



The New Roadrunner Supercomputer: What, When, & How

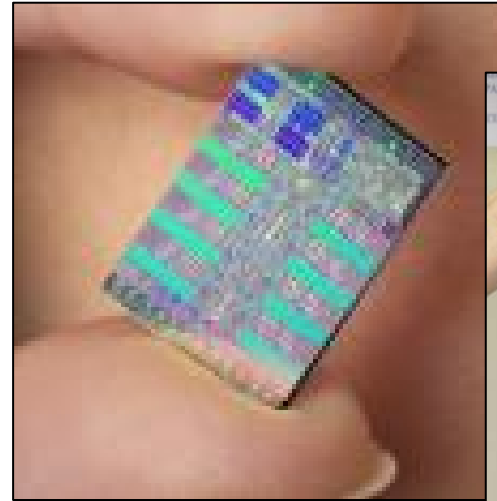
SC06
November 14, 2006

Ken Koch
Los Alamos National Laboratory

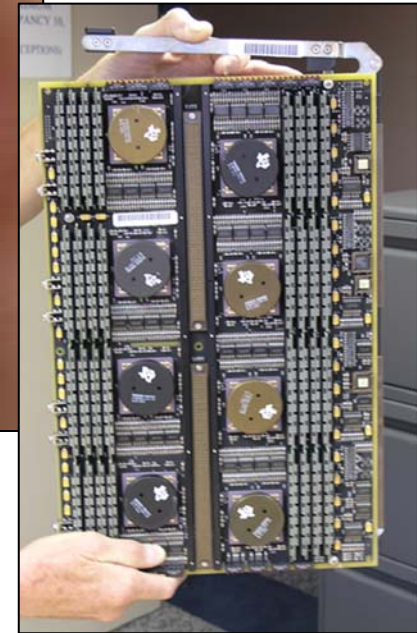
Roadrunner is an Important Supercomputer Asset for Los Alamos



- Contract awarded to **IBM** in early September 2006 with delivery starting soon thereafter
- Critical component of stockpile stewardship
 - Initial system supports near-term mission deliverables
 - Hybrid final system achieves PetaFlops level of performance
- Vision of the future
 - Faster computation by using hybrid/heterogeneous processors as accelerators



Cell processor (2007, ~100 GF)



CM-5 board with vector units (1994, 1 GF)



Roadrunner's Goals

- Provide a large “capacity-mode” computing resource for LANL weapons simulations
 - Cycles for throughput of 64-way to 1000-way parallel jobs
 - Purchase in FY2006 and stand up quickly
 - Robust HPC architecture with known usability for LANL codes
- Possible upgrade to petascale-class hybrid “accelerated” architecture in a year or two
 - Follow future trends toward hybrid/heterogeneous computers
 - More and varied processors with special function units
 - Capable of supporting future LANL weapons physics and system design workloads
 - Capable of achieving a sustained PetaFlop



Roadrunner Comes in Stages

- Phase 1 **2006** **Stage 1 Deployment**
 - Multiple non-accelerated clustered systems Oct. 2006
 - Provides a large classified capacity at LANL
 - One cluster with 7 Cell-accelerated nodes for development & testing (Advanced Architecture Initial System — AAIS)
- Phase 2: Technology Refresh & Assessment **2007**
 - Improved Cell Blades & Cell software on 6 more nodes of AAIS
 - Supports pre-Phase 3 assessment
- Phase 3 **2008** **Stage 2 Deployment**
 - Populate entire system with Cell Blades
 - Achieve a **sustained** 1 PetaFlop Linpack
 - Run at least some accelerated LANL codes initially
 - Contract Option

Roadrunner starts as a fairly
standard collection of

Base System Clusters

which we call Connected Units



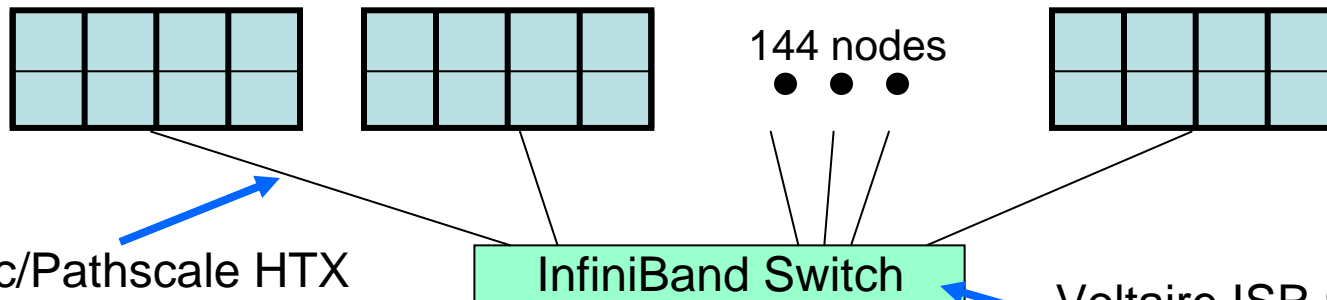
Roadrunner Connected Unit

8-way (quad-socket dual-core) Opteron Node



IBM x3755 server w/
4 dual-core Opterons
& 32 GB memory

Base System Connected Unit (CU) 144-way Cluster



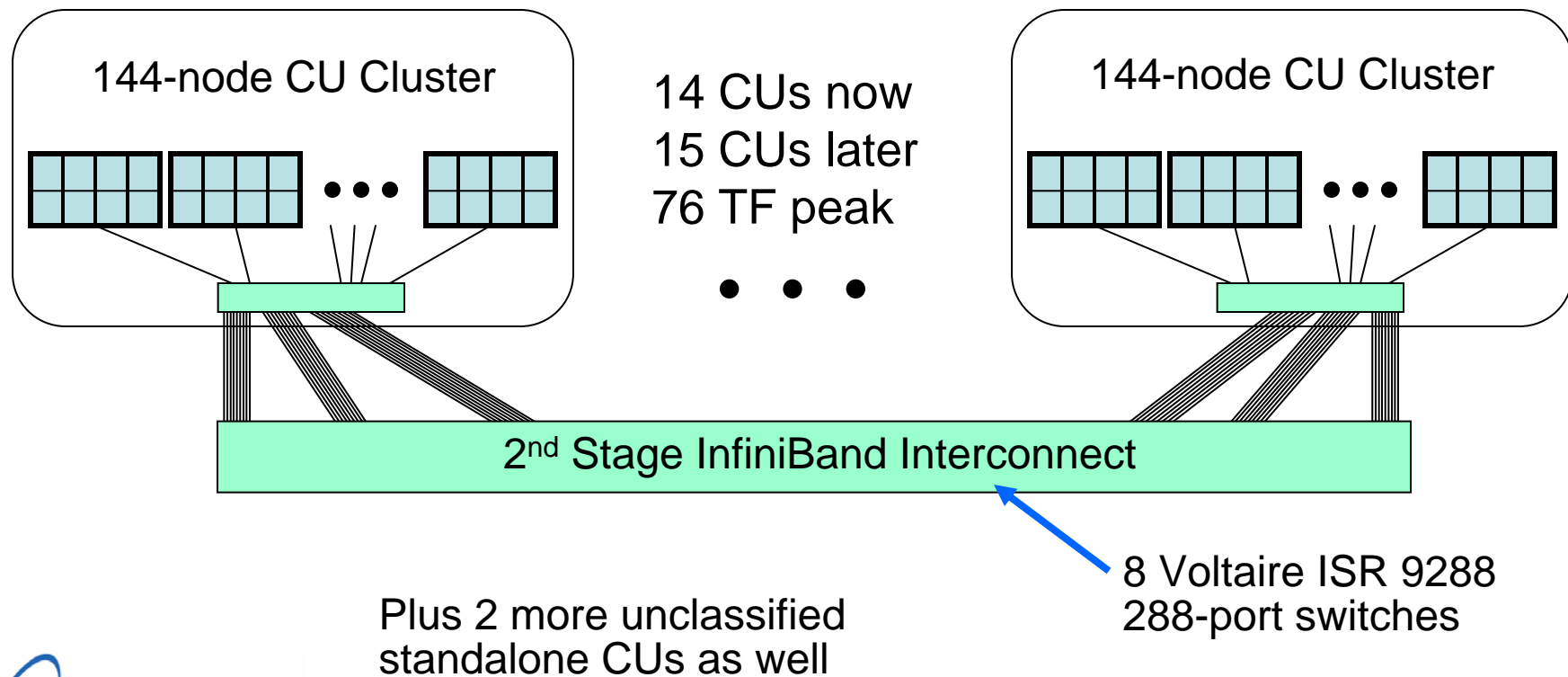
QLogic/Pathscale HTX
InfiniBand 4x SDR links

Voltaire ISR 9288
288-port switch



Roadrunner Base System

Multiple Cluster Classified Base System

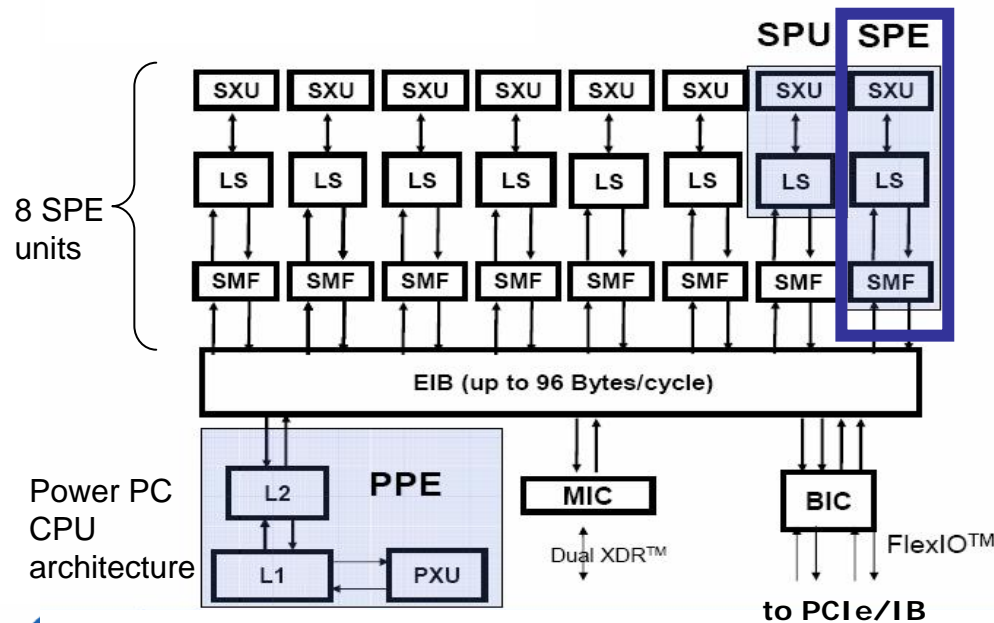
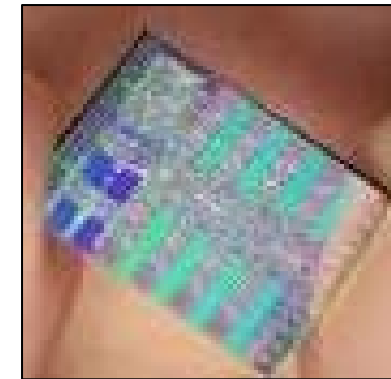


Final Roadrunner accelerated architecture will attach IBM Cell blades (e.g. QS20) as Accelerators using multiple InfiniBand links to each Opteron cluster node



Cell Chip

- Cell Broadband Engine™ * (Cell BE)
 - Developed under Sony-Toshiba-IBM efforts
 - Current Cell chip is used in the Sony PlayStation 3
- An 8-way heterogeneous parallel engine



Each of the 8 SPEs are 128 bit (e.g. 2-way DP-FP) vector engines w/ 256KB of Local Store (LS) memory & a DMA engine.

They can operate together or independently (SPMD or MPMD).

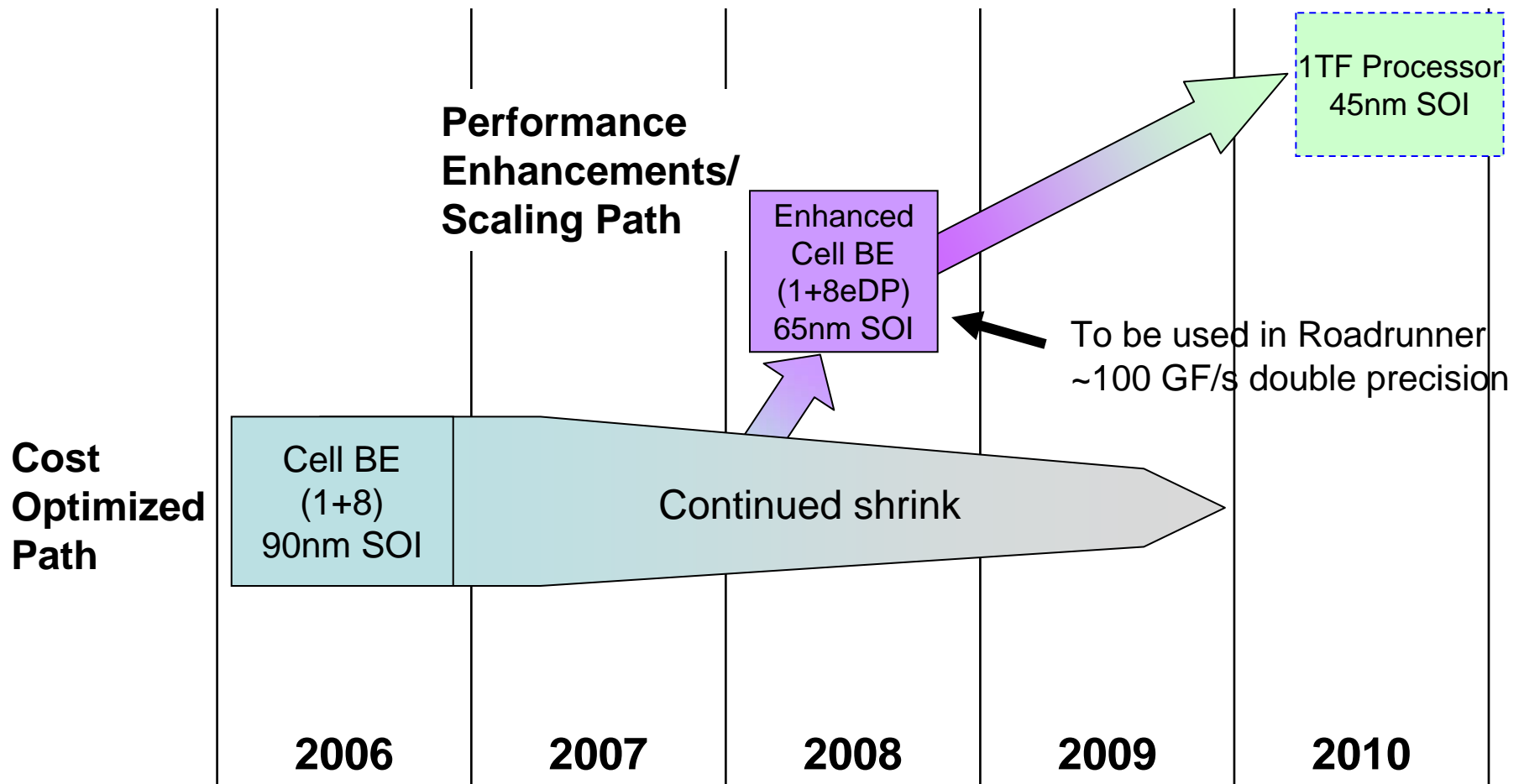
~200 GF/s single precision

~ 15 GF/s double precision (current chip)

* Trademark of Sony Computer Entertainment, Inc.



Cell Broadband Engine Architecture™ Technology Competitive Roadmap

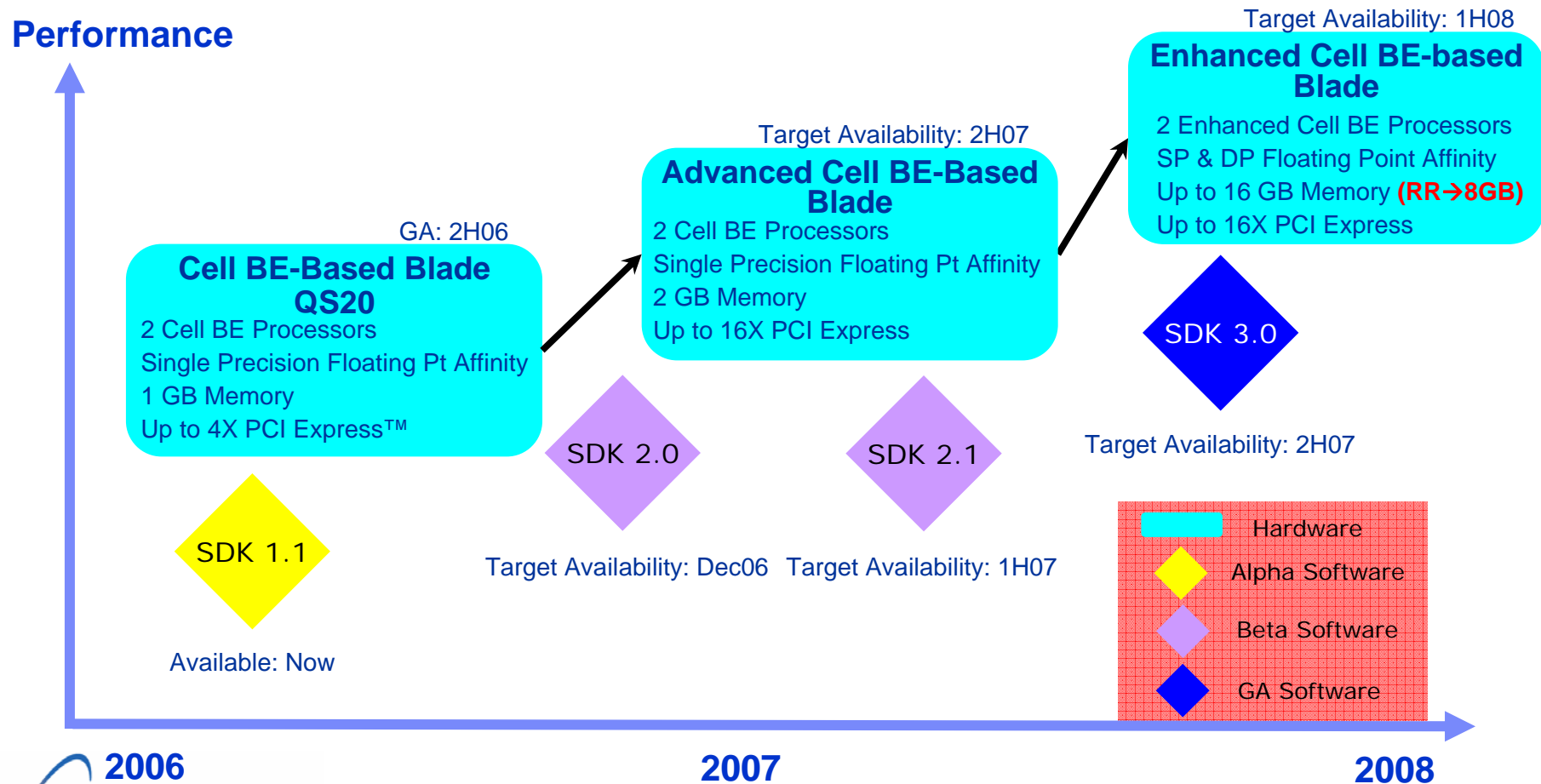


Cell BE Roadmap Version 5.0 24-Jul-2006

All future dates are estimations only; Subject to change without notice.



Cell Blade Roadmap



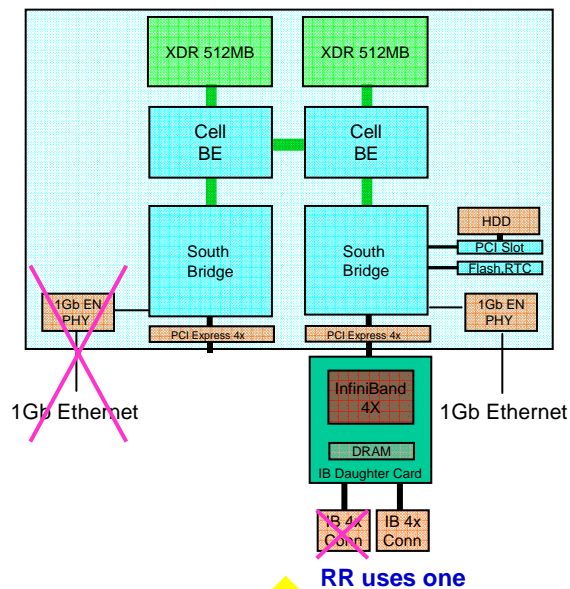
All future dates are estimations only; Subject to change without notice.



Roadrunner Cell HW & SW

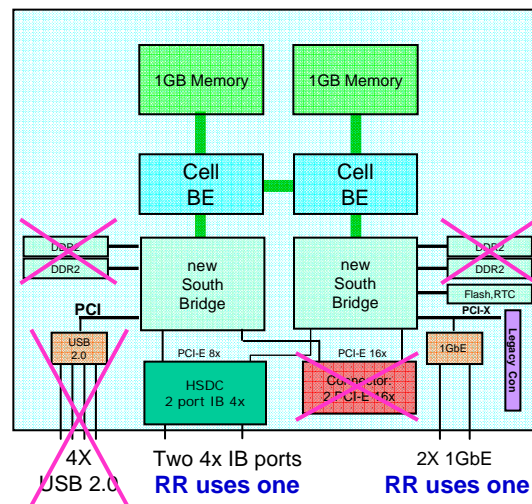
Phase 1 Oct. 2006

2 Cell processors, 3.2 GHz
 512MB XDR memory each
 1 connected IB 4X SDR link



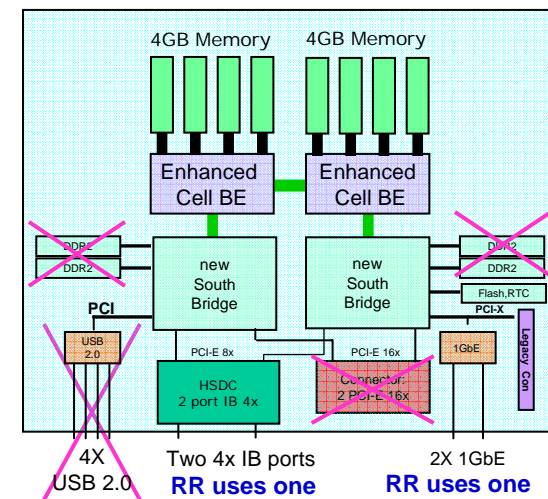
Phase 2 Refresh June 2007

2 Cell processors, 3.2 GHz
 1 GB XDR memory each
 1 connected IB 4X SDR link



Phase 3 Option Q1 2008

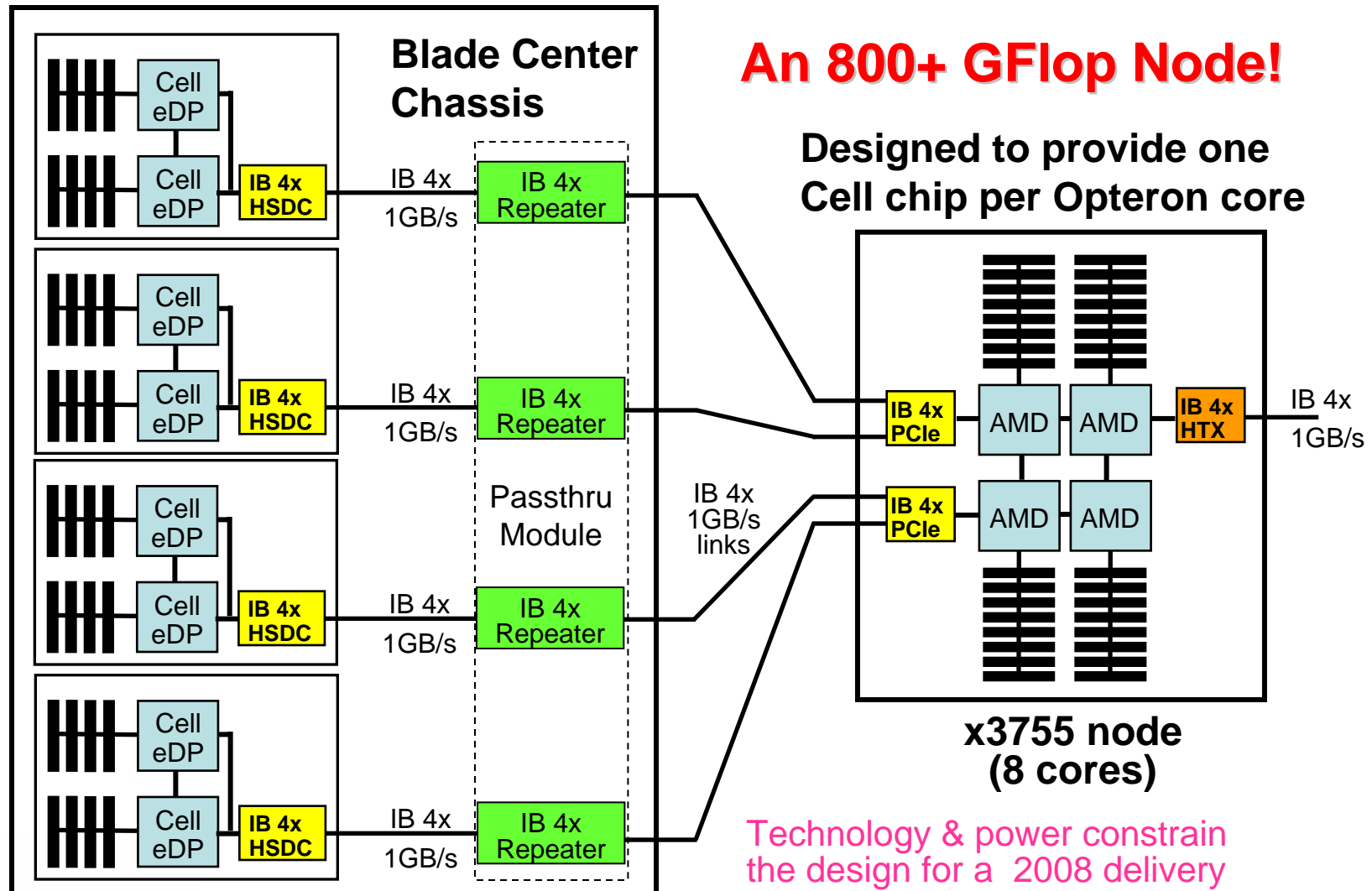
2 Cell eDP processors, 3.2 GHz
 4 GB DDR2 memory each
 1 connected IB 4x SDR link



Roadrunner does not use the X-pink components



Accelerated Node



An 800+ GFlop Node!

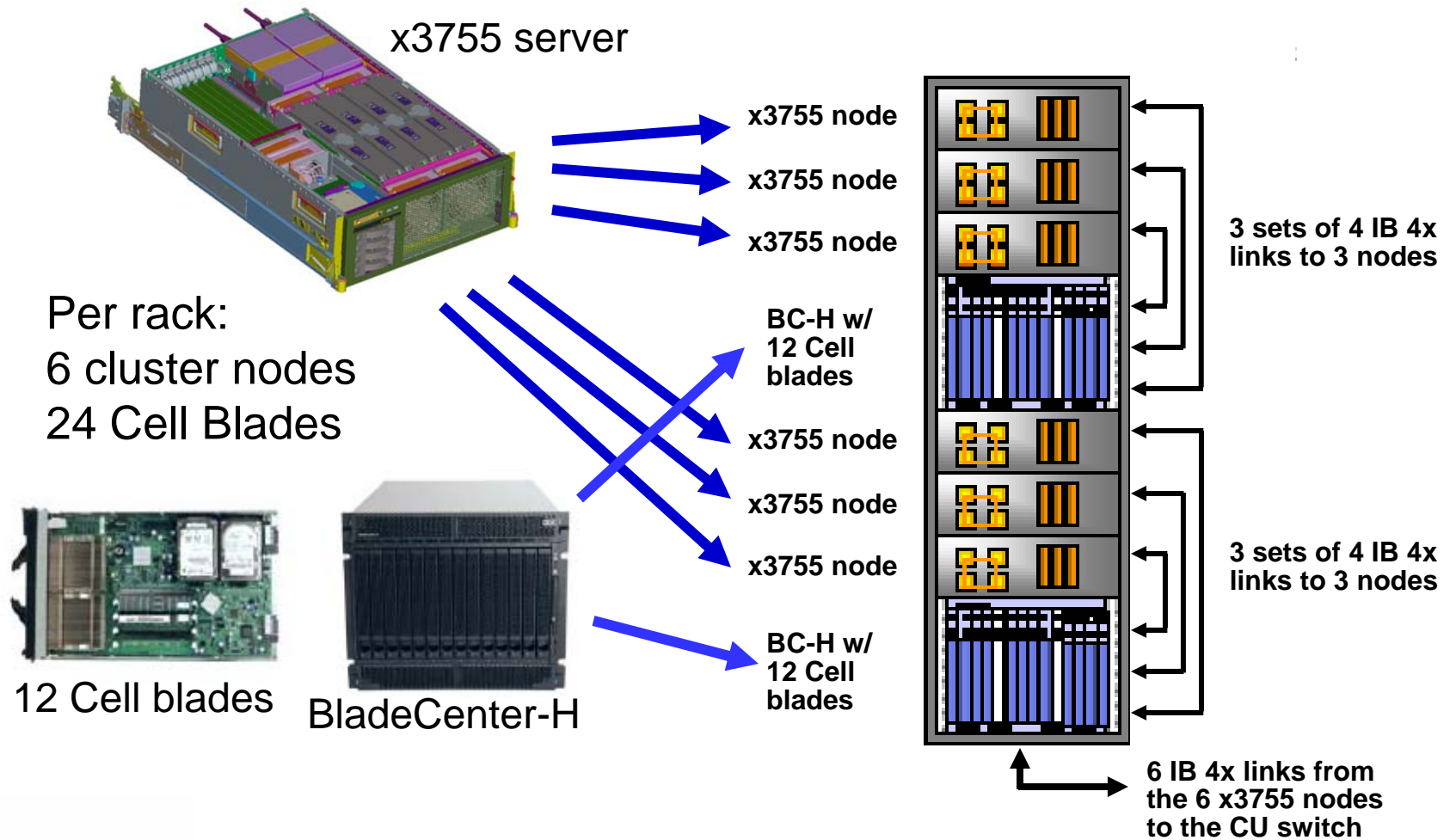
Designed to provide one Cell chip per Opteron core

**x3755 node
(8 cores)**

Technology & power constrain the design for a 2008 delivery timeframe

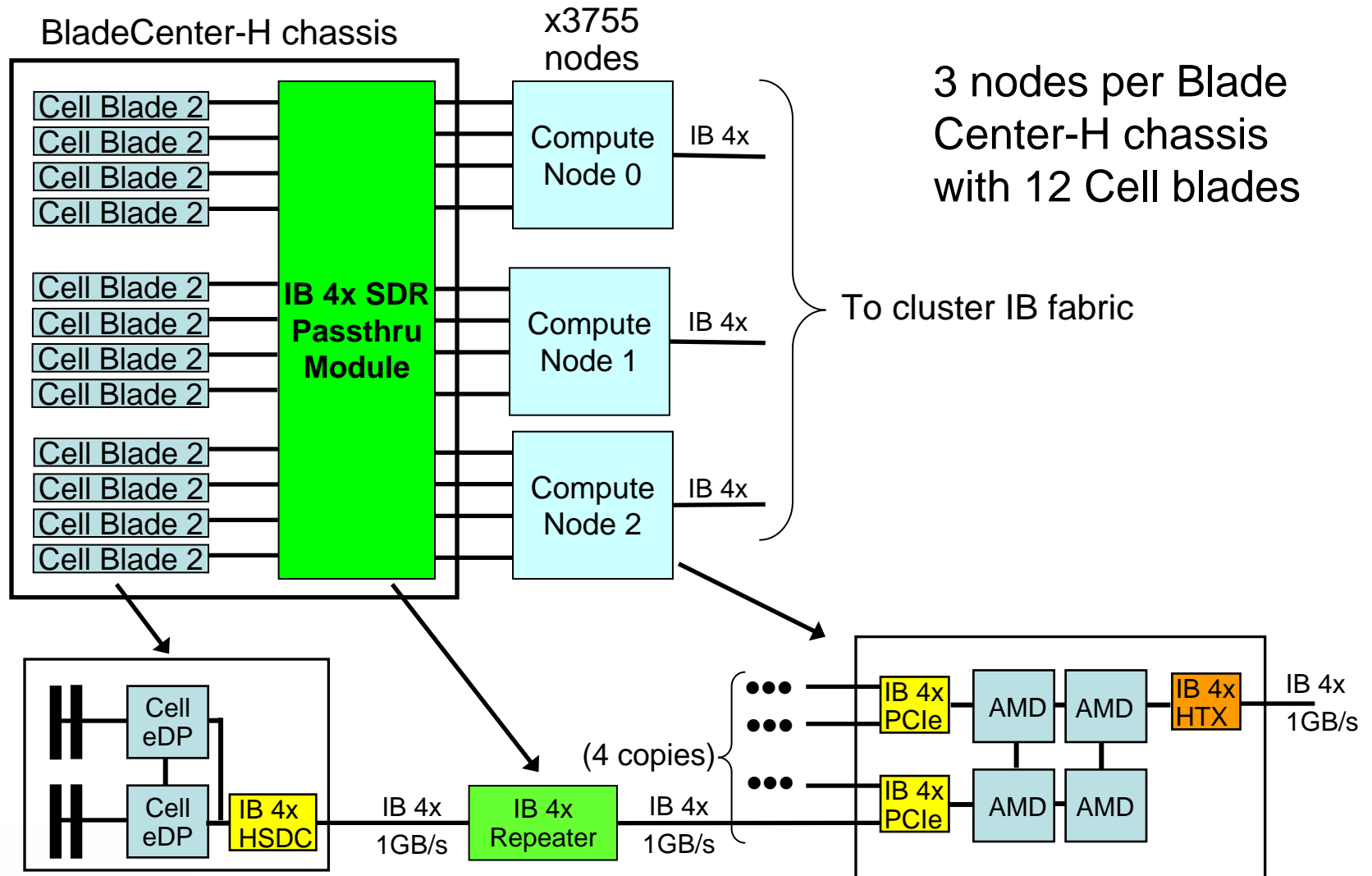


Compute Rack





Cell Blade Attachment

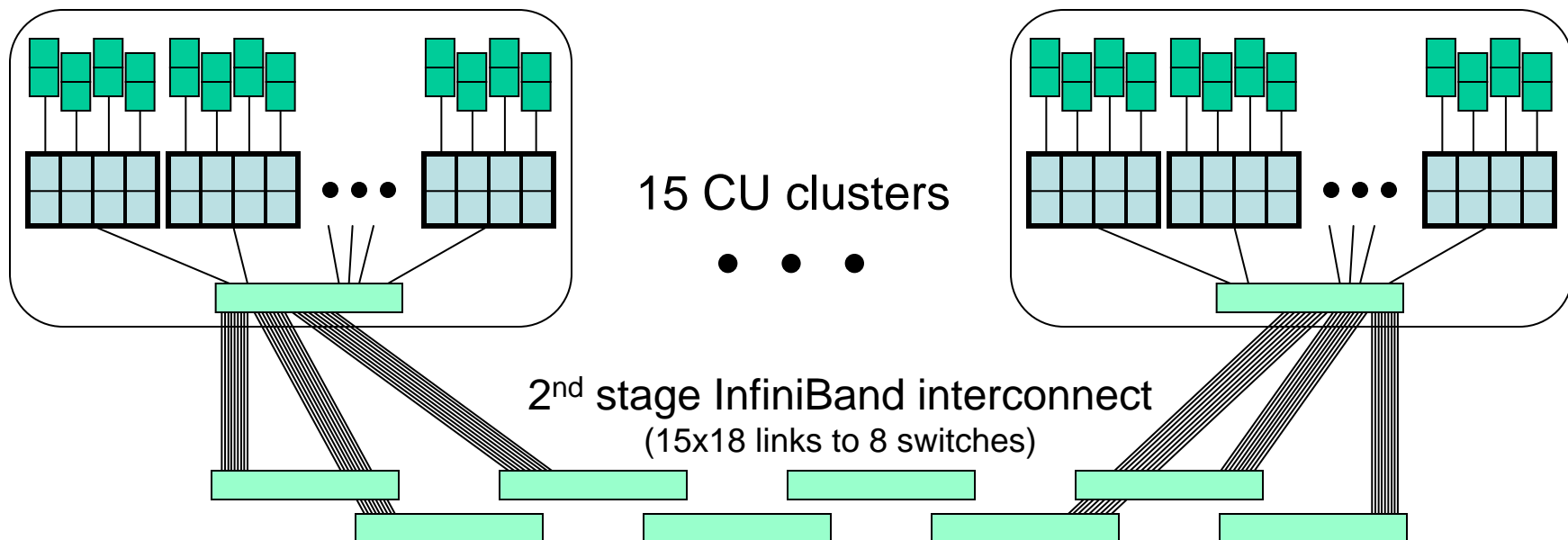




Accelerated Roadrunner

“Connected Unit” cluster
144 quad-socket
dual-core nodes
(138 w/ 4 dual-Cell blades)
InfiniBand interconnects

8,640 dual-core Optrons
• **76 TeraFlops**
16,560 eDP Cell chips
• **1.7 PetaFlops Cell**





Cells as Accelerators

- Eight Cells attached to each of 138 nodes of each Roadrunner CU
 - 16,560 total Cells in the Phase 3 Roadrunner system
 - Designed to provide one Cell chip per Opteron core
 - Four IBM Blade Center blades each with dual-Cell chips per node
 - 12 blades per BladeCenter-H 9U chassis (2 empty slots)
 - Each blade is directly connected with one IB 4X SDR link to its host node
 - Ratios:
 - 1 host node = 8 Opteron cores & 4 Cell Blades = 8 Cell cores
 - 4 IB links = 2 dual-IB cards per host & 1 IB card per blade
 - 32 GB of Opteron node memory and 32GB of Cell blade memory



Roadrunner & Cell Roadmap

- Current Cell chips & QS20 blades in AAIS cluster in Phase 1 (November 2006)
 - Use these to develop a few Cell and hybrid applications
- Next generation blade w/ same Cell chips in Phase 2 AAIS upgrade (June 2007)
 - Use these to test performance and make projections for a larger system with eDP Cell chips
- Next generation blade w/ enhanced double precision (eDP) Cell chips in Phase 3 (January 2008)
 - New eDP “rev” provides double-precision performance of ~100GFlops
 - 4 GB of memory per Cell chip

Programming Roadrunner can be
easy

or it can be challenging but with
great potential benefit



Roadrunner Programming

- Roadrunner is a cluster
 - Existing Opteron-only parallel codes run **unaltered** on cluster nodes
 - MPI or other cluster codes run unmodified
- Roadrunner is hybrid/heterogeneous
 - Computationally intense kernels or entire modules or pieces are partially modified or rewritten to take advantage of Cells
 - Accelerate large granularity computation kernels/modules
 - Source code impact can hopefully be minimized



Proposed Hybrid Programming

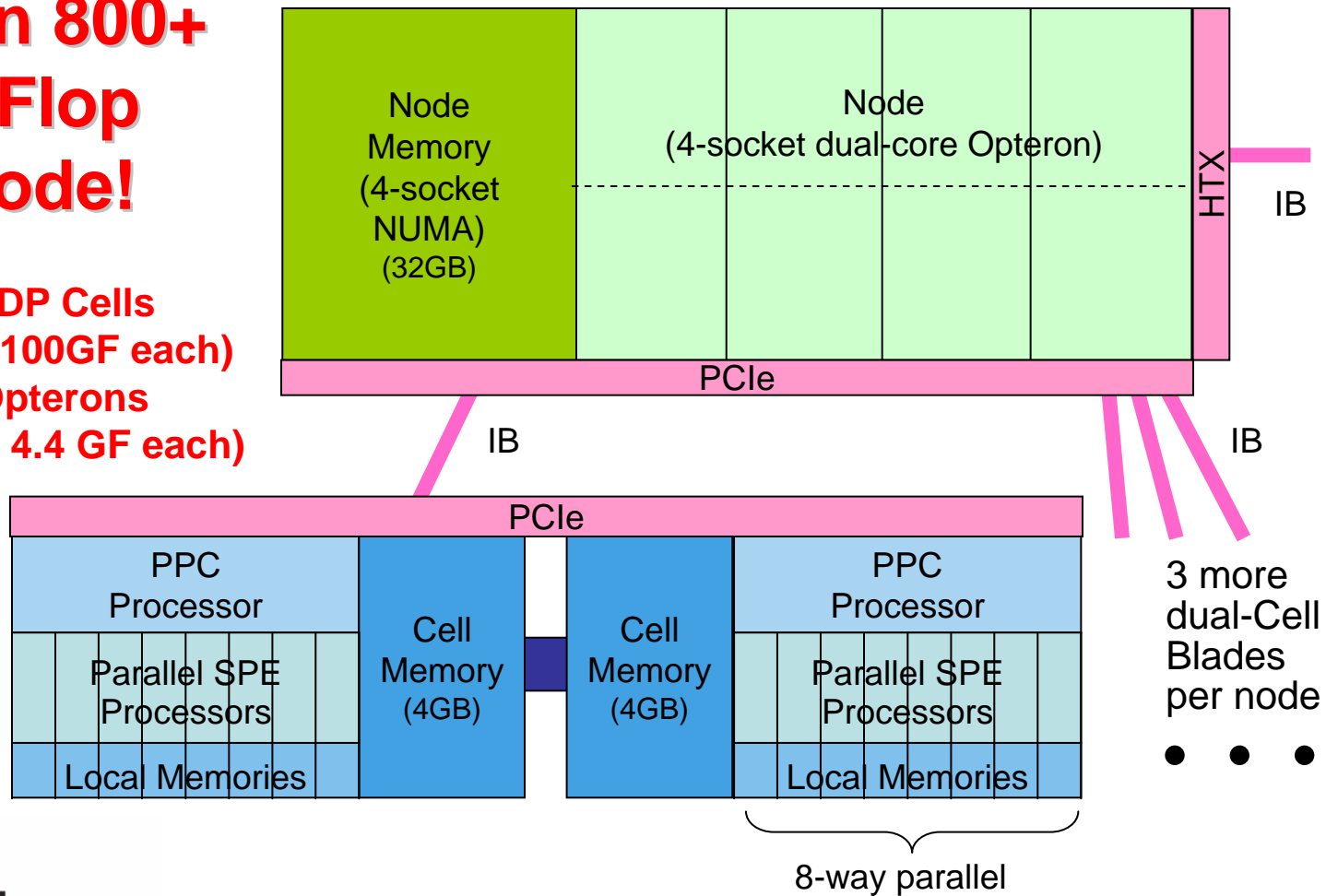
- A hybrid code would have 3 distinct cooperating code pieces
 1. Main code runs on an Opteron of a node
 - Uses node memory and passes data up/down and across the cluster to other Opteron processes
 - Runs “non-accelerated normal” cluster code
 2. A Cell PPE code
 - Uses blade memory and passes data up/down and monitors SPE computations
 3. A Cell SPE code
 - Performs most of computation
 - Uses SPE local store memory and DMAs to/from Cell blade common memory
- Developer architects the cooperation now; tools may be able to help some in the future



Roadrunner Heterogeneity

**An 800+
GFlop
Node!**

**8 eDP Cells
(~100GF each)
8 Opteron
(~ 4.4 GF each)**



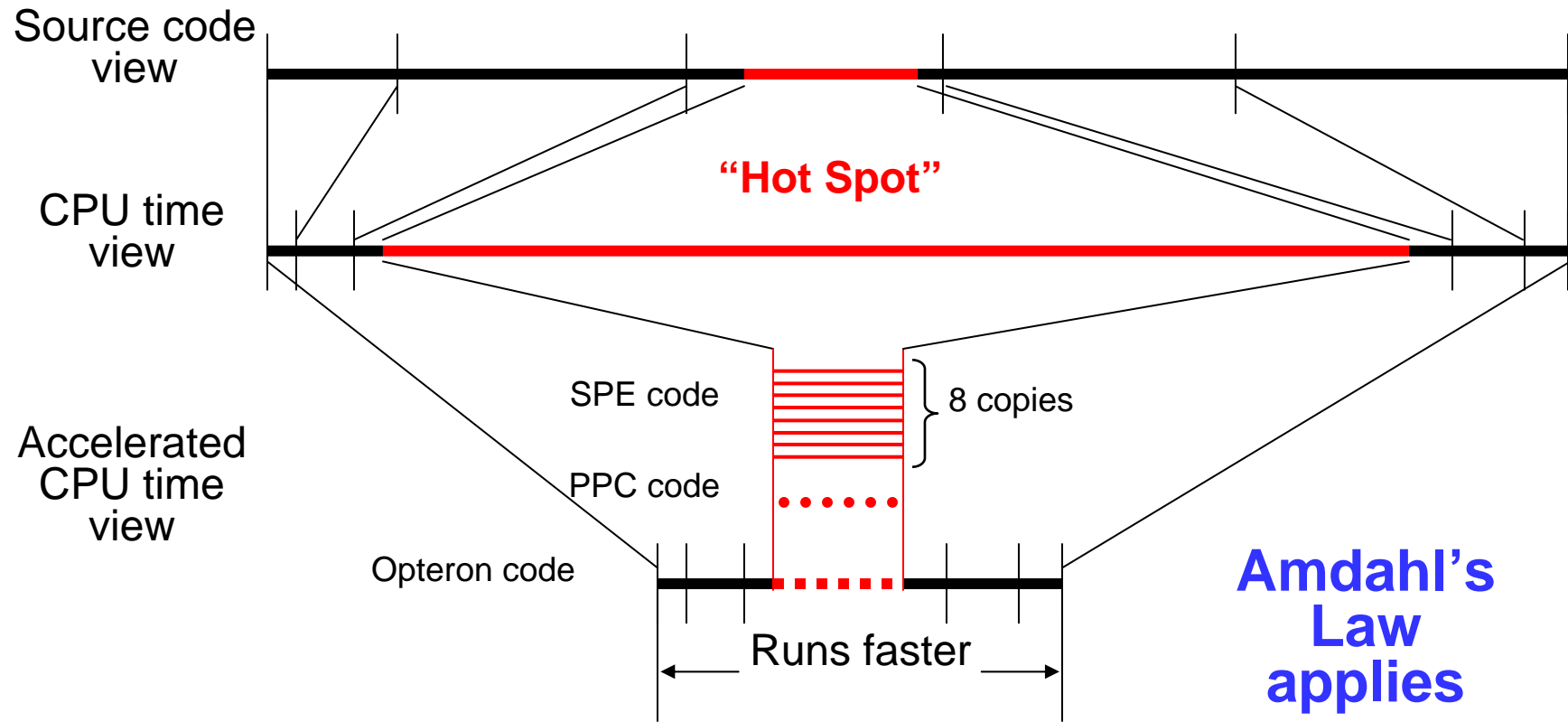


Hybrid Programming

- Decomposition of an application for Cell-acceleration
 - Opteron code
 - Runs non-accelerated parts of application
 - Participates in usual cluster parallel computations
 - Controls and communicates with Cell PPC code for the accelerated portions
 - Cell PPC code
 - Works with Opteron code on accelerated portions of application
 - Allocates Cell common memory
 - Communicates with Opteron code
 - Controls and works with its 8 SPEs
 - Cell SPE code
 - Runs on each SPE (SPMD) (MPMD also possible)
 - Shares Cell common memory with PPC code
 - Manages its small Local Store (LS) memory, transferring data blocks in/out as necessary
 - Performs vector computations from its LS data
- Each code is compiled separately (currently)



Cell Acceleration & Speedup





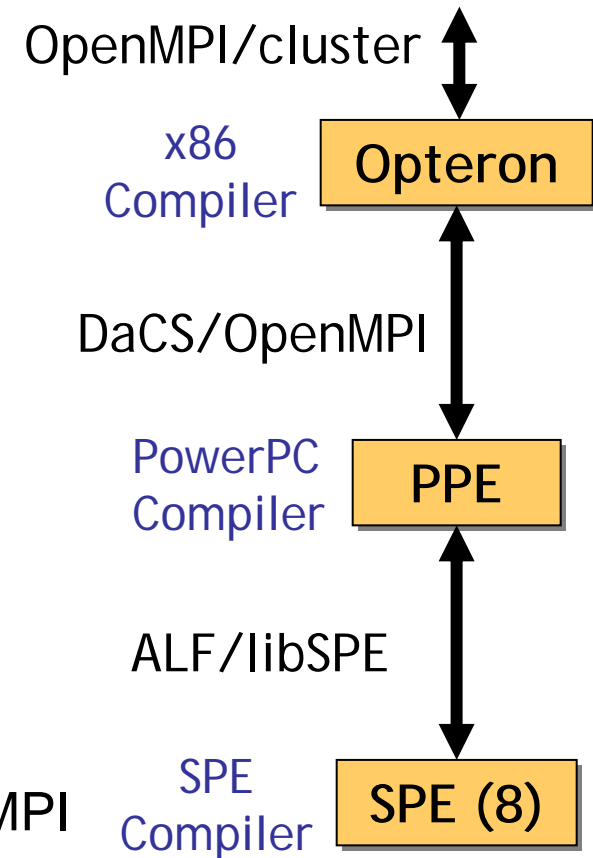
Using Cells Effectively

- Hide Cell blade communications costs
 - Load-compute-store using Cell blades would have to amortize the communications costs
 - All of a Cell blades memory can be transferred over IB 4X SDR in ~10-15 seconds
 - Compute phase should be of order of 2 mins or longer
 - Promote the computation onto the Cell blades
 - May require passing small amounts back to/from and between Opterons and other Cell blades
- Dealing with SPE local store size
 - Stream work blocks through the SPEs to/from Cell blade memory in a multi-buffered sequence
 - Overlap DMA reads, compute, & DMA writes
 - Application-specific software data caching for non-regular data
 - Text overlays for large programs



Roadrunner Hybrid APIs

- Computational Library (ALF w/ IBM)
 - Data partitioning
 - Task & work queue management
 - Process management
 - Error handling
- Communication Library (DaCS w/ IBM)
 - Data movement & synchronization
 - Process management & synchronization
 - Topology description
 - Error handling
 - First implementation may leverage OpenMPI



Opteron-Cell Programming Environment



- Minimum requirements:
 - Job launch & control, including delivery of executable image
 - I/O and error forwarding
 - Asynchronous data communication, DMA & MP styles
 - Double-buffered data transfers with computation
 - Synchronization primitives
- “Simple” Leverage Approach is Open MPI, but it...
 - Doesn’t deliver executables to Cell Blades
 - Currently has some lingering problems with heterogeneous MPI_Comm_spawn()
 - Opteron->PPC
 - Makes attached accelerator explicit
 - 2 levels of communications



IBM/LANL Communication API

- API being developed to meet minimum requirements.
 - Support Roadrunner's IB connected Cell Blades
 - Primarily in C, but is friendly to C++ and F9x
- Hides the particulars of the interconnect fabric
 - more future-proof.
- Processor topology and reservation system
 - Allows precise process placement for MPMD
 - Good for managing communications links and NUMA issues
 - Adapts to future hardware configurations
- Not specific to Cell or Roadrunner

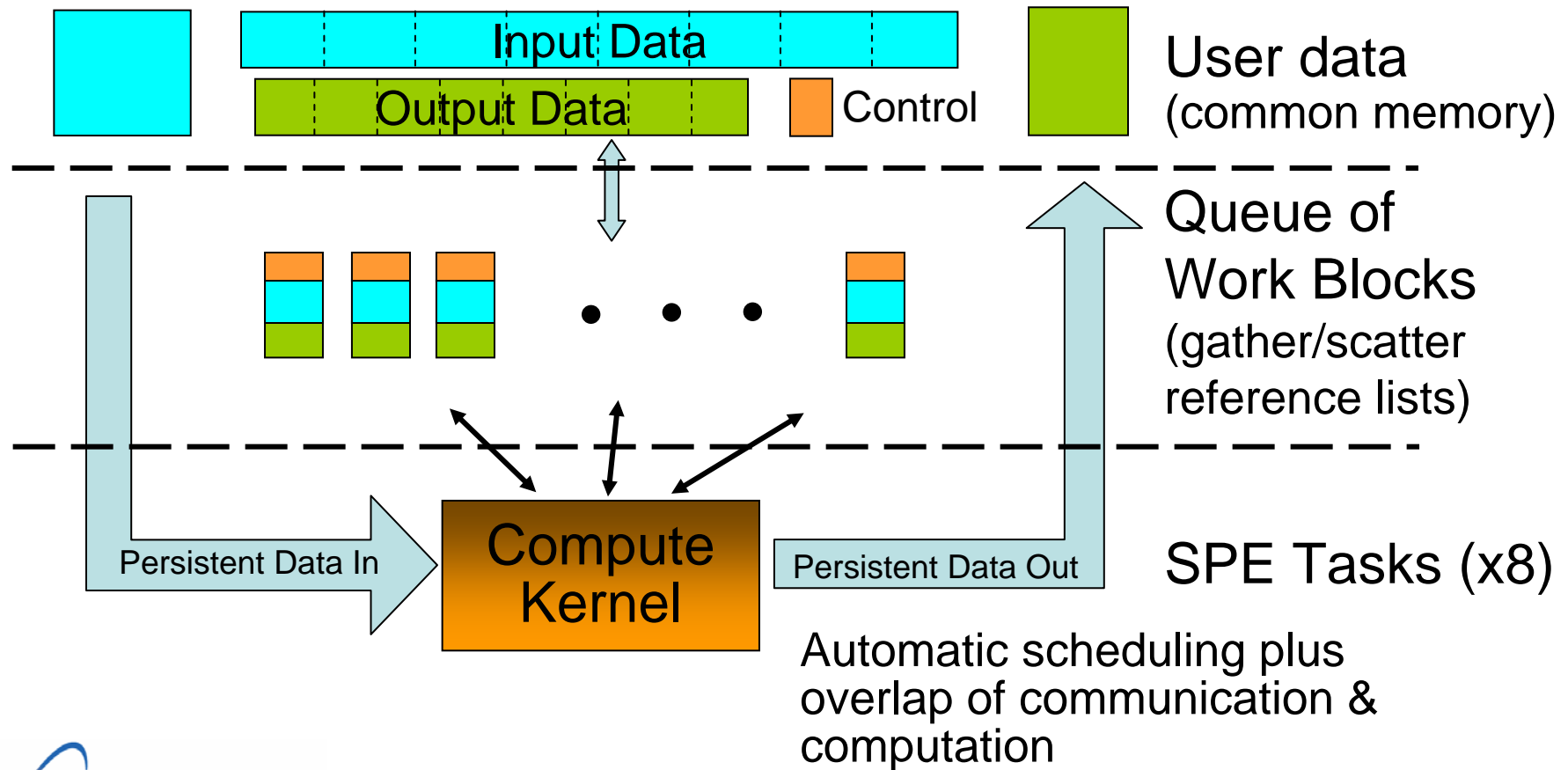


Work Queue API

- High-level API
 - Should be good for data-parallel operations
 - Option to programming to the hardware using low-level intrinsics
- Implements a common communication paradigm to increase programmer productivity and robustness
- Automatically partitions work among accelerators.
- Overlaps DMA operations with compute kernel
- No extra data copies
 - Working data defined by gather/scatter lists



Work Queue Paradigm



Los Alamos National Laboratory
Hybrid/Heterogeneous Experience
& Target Applications



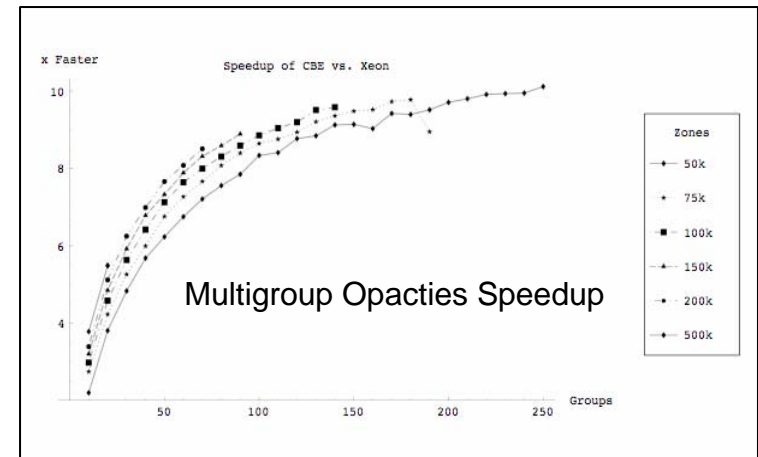
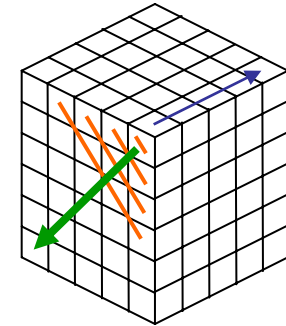
LANL's Heterogeneous Architectures Experience

- We have been pursuing heterogeneous computing for several years
 - Results thus far (GPU, FPGA, Cell) are mostly encouraging
 - Roadrunner is simply the first large-scale system hardware
- Focus on applications of direct interest to LANL
 - Develop algorithms and tools not just for Roadrunner but for heterogeneous computing in general
 - *Re-think* algorithms rather than simply *re-implement*
- Ultimate goal is improved simulation capabilities
 - Maybe “better” rather than simply “faster” is an option

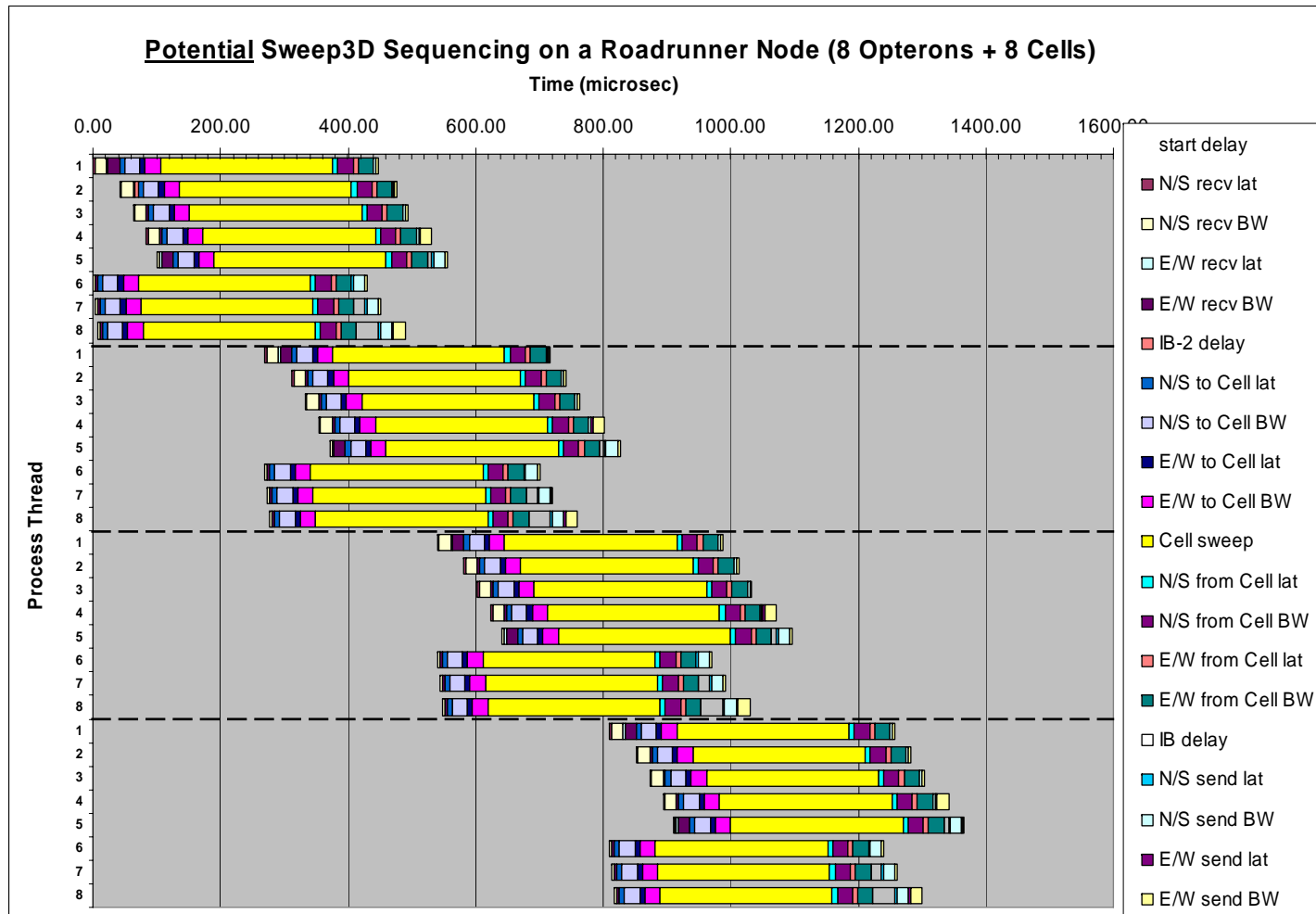


Initial LANL Cell Investigations

- Transport
 - Neutron transport via S_n (PartiSn)
 - Sweep3D – 5x speedup on Cell seen in IBM/PNNL work
 - Sparse linear solver (PCG) – not started
 - Radiation transport via Implicit Monte Carlo (Milagro)
 - Up to 10x speedup for opacity calculation on Cell
- Particle methods
 - Molecular Dynamics (SPaSM)
 - 7x speedup on Cell on simple MD
- Eulerian shock-hydrodynamics
 - No Cell work started yet
 - Pagosa
 - Data parallel CM-5 implementation
 - Advanced methods
 - Mesh-free / particle methods



Overlapped Communications & Computation





Roadrunner is the Path to the Future

- View Roadrunner as simply “rev. 1” of large-scale hybrid/heterogeneous computing
 - rev. 2 \Rightarrow processors on boards in PCI-E slots
 - rev. 3 \Rightarrow processors in traditional CPU sockets
 - rev. 4 \Rightarrow different processors on die
- Develop algorithms and software tools for heterogeneous computing – not just RR.



Thank you for your attention

Questions & Answers?



Accelerated Roadrunner

“Connected Unit” cluster
144 quad-socket
dual-core nodes
(138 w/ 4 dual-Cell blades)
InfiniBand interconnects

8,640 dual-core Optrons
• **76 TeraFlops**
16,560 eDP Cell chips
• **1.7 PetaFlops Cell**

