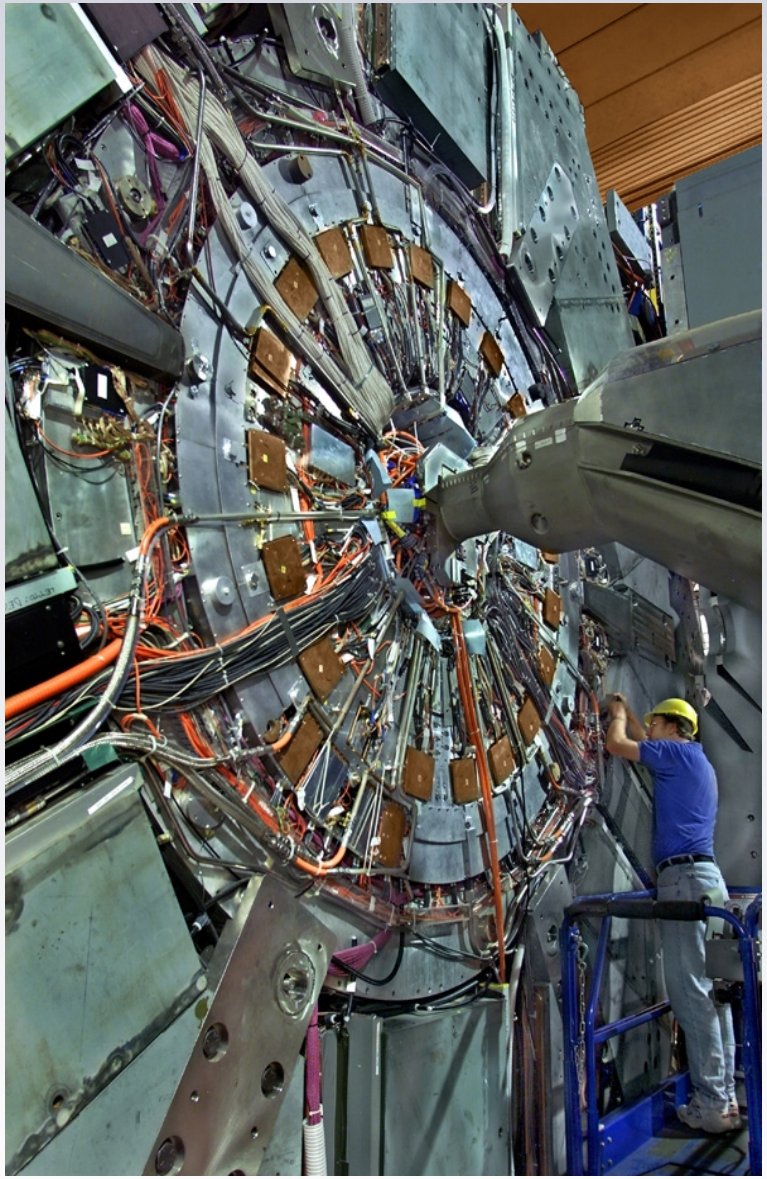
Simulation and user analysis of BaBar data in a distributed cloud

Colin Leavett-Brown, University of Victoria

A. Agarwal, University of Victoria
M. Anderson, University of Victoria
P. Armstrong, University of Victoria
A. Charbonneau, National Research Council
R. Desmarais, University of Victoria
K. Fransham, University of Victoria
I. Gable, University of Victoria
D. Harris, University of Victoria
R. Impey, National Research Council
M. Paterson, University of Victoria
W. Podaima, National Research Council
R. J. Sobie, University of Victoria
M. Vliett, University of Victoria

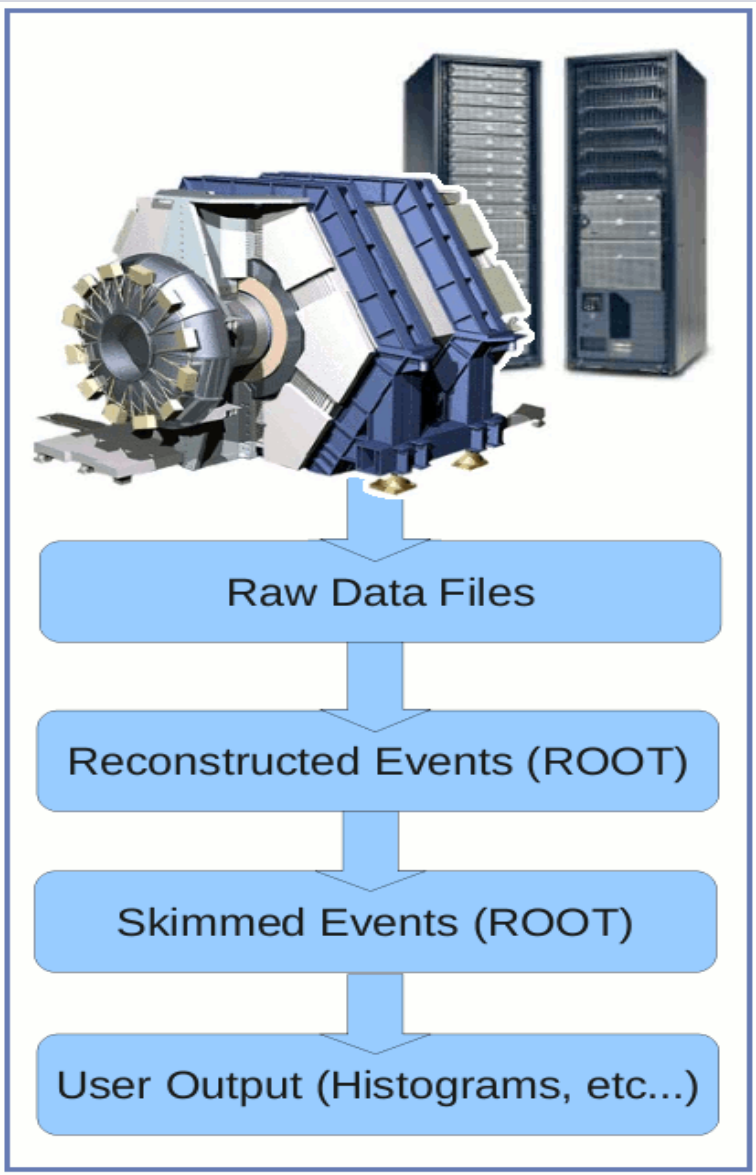
Overview

- About BaBar
  - The experiment
  - Analyzing the data
  - Why use clouds?
  - Analysis versus Simulation
- The Interactive System
- The Batch System
- Analysis with User Developed Images
- Simulation Production
- Components and References
- Summary
- Addenda



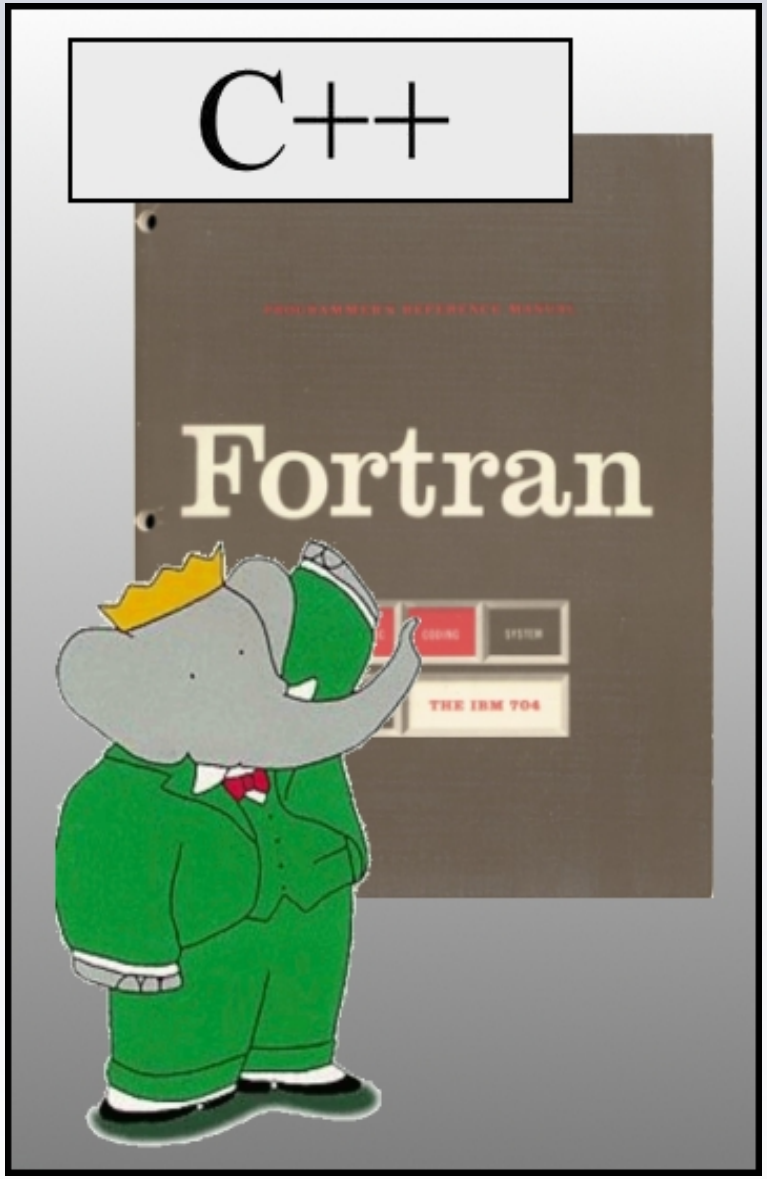
About BaBar

- HEP experiment based at the SLAC, Stanford, California.
- Recorded electron/positron particle collisions.
- After the Big Bang, where did all the anti-matter go?
- Experiment ran from 2000 to 2008.
- Collected ~1PB.



About BaBar

- ~1PB of simulated data in addition to the experimental data.
- BaBar analysis; a four step process:
  - Raw data collection, both experimental and simulated.
  - Pattern recognition on raw data to form tracks.
  - Track topology characterized and specific events selected.
  - User write code to examine the events and extract new physics.
- 9M lines of C++ and Fortran application code.



About BaBar

- Why use clouds?
  - Much analysis still to be done.
  - Code developed over the last 10 years.
  - Highly complex and difficult to port to new operating systems.
  - Diminishing resources, fewer people.
  - Application environment encapsulated through virtualization; install once, run many times, anywhere.

About BaBar

Analysis:

- Mostly batch processing
- Embarrassingly parallel
- BaBar application code
- Validation required
- User code/development
- Substantial amount of input
- Small amount of output
- Output belongs to user

Simulation:

- Entirely batch processing
- Embarrassingly parallel
- BaBar application code
- Validation required
- No User code
- Small amount of input
- Substantial amount of output
- Output goes to collaboration

About BaBar

Analysis:

- Mostly batch processing
- Embarrassingly parallel
- BaBar application code
- Validation required
- User code/development
- Substantial amount of input
- Small amount of output
- Output belongs to user

Simulation:

- Entirely batch processing
- Embarrassingly parallel
- BaBar application code
- Validation required
- No User code
- Small amount of input
- Substantial amount of output
- Output goes to collaboration

About BaBar

Analysis:

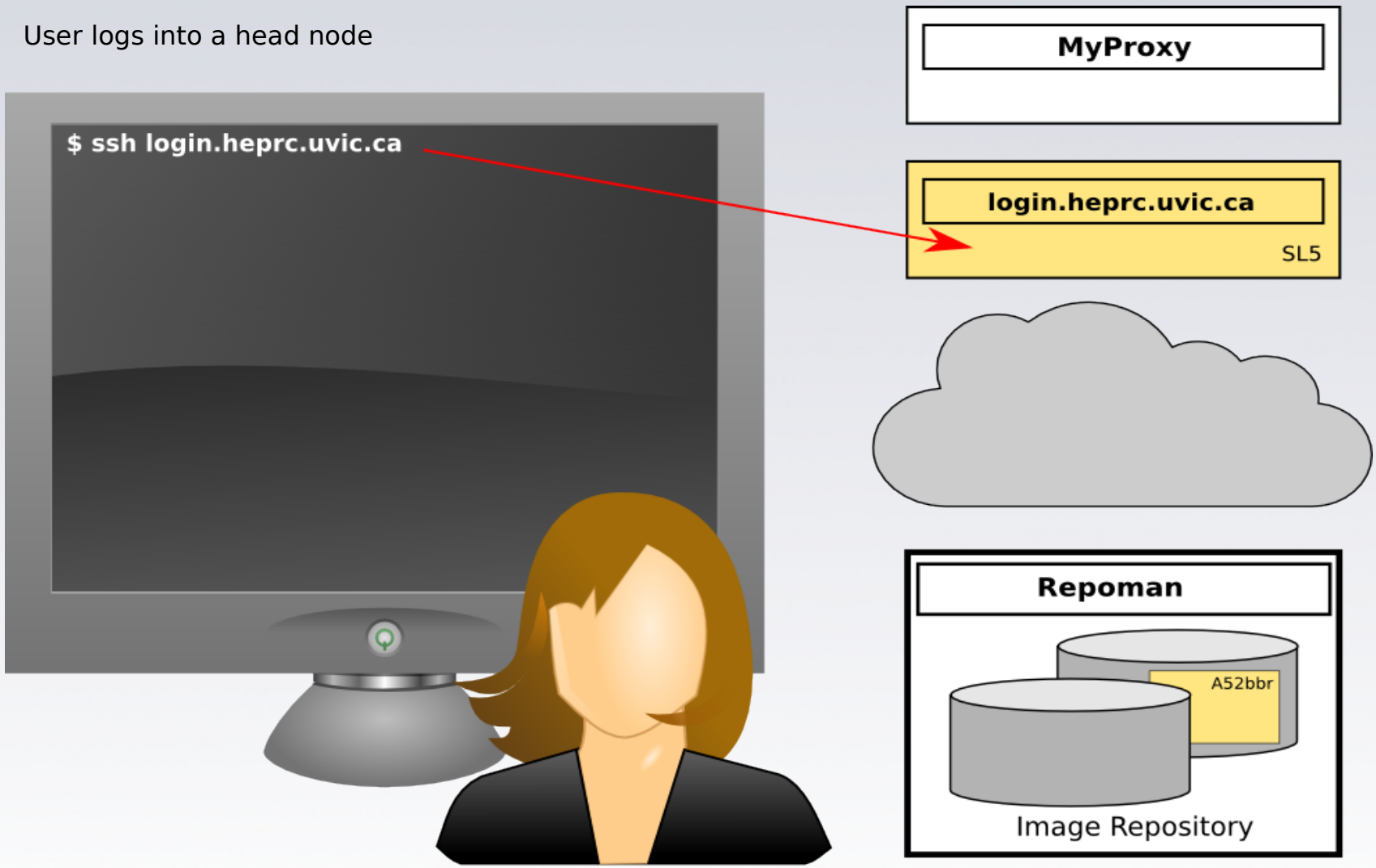
- Mostly batch processing
- Embarrassingly parallel
- BaBar application code
- Validation required
- User code/development
- Substantial amount of input
- Small amount of output
- Output belongs to user

Simulation:

- Entirely batch processing
- Embarrassingly parallel
- BaBar application code
- Validation required
- No User code
- Small amount of input
- Substantial amount of output
- Output goes to collaboration

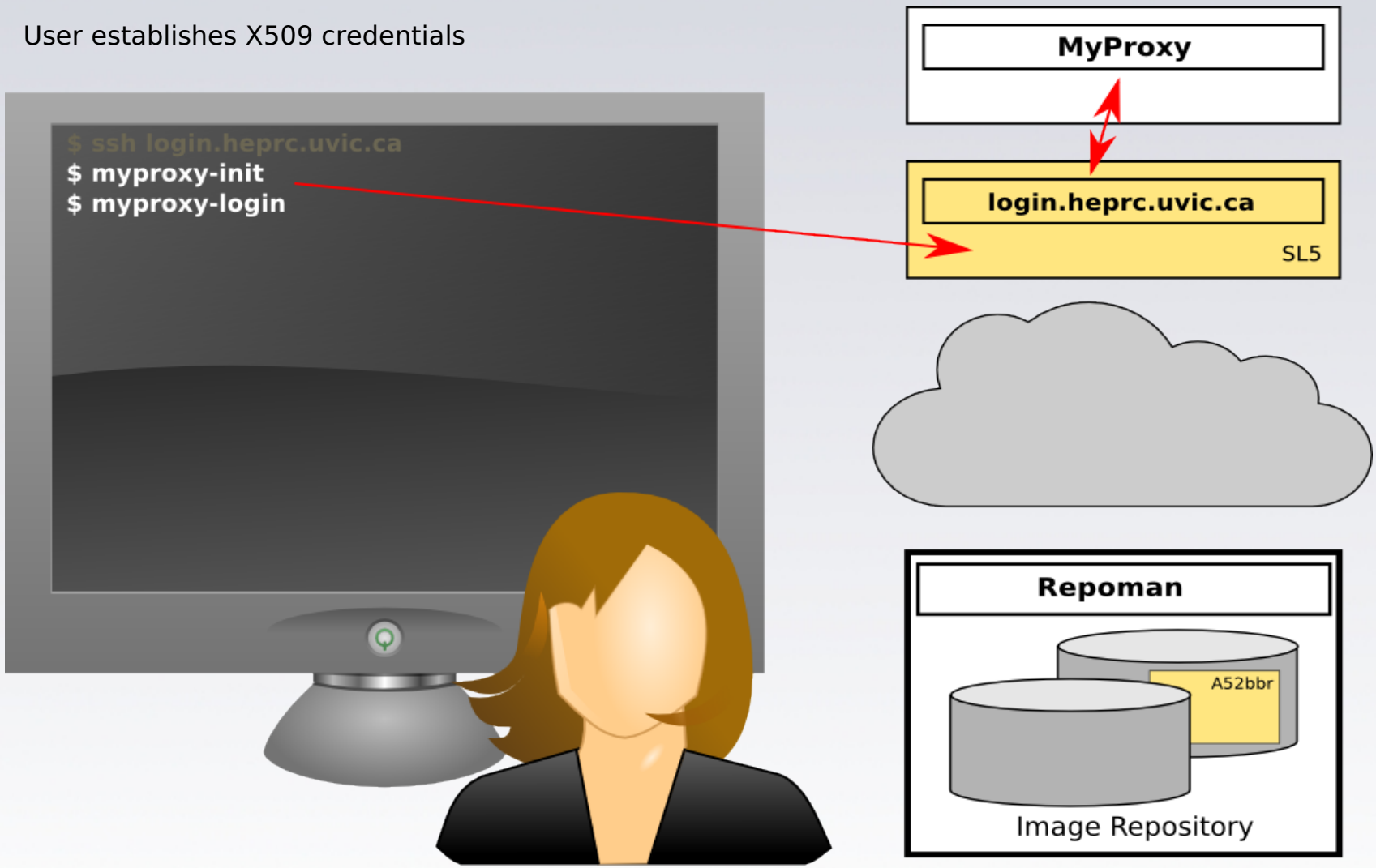
The Interactive System

User logs into a head node



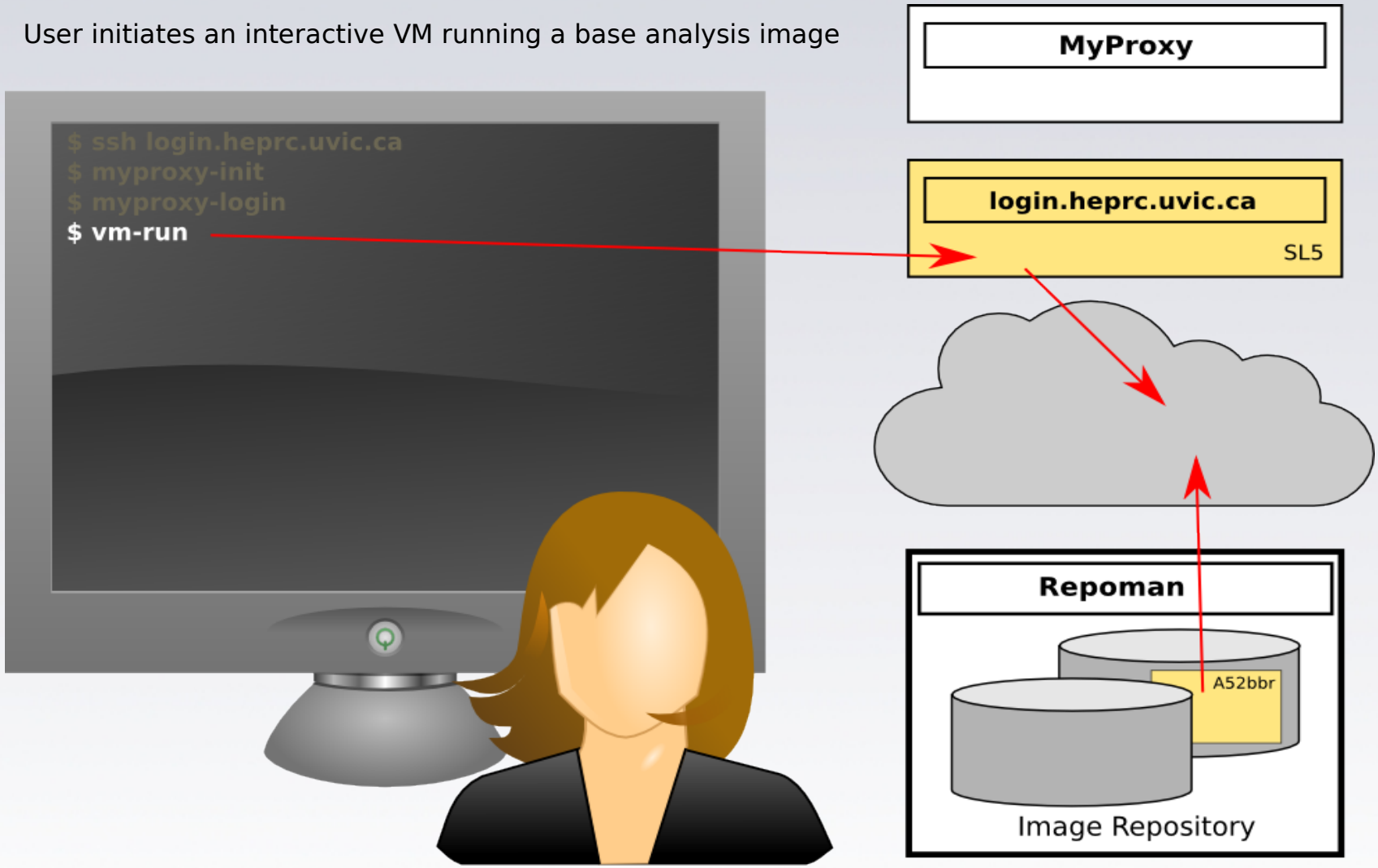
The Interactive System

User establishes X509 credentials



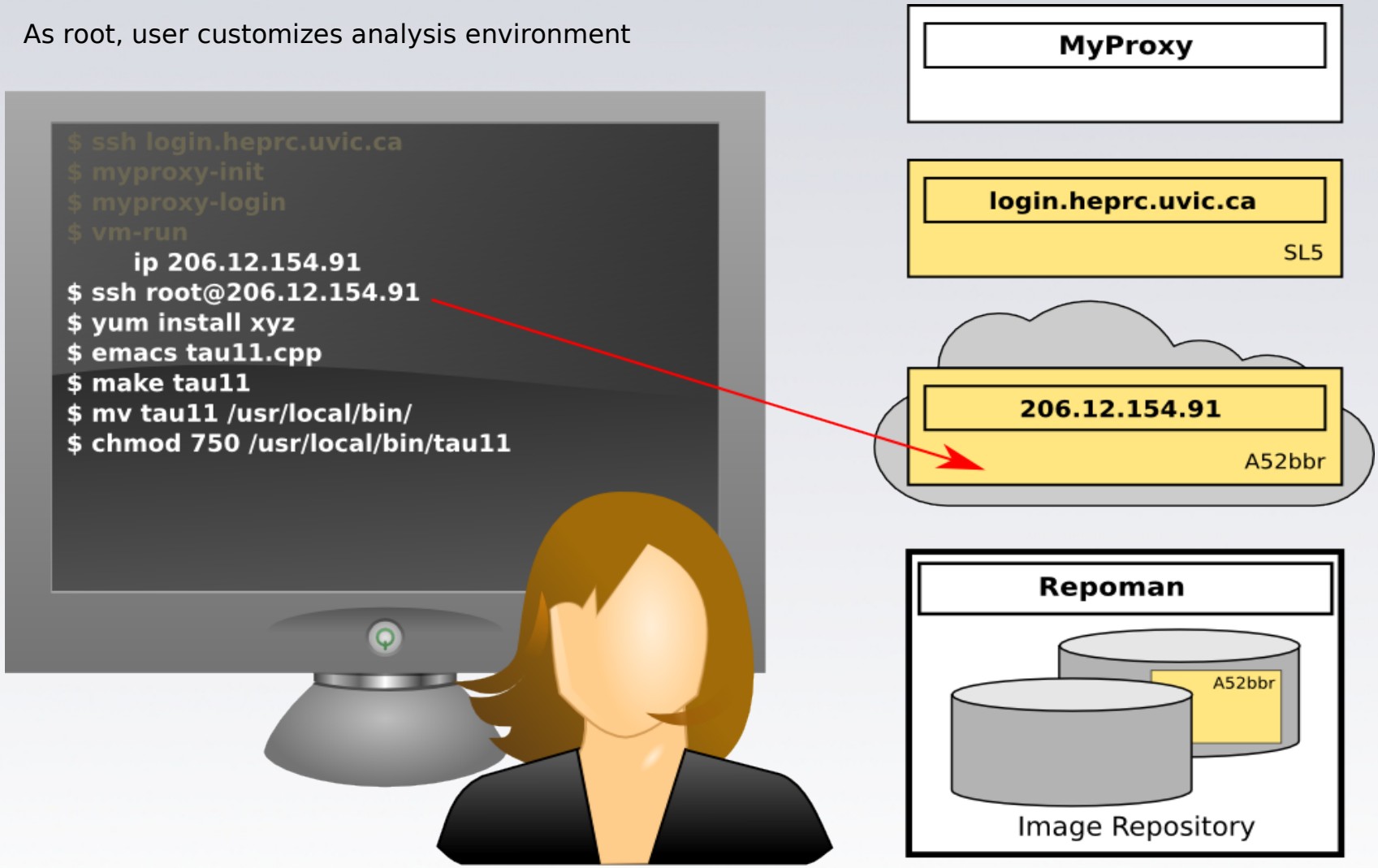
The Interactive System

User initiates an interactive VM running a base analysis image



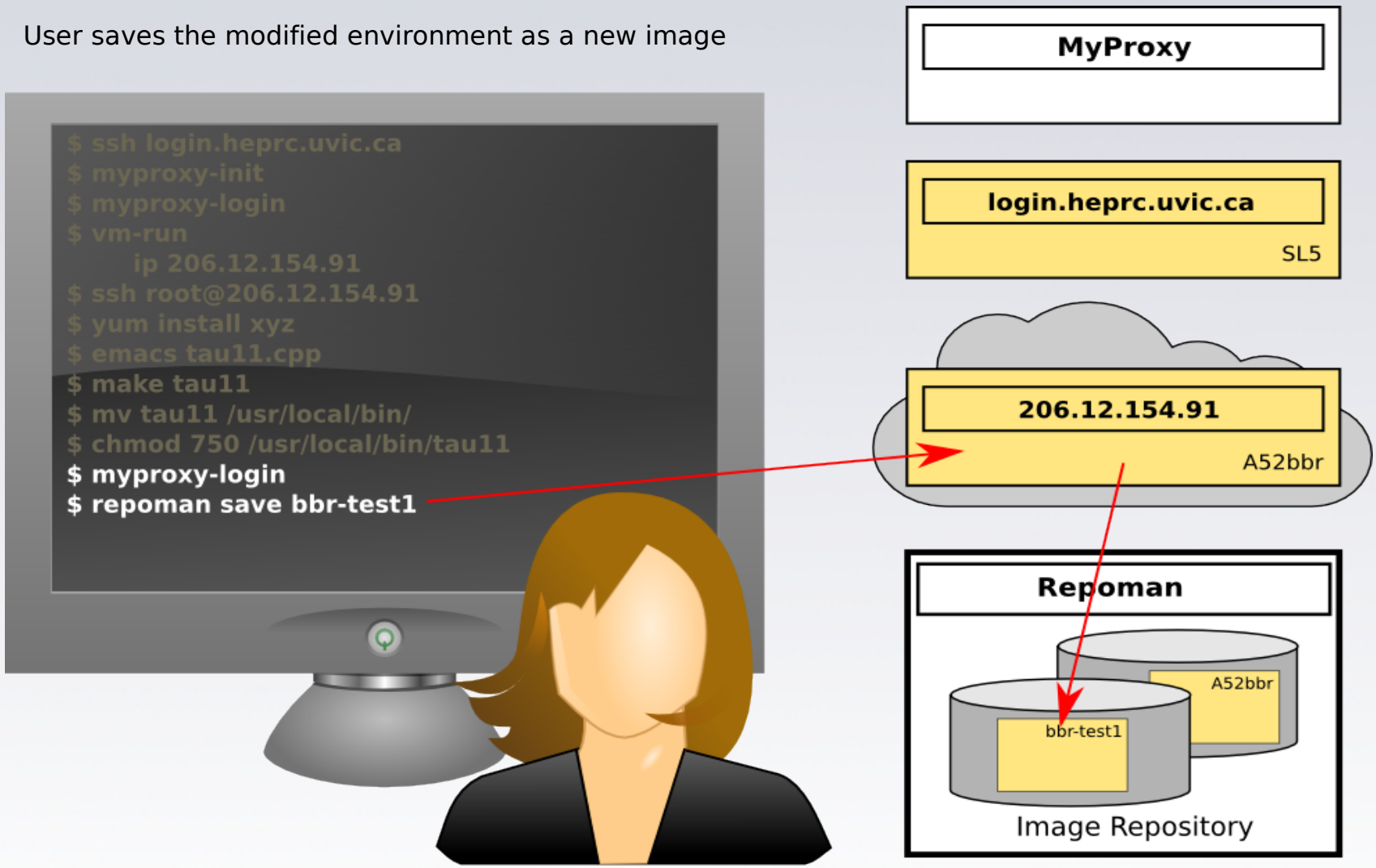
The Interactive System

As root, user customizes analysis environment



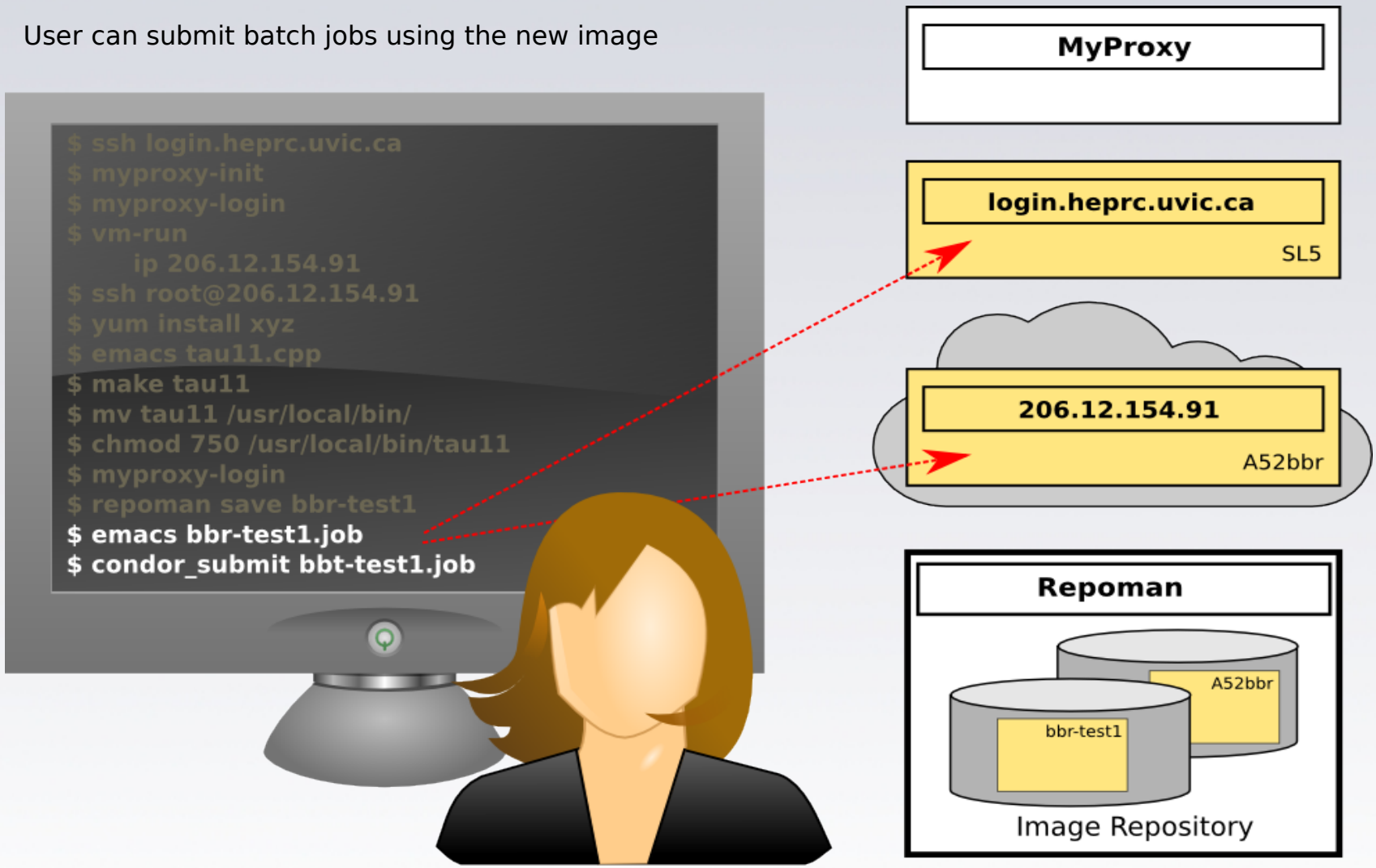
The Interactive System

User saves the modified environment as a new image



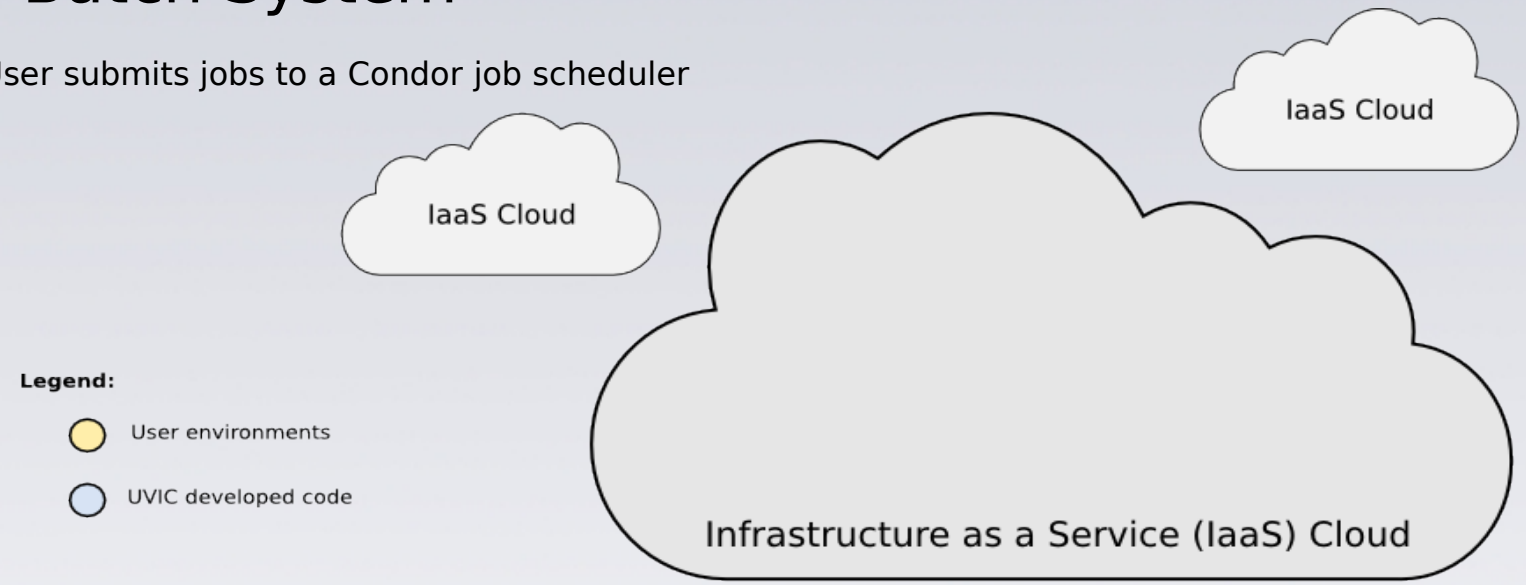
The Interactive System

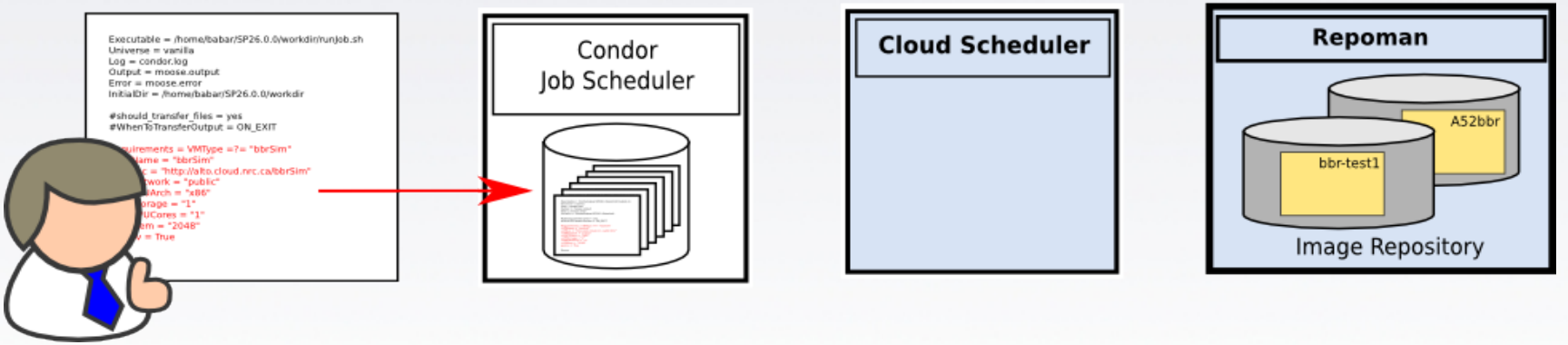
User can submit batch jobs using the new image



The Batch System

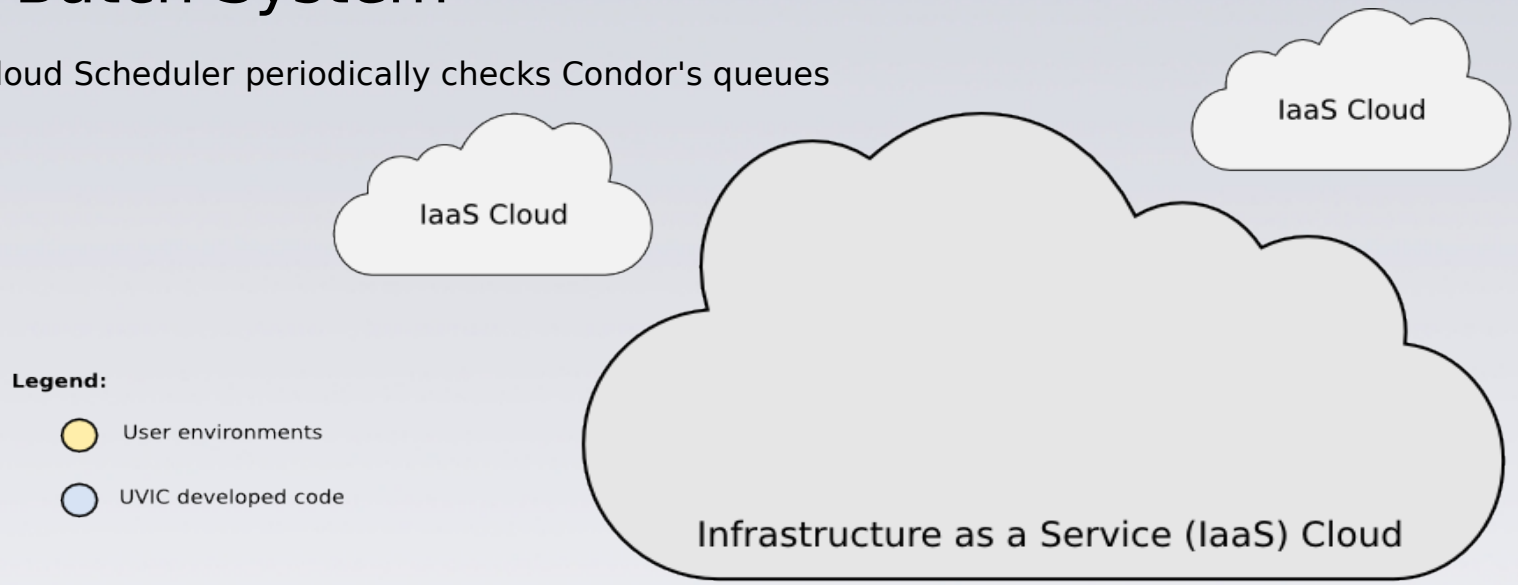
User submits jobs to a Condor job scheduler

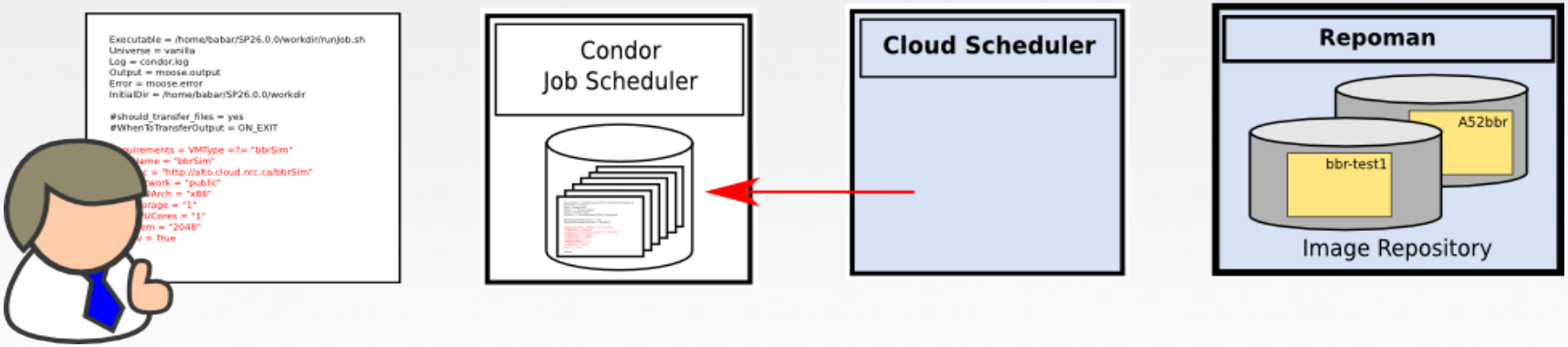




The Batch System

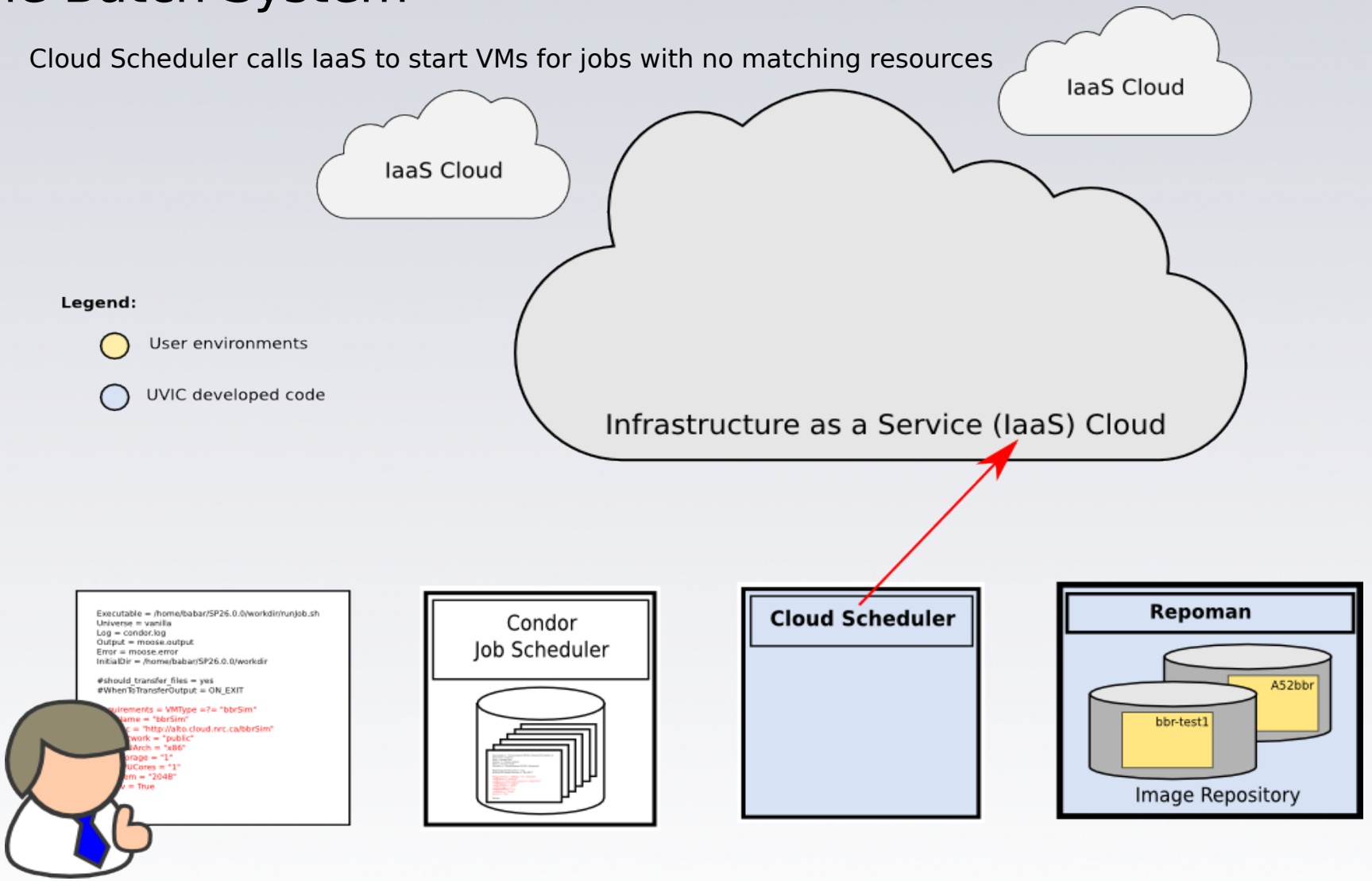
Cloud Scheduler periodically checks Condor's queues





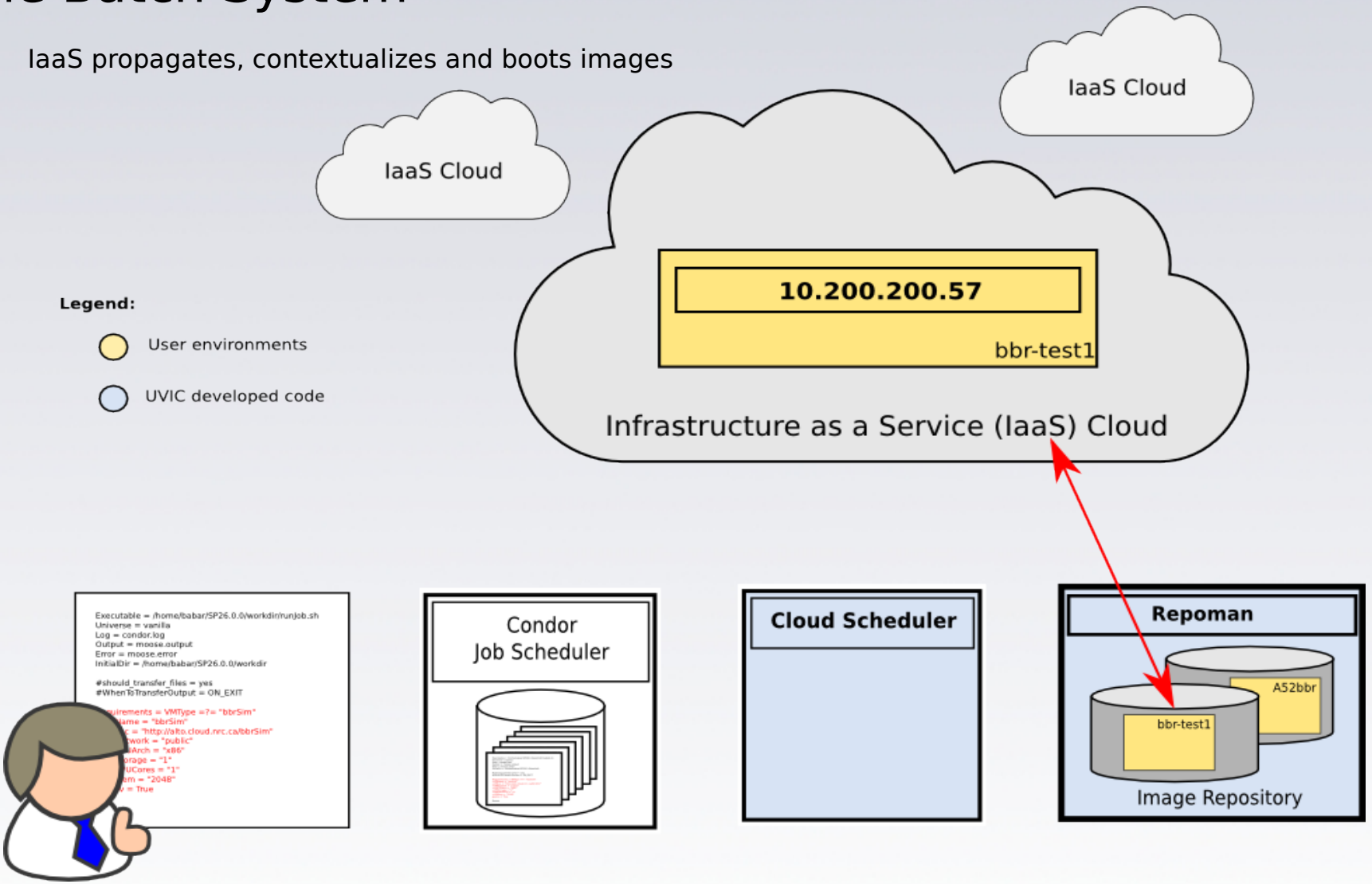
The Batch System

Cloud Scheduler calls IaaS to start VMs for jobs with no matching resources



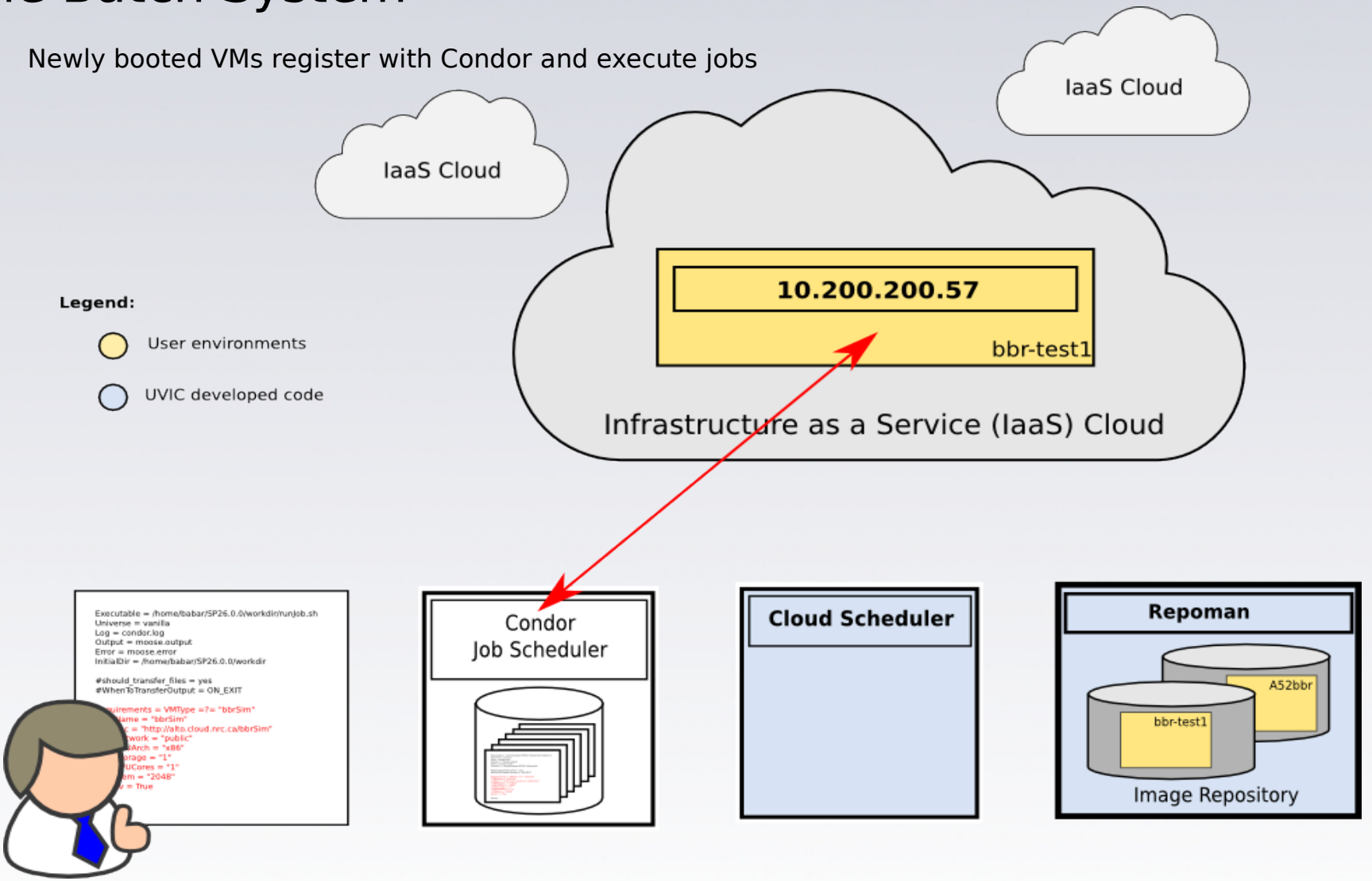
The Batch System

IaaS propagates, contextualizes and boots images



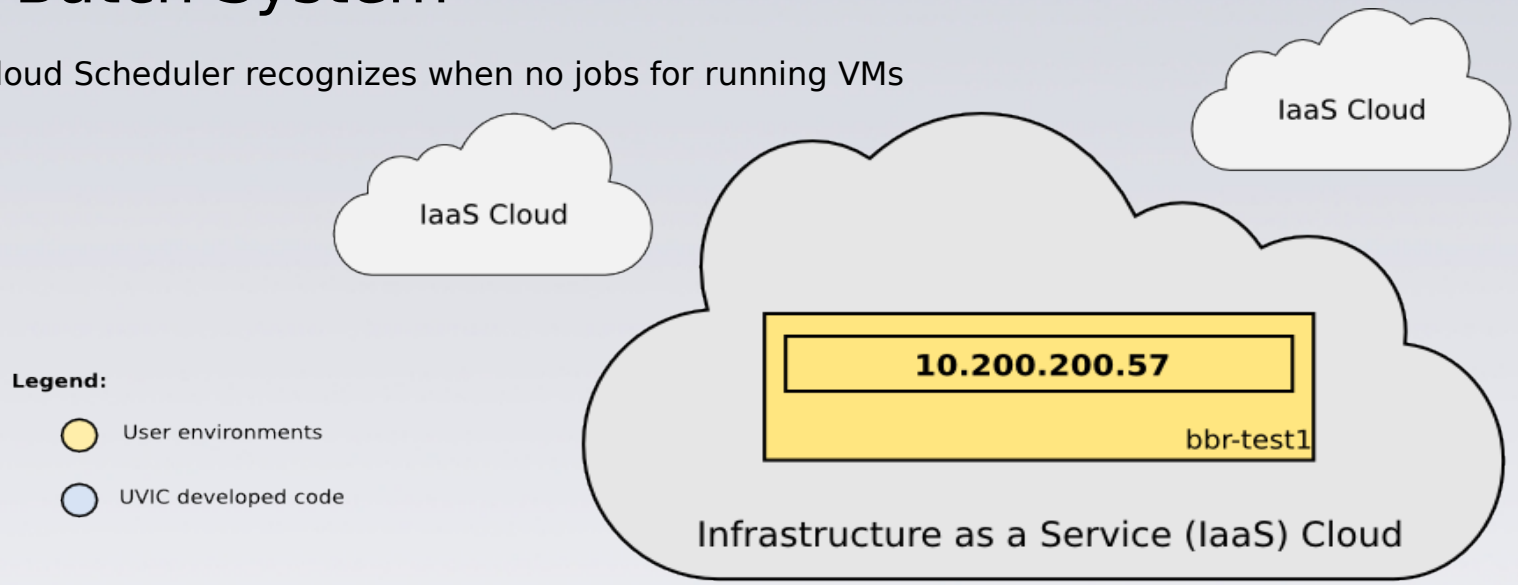
The Batch System

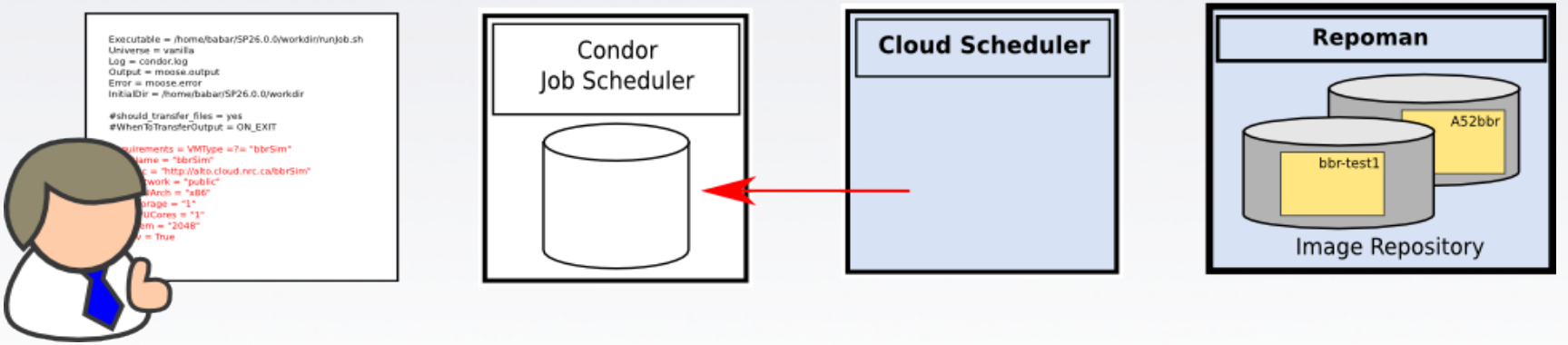
Newly booted VMs register with Condor and execute jobs



The Batch System

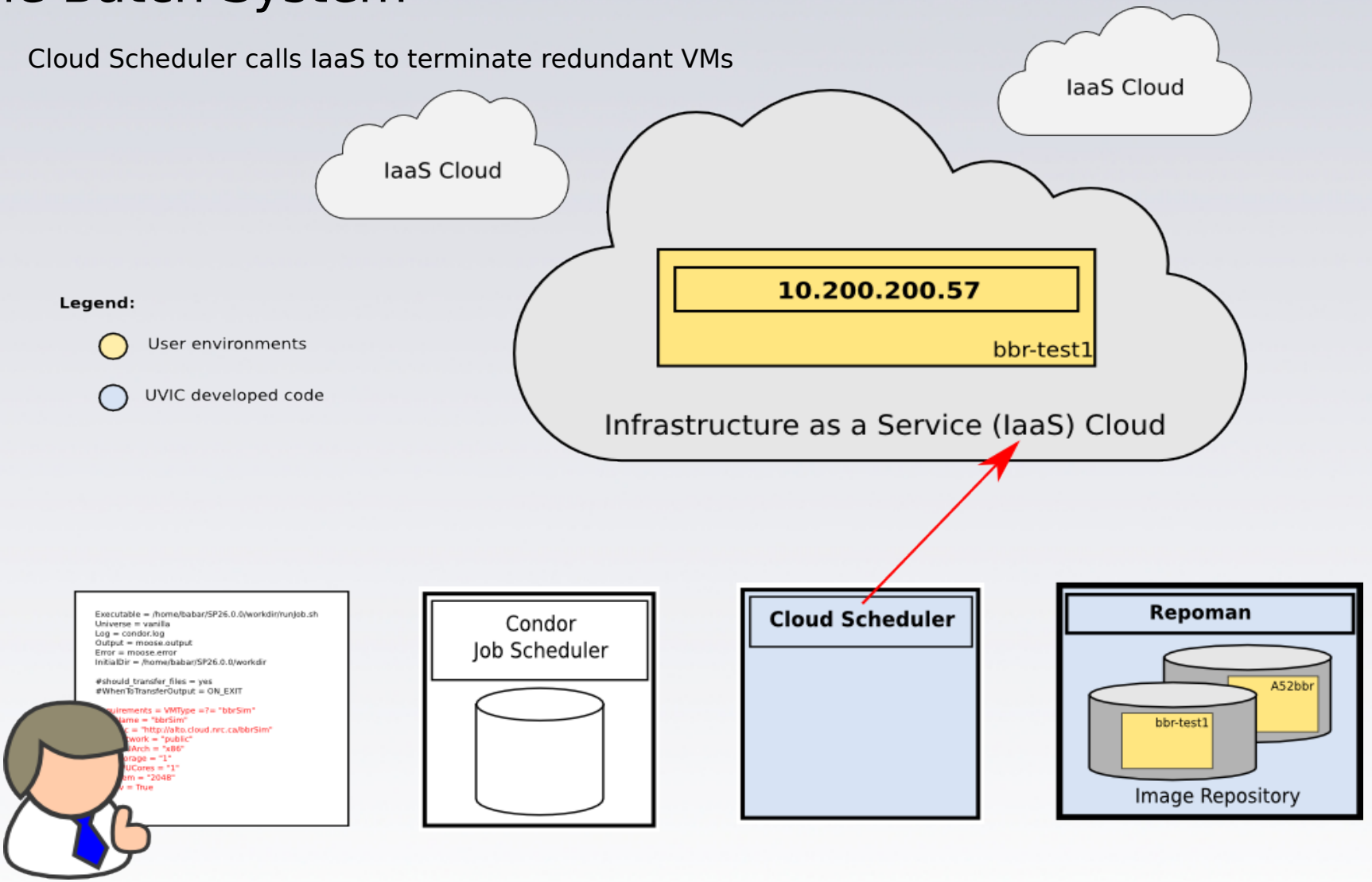
Cloud Scheduler recognizes when no jobs for running VMs





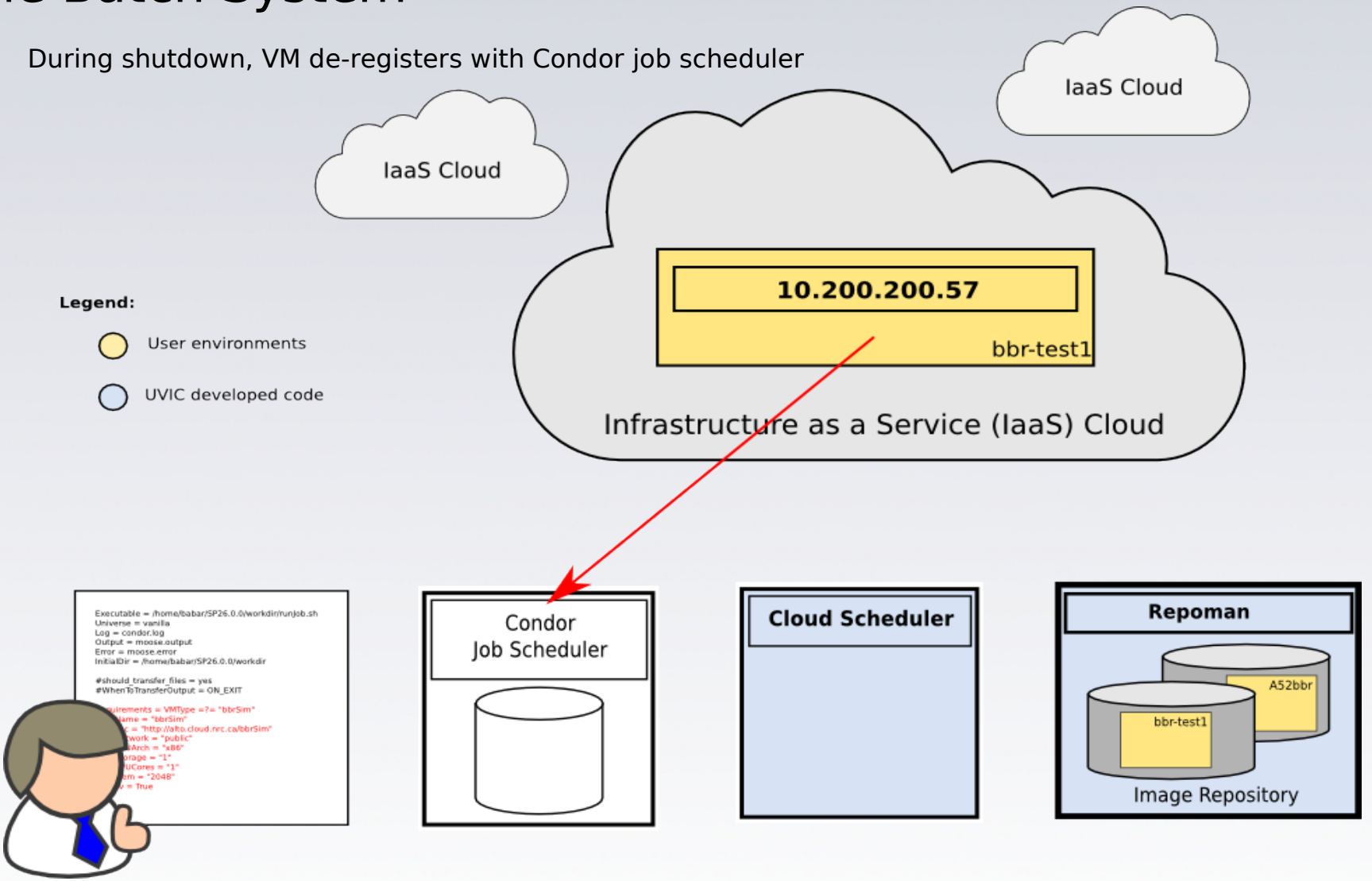
The Batch System

Cloud Scheduler calls IaaS to terminate redundant VMs



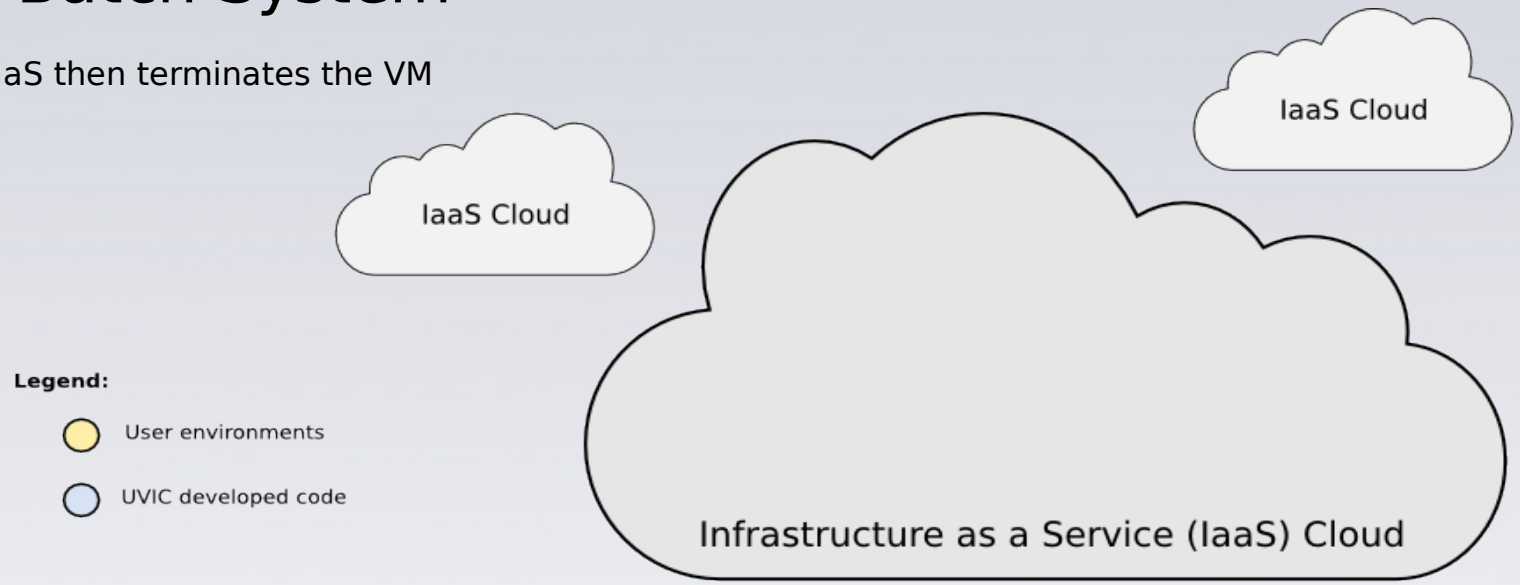
The Batch System

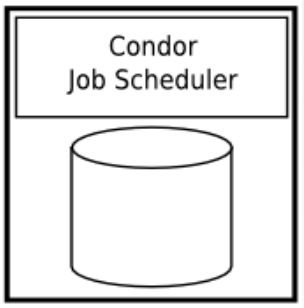
During shutdown, VM de-registers with Condor job scheduler

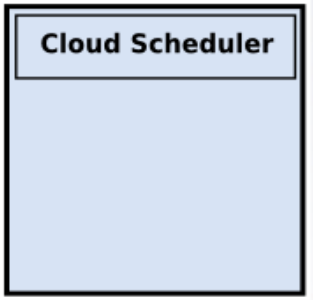


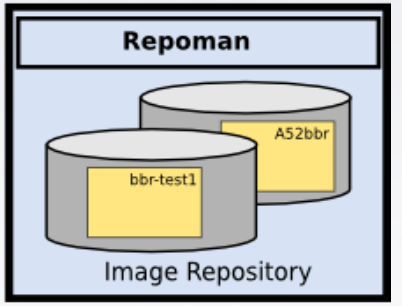
The Batch System

IaaS then terminates the VM

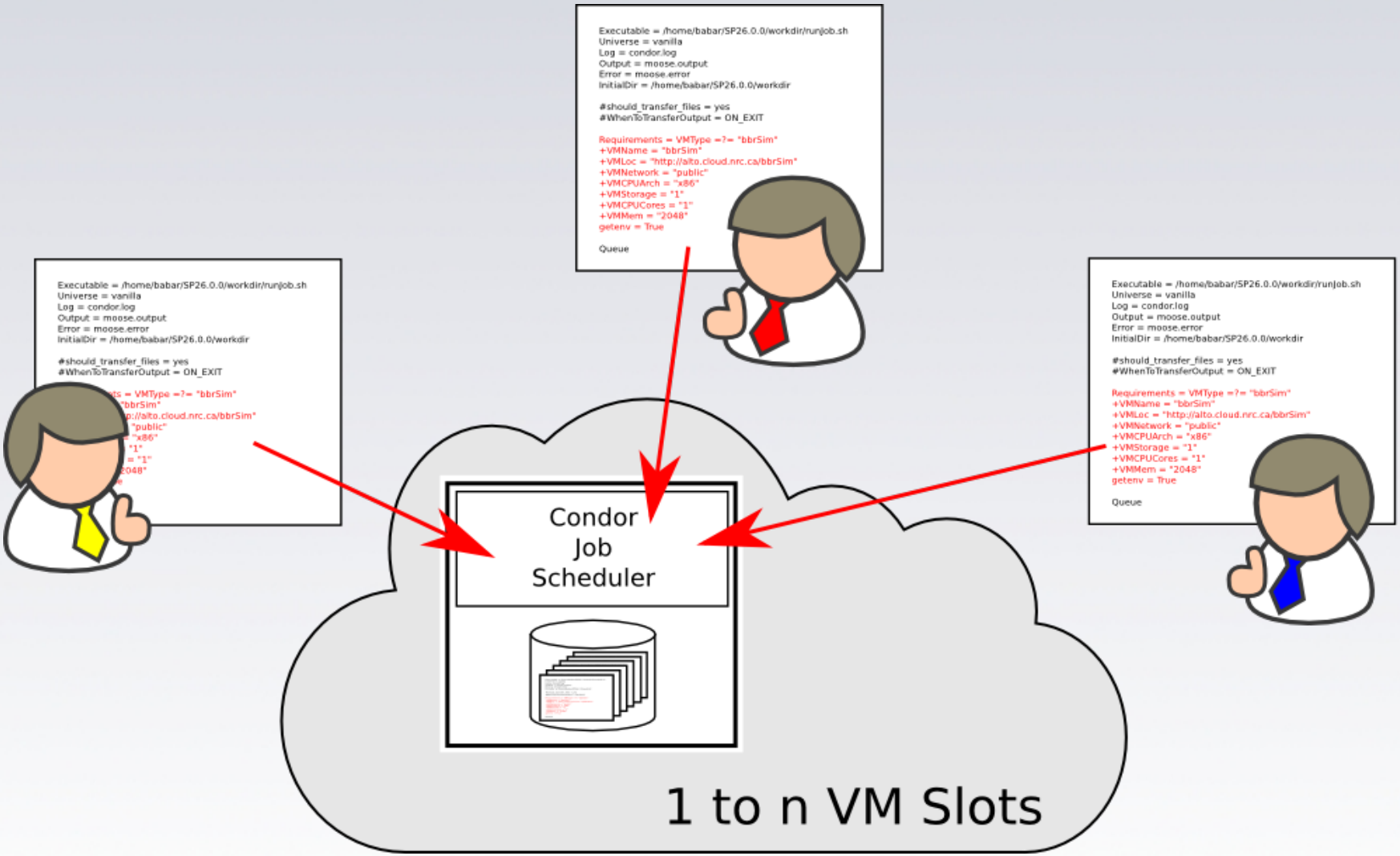








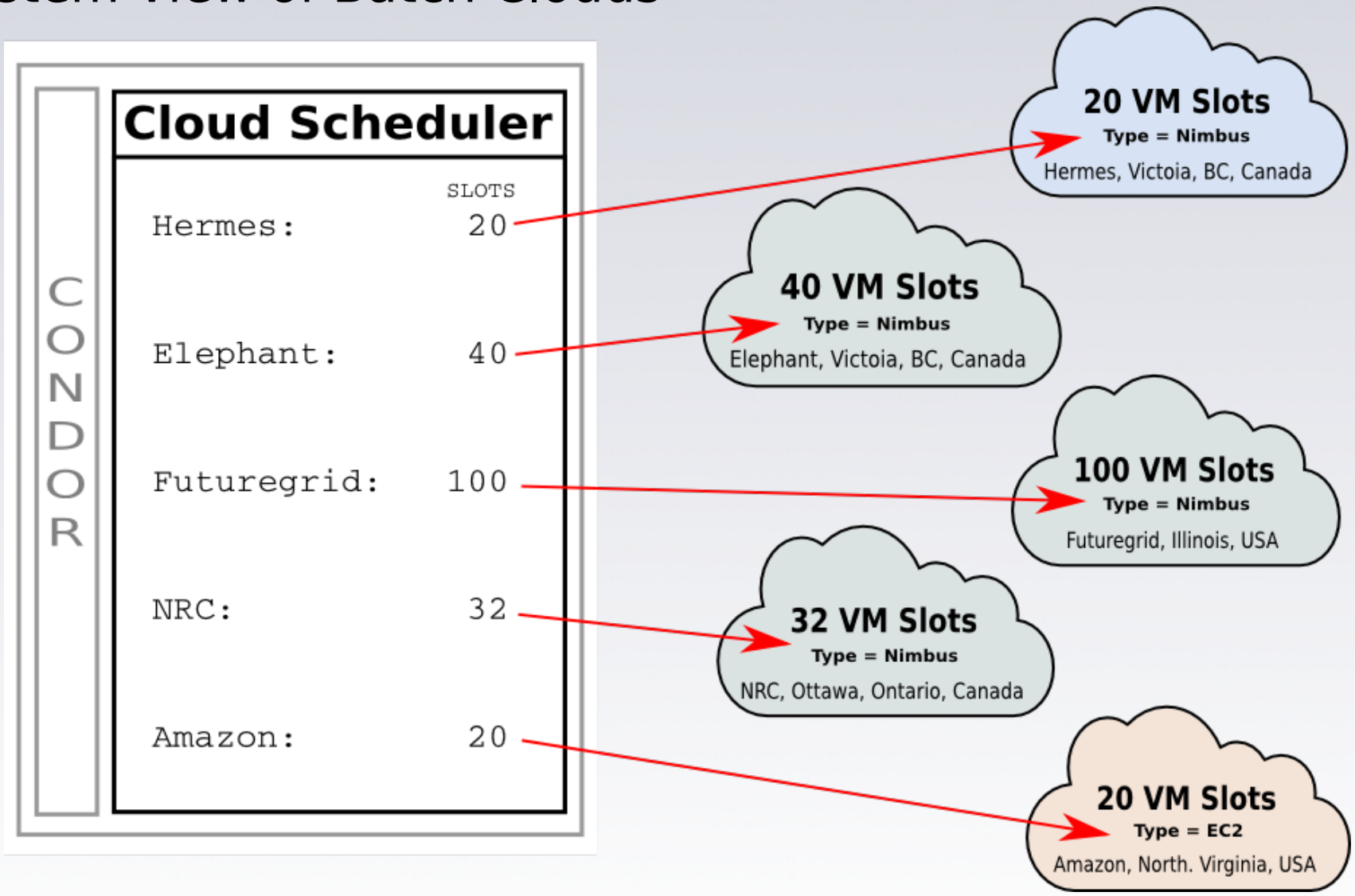
User View of the Batch System



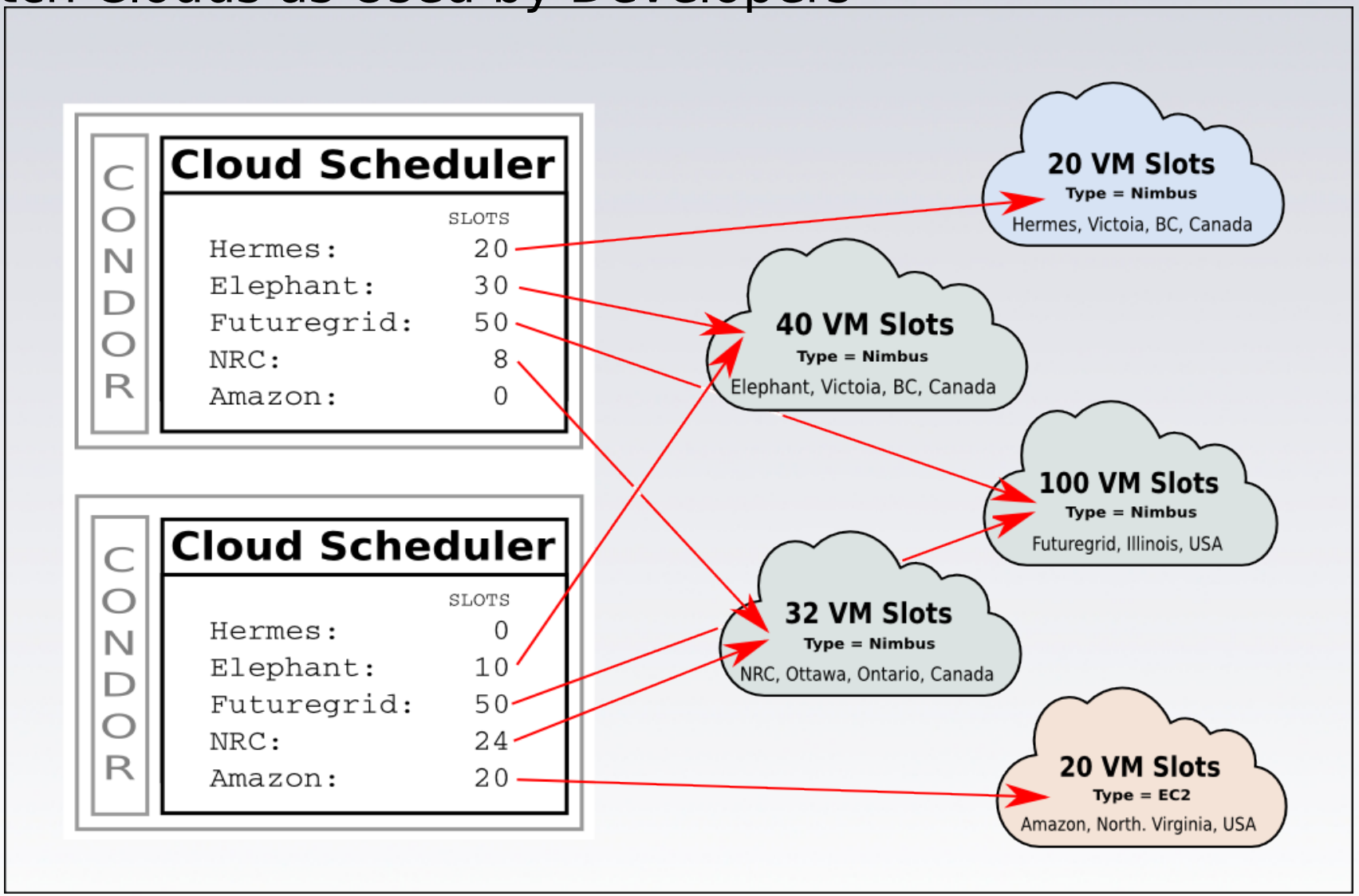
Sample Condor Job File (red text required for batch clouds/Cloud Scheduler)

```
Universe                =       vanilla
Log                     =       SP-3429-Tau11-Run2-R24a3-3.l1341
Output                  =       SP-3429-Tau11-Run2-R24a3-3.o1341
Error                   =       SP-3429-Tau11-Run2-R24a3-3.e1341
Input                   =       a52.tcl
should_transfer_files   =       YES
when_to_transfer_output =       ON_EXIT
environment             =       CLUSTERID=1341


Requirements = VMType =?= "rsobie/rjs1"
+VMLoc           =       "http://elephant01.heprc.uvic.ca/api/images/raw/rsobie/rjs1"
+VMCPUArch       =       "x86"
+VMStorage       =       "1"
+VMCPUCores      =       "1"
+VMMem           =       "2555"
+VMAMI           =       "ami-64ea1a0d"
+VMInstanceType  =       "m1.small"
+VMJobPerCore    =       True

getenv          =       True

Queue
```

System View of Batch Clouds

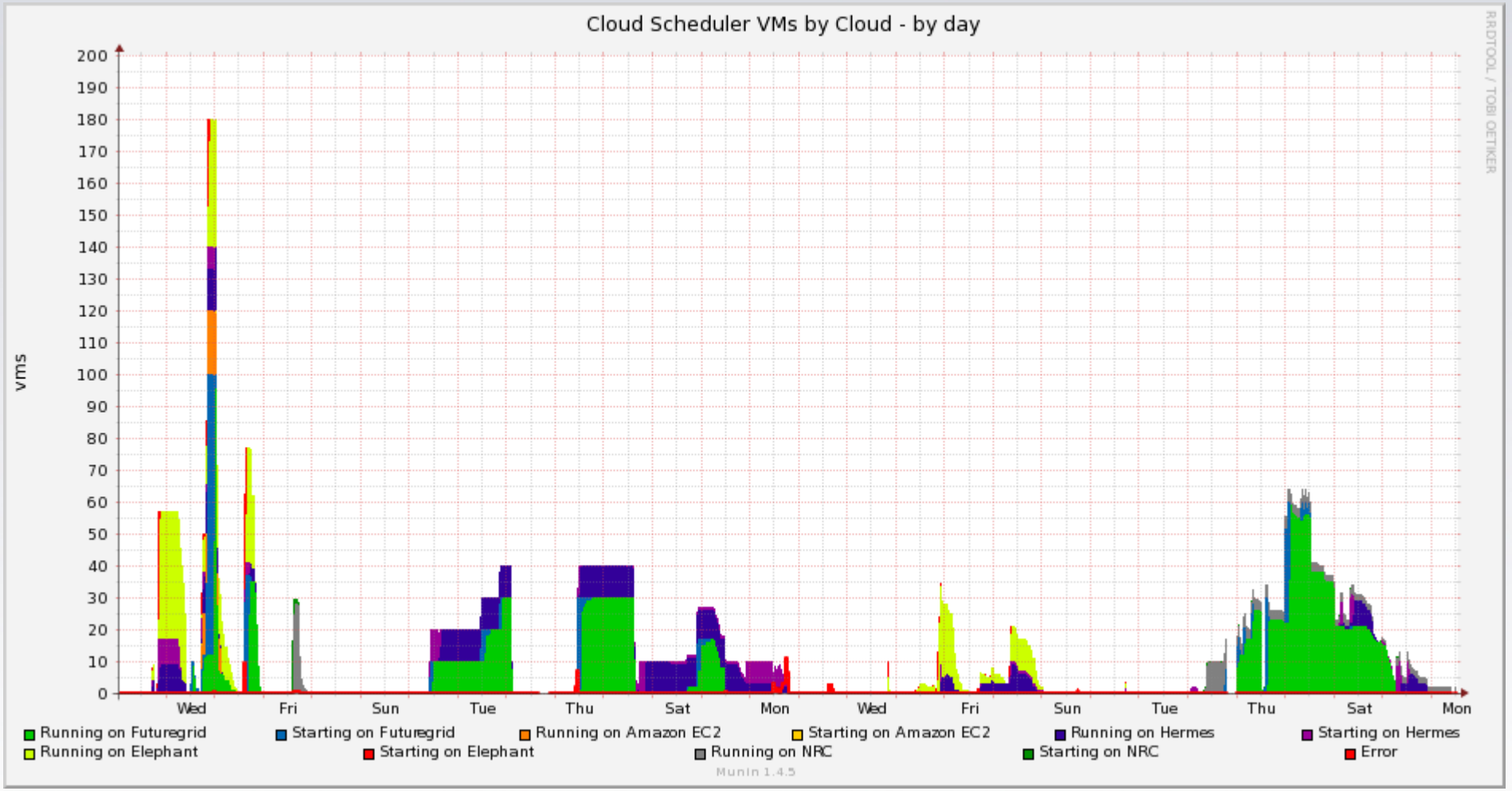


Batch Clouds as Used by Developers



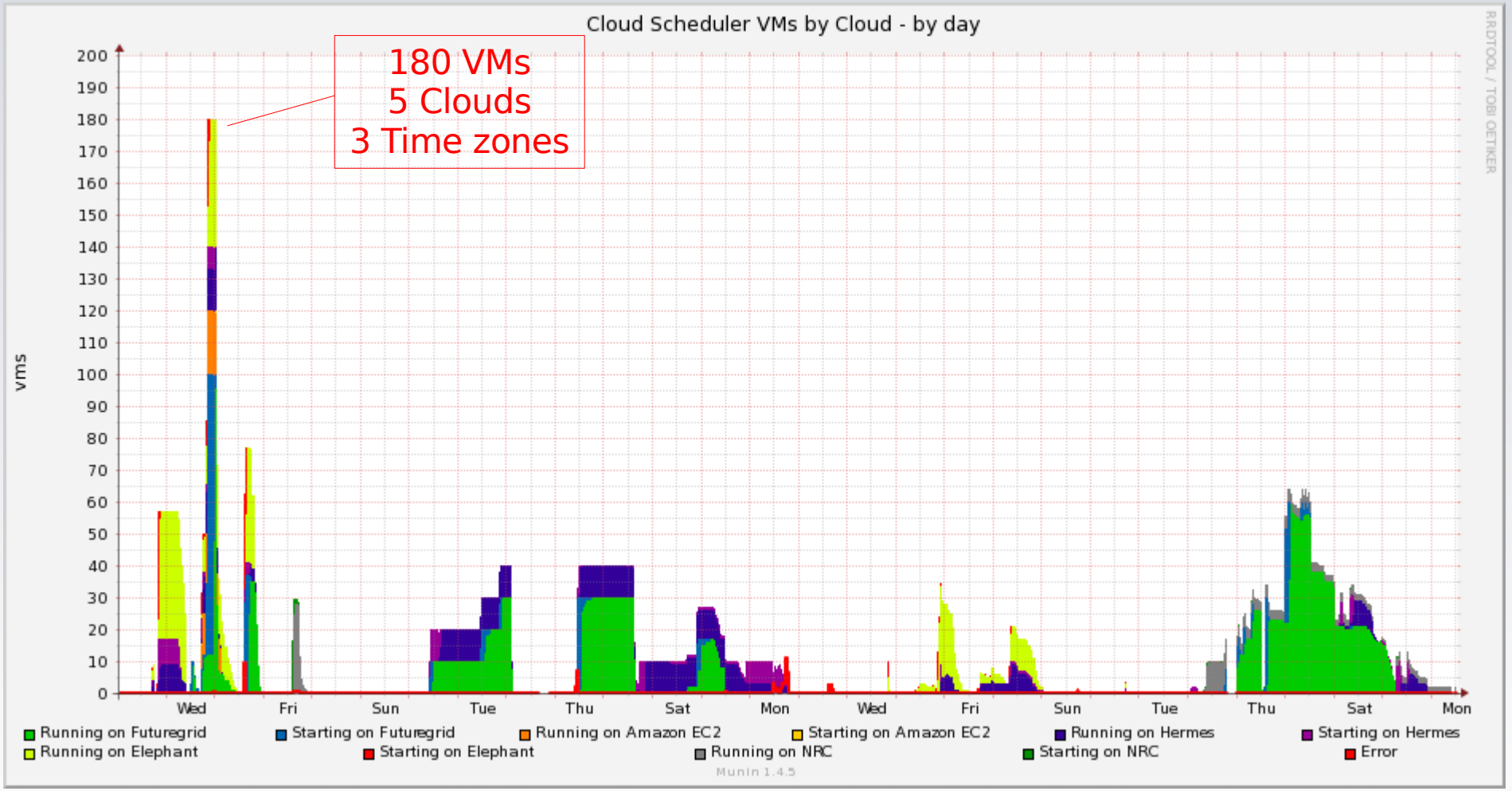
Analysis with User Developed Images

- Batch usage during the month of February.



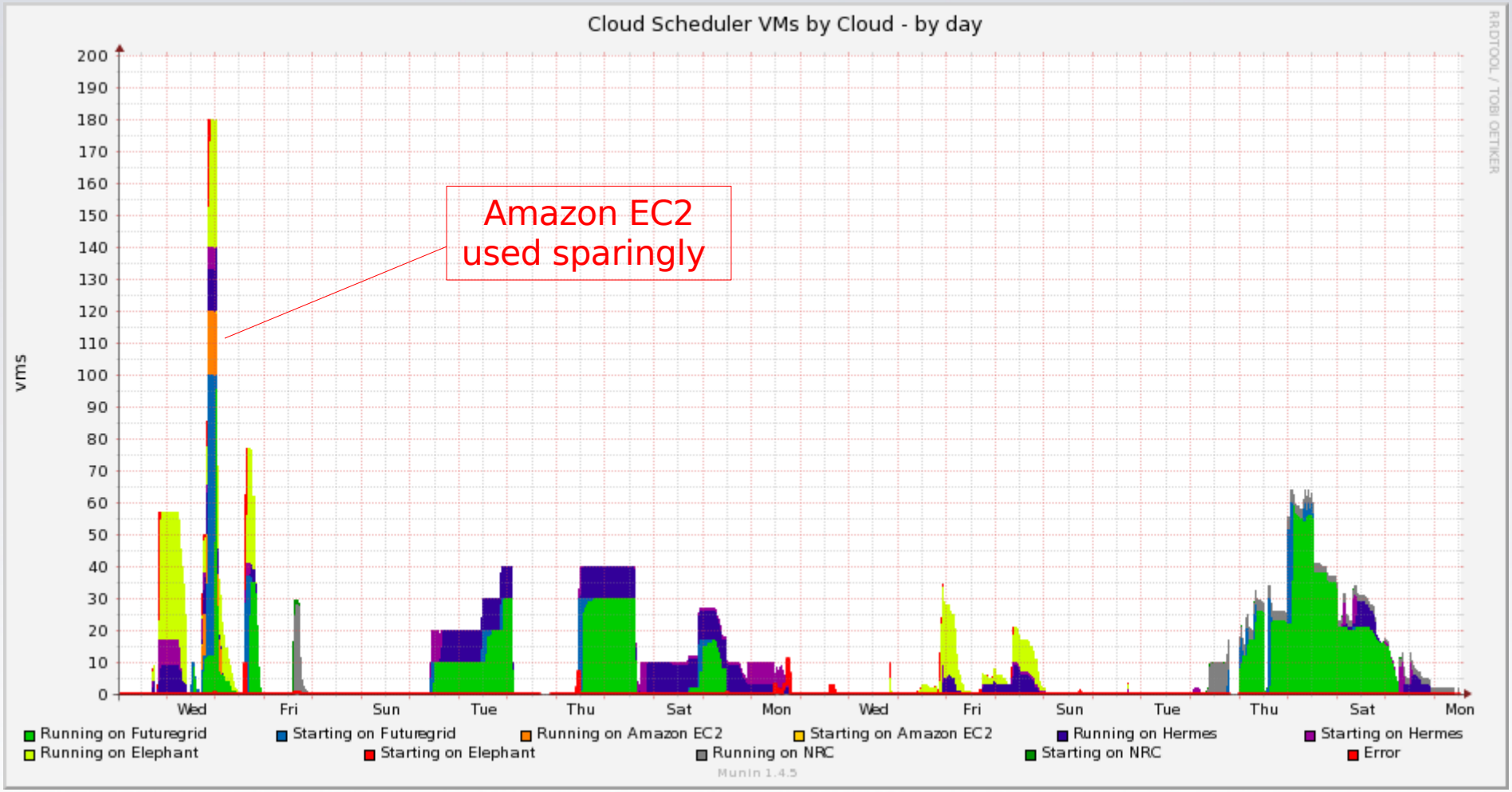
Analysis with User Developed Images

- Batch usage during the month of February.



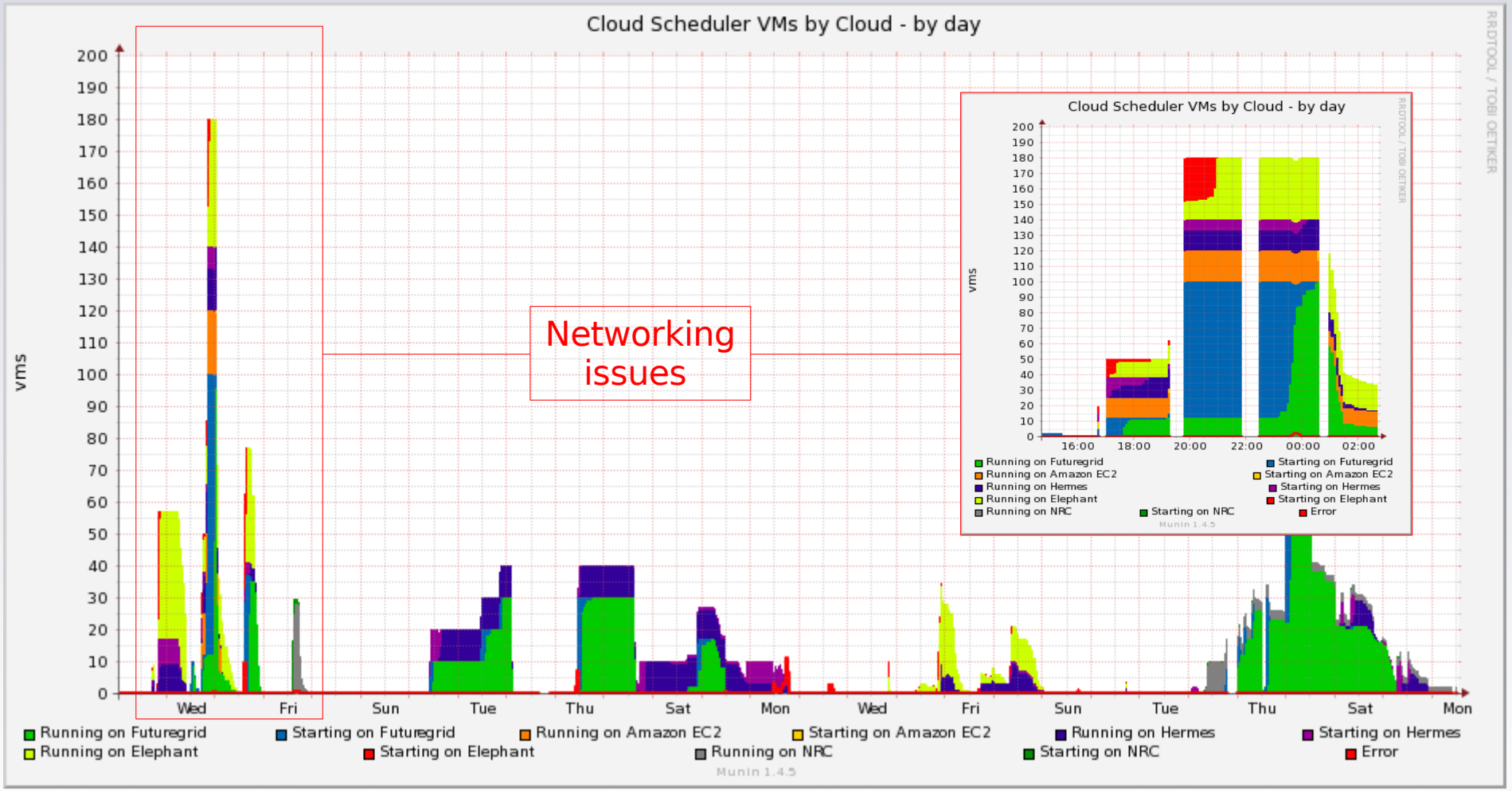
Analysis with User Developed Images

- Batch usage during the month of February.



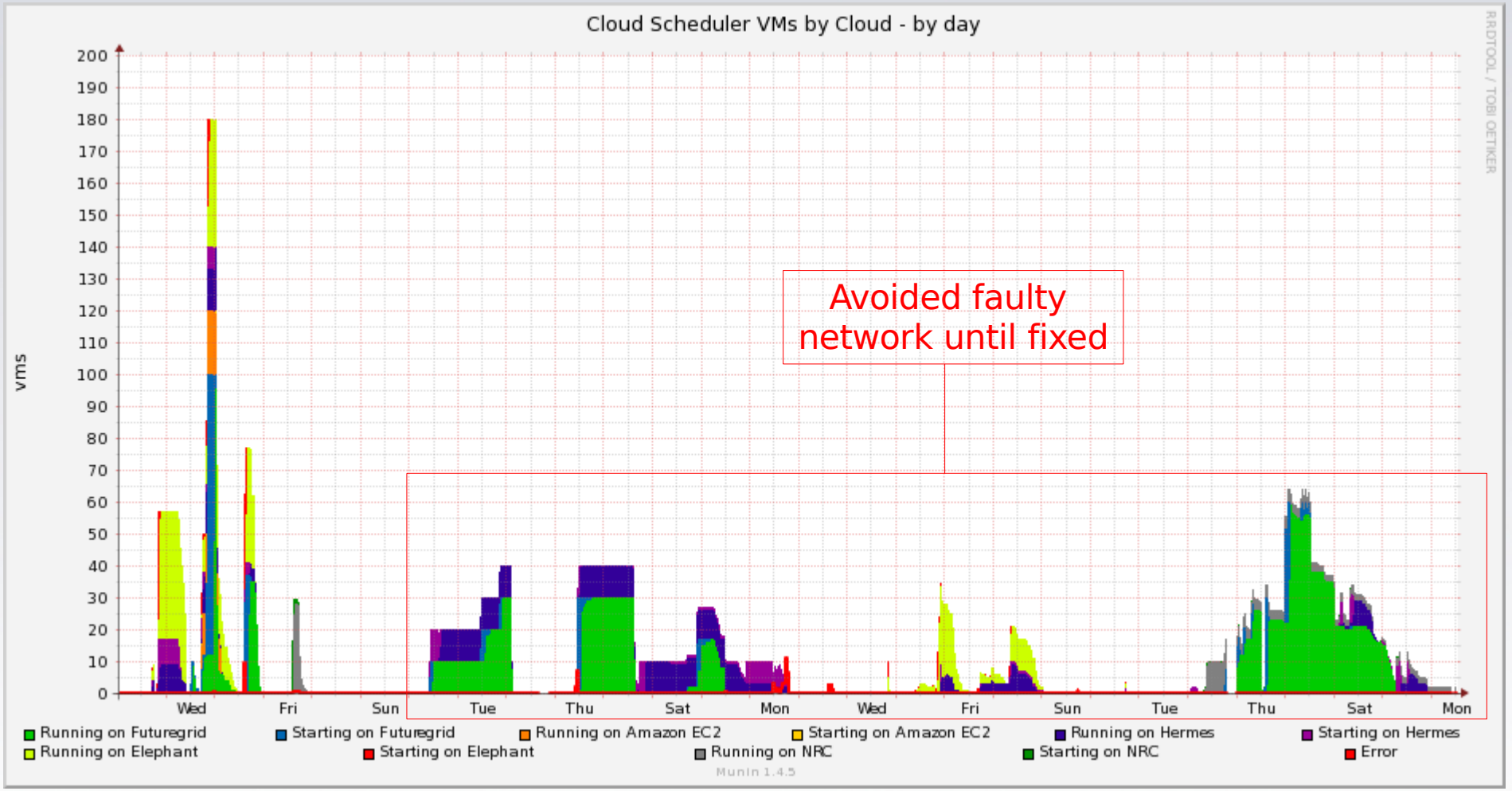
Analysis with User Developed Images

- Batch usage during the month of February.



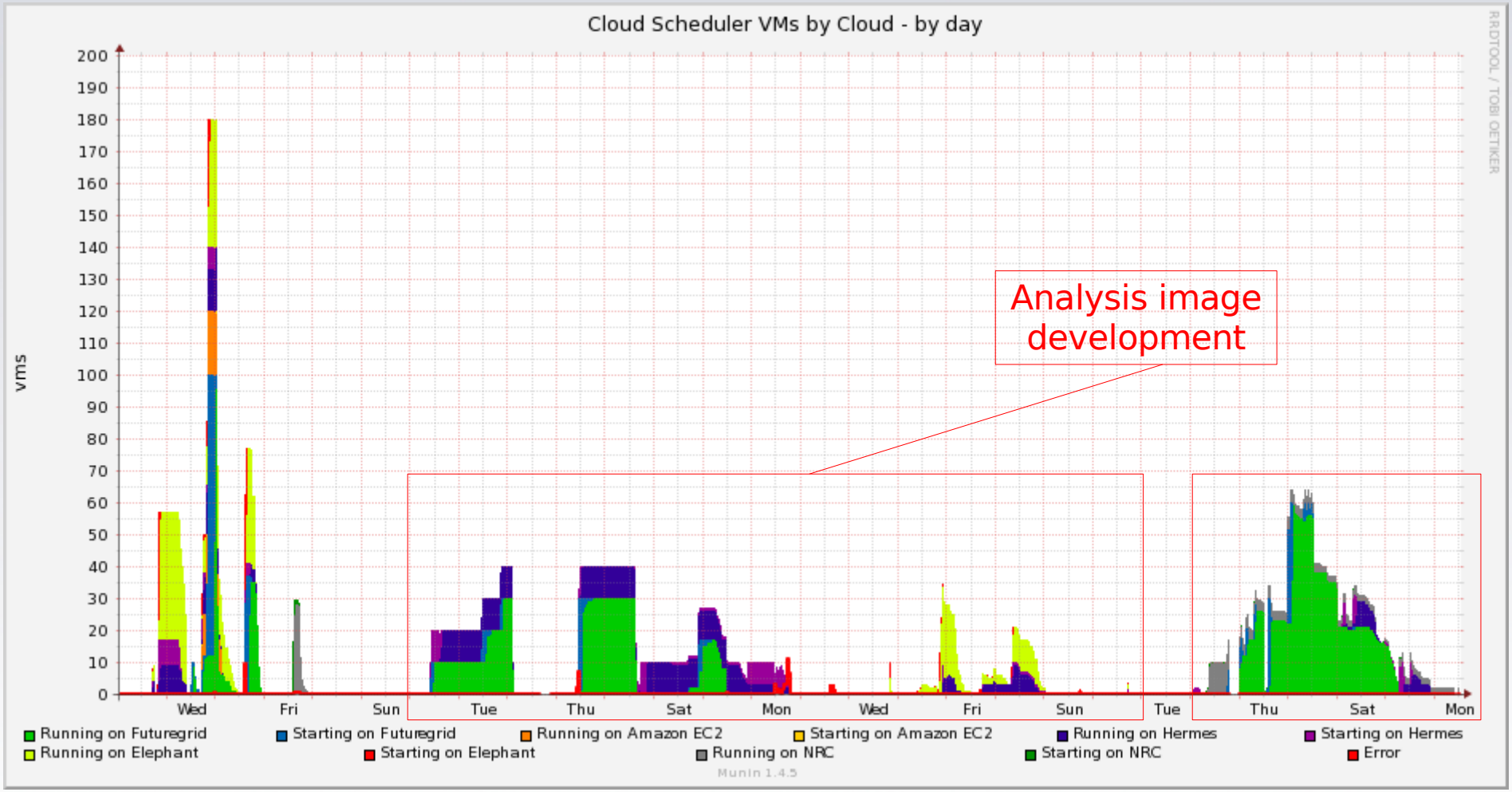
Analysis with User Developed Images

- Batch usage during the month of February.



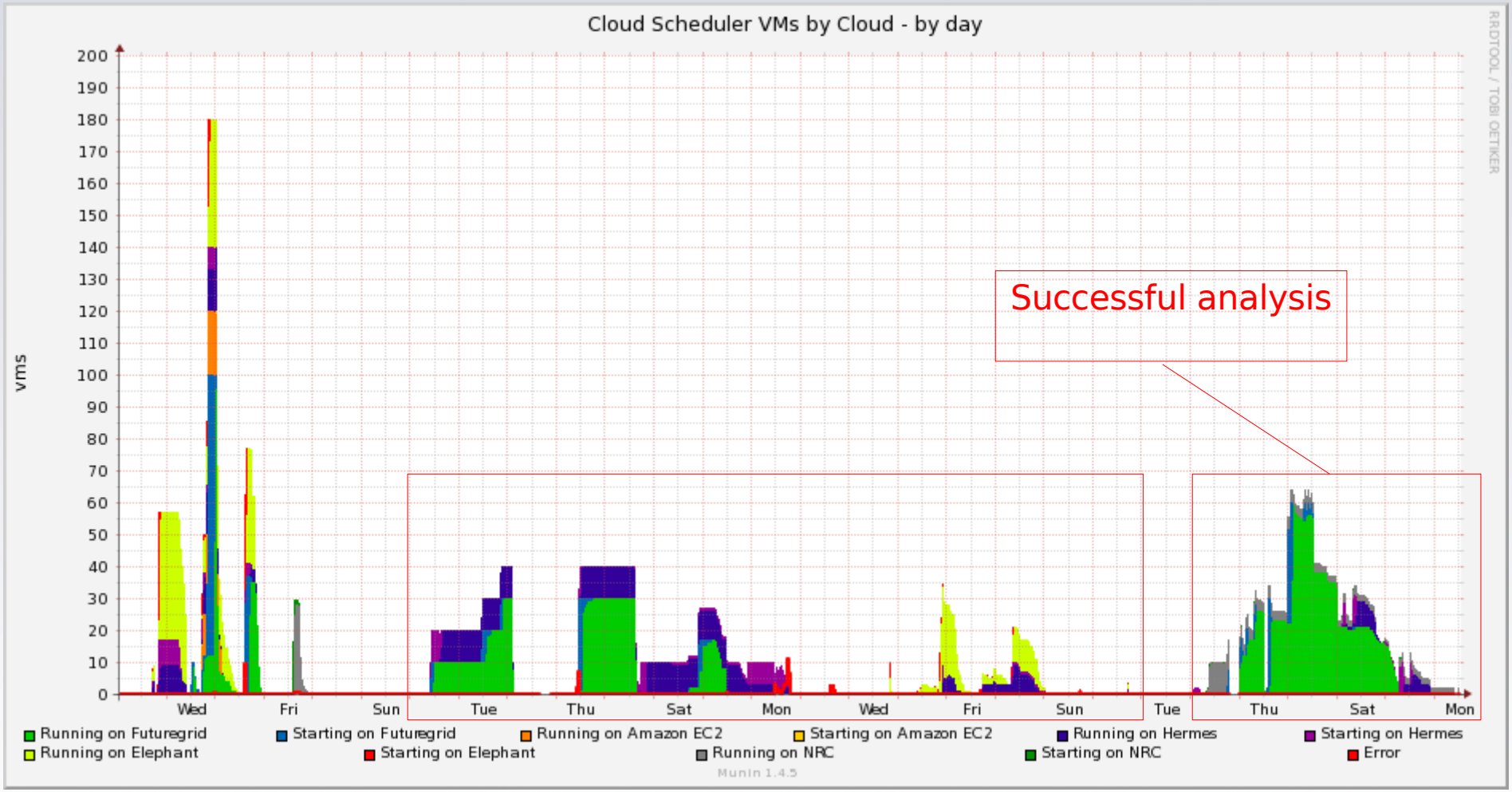
Analysis with User Developed Images

- Batch usage during the month of February.



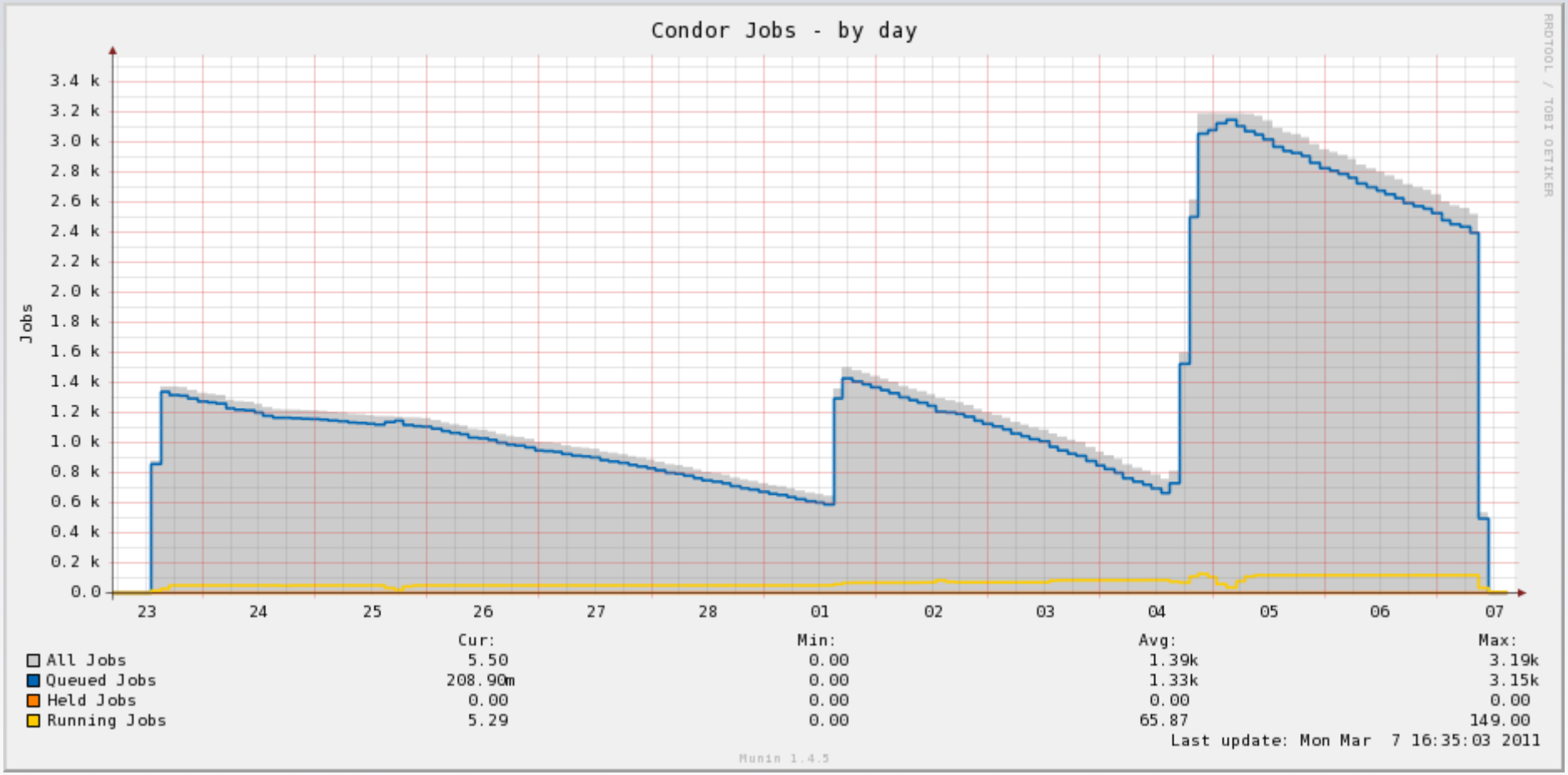
Analysis with User Developed Images

- Batch usage during the month of February.

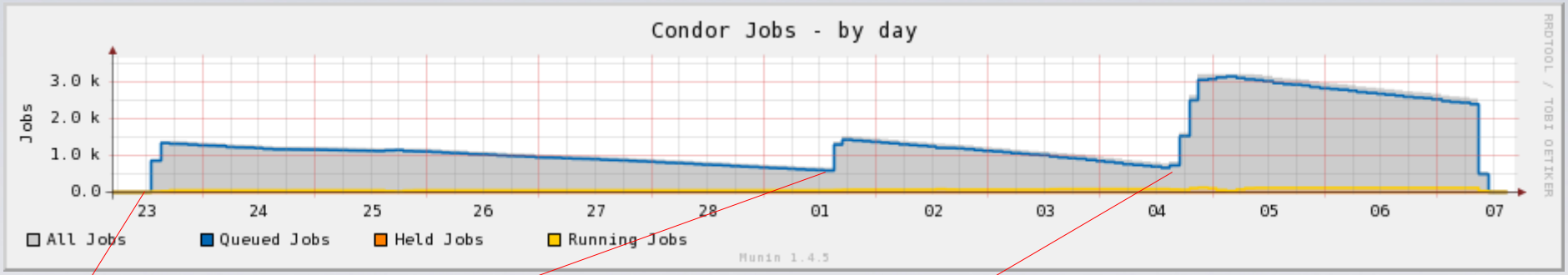


Simulation Production

- February 23 to March 7



Simulation Production

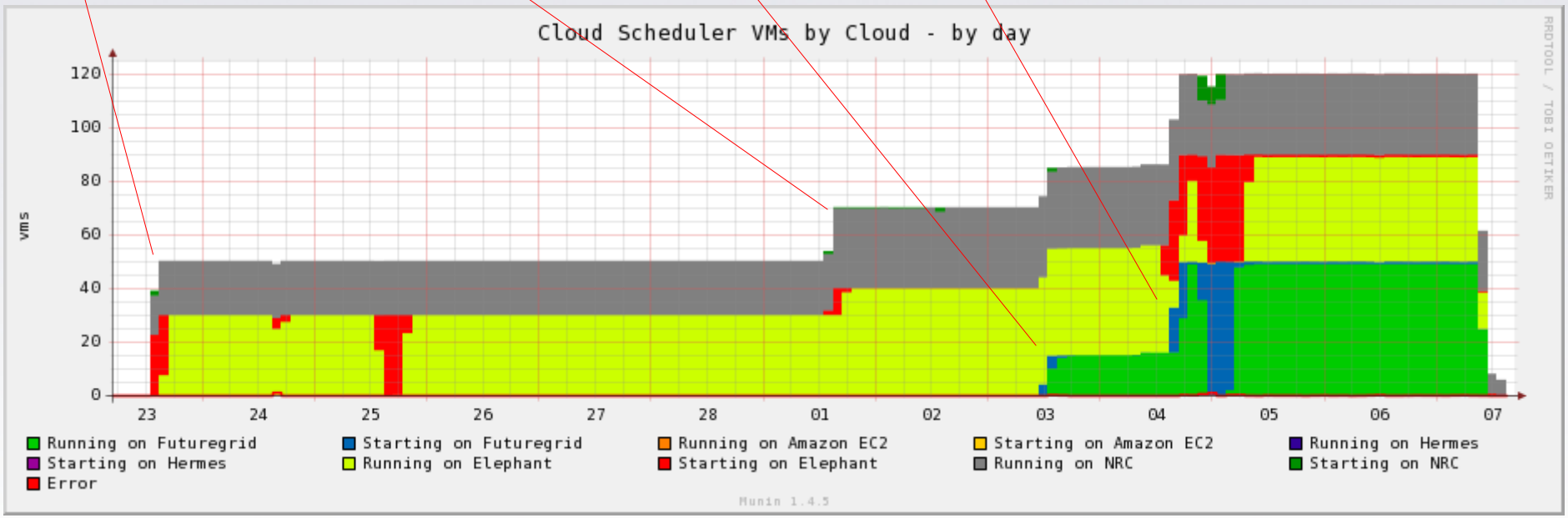


~1300 jobs
50 slots
2 time zones

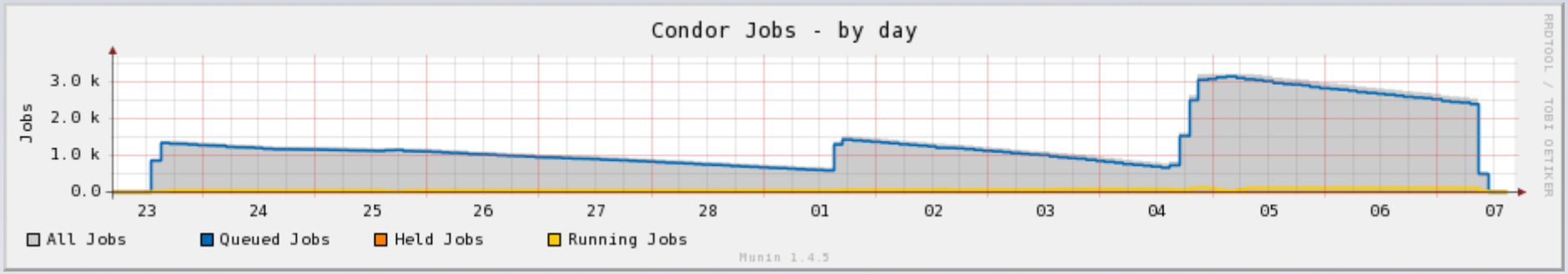
~850 more jobs, 20 more slots

15 more slots, 3rd time zone

2500 more Jobs, 35 more slots



Simulation Production

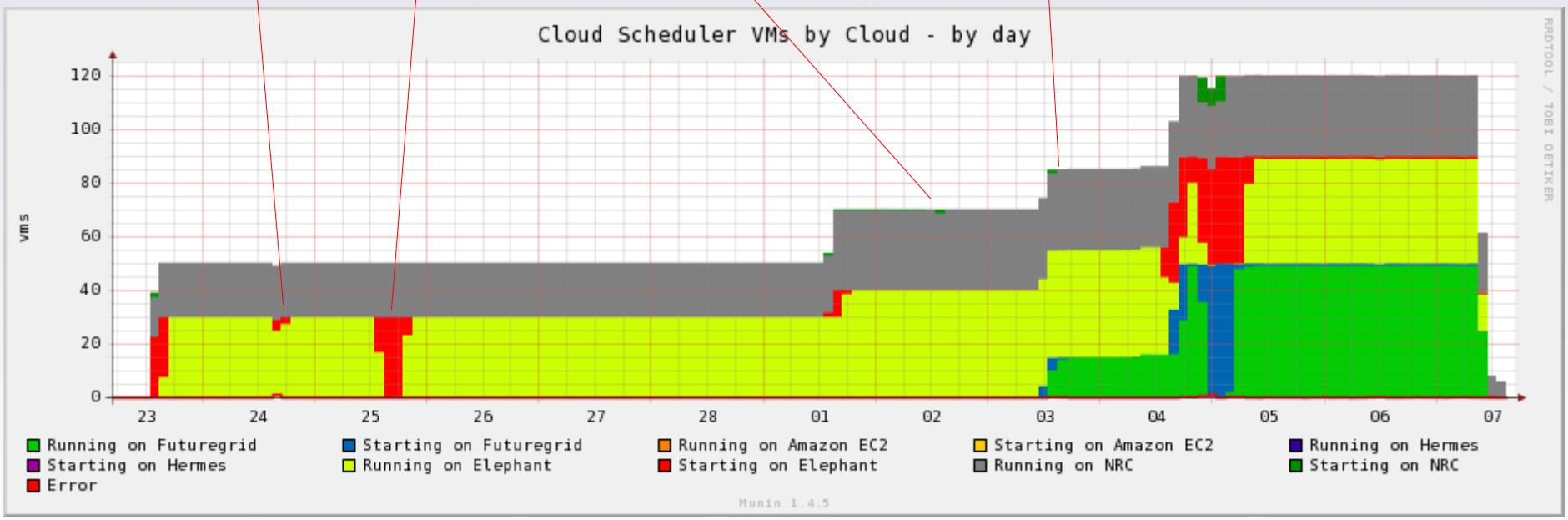


Mother-board re-placed

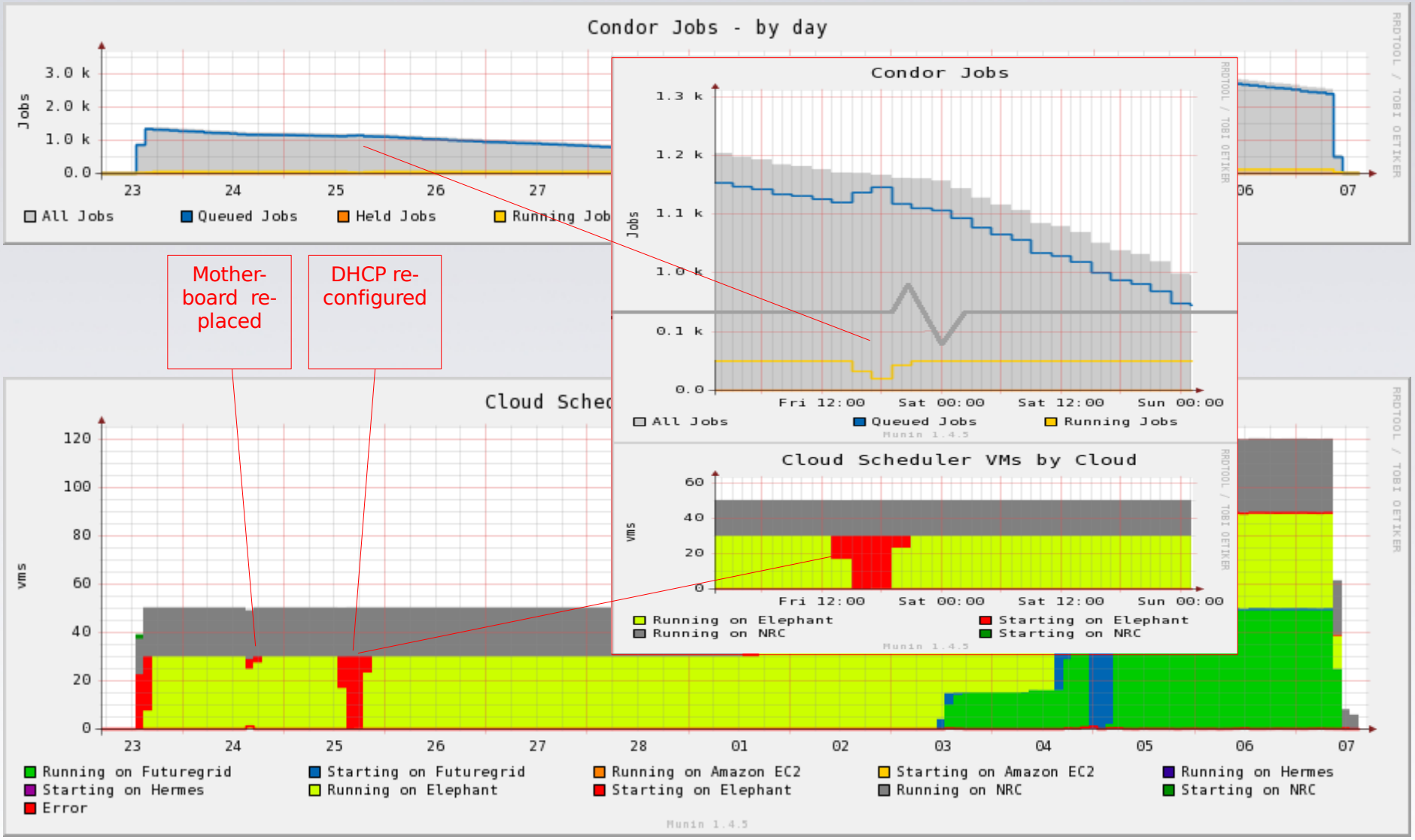
DHCP re-configured

DHCP re-configured

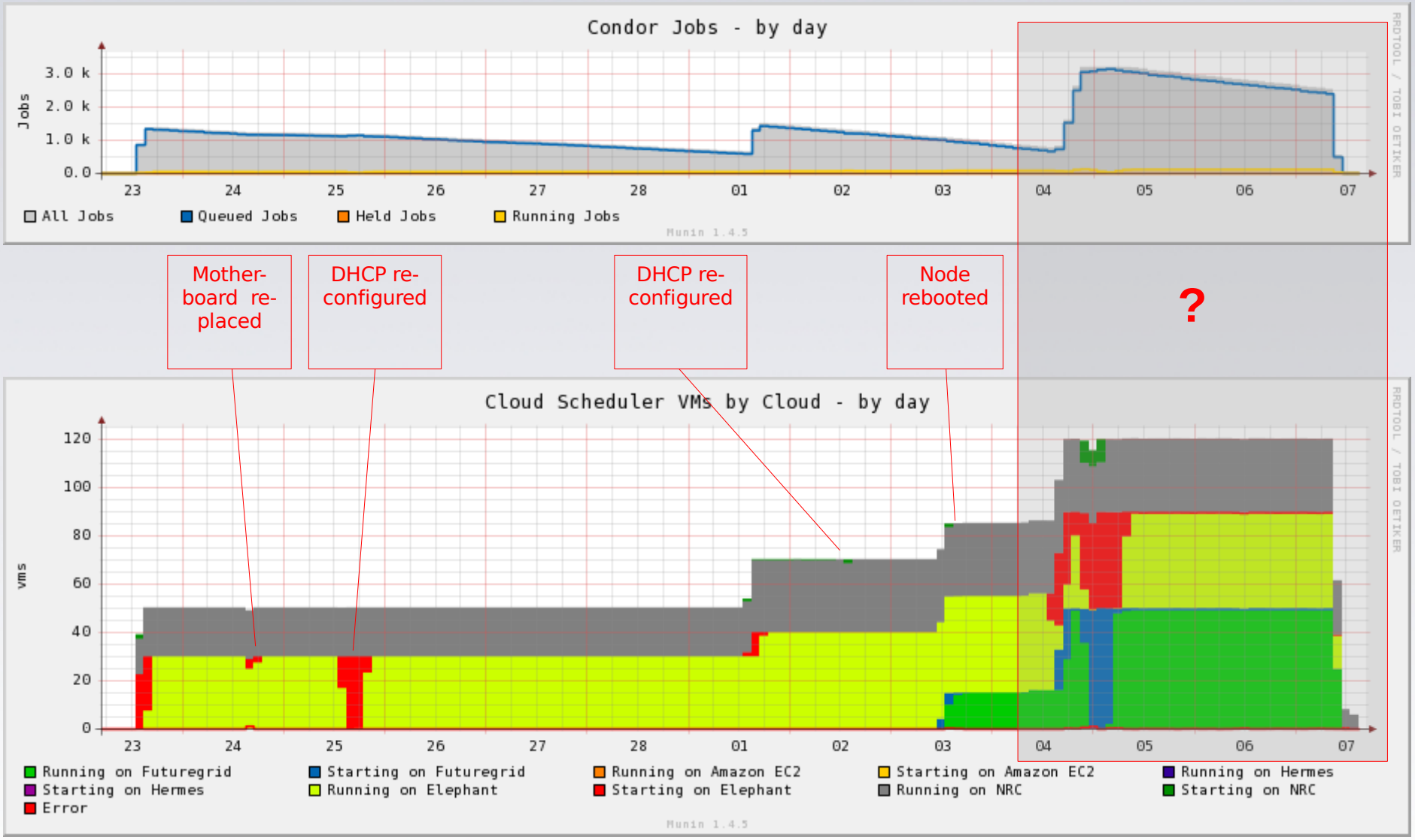
Node rebooted



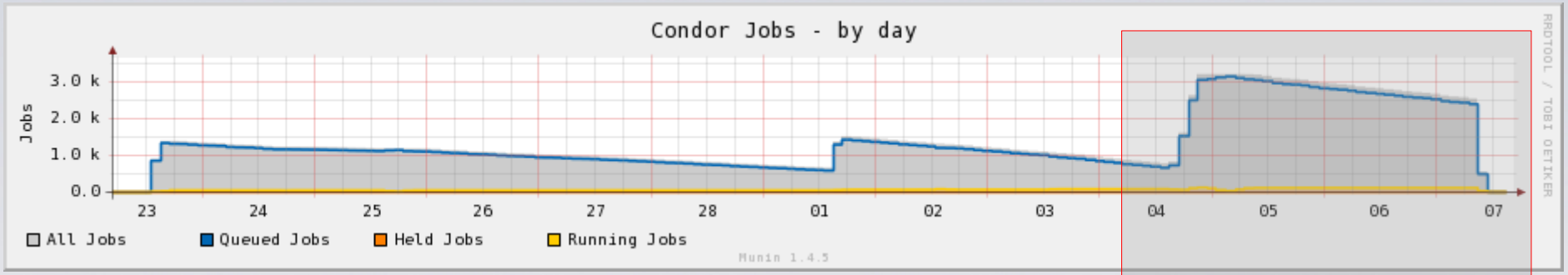
Simulation Production



Simulation Production

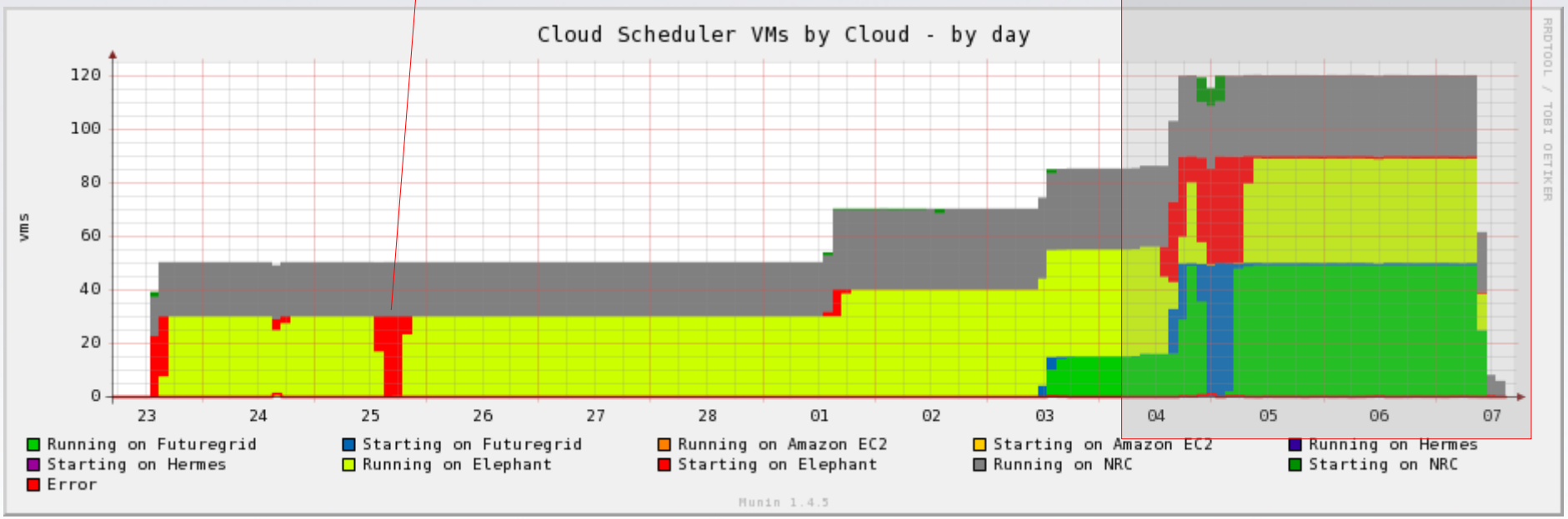


Simulation Production

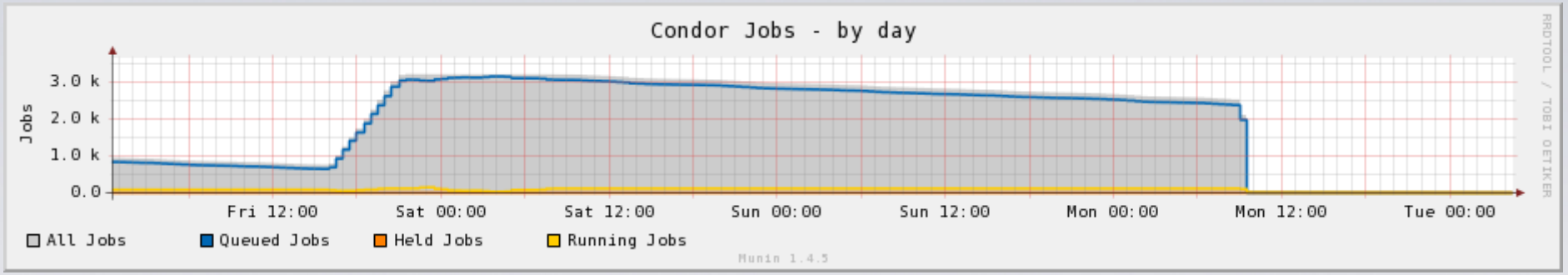


DHCP re-configured

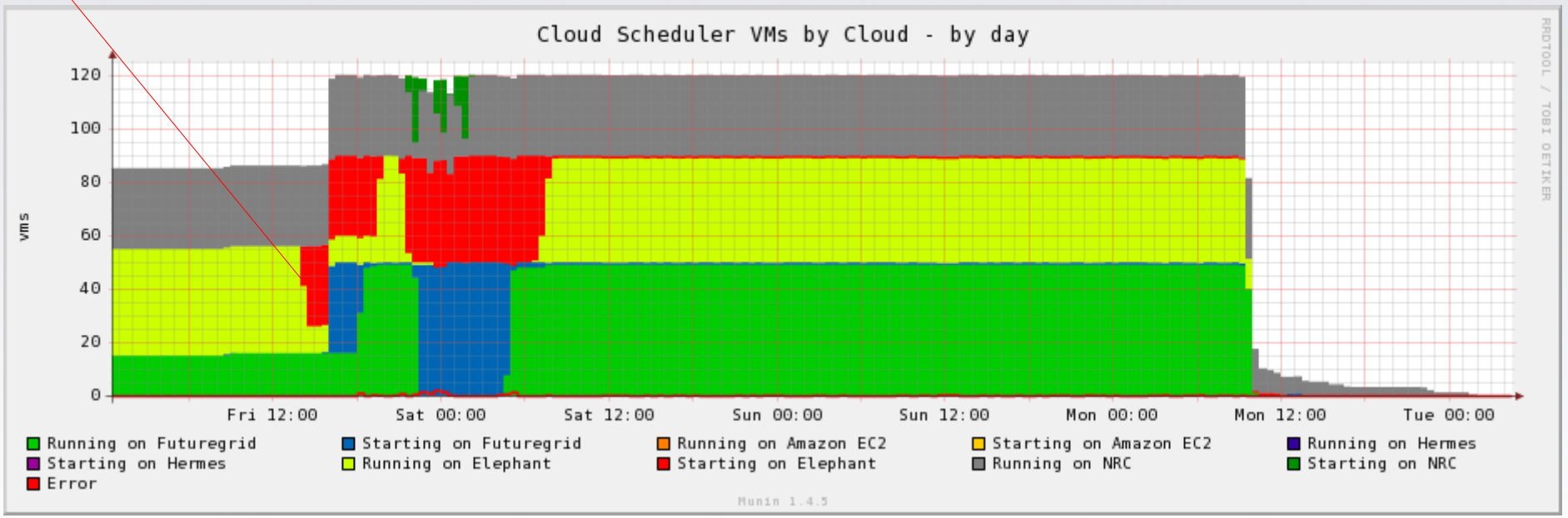
These events occurred 7 days after the first DHCP reconfiguration



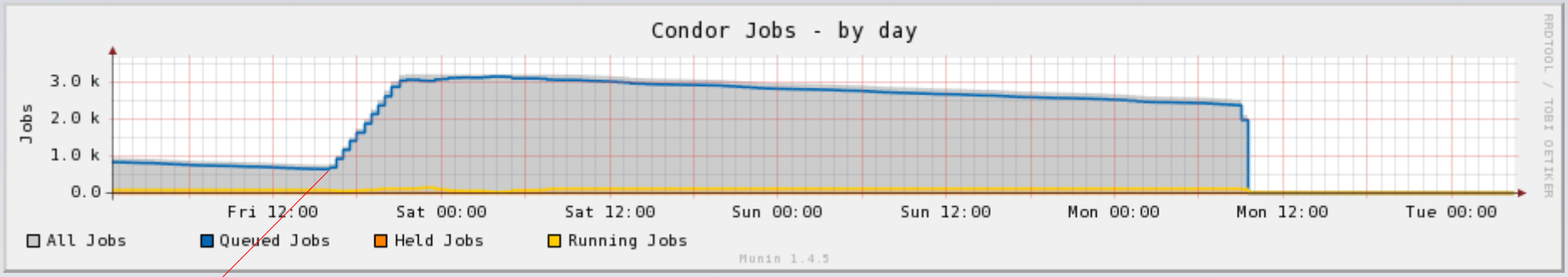
Simulation Production



Leases expired

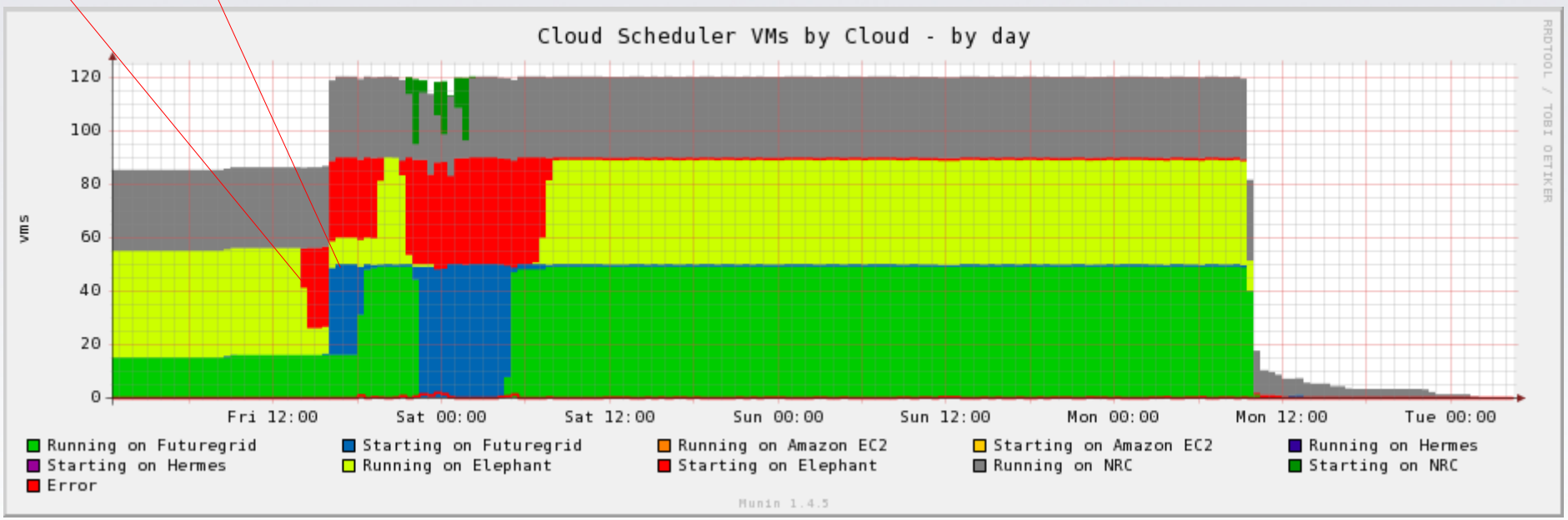


Simulation Production

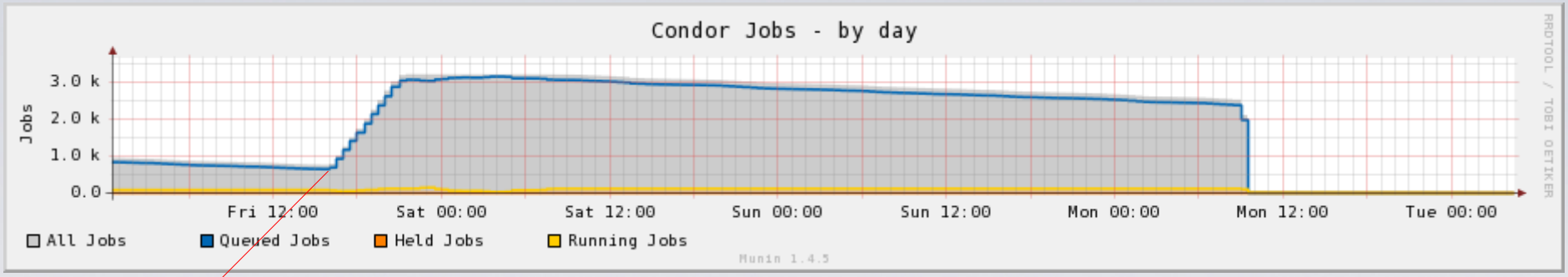


Leases expired

2500 more Jobs, 35 more FG slots



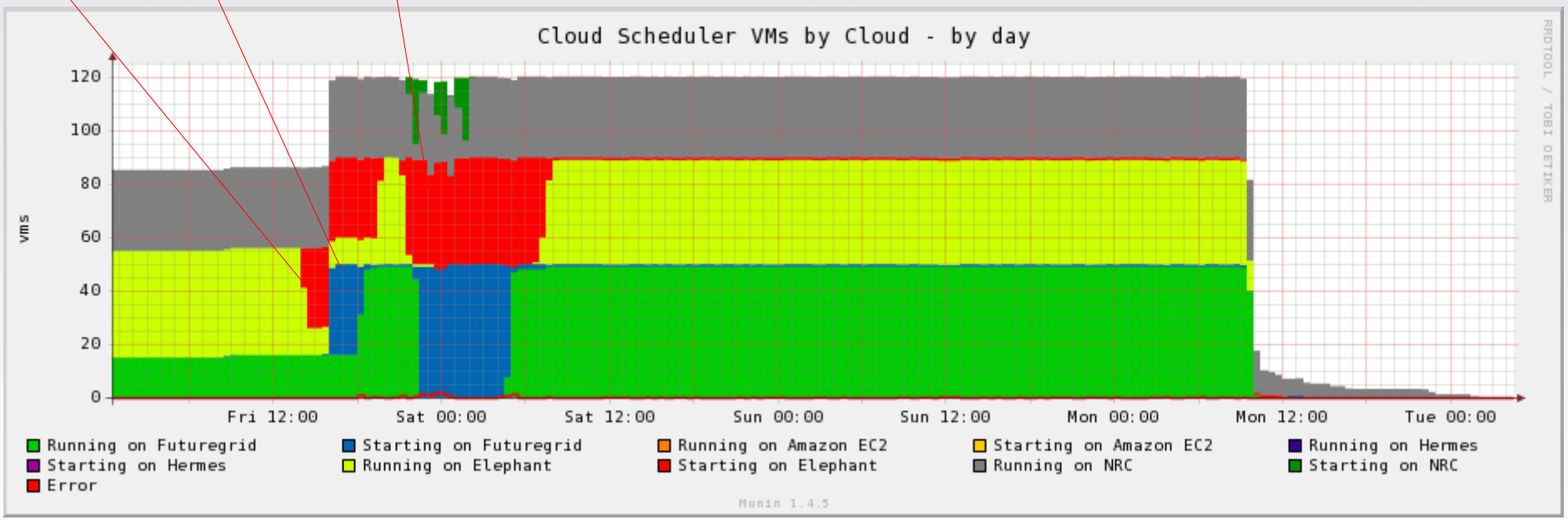
Simulation Production



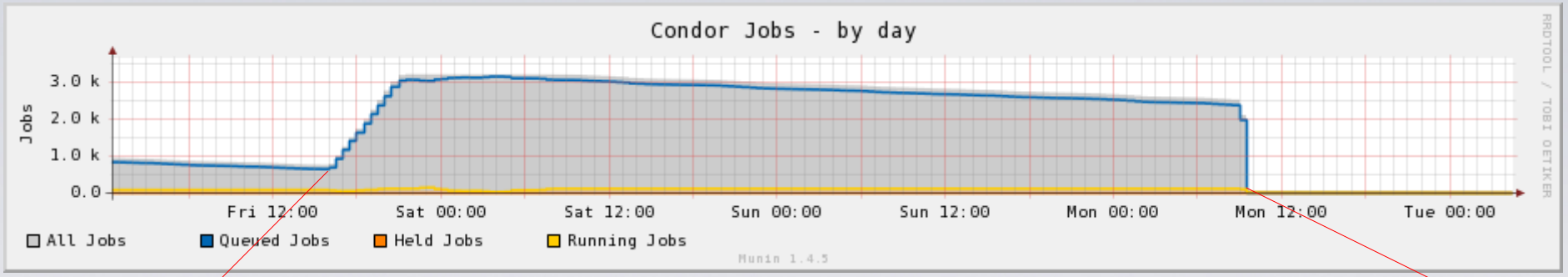
Leases expired

2500 more Jobs, 35 more FG slots

Job submissions took nearly 6 hour. Effective Condor DoS



Simulation Production

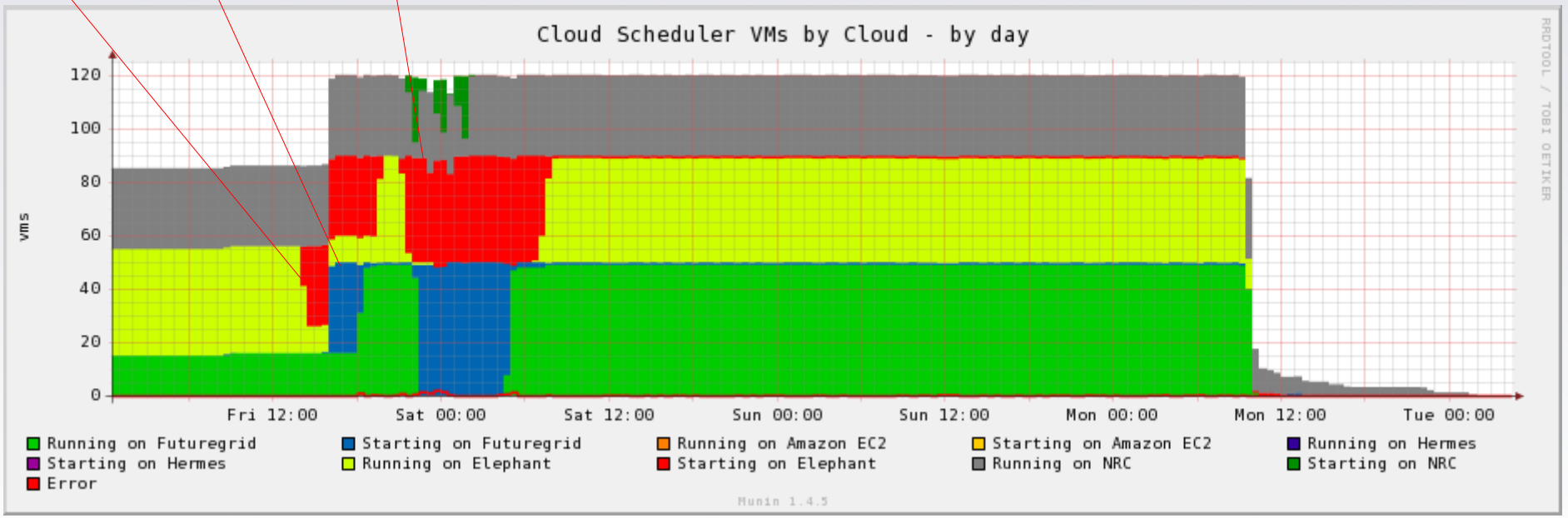


Leases expired

2500 more Jobs, 35 more FG slots

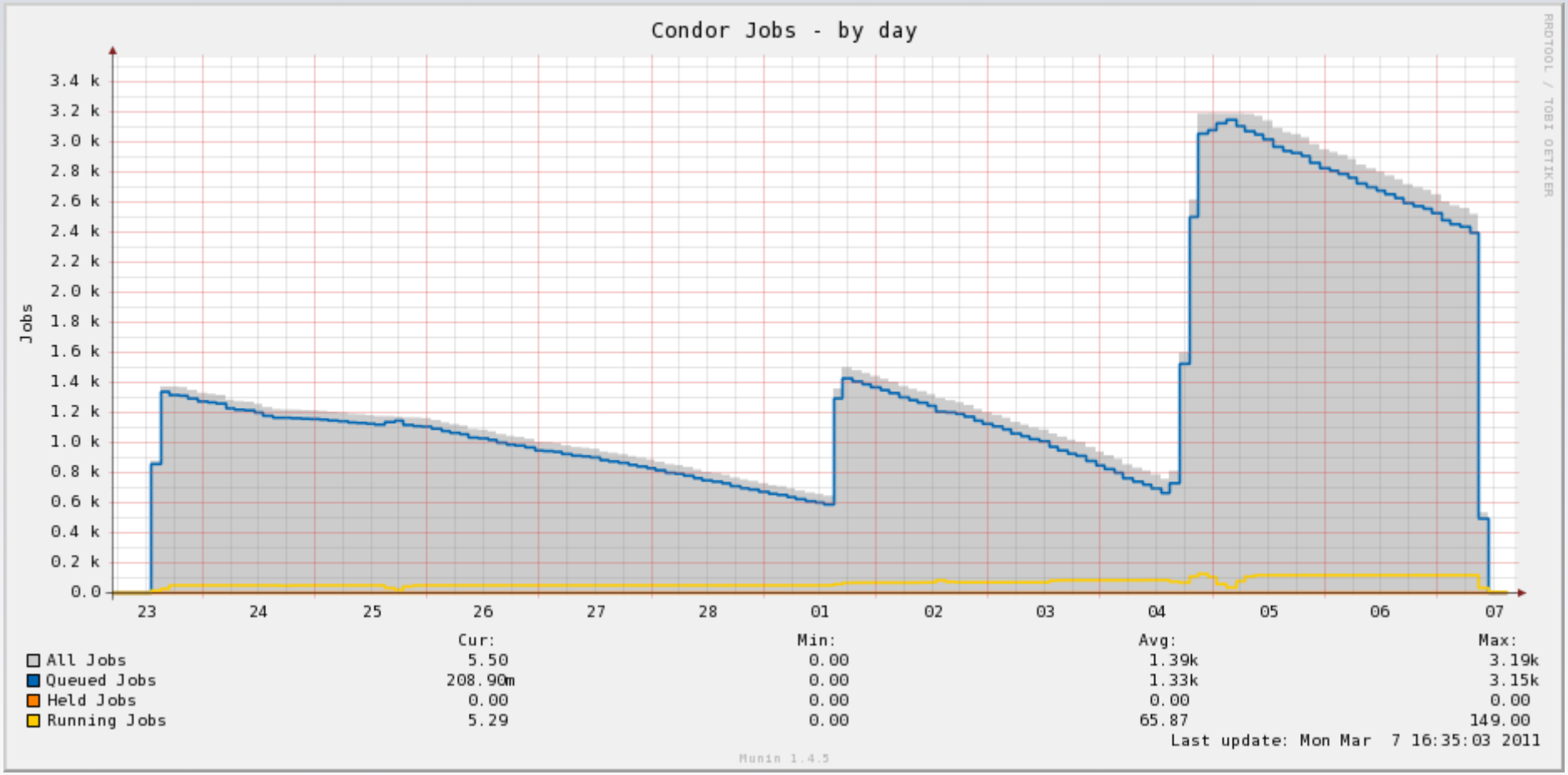
Job submissions took nearly 6 hour. Effective Condor DoS

Shutdown; condor_rm used to remove jobs from queue



Simulation Production

- ~2100 two hour jobs completed in 12 days



Components & References

- Open Source code developed by University of Victoria:
  - Cloud Scheduler >=0.11.1, https://github.com/hep-gc/cloud-scheduler
  - Repoman, https://github.com/hep-gc/repoman
- Other Open Source components used:
  - Scientific Linux 5.x (Xen, KVM), http://www.scientificlinux.org
  - Nimbus >=2.5, http://www.nimbusproject.org
  - Condor >=7.4, http://www.cs.wisc.edu/condor
  - MyProxy, http://grid.ncsa.illinois.edu/myproxy
  - Xrootd, http://xrootd.slac.stanford.edu
  - Lustre >=1.8.3, http://wiki.lustre.org/index.php/Main_Page
  - Squid 2.7.STABLE8, http://www.squid-cache.org
  - Munin 1.4.5 (epel repository), http://munin-monitoring.org

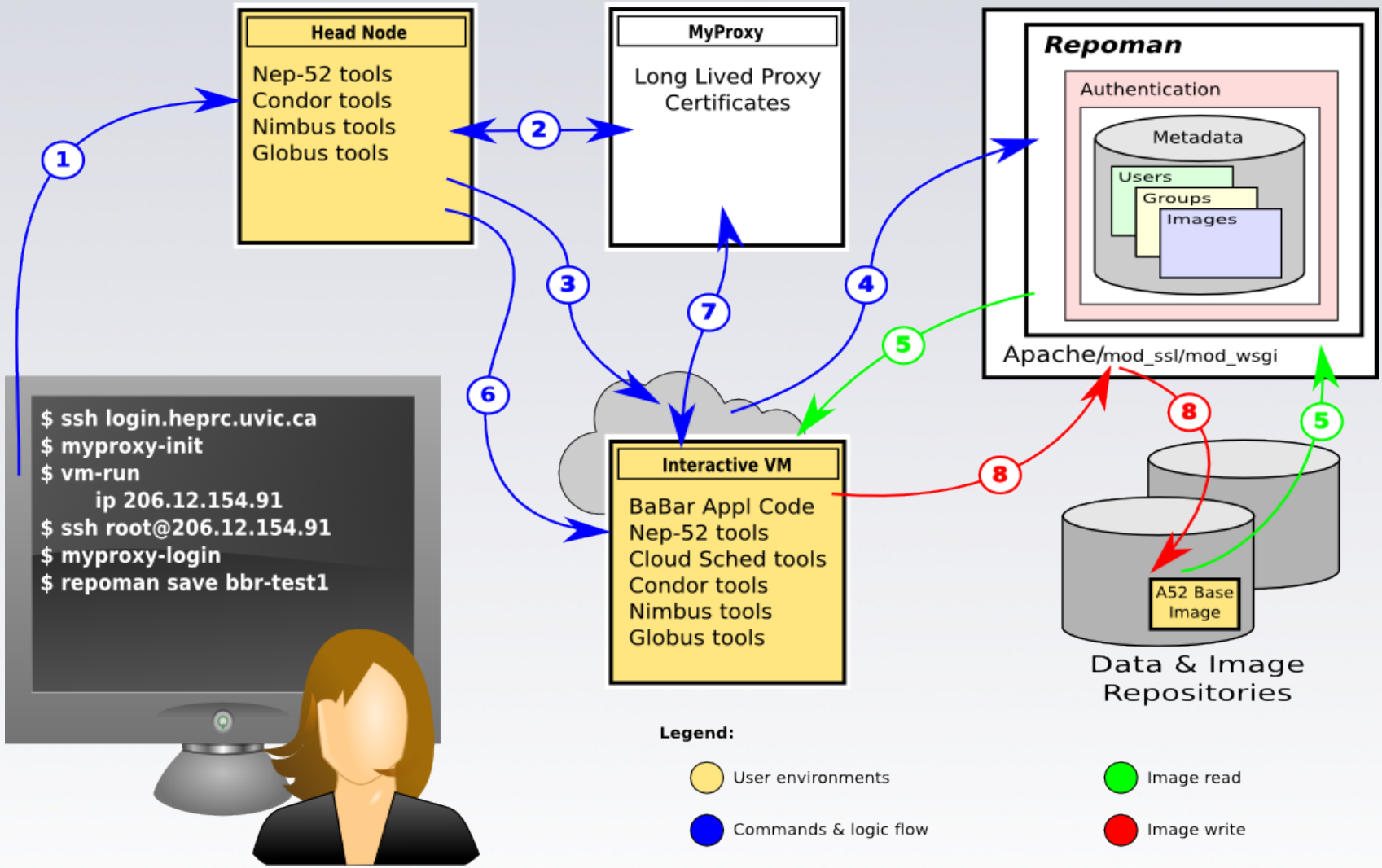
Summary

- Comprehensive solution allowing:
  - Interactive user development of images.
  - Large volumes of batch processing.
- Solution is flexible, easy to use, efficient, scalable, and fault tolerant.
- Systems employs open source components which are readily available.
- Missing pieces developed in-house as open source projects.
- Supported by: CANARIE, Futuregrid, Google, & Amazon EC2

End of Presentation

The following additional slides provide more details of the interactive and batch systems.

The Interactive System (see notes on next page)

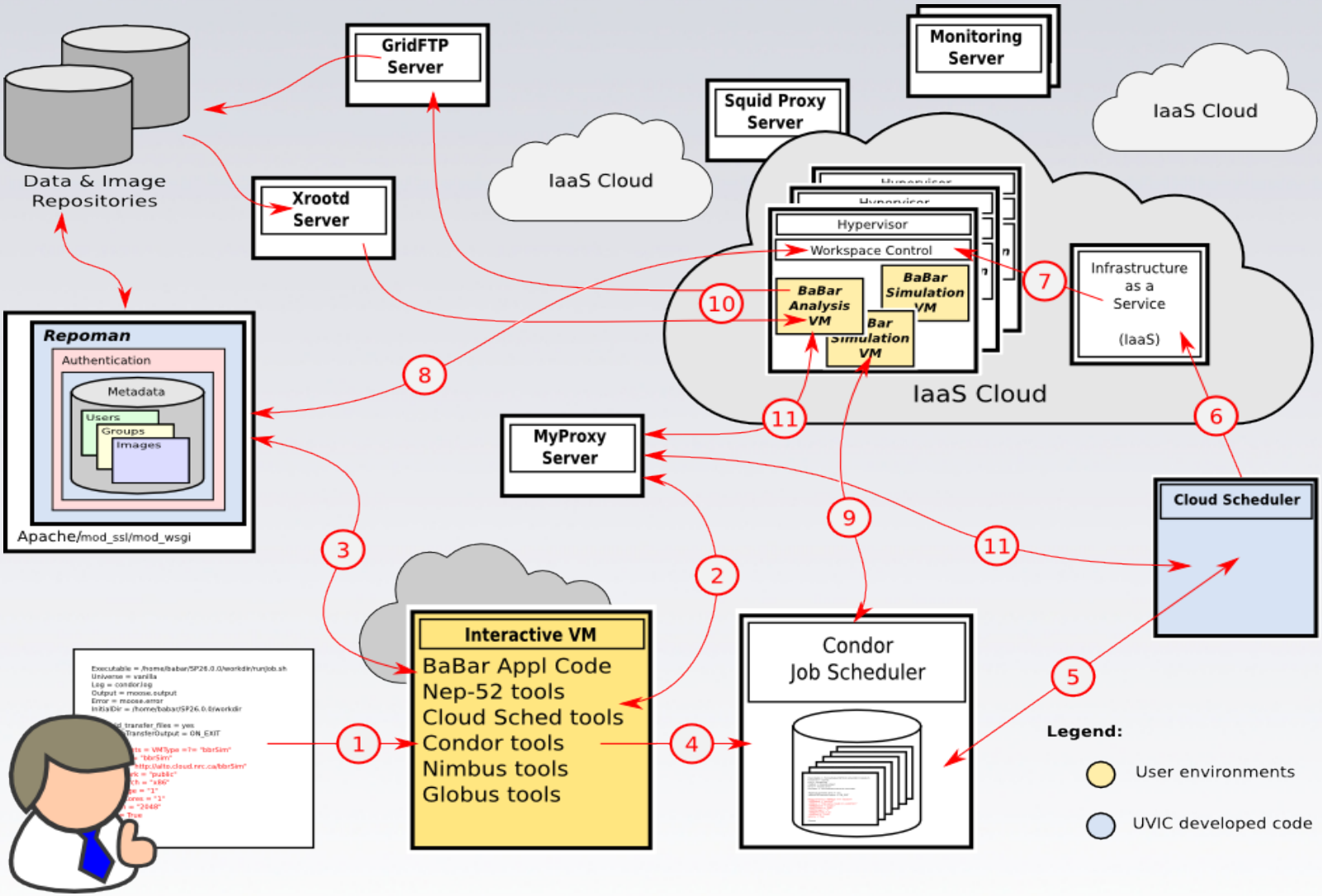


The Interactive System

Numbers refer to the numbered arrows on previous diagram:

1. User remotely logs into the head node.
2. User stores a long-lived X509 proxy certificate on the MyProxy server.
3. The “vmrun” command requests IaaS deploy an interactive VM (base or user image).
4. IaaS/workspace control retrieves the image specified in the Condor job file via HTTP/HTTPS.
5. The URL specified in the Condor job file is serviced by Repoman. Once the image has booted, the IP address of the VM is returned to the user.
6. The user remotely logs in as root to the newly booted VM. No password will be required; root has user's public ssh key. As root, user has the ability to customize the environment to suit.
7. User obtains a short-lived X509 proxy certificate so that they can interact with Repoman and Condor/Cloud Scheduler.
8. User can issue the “repoman save” command at any time to save the an image of the currently running VM. Images created in this way can be used interactively or in batch.

The Batch System (see notes on next 3 pages)



The Batch System

Numbers refer to the numbered arrows on previous diagram:

1. The user remotely logs into an interactive/head node and prepares the environment and Condor job files.

2. The user will need to establish an X509 proxy certificate in order to submit jobs. We use a MyProxy server to manage certificates.

3. Having a proxy certificate allows the user to interact with the Repoman repository manager. Repoman provides a full range of functions to modify and manage images and, minimally, would be used to determine the URL of the image to be used.

4. The user issues the `condor_submit` command to queue jobs and can use Condor functions to query and manage their workload. However, Condor is unable to execute jobs until an appropriate VM/Image becomes available.

5. Cloud Scheduler periodically inspects the Condor queues to find jobs for which there are no matching VMs/images.

6. Cloud Scheduler requests Infrastructure as a Service (IaaS) to deploy a a VM and boot the image specified by the job file.

Continued on next slide.

The Batch System

Numbers refer to the numbered arrows on previous diagram:

7. IaaS selects a nodes and instructs Workspace Services (node control) to boot the image in a new VM.

8. Workspace Services fetches the image via HTTP/HTTPS (serviced by Repoman) and calls the node's hypervisor to deploy the VM and boot the Image.

9. At boot completion, the Condor initialization script runs and registers the VM with the Condor server which responds by dispatching jobs requiring corresponding attributes (eg. image type, cpus, memory, etc.)

10. Special purpose services are provided for BaBar jobs to read and write data. Data is normally read through an Xrootd server, while a GridFTP server is planned for the return of output data to the data repository.

11. For long running jobs/workloads, it may be necessary to renew X509 proxy credentials via the MyProxy server.

Continued on next slide.

The Batch System

Numbers refer to the numbered arrows on previous diagram:

5. When scanning the Condor queues, Cloud Scheduler recognizes when there are no more jobs for a particular image type in a running VM.

6. Cloud Scheduler requests IaaS to shutdown and destroy redundant Vms.

7. IaaS directs Workspace Services on the corresponding nodes to shutdown and destroy redundant VMs.

9. A Condor shutdown script unregisters terminating VMs from the Condor server.