

# Differences Between HPCx and CSAR

Jon Gibson

Manchester Computing  
University of Manchester



- Service overviews
  - Applying for resources
  - File systems
  - Batch systems
  - Compilers, parallelism and tools
  - Software
  - Porting codes
  - Grid
-

- Started in 1998, 6 years (plus 18 months extension)
  - PFI contract
    - value of £29M
    - CSC & SGI take financial risk and buy machine(s), decide size etc.
    - UoM is public body & not allowed to take financial risk - income is fixed
  - Grows according to user demand
    - Bid for what you need, not a portion of what exists
  - Very successful
    - PFI awards
    - 'Satisfied' customers
-

- Funded by EPSRC
  - Started in 2003
  - Six-year contract worth £53M
  - Technical support provided by EPCC & CCLRC Daresbury Lab
  - Three-phase hardware roadmap supplied by IBM
-

- Newton (SGI Altix3700/Itanium2)
- Green (SGI Origin3800/MIPS)
- Fermat (SGI Origin2000/MIPS)
- Wren (SGI Origin300/MIPS)
- Powderhorn tape silo - 125 Tbytes
- Storage Area Network, with four different disk types

## HPCx

- 96 IBM eServer 575 LPARs for computation
  - 6 IBM eServer 575 LPARs for login and disk I/O
  - Each eServer LPAR contains 16 processors
-

- **SGI Altix 3700**
    - 512 processors
    - 384 1.3GHz Itanium2 CPUs
    - 128 1.5GHz Itanium2 CPUs
    - 2.7 Tflops peak
    - 1TB shared memory
    - On the SAN
  - **Numaflex interconnect**
    - Lowest latency available
    - High bandwidth
    - Near uniform across whole system
  - **Batch and interactive access**
-

- **SGI Origin 3800**
    - 512 processors
    - 0.4GHz SGI R12000 MIPS CPUs
    - 0.41 Tflops peak
    - 512GB shared memory
    - On the SAN
  - **Numaflex interconnect**
    - Lowest latency available
    - High bandwidth
    - Near uniform across whole system
  - **Batch access**
-

- SGI Origin 300
    - 16 processors
    - 0.5GHz SGI R12000 MIPS CPUs
    - On the SAN
  - Interactive access
-

- “Class 1” applications for resources for both services done through research councils and subject to review
  - CSAR had four classes of application
    - Class 1: Full Peer Review
    - Class 2: Pump-priming
    - Class 3: New application areas
    - Class 4: Commercial use
  - HPCx currently only allows Class 1 applications
-

- CPU-hours
  - SAN disk
  - Tape storage
    - (much cheaper than disk)
  - Training Courses
  - Optimisation/Application support
  - Trading possible between resources
-

- Awarded with grant (EPSRC, BBSRC, NERC)
    - CPU time
    - Disk
    - Tape storage
  - On application
    - Optimisation/Application support
  - Free
    - Training courses
    - Helpdesk support
  - Trading between resources is not possible
-

- Home filesystems
    - /**sanhp** or /**m** on CSAR
    - /**hpcx/home** on HPCx
  - Temporary files:
    - /**santmp** on CSAR
    - /**hpcx/work** on HPCx
  - Offline archive:
    - /**hold** on CSAR
    - Archives created with TSM on HPCx
  - Cross-mounting of directories on both services
-

- **MP-SAN – medium performance disk**
    - Home directories
    - Use for compute-oriented jobs
    - The default type
    - Backed up
  - **HP-SAN – high performance disk**
    - Use for jobs with significant I/O
    - Backed up
  - **UHP-SAN – ultra-high performance disk**
    - Use for I/O bound applications
    - Not backed up
  - **HV-SAN – high volume disk**
    - Use for storing large volumes of data cheaply on disk
    - Not backed up
-

- **Homespace**
  - Home directories
  - Backed up
- **Workspace**
  - Not backed up

- All CSAR machines use LSF
    - Job script submitted using `bsub < script`
    - Jobs monitored with `bjobs` and `qs`
    - Jobs killed with `bkill <JOB_ID>`
  - HPCx uses IBM's LoadLeveler
    - Job script submitted using `llsubmit script`
    - Jobs monitored with `llq`
    - Jobs killed with `llcancel <JOB_ID>`
    - Refer to course notes for further details
-

# Example LSF Script

---

```
#BSUB -m green
#BSUB -W 12:00
#BSUB -n 256
#BSUB -o outfile
#BSUB -J test_job

mpirun -np 256 ./my_job
```

---

# Example LoadLeveler Script

---

```
#@ shell = /bin/ksh
#@ job_name = myrun
#@ job_type = parallel
#@ cpus = 256
#@ node_usage = not_shared
#@ network.MPI = csss,shared,US
#@ bulkxfer = yes
#@ wall_clock_limit = 12:00:00
#@ account_no = z001
#@ output = $(job_name).$(schedd_host).$(jobid).out
#@ error = $(job_name).$(schedd_host).$(jobid).err
#@ notification = never
#@ queue

# suggested environment settings:
export MP_EAGER_LIMIT=65536
export MP_SHARED_MEMORY=yes
export MEMORY_AFFINITY=MCM
export MP_TASK_AFFINITY=MCM

poe ./my_executable
```

---

- On CSAR
    - Development queue for jobs of up to 16 processors and 30 minutes
    - Jobs can be submitted of any size up to the capacity of the machine
  - On HPCx
    - Various queues exist – see course notes for full details
    - Maximum job size is 1024 processors
-

- Fortran 90, C/C++ and Java on all systems.
  - Use compiler options to get optimal performance.
  - SGI compilers on CSAR's Origins.
    - Use `f90` and `cc`
  - Intel compilers on CSAR's Altix.
    - Use `ifort` and `icc` for version 8 compilers.
    - Use `efc` and `ecc` for version 7 compilers.
  - IBM compilers on HPCx
    - Use `xlf_r`, `xlf90_r`, `xlc_r`, `xlc_r` and `javac` for serial codes
    - Use `mpxlf90_r`, `mpcc_r` and `mpCC_r` for MPI codes
    - Refer to course notes for further details
-

- **Message Passing**
    - Both machines provide vendor MPI
    - CSAR machine has SHMEM
    - HPCx has LAPI
  
  - **Shared Memory**
    - Both machines allow OpenMP use
    - Only possible within a node (16 processors) on HPCx
-

- **CSAR**
    - Totalview – graphical based parallel debugger
    - CVD – graphical debugger
    - Vampir – profiler for parallel codes
    - Perfex – shows hardware counters after a program run
    - Speedshop – SGI’s suite of performance tools
    - Histx – parallel profiler
  - **HPCx**
    - pdbx – parallel debugger
    - Totalview – graphical based parallel debugger
    - Vampir – profiler for parallel codes
    - Paraver - performance visualisation and analysis tool
    - gprof and xprofiler – parallel profiling tool
    - HPM toolkit – Hardware performance monitoring tools
    - KOJAK – tools for analysing performance of parallel codes
-

- Both machines have a range of software packages installed
  - Please check for specific packages
-

- **CSAR uses modules**
    - The user loads the module associated with a given piece of software
    - This sets all required paths and environment variables
    - Allows several versions of software and compilers to be available on the system with minimal risk of conflict
  - **HPCx does not use modules**
    - Paths to compilers included by default but this means that multiple versions are not available on the system
    - All other software paths and all environment variable have to be set explicitly by the user
-

- Default data sizes are the same on both machines.
  - Data storage varies
    - CSAR's Origins store data in a big-endian format
    - CSAR's Altix stores data in a little-endian format
    - HPCx stores data in a big-endian format
    - This means consideration will have to be given to moving data from Newton to HPCx, as explained in the data migration lecture
  - Some compiler options will vary between machines.
  - Non-standard Fortran extensions
    - Some may become apparent when codes ported
  - External libraries
    - May behave differently or have different routine names
-

- CSAR machines support basic Globus and Unicore access
- Globus 2.4.3 is running on HPCx
  - For details see  
<http://www.hpcx.ac.uk/services/grid>