Notos system – access and usage

PRACE Spring School 2012

CAPACITIES

e-infrastructure

Notos system - introduction

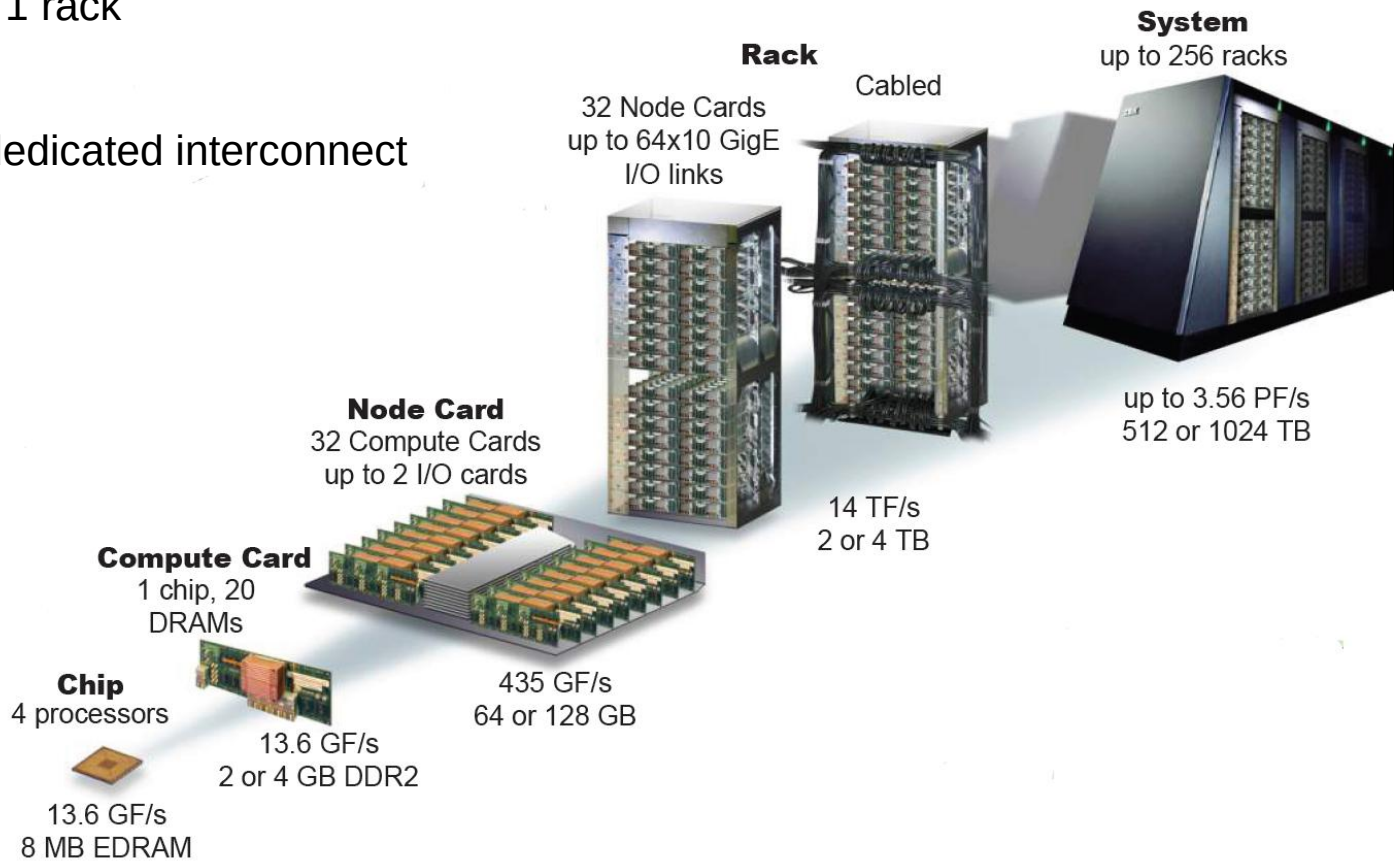
- IBM Blue Gene/P system located at ICM, University of Warsaw

- **System configuration:**
  - Single BG/P rack – 1024 compute nodes
  - Each node consists of four PowerPC 450 cores and 4GB memory

- **Software and tools:**
  - IBM XL C/C++ and Fortran compilers
  - MPI compilers
  - Libraries and scientific codes

System configuration details

- IBM Blue Gene/P – 1 rack
- ~14 Tflop/s
- High performance dedicated interconnect
- 4 TB RAM
- 178 TB storage
- Installation date:
  December 2010



Access to Notos

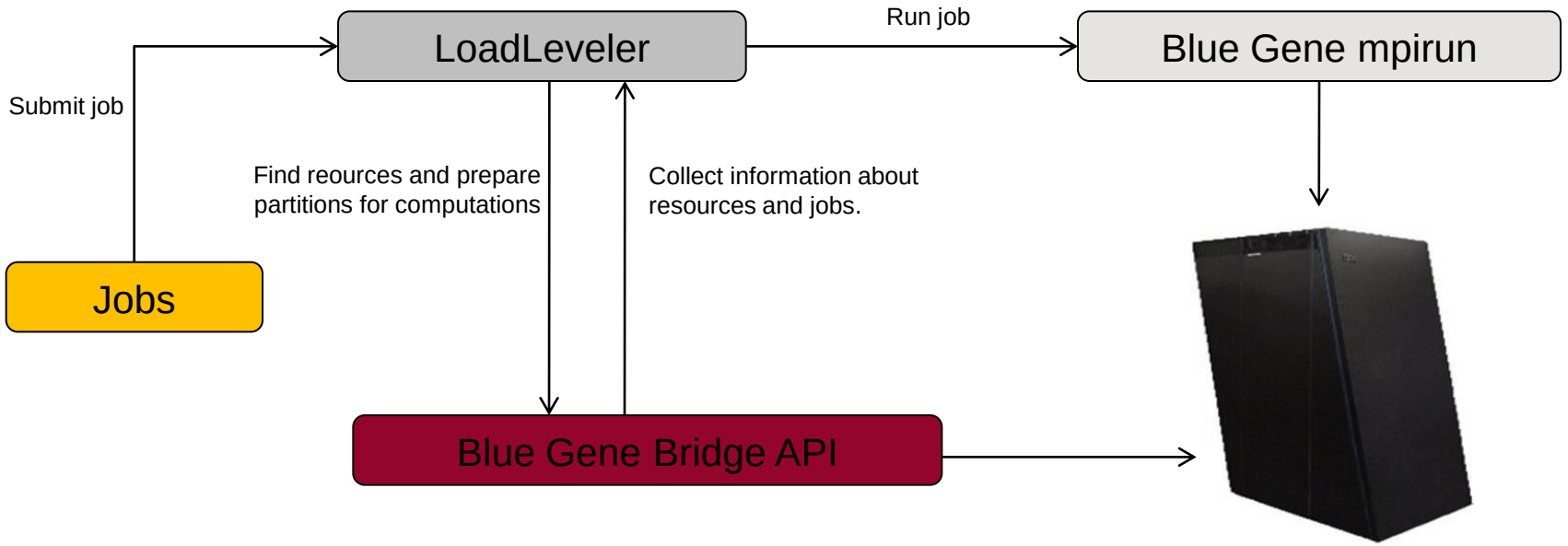
- **Login to ICM**: ssh login@atol.icm.edu.pl
- **Login to Notos**: ssh notos
- **ICM front-end system:** delta
- **X Forwarding**: add „-Y” option to both ssh commands

- **Copying data from Notos system (2 steps):**
  user@notos:~> scp file delta:~/
  user@laptop:~> scp login@atol.icm.edu.pl:file .

- **Laptops with Microsoft systems**: PuTTY and WinSCP

Running jobs – IBM LoadLeveler

- Notos resources are controlled by IBM's queueing system called LoadLeveler
- LoadLeveler consists of resource manager and scheduler
- Users define jobs in their home directory and submit them to LoadLeveler
- Users store all data (input, output,..) in their home directory



Running jobs – IBM LoadLeveler

- Basic commands:

Command	Short description
llsubmit	Submits the job to the LoadLeveler.
llq	Queries job status in the LoadLeveler queues. Prints additional information about running and submitted jobs.
llq –s <job_id>	Prints extended information about jobs: where the job is running, when it was submitted, why it is not running…
llcancel <job_id>	Removes the job from the LoadLeveler.
llclass	Prints information about available classes.

- Submitting jobs: *llsubmit <jobfile name>*

Example command: llq

```
sheed@notos:~> llq
Step Id                   Owner       Account  Job Name             Class       Size   ST
------------------------- ----------- -------- -------------------- ----------- ------ --
notos.1904.0              panecka     G31-4    R53A_v3_b_P          kdm-large          I
notos.1903.0              panecka     G31-4    R53A_3_b_nP          kdm-large          I
notos.1902.0              panecka     G31-4    f_3_b_P              kdm-large          I
notos.1901.0              panecka     G31-4    K42A_3_b_nP          kdm-large          I
notos.1900.0              panecka     G31-4    full_v3_b_nP         kdm-large          I
notos.1899.0              panecka     icm-staf f_v3_99_nP           kdm-large          I
notos.1898.0              panecka     G31-4    K42A_v3_b_P          kdm-large          I
notos.1897.0              panecka     G31-4    K43Q_v3_b_P          kdm-large          I
notos.1896.0              panecka     G31-4    K43Q_3_b_nP          kdm-large          I
notos.1890.0              memar       icm-staf nwchem1              workshop   32      R
notos.1863.0              fleon       G31-4    namd_ab              kdm-large  512     R

11 job step(s) in queue, 9 waiting, 0 pending, 2 running, 0 held, 0 preempted
```

Example command: llq -s

```
sheed@notos:~> llq -s 1863
=============== Job Step notos.icm.edu.pl.1863.0 ===============
        Job Step Id: notos.icm.edu.pl.1863.0
           Job Name: namd_ab
              Owner: fleon
         Queue Date: Fri 03 Jun 2011 11:05:13 AM CEST
             Status: Running
      Dispatch Time: Tue 07 Jun 2011 07:00:47 PM CEST
     Size Requested: 512
     Size Allocated: 512
Partition Allocated: LL11060703200913
Base Partition List: R00-M0
     IONodes Per BP: N00-J00,N01-J00,N02-J00,N03-J00,N04-J00,N05-J00,N06-J00,N07-J00,N08-J00,N09-J00,N10-J00,N11-
    J00,N12-J00,N13-J00,N14-J00,N15-J00
        Notify User: fleon@icm.edu.pl
  LoadLeveler Group: G31-4
              Class: kdm-large
Wall Clk Hard Limit: 8+08:00:00 (720000 seconds)
Wall Clk Soft Limit: 8+08:00:00 (720000 seconds)
            Account: G31-4

==================== EVALUATIONS FOR JOB STEP notos.icm.edu.pl.1863.0 ====================
The status of job step is : Running
Since job step status is not Idle, Not Queued, or Deferred, no attempt has been made to determine why this job
     step has not been started.
```

LoadLeveler – queue script structure

- LoadLeveler queue scipt - text file consisting of LoadLeveler parameters
- LoadLeveler parameter lines begin with #@
- Script files can include additional standard bash script commands (Note: executed on Front End Node)
- Comment lines begin with #
- Most important part of the script: mpirun command

LoadLeveler – queue script structure

Required parameters:

Parameter	Description
# @ **job_type = bluegene**	Defines job type. Should be always set to „bluegene”.
# @ **bg_size = N**	Number of Blue Gene/P computational nodes requested, partition size.
# @ **account_no = grant no**	Users computational grant ID. Should be set to „G47-17” during PRACE Spring School 2012.
# @ **wall_clock_limit = HH:MM:SS**	Limits the amount of real computational time that the job can run for.
# @ **class = class**	Defines the class name. Should be set to „prace” during PRACE Spring School 2012.
# @ **output = file**	Defines the name of the output file (standard output). Default /dev/null.
# @ **error = file**	Defines the name of the error file. Default /dev/null.

LoadLeveler – queue script structure

Optional parameters:

Parameter	Description
# @ **bg_connection = MESH/TORUS/PREFER_TORUS**	Defines the connection type between nodes. Default: MESH.
# @ **environment = env1; env2; ..**	Defines the method for copying the environment variables to computational environment. Available methods: • COPY_ALL – copy all the variables, • $var - copy variable var and its value to computational environment, • !var – do not copy variable var and its value to computational environment, • var=value - variable var should be set to value in computational environment.
# @ **job_name = job name**	Defines the job name.
# @ **initialdir = dir**	Defines the work directory.

LoadLeveler – queue script structure

Optional parameters:

Parameter	Description
# @ **notification = typ**	Defines the method of user notification on the job status. Possible attributes: • error – notify when job has ended with an error, • start – notify when job has started, • complete – notify when job has completed, • always – always notify, • never – never notify.
# @ **notify_user = email**	User email address. Format: user[@host][,user[@host],...].

Example script

This example can be found in /opt/prace/loadl

```
# @ job_name = Hello_World
# @ account_no = G47-17
# @ class = prace
# @ error = hello.err
# @ output = hello.out
# @ environment = COPY_ALL
# @ wall_clock_limit = 00:20:00
# @ job_type = bluegene
# @ bg_size = 32
# @ queue
```

mpirun -exe hello_world -mode VN -np 128

Basic mpirun options *(mpirun –h)*

Option	Description
-np ranks	Number of MPI ranks.
-exe <executable>	Executable file name.
-args „program args”	Program arguments.
-cwd <path>	Work directory.
-mode <SMP,DUAL,VN>	Blue Gene/P execution mode. • SMP – 1 rank, 4 threads • DUAL – 2 ranks, 2 threads each • VN – 4 ranks, 1 thread each

Relationship between execution mode, partition size and number of MPI ranks:

- VN mode: number of MPI ranks = 4 x partition size
- DUAL mode: number of MPI ranks = 2 x partition size
- SMP mode: number of MPI ranks = partition size

Computational environment - libraries

Library	Usage	Installation directory
Message Passing Interface (MPI)	module load mpi_fast module load mpi_default	/bgsys/drivers/ppcfloor/comm/fast /bgsys/drivers/ppcfloor/comm/default
Mathematical Acceleration Subsystem (MASS) – optimized basic mathematical functions (cos,sin,powf,…)	-lmass -lmassv	/opt/ibmcmp/xlmass
Engineering and Scientific Subroutine Library (ESSL) – optimized mathematical and scientific functions (linear algebra, FFT, random number generator,…)	**C/C++**: -lesslbg –lesslsmpbg -lxlf90_r -lxlopt -lxl –lxlfmath **Fortran**: -lesslbg -lesslsmpbg	/opt/ibmmath
FFTW (v2.1.5,v3.2.2) single + double	module load fftw_2.1.5 module load fftw_3.2.2	/opt/fftw/2.1.5 /opt/fftw/3.2.2
LAPACK v3.1	-L/opt/lapack/lapack_bgp.a	/opt/lapack
CBLAS	/opt/cblas/cblas_bgp.a	/opt/cblas
BLACS	/opt/blacs/blacsCinit_MPI-BGP-0.a /opt/blacs/blacsF77init_MPI-BGP-0.a /opt/blacs/blacs_MPI-BGP-0.a	/opt/blacs
SCALAPACK	/opt/scalapack/libscalapack.a	/opt/scalapack
HYPRE v2.7.0b	-L/opt/hypre/lib –lHYPRE –I/opt/hypre/include	/opt/hypre

Computational environment - libraries

Library	Usage	Installation directory
HDF5 (1.8.3)	/opt/hdf5/XL/lib	/opt/hdf5/XL
NetCDF (4.0.1), Parallel-NetCDF (1.1.1)	/opt/netcdf/XL/lib /opt/parallel-netcdf/lib	/opt/netcdf /opt/parallel-netcdf
GNU Scientific Library (GSL)	/opt/gsl/lib	/opt/gsl
PETSc (3.0.0-p7)	/opt/petsc/bgp-ibm-opt/lib/	/opt/petsc /opt/petsc/bgp-ibm-opt
Zoltan	/opt/zoltan/XL/lib	/opt/zoltan
P3DFFT (2.3.2)	-L/opt/p3dfft/lib –lp3dfft	/opt/p3dfft

Compilers

- Compilation on Blue Gene/P system is achieved with the use of cross-compilers available on front end node
- C, C++ and Fortran compilers available

- **XL compilers family: XL C/C++ and XL Fortran**
  - Specific Blue Gene/P code optimizations
  - Double FPU optimizations, hierarchical memory optimizations

- **GNU compilers family: C, C++ and Fortran**
  - No specific Blue Gene/P code optimizations
  - Do not allow OpenMP parallelization

Cross-compilers

Very important notice

- Compilation and linking is achieved on front end node:
  - POWER6, 64-bit architecture
- Binaries are executed on Blue Gene/P computational nodes:
  - PowerPC 450, 32-bit architecture

- Running Blue Gene/P binaries on front end node will fail.

Compiler names

- **MPI default**
  - default MPI compilers
  - run „module load mpi_default" to setup paths
  - `mpixlc, mpixlcxx, mpixlf{77|90|95|2003}, mpixlc_r, mpixlcxx_r, mpixlf{77|90|95|2003}_r`

- **MPI fast**
  - MPI with no error checking implemented
  - run „module load mpi_fast" to setup paths
  - `mpixlc, mpixlcxx, mpixlf{77|90|95|2003}, mpixlc_r, mpixlcxx_r, mpixlf{77|90|95|2003}_r`

Notice: `mpicc, mpicxx, mpif77` is based on GNU compiler

Notice: use **mpixl*** wherever possible (even for scalar codes)

Please check your environment…

- Login to ICM
- Login to Notos
- Copy example LL job **(/opt/prace/loadl**) into your local folder
- See the **README** file to complete testing