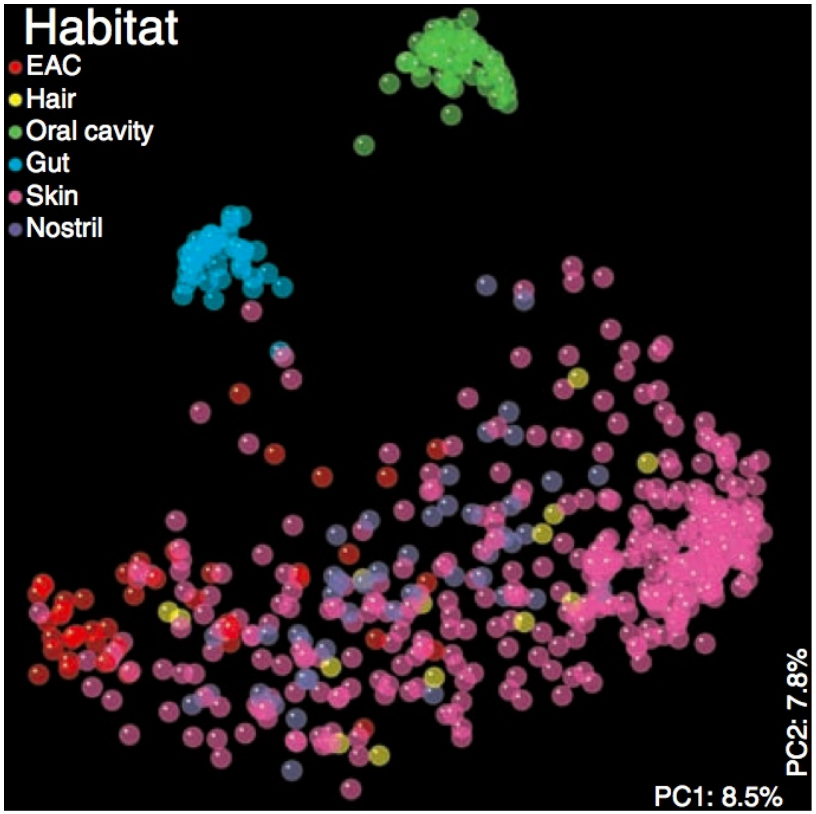
Dynamically Extending Your Cluster with Infrastructure Clouds

A Bioinformatics Use Case

Paul Marshall
University of Colorado at Boulder

QIIME

- Quantitative Insights Into Microbial Ecology
  - How can we manipulate the microbiota to improve health?



Need to understand spatial and temporal differences in healthy and disease states.

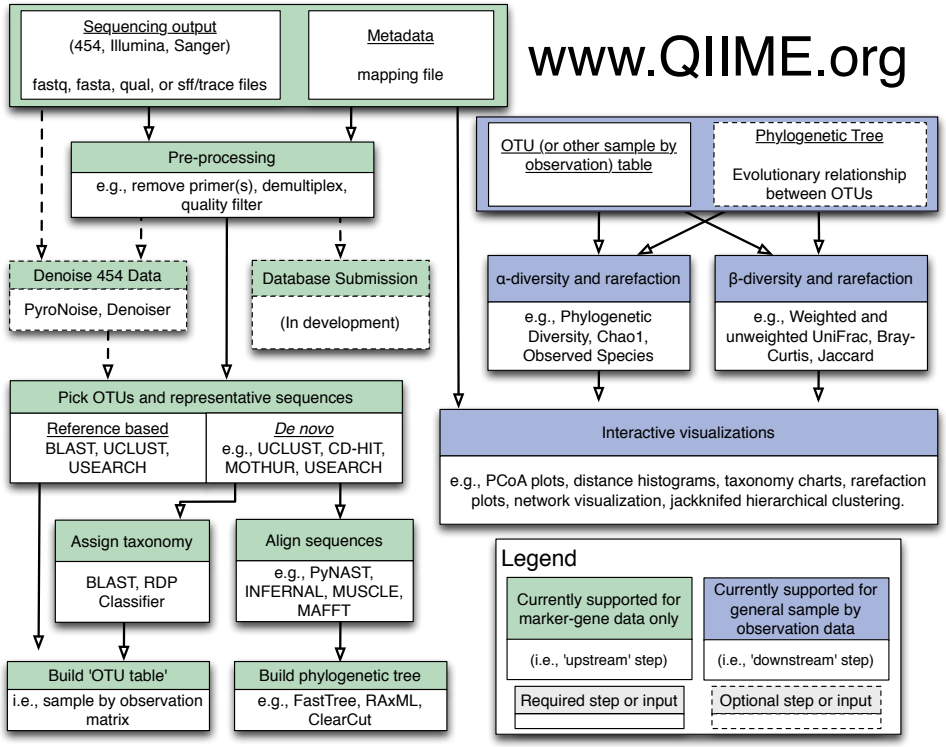
Bacterial Community Variation in Human Body Habitats Across Space and Time, Costello et al., Science 2009

Slide acknowledgement: Antonio Gonzalez Pena

More specifically…

- QIIME provides a complete analysis pipeline
  - Processing raw sequence data ➡ generating publication-quality graphics

QIIME workflow:



www.QIIME.org

Slide acknowledgement: Knight Lab, CU-Boulder

Deploying QIIME on New Resources

- 30+ dependencies: C, Java, Python, Haskell, x86 binaries, …

Virtual machines are ideal for complex software stacks.

Python 2.7.1

PyCogent 1.5.1

Numpy 1.5.1

biom-format 0.9.3

uclust 1.2.22q

PyNAST 1.1

greengenes core set data file

greengenes alignment lanemask file

fasttree 2.1.3

jre1.6.0_05

rdp_classifier-2.2

usearch v5.2.32

blast-2.2.22

cd-hit 3.1.1

ChimeraSlayer

mothur 1.25.0

clearcut v1.0.9

raxml 7.3.0

infernal 1.0.2

cdbtools

muscle 3.8.31

rtax 0.981

pplacer 1.1

ParsInsert 1.04

sfffile and sffinfo

GNU Science Library

AmpliconNoise 1.25

ghc 6.8

MatPlotLib 1.1.0

cytoscape v2.7.0

R 2.12.0

Greengenes 97% OTUs, taxonomies, and tree

Sphinx 1.0.4

At the same time…

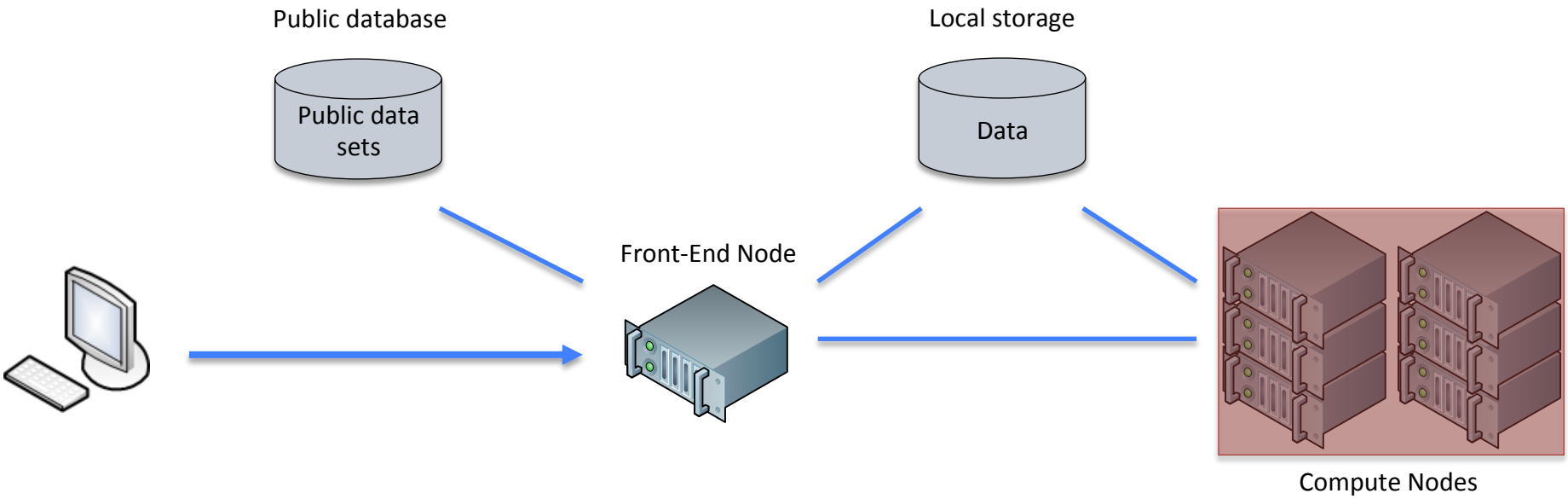
- Data is exploding

First 40
sequences of
91,037,412

```
-bash-3.2$ head -n 40 splitLibs/allSeqs.fna
>mck296Post.1Bubo.8.452375_0 HWI-ST753_83:1:1101:3301:1990#0/1 orig_bc=TAGACTGTACTC new_bc=TAGACTGTACTC bc_diffs=0
TACAGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCGCGTAGGTGGTTCGTTAAGTTGGATGTGAAATCCCCGGGCTCAAC
>mck438Post.4Buma.11.452834_1 HWI-ST753_83:1:1101:3818:1992#0/1 orig_bc=GGTGCGTGTATG new_bc=GGTGCGTGTATG bc_diffs=0
TACGGAGGGGGCTAGCGTTGTTCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGATCAGAAAGTCAGAGGTGAAATCCCAGGGCTCAAC
>mck872Treat.8Bubo.3.452008_2 HWI-ST753_83:1:1101:3795:1994#0/1 orig_bc=ATAGCTCCATAC new_bc=ATAGCTCCATAC bc_diffs=0
CGGTATAAAGTGGGCCATGTTGCTGTAGCGGTCCACTACCACATTAGAAACCCTGGTAGTCCGGCTGACTGACTATAGCTCCATACATCT
>mck446Post.4Rapi.8.451771_3 HWI-ST753_83:1:1101:3885:1998#0/1 orig_bc=CTTGACTGAGGT new_bc=CTTGACTGAGGT bc_diffs=0
TACAGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCGCGTAGGCGGTTGTATAAGTTGGAGGTGAAATCCCCGGGCTCAAC
>mck763Post.10Buma.1.451864_4 HWI-ST753_83:1:1101:6000:2000#0/1 orig_bc=TAGTACCCGAGG new_bc=TAGTACCCGAGG bc_diffs=0
TACGGAGGGTGCAAGCGTTATCCGGATTTATTGGGTTTAAAGGGTCCGTAGGCGGATCTGTAAGTCAGTGGTGAAATCTCACAGCTTAAC
>mck748Post.10Bubo.5.452903_5 HWI-ST753_83:1:1101:6205:1996#0/1 orig_bc=AGAGTAGCTAAG new_bc=AGAGTAGCTAAG bc_diffs=0
GATCTCGTATGCCGTCTTCTGCTTGATTAAAAACCCGGGTAGTCCGGCTGACTGACTAGAGTAGCTAAGATCTCGTATGCCGTCTTCTGC
>mck86Pre.2Bubo.10.452793_6 HWI-ST753_83:1:1101:7044:1994#0/1 orig_bc=AGCTGTTGTTTG new_bc=AGCTGTTGTTTG bc_diffs=0
TCCGGCTGACTGACTAGCTGTTGTTTGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAACACGAGTAGTCCGGCTGACTGACTAGC
>mck378Post.3Buma.7.451590_7 HWI-ST753_83:1:1101:8252:1991#0/1 orig_bc=GTCAACGCGATG new_bc=GTCAACGCGATG bc_diffs=0
TACGGAGGGTGCAAGCGTTATCCGGATTTATTGGGTTTAAAGGGTCCGTAGGCGGATTTATAAGTCAGTGGTGAAATCCGATAGCTTAAC
>mck254Pre.4Osse.9.452071_8 HWI-ST753_83:1:1101:9163:1991#0/1 orig_bc=CCATAATCCGTA new_bc=CCATAATCCGTA bc_diffs=0
TCTCGTATGCCGTCTGATTAGCAACCCCAGTAGTCCGGCTGACTGACTCCATAATCCGTAATCTCGTATGCCGTCTTCTGCT
>mck185Pre.4Bubo.3.451052_9 HWI-ST753_83:1:1101:10588:1993#0/1 orig_bc=TAAAGTCACCCT new_bc=TAAAGTCACCCT bc_diffs=0
TACGTAGGGTGCGAGCGTTAATCGGAATTACTGGGCGTAAAGCGTGCGCAGGCGGTTTGGCAAGTCAGATGTGAAATCCCCGAGCTCAAC
>mck72Pre.2Rapi.8.452208_10 HWI-ST753_83:1:1101:11258:2000#0/1 orig_bc=CGATCCGTATTA new_bc=CGATCCGTATTA bc_diffs=0
ACTCACCGTCATTCACAACTTCAAACGGGATTAGATACCCTAGTAGTCCGGCTGACTGACTCGATCCGTATTAATCTCGTATGCCG
>mck297Post.1Bubo.9.451289_11 HWI-ST753_83:1:1101:11649:1997#0/1 orig_bc=CGCATGAGGATC new_bc=CGCATGAGGATC bc_diffs=0
TACGTAGGGTGCGAGCGTTAATCGGAATTACTGGGCGTAAAGCGTGCGCAGGCGGTGATGTAAGACAGATGTGAAATCCCCGGGCTCAAC
>mck511Post.5Bubo.8.451238_12 HWI-ST753_83:1:1101:12927:1996#0/1 orig_bc=GAGTCAACCGCA new_bc=GAGTCAACCGCA bc_diffs=0
TACAGAGGGTGCAAGCGTTAATCGGAATTACTGGGCGTAAAGCGCGCGTAGGTGGTTCGTTAAGTTGGATGTGAAATCCCCGGGCTCAAC
>mck735Post.9Osse.6.451725_13 HWI-ST753_83:1:1101:14891:1994#0/1 orig_bc=CGCCGTACATAA new_bc=CGCCGTACATAA bc_diffs=0
TACGGAGGGGGCTAGCGTTGTTCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGACCAGAAAGTTGGAGGTGAAATCCCAGGGCTCAAC
>mck85Pre.2Bubo.9.451105_14 HWI-ST753_83:1:1101:15445:1994#0/1 orig_bc=GTACATACCGGT new_bc=GTACATACCGGT bc_diffs=0
TACAGAGGGTGCGAGCGTTAATCGGATTTACTGGGCGTAAAGCGTGCGTAGGCGGCTTTTTAAGTCGGATGTGAAATCCCTGAGCTTAAC
>mck5Pre.1Raca.5.452683_15 HWI-ST753_83:1:1101:16315:1995#0/1 orig_bc=TCTAGCGTAGTG new_bc=TCTAGCGTAGTG bc_diffs=0
TCCGGCTGACTGACTTCTAGCGTAGATTAGATACCCTGGTAGTCCGGCTGACTGACTTCTAGCGTAGTGATCTCGTATGCCGTCTTCTGC
>mck473Post.5Raca.3.451376_16 HWI-ST753_83:1:1101:16288:1998#0/1 orig_bc=ACCCATGACGGA new_bc=ACCCATGACGGA bc_diffs=0
TACGAAGGGGGCTAGCGTTGCTCGGAATCACTGGGCGTAAAGGGCGCGTAGGCGGACTCTTAAGTCGGGGGTGAAAGCCCAGGGCTCAAC
>mck771Post.10Buma.23.451780_17 HWI-ST753_83:1:1101:19838:1997#0/1 orig_bc=TGATAGTGAGGA new_bc=TGATAGTGAGGA bc_diffs=0
TACGTAGGGTGCGAGCGTTAATCGGAATTACTGGGCGTAAAGCGTGCGCAGGCGGTTATGTAAGACAGATGTGAAATCCCCGG
>mck549Post.6Rapi.5.452313_18 HWI-ST753_83:1:1101:20505:1996#0/1 orig_bc=AGGATCCGCATG new_bc=AGGATCCGCATG bc_diffs=0
TCCGGCTGACTGACTAGGCTCCGCATGATCTCGTATGCCGTCTTCTGCTTGAAAAAAAAAACTAGTAGTCAGGCTGACTGACTAG
>mck293Post.1Bubo.5.453112_19 HWI-ST753_83:1:1101:1915:2017#0/1 orig_bc=TTCAGCGCCCTT new_bc=TTCAGCGCCCTT bc_diffs=0
TACGAAGGGGGCTAGCGTTGTCCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGGTATTTAAGTCAGGGGTGAAATCCCGGAGCTCAAC
```

Slide acknowledgement: Antonio Gonzalez Pena

Local Cluster Deployment



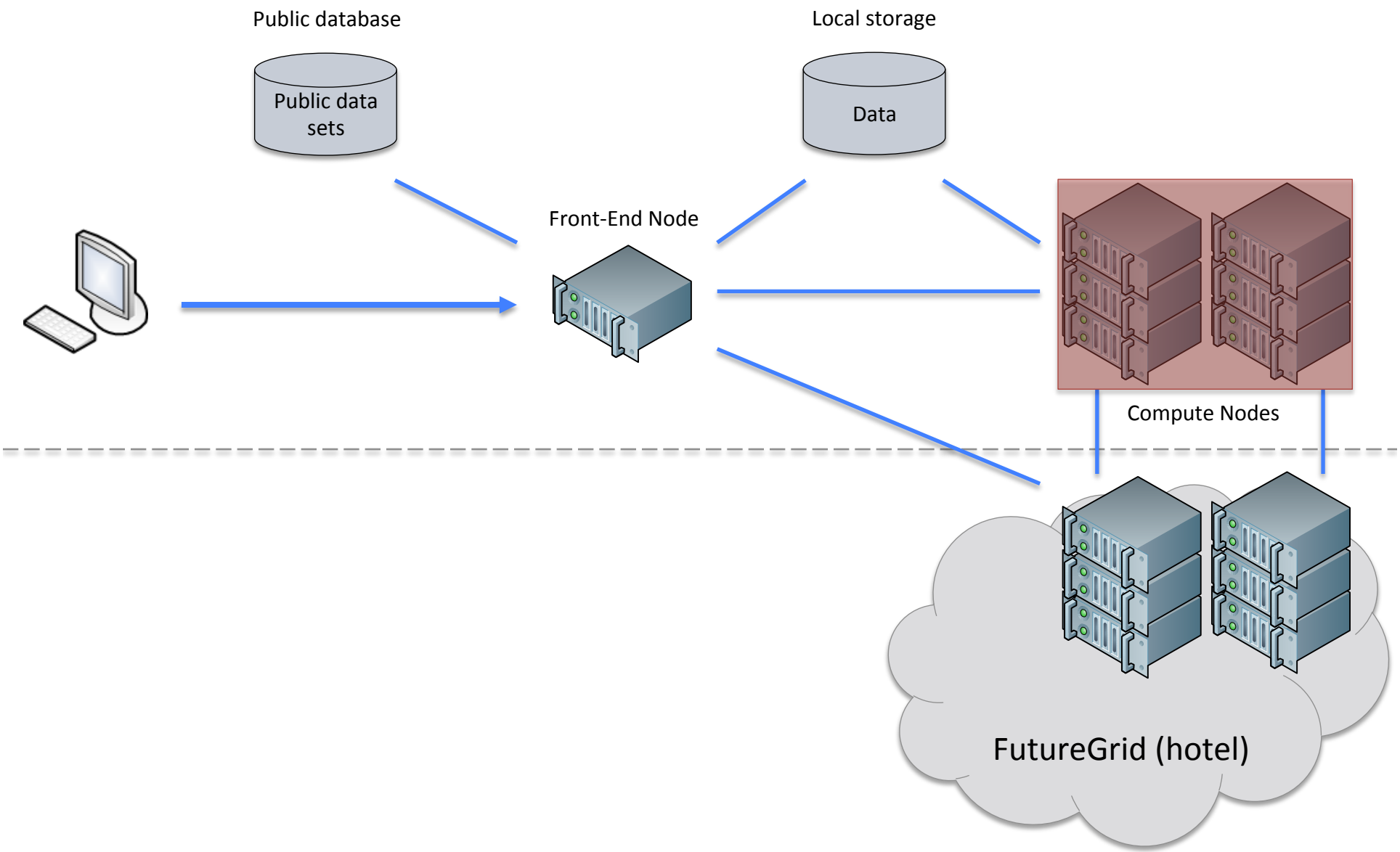
- We want to augment the cluster when it's heavily utilized.

Extending Clusters with Clouds

- Launch standalone environments in the cloud for users.
  - StarCluster (EC2 only, OGE, NFS), et al.
  - Manually transfer data and setup environment.

- Seamlessly integrate IaaS resources with local cluster.

An Elastic Cluster



Implementation

Challenges

- Adapt to demand
- Recontextualize nodes as they join and leave
- Monitor the queue and cluster status
- Provision resources
- Data movement

Implementation

Challenges

- Adapt to demand
- Recontextualize nodes as they join and leave
- Monitor the queue and cluster status
- Provision resources
- Data movement

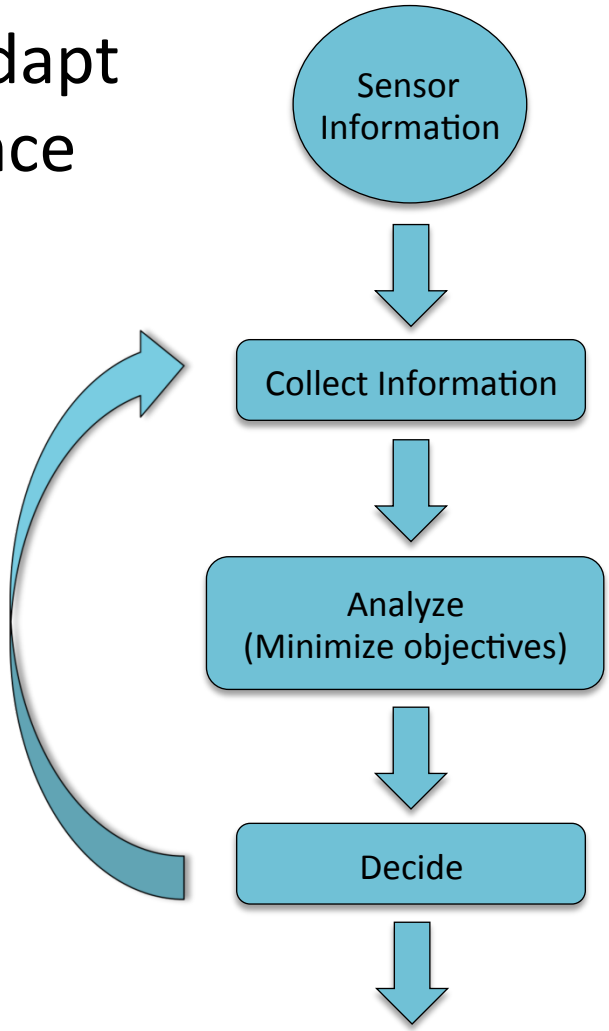
Solutions

- Policies
- Recontextualization Broker
- Sensor to monitor the status of the cluster
- Amazon Auto Scaling
- Gluster FS

Provisioning Policies

- Elastic environments need to adapt efficiently to demand and balance conflicting needs of users and administrators.

- Requirements
  - Reduce time jobs are queued
  - Minimize costs



Published in IPDPS HPGC 2012: Provisioning Policies for Elastic Computing Environments

Implementation

Challenges

- Adapt to demand
- Recontextualize nodes as they join and leave
- Monitor the queue and cluster status
- Provision resources
- Data movement

Solutions

- Policies
- Recontextualization Broker
- Sensor to monitor the status of the cluster
- Amazon Auto Scaling
- Gluster FS

Recontextualization

- All nodes in the environment need to be aware of each other and trust each other

- REST-based recontextualization broker
  - Exchange host information (e.g. IP addresses, SSH host keys) between nodes in a deployment
  - All communication over HTTPS
  - Symmetric keys used for both user and context security

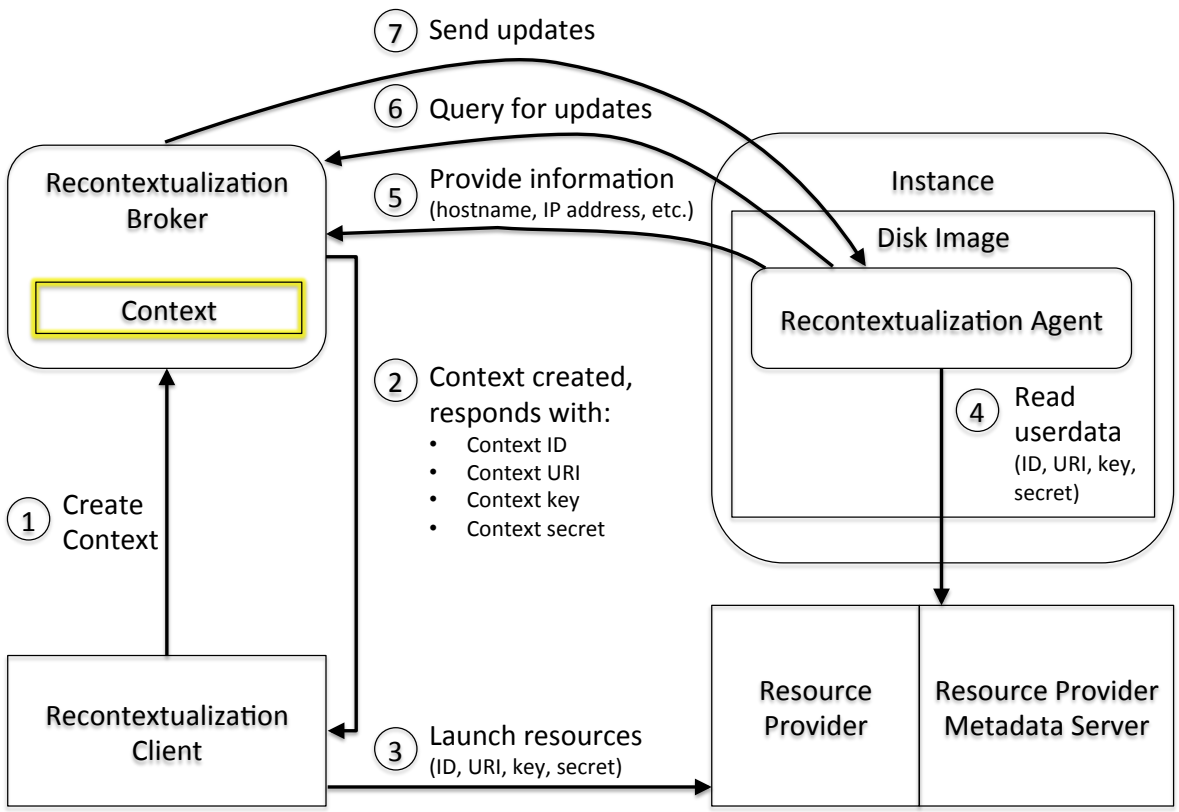
Recontextualization

Actions

1. Create context
2. Broker responds with context information
3. Deploy VMs
4. VMs read context information from metadata server
5. VM provides its information to broker

Loop periodically:

6. Query for updates to the context
7. Broker sends updates



Implementation

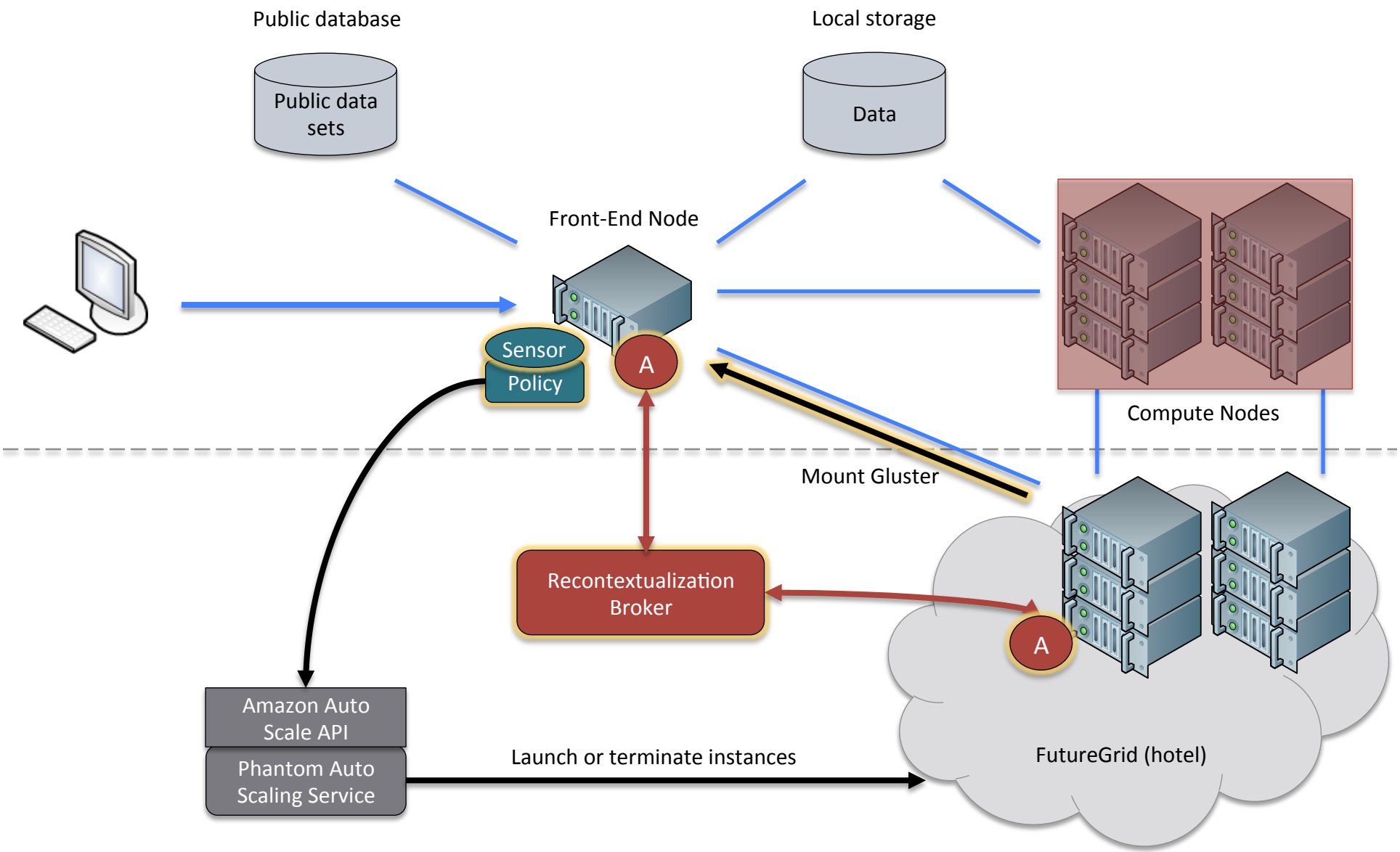
Challenges

- Adapt to demand
- Recontextualize nodes as they join and leave
- Monitor the queue and cluster status
- Provision resources
- Data movement

Solutions

- Policies
- Recontextualization Broker
- Sensor to monitor the status of the cluster
- Amazon Auto Scaling
- Gluster FS

An Elastic Cluster with Auto Scaling



Running QIIME on Your Laptop

- A few example steps: check mapping file, assign reads to samples, and denoise (parallel with –n flag):

```
$check_id_map.py -m Fasting_Map.txt -o mapping_output
$split_libraries.py -o run1 -f run1.fasta -q run1.qual –m mapping.txt -w 50 -r -l 150 -L 350
$denoise_wrapper.py -v -i run1.sff.txt -f run1/seqs.fna -o run1/denoised/ -m mapping.txt –n 4
```

QIIME will simply use available cores when running in parallel on your laptop.

Running QIIME on an Elastic Cluster

- On cluster front-end node: Just change the number of processes you want to run (-n):

```
$check_id_map.py -m Fasting_Map.txt -o mapping_output
$split_libraries.py -o run1 -f run1.fasta -q run1.qual –m mapping.txt -w 50 -r -l 150 -L 350
$denoise_wrapper.py -v -i run1.sff.txt -f run1/seqs.fna -o run1/denoised/ -m mapping.txt –n 64
```

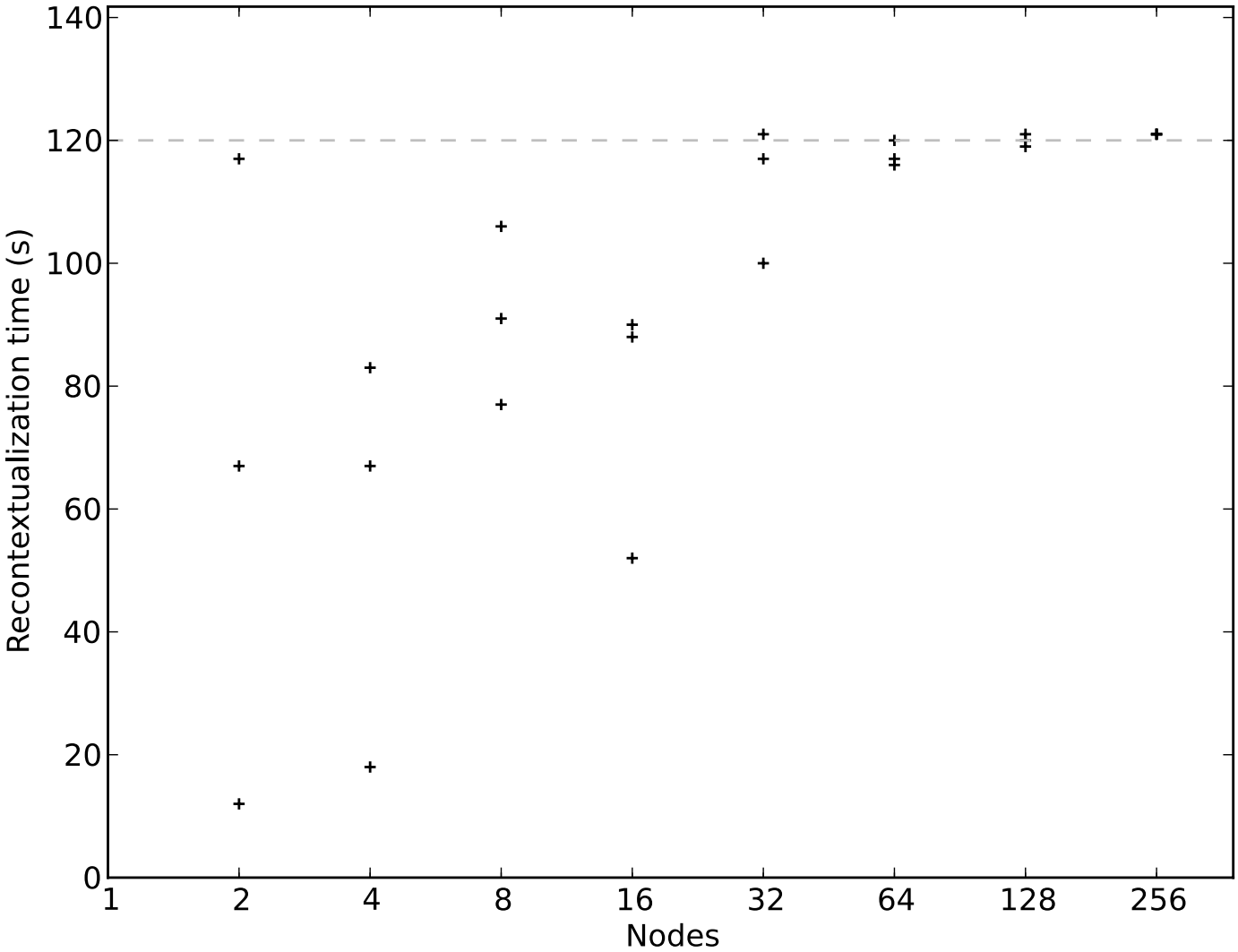
QIIME includes built-in support for clusters and will automatically submit jobs to the queue.

Everything else just works, like magic.

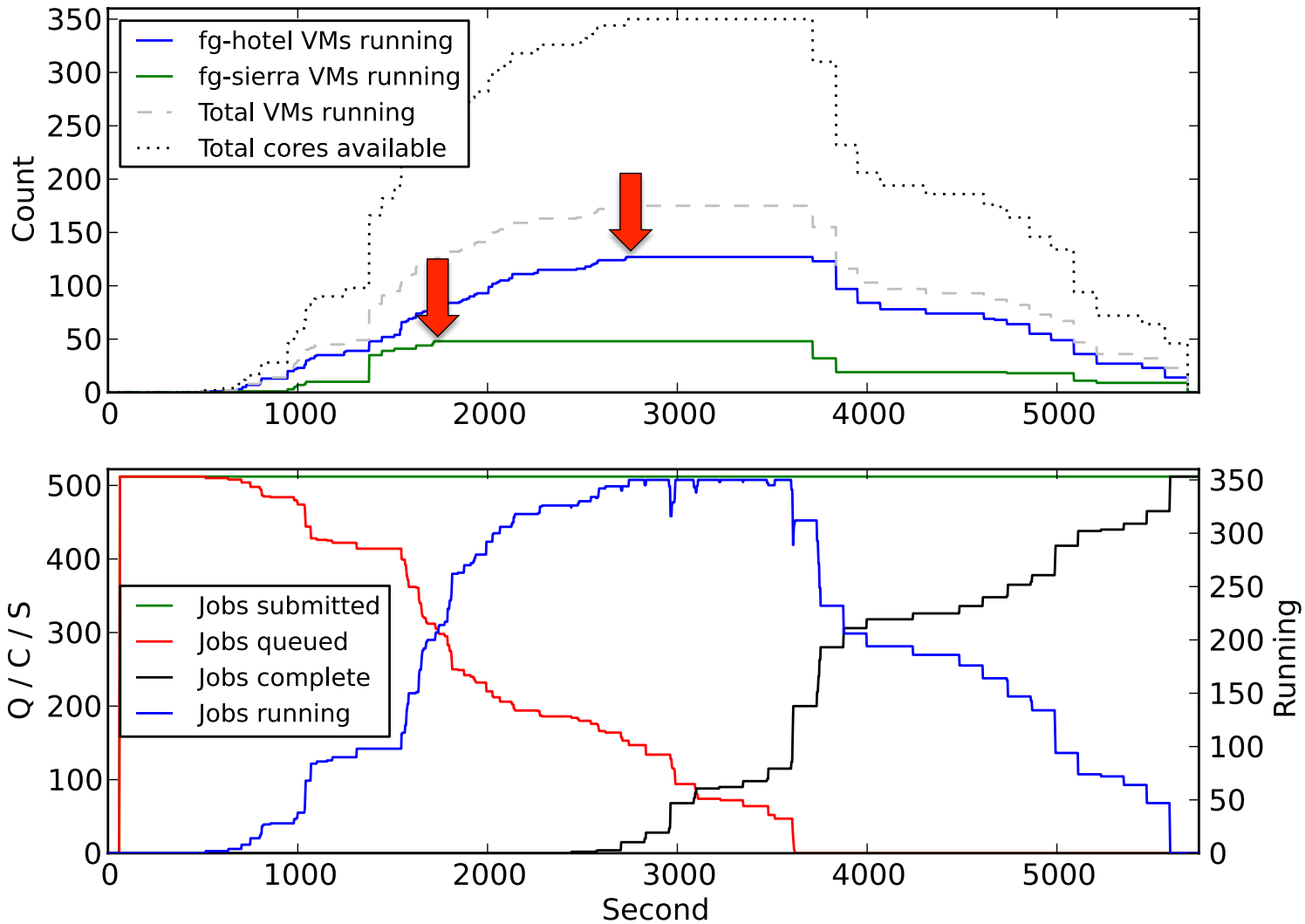
Evaluation Environment

- FutureGrid
  - Hotel at the University of Chicago (UC)
  - Sierra at San Diego Supercomputing Center (SDSC)
- Amazon EC2
  - Micro instances are $0.02 per hour
- Recontextualization
  - UC: 8 2.93 GHz Xeon cores and 16 GB RAM
- Torque head node
  - UC: 2 2.93 GHz Xeon cores and 2 GB RAM
- Workers
  - EC2: 64-bit EC2 east micro instances
  - UC/SDSC: 2 2.93 GHz Xeon cores and 2 GB RAM

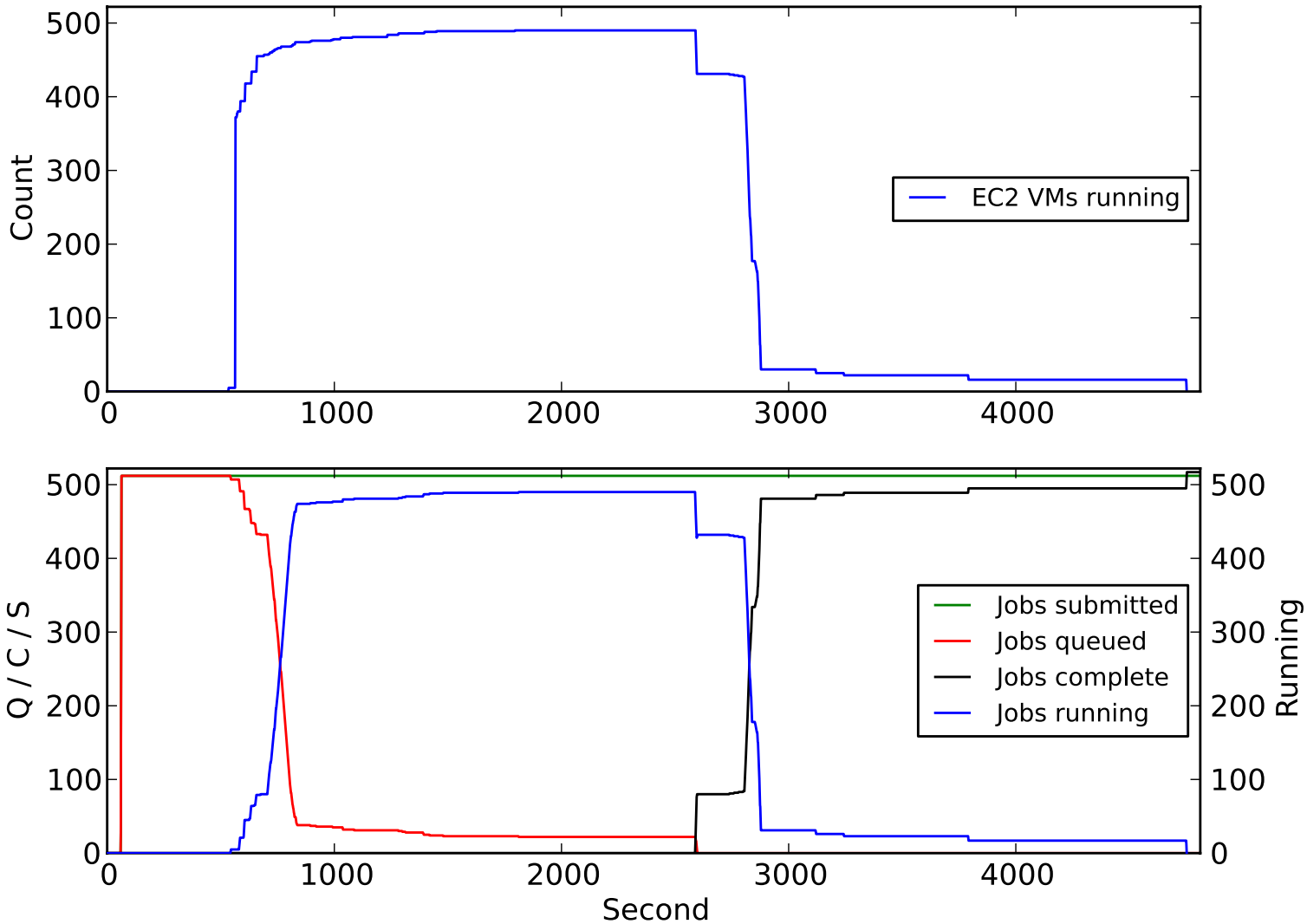
Recontextualization



Multi-cloud: Hotel (UC) + Sierra (SDSC)



Scalability: Amazon EC2



Future Work

- Develop additional policies
- Develop efficient data movement solutions for elastic cloud environments
  - Avoid sending unnecessary data both for performance and cost ($) concerns
- Evaluate using additional user applications and workflows

Thank you

Paul Marshall
University of Colorado at Boulder

paul.marshall@colorado.edu