BlueGene Application Performance Optimization

Roy Musselman
Other Contributors: Brent Swartz, Kurt Pinnow, Dave Hermsmeier
BlueGene Software Development
IBM Rochester, Minnesota

ScicomP 12
IBM System Scientific Computing User Group
Boulder, CO, July 18-21, 2006
Agenda

- BlueGene Software Development
- BlueGene System Performance Characteristics
  - Optimized for Computation, Communication, & IO
  - V1R3 Features and Enhancements
- Application Optimization
  - Compiler Enhancements
  - Toolchain Enhancements
- Performance Analysis Tools
- Resources
BlueGene Software Development

- Collaboration of teams
  - STG at Rochester, Minnesota
  - IBM Research at T.J. Watson Research Center
  - IBM XL Compilers, Toronto Lab
- Project Management
- Kernel development
- Control system development
- Diagnostic development
- Communication libraries (MPI, ARMCI, etc.)
- Compiler and Toolchain enhancements
- Software driver build, test, package, and release
- Extensive functional and performance testing
- Application enablement (Life Sciences)
- Manufacturing tests
- Staff the BlueGene advocate program
Optimized for Computation

- **Compute Node Architecture**
  - Two embedded PowerPC 440 processor cores
  - Two dual 16 byte FPUs (Double Hummer)
  - On-chip Cache Hierarchy
    - L1 (32KB Instruction, 32KB Data)
    - L2 (16 128-byte lines used as prefetch buffer)
    - L3 (4MB)
  - Up to 1 GB main memory per node

- **Selectable L1 Cache Runtime Behavior**
- **Benchmark results**
  - STREAM
BlueGene/L Compute System-on-a-Chip ASIC

- **PLB (4:1)**
  - 2.7GB/s

- **32k/32k L1**
  - 440 CPU
  - Double FPU

- **440 CPU I/O proc**
  - Double FPU

- **L2**
  - 128
  - 256

- **Ethernet**
  - Gbit
  - JTAG

- **JTAG Access**
  - 2.8 Gbit/s link

- **Torus**
  - 6 out and 6 in, each at 1.4 Gbit/s link

- **Tree**
  - 3 out and 3 in, each at 2.8 Gbit/s link

- **Global Interrupt**
  - 4 global barriers or interrupts

- **Shared L3 directory for EDRAM**
  - Includes ECC
  - 1024+144 ECC
  - 22GB/s

- **4MB EDRAM**
  - L3 Cache or Memory

- **DDR Control with ECC**
  - 5.5 GB/s
  - 144 bit wide DDR 256MB

- **5.5GB/s**

- **11GB/s**

- **256**

- **5.6GF peak node**
Flexible L1 Data Cache Runtime Behavior

- Default L1 data cache modes: Store with Allocate and Write-back
- Use environment variables to select:
  - Store with Allocate (SWA) vs. Store without Allocate (SWOA)
  - Write-back (WB) vs. Write-Through (WT)
- Store without Allocate: `BGL_APP_L1_SWOA=1`
  - For store operations, if there is a L1 cache miss, the L1 cache is bypassed and the data is stored directly to lower levels of the memory subsystem.
- Write-Though: `BGL_APP_L1_WRITE_THROUGH=1`
  - For store operations, data is written to L1 and lower level caches and L1 is marked as clean.
- Some applications may see improved performance depending on their memory access patterns.
- Users are encouraged to benchmark their applications to find the optimum settings.
- Mileage will vary. Performance improvements are not guaranteed!
BlueGene STREAM Performance

BlueGene/L STREAM Performance
Main Memory, N=2000000

<table>
<thead>
<tr>
<th>Modes / Test</th>
<th>Copy</th>
<th>Scale</th>
<th>Add</th>
<th>Triad</th>
<th>Copy</th>
<th>Scale</th>
<th>Add</th>
<th>Triad</th>
<th>Copy</th>
<th>Scale</th>
<th>Add</th>
<th>Triad</th>
</tr>
</thead>
<tbody>
<tr>
<td>WT=0,SWOA=0</td>
<td>V1R2M2_060328</td>
<td>V1R3M0_060623</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT=0,SWOA=1</td>
<td>V1R2M2_060328</td>
<td>V1R3M0_060623</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT=1,SWOA=0</td>
<td>V1R2M2_060328</td>
<td>V1R3M0_060623</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT=1,SWOA=1</td>
<td>V1R2M2_060328</td>
<td>V1R3M0_060623</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
STREAM Performance

BlueGene/L STREAM Performance
L1 Cache, N=1000, V1R3M0

BlueGene/L STREAM Performance
L3 Cache, N=5000, V1R3M0

BlueGene/L STREAM Performance
Main Memory, N=500000, V1R3M0

BlueGene/L STREAM Performance
Main Memory, N=2000000
Dual Floating Point Unit (“Double Hummer”)

- One Dual FPU for each PPC440
- Two Dual FPU’s per CN chip
- 32 8-byte Primary registers
- 32 8-byte Secondary registers
- Careful use of compiler options and data alignment is necessary to effectively utilize this feature.
- Codes with significant instruction level parallelism can benefit.
  - Matrix multiply
  - Complex arithmetic

"Exploiting the Dual Floating Point Units in Blue Gene/L."
Optimized for Communication

- **Multiple communication networks**
  - Targeted to specific message passing operations
  - Network interfaces are integrated into the compute nodes

- **Communication API Libraries**
  - MPI, KEKAPI (QCD), ARMCI, Global Arrays
  - Use the appropriate networks efficiently

- **Communication Performance Results**

- **Interrupt-Driven mode available in V1R3**
Multiple Communication Networks

- **3D Torus Network**
  - Each node has 6 neighbors (+-X,Y,Z)
  - Virtual cut-through hardware routing
  - Primary network for point-to-point messages
  - Ideal for 3D and QCD applications

- **Global Collective Network**
  - Optimized for Broadcast and Reduction operations
  - Connection between IO node and Compute nodes

- **Global Barrier & Interrupt Network**
  - Very low latency
BlueGene/L Barrier Latency

Latency (Seconds)

System size (number of nodes)

512
1K
2K
4K
8K
16K

8.0E-07
9.0E-07
1.0E-06
1.1E-06
1.2E-06
1.3E-06
1.4E-06
Interrupt-driven Communication

- Interrupt-driven mode was added in V1R3 to support one-sided communication mechanisms such as ARMCI and Global Arrays.
- Can also potentially increase performance of applications that use non-blocking point-to-point operations
  - MPI_Isend, MPI_Irecv, MPI_Wait
- New environment variable available in V1R3. **BGLMPI_INTERRUPT**
- Four options:
  - **Y** - turn on both send and receive interrupts.
  - **N** - turn off both send and receive interrupts.
  - **S** - turn on only send interrupts (e.g., MPI_Isend() can interrupt to copy data to the network).
  - **R** - turn on only receive interrupts (e.g., data coming in can generate an interrupt to copy the data from the network).

- It is very difficult to provide concrete guidelines that specify when enabling interrupts will result in improved performance.
- Users are encouraged to benchmark their applications with and without interrupts enabled. Use profiling tools to see the effect on communication time.
BlueGene/L Interrupt-Driven Performance
Application=MILC, 32 processors
Results obtained using the MPI_profiler tool

Mode / Range of Processor Performance

Interrupts Disabled (both S & R) Interrupts Enabled (both S & R)
Optimized for I/O

- **Scalable Configurations**: Compute / IO Node ratio
  - 8, 16, 32, 64 to 1.
- **IO node specs**
  - Max bandwidth per IO Node = 125 MB/s (1 Gb/s Ethernet)
- **Streaming IO (Sockets) performance**
  - Driving very close to max available (~120 MB/s)
  - IO can scale linearly due to parallel IO
- **V1R3 file and socket I/O performance enhancements**
  - Single Compute node IO performance has doubled.
  - A memory copy operation was eliminated during read, readv, recv and recvfrom system calls by receiving the data directly to the user’s buffer.
  - The default read/write buffer size was increased from 87600 bytes to 262144 bytes
  - New CIOD environment variables to fine tune file system performance
Application Optimization: Compiler Improvements

- New XL compilers for BlueGene
  - XL C/C++ V8.0 Advanced Edition for Blue Gene
  - XL Fortran V10.1 Advanced Edition for Blue Gene
  - PTF1 supports the release V1R3 toolchain
- Compiler improvements including:
  - Loop transformations at –O3 level
  - Improved performance of quad precision floating point
  - Tuning for specific benchmarks, 440d complex arithmetic, and MASS library
  - New SIMDization features and tuning items
- “BlueGene Compilers and Optimization” by Allan Martin and Mark Mendell, IBM Toronto Lab
- “Using XL Compilers for Blue Gene”
Application Optimization: Toolchain Improvements in V1R3

- **gcc**: from 3.2 to 3.4.3.
  - The new version contains significant improvements in compiler
- **binutils**: from 2.13 to 2.16.1
- **glibc**: from 2.2.5 to 2.3.6
- **gmon/gprof** enhancements
  - BG/L generates 1 gmon.out file per processor
  - Files can now be saved in an alternate format to conserve space.
    - Controlled by an environment variable
    - Can be converted back to the standard gmon file format
  - gprof can now handle the large number of files generated by the largest BGL configuration
- **All user libraries that an application links with must be recompiled if the application itself is recompiled using V1R3.**
  - Tip: If you see error: “1498: undefined reference to ‘__ctype_b’,” then you have not rebuilt all libraries your application links with. Rebuild those libraries and then retry linking your application
- **V1R2 binaries may still possibly run on a system running with V1R3**
BlueGene Performance Analysis Tools

- **High Performance Computing (HPC) Toolkit**
  - MPI_profiler – Tabulates time spent in MPI functions
  - MPI_tracer – Captures time-stamped events for later analysis via Peekperf
  - Xprofiler – GUI to view the breakdown of elapsed time based on function calls
  - Peekperf – GUI to browse trace output and correlate events back to source code

- **External Performance Instrumentation Facility**
  - Available within BlueGene/L V1R3 driver
  - Count hardware events in the processor, memory, and network components.
  - No application alteration other than linking in a performance counter library

- **Several others**

Spiral in on Application Optimization

- Problem Decomposition & Algorithm Design, Targeting BG/L
  - Task size & workload
  - Mapping Tasks to nodes to minimize communication
- Use optimized function libraries: MASS, MASSV, BLAS, ESSL
- Profiling – use tools to find where time is being spent?
  - Communication: MPI_profiler, MPI_tracer, Peekperf
  - Computation & IO: gprof, Xprofiler
- Feature exploitation
  - Compiler optimization levels
  - Dual FPU & SIMD utilization
  - Runtime environment switches
- Redbook: “BlueGene/L Application Development”
- “BlueGene/L Optimizing Tips” from Bob Walkup, IBM Research
Resources: Support Website

Resources: Redbooks
References

1. “BlueGene Compilers and Optimization” by Allan Martin and Mark Mendell, IBM Toronto Lab
2. “Using XL Compilers for Blue Gene”
3. “BlueGene/L Optimization Tips” by Bob Walkup, IBM Research
4. "Exploiting the Dual Floating Point Units in Blue Gene/L." by Mark Mendell, IBM Toronto Lab
5. “Compiling for the Double Hummer” by Mark Mendell, IBM Toronto Lab