Using NeCTAR Resources

Lucien Boland[1], Paul Coddington[2], Sean Crosby[1], Joanna Huang[1], **Antonio Limosani**[1,3], Martin Sevior[1], Sachin Wasnik[3], Tony Williams[4], Ross Wilson[2], Kevin Varvell[3] ,Shunde Zhang[2]

July 10th, 2013

1. University of Melbourne
2. eResearch SA
3. University of Sydney
4. University of Adelaide

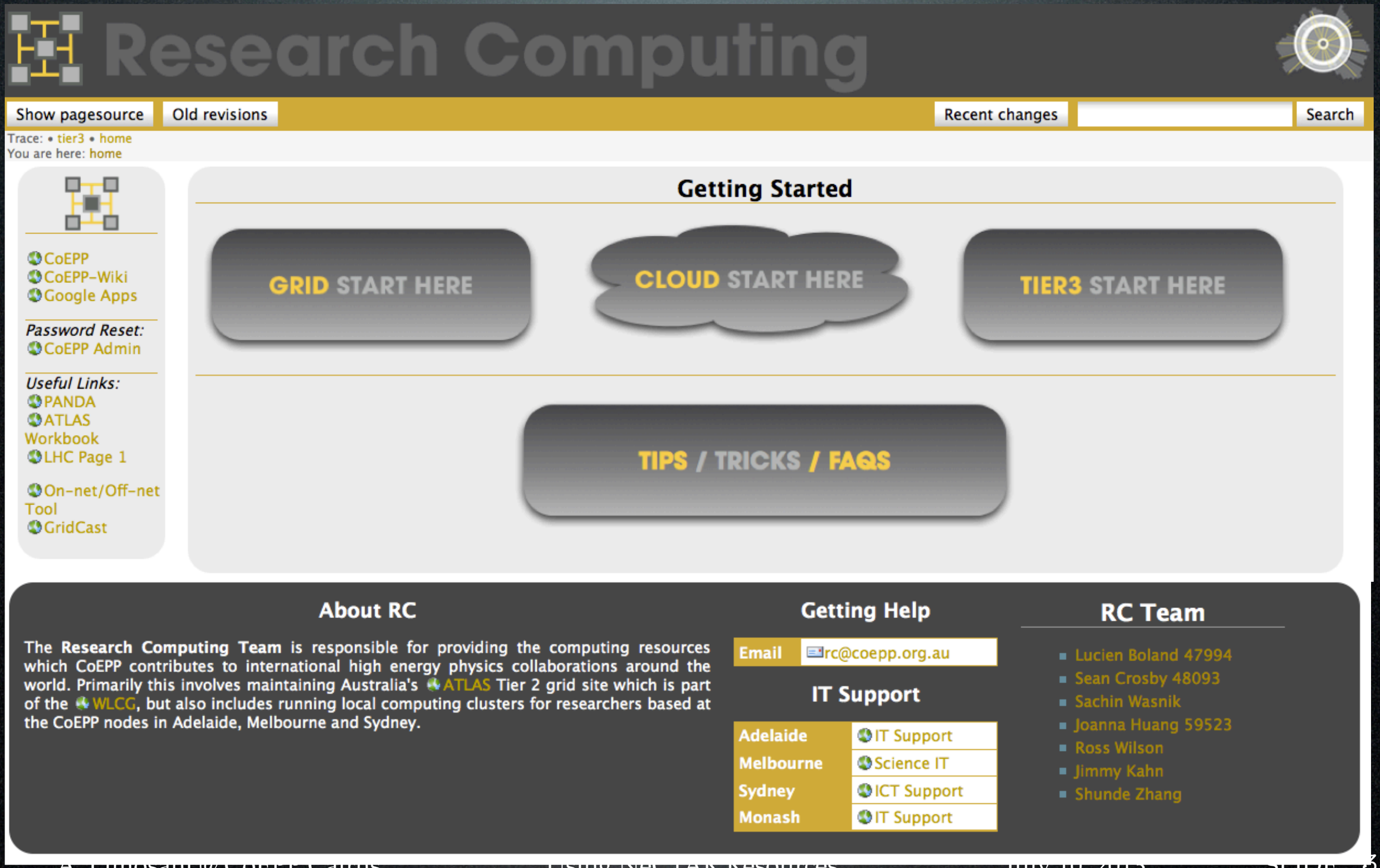
Australian Government
Department of Industry, Innovation, Climate Change, Science, Research and Tertiary Education

Australian Government
Australian Research Council

Outline

- How we currently use computing
- NeCTAR cloud and RDSI projects
- What a user can do now on the Pilot system utilising NeCTAR cloud
- Potential future user experiences

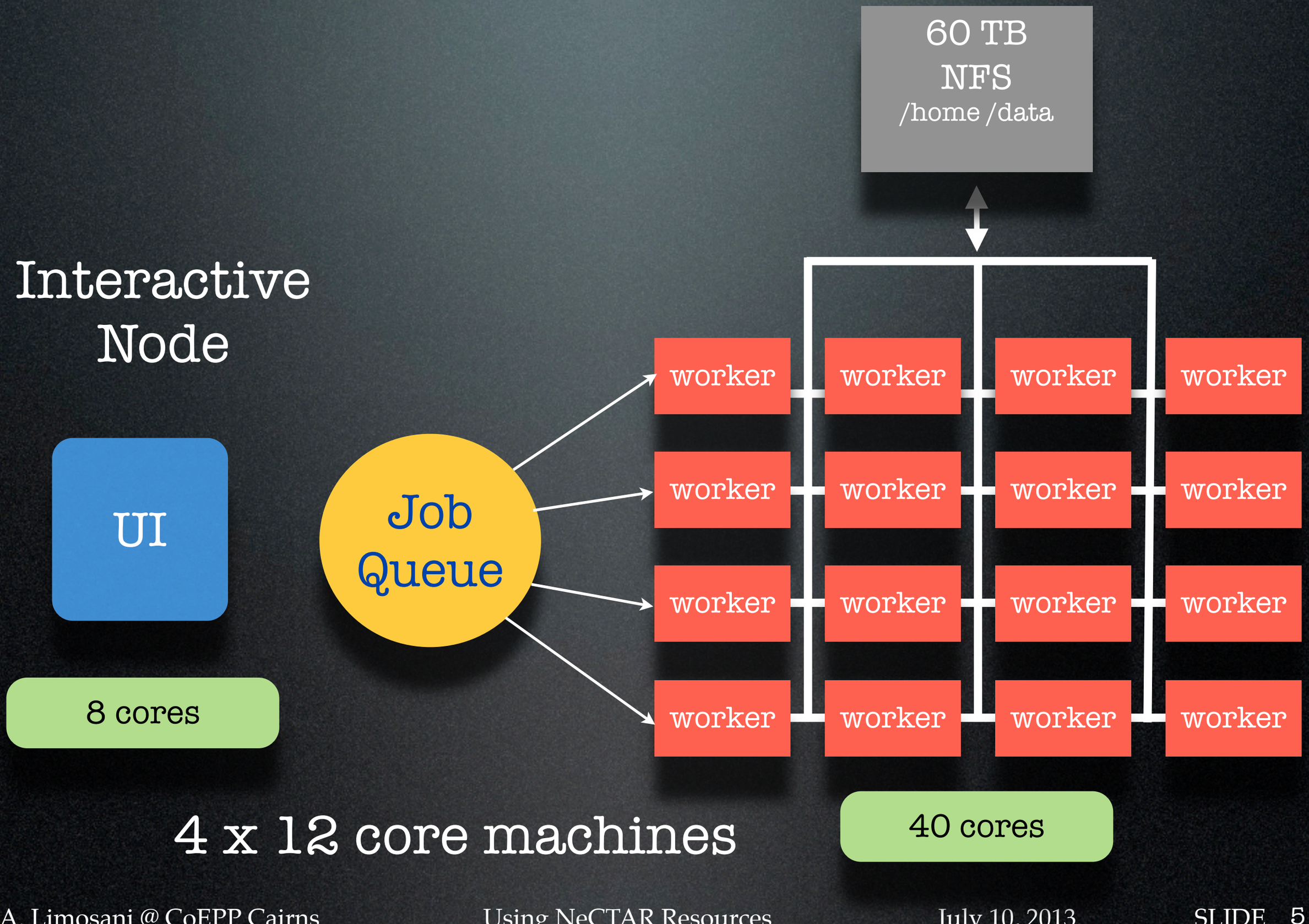
https://rc.coepp.org.au/



Working model

- Experimentalists
  - ATLAS Tier 2 site in Melbourne
    - ScratchDisk, LocalGroupDisk
  - ATLAS Tier 3 sites
    - ui.atlas.unimelb.edu.au (28 cores, 60 TB)
    - sydpp.physics.usyd.edu.au (48 cores, 50 TB)
    - coepp1.ersa.edu.au (20 cores, 4 TB)
- CoEPP Theorists
  - Local physics school resources or other

Melbourne Physical Tier 3



4 x 12 core machines

Our experience

- Experimentalists
  - Develop and test analysis code on Tier 3 interactive machines (UI)
  - Submit jobs to the Grid to run over big datasets / Download data using Grid tools to Tier 3 storage
  - Process reduced data to make histograms etc interactively or via batch queues
- Theorists and experimentalists
  - Simulation of physics processes
  - Toy Monte Carlo studies
  - Large scale fitting to test and constrain large parameter spaces
- Is there ever enough disk storage and processors? Storage at ≥ 90% now!

Challenges

- Shelf/warranty life of our physical computing infrastructure 3-5 years
- Research computing needs grow exponentially (ATLAS will increase data rate by ~3 times in 2015)
- Significant federal government funding ($100 M) is available in compute and storage using **virtualisation** platforms
  - NeCTAR Cloud 20,000 cores
  - RDSI 40 PetaBytes
- Less operations, focus more on the user experience
- Our needs challenge and grow the capabilities of the national infrastructure

NeCTAR

- Cloud computing at a national scale, accessible for all researchers for the first time
- CoEPP project “High Throughput computing for globally connected science” Project Leader: Martin Sevior
- NecTAR project team: Joanna Huang (Leader), Ross Wilson, Shunde Zhang & AL - with significant in-kind support from CoEPP RC team: Lucien, Sean and Sachin.
- This workshop : CoEPP Partner Institute Duke University: support from Doug Benjamin
- Commenced March 2012
- Project will run to March 31, 2014*

Pilot Nectar Cloud

CVMFS

CERN VM File System
ATLAS/ Belle II / Other

/cvmfs

1.2 TB
xtreemfs
/data

CXUI

CXUI

Static Job Queue

Dynamic Job Queue

worker worker worker worker
worker worker worker worker
worker worker worker worker
worker worker worker worker

16 cores each / 64 GB

Max. 120 cores NeCTAR MORE AVAILABLE!

30 cores static / 30 cores dynamic

Using the servers

Log into the **cxui** job sumission node:

- ssh <user_name>@cxui.cloud.coepp.org.au

or log into **cxin01** or **cxin02** interactive nodes:

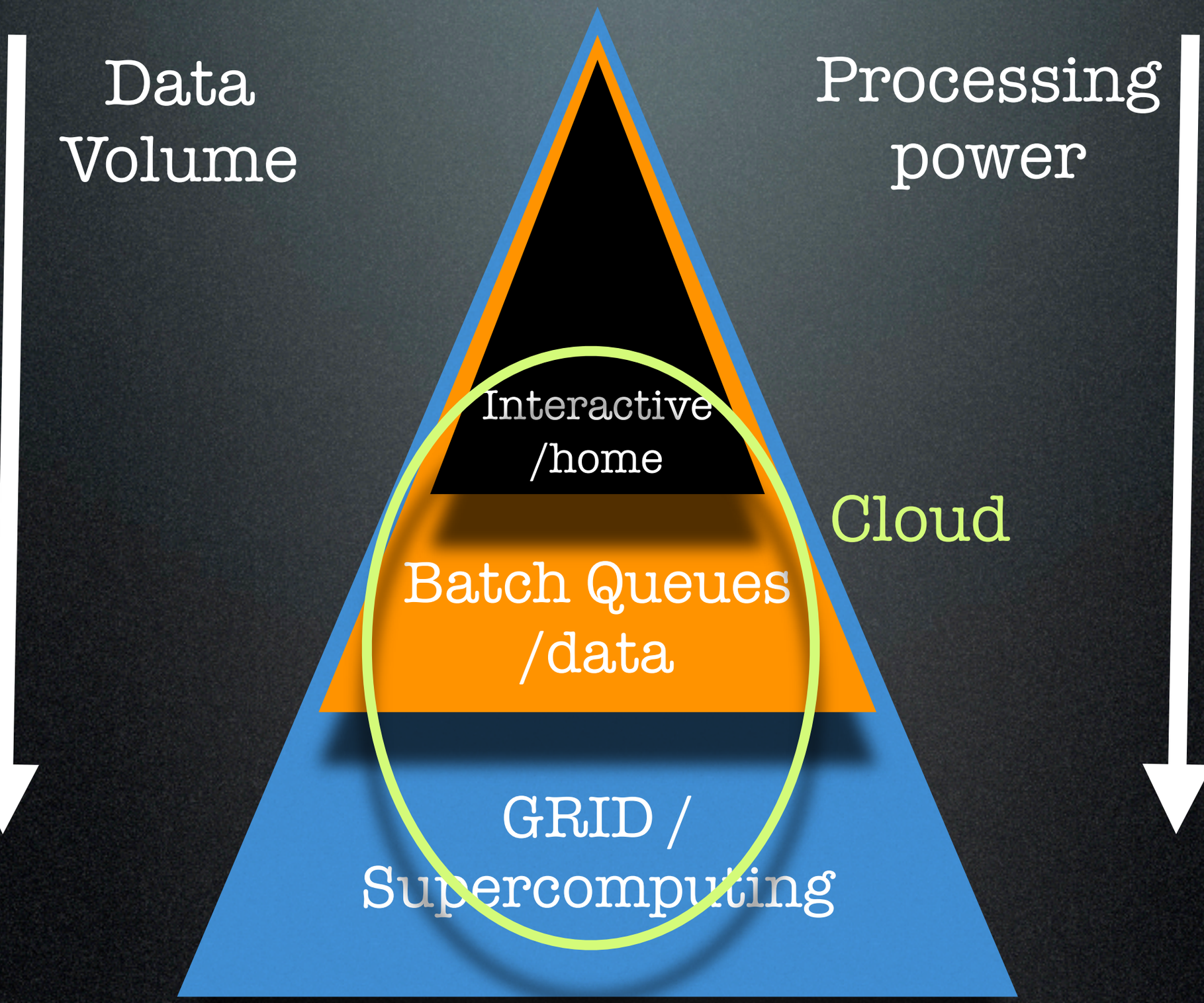
- ssh -Y <user_name>@cxin01.cloud.coepp.org.au
- ssh -Y <user_name>@cxin02.cloud.coepp.org.au

The cxui node is a lower power machine and should be used only for batch job submission.

cxin01 and cxin02 are for interactive use with 16 cores and 64GB memory.

- Users can use these as they wish. If we find demand exceeds capacity for these two, they will be upgraded.

User Resources



Batch Jobs

- Every cloud job has 4 steps:
  - Identify the program you want to run and the data that is input to and output from it and create a batch job script
  - Prepare files in the /data partition
  - Run the program on a batch node
  - Retrieve files from the /data partition

A simple batch job

- Create a directory **job_test** under your home directory. In this directory create a file **run_job.sh** containing:

```
#!/bin/bash

# Set the name of this batch job
#PBS -N my_test

# Join standard and error job outputs into one file
#PBS -j oe

# Set the maximum resource usage we expect from the job.
# This usually helps the scheduler work more
efficiently.
#PBS -l ncpus=1
#PBS -l mem=512MB
#PBS -l vmem=512MB
#PBS -l walltime=0:01:00

cd /data/antonio/job_test
cat job_test.data > job_test.output
echo "Done!"
```

Prepare the /data partition

- We have a job script that copies one file into another, but we can't run it yet since our home directory (where you created the job script and input data file) doesn't exist on the cloud worker nodes.

- The only filesystem common the the cxui and cxin nodes is that under **/data**. If you look there you will see directories with the names of CoEPP users:

```
cxui:./job_test>ls -l /data/
total 0
drwxr-xr-x 1 abangert   people 0 Jun 26 04:22 abangert
drwxr-xr-x 1 adunn      people 0 Jun 26 04:22 adunn
drwxr-xr-x 1 alexss     people 0 Jun 26 04:23 alexss
....
```

Submitting the job

- Test the job on the cxin01 or cxin02 nodes before running in the cloud by just running the job script file
- You might want to change a real job so that it doesn't take a lot of time or other resources when you do.

```
cxui:./job_test>ls -l /data/antonio/job_test
total 0
-rw-r--r-- 1 antonio people  20 Jul  8 23:30 job_test.data
-rw-r--r-- 1 antonio people 335 Jul  8 23:30 run_job.sh

cxui:./antonio>qsub /data/antonio/job_test/run_job.sh
11430.c3torque.cloud.coepp.org.au
```

Checking job status

```
cxui:./antonio>qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
11430.c3torque            my_test          antonio                0 Q short

cxui:./antonio>qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
11430.c3torque            my_test          antonio                0 R short

cxui:./antonio>qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
11430.c3torque            my_test          antonio                0 E short
```

Examining output

```
cxui:./~>ls -l /data/antonio/job_test/
total 0
-rw-r--r-- 1 antonio people  20 Jul  8 23:30 job_test.data
-rw-r--r-- 1 antonio people  20 Jul  8 23:40 job_test.output
-rw------- 1 antonio people   0 Jul  8 23:40 my_test.e11428
-rw------- 1 antonio people   6 Jul  8 23:40 my_test.o11428
-rw-r--r-- 1 antonio people 335 Jul  8 23:30 run_job.sh

cxui:./~>cat /data/antonio/job_test/my_test.e11428

cxui:./~>cat /data/antonio/job_test/my_test.o11428
Done!

cxui:./~>cat /data/antonio/job_test/job_test.output
A simple text file.
```

Available queues

```
cxui:./~>qstat -q

server: c3torque.cloud.coepp.org.au

Queue            Memory CPU Time Walltime Node  Run Que Lm  State
---------------- ------ -------- -------- ----  --- --- --  -----
cloud_monash       --   24:00:00 24:00:00   --    0   0 --   E R
batch              --      --       --       --    0   0 --   E R
long               --      --       --       --    0   0 --   E R
short              --      --    01:00:00 01:00:00   --    0   4 --   E R
                                               ----- -----
                                                   0    4
```

- For cloud_monash use “qsub.cloud” instead of qsub
- /data is currently 1.2 TB … generated output should be copied back to storage elsewhere
- There is also a mechanism to **stage-in** your input files into the worker nodes and **stage-out** your output back to your home directory
- https://rc.coepp.org.au/cloud/advanced_staging

Root Benchmarks

- Test programs to benchmark performance of root http://root.cern.ch/drupal/content/benchmarking

Benchmark	Detail	System	queue	CPU time (s)	Wall time (s)
Fitting	./stressFit Minuit 1000000	Cloud ui	batch	1875	1890
Fitting	./stressFit Minuit 1000000	Melb Tier 3	batch	1897	1897
Read/Write + CPU	./stress -b 50000	Cloud ui	batch	1491	3406
Read/Write + CPU	./stress -b 50000	Melb Tier 3	batch	1118	1164

- /data area on cxui mounted using xtreemfs

My Analysis Results

- Input file size (infile.root TTree) 10 GB
- Output file size (outfile.root THist) 40 MB

Machine	Description	queue	duration
MacBook Pro	**MacBookPro**	**interactive**	**31m 55s**
cxin01.cloud.coepp.org.au	**Interactive Cloud ui**	**interactive**	**52m 35s**
ui.atlas.unimelb.edu.au	**Melb Tier 3**	**interactive**	**31m 5s**
sydpp.physics.usyd.edu.au	**Syd Tier 3**	**interactive**	**30m 7s**
cxui	**Cloud ui**	**batch**	**52m 7s**
cxui	**Cloud ui**	**dynamic**	**49m 49s**
ui.atlas.unimelb.edu.au	**Melb Tier 3**	**batch**	**29m 9s**
sydpp.physics.usyd.edu.au	**Syd Tier 3**	**batch**	**30m 0s**

Code other than CERN Root

- Any software can be placed in CVMFS server to share
- Can be configured inside a VM
- VM is made to perfectly suit the needs of your job

Summary / Comments

- Significant additional resources are available in NeCTAR cloud (Pilot system - encourage early adopters)
- Same performance on CPU intensive jobs as physical cluster
- Read/Write not as fast yet still very useable
- With some startup costs can utilise stage-in/ stage-out for jobs for read/write
- NeCTAR upgrades at Melbourne “Cell” completed a week or so ago have vastly improved the performance

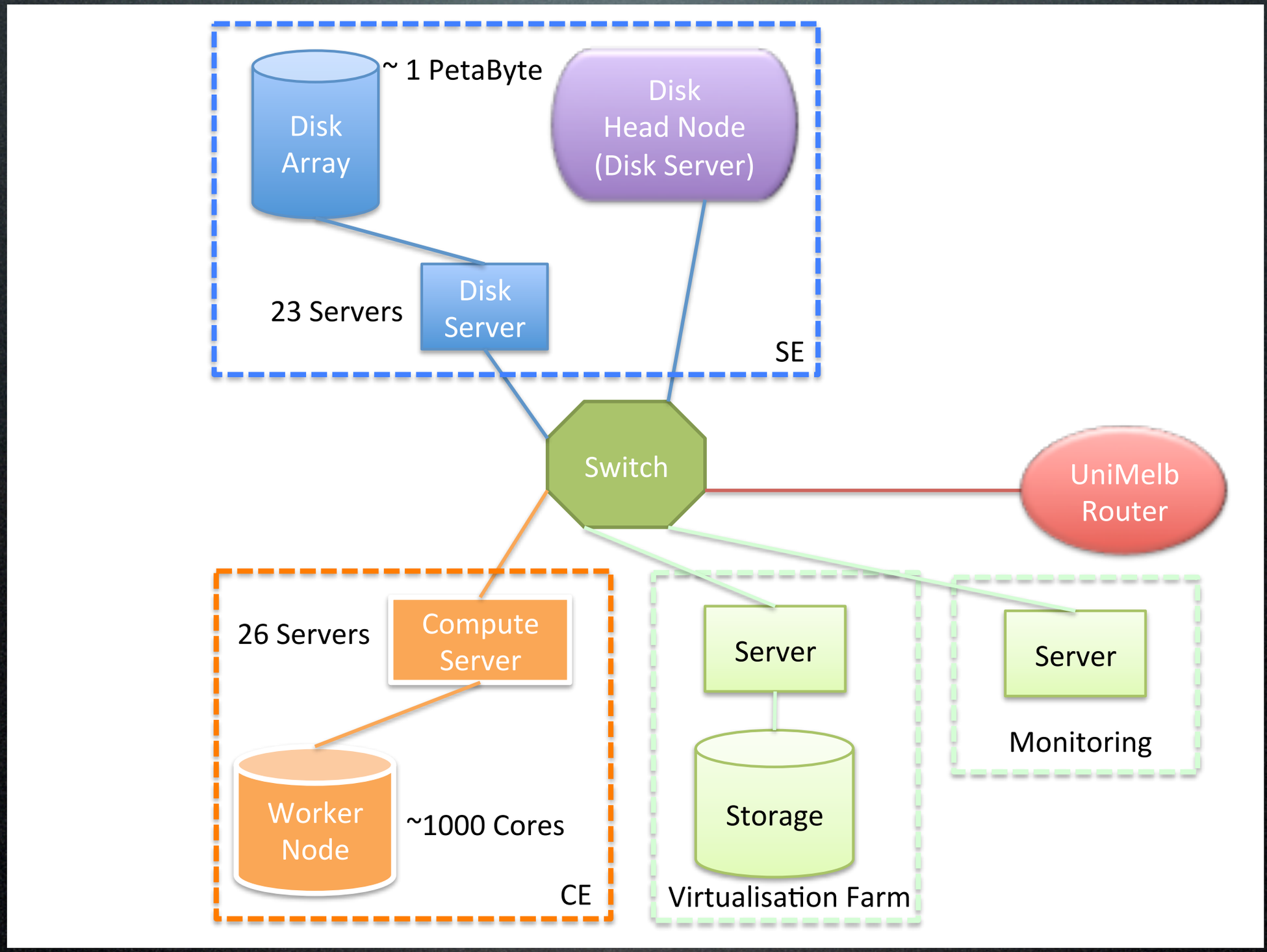
Problems

- Data is being duplicated on local group disk and local Tier 3 storage - not sustainable
- All data being stored on /home is not sustainable
- We don't effectively share resources across nodes
- NeCTAR and RDSI long term can be a part of the solution

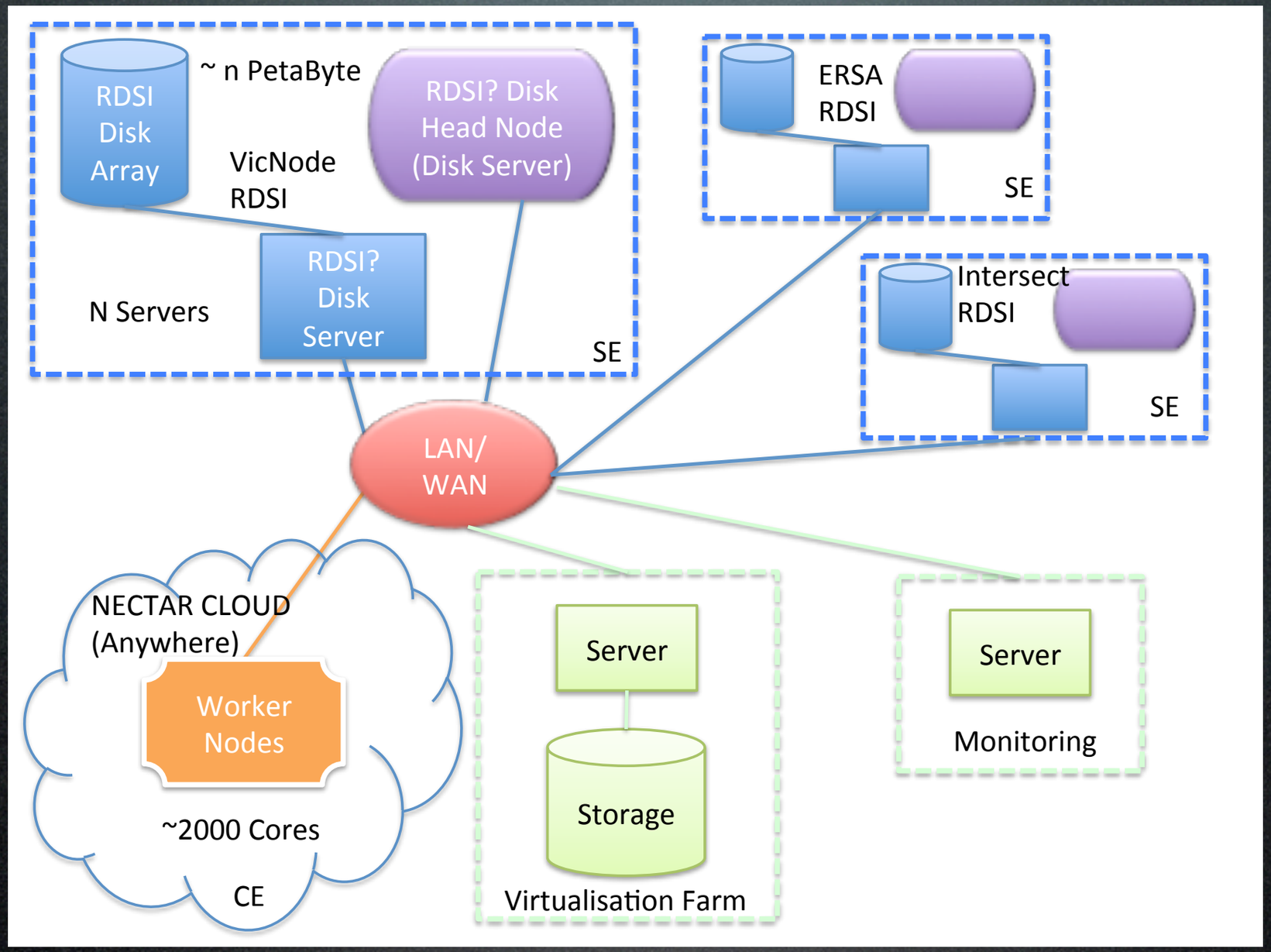
Future directions

- Access ATLAS data stored on the GRID in batch queue jobs via xrootd
- Pilot system at Intersect (NSW) coming online later this month - production system available in November/December
- eRSA Nectar/RDSI resources available in August for us to test.
- VicNode - storage available to test ~August?

Physical Cluster



Cloud-based Cluster



- Migrate to a Wide Area Network (WAN) to share resources nationally
- Challenge is to maximise data throughput and minimise latency

Virtual Machines

- Virtual machine (VM) is a software implementation of machine that executes programs like a physical machine
- Emulate an existing architecture
- Multiple instances of VMs leading to more efficient use of resources leading to cloud computing
- Advantages
  - multiple OS on a single machine but entirely isolated
  - application provisioning, maintenance, high availability and disaster recovery
- Disadvantages
  - VM is less efficient than a real machine
  - When VMs run concurrently on the same physical host (varying and unstable performance)

Nodes


Timor Sea
Northern Territory
Coral Sea
Queensland
Australia
QCIF
Brisbane
Western Australia
iVEC
Perth
South Australia
eRSA
New South Wales
Intersect
ANU-NCI
Great Australian Bight
VicNode
Australian Capital Territory
Melbourne
Uni. Tas.

Pilot Cloud ui