HPC Resources at UVT

BlueGene/P Supercomputer

Access
Overview
Usage

Silviu Panica, Alexandra Popescu
bgsupport@hpc.uvt.ro
http://hpc.uvt.ro/

Universitatea de Vest din Timişoara

Overview

- BlueGene/P Supercomputer
  - access information
  - technical details / architecture
  - how to use it
- For developers
  - parallel/multi-core environment
  - compiling tips

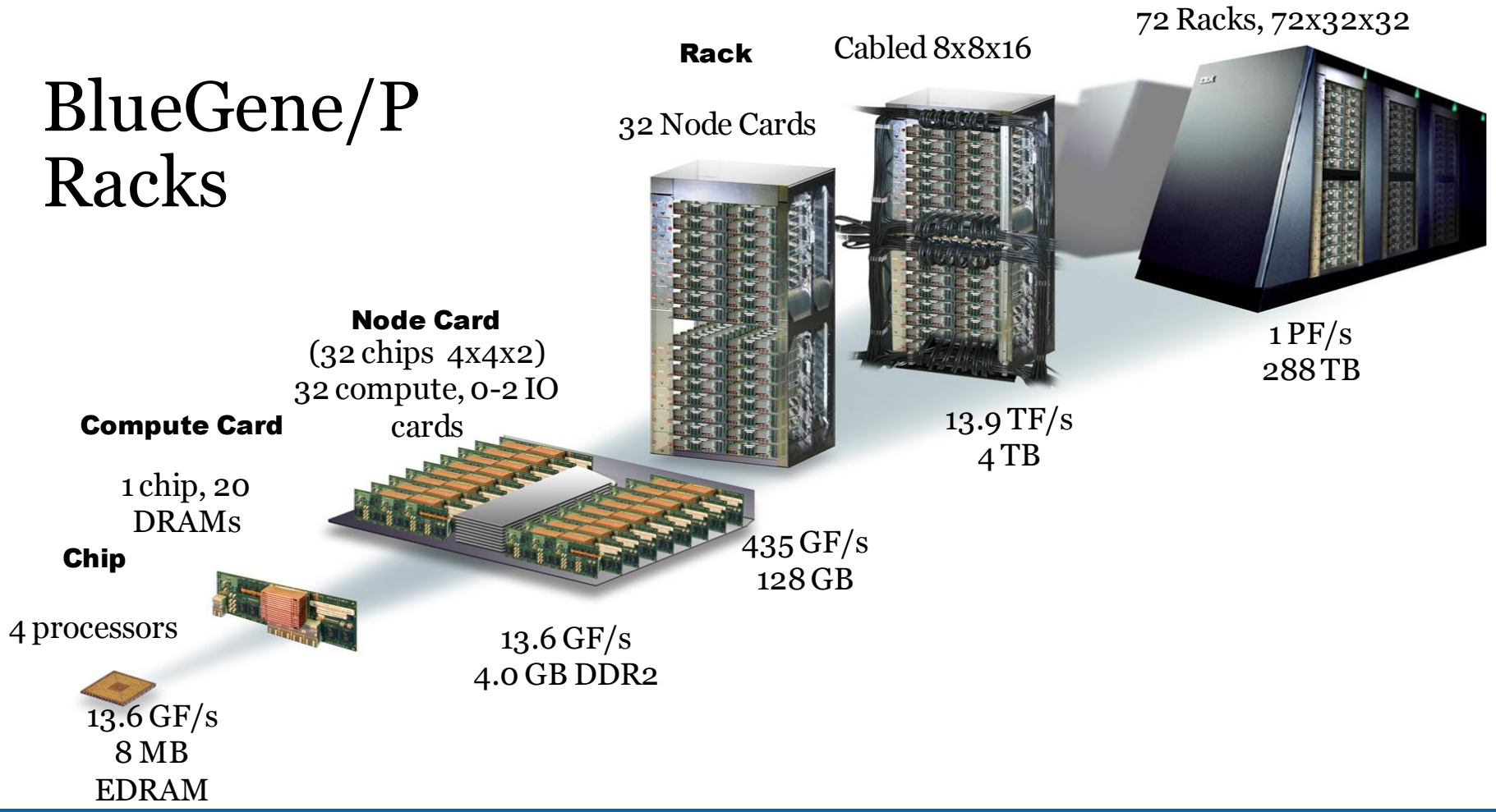
BlueGene/P Supercomputer

- supercomputer information:
  - https://hpc.uvt.ro/infrastructure/bluegenep/
- „open“ access;
- sign-up form;
- resource scheduling:
  - no special policy: *based on res. availability;*
- **one simple request**: acknowledge the use of BG/P;

BlueGene/P Supercomputer

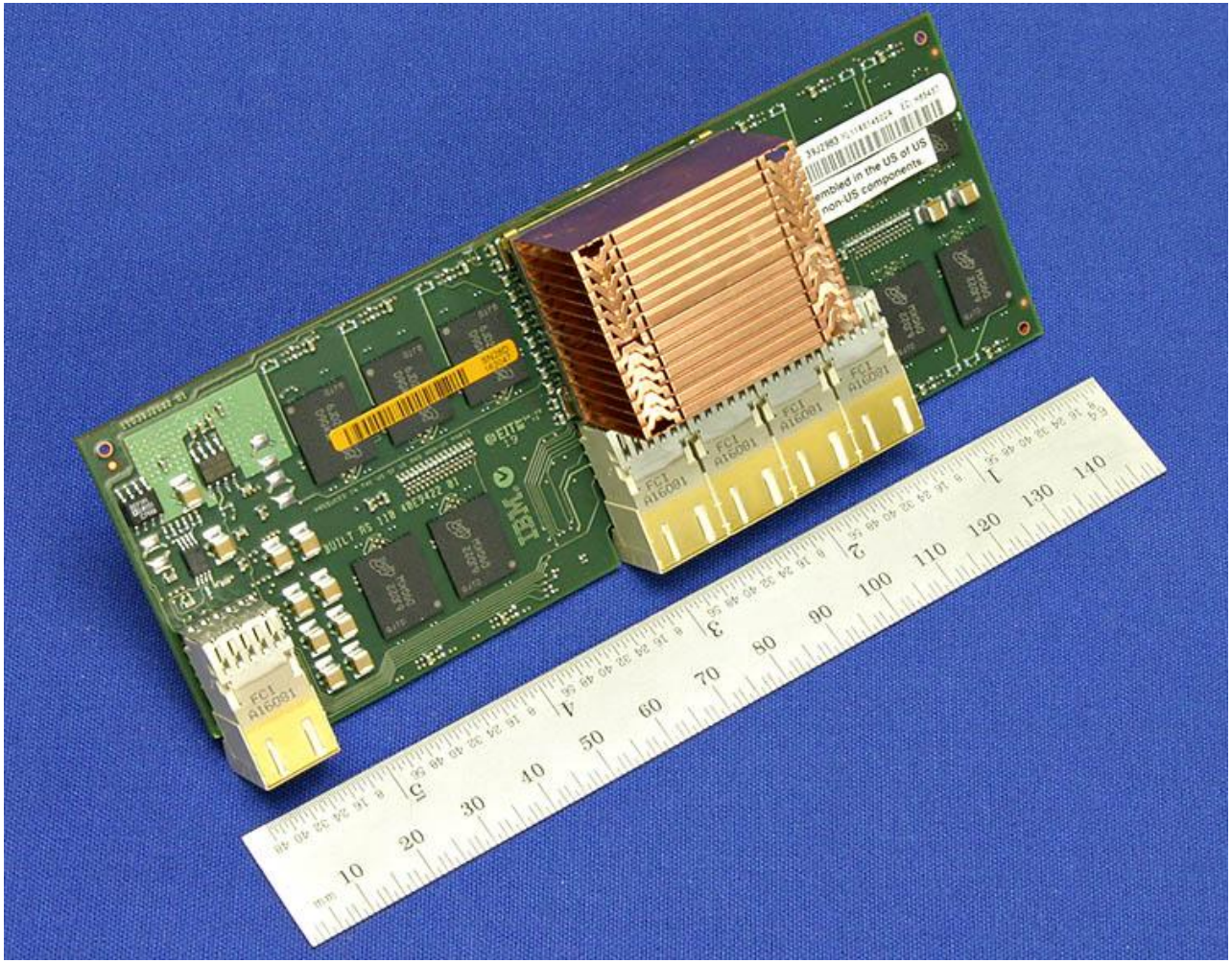
- technical details:
  - 1x rack
  - 32x compute nodes
    - 32x compute cards
    - ~4TB RAM
  - 1024x compute cards
    - 4x cores – IBM 450 PowerPC
    - 4GB RAM
  - high-speed and scalable inter-connect;

BlueGene/P Supercomputer

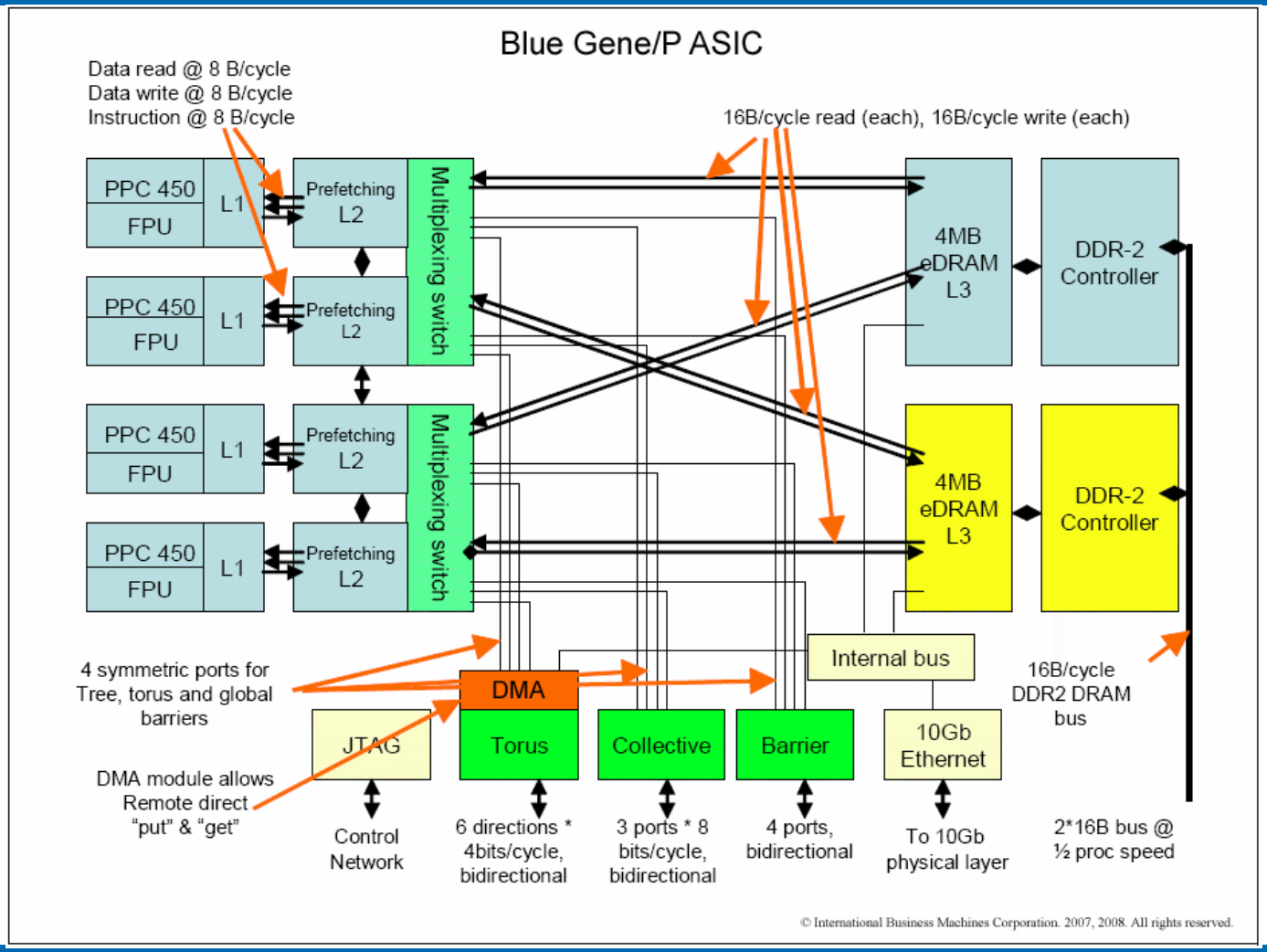


BlueGene/P Supercomputer

Compute card



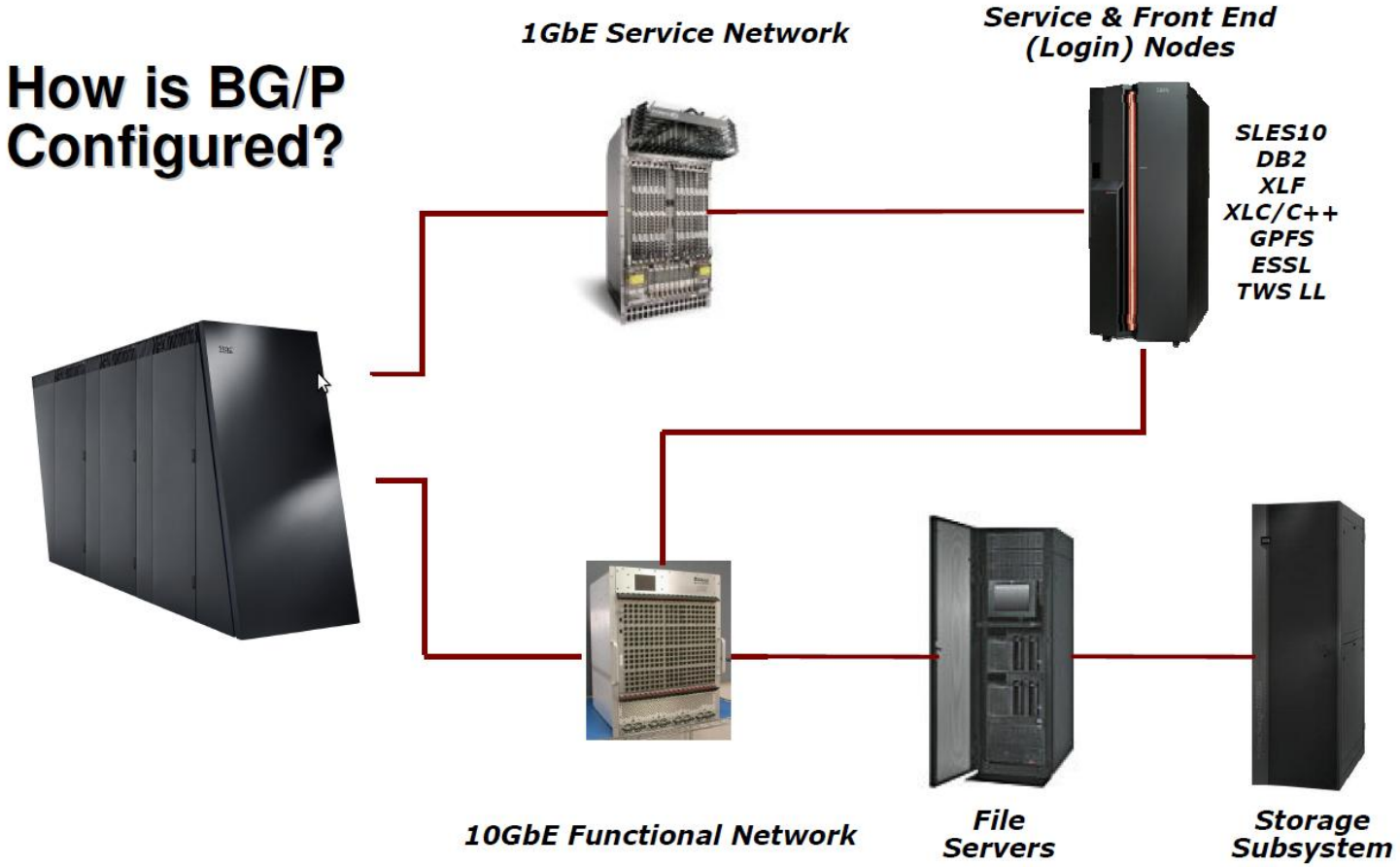
BG/P - Architecture



BlueGene/P Supercomputer

Node card

BG/P - Configuration

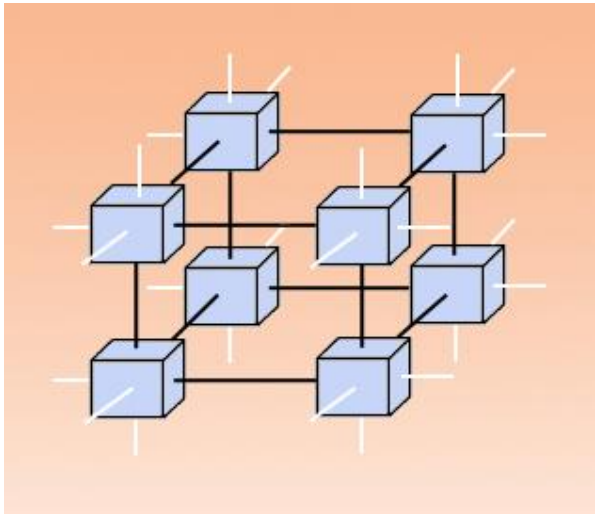


BG/P - Numbers

Node processors :	quad-core 450 PowerPC
Processor frequency:	850 MHz
Coherency:	Symmetrical multiprocessing
L1 Cache (private):	32 KB per core
L2 Cache (private)	14 stream pre-fetching
L3 Cache size (shared)	8 MB
Main store memory/node:	4GB
Total number of CPUs:	1024
Main store memory bandwidth:	16GBps
Peak performance:	13.6TFlops
Sustained performance:	11.7 TFlops
Storage:	28TB

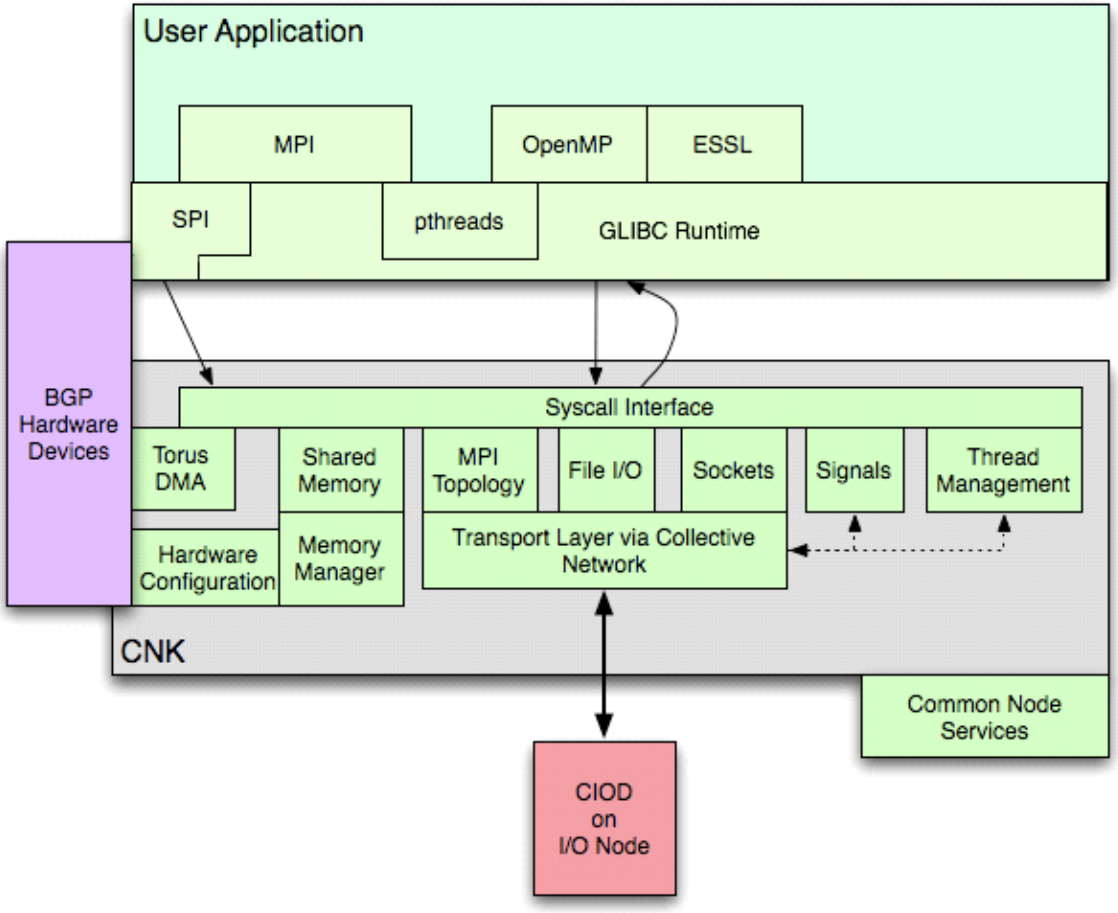
BG/P Inter-connect

- 3D Torus
  - MPI point-to-point comm.;
  - every compute node is connected to each six neighbors;
  - **bandwidth:** 3.4Gb/s bidirectional * 6 links/node; 5.1GiB/s per node;
  - **latency:** 3μs – 1 hop; 10μs farthest;
  - **routing:** adaptive/dynamic hardware;
  - DMA support;
- Global collective
  - MPI Broadcasts (one-to-all and all-to-all);
  - MPI Reduction;
  - interconnects all compute and I/O;
  - **bandwidth**: 6.8Gb/s bidirectional * 3 links/node;
  - **latency:** 5μs – tree traversal (one way);
- Global Barrier/Interrupt
  - connects all compute nodes;
  - low latency for MPI: 1.3μs to reach 72k nodes (max BG/P configuration)
  - bandwidth: 3.4Gb bidirectional * 4 links/node (not so important; no data carried)
- I/O network
  - 10Gbps Ethernet: connects I/O nodes with the storage system;



BG/P – CNK

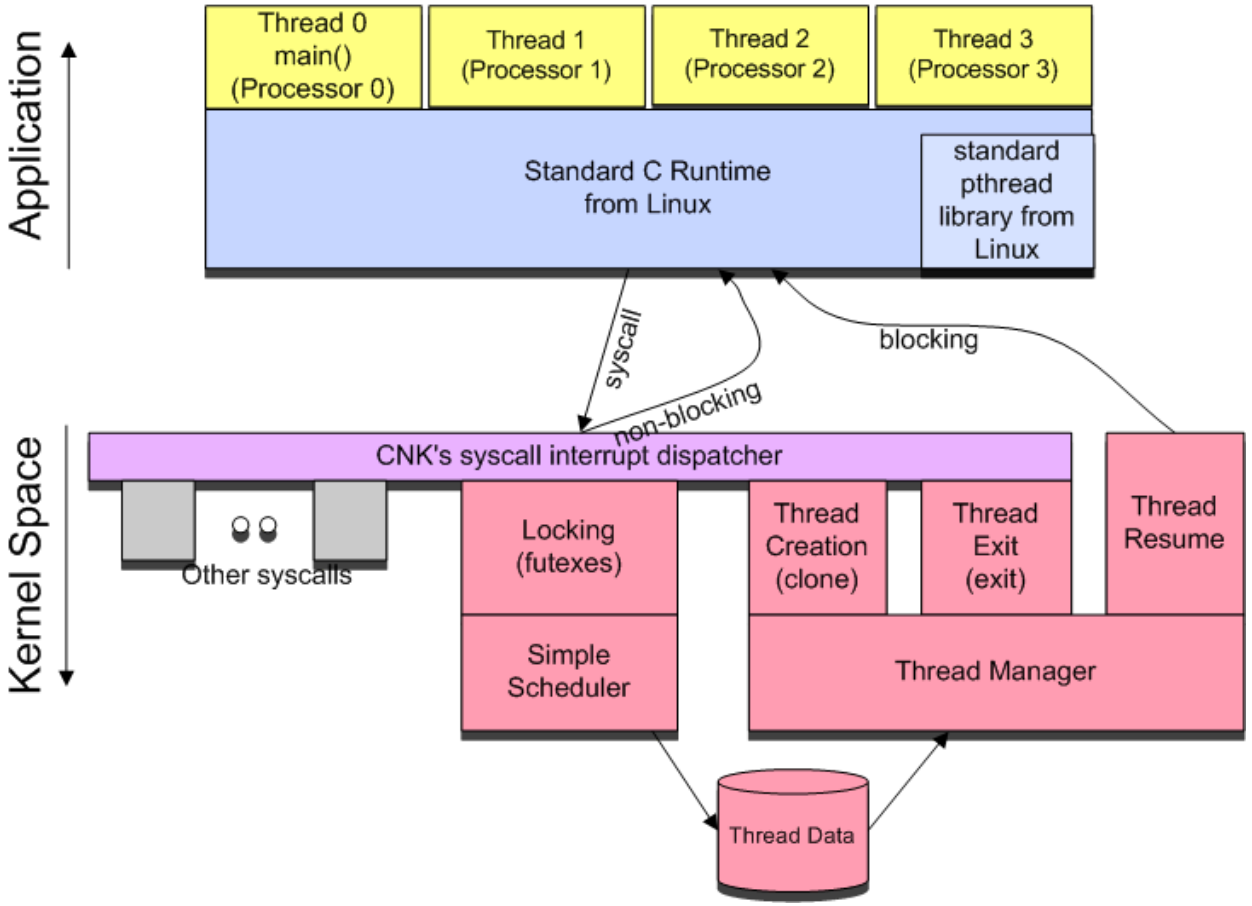
Compute Node Kernel



BG/P – Execution modes

SMP

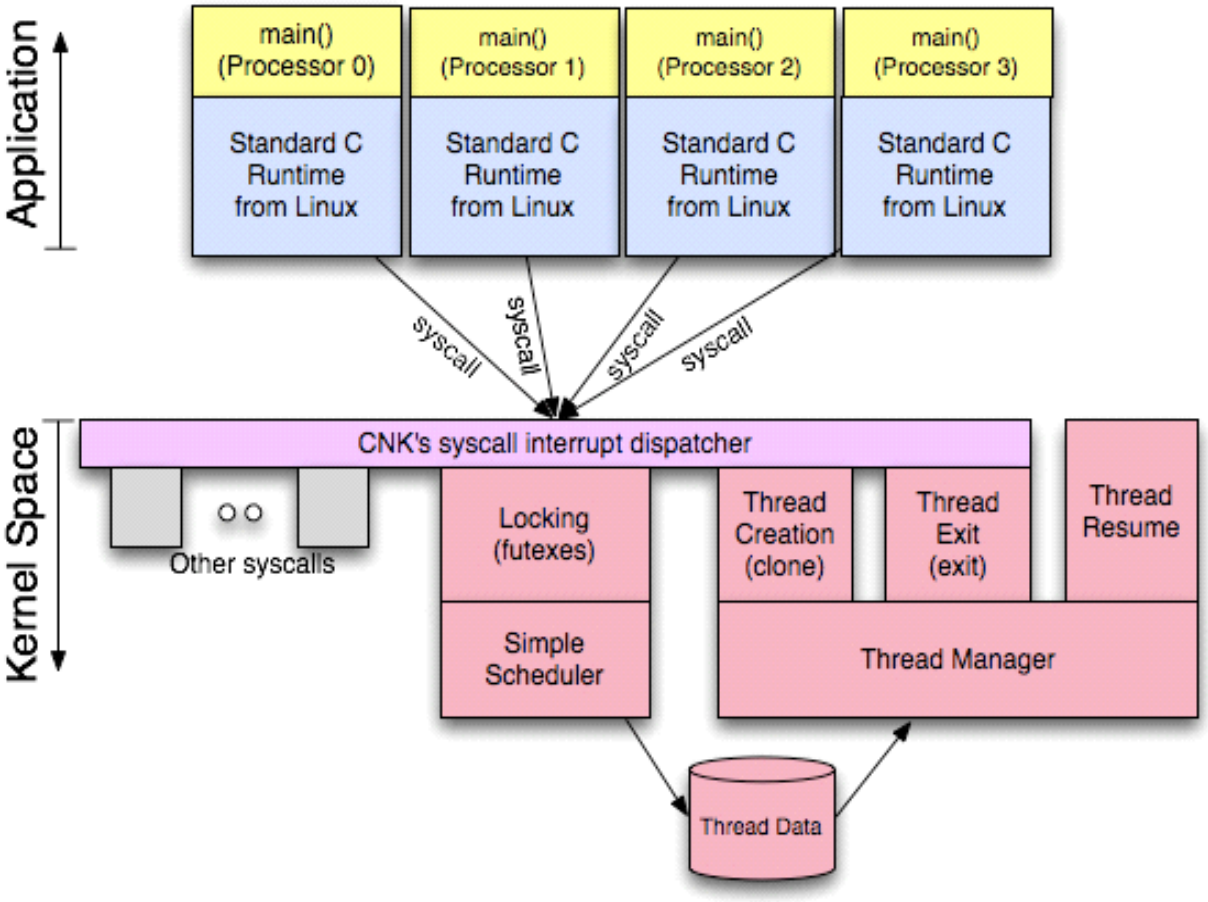
- 1 MPI per node
- up to 4 Pthreads/OpenMP
- maxim memory availability
- running only one kernel image;



BG/P – Execution modes

VN – Virtual Node

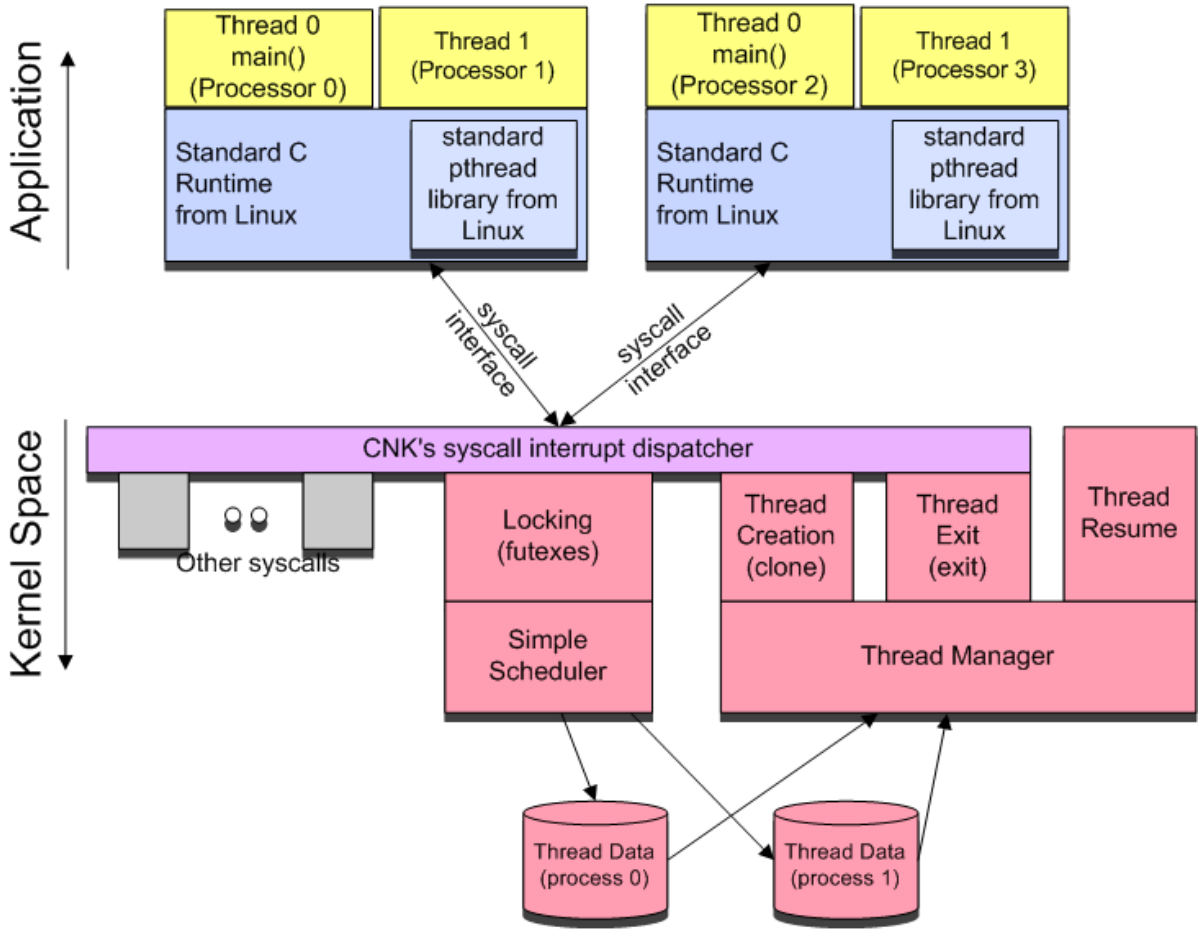
- 4 MPI per node
- NO threads
- 1GB memory available for each process;
- running one kernel image for each process;
- (L3 cache / 2) for each two cores;
- (memory bandwith / 2) for each two cores;



BG/P – Execution modes

DUAL - Dual Mode

- 2 MPI per node
- up to 2 Pthreads/OpenMP
- 2GB memory for each process;
- running one kernel image for each process;



BG/P – How to use it

- front-end node access:
  - username@ui.hpc.uvt.ro (ssh)
- CLI access;
- web interface for information:
  - **https://sn.hpc.uvt.ro:32072/BlueGeneNavigator/**
- compilers:
  - C/C++ and Fortran;
  - python limited support;
- job submission:
  - *job scheduler:* LoadLeveler;
  - *direct submit:* mpirun

BG/P - Environment

http://hpc.uvt.ro/wiki/ BlueGene - up to date

- **user_home:** /u/invites/username
- **storage_partition:** $user_home
- **compilers location:**
  - Fortran
    - /opt/ibmcmp/xlf/bg/11.1/bin/
  - C
    - /opt/ibmcmp/vac/bg/9.0/bin/
  - C++
    - /opt/ibmcmp/vacpp/bg/9.0/bin/
- **Optimized libraries:**
  - IBM ESSL (Engineering and Scientific Subroutines)
    - /bgsys/ibm_essl/sles10/prod/opt/ibmmath/essl/4.4/
  - IBM MASS (Mathematical Acceleration Subsystem)
    - /opt/ibmcmp/xlmass/bg/4.4/bglib
    - /opt/ibmcmp/xlmass/bg/4.4/include
- **Custom libraries** (users)
  - /u/sdk/bg/
- **module support**: soon!

BG/P - compilers

- supported compilers:
  - IBM XL C/C++
    - standard C/C++
      - bgxlc
      - bgxlC
    - MPI-2
      - mpixlc (_r - thread safe);
      - mpicxx
  - IBM XL Fortran
    - standard Fortran:
      - bgxlf
    - MPI-2
      - mpixlf[70,77,90,95,2003] (_r - thread safe)
- the code must be statically built

BG/P – job submission

Load leveler

```
#!/bin/sh

# @ job_name = sfcmGridFragm_2iunie
# @ job_type = bluegene
# @ requirements = (Machine == "$(host)")
# @ error = $(job_name)_$(jobid).err
# @ output = $(job_name)_$(jobid).out
# @ environment = COPY_ALL;

# @ notification = always
# @ notify_user = silviu@info.uvt.ro

# @ wall_clock_limit  = 3:59:00

# @ class = parallel
# @ bg_size = 32
# @ queue

/bgsys/drivers/ppcfloor/bin/mpirun -mode VN –np 128 -cwd "/u/path/to/dir" -args
"config_sfcm.cnf 4 4" -exe /u/path/to/exe
```

BG/P – job submission

Load leveler

- llclass
- llstatus
- llsubmit job_desc
- llq
  - -l – verbose information
- llcancel

BG/P – job submission

mpirun

```
mpirun -partition R00-M0-64-2 \
       -cwd ~/projects/bgp/loadl/sample_1 \
       -mode SMP \
       -np 64  \
       -exe ~/projects/bgp/loadl/sample_1/sample_1
```

- **partition:** a group of resources (CPUs, I/O nodes) allocated to an user;
- partition must exist! (check BlueGeneNavigator)
- if cancelling run ctrl+c once and wait!
- partitioning support – soon!

Documentation

- https://hpc.uvt.ro/wiki/BlueGene/
- IBM RedBooks (free for download):
  - BlueGene/P for System Administration
    - id: sg247417
  - BlueGene/P Application Development
    - id: sg247287

BG/P – For developers

MPI

- MPI-2 implementation;
  - -I/bgsys/drivers/ppcfloor/comm/include
  - mpi.h
  - mpif.h
- MPI_X extension to support BG/P personality (CPU mapping);
  - mpix.h

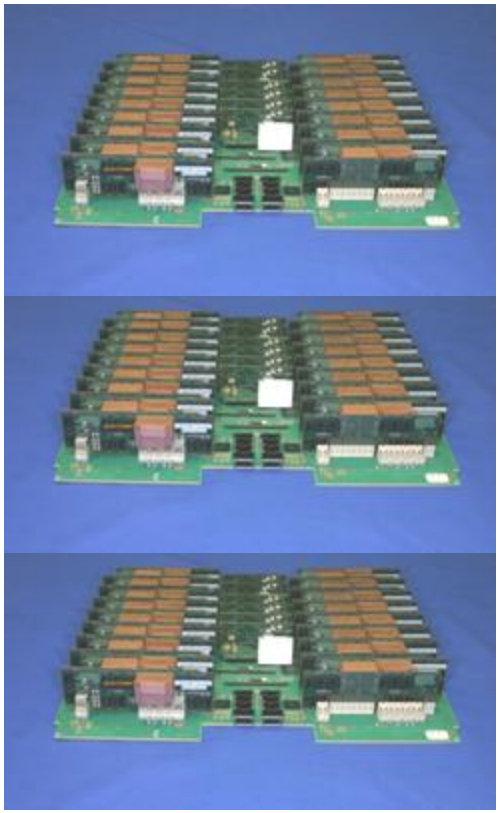
BG/P – For developers

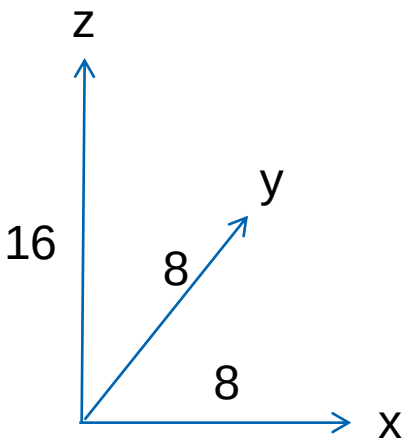
- personality example:

```
#include <spi/kernel_interface.h>
#include <common/bgp_personality.h>
#include <common/bgp_personality_inlines.h>

_BGP_Personality_t personality;
int myX, myY, myZ, coreID;

Kernel_GetPersonality(&personality, sizeof(personality));

myX = personality.Network_Config.Xcoord;
myY = personality.Network_Config.Ycoord;
myZ = personality.Network_Config.Zcoord;
coreID = Kernel_PhysicalProcessorID();
node_config = personality.Kernel_Config.ProcessConfig;
pset_size = personality.Network_Config.PSetSize;
pset_rank = personality.Network_Config.RankInPSet;
BGP_Personality_getLocationString(&personality, location);

printf("MPI rank %d has torus coords <%d,%d,%d> cpu = %d, location = %s\n", rank, myX, myY, myZ, coreID,
location);
```

BG/P – For developers

Mapping





BG/P – For developers

OpenMP

- OpenMP 2.5;
  - -qsmp=omp to activate;
  - -qsmp – activates auto parallalization;
    - activates –O2 –qhot (use –qsmp=omp:noopt to disable)
- use _r* compilers for thread safe compilation support;

BG/P – For developers

- compiler optimization:
  - default: NONE (very very slow);
  - -O: first level,
    - –qmaxmem=128000 (amount of mem. for memory-intensive optimization)
  - -O3 –qstrict: more aggressive optimization but it doesn't modify program semantics;
  - -O3: aggressive; replace division by multiplication with the inverse; allows re-association etc.;
  - -qhot: high-order transformation module; adds vector routines (-qhot=novector to disable); optimizes loops; etc.
  - -qipa: inter-procedure analysis; check documentation for further sub-options; ex. –qipa=level=2
  - -qinline: inline subroutines to reduce overhead;

BG/P – For developers

- compiler optimization
  - architecture flags:
    - -qarch=450: single FPU -> standard floating point code;
    - -qarch=450d: double FPU;
    - test you application with both flags (450d sometimes needs more time to compute if not used correctly);
    - -qtune=450 (by default comes with –qarch);
- BG/P recommendations:
  - start with: -g -O -qarch=450 -qmaxmem=128000;
  - next: -O3 -qarch=450/450d
  - next: -O4 -qarch=450/450d (even –O5)
    - -O4 = -O3 -qhot -qipa=level=1
    - -O5 = -O3 -qhot -qipa=level=2
  - basic debugging: -qdebug=diagnostic (only with –qhot)

Thank you!

QUESTIONS ?!