Robert W. Wisniewski  
Chief Software Architect  
Blue Gene Supercomputer Research  
On behalf the Blue Gene Team

BlueGene/Q:  
Architecture, CoDesign;  
Path to Exascale
Blue Gene/Q

Industrial Design

BQC DD2.0

32 Node Board

5D torus

4-rack system
Top 10 reasons that you need Blue Gene/Q

1. **Ultra-scalability for breakthrough science**
   – System can scale to 256 racks and beyond (>262,144 nodes)
   – Cluster: typically a few racks (512-1024 nodes) or less.

2. **Highest capability machine in the world (20-100PF+ peak)**

3. **Superior reliability: Run an application across the whole machine, low maintenance**

4. **Highest power efficiency, smallest footprint, lowest TCO (Total Cost of Ownership)**

5. **Low latency, high bandwidth inter-processor communication system**

6. **Low latency, high bandwidth memory system**

7. **Open source and standards-based programming environment**
   – Red Hat Linux distribution on service, front end, and I/O nodes
   – Lightweight Compute Node Kernel (CNK) on compute nodes ensures scaling with no OS jitter, enables reproducible runtime results
   – Automatic SIMD (Single-Instruction Multiple-Data) FPU exploitation enabled by IBM XL (Fortran, C, C++) compilers
   – PAMI (Parallel Active Messaging Interface) runtime layer. Runs across IBM HPC platforms

8. **Software architecture extends application reach**
   – Generalized communication runtime layer allows flexibility of programming model
   – Familiar Linux execution environment with support for most POSIX system calls.
   – Familiar programming models: MPI, OpenMP, POSIX I/O

9. **Broad range of scientific applicability at superior cost/performance**

10. **Key foundation for exascale exploration**
## Examples of Applications Running on Blue Gene
Developed on L, P; many ported to Q

<table>
<thead>
<tr>
<th>Application</th>
<th>Owner</th>
<th>Application</th>
<th>Owner</th>
<th>Application</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFD Alya System</td>
<td>Barcelona SC</td>
<td>DFT iGryd</td>
<td>Jülich</td>
<td>BM: SPEC2006, SPEC openmp</td>
<td>SPEC</td>
</tr>
<tr>
<td>CFD (Flame) AVBP</td>
<td>CERFACS Consortium</td>
<td>DFT KKRnano</td>
<td>Jülich</td>
<td>BM: NAS Parallel Benchmarks</td>
<td>NASA</td>
</tr>
<tr>
<td>CFD dns3D</td>
<td>Argonne National Lab</td>
<td>DFT ls3df</td>
<td>Argonne National Lab</td>
<td>BM: RZG (AIMS, Gadget, GENE, GROMACS, NEMORB, Octopus, Vertex)</td>
<td>RZG</td>
</tr>
<tr>
<td>CFD OpenFOAM</td>
<td>SGI</td>
<td>DFT PARATEC</td>
<td>NERSC / LBL</td>
<td>Coulomb Solver - PEPC</td>
<td>Jülich</td>
</tr>
<tr>
<td>CFD NEK5000, NEKTAR</td>
<td>Argonne, Brown U</td>
<td>DFT CPMD</td>
<td>IBM/Max Planck</td>
<td>MPI PALLAS</td>
<td>UCB</td>
</tr>
<tr>
<td>CFD OVERFLOW</td>
<td>NASA, Boeing</td>
<td>DFT QBOX</td>
<td>LLNL</td>
<td>Mesh AMR</td>
<td>CCSE, LBL</td>
</tr>
<tr>
<td>CFD Saturne</td>
<td>EDF</td>
<td>DFT VASP</td>
<td>U Vienna &amp; Duisburg</td>
<td>PETSC</td>
<td>Argonne National Lab</td>
</tr>
<tr>
<td>CFD LBM</td>
<td>Erlanger-Nuremberg</td>
<td>Q Chem GAMESS</td>
<td>Ames Lab/Iowa State</td>
<td>MpiBlast-pio Biology</td>
<td>VaTech / ANL</td>
</tr>
<tr>
<td>MD Amber</td>
<td>UCSF</td>
<td>Nuclear Physics GFMC</td>
<td>Argonne National Lab</td>
<td>RTM – Seismic Imaging</td>
<td>ENI</td>
</tr>
<tr>
<td>MD Dalton</td>
<td>Univ Oslo/Argonne</td>
<td>Neutronics SWEET3D</td>
<td>LANL</td>
<td>Supernova 1a FLASH</td>
<td>Argonne National Lab</td>
</tr>
<tr>
<td>MD ddcMD</td>
<td>LLNL</td>
<td>QCD CPS</td>
<td>Columbia U/IBM</td>
<td>Ocean HYCOM</td>
<td>NOPP / Consortium</td>
</tr>
<tr>
<td>MD LAMMPS</td>
<td>Sandia National Labs</td>
<td>QCD MILC</td>
<td>Indiana University</td>
<td>Ocean POP</td>
<td>LANL/ANL/Consortium</td>
</tr>
<tr>
<td>MD MP2C</td>
<td>Jülich</td>
<td>Plasma GTC</td>
<td>PPPL</td>
<td>Weather/Climate CAM</td>
<td>NCAR</td>
</tr>
<tr>
<td>MD NAMD</td>
<td>UIUC/NCSA</td>
<td>Plasma GYRO (Tokamak)</td>
<td>General Atomics</td>
<td>Weather/Climate Held-Suarez Test</td>
<td>GFDL</td>
</tr>
<tr>
<td>MD Rosetta</td>
<td>U Washington</td>
<td>KAUST Stencil Code Gen</td>
<td>KAUST</td>
<td>Climate HOMME</td>
<td>NCAR</td>
</tr>
<tr>
<td>DFT GPAW</td>
<td>Argonne National Lab</td>
<td>BM: sppm, raptor, AMG, IR5, sphot</td>
<td>Livermore</td>
<td>Weather/Climate WRF, CM1</td>
<td>NCAR, NCSA</td>
</tr>
</tbody>
</table>

### Accelerating Discovery and Innovation in:

- **Materials Science**
- **Energy**
- **Engineering**
- **Climate & Environment**
- **Life Sciences**

- Silicon Design
- Next Gen Nuclear
- High Efficiency Engines
- Oil Exploration
- Whole Organ Simulation
Blue Gene/Q Expanded Apps Reach

- **Ease of Programming**
  - More memory/node
  - Enhanced I/O
  - Ease of porting

- **BROADER Application Front**
  - Graph 500
  - Life Sciences
  - Uncertainty Quantification

- **Increasing capability- Example**
  - L: a few fuel rods (5x5)
  - P: fuel assembly (17x17)
  - Q: nuclear reactor (~200 assemblies)
October 7, 2009: President Obama presented the 2008 National Medal of Technology and Innovation to IBM, the only company so honored, for the Blue Gene family of supercomputers...

The US Government and IBM represent world leadership in high performance computing.
System Power Efficiency (Green500 06/2011)

At $0.10/kWh => 1MW savings in power saves $1M/year. TCO saving is much more.
Low power is key to scaling to large systems

Source: www.green500.org

© 2011 IBM Corporation
Annualized TCO of HPC Systems (Cabot Partners)

BG/Q saves ~$300M/yr!
Annualized TCO & Component Costs vs Peak Performance
Blue Gene Evolution

- **BG/L (5.7 TF/rack, 210 MF/W) – 130nm ASIC (2004 GA)**
  - Scales >128 racks, 0.734 PF/s, dual-core system-on-chip,
  - 0.5/1 GB / Node

- **BG/P (13.9 TF/rack, 357 MF/W) – 90nm ASIC (2007 GA)**
  - Scales >256 racks, 3.5 PF/s, quad core SOC, DMA
  - 2/4 GB / Node
  - SMP support, OpenMP, MPI

- **BG/Q (209 TF/rack, 2000 MF/W) – 45nm ASIC (Early 2012 GA)**
  - Scales >256 racks, 53.6 PF/s, 16 core/64 thread SOC
  - 16 GB / Node
  - Speculative execution, sophisticated L1 prefetch, transactional memory, fast thread handoff, compute + IO systems
Blue Gene/Q

1. Chip:
   16+2 μP cores

2. Single Chip Module:
   Heat Spreader for H₂O Cooling

3. Compute card:
   One chip module,
   16 GB DDR3 Memory,
   Heat Spreader for H₂O Cooling

4. Node Card:
   32 Compute Cards,
   Optical Modules, Link Chips; 5D Torus

5a. Midplane:
   16 Node Cards

5b. IO drawer:
   8 IO cards w/16 GB
   8 PCIe Gen2 x8 slots
   3D I/O torus

6. Rack: 2 Midplanes

7. System:
   96 racks, 20PF/s

- Sustained single node perf: 10x P, 20x L
- MF/Watt: (6x) P, (10x) L (~2GF/W, Green 500 criteria)
- Software and hardware support for programming models
  for exploitation of node hardware concurrency
BlueGene/Q Compute chip

System-on-a-Chip design: integrates processors, memory and networking logic into a single chip

- 360 mm² Cu-45 technology (SOI)
- 16 user + 1 service PPC processors
  - plus 1 redundant processor
  - all processors are symmetric
  - 11 metal layer
  - each 4-way multi-threaded
  - 64 bits
  - 1.6 GHz
  - L1 I/D cache = 16kB/16kB
  - L1 prefetch engines
  - each processor has Quad FPU
    (4-wide double precision, SIMD)
  - peak performance 204.8 GFLOPS @ 55 W
- Central shared L2 cache: 32 MB
  - eDRAM
  - multiversioned cache – supports transactional memory, speculative execution.
  - supports scalable atomic operations
- Dual memory controller
  - 16 GB external DDR3 memory
  - 42.6 GB/s DDR3 bandwidth (1.333 GHz DDR3)
    (2 channels each with chip kill protection)
- Chip-to-chip networking
  - 5D Torus topology + external link
    → 5 x 2 + 1 high speed serial links
  - each 2 GB/s send + 2 GB/s receive
  - DMA, remote put/get, collective operations
- External (file) IO -- when used as IO chip.
  - PCIe Gen2 x8 interface (4 GB/s Tx + 4 GB/s Rx)
  - re-uses 2 serial links
  - interface to Ethernet or Infiniband cards
Inspired by array redundancy

- PUUnit N+1 redundancy scheme substantially increases yield of large chip
- Redundancy can be invoked at any manufacturing test stage
  - wafer, module, card, system
- Redundancy info travels with physical part -- stored on chip (eFuse) / on card (EEPROM)
  - at power-on, info transmitted to PUUnits, memory system, etc.

- Single part number flow
- Transparent to user software: user sees N consecutive good processor cores.
Main Memory Capacity per Rack
Main Memory Bandwidth per Rack

![Bar chart showing memory bandwidth per rack for various systems.](chart.png)
Inter-Processor Communication

- **Integrated 5D torus**
  - Virtual Cut-Through routing
  - Hardware assists for collective & barrier functions
  - FP addition support in network
  - RDMA
    - Integrated on-chip Message Unit

- **2 GB/s raw bandwidth on all 10 links**
  - each direction -- i.e. 4 GB/s bidi
  - 1.8 GB/s user bandwidth
    - protocol overhead

- **5D nearest neighbor exchange measured at 1.76 GB/s per link (98% efficiency)**

- **Hardware latency**
  - Nearest: 80ns
  - Farthest: 3us
    - (96-rack 20PF system, 31 hops)

- **Additional 11th link for communication to IO nodes**
  - BQC chips in separate enclosure
  - IO nodes run Linux, mount file system
  - IO nodes drive PCIe Gen2 x8 (4+4 GB/s)
    - IB/10G Ethernet <-> file system & world

**Network Performance**
- All-to-all: 97% of peak
- Bisection: > 93% of peak
- Nearest-neighbor: 98% of peak
- Collective: FP reductions at 94.6% of peak
Inter-Processor Peak Bandwidth per Node

- Roadrunner
- SGI Xeon Pleiades
- Sun TACC
- Itanium 2
- Power 6
- Fujitsu K
- Cray XT6 12C
- Cray XT5 4C
- BG/Q
- Tianhe-1A

Byte/Flop

© 2011 IBM Corporation
Blue Gene/Q Compute Card Assembly

- Basic field replaceable unit of a Blue Gene/Q system
- Compute Card has 1 BQC chip + 72 SDRAMs (16GB DDR3)
- Two heat sink options: Water-cooled → “Compute Node” / air-cooled → “IO Node”
- Connectors carry power supplies, JTAG etc, and 176 Torus signals (4 and 5 Gbps)
- Power efficient processor chips allow dense packaging
- High bandwidth / low latency electrical interconnect on-board
- 18+18 (Tx+Rx) 12-channel optical fibers @10Gb/s
  - Recombined into 8*48-channel fibers for rack-to-rack (Torus) and 4*12 for Compute-to-IO interconnect
- Compute Node Card assembly is water-cooled (18-25°C – above dew point)
- Redundant power supplies with distributed back-end ~ 2.5 kW
Full height, 25W PCI cards, NOT hot serviceable.

~1 KW per I/O Drawer

8 compute cards (different PN than in compute rack because of heatsink vs cold plate)

Axial fans – same as BGP

Fiber connections

Clock input

Ball bearing slides for field maintenance

48V power input (not shown)

Picture by Shawn Hall
## Failures per Month per TF


<table>
<thead>
<tr>
<th></th>
<th>Scale Demonstrated Factor to PF</th>
<th>Failures per month per TF</th>
<th>Power Consumption @PF</th>
<th>Estimated System Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray XT3/XT4</td>
<td>10880 CPUs 10X to PF 100,000 CPUs</td>
<td>~1 - ~1</td>
<td>~8MW XT4</td>
<td>&gt;$150M XT4</td>
</tr>
<tr>
<td>Clusters X86/AMD64</td>
<td>8000 CPUs 12X to PF 100,000 CPUs</td>
<td>2.6 - 8.0</td>
<td>~6MW</td>
<td>&gt;$150M x86</td>
</tr>
<tr>
<td>Blue Gene L/P</td>
<td>131,720 CPUs 2.2x to PF 294,912</td>
<td>.01-0.03</td>
<td>~2.3MW BG/P</td>
<td>&lt;$100M</td>
</tr>
</tbody>
</table>

Example: A 100 hr job => BG/Q architecture has 2x advantage in TCO  
-MTBF 70 hrs  150 hrs to complete (96 rack BG/Q MTBF target)  
-MTBF 7 hrs  309 hrs to complete
Blue Gene/Q Software High-Level Goals & Philosophy

- Facilitate extreme scalability
  - Extremely low noise on compute nodes
- High reliability: a corollary of scalability
- Standards-based when possible, leverage other IBM HPC
- Open source where possible
- Facilitate high performance for unique hardware:
  - Quad FPU, DMA unit, List-based prefetcher
  - TM (Transactional Memory), SE (Speculative Execution)
  - Wakeup-Unit, Scalable Atomic Operations
- Optimize MPI and native messaging performance
- Optimize libraries
- Facilitate new programming models
### Software comparison: BG/Q is more general purpose

<table>
<thead>
<tr>
<th>Property</th>
<th>BG/L</th>
<th>BG/Q</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overall Philosophy</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td>Scale infinitely, minimal functionality</td>
<td>Scale infinitely, added more functionality</td>
</tr>
<tr>
<td>Openness</td>
<td>closed</td>
<td>almost all open</td>
</tr>
<tr>
<td><strong>Programming Model</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shared Memory</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Hybrid</td>
<td>2 processes 1 thread (software managed)</td>
<td>1-64 processes 64-1 threads</td>
</tr>
<tr>
<td>Low-Level General Messaging</td>
<td>No</td>
<td>PAMI, generic parallel program runtimes, wake-up unit</td>
</tr>
<tr>
<td>Programming Models</td>
<td>MPI, ARMCi, global arrays</td>
<td>MPI, OpenMP, UPC, ARMCi, global arrays, Charm++</td>
</tr>
<tr>
<td><strong>Kernel</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System call interface</td>
<td>proprietary</td>
<td>Linux/POSIX system calls</td>
</tr>
<tr>
<td>Library/threading</td>
<td>glibc/proprietary</td>
<td>glibc/pthreads</td>
</tr>
<tr>
<td>Linking</td>
<td>static only</td>
<td>static or dynamic</td>
</tr>
<tr>
<td>Compute Node OS</td>
<td>CNK</td>
<td>CNK, Linux, Red Hat</td>
</tr>
<tr>
<td>I/O Node OS</td>
<td>Linux</td>
<td>SMP Linux with SMT, Red Hat</td>
</tr>
<tr>
<td><strong>Control</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scheduling</td>
<td>generic API</td>
<td>generic and real-time API</td>
</tr>
<tr>
<td>Run Mode</td>
<td>HPC, prototype HTC</td>
<td>Integrated HPC, HTC, MPMD, and sub-blocks, HA with job cont</td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td>Tools</td>
<td>HPC Toolkit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPC Toolkit, Dyninst, Valgrind, PAPI</td>
</tr>
<tr>
<td><strong>Research Initiatives</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>Scaling Linux</td>
<td>ZeptoOS, Plan 9</td>
</tr>
<tr>
<td>Big Data</td>
<td>N/A</td>
<td>BGAS (Blue Gene Active Storage), Large memory nodes</td>
</tr>
<tr>
<td>Commercial</td>
<td>N/A</td>
<td>Kittyhawk, Cloud, SLAcc</td>
</tr>
</tbody>
</table>
Blue Gene System Architecture

Service Node
- System Console
- LoadLeveler
- MMCS
- DB2

Front-end Nodes
- File Servers
- Functional Network 10Gb QDR

I/O Node
- Linux
- fs client
- ciod

Control Ethernet (1Gb)

FPGA

Control
Ethernet

Collective network
- Control
Ethernet
- Optical
- Torus

I/O Node
- Linux
- fs client
- ciod

C-Node 0
- app
- CNK

C-Node n
- app
- CNK

File Servers

 JTAG
I/O on Blue Gene/Q

Application
fscanf

libc
read

CNK

BG ASIC
cn packets

Optical Network

CIOS
Full Red Hat Linux
On I/O node

Linux
FS
IP
read
cn packets

BG ASIC

Ethernet or Infiniband

File server

© 2011 IBM Corporation
Blue Gene Q Software Innovations

- Standards-based programming environment
  - Linux™ development environment
  - Familiar GNU toolchain with glibc, pthreads, gdb
  - Red Hat on I/O node
  - XL Compilers C, C++, Fortran with OpenMP 3.1
  - Debuggers: Totalview
  - Tools: HPC Toolkit, PAPI, Dyinst, Valgrind, Open Speedshop

- Message Passing
  - Scalable MPICH2 providing MPI 2.2 with extreme message rate
  - Efficient intermediate (PAMI) and low-level (SPI) message libraries, documented, and open source
  - PAMI layer allows easy porting of runtimes like GA/ARMCI, Berkeley UPC, etc,

- Compute Node Kernel (CNK) eliminates OS noise
  - File I/O offloaded to I/O nodes running full Linux
  - GLIBC environment with a few restrictions for scaling

- Flexible and fast job control – with high availability
  - Integrated HPC, HTC, MPMD, and sub-block jobs
  - Noise-free partitioned networks as in previous BG

- New for Q
  - Scalability Enhancements: the 17th Core
    - RAS Event handling and interrupt off-load
    - Event CIO Client Interface
    - Event Application Agents: privileged application processing
  - Wide variety of threading choices
  - Efficient support for mixed-mode programs
  - Support for shared memory programming paradigms
  - Scalable atomic instructions
  - Transactional Memory (TM)
  - Speculative Execution (SE)
  - Sub-blocks
    - Integrated HTC, HPC, MPMD, Sub-blocks
  - Integrated persistent memory
  - High availability for service nodes with job continuation
  - I/O nodes running Red Hat

© 2011 IBM Corporation
IB MS Y S T E M  T E C H N O L O G Y  G R O U P

B G / Q  S O F T W A R E  S T A C K  O P E N N E S S

I/O and Compute Nodes

Application
- GNU Runtime
- XL Runtime
- ESSL
- MPI
- Global Arrays
- Charm++
- MPI-IO

System
- PAMI (Converted Messaging Stack)
- Compute Node Kernel (CNK)
- CIO Services
- totalviewed
- Messaging SPIs
- Node SPIs
- Linux kernel

Firmware
- Node Firmware
  - Init, Bootloader, RAS, Recovery Mailbox
- Diagnostics

HW
- Compute nodes
- I/O nodes

Service Nodes/Login Nodes

Application
- BGMon
- BG Nav
- GNU Compilers
- XL Compilers
- runjob
- Sched API
- ISV Schedulers, debuggers
- Loadleveler
- HPC Toolkit

System
- High Level Control System (MMCS)
  - Partitioning, Job management and monitoring, RAS, Administrator interface

User/Sched
- TEL
- Diag Harness
- BGWS
- BG master

Firmware
- Low Level Control System
  - Power On/Off, Hardware probe, Hardware init, Parallel monitoring, Parallel boot, Mailbox

HW
- Node cards
- Service cards
- SN
- SSNs
- LNs

New open source reference implementation licensed under CPL.
Existing open source community under CPL license. Active IBM participation.
Existing open source communities under various licenses. BG code will be contributed and/or new sub-community started.
Closed. No source provided. Not buildable.

© 2011 IBM Corporation
Execution Modes in BG/Q per Node

Next Generation HPC
- Many Core
- Expensive Memory
- Two-Tiered Programming Model

Hardware Abstractions Black
Software Abstractions Blue

64 Processes
1 Thread/Process

2,4,8,16,32 Processes
32,16,8,4,2 Threads

1 Process
64 Threads

© 2011 IBM Corporation
- Message Layer Core has C++ message classes and other utilities to program the different network devices
- Support many programming paradigms
- PAMI runtime layer allows uniformity across IBM HPC platforms

* describes capability not necessarily product support
Advantages of Software/Hardware Co-Design on BG/Q (helping take advantage of multi-core environment)

- **Scalable atomic instructions**
  - Enables development of lock-less producer consumer queues with N producers and 1 or more consumers

- **Hardware wake-up mechanism**
  - Support for OpenMP/MPI and other hybrid programming models

- **List-based prefetching**
  - Allows efficient use of cache for broader applications

- **Multi-valued L2 cache**
  - TM and TLS
Standard Atomic Operation
(Iwarx stx on PowerPC)

- N round trips
  - Where N is the number of threads
  - For N=64 and L2 74 cycles $\rightarrow$ ~9500 cycles
Scalable Atomic Operation (fetch_and_inc for example)

- 1 round trips + N L2 cycles
  - Where N is the number of threads
  - For N=64 and L2 74 cycles → ~800 cycles
  - Compared to ~9500 cycles for standard
Use of Scalable Atomic ops

Barrier speed using different synchronizing hardware

- atomic: no -invalidates
- atomic: invalidates
- lwarx/stwcx

number of processor cycles

number of threads
Wakeup Unit

- Allow hardware threads to stop executing instructions
  - Only two threads needed to keep A2 utilized
- Avoids software polling
- Waiting thread configured to wake up on choice of interrupt

<table>
<thead>
<tr>
<th>node</th>
<th>core_0</th>
<th>core_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0</td>
<td>t1</td>
<td>t2</td>
</tr>
</tbody>
</table>

core_i

- IPI (Inter Processor Interrupt)
- MU (Messaging Unit Interrupt)
- L2
List-Based prefetching for LLNL IRS Sequoia kernel

- Benchmark with distributed pattern:
  
  ```
  for ( kk = kmin ; kk < kmax ; kk++ ) {
    for ( jj = jmin ; jj < jmax ; jj++ ) {
      for ( ii = imin ; ii < imax ; ii++ ) {
        i = ii + jj * jp + kk * kp ;
      }
    }
  }
  ```

  ![Start list]

  ![Stop list]
“Perfect” Prefetching

- Tolerance to missing or extra cache misses
  - Possible asynchronous A2 behavior may cause out-of-sync addresses to be issued or initially recorded.
  - An L1 miss not matching the next list address will be compared to the next N addresses in the list. A match will cause list prefetching to continue from the point of match.
  - An L1 miss not matching these N+1 addresses will be discarded and the next miss addressed compared. M sequential such failures will cause the list to be abandoned.
  - When a list is abandoned:
    - Stream prefetching is activated.
    - List recording continues.
    - In all cases list recording of each L1 miss address continues until stop list is asserted. The new list then overwrites the original one.
    - Such self-healing and adaptation is likely needed since the address pattern will change as the list repeats and prefetching becomes more accurate.
Concurrency Design Space
Putting TM in Perspective

System Performance

Programmer Effort

Canned NBDS

Coarse Locks

Fine Locks

Ad Hoc NBDS

HW TM

Software TM (STM)

Credit: Bill Scherer
BlueGene/Q transactional memory mode

- **User program model:**
  - User defines parallel work to be done
  - User explicitly defines start and end of transactions within parallel work that are to be treated as **atomic**

- **Compiler implications:**
  - Interpret user program annotations to spawn multiple threads
  - Interpret user program annotation for start of transaction and save state to memory on entry to transaction to enable rollback
  - At end of transaction program annotation test for successful completion and optionally branch back to rollback pointer.

- **Hardware implications:**
  - Transactional memory support required to detect transaction failure and rollback
  - L1 cache visibility for L1 hits as well as misses allowing for ultra low overhead to enter a transaction
BlueGene/Q 0’th compiler support for TLS

Program Execution Flow with TLS

Runtime optimized decision branch points

Potentially independent "serial" fragments
Summary Blue Gene/Q

1. Ultra-scalability for breakthrough science
   - System can scale to 256 racks and beyond (>262,144 nodes)
   - Cluster: typically a few racks (512-1024 nodes) or less.

2. Lowest Total Cost of Ownership
   - Highest total power efficiency, smallest footprint
   - Typically 2 orders of magnitude better reliability

3. Broad range of applications reach
   - Familiar programming models
   - Easy porting from other environments

4. Foundation for Exascale exploration