Sequoia Code Development Challenges and Tools Infrastructure

CScADS Tools Workshop
June 26, 2012

Dong H. Ahn and Matthew P. LeGendre
Livermore Computing Division
We are on track to deliver Sequoia in 2012.

- **Sequoia statistics**
  - 20 petaFLOP/s target
  - Memory 1.5 PB, 4 PB/s bandwidth
  - 1.5M cores
  - 3 PB/s link bandwidth
  - 60 TB/s bi-section bandwidth
  - 0.5–1.0 TB/s Lustre bandwidth
  - 50 PB disk

- 9.6MW power, 4,000 ft²

- Third generation IBM BlueGene

- **Challenges**
  - Hardware scalability
  - Software scalability
  - Applications scalability

- **Sequoia 20 petaFLOP/s**
  - Deliveries completed on April 16
  - Unclassified science throughout 2012
  - Acceptance: September 2012
  - Classified: January 2013
  - Tri-lab production use follows

**Acceptance Criteria:**

- **UQ**: Run 24 simultaneous Purple-class Integrated Design Code calculations while also running…
- **Weapons science**: at 4 PF sustained
Sequoia builds on 96 racks of the IBM Blue Gene /Q architecture.

1. Chip
   16 cores

2. Module
   Single chip

3. Compute card
   One single chip module, 16 GB DDR3 memory

4. Node card
   32 compute cards, Optical modules, link chips, torus

5a. Midplane
   16 node cards

5b. I/O drawer
   8 I/O cards
   8 PCIe Gen2 slots

6. Rack
   2 midplanes
   1, 2, or 4 I/O drawers

7. System
   20 PF/s
Threading is key to BG/Q’s energy efficiency and performance.

- Mitigates single-issue, in-order processor constraints
  - Four HW threads/core maximizes function unit utilization
  - Latency hiding
  - More opportunities than out-of-order execution

- Supported by new hardware features
  - 16 L2 instructions supporting fast barriers, work queues, etc.
  - Transactional memory
  - Speculative execution

- Works well with other BG/Q technologies
  - 4-wide SIMD unit
  - Prefetching
Problems that arise at large-scale are the rule rather than the exception.

- It took less than 5 minutes in resolving errors that occurred with 24 Sequoia racks.
- Efficient mechanisms like STAT to diagnose and to fix Sequoia development challenges are key to success.
- Co-design of various BGQ tools interfaces such as CDTI has been essential to enabling those tools on Sequoia.

STAT showing an outliner on a hang with 393,216 cores on Sequoia.
CDTI – Code Development and Tools Infrastructure

- Standard Debugging Operations
- New Types of Capabilities:
  - Scalability
  - Call Stack Unwinding
  - File IO
  - Multi-tool Support
  - Tool Launching
Architecture and Scalability

- 3rd Party Tools run on IO-Node
  - Linux
  - Network Connectivity
  - Away from application
Architecture and Scalability

- Message Interface between Tool and CN
  - Packages 16 commands into each message
  - Commands: Debug Operations, Stackwalks, File IO, …

- Asynchronous communication
Asynchronous Communication allows parallelism

- IO Node/Process ratio: 1:2048 (common), 1:8192 (theoretical max)
Parallel Performance

**Multiple 4k Reads Per Message**

Cost to send 16 commands per message only x2 cost to send 1.

**Read 4k data from each CN**

Cost to read from 128 CNs only x7 cost to read from 1 CN.
Stack Unwinding via CDTI

- Single command to collect call stack
  - CNK collects call stack, returns it via CDTI.
  - Lower latency than traditional: pause, read register, read mem, read mem, read mem, ...

- Uses basic stack walking techniques
  - No DWARF or binary analysis

- Trade off: Scalability vs. Accuracy
Stack Unwinding Performance

Time to Stackwalk Job

- 1 Thread/Process
- 2 Threads/Process
- 4 Threads/Process
- 8 Threads/Process
- 16 Threads/Process

Time in Milliseconds

Threads Per BG/Q Rack
Ramdisk IO

- Asynchronous File Reads
  - Read file data, `stat` files, get directory contents.
  - Peak read speed of ~2GB/Sec

- Useful for getting data off CN’s ramdisk
  - Trace data
  - Checkpoint data (current plans for SCR)
Multi-Tool Support

- Attach up to Four Tools at once
- Each Tool has **Priority 0-99**
  - Highest priority tool has **Control Authority**
  - Other tools have read access
Operations Available

Control Authority

- Receive Events
- Stop/Continue Threads
- Install Breakpoints
- Allocate/Free Memory
- Send/Manage Signals
- Write Memory
- Write Registers

Read Access

- Get Thread Data (inc. Callstack)
- Get Process Data
- Ramdisk IO
- Read Memory
- Read Registers
Job Launch

- Launches Tools via MPIR and `start_tool`
  - MPIR provides app’s process table
  - `start_tool` runs processes on IO-Node

![Launch Time of Application and Tool](image)
CDTI Status

- Supported by ProcControlAPI and LaunchMON tool components.
- Used by TotalView

- Interface Available from IBM Redbook:
  Blue Gene/Q Code Development and Tools Interface
  at
Transactional Memory and Speculative Execution

- Expect shift towards threading in BG/Q

- New BG/Q Hardware Features
  - Transactional Memory
    - Synchronization mechanism, alternative to lock
  - Speculative Execution
    - Auto-parallelization of loops
Transactional Memory

- Fast Synchronization Primitive
  - Allows multiple threads into critical region
  - Dynamically detect and roll back conflicts
  - Low overhead in no conflict case

- Ideal when conflicts are non-zero, but rare.

//Remove element from list
lock()
elem = head;
head = head->next;
unlock();

#pragma tm_atomic

//Remove element from list
elem = head;
head = head->next;
Speculative Execution

- Auto-parallelize loops
  - Breaks loops into tasks
  - Runs tasks in parallel
  - Auto-detects dependencies between tasks

- Ideal when there are some conflicts, but not too many.

SE:

```c
#pragma speculative for private(i) schedule (static)
for (i=0; i < num_regions; i++) {
    regions[i] = update_region(i);
}
```
SE and TM Performance Questions

- What critical sections in my code would benefit from TM?
- Are my TM regions performing well?
- What loops would benefit from SE?
- Are my SE loops performing well?
TM/SE Performance Counters

- TM/SE Counters
  - Transactions/Commits
  - Retries
  - Serialization
  - ...

- Details forthcoming
Summary

- Sequoia brings in new challenges and capabilities.
  - BGQ CDTI provides new kinds of functionality for tools.
  - Need tool support for TM/SE.

- Users need our help!