Aggregating local clouds with reusable tools

- **Opportunities**
- **STAR experience & experiments**
- **Owning a pet local cloud**

Massachusetts Institute of Technology

Jan Balewski
for STAR Collaboration

U.S. DEPARTMENT OF ENERGY

2014 OSG All Hands

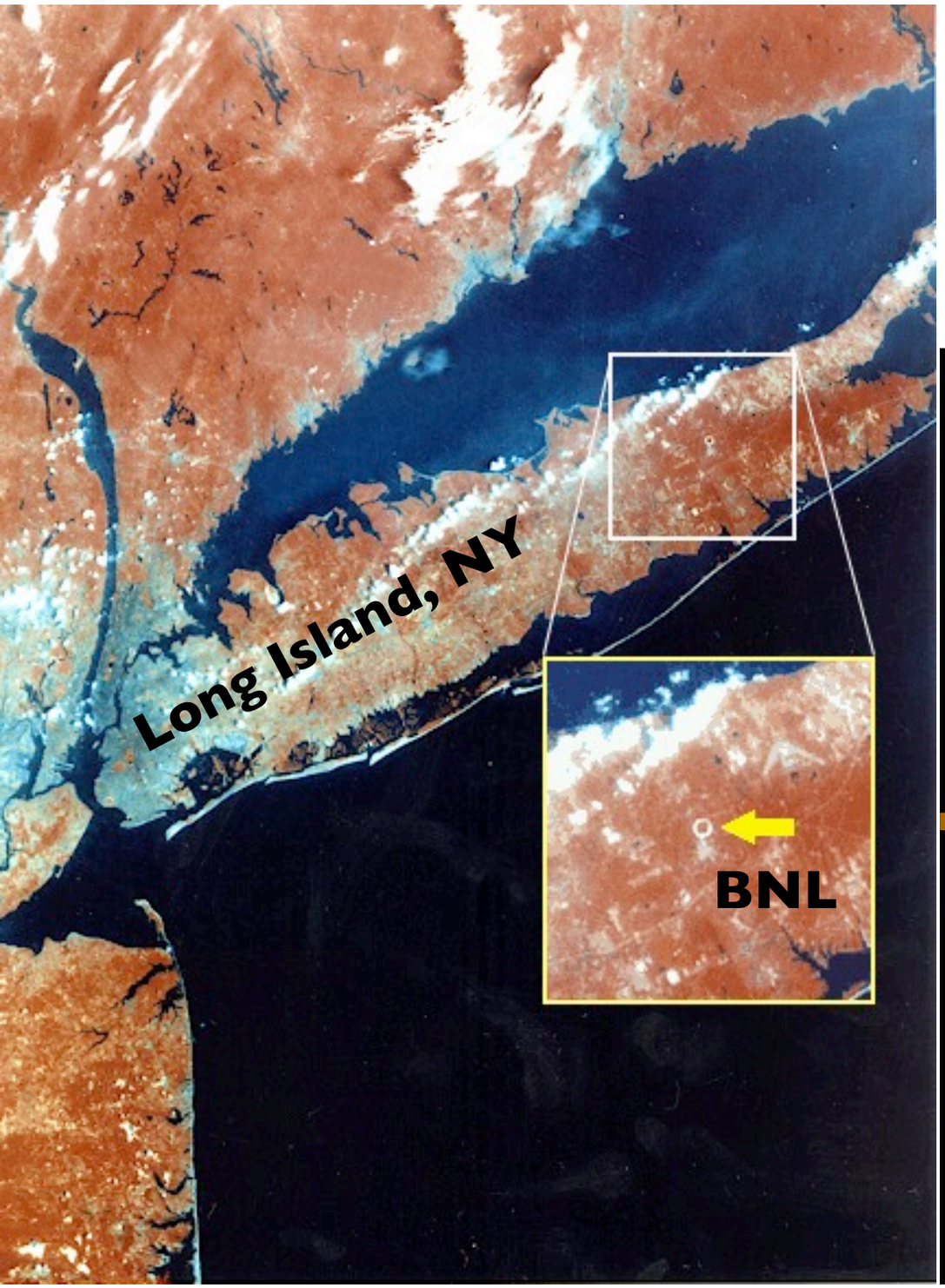
April 7-11, 2014

SLAC, Menlo Park, CA

Why Cloud Computing?

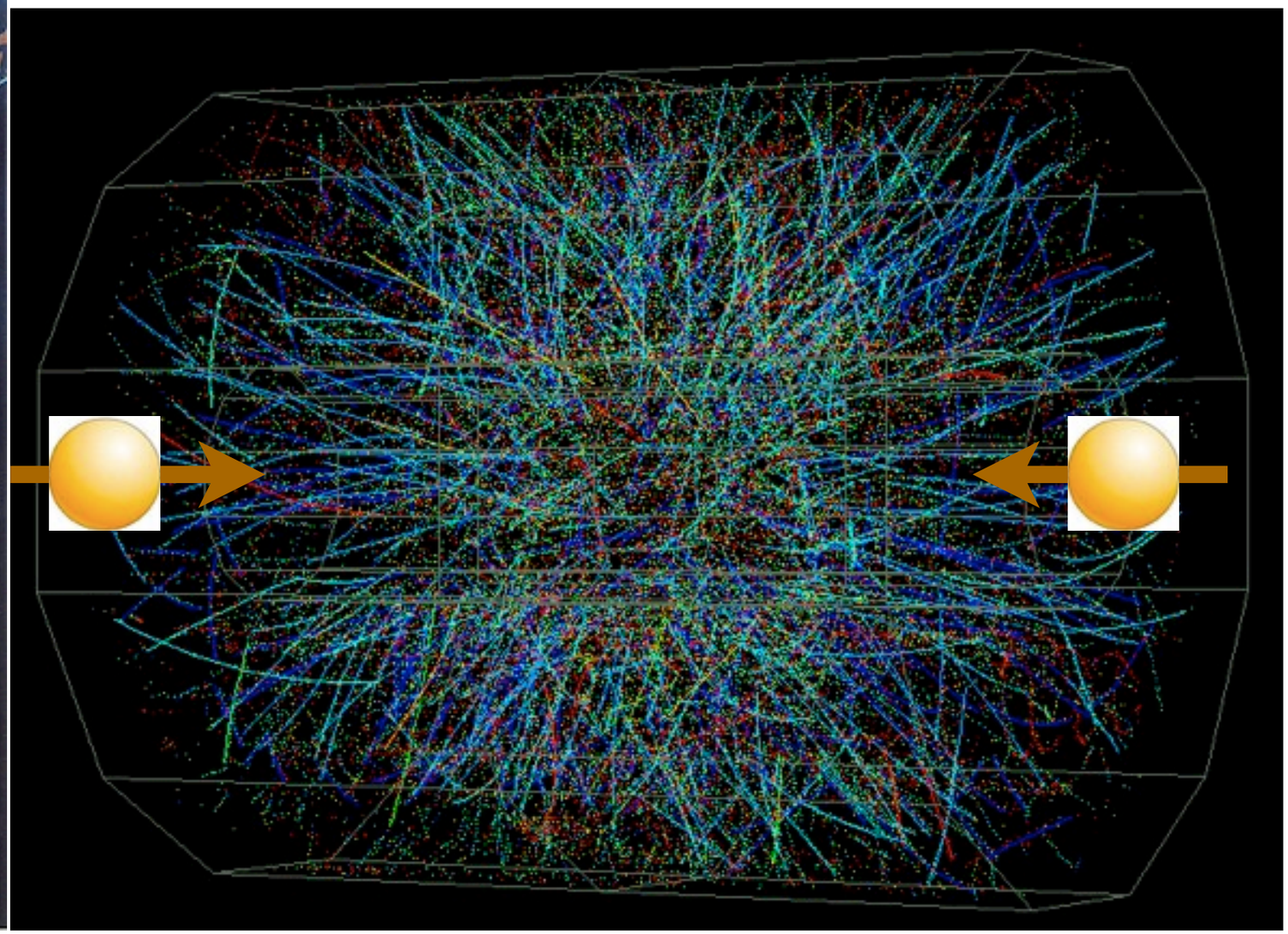
- **significant computing resources ‘for rent’, ‘free?’**
- **virtualization benefits**
  - hardware-agnostic computation
  - easy recirculation of resources for different users (opportunistic resources)
  - natural preservation of software in VM ,‘as used’, for future re-use
- **challenges**
  - image config management : before of after launch ?
  - security, authorization, job scheduling
  - adopt to site-specific clouds ‘flavors’, aggregation (e.g. Phantom)
  - I/O data transfer, multi-site balancing to match variable computing resources
  - time : resources stability, startup , lease-time (on-availability)
  - cost: EC2, grants (Magellan), on-availability (STAR online 100% available during off-run periods)
  - education : ‘when will you restore my files on VM after last crash?’

STAR experiment at RHIC



~600 collaborators from
~50 institutions and ~12 countries

Reconstruction of particles emerging from collision of two protons is a computational challenge



Brookhaven National Laboratory, Upton NY, USA

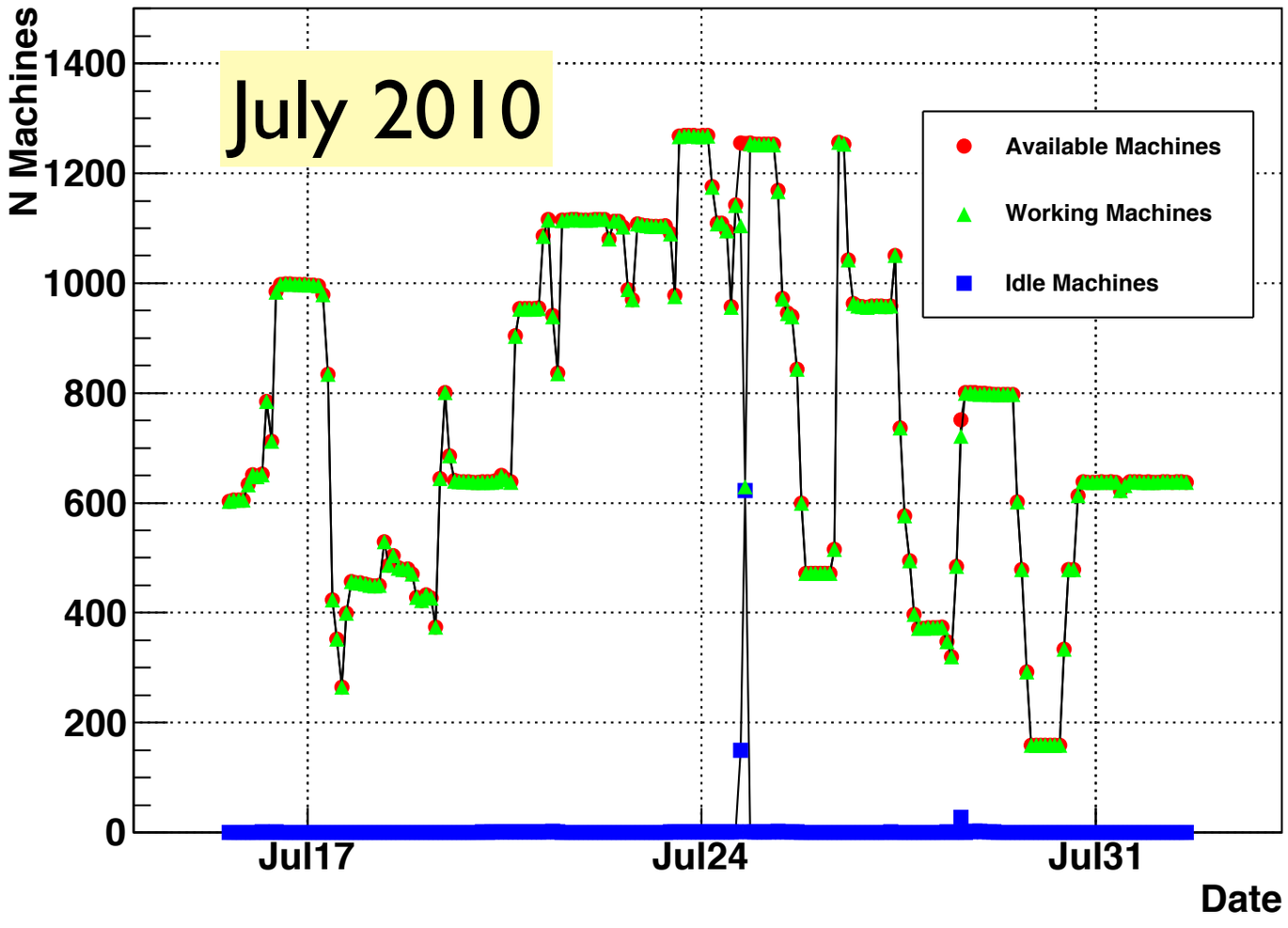
Former STAR encounters with VMs

date	Facility	tools	type of task	# of VMs	# jobs per VM	total CPU days	calendar days	total input (TB)	total output (TB)	remarks
2009, March	Amazon EC2	Nimbus Globus PBS batch	simu	100	1	500	5	0	0.3	works like normal globus GK grid site
2009, November	Amazon EC2	EC2	simu	10	1 or 2	1	1	0	0.01	use commercial interface
2010, February	GLOW Madison Uni Wisconsin	CondorVM	simu	430	1	130	0.6	0	0.1	call home model
2010, July	Clemson Uni, SC	Kestrel, QEMU-KVM	simu	1000	1	**17,000**	20	0	7	VM lifetime 24 h, no ssh to VM
2011, [Feb-May]	NERSC/ANL	Eucalyptus/ OpenStack	**data reco**	20...120	8	25,000+	120+	60	40	**near real-time processing**

Largest STAR simulations (ever) at Clemson

Matt Walker, MIT

- STAR MC simulations with partonic $p_T > 2$ GeV, PYTHIA event generator
- **Virtual Machine** prepared with STAR software stack and **deployed to over 1000 machines**
- Using cloud computing at **Clemson University in South Carolina** (Ranked #85 best supercomputer)



Available (red) and working (green) VMs track EXACTLY.

We clearly demonstrated that efficient use of truly **opportunistic** resources is possible and had a net gain/impact with no negatives for local jobs.

Multi-site STAR near real time data reco , June 2011

Magellan project resources
STAR experiment data/code

63 VMs
500 jobs

31 VMs
250 jobs

71 VMs
550 jobs

NERSC
Magellan

ANL
OpenStack

ANL
Nimbus

STAR
DB
snapshot

BNL

STAR
experiment

NERSC
cache
20 TB
gpfs

raw data: 3TB/day

reco events 2TB/day

RCF
HPSS

- coordination of data/storage and computation is not trivial.
- use the ANL resources efficiently, we had to transfer data from BOTH BNL->ANL and NERSC->ANL.

Irmo : the cloud of clouds

http://press3.mcs.anl.gov/irmo/

The Irmo project develops a vision for an infrastructure strategy that seeks to combine the different usage modalities present in DOE communities under one model that encourages collaborative sharing, supports community growth, accommodates emergent usage patterns such as on-demand computing, and lowers the entry barrier to the use of DOE facilities from desktop to exascale.

Highlights :

- develop best practices
- provide (identify exiting) tools to handle ease customization
- apply the same recipe even if the Cloud is made of several clouds (e.g. Phantom service)

VM formats - similar but different

amazon web services™

master copy

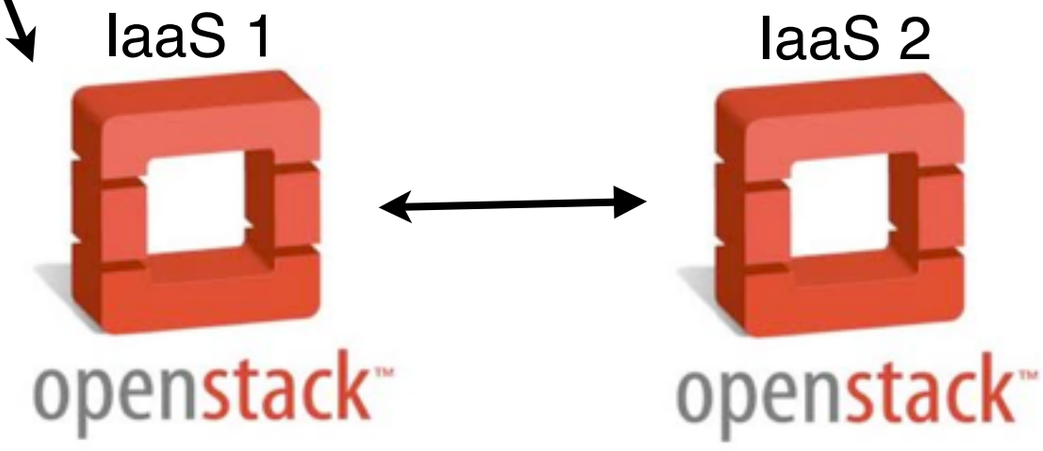
vmware

3-part format (ami/aki/ari)

- 3 files must match
- bundling (i.e. chopping) is needed
- any kernel update requires re-bundling
- may exclude some (confidential) content
- (3x less) easy to copy
- VM disc size matches launch ‘flavor’

vmdk format

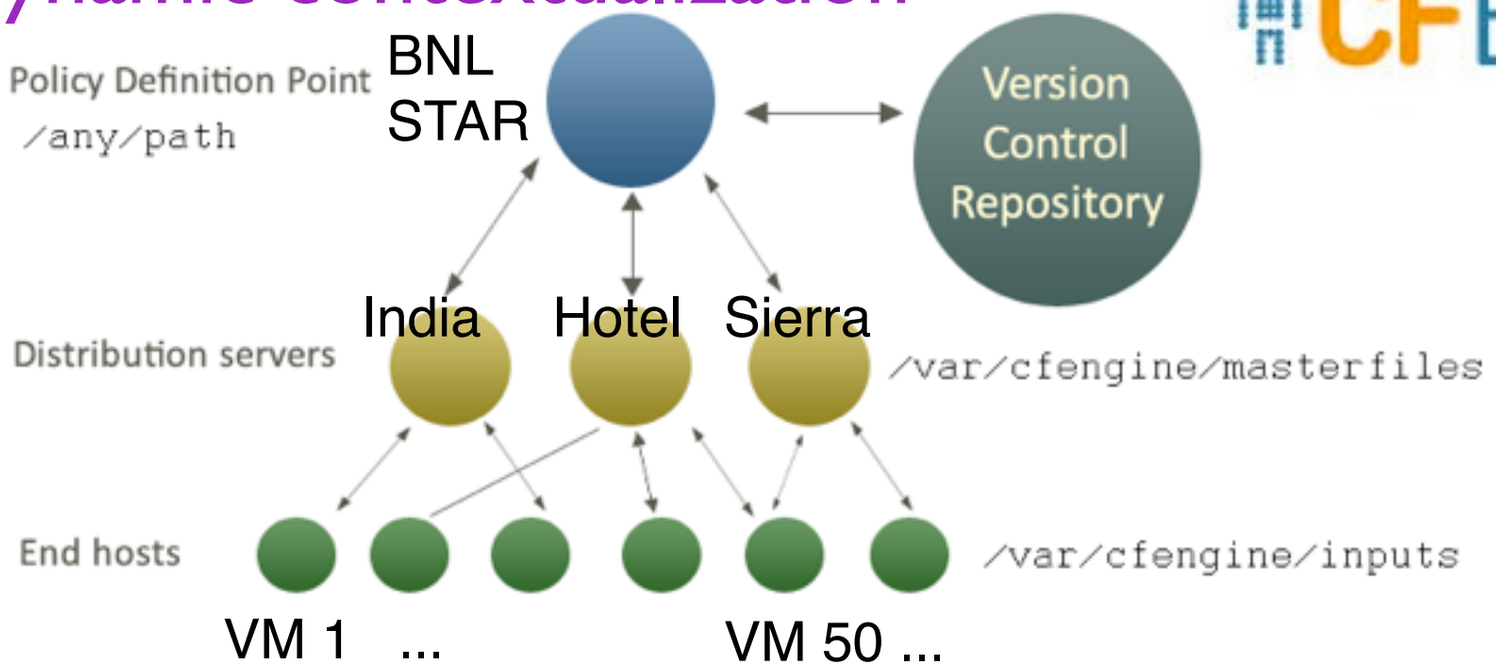
- single image file (10-20 GB)
- whole VM image ‘from outside’
- no bundling step
- easy to copy
- not elastic VM disc



VM content management

Dynamic contextualization

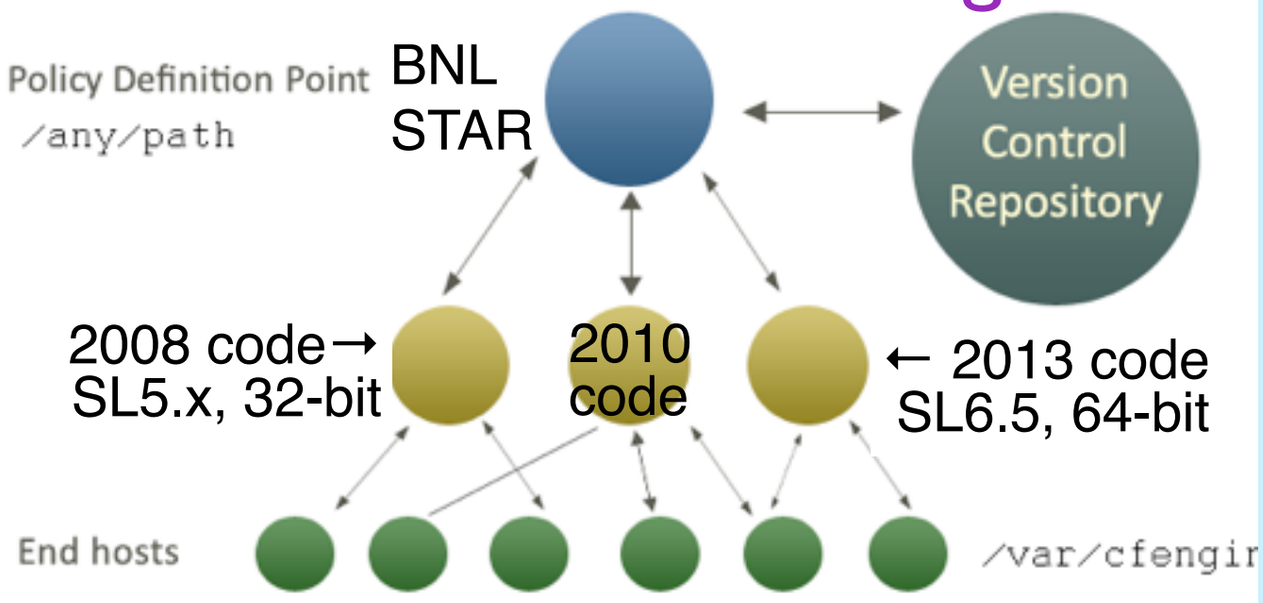
Michael Poat, BNL



- lightweight
- scripted (reusable)
- widely scalable to 1000's VMs
- policy syntax non-trivial
- tested even on submarines (limited internet access)

CFE is complementary to user-data injection at launch

Automatic code versioning



Dilemmas:

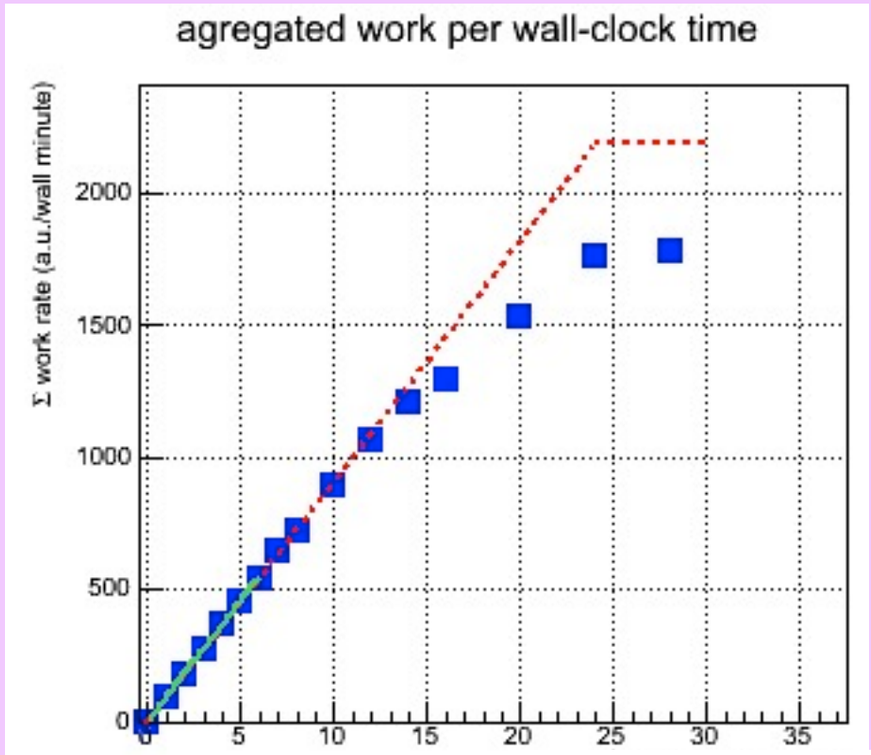
- 1 universal CFE policy & long VM boot time (which may fail), or
- archive each VM version (build once, use many times) & book-keep 100s GB of VM images
- IaaS may need site-specific VM changes

→ book-keep & archive 100s of VM images

We are exploring & learning

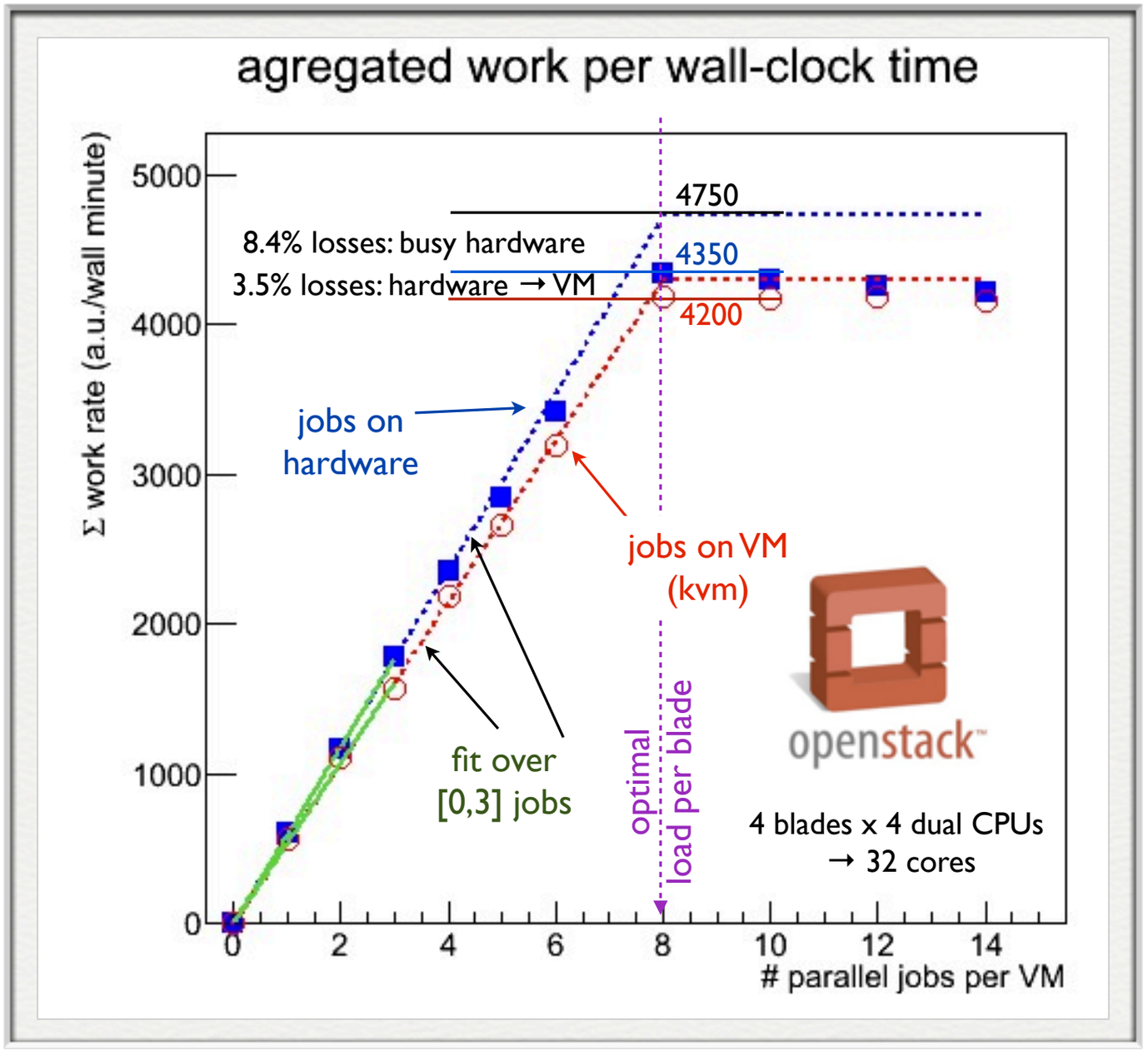
3.5% CPU losses due to virtualization

Reference : rcas6001 @ RCF/BNL
24 core, 48 GB RAM



Hardware: 4 **Workers**: Dell 1950,
dual quad CPU (8cores) + 16GB RAM + 140 GB disc

Test 1) run jobs on 8-core hardware, Ubuntu 12
Test 2) run jobs inside 8vCore VM, SL 6.4

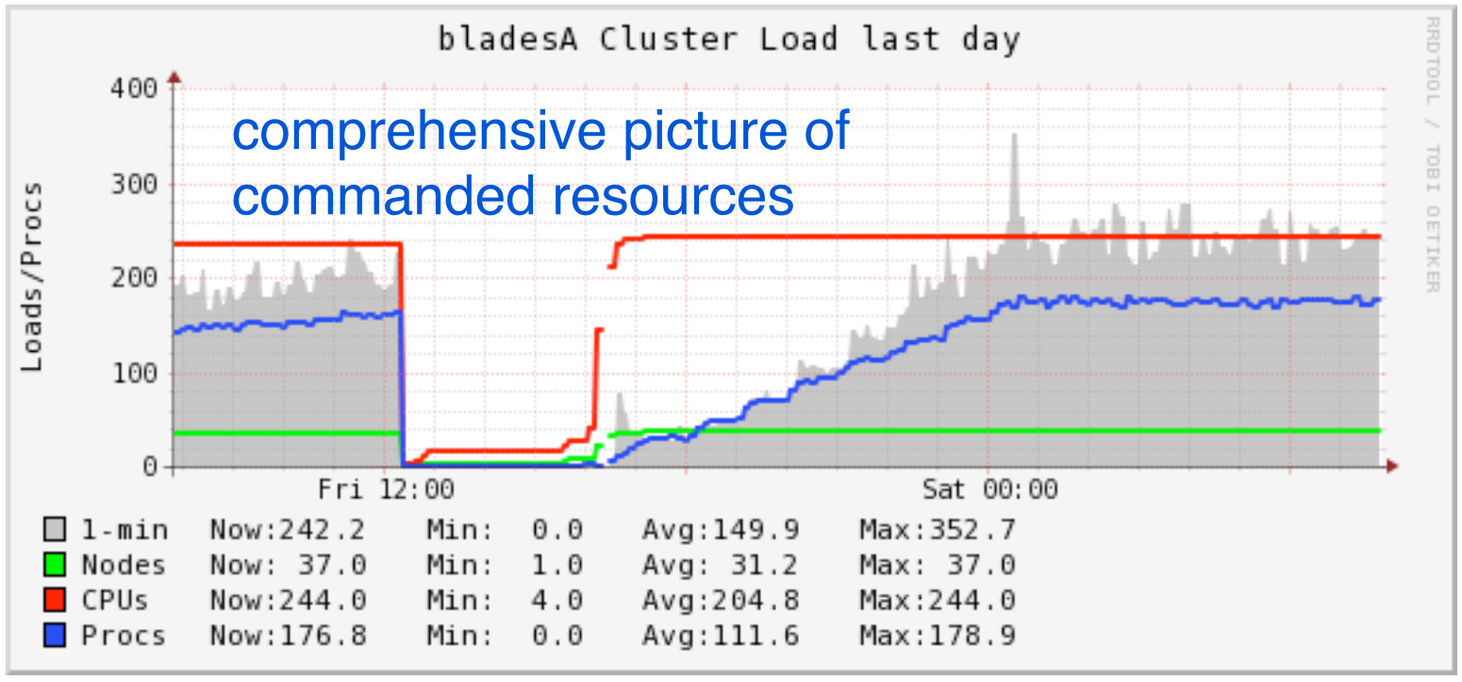



```
bigArray= (double*)malloc(sizeB); // size=300MB

WORK UNIT: math & memory ops, no I/O
 for(int i=0;i<nOps;i++) { //  'work'=math & memory operations
    int j=random()%sizeN;
    double x=random()*1./RAND_MAX,  y=random()*1./RAND_MAX;
    double z=sin(x)*pow(y,1.123);
    bigArray[j]=z;
    j=random()%sizeN;
    bigArray[j]=pow( bigArray[j],2.3);
  }
```

Timing measured by
clock_t begin = clock(); // CPU time in usec
time_t beginWallT; time(& beginWallT); // 1 second accuracy

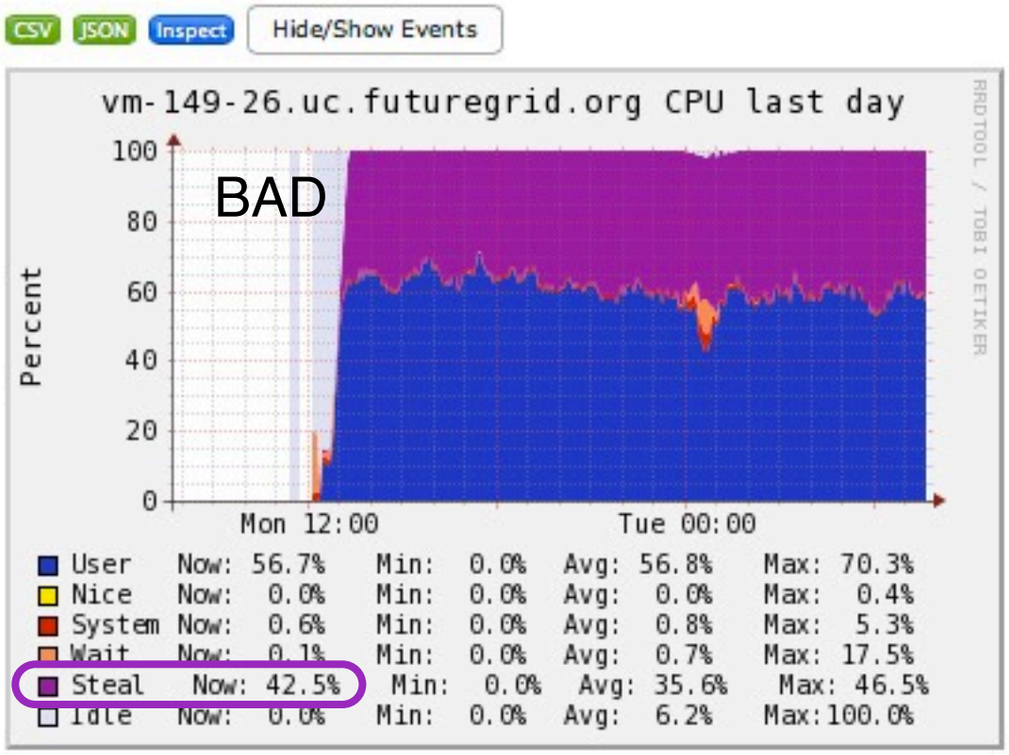
Ganglia monitoring inside each VM

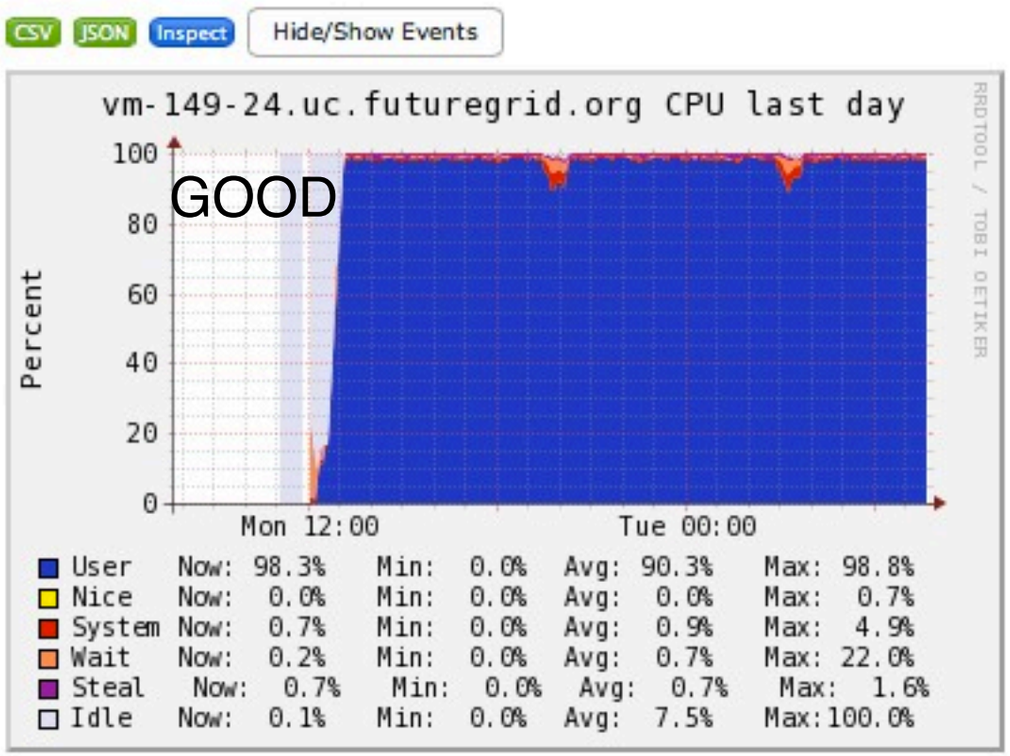


Access to DNS name of host blade

149.165.158.65
149.165.159.43
149.165.159.48
guest21.eucalyptus.sierra.futuregrid.org
guest3.eucalyptus.sierra.futuregrid.org
reuse40.lns.mit.edu
reuse44.lns.mit.edu
reuse46.lns.mit.edu
reuse49.lns.mit.edu
reuse51.lns.mit.edu
reuse53.lns.mit.edu
reuse55.lns.mit.edu
reuse65.lns.mit.edu
reuse67.lns.mit.edu
s64r.idp.sdsc.futuregrid.org
s66r.idp.sdsc.futuregrid.org

detect inefficient use of blades





FutureGrid
an XSEDE resource provider

2013++

Multiple clouds & interfaces

https://portal.futuregrid.org

(selected information about selected participants)

Selected Details of the Clusters

Name	Alamo	Bravo	Delta	Foxtrot	Hotel	India	Sierra
Organization	Texas Advanced Computing Center	Indiana University	Indiana University	University of Florida	University of Chicago	Indiana University	San Diego Supercomputer Center
Number of CPUs	192	32	32	64	168	256	168
Number of nodes	96	16	16	32	84	128	84
Total RAM (GB)	1152	3072	3072	768	2016	3072	2688
Number of cores	768	128		256	672	1024	672

Our focus: [site]-**openStack** sub-clusters, fraction of CPUs, **unified VM format**, similar to openStack at STAR-online, MIT-reuse, Amazon-EC2 clusters

One Phantom to rule them all

https://phantom.nimbusproject.org/

Pierre Riteau, U. Chicago

NIMBUS

Phantom | Profile | App

Cloud Credentials

Cloud	Status
hotel-kvm	Disabled
india-openstack	Enabled
ec2-eu	Disabled
alamo	Enabled
hotel	Enabled
mit-openstack	Enabled
sierra-openstack	Enabled
wispy	Disabled
alamo-openstack	Enabled
ec2	Disabled
sierra	Enabled
hotel-openstack	Enabled
foxtrot	Enabled

hotel solo Launch Configuration

Appliance:

Contextualization Type: User Data

User Data

```
#!/bin/sh
echo "*********JAN: injecting test START *********"
whoami
pwd ; ls -l
```

Save Delete

Available Clouds

Cloud	Status
hotel-openstack	Enabled
alamo	Disabled
hotel	Disabled

Hotel-openstack Options

Maximum VMs: 6

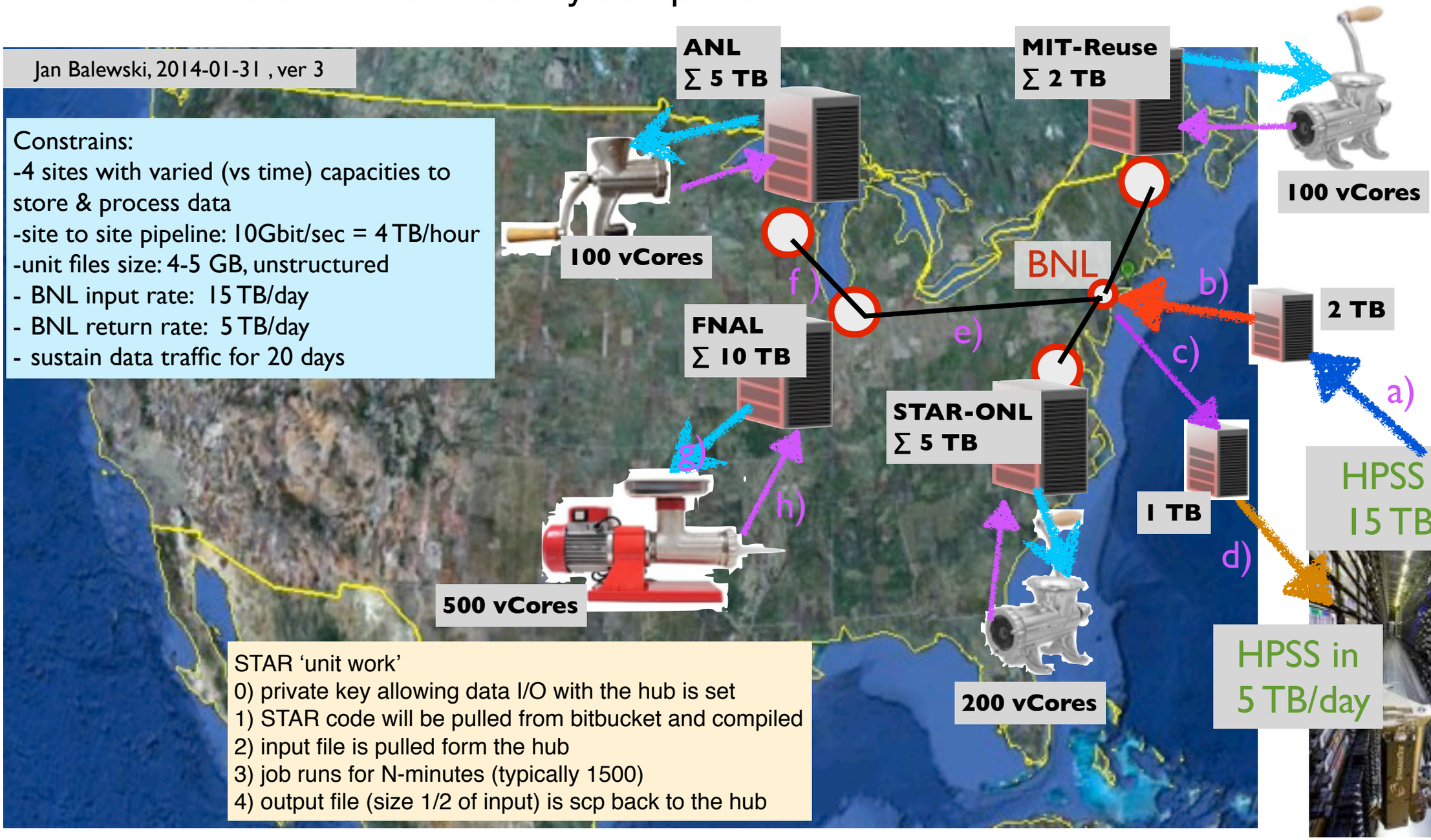
Instance Type: m1.large

Public Image ami-00000034

Goal: distributed STAR data processing

'Cloud of the clouds' demonstrator

- outreach to Fermi/Cloud for a common activity and project
- exercise Phantom within Irmo project
- include on-availability component



Deploying local cloud at MIT/LNS

Motivation

- participate in STAR data challenge on the cloud
- learn how to admin a local cloud
- let students learn how to use cloud resources (for free)

Choice: OpenStack (recycled hardware & free software)

- blades : Ubuntu 12.04 STL
- virtualization: devstack.org , **single script deployment**
- load monitoring : ganglia 2x
- blades (Dell 1950, 2850, 1650):
  - 8-core, kvm, small disc, 64-bit → **workers**
  - 4-core, qemu, large disc → **controller**, NFS storage
  - 32-bit blades → ganglia reporting, NFS storage
- blades connected at 1 GBit/sec : Catalyst 4000 switch
  - flat network for VMs
- public class-C network: blades IP<100, VMs on demand

Networking

iperf network bandwidth test

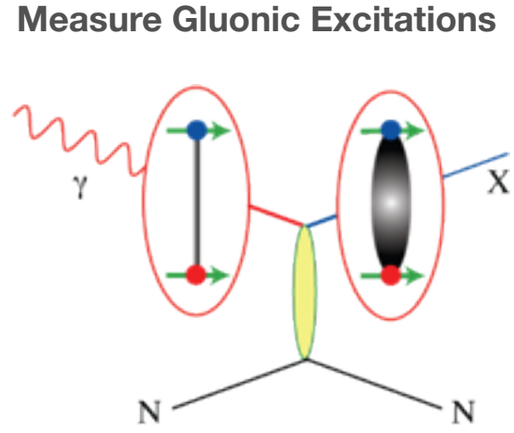
a) MIT blade => FG: 707 Mbits/sec
b) MIT blade => LBL: 267 Mbits/sec
c) MIT VM => FG: 680 Mbits/sec
d) MIT VM => LBL: 233 Mbits/sec

Paul Acosta, MIT

GlueX simulation: 10 billion events

Simulations code

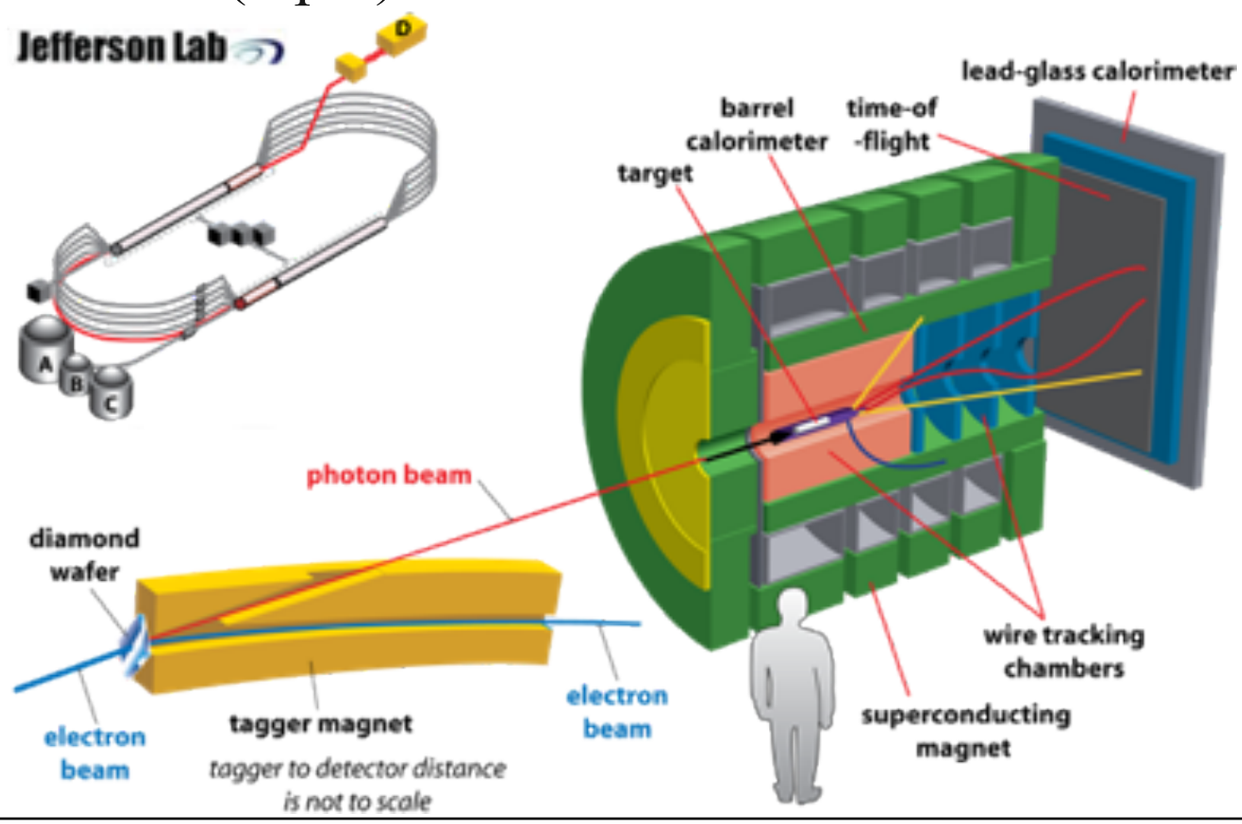
- SL6.5, Geant3, CERN ROOT
- GlueX software (C++)
- job profile: CPU intensive, large intermediate file, small output, ~no input



Strategy

- collaboration wide simu : JLAB, local computing centers, OSG
- each site assigned different range of random seeds (input)
- common output collection host
- simultaneous start of simulation
- produce as much as you can

Target MIT/FG contribution of 5 %



Job authorization,scheduling,monitoring

Basic idea: user passes an individualized script to VM at launch, (is executed as root)

**euca-run-instances -n 1 -t m1.small ami-00000009 **
-f helloWorld.sh

```
# cat helloWorld.sh
#!/bin/sh
echo "********JAN: injecting test  *********"
pwd ; whoami;
MYIP=`/sbin/ifconfig eth0 |grep Mask |cut -f2 -d: |cut -f1 -d\ `
echo " Hello World from VM with  local IP=$MYIP "
echo "********JAN:  test END  *********"
```

Instance Console Log

```
Cloud-init v. 0.7.4 running 'modules:final' at Tue,
********JAN: injecting test? *********
/
root
 Hello World from VM with? local IP=10.50.0.23
********JAN:? test END? *********
```

```
#1 store ssh key for copying output
cat > id_rsa_tier2 <<EOF
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAsqiRlBX2xOUnLnqxk0vRuA0jGAiie
yAiOwl7xb8b8YPa8/HlAtFmOYNXSxjYct9y8TSTu+PEX6
....

#2 redirect ganglia monitoring
sed -i "s/8663/${GANGLIAPORT}/g" /etc/ganglia/gmond.conf
service gmond restart
....
#3 install new GlueX binaries
wget http://reuse38.lns.mit.edu/gluex-dc-2/sim-recon-dc-2.7.tgz
tar -xzvf sim-recon-dc-2.7.tgz
...
#4 copy in GlueX configuration files
wget http://reuse38.lns.mit.edu/gluex-dc-2/conditions/getSeed.sh
chmod a+x getSeed.sh
...

#5 write input  script controlling pythia simulations
cat >> run.ffr <<EOF
RUNNO    ${RUNNO}      run number of generated events
CRNDMSEQ  \${SEED}       random number sequence
...

#6 runs the simulation <= THE JOB
hdgeant run.ffr
...
#7 copy the output from VM to other location Tier2/Bates
scp -i id_rsa_tier2   -r dana_rest_0${RUNNO}_\${SEED}.hddm  ....
```

loop N times

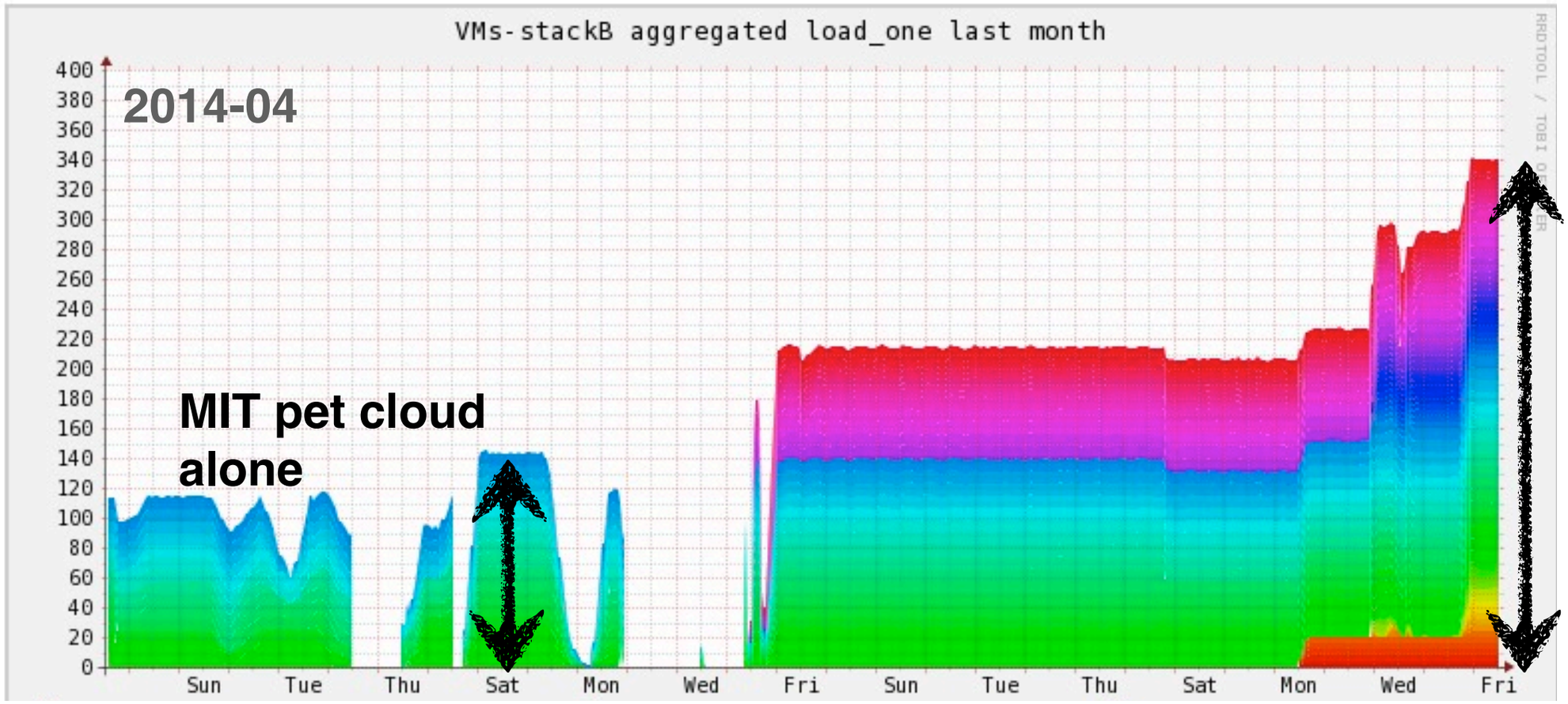
Justin Stevens, MIT

GlueX simulation on 'Cloud of Clouds'

Justin Stevens, MIT

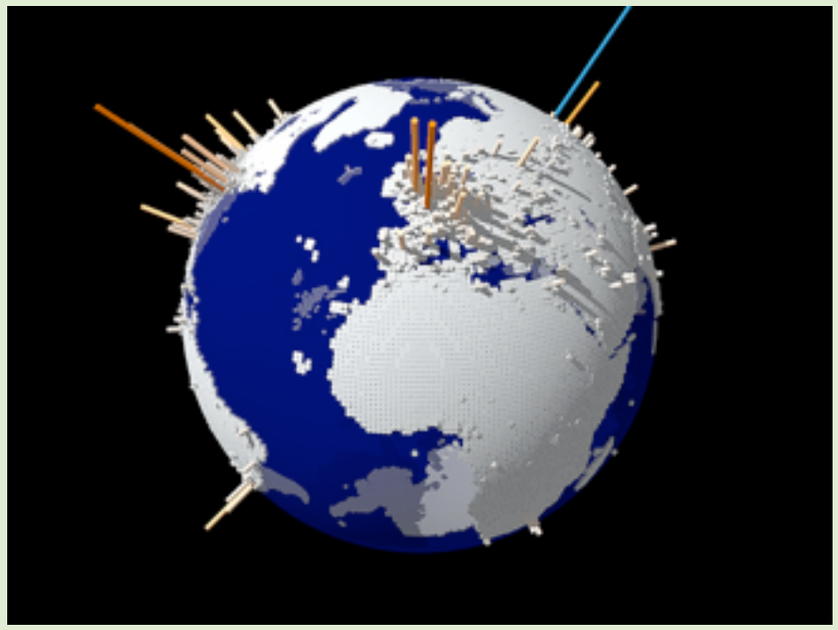
tally of resources allocated on various IaaS using Phantom

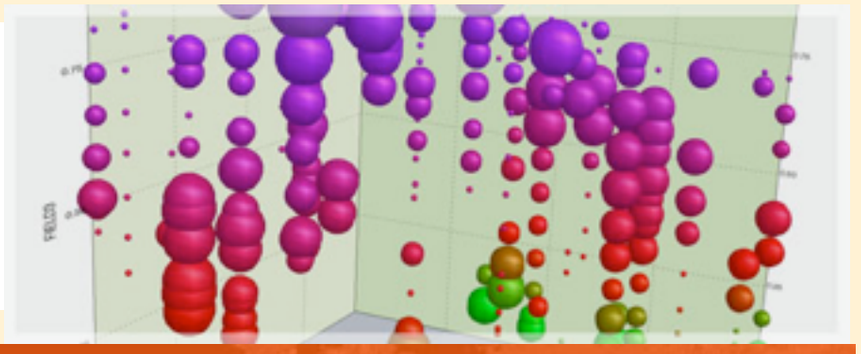
Site	Location	Openstack flavor	# Cores
mit-stackb	MIT	grizzly	136
hotel	University of Chicago	havana	100
india	Indiana University	havana	60
sierra	UC - San Diego	grizzly	56
alamo	University of Texas - Austin	folsom	0
total			352



Licensed software on VMs

VM/cloud help students/postdocs in general research



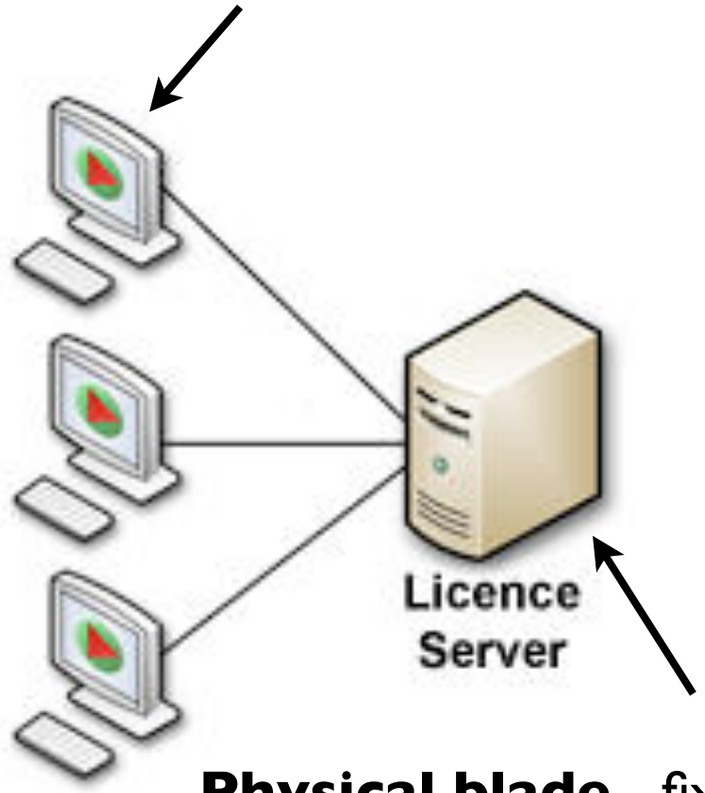


IDL. Discover What's In Your Data.

IDL is the trusted scientific programming language used across disciplines to extract meaningful visualizations from complex numerical data. With IDL you can interpret your data, expedite discoveries, and deliver powerful applications to market. Additionally, IDL is a truly cross-platform solution, providing support for today's most popular operating systems, including Microsoft Windows®, Mac OS X, Linux, and Solaris.

VMs on high core count blades
volatile MAC address
base VM image:

- SL6.5 (6 GB)
- Matlab (12 GB)
- IDL (2 GB)
- created ~10 users accounts
- assigned public IP after launch



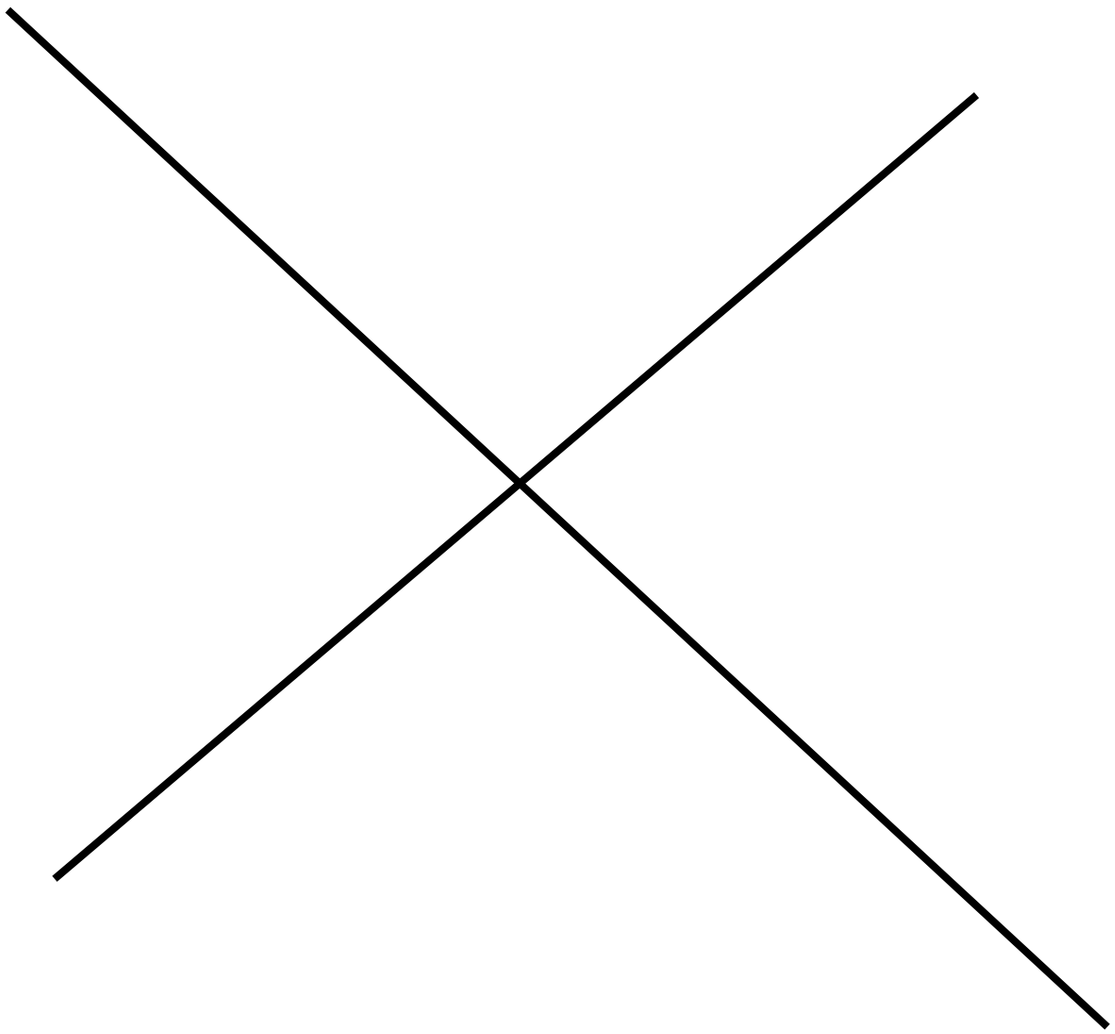
Physical blade, fixed IP
hardware MAC adr.
FlexNet License Manager

- 3 Matlab licenses
- 3 IDL licenses

Summary

- demonstrated use of opportunistic resources, 'on-availability' is at reach
- building 'cloud of clouds' (Irmo) is complex task, many challenges
  - multi-site integration (e.g. Phantom) and site-to-site similarity (e.g. OpenStack)
  - VM content management : static VM master copy, in-fly CFEngine policy, user-data injection
  - security models: ssh-key, license server, Grid/GSI ?
  - job scheduling: condor, user-data injection, CFEngine policy
- virtualization provides time-capsules for task specific code & OS
  - small local clouds based on surplus hardware allows for experimentation
  - multi-thousand cores virtual cloud farms needed for practical applications

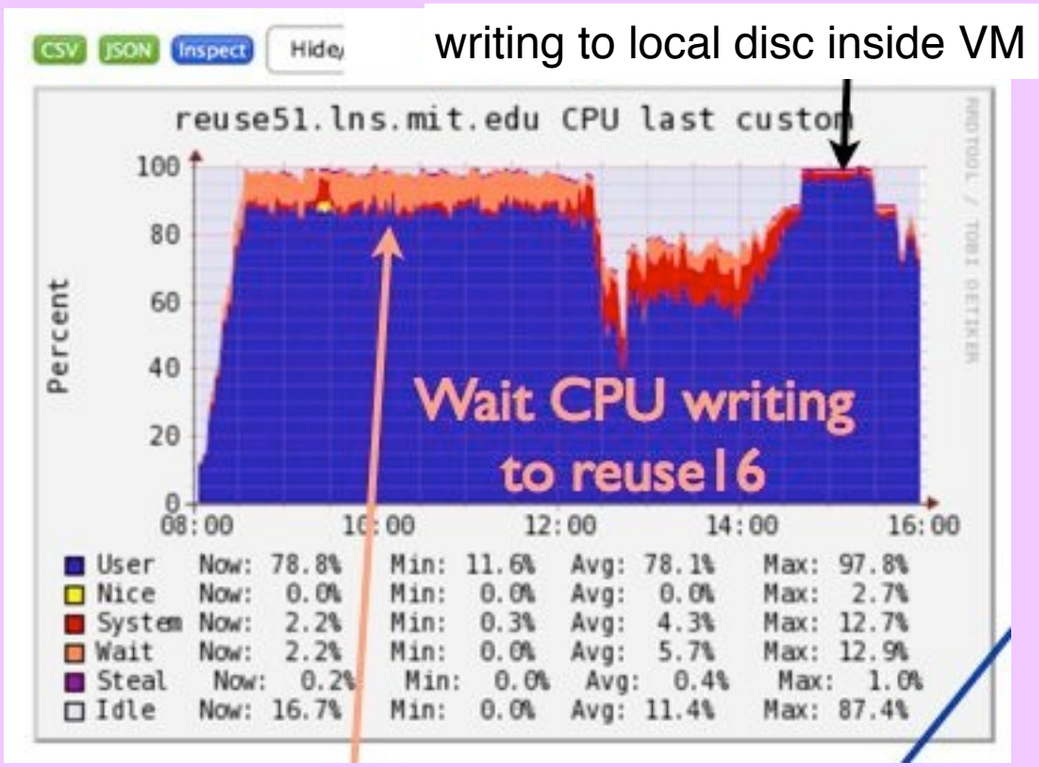
Backup



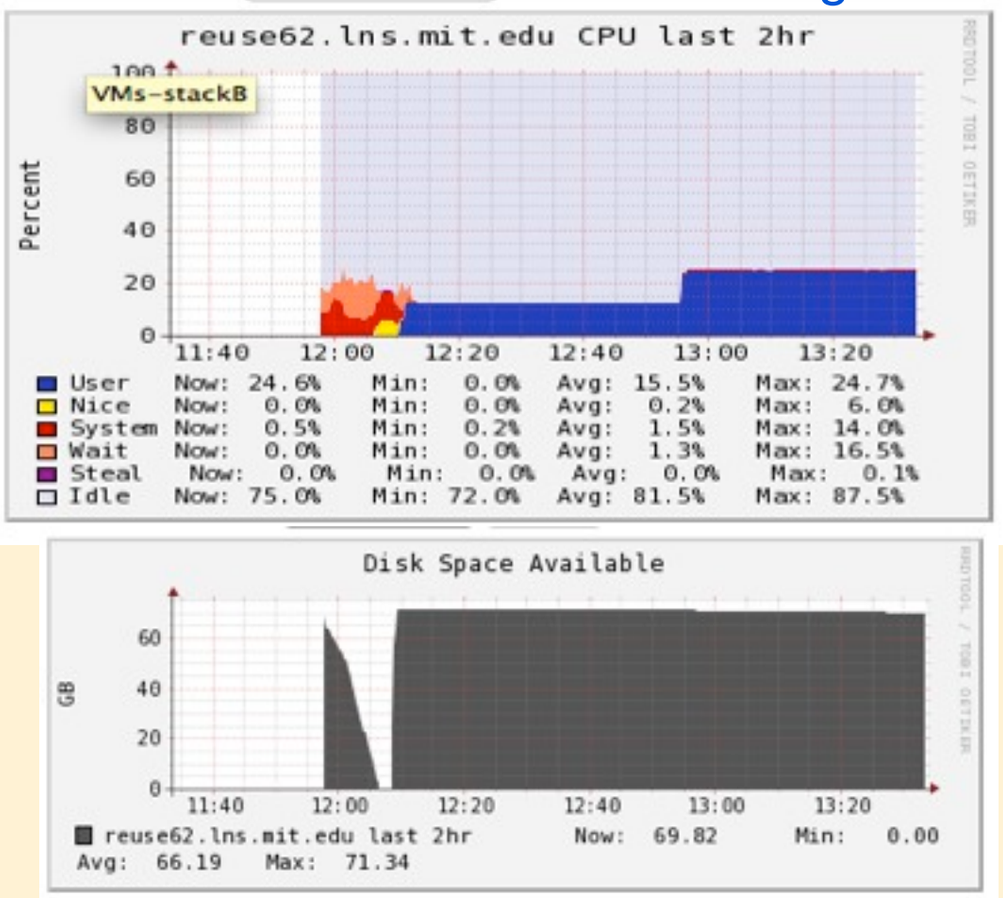
Optimizing VM 'flavor' for GlueX simu

Justin Stevens, MIT

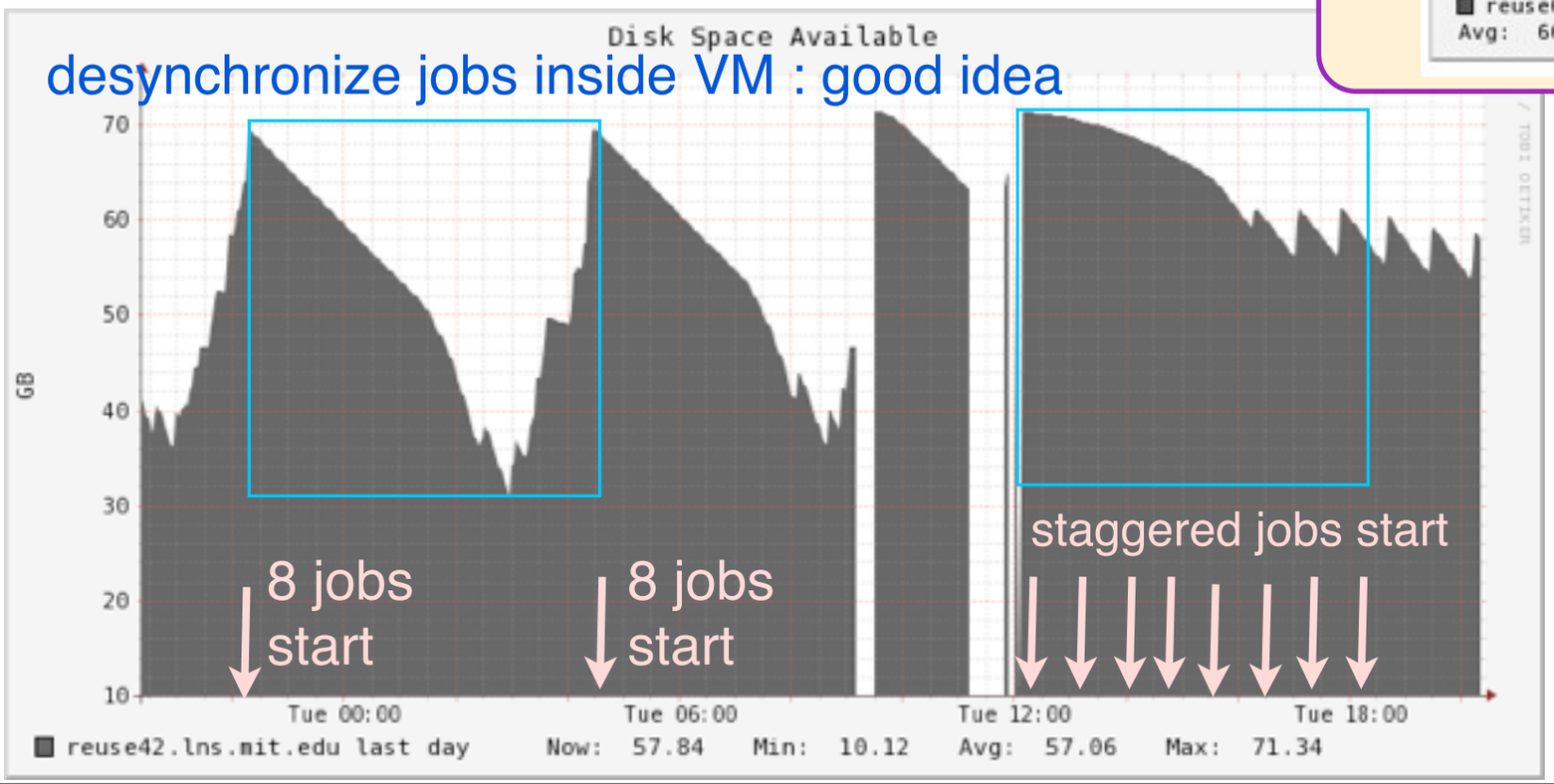
continuos write to NFS disc: Bad idea



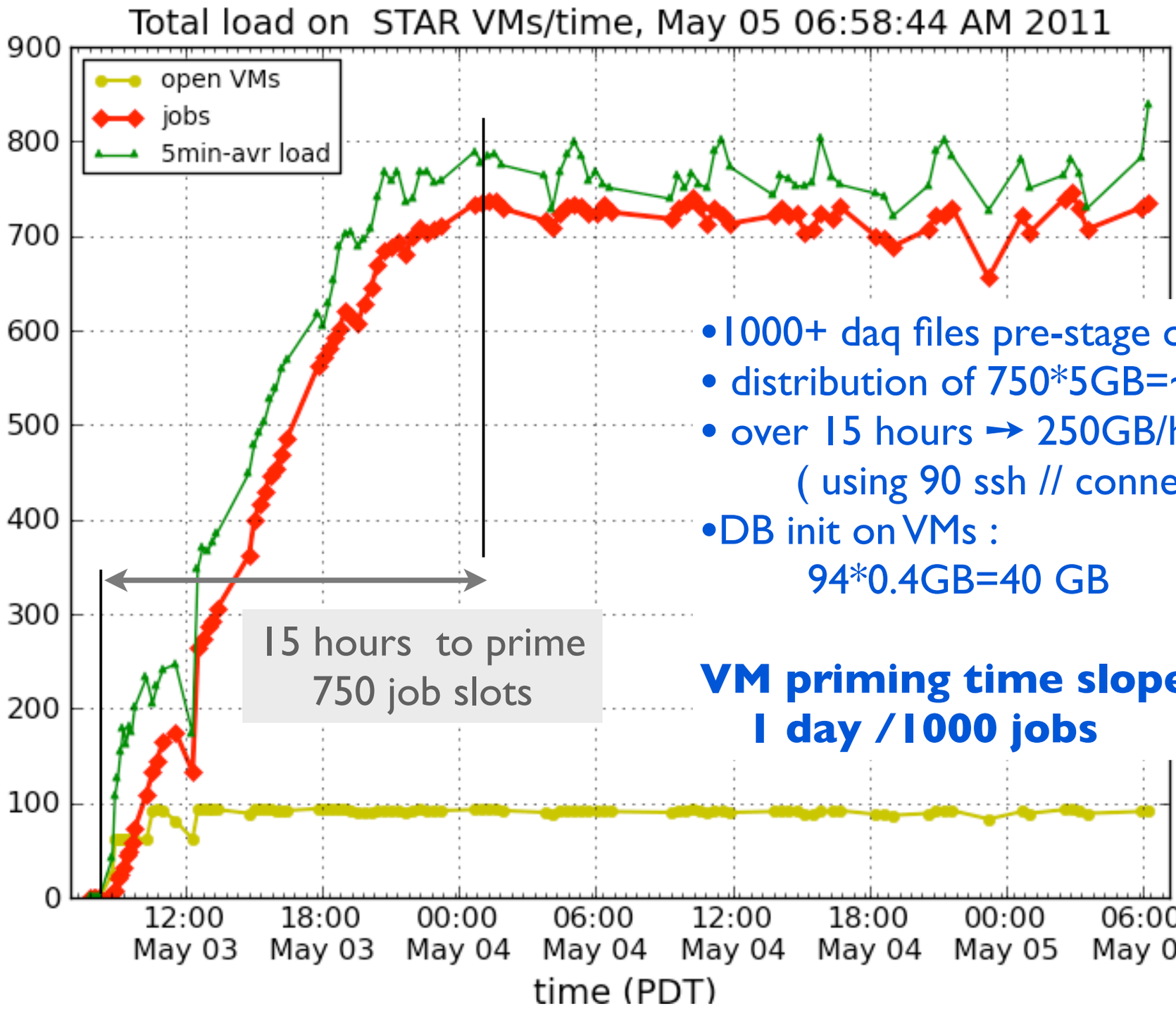
'stretch' elastic VM disc at launch: good idea



desynchronize jobs inside VM : good idea



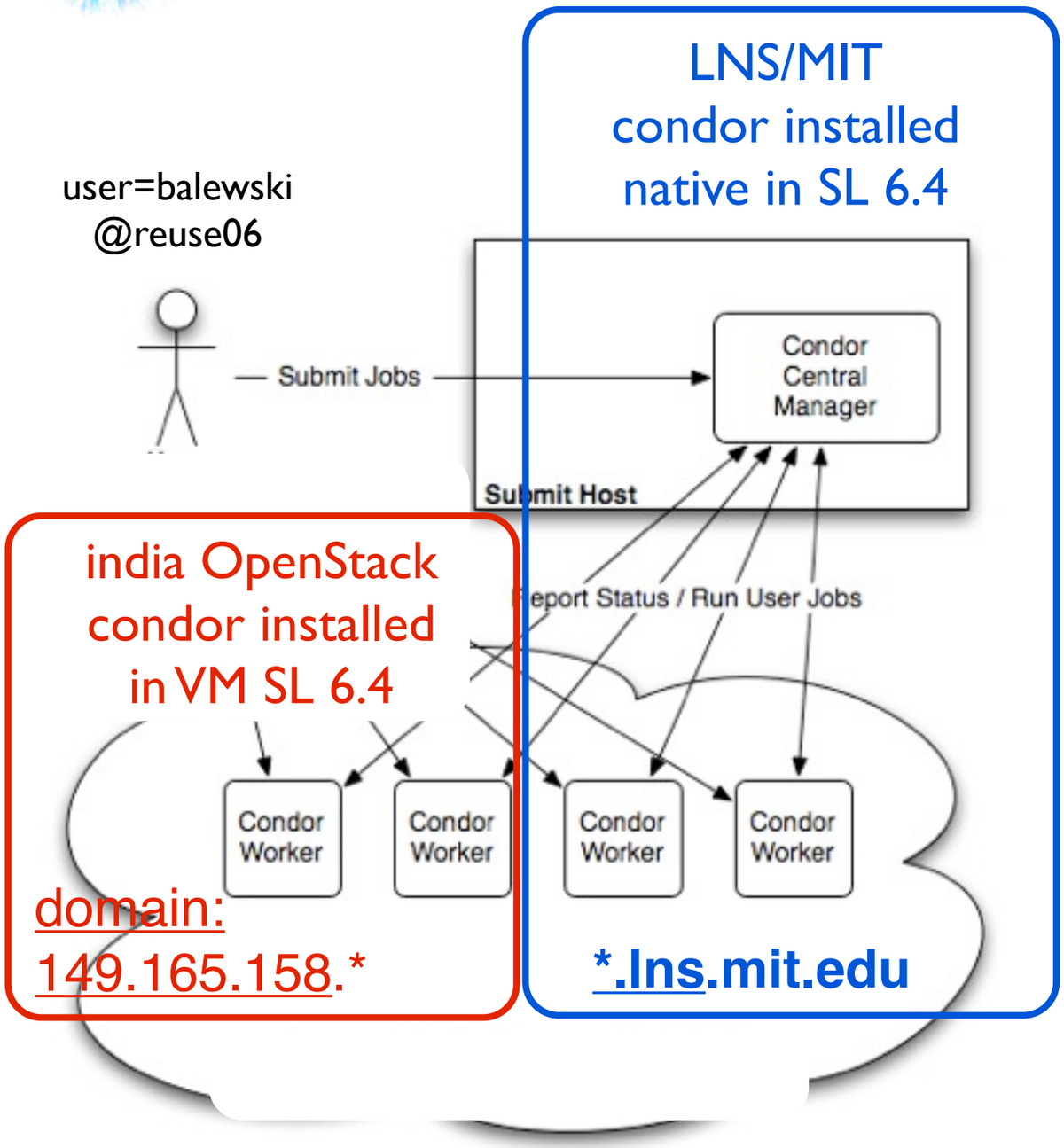
"Cold start" of 94 VMs w/ 750 slots



- 1000+ daq files pre-stage on carver
- distribution of 750*5GB=~4TB
- over 15 hours ➙ 250GB/hour
  (using 90 ssh // connections)
- DB init on VMs :
  94*0.4GB=40 GB

VM priming time slope:
1 day /1000 jobs

multi-site condor inside VMs



Original idea, discussion of implementation

https://sites.google.com/site/patrickbwarren/ec2-condor-cluster

http://www.isi.edu/%7Egideon/condor-ec2/

Example of fully loaded condor cluster w/ one controller, consisting of CPU 20 slots (5 machines x 4 slots)

```
[root@reuse06 ~]# condor_status
Name                OpSys      Arch   State     Activity LoadAv Mem   ActvtyTime

Condor-Controller+worker_0 @  native SL6.4_MIT
slot1@reuse06.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:04
slot2@reuse06.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:05
slot3@reuse06.lns.mit LINUX      X86_64 Claimed   Busy     0.140 8026  0+00:00:06
slot4@reuse06.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:07

Condor-worker_1 @  native SL6.4_MIT
slot1@reuse05.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:04
slot2@reuse05.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:04
slot3@reuse05.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:05
slot4@reuse05.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:06

Condor-worker_2 @  native SL6.4_MIT
slot1@reuse07.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:04
slot2@reuse07.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:05
slot3@reuse07.lns.mit LINUX      X86_64 Claimed   Busy     0.000 8026  0+00:00:06
slot4@reuse07.lns.mit LINUX      X86_64 Claimed   Busy     0.080 8026  0+00:00:07

Condor-worker_3 @ VM_india_openstack
slot1@149.165.158.49 LINUX       X86_64 Claimed   Busy     0.340 1999  0+00:00:04
slot2@149.165.158.49 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:05
slot3@149.165.158.49 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:06
slot4@149.165.158.49 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:07

Condor-worker_4 @ VM_india_openstack
slot1@149.165.158.55 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:04
slot2@149.165.158.55 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:05
slot3@149.165.158.55 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:06
slot4@149.165.158.55 LINUX       X86_64 Claimed   Busy     0.000 1999  0+00:00:07

               Total Owner Claimed Unclaimed Matched Preempting Backfill
X86_64/LINUX      20     0      20         0       0          0        0
       Total      20     0      20         0       0          0        0
```