Porting Applications to Blue Gene/P

Dr. Christoph Pospiech  pospiech@de.ibm.com
Agenda

- What beast is this?
- Compile - link – go!
- MPI subtleties
- Help! It doesn't work (the way I want)!
Blue Gene/P

- **System**: 72 Racks
  - 1 PF/s
  - 144 TB

- **Rack**: Cabled 8x8x16
  - 14 TF/s
  - 2 TB

- **Node Card**: (32 chips 4x4x2)
  - 32 compute, 0-1 IO cards

- **Compute Card**: 1 chip, 20 DRAMs
  - 435 GF/s
  - 64 GB

- **Chip**: 4 processors
  - 13.6 GF/s
  - 8 MB EDRAM
  - 13.6 GF/s
  - 2.0 (or 4.0) GB DDR
Blue Gene/P Spider Webs

3 Dimensional Torus
- Interconnects all compute nodes (73,728)
- Virtual cut-through hardware routing
- 3.4 Gb/s on all 12 node links (5.1 GB/s per node)
- 0.5 µs latency between nearest neighbors, 5 µs to the farthest
- MPI: 3 µs latency for one hop, 10 µs to the farthest
- Communications backbone for computations
- 1.7/3.9 TB/s bisection bandwidth, 188TB/s total bandwidth

Collective Network
- One-to-all broadcast functionality
- Reduction operations functionality
- 6.8 Gb/s of bandwidth per link
- Latency of one way tree traversal 1.3 µs, MPI 5 µs
- ~62TB/s total binary tree bandwidth (72k machine)
- Interconnects all compute and I/O nodes (1152)

Low Latency Global Barrier and Interrupt
- Latency of one way to reach all 72K nodes 0.65 µs, MPI 1.6 µs

Other networks
- 10Gb Functional Ethernet
- I/O nodes only
- 1Gb Private Control Ethernet
- Provides JTAG access to hardware. Accessible only from Service Node system
You have the choice!

Node
- core
- core

Software Abstractions Blue

Quad Mode (VNM)
- 4 Processes
- 1 Thread/Process

P0
- T0
- T1
- P2
- T0
- P1
- T0
- P3
- T0

Dual Mode
- 2 Processes
- 1-2 Threads/Process

P0
- T0
- T1
- P1
- T0
- P1
- T1

SMP Mode
- 1 Process
- 1-4 Threads/Process

P0
- T0
- T2
- T1
- T3

P0
- T0
- T2
- T1

P0
- T0
- T2
- T1
Dual FPU Architecture

- SIMD instructions over both register files
- FMA operations over double precision data
- Parallel (quadword) loads/stores
- Data needs to be 16-byte aligned
# Caches

<table>
<thead>
<tr>
<th>Cache</th>
<th>Total per node</th>
<th>Size</th>
<th>Replacement Policy</th>
<th>Associativity</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1 Instruction</td>
<td>4</td>
<td>32 KB</td>
<td>Round-Robin</td>
<td>64-way set-associative 16 sets 32B line size</td>
</tr>
<tr>
<td>L1 Data</td>
<td>4</td>
<td>32 KB</td>
<td>Round-Robin</td>
<td>64-way set-associative 16 sets 32B line size</td>
</tr>
<tr>
<td>L2 Prefetch</td>
<td>4</td>
<td>14x256 B</td>
<td>Round-Robin</td>
<td>Fully associative (15-way) 128 B Line size</td>
</tr>
<tr>
<td>L3</td>
<td>2</td>
<td>2x4 MB</td>
<td>Least Recently Used</td>
<td>8way associative 2 Bank Interleaved 128 B Line</td>
</tr>
</tbody>
</table>
Jitter-free Execution

- Compute node runs nothing but application
- I/O delegated to I/O nodes
- Cross-Compiling on the front end node
Software Stack in Blue Gene Compute Node

- Compute Node Kernel (CNK) controls access to hardware, and enables bypass for application use.
- User-space libraries and applications can directly access torus and collective network through bypass.
- Application code can use all processors in a compute node.
Processing Sets (Psets)

- I/O node dedicated to a fixed group of compute nodes
- Compute to I/O ratio is fixed within a partition
  - 128:1, 64:1, 32:1, 16:1
I/O Node Kernel

- SMP Linux
- No persistent store (network filesystems only; no swap)
- 10Gb Ethernet interface
- Several CNK System calls are function shipped to here
  - Linux compatibility by executing these syscalls on Linux
  - Function ship occurs over Collective network
  - The `ciorg` daemon manages a fixed set of compute nodes in a processing set (pset)
  - Linux provides the portable filesystem and network layer interfaces
Agenda

- What beast is this?
- **Compile - link – go !**
- MPI subtleties
- Help ! It doesn't work (the way I want) !
Getting started is simple

- ...for simple cases
- **BGP_SYS=/bgsys/drivers/ppcfloor**
- `make CC=$BGP_SYS/comm/bin/mpixlc_r`
- `llrun -mode VN -np 512 ./hello_par`
IBM XL Compilers for Blue Gene

- **XLF 11.1/VACPP 9.0** will be the compiler releases
  - /opt/ibmcmp/xlf/bg/11.1/bin
  - /opt/ibmcmp/vaccpp/bg/9.0/bin

- **Differences in this release:**
  - xlf2003 (the 2003 Fortran standard) is available
  - BGP wrapper names are different
    - blrts_ is replaced by bg
    - bgxlf, bgxlc, bgcc, etc.
  - On BG/L for xlf 11.1/vacpp 9.0 both blrts_ and bg will be supported.
    - -qarch=450d/450 are accepted in addition to 440d/440
Some key options for IBM compilers

- **-qarch=440, 450** generates only instructions for one floating point (option minimal option with blrts_)
- **-qarch=440d, 450d** generates only instructions for 2 floating point pipes
- **-qtune=440**
- **-O3 (-qstrict)** minimal level for SIMDization
- **-O3 –qhot (=simd)**
- **-O4 (-qnoipa)**
- **-O5**
- **-qdebug=diagnostic** provide details about SIMDization, only with – qhot
- **-qreport –qlist –qsource** provide pseudo-assembler code .lst
What’s new from BG/L …

- pthreads and OpenMP support
- Dynamic linking
- Use of mmap for shared memory
- Protected readonly data and application code
- Protection for stack overflow
- Full socket support (client and server)
ESSL for Blue Gene

- Engineering and Scientific Subroutine Library
- Optimization library and intrinsics for better application performance
- Serial Static Library supporting 32-bit applications
- Callable from FORTRAN, C, and C++
- SMP support and ppc450 tuning done for BG/P
- libesslbg.a (.so) and libesslsmplibg.a (.so)
Lib Mass for Blue Gene

- **Mathematical Acceleration Subsystem (MASS) libraries** consists of libraries of tuned mathematical intrinsic functions

- **Location:**
  - `/opt/ibmc/mp/xlmass/bg/4.4/bglib`
    - `libmass.a` `libmassv.a`
  - `/opt/ibmc/mp/xlmass/bg/4.4/include`
MPIRUN implementation on BGP

- no rsh/ssh mechanism
- Option -free
- STDIN handling
Partitioning

- Subdivision of a single Blue Gene system
- Partitions are software defined
- Torus, Collective and Barrier networks are completely isolated from traffic from other partitions
- A single job runs on a partition
  - i.e. jobs never share resources or interfere with each other
- Custom kernels may be booted in a partition
Agenda

- What beast is this?
- Compile - link – go!
- MPI subtleties
- Help! It doesn't work (the way I want)!
Blue Gene/P Spider Webs

3 Dimensional Torus
- Interconnects all compute nodes (73,728)
- Virtual cut-through hardware routing
- 3.4 Gb/s on all 12 node links (5.1 GB/s per node)
- 0.5 µs latency between nearest neighbors, 5 µs to the farthest
- MPI: 3 µs latency for one hop, 10 µs to the farthest
- Communications backbone for computations
- 1.7/3.9 TB/s bisection bandwidth, 188TB/s total bandwidth

Collective Network
- One-to-all broadcast functionality
- Reduction operations functionality
- 6.8 Gb/s of bandwidth per link
- Latency of one way tree traversal 1.3 µs, MPI 5 µs
- ~62TB/s total binary tree bandwidth (72k machine)
- Interconnects all compute and I/O nodes (1152)

Low Latency Global Barrier and Interrupt
- Latency of one way to reach all 72K nodes 0.65 µs, MPI 1.6 µs

Other networks
- 10Gb Functional Ethernet
- I/O nodes only
- 1Gb Private Control Ethernet
- Provides JTAG access to hardware. Accessible only from Service Node system
Collectives

- Use the hardware support in the collective network and global interrupt networks

- Supported operations
  - Barrier
  - Broadcast
  - Allreduce
  - Alltoall
  - Allgather
Messaging Framework

Multiple programming paradigms supported
MPI, Charm++, ARMCI, GA, UPC (as a research initiative)

SPI : Low level systems programming interface
DCMF : Portable active-message API
Agenda

 What beast is this?
 Compile - link – go!
 MPI subtleties
 Help! It doesn't work (the way I want)!
Debugger Interfaces

- **ptrace-like interfaces available via ciod**
  - non-parallel: gdbserver for direct use with gdb
  - parallel: Totalview, or other tools

- **lightweight core files**
  - each node writes a small file with regs, traceback, etc
  - superset of parallel tools consortium format
  - Use addr2line for translating HEX into source lines

- **Coreprocessor**
Coreprocessor

All details in SG24-7417 IBM Blue Gene Solution: System Administration

Located in /bgsys/drivers/ppcfloor/tools/coreprocessor

Originally written to view lightweight core files

Can attach to a hanging program
Coreprocessor showing program counter on 4 racks
GNU Debugger

- Simple debug server call «gdbserver »
- Only one gdb instance for one compute node (to debug multiple CNs at the same time you need to launch multiple GDB clients)
- Limited subset of primitives (however enough to be useful)
- Standard Linux gdb client, not aware about Double FPU.
- Gdserver must start before the application; mpirun has a special option «-start_gdbserver »
- Compile and link with –g (–O2)
- Location: /bgsys/drivers/ppcfloor/gnu-linux/bin
Performance Tools

- Low level SPI provided to configure, reset and read hardware perf counters
- PAPI interface to the perf counters
- HPC Toolkit
- Considering addition of per-job performance metrics recorded via job history
- Unix gprof command (compiler with `-g -pg`)
IBM High Performance Computing Toolkit on Blue Gene

- CPU performance: Xprofiler, HPM
- MPI performance: MPI Profiler/Tracer
- Visualization and analysis: PeekPerf
- Work in progress
  - High Productivity Computing System Toolkit (HPCST)
  - Currently only for AIX operating system
More Information

- IBM Redbooks for Blue Gene
  - Application Development Guide
  - System Administration Guide
  - Performance Tools
- Open Source Communities (Argonne website, …)
- BlueGene Rochester website
- Doxygen documentations (DCMF, SPI, …)