion
What is MapReduce?

• **Ease of Use**
  – Primitives from Lisp: Map and Reduce
  – Automatic parallel execution, fault-tolerance by runtime

• **Efficient for Large-Scale Data Processing**
  – Deliver computation to data

• **Example: Word Count**

```
Hello World
Hadoop
Hello World
```

```
<Hello, 1>
<World, 1>
<Hadoop,1>
<Hello, 1>
<World, 1>
```

```
<Hello, 2>
<World, 2>
<Hadoop,1>
```
Many Applications to Bio

• Computational Biology
  – Sequence alignment
  – Short-read sequence mapping

• Data Mining
  – Temporal data mining
  – K-means clustering
  – Genetic Algorithms

Cognitive Neuroscience

Bioterrorism

Genomics

Images: Courtesy of http://images.google.com/

Virginia Tech
Invent the Future

Bio-IT World Cloud Summit
09/12/2012

SyNergy synergy.cs.vt.edu
Hadoop

- Open-Source MapReduce Implementation
  - Widely used: Yahoo!, Facebook, Amazon and many others
- Master-Slave Architecture
  - Coupled with Hadoop Distributed File System (HDFS)
What is an Opportunistic Environment?

• Resources come and go without notice
  – E.g., Condor yield for 15 minutes after keyboard/mouse events

• Examples: BOINC and Condor

• Limitations
  – Limited programming models
    • Embarrassingly parallel
    • Master-worker programming model
  – Inefficient support for data-intensive applications
Our Solution: MOON

• Combining the expressiveness of MapReduce with the latent computing capability of idle compute resources, i.e., opportunistic environments

• MapReduce + Opportunistic Environments

or

• MapReduce On Opportunistic Environments
MOON Overview

- Observation
  - Opportunistic resources not dependable enough to provide reliable service
MOON Overview (Cont.)

• Hybrid Resource Provisioning
  – Supplement volatile PCs with a small # of dedicated computers

• Extend Hadoop Task Scheduling & Data Management
Roadmap

• Introduction

• MOON: MapReduce On Opportunistic eNvironments
  – What is MapReduce?
  – What is an Opportunistic Environment?
  – Overview of MOON
  – Data Management
  – Task Scheduling

• Results

• Conclusion
MapReduce Data Model

- **Data Dependencies**
  - A Map task depends on its corresponding input data
  - A Reduce task depends on intermediate data of ALL map tasks
Hadoop Data Management

- **Design Summary**
  - Uniform replication of input/output data
  - No replication for intermediate data

- **Limitations on Opportunistic Environments**
  - Prohibitively high replication cost for reliable data service
    - E.g., 11 replicas to achieve 99.99% availability on resources with 0.4 unavailability rate: $1 - 0.4^{11} = 0.99996$
  - Frequent Map task re-execution caused by loss of intermediate data
    - Too many re-execution could cause *job failure*
MOON Data Management Enhancement

• Reduce Replication Cost with Hybrid Replication
  – Two dimensional replication factor \(<d, v>\)
  – E.g., 1 dedicated and 3 volatile copies to achieve 99.99% availability
    (0.001 unavailability rate on dedicated node)
    • \(1 - 0.001 \times 0.4^3 = 0.99994\)

• Design Challenges
  – \# dedicated nodes << \# volatile nodes
  – Dedicated nodes can be overloaded with incautious I/O
Cost-Efficient Replication

• Reserve Dedicated Resources for Important Data

• Differentiate Data in the File System
  – Reliable Files: Cannot afford loss
    • System data, input data
  – Opportunistic Files: Can be regenerated
    • Intermediate data – rerun map tasks
    • Output data – rerun reduce tasks

• Avoid Overloading Dedicated Nodes by Prioritizing I/O
  – Write access: Opportunistic files yield to reliable files on dedicated nodes
  – Read access: Data supplied by the volatile nodes first
Hadoop Task Scheduling

• Speculative Task Dispatching for *Stragglers*
  – Task progress score proportional to processed data
  – Straggler: progress score 20% slower than average
  – Uniform replication: each task replicated at most once

• Issue: Design Assumption Broken
  – Original assumption: Tasks run smoothly till completion
  – Opportunistic environment: Frequent task suspension/resume

• Result: Misidentification of Stragglers
Hadoop Task Suspension Handling

- **Heartbeat Mechanism**
  - Mark a TaskTracker dead when no heartbeat in expiring interval
  - All tasks on a dead node killed and rescheduled

- **Inflexible**
  - If expiring interval too long, speculative copy too slow
  - If expiring interval too short, tasks killed prematurely
MOON Task Suspension Handling

• Introduce *hibernated* state for TaskTracker
  – Give replication priority to *frozen* tasks, i.e., all copies on hibernated nodes
  – Configure hibernating interval much shorter than expiring interval

• Advantages
  – Fast response to task suspension
  – Prevent killing tasks prematurely
Leverage Dedicated Resources

• Assign Tasks to Dedicated Nodes when Possible

• Advantages
  – Save replication cost
    • Tasks with dedicated copy do not participate homestretch phase
  – Improve efficiency of long-running tasks
    • No suspension/interruption
    • Guarantee completion
Roadmap

• Introduction
• MOON: MapReduce On Opportunistic eNvironments
  – What is MapReduce?
  – What is an Opportunistic Environment?
  – Overview of MOON
  – Data Management
  – Task Scheduling

• Results
• Conclusion
Experiment Setup

• Methodology
  – Emulate opportunistic environments on clusters with configuration similar to student labs
  – Control degree of volatility with randomly generated machine unavailability traces

• Platform
  – System X at Virginia Tech
  – Dual 2.3GHz PowerPC 970FX processors
    • 4GB of RAM
    • Gigabit Ethernet
Overall Performance

- Extended Hadoop with intermediate data replication
- MOON hybrid setting: 20:1, 15:1, 10:1

MOON outperforms extended Hadoop by 3x
Acknowledgements

• Seed funding was provided in part by the Virginia Tech Foundation (VTF).

• We actively seek additional collaborations, partnerships, funding, and customers to extend and harden MOON.
Conclusion

• Ubiquity of parallel computing and the importance of high-end computing for scientific discovery

• MOON provides cost-efficient parallel computing solutions on private clouds
  – High-quality MapReduce services
  – Reliable data storage

• *Forget about the clouds, shoot for the MOON!*