MG-RAST: A Case Study in Large Scale Metagenomics Analysis

Narayan Desai
desai@mcs.anl.gov
Mathematics and Computer Science Division
Argonne National Laboratory
January 15, 2012
Talk Overview

- Definition of terms
- MG-RAST overview
- Our cloud adoption process
- Strategies for cloud adoption
Terms

- Clouds are computational utilities
  - Provide metered access to processors, storage, networking
  - Access to all resources via APIs
    - Disintermediated access
  - Enables specialization and competition
- Three basic models
  - IaaS
    - Rental access to hardware
  - PaaS
    - Framework for performing a particular set of tasks
  - SaaS
    - Remotely deployed and managed software
Range of Cloud Options

General

Complicated

IaaS

PaaS

SaaS

Special Purpose

Easy to Use
Metagenomics ...

Definition:
“random shotgun DNA sequencing applied directly to environmental samples”

- whole shotgun metagenomics
- Result is a combination of short reads of DNA from all organisms in sampled community
  - Mixed together

Who are they?

What are they doing?
Metagenomics and Discovery

Today:
Mapping our knowledge to help understand microbial ecology
→ using existing knowledge

Future:
Discover new biology from computationally mining the unknowns
→ Patterns
→ co-occurrence
→ Exclusion
→ ..

Via:
Mapping to curated databases

Example
Systematic discovery of patterns
e.g. CRSPR by Jill Banfield
This data was calculated for metagenomes 4441679.3, 4441680.3, 4441681.3 and 4441682.3. The data was compared to MSNR using a maximum e-value of None and a minimum identity of None%. The data has been normalized to values between 0 and 1. If you would like to view raw values, redraw using the form below.

<table>
<thead>
<tr>
<th>metagenome</th>
<th>source</th>
<th>domain</th>
<th>phylum</th>
<th>class</th>
<th>abundance</th>
<th>avg eValue</th>
<th>%Ident</th>
<th>#proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>4441681.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Proteobacteria</td>
<td>Gammaproteobacteria</td>
<td>332902</td>
<td>-2.92</td>
<td>83.23</td>
<td>12148</td>
</tr>
<tr>
<td>4441682.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Firmicutes</td>
<td>Clostridia</td>
<td>50026</td>
<td>-3.76</td>
<td>77.94</td>
<td>24605</td>
</tr>
<tr>
<td>4441679.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Firmicutes</td>
<td>Clostridia</td>
<td>46641</td>
<td>-3.67</td>
<td>78.77</td>
<td>24069</td>
</tr>
<tr>
<td>4441680.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Firmicutes</td>
<td>Clostridia</td>
<td>31808</td>
<td>-3.87</td>
<td>76.34</td>
<td>17701</td>
</tr>
<tr>
<td>4441681.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Firmicutes</td>
<td>Clostridia</td>
<td>26796</td>
<td>-3.55</td>
<td>78.18</td>
<td>15415</td>
</tr>
<tr>
<td>4441680.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Bacteroidetes</td>
<td>Bacteroidia</td>
<td>21190</td>
<td>-5.18</td>
<td>81.07</td>
<td>11495</td>
</tr>
<tr>
<td>4441679.3</td>
<td>MSNR</td>
<td>Bacteria</td>
<td>Bacteroidetes</td>
<td>Bacteroidia</td>
<td>20000</td>
<td>-4.74</td>
<td>80.47</td>
<td>10594</td>
</tr>
</tbody>
</table>
**Brief history of MG-RAST**

- **December 2007 (v1)**
  - 100+ groups and ~250 data submitters
  - 100+ data sets, ~10+ GBp total size

- **October 2009 (v2)**
  - Pre-publication sharing available
  - ~1500 data submitters, ~300 public data sets
  - 6000+ data sets
  - 200+ GBp total data sets
  - About ~30 GBp/month throughput

- **March 2011 (release v3)**
  - 2500+ data submitters
  - ~2000 public data sets
  - 25,000 data sets total
  - Throughput:
    - 47GBp in 24h
    - 3000 submissions in 24h

- **January 2012 (v3.1.2)**
  - 36,139 data sets
  - 9.31 TBp analyzed
  - 500 users / day
  - 7500+ users total
The Most Overused Slide in Genomics

Cost per Megabase of DNA Sequence

Moore’s Law

National Human Genome Research Institute

genome.gov/sequencingcosts
Observations (circa 2009)

- We’re in trouble
  - Data set size is outpacing our throughput
  - The pipeline is getting further and further behind

- Gap between sequencing costs and computation costs is growing
  - We’re in a hole, and it is getting deeper

- Computational capacity is a serious concern
  - Computing is growing its share of the budget
  - But the scaling curves suggest we won’t be able to keep up solely with hardware additions

- Any sustainable solution will require algorithm changes
  - What is the most relevant/useful analysis we can afford to compute?

- Public portals have a complicated resource consumption dynamic
  - Users abstracted from resource costs
MG-RAST v2 Computational Architecture

=> Purpose built dedicated system

- Standard bioinformatics pipeline design
- Dedicated compute resources
  - ~64 nodes
- Tightly integrated system/runtime environment
  - Global filesystems
  - Single resource manager (SGE)
- Shared with another project
  - Configuration/change difficulties
  - Capacity problems

shares NFS filesystem with all systems

Local Cluster
FPGA
MG-RAST Cloud Adoption Strategy

Goals: Scale the MG-RAST pipeline to work across distributed and shared resources
Get the most work done with the resources we can access

- Pursue shared computing resources
  - Dedicated resources are a luxury we can’t afford
- Perform detailed performance analysis
  - Where are our bottlenecks?
- Redesign analysis workflow
  - Negotiate tradeoffs between performance and sensitivity
  - Continuous refinement
- Use resources efficiently
  - Nebulous concept; varies between resource types
  - Minimize computational (and financial) costs
- Move the most expensive computational stages to the cheapest resources
  - Sequence similarity search
Software requirements

- Portability of analysis was the key problem to solve
  - Analysis orchestration
  - Data mobility
  - Tool execution portability

- Replaced system resource manager with a distributed workload management system
  - Argonne/Another Workflow Engine (AWE)
  - Data portability layer (Shock)

- AWE provides encapsulation for analysis work flows
  - Analysis environment and package prerequisites sent to compute elements on the fly
  - Minimal requirements on host environment
  - Can be run under system resource manager

- Enabled execution of the sims calculation across a range on shared systems
  - HPC clusters
  - Cloud resources
Our new architecture

- Centralized web portal
  - Database server
  - Archival dataset access
- Centralized orchestration of computation
  - Workload management
  - Storage services
- Small dedicated cluster
  - For data intensive operations only
  - Moving these tasks towards the data archival infrastructure
- Dynamic cloud resources (IaaS)
  - Large scale similarity searches
  - Clustering
  - QC
- Focusing on driving more systems into the cloud
  - Let our sysadmin focus on running the software, not the hardware
  - Need to address performance issues for specialized systems on the way
How did this work?

- Enabled MG-RAST to use any resources we could get access to
  - Regardless of geographic location (ANL or NERSC)
  - For any available duration (workunits run for 10-30 minutes each)
  - With any kind of system environment
- This expansion helped, but not a panacea
  - Access to more capacity with growth helps, but we can’t expect an ever increasing slice of the pie
- Overall impact was about a 10X improvement in throughput
  - More or less a one trick pony, aside from Moore’s Law improvements
- Initially targeting a single work type (similarity search) proved to be a long term liability
  - Should have pushed more of our system into the cloud initially
- Infrastructure flexibility enabled easy experimentation with new analysis approaches
  - Biggest overall payoff for our group
Lessons Learned

- Access to large shared resources wins
  - Better than dedicated access to small resources
- Cloud APIs and sandboxing enable rapid improvement of computing architecture
- Clouds simplify experimentation
  - Lets you focus on the hard thinking
    - In our case, analysis/algorithm development
  - Isn’t this why we are all here?
- These factors have caused us to start moving most of our development activities towards clouds
  - Moving emphasis away from dedicated resources, even for ad hoc development
Three Architectures for Cloud Adoption

- Pile of instances
  - Everybody builds their own
  - Maximum flexibility, maximum complexity
  - Who wants to be a sysadmin!

- Curated set of images
  - Small set of image maintainers
  - Large set of image instantiators
  - Limits administration/maintenance costs

- Persistent, managed group infrastructure
  - Sysadmin runs resources inside the cloud
  - More flexible and dynamic than dedicated hardware
  - Less flexible than previous two
  - Most similar to the status quo
Cloud Adoption Architectures (cont)

- No right answer
- Many tradeoffs between approaches
  - Flexibility versus Cost
- Some require particular resources
  - System administration effort
  - Technical expertise
- Options vary with group size
  - DIY is popular for small groups, but gets costly in large groups
Contributors

- Folker Meyer
- Andreas Wilke
- Elizabeth Marland
- Jared Wilkening
- Travis Harrison
- Mark D’Souza
- Tobias Paczian
- Kevin Keegan
- Will Trimble
- Hunter Matthews
- Jason Hedden
Questions?