vlsci

LIFE SCIENCES COMPUTATION

SC14: VLSCI Site Report

Chris Samuel

20/11/2014

The Victorian Life Sciences Computation Initiative is funded by the Victorian Government and contributing institutions, is hosted by the University of Melbourne and includes the first IBM Research Collaboratory for Life Sciences. It exists for all Victorian life science researchers and as at July 2012 is the biggest supercomputer facility devoted to life sciences in the world.

Victoria

THE UNIVERSITY OF MELBOURNE

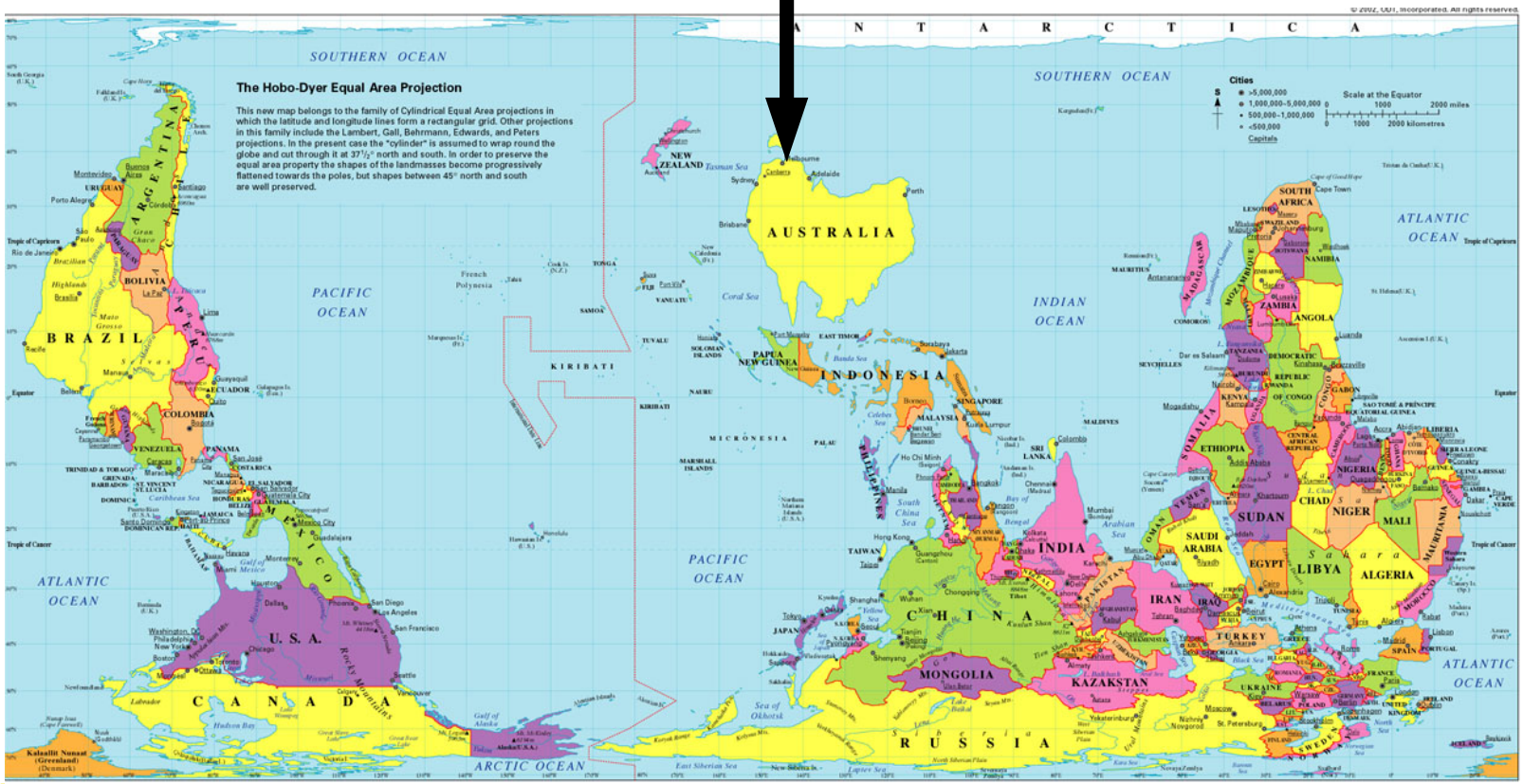
IBM

Overview

- VLSCI Overview
- Hardware (HPC and storage)
- Software stack
- History of Slurm at VLSCI
- Torque to Slurm transition
- Current use of Slurm
- Future plans

vlsci
LIFE SCIENCES COMPUTATION

VLSCI



VLSCI Overview

- Dedicated to computation in the life sciences
- State Govt and University of Melbourne funded
- In-kind contribution from UoM, Monash, and La Trobe universities
- Most powerful HPC facility in the world dedicated to life sciences computation

VLSCI Overview

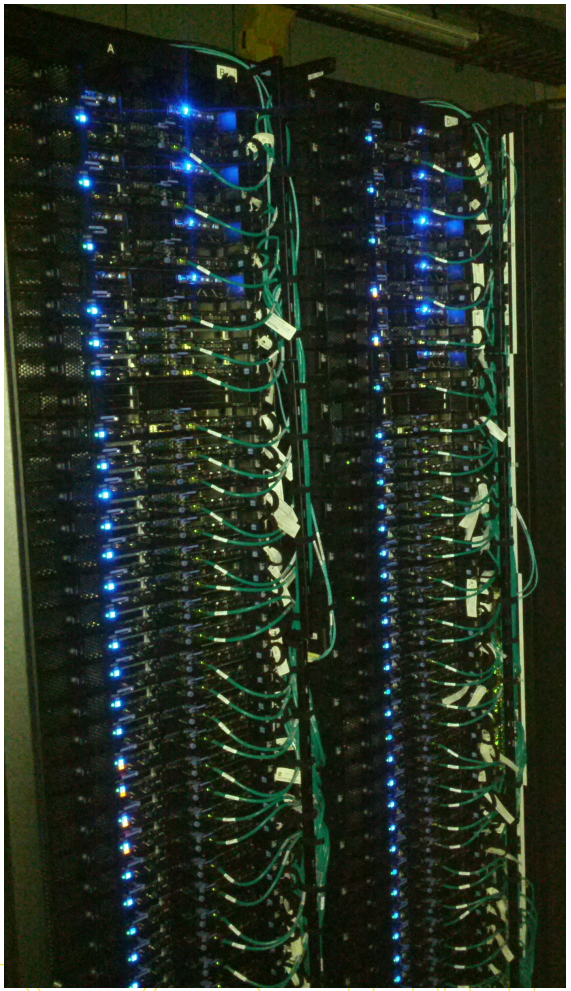
- VLSCI has four parts:
  - Directorate: the director, finance and admin
  - PCF: Peak Computing Facility – system administration and user support
  - LSCC: Life Science Computation Centre – scientists spread across three precincts around Melbourne
  - Communications: PR, sponsorships & outreach

PCF Overview

- PCF manages the VLSCI HPC systems, storage, networks & supporting systems.
- Provides user & software support
- Purpose built data centre @ UniMelb
  - Built 2010, upgraded 2012
- 4 sysadmins, 3 on-site IBMers, 2 specialist programmers, 1 scientist and our boss.

vlsci
LIFE SCIENCES COMPUTATION

Hardware



Merri: Intel Cluster

- IBM iDataplex, Nehalem/Westmere
  - In service June 2010, upgrade 2012.
  - 84 compute nodes (720 cores)
    - 44 nodes with 8 cores and 48GB RAM
    - 36 nodes with 8 cores and 96GB RAM
    - 1 SGI UV10 with 32 cores and 1TB RAM
    - 3 nodes with 16 cores and 1TB RAM
  - Mellanox FDR14 Infiniband

Barcoo: Intel Cluster

- IBM iDataplex, 2 x 2.7 GHz SB
  - In service June 2013.
  - 70 compute nodes (1,120 cores)
    - 67 nodes with 256GB RAM
    - 3 nodes with 512GB RAM
    - 10 nodes with dual Intel Xeon Phi accelerators
  - Mellanox FDR14 Infiniband
  - Diskless nodes

Avoca: Blue Gene/Q

- 4 racks of IBM Blue Gene/Q
  - In service June 2012.
  - Each rack has 1024 nodes
  - Each node has 16 cores and 16GB RAM
  - Total of 65,535 cores and 64TB RAM!
  - 5D torus interconnect.
  - 715TF measured of 838TF theoretical
  - Still in the top 100 of Top500 (#76).

vlsci
LIFE SCIENCES COMPUTATION

vlsci
LIFE SCIENCES COMPUTATION

vlsci
LIFE SCIENCES COMPUTATION

vlsci
LIFE SCIENCES COMPUTATION

Storage

DDN Disk Storage

- Two DDN SFA10K storage systems
  - Both have dual redundant controllers
  - Both talk to 10 shelves of drives
  - One pair control 600 1TB SATA drives
  - One pair control 600 900GB SAS drives
  - RAID arrays run vertically across shelves
  - Controllers run a Linux based OS

IBM Tape Storage

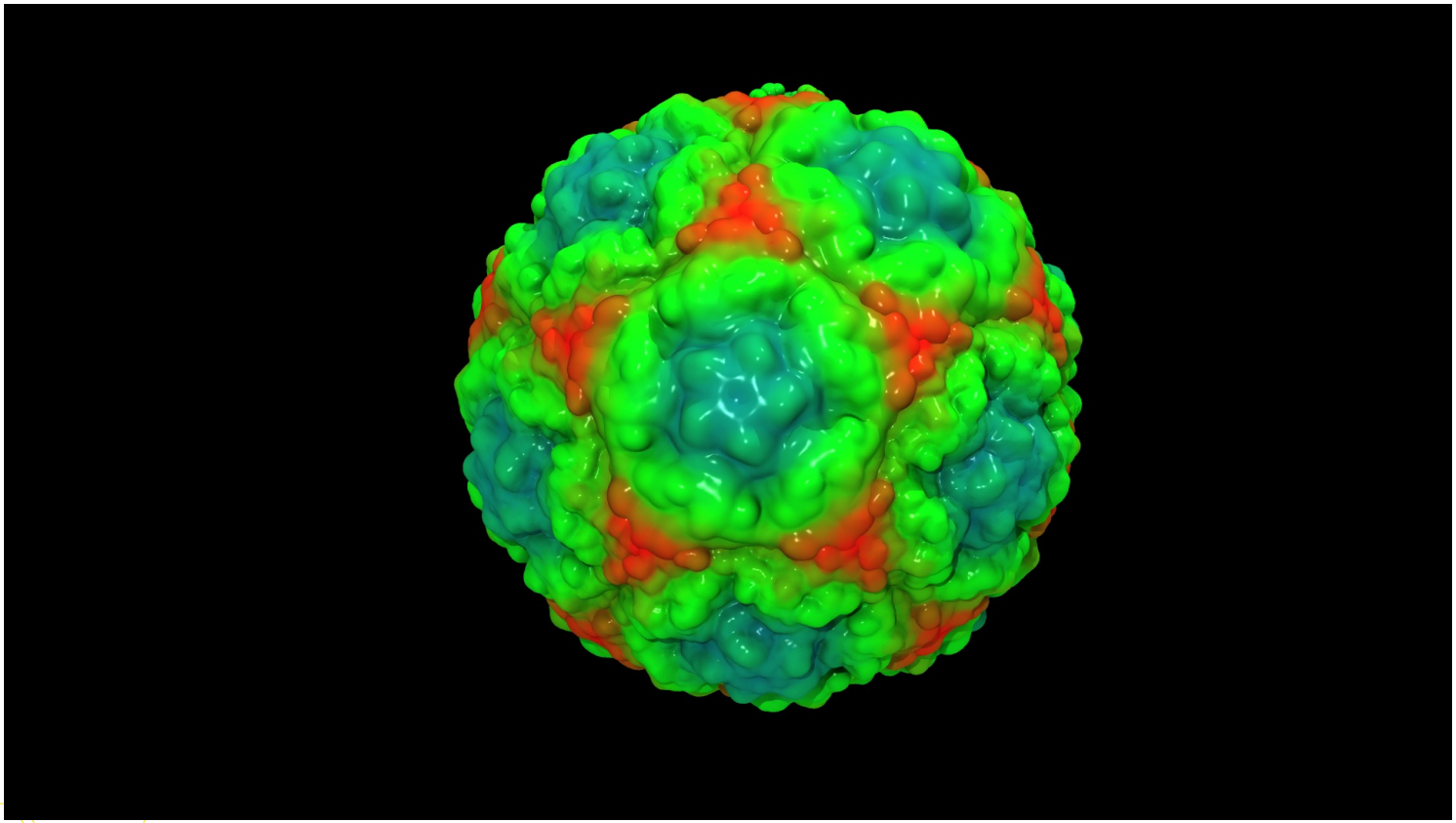
- IBM TS3500 tape library
  - 6 frames (racks)
  - 16 LTO5 drives – 1.5TB per tape native
  - 3,000 LTO5, 100 LTO4 tapes loaded
  - 180 free slots
  - >4.5 PB current capacity
  - Dual robots to load/unload/move tapes
  - For backups and hierarchical filesystem

GPFS Filesystems

- Multiple tiers of storage for projects
  - /scratch (357 TB)
    - fastest, not backed up
  - /vlsci (170TB)
    - Per-project areas with space quotas
    - Snapshots for rapid recovery, tape backups
  - /hsm (170TB disk, PB's of tape)
    - For large data needs
    - Uses TSM for hierarchical storage management

vlsci
LIFE SCIENCES COMPUTATION

Software



Software Stack

- OS: Linux (almost) everywhere
  - Red Hat Enterprise Linux 6 for HPC
  - Debian 6 and 7 for infrastructure
- HPC software
  - Compilers, MPI stacks, applications
- Cluster filesystems
- Management/deployment
  - xCAT for HPC bare metal
  - VMware for infrastructure

HPC Applications

- 626 combinations of application, compiler and versions across three architectures (Nehalem, SB, BG/Q)
  - Not inc. billions of Perl, Python & R modules
- Use “Environment modules” to allow users to choose which they want
  - Schema: application-compiler/version
  - module load bwa-intel/0.7.5a
  - http://modules.sf.net/

History of Slurm@VLSCI

- 2010: VLSCI Stage 1
  - Intel (RHEL5) clusters were Torque + Moab + Gold
  - BlueGene/P “Tambo” running Slurm + Moab + Gold
  - Moab didn't work so well on BG/P so switched to pure Slurm there in 2011

- 2012: VLSCI Stage 2
  - BlueGene/Q “Avoca” running pure Slurm
  - Intel clusters running Torque+Moab+Gold
- 2013: VLSCI Stage 2
  - New Intel cluster, decision made to go RHEL6 and pure Slurm (2.6.0)
  - Other Intel clusters transitioned later in 2013

Transitioning to Slurm

- PUSH: Painted into a corner with Torque + Moab + Gold
  - Fixed allocations to projects, hence Gold
  - Gold support dropped in Moab 7
  - No (supported) upgrade path to MAM
  - Lot of work to upgrade, plus proprietary
- PULL: BG/P then BG/Q running Slurm
  - Works, GPL, prospect of using the same system everywhere

Crash test dummy

- New x86 cluster, green field site..
- Started off with 2.6.0-pre releases
  - Needed job array support, GRES for MIC
  - Remarkably stable, moved to 2.6.0 just before going public
- Javascript job script generator on website
- Python PBS->SLURM script translator
  - Basics only, but enough for most.

Existing clusters

- We relied heavily on cpusets in Torque (in RHEL5) but Slurm needed control groups, only in RHEL6
- Reinstall from scratch (sigh)
- One week allocated downtime for each cluster, one month apart
- Each outage announcement an opportunity to educate on change

...results

- Users have coped remarkably well
  - Only a few “where is qsub?” emails
- Scheduling not as flexible as Moab
  - But seems to be coping
  - Takes time to adjust brain to match :-)
- Open-MPI 1.6 needs to use mpirun, not srun, to get good scaling
  - Fixed in 1.8, currently planning change

Current Use of Slurm

- Recently upgraded to 14.03.10
- Managing Xeon Phi cards as GRES
  - sbatch --gres=mic:2 ./myjob.sh
  - Offload only
- Rubra bioinformatics pipeline
  - Written locally, available from Github
  - Has Slurm and DRMAA branches
  - Python

Current Use of Slurm

- Lots of smaller jobs, some very big jobs
  - One user pushed through 95,000+ jobs on our BG/Q in one day.
  - Bioinformatics tends to be dominated by small tasks that form pipelines
  - Lots of job arrays too
- Balancing equitable access to resources with throughput is a hard problem

Future Plans

- 14.11.x !
  - Reporting on MIC card usage
    - Info in DB already, but sacct needed updating
  - Optimisations for large array jobs
    - Array jobs only created when running
  - BlueGene/Q enhancements
    - cnode level reservations
- Monitor cgroup/systemd development

vlsci
LIFE SCIENCES COMPUTATION

THANKS

Any questions?

www.vlsci.org.au

Australian HPC Booth #3833