Evaluating Cloud Computing for Science

Lavanya Ramakrishnan

Lawrence Berkeley National Lab

June 2011
Magellan Research Questions

• Are the *open source* cloud software stacks ready for DOE HPC science?
• Can DOE cyber security requirements be met within a cloud?
• How usable are cloud environments for scientific applications?
• Are the new cloud programming models useful for scientific computing?
• Can DOE HPC applications run efficiently in the cloud? What applications are suitable for clouds?
• When is it cost effective to run DOE HPC science in a cloud?
• What are the ramifications for data intensive computing?
Cloud Deployment Models

**Physical Resource Layer**

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

**Linkedin**

**Twitter**

**Google**

**Hadoop**

**Windows Azure**

**Amazon Web Services**

**Eucalyptus Systems**

**CRD**

**NERSC**
Magellan User Survey

User interfaces/Science Gateways: Use of clouds to host science gateways and/or access to cloud resources through science

- Access to additional resources
- Exclusive access to the computing resources/ability to schedule independently of other groups/users
- Ability to control software environments specific to my application
- Ability to control groups/users
- Ability to share setup of software or experiments with collaborators
- Easier to acquire/operate than a local cluster
- Cost associativity? (i.e., I can get 10 cpus for 1 hr now or 2 cpus for 5 hrs at the same cost)
- MapReduce Programming Model/Hadoop
- Hadoop File System

Program Office

<table>
<thead>
<tr>
<th>Program Office</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Scientific Computing Research</td>
<td>17%</td>
</tr>
<tr>
<td>Biological and Environmental Research</td>
<td>9%</td>
</tr>
<tr>
<td>Basic Energy Sciences -Chemical Sciences</td>
<td>10%</td>
</tr>
<tr>
<td>Fusion Energy Sciences</td>
<td>10%</td>
</tr>
<tr>
<td>High Energy Physics</td>
<td>20%</td>
</tr>
<tr>
<td>Nuclear Physics</td>
<td>13%</td>
</tr>
<tr>
<td>Advanced Networking Initiative (ANI) Project</td>
<td>3%</td>
</tr>
<tr>
<td>Other</td>
<td>14%</td>
</tr>
</tbody>
</table>
Magellan Software

- Batch Queues
- Private Clusters
- Virtual Machines (Eucalyptus)
- Hadoop
- Science and Storage Gateways
- ANI

Public or Remote Cloud
Are the *open source* cloud software stacks ready for DOE HPC science?

Can DOE cyber security requirements be met within a cloud?
Amazon Web Services

- Web-service API to IaaS offering
- Non-persistent local disk in VM
- Simple Storage Service (S3)
  - scalable persistent object store
- Elastic Block Storage (EBS)
  - persistent, block level storage
- Offers different instance types
  - standard, micro, high-memory, high-cpu, cluster computer, cluster GPU
Private Cloud Software

• Eucalyptus
  – open source IaaS implementation, API compatible with AWS
  – KVM and Xen can be used as hypervisors
  – Walrus & Block Storage
    • interface compatible to S3 & EBS
  – experience with 1.6.2 and 2.0

• Other options
  – OpenStack, Nimbus, etc
Experiences with Eucalyptus (1.6.2)

• **Scalability**
  – VM network traffic is routed through a single node
  – limit on concurrent VMs due to messaging size

• **Requires tuning and tweaking**
  – co-exist with services such as DHCP
  – advanced Nehalem CPU instructions

• **Allocation and Accounting**
  – hard to ensure fairness since first come first serve

• **Limited Logging and Monitoring**
Security in the Cloud

• Trust issues
  – User provided images uploaded and shared
  – Root privileges by untrained users opens the door for mistakes

• Effective Intrusion Detection System (IDS) strategy challenging
  – Due to the ephemeral nature of virtual machine instances

• Fundamental threats are the same, security controls are different
Can DOE HPC applications run efficiently in the cloud? What applications are suitable for clouds?
Workloads

• High performance computing codes
  – supported by NERSC and other supercomputing centers

• Mid-range computing workloads
  – that are serviced by LBL/IT Services, other local cluster environments

• Interactive data intensive processing
  – usually run on scientist’s desktops
Experiment Setup

• Workloads
  – HPCC
  – Subset of NERSC-6 application benchmarks for EC2 with smaller input sizes
    • represent the requirements of the NERSC workload
    • rigorous process for selection of codes
    • workload and algorithm/science-area coverage

• Platforms
  – Amazon
  – Lawrencium (IT cluster)
  – Magellan
Application Benchmarks

![Bar chart showing runtime relative to Magellan (non-VM) for various applications.]

- **GAMESS**
- **GTC**
- **IMPACT**
- **fvCAM**
- **MAESTRO256**

**Legend:**
- Franklin
- Lawrencium
- EC2-Beta-Opt
- Amazon EC2
- Amazon CC
Application Benchmarks

![Graph showing runtime relative to Magellan for MILC and PARATEC with different cloud services: Franklin, Lawrencium, Amazon CC, and Amazon EC2.]
Application Scaling

PARATEC

- TCPoIB
- TCPoEth
- AmazonCC

MILC

- TCPoIB
- TCPoEth
- AmazonCC

Performance relative to native (IB)

Number of cores
Comparison of PingPong BW

Percentage Bandwidth relative to Native (IB)

Number of cores

AvgPingPongBW_TCPoIB
AvgPingPongBW_TCPoEth
AvgPing_Amazon

17
Amazon Reliability: A Snapshot

![Graph showing the number of instances failed at startup and successful over time of day.]

- Failed at startup
- Successful

<table>
<thead>
<tr>
<th>Time of day</th>
<th>Number of instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tue Mar 15 14:14:54</td>
<td>Failed at startup</td>
</tr>
<tr>
<td>Tue Mar 15 14:55:54</td>
<td>Successful</td>
</tr>
<tr>
<td>Tue Mar 15 15:03:39</td>
<td>Failed at startup</td>
</tr>
<tr>
<td>Wed Mar 16 17:59:05</td>
<td>Successful</td>
</tr>
</tbody>
</table>

CRD computational research division

U.S. Department of Energy Office of Science

NERSC
Queue Wait Time Vs VM Startup Overhead

Batch queue prediction times from QBETS service on NSF TeraGrid resources, 2010

BReW: Blackbox Resource Selection for eScience Workflows collaboration w/ Emad Soroush, Yogesh Simmhan, Deb Agarwal, Catharine van Ingen

Windows Azure VM startup time, 2010
Yogesh Simmhan, Microsoft
What codes work well?

• Minimal synchronization, modest I/O requirements
• Large messages or very little communication
• Low core counts (non-uniform execution and limited scaling)
• Generally applications that would do well on midrange clusters
  – Future: Analyzing data from our batch queue profiling (through IPM)
How usable are cloud environments for scientific applications?
Application Case Studies

• Magellan has a broad set of users
  – various domains and projects (MG-RAST, JGI, STAR, LIGO, ATLAS, Energy+)
  – various workflow styles (serial, parallel) and requirements

• Two use cases discussed today
  – MG-RAST - Deep Soil sequencing
  – STAR – Streamed real-time data analysis
MG-RAST: Deep Soil Analysis

Background: Genome sequencing of two soil samples pulled from two plots at the Rothamsted Research Center in the UK.

Goal: Understand impact of long-term plant influence (rhizosphere) on microbial community composition and function.

Used: 150 nodes for one week to perform one run (1/30 of work planned) and used NERSC for fault tolerance and recovery.

Observations: MG-RAST application is well suited to clouds. User was already familiar with the Cloud.
Early Science - STAR

Details

- STAR performed Real-time analysis of data coming from Brookhaven Nat. Lab
- First time data was analyzed in real-time to a high degree
- Leveraged existing OS image from NERSC system
- Started out with 20 VMs at NERSC and expanded to ANL.

Image Courtesy:
Jan Balewski, STAR collaboration
Application Design and Development

• Image creation and management
  – system administration skills
  – determining what goes on image etc

• Workflow and data management
  – need to manage job distribution and data storage, archiving explicitly

• Performance and reliability needs to be considered
Are the new cloud programming models useful for scientific computing?
What are the ramifications for data intensive computing?
Hadoop Stack

- Open source reliable, scalable distributed computing
  - implementation of MapReduce
  - Hadoop Distributed File System (HDFS)

Source: Hadoop: The Definitive Guide
Hadoop for Science

• Advantages of Hadoop
  – transparent data replication, data locality aware scheduling
  – fault tolerance capabilities

• Hadoop Streaming
  – allows users to plug any binary as maps and reduces
  – input comes on standard input
Application Examples

• Bioinformatics applications (i.e., BLAST)
  – parallel search of input sequences
  – Managing input data format

• Tropical storm detection
  – binary file formats can’t be handled in streaming

• Atmospheric River Detection
  – maps are differentiated on file and parameterC

U.S. DEPARTMENT of ENERGY Office of Science CRD computational research division NERSC
HDFS vs GPFS (Time)

Teragen (1TB)

- HDFS
- GPFS
- Linear (HDFS)
- Expon. (HDFS)
- Linear (GPFS)
- Expon. (GPFS)

Time (minutes) vs Number of maps
Challenges

• **Deployment**
  - all jobs run as user “hadoop” affecting file permissions
  - less control on how many nodes are used - affects allocation policies

• **Programming: No turn-key solution**
  - using existing code bases, managing input formats and data

• **Additional benchmarking, tuning needed, Plug-ins for Science**
Comparison of MapReduce Implementations

Collaboration w/ Zacharia Fadika, Elif Dede, Madhusudhan Govindaraju, SUNY Binghamton

Producing random floating point numbers

Load balancing

Processing 5 million 33 x 33 matrices

Cluster size (cores)
MARIANE

Collaboration w/ Zacharia Fadika, Elif Dede, Madhusudhan Govindaraju, SUNY Binghamton
Data Intensive Science

• Goal: Evaluating hardware and software choices for supporting next generation data problems

• Evaluation of Hadoop
  – using mix of synthetic benchmarks and scientific applications
  – understanding application characteristics that can leverage the model
    • data operations: filter, merge, reorganization
    • compute-data ratio

(collaboration w/ Shane Canon, Nick Wright, Zacharia Fadika)
Tools for managing code ensembles/UQ

• Code ensembles
  – problem that is decomposed into a large number of loosely coupled tasks
    • Running VASP on 125K crystals in the Materials Genome database
    • Uncertainty Quantification

• Evaluate Hadoop for managing CEs
  – compare with database (MySQL, MongoDB), workflow tools, Message Queues

  (collaboration w/ Dan Gunter, Elif Dede)
Unique Needs and Features of a Science Cloud

• Access to parallel filesystems and low-latency high bandwidth interconnect
  – access to legacy data sets

• Bare metal provisioning for applications that require custom environments
  – that cannot tolerate the performance hit from virtualization

• Preinstalled, pre-tuned application software stacks
  – specific libraries and performance considerations

• Customizations for site-specific policies
  – authentication, fairness

• Alternate MapReduce implementations
  – account for scientific data and analysis methods
Conclusions

• Current day cloud computing solutions have gaps for science
  – performance, reliability, stability
  – programming models are difficult for legacy apps
  – security mechanisms and policies

• HPC centers can adopt some of the technologies and mechanisms
  – support for data-intensive workloads
  – allow custom software environments
  – provide different levels of service
Acknowledgements

• US Department of Energy DE-AC02-05CH11232
• Magellan
  – Shane Canon, Tina Declerck, Iwona Sakrejda, Scott Campbell, Brent Draney
• Amazon Benchmarking
  – Krishna Muriki, Nick Wright, John Shalf, Keith Jackson, Harvey Wasserman, Shreyas Cholia
• Magellan/ANL
  – Susan Coghlan, Piotr T Zbiegiel, Narayan Desai, Rick Bradshaw, Anping Liu, Ed Holohan
• NERSC
  – Jeff Broughton, Kathy Yelick
• Applications
  – Jared Wilkening, Gabe West, Doug Olson, Jan Balewski, STAR collaboration, K. John Wu, Alex Sim, Prabhat, Suren Byna, Victor Markowitz
Questions?

Lavanya Ramakrishnan
LRamakrishnan@lbl.gov