

The Los Alamos Roadrunner Petascale Hybrid Supercomputer

Overview of Applications, Results, and Programming

Roadrunner Technical Seminar Series

18 March 2008

John A. Turner

Group Leader, Computational Physics Group ([CCS-2](#))
Computer, Computational, & Statistical Sciences Division ([CCS](#))
turner@lanl.gov

Work presented was performed by a large team of Roadrunner project staff!



Operated by the Los Alamos National Security, LLC for the DOE/NNSA

LA-UR-08-2412



The Roadrunner Technical Seminar Series

- March 13: "Roadrunner Platform Overview," Ken Koch, CCS-DO
- March 18: "Overview of Applications, Results, and Programming," John Turner, CCS-2
- March 19: "Overview of Modeling, Performance, and Results," Darren Kerbyson, CCS-1
- April 10: "Application 1: SPaSM," Sriram Swaminarayan, CCS-2
- April 22: "Application 2: VPIC," Ben Bergen, CCS-2
- April 23: "Application 3: SWEEP3D," Mike Lang, CCS-1
- April 24: "Application 4: Milagro I," Tim Kelley, CCS-2
- May 6: "Application 5: Milagro II," Paul Henning, CCS-2
- May 8: "Application 6: DNS," Jamal Mohd-Yusof, CCS-2
- May 29: "Panel Discussion: Hybrid Computing Programming Models"
- June 3: "Panel Discussion: Future Platforms"

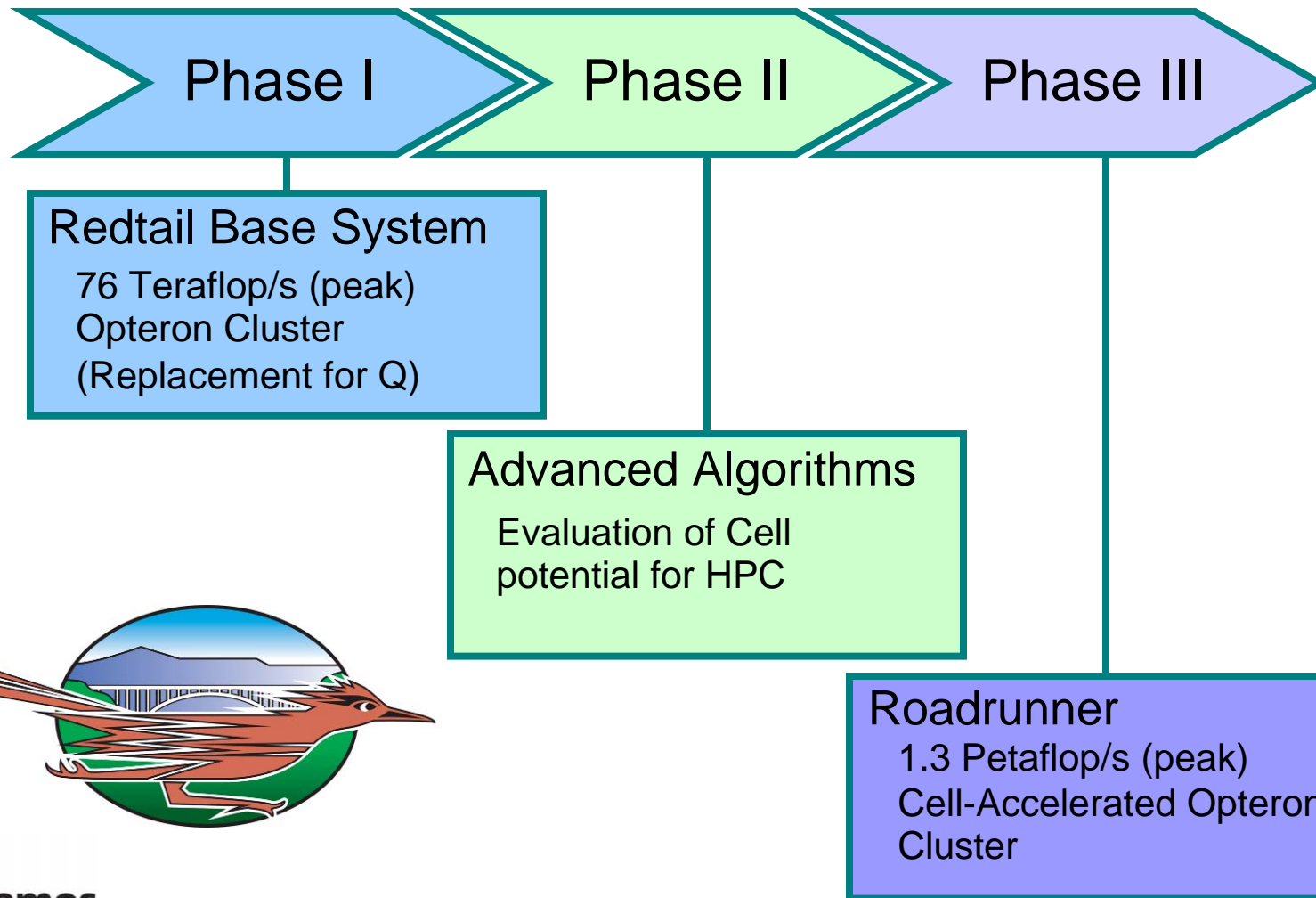
System Overview



Operated by the Los Alamos National Security, LLC for the DOE/NNSA



Roadrunner Project Phases



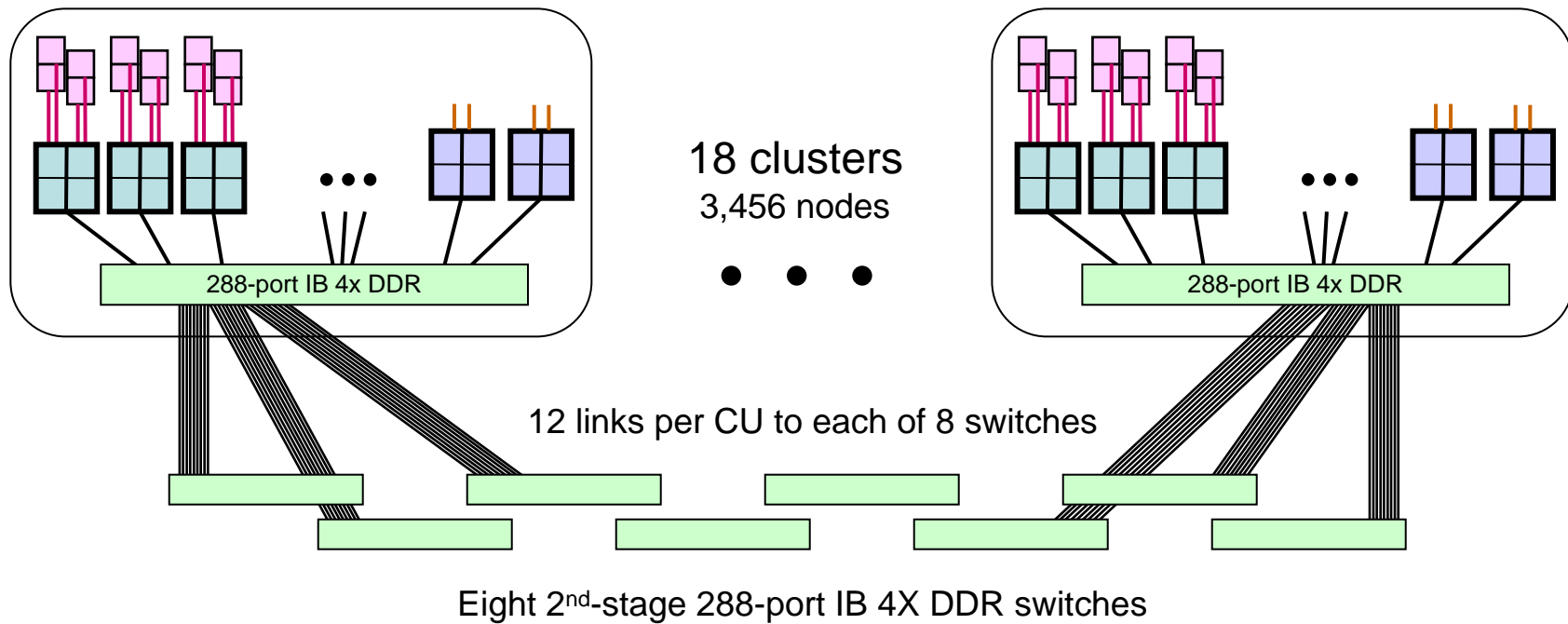
The LANL Roadrunner system has two primary goals.

- Science at Scale
 - *Multi-scale unit physics for weapons & open science*
 - Validate model assumptions
 - Better understand physics
 - Cross-validate physics models at overlapping resolutions
 - *Run at Petascale (machine and phenomenological scales)*
- Advanced Architecture for algorithms and applications
 - *Target select physics and work on algorithms and implementations*
 - Convert and re-think algorithms for hybrid and many-core architectures
 - *Provide faster solutions, improved accuracy*
 - *Incrementally accelerate existing ASC integrated codes to target key uncertainties*
 - *Target focused simulations, not general usage*
 - Focus is more on improved science (predictivity) than speeding up production jobs

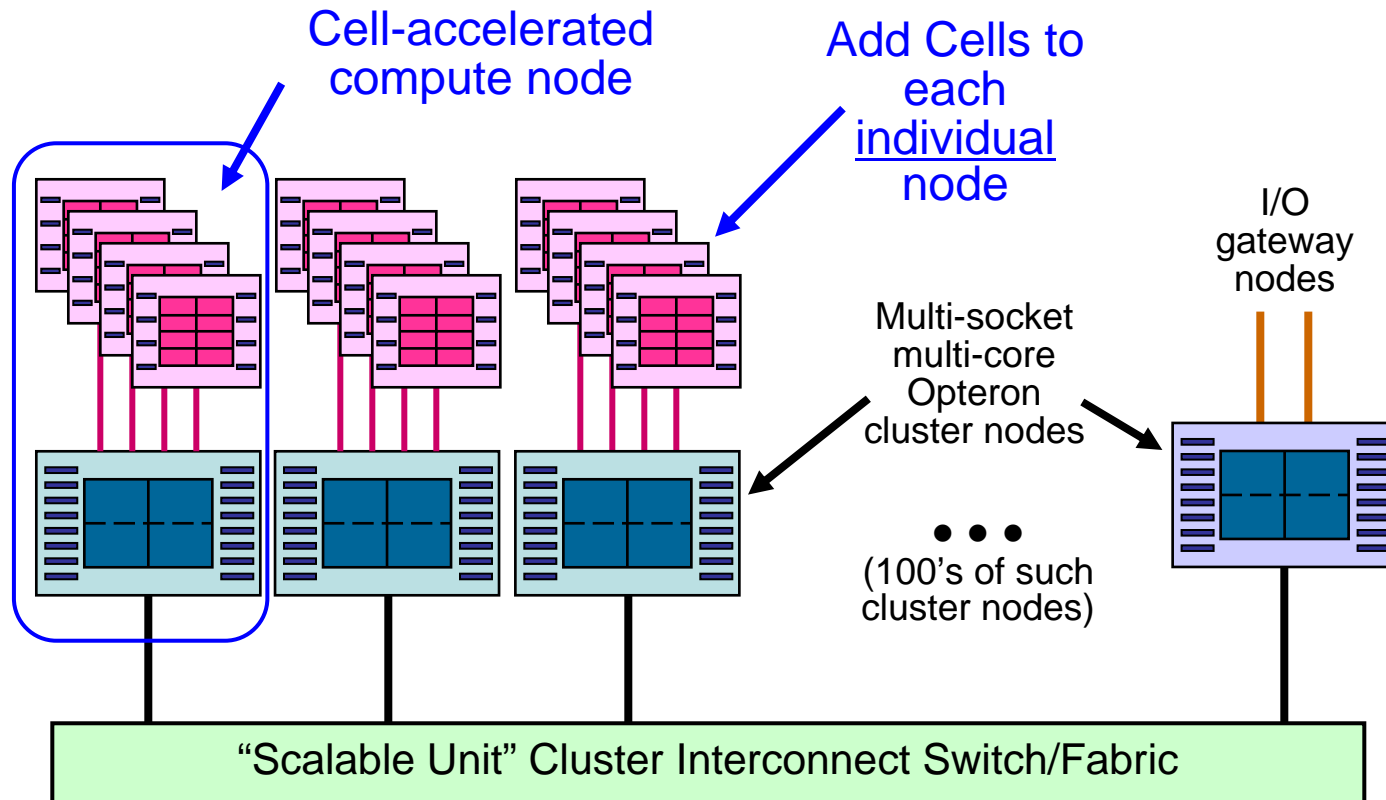
Roadrunner is a hybrid petascale system of modest size delivered in 2008

Connected Unit cluster
180 compute nodes w/ Cells
12 I/O nodes

12,960 Cell eDP chips \Rightarrow 1.3 PF, 52 TB
6,912 dual-core Opteron \Rightarrow 50 TF, 52 TB



Roadrunner is Cell-accelerated, not a cluster of Cells

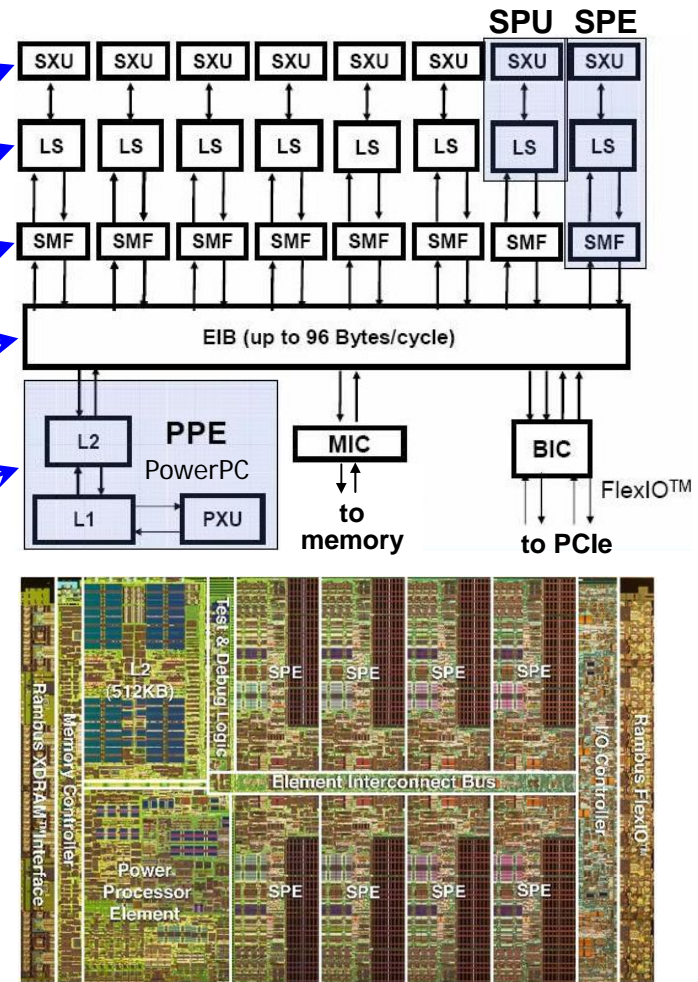


Node-attached Cells is what makes Roadrunner different!

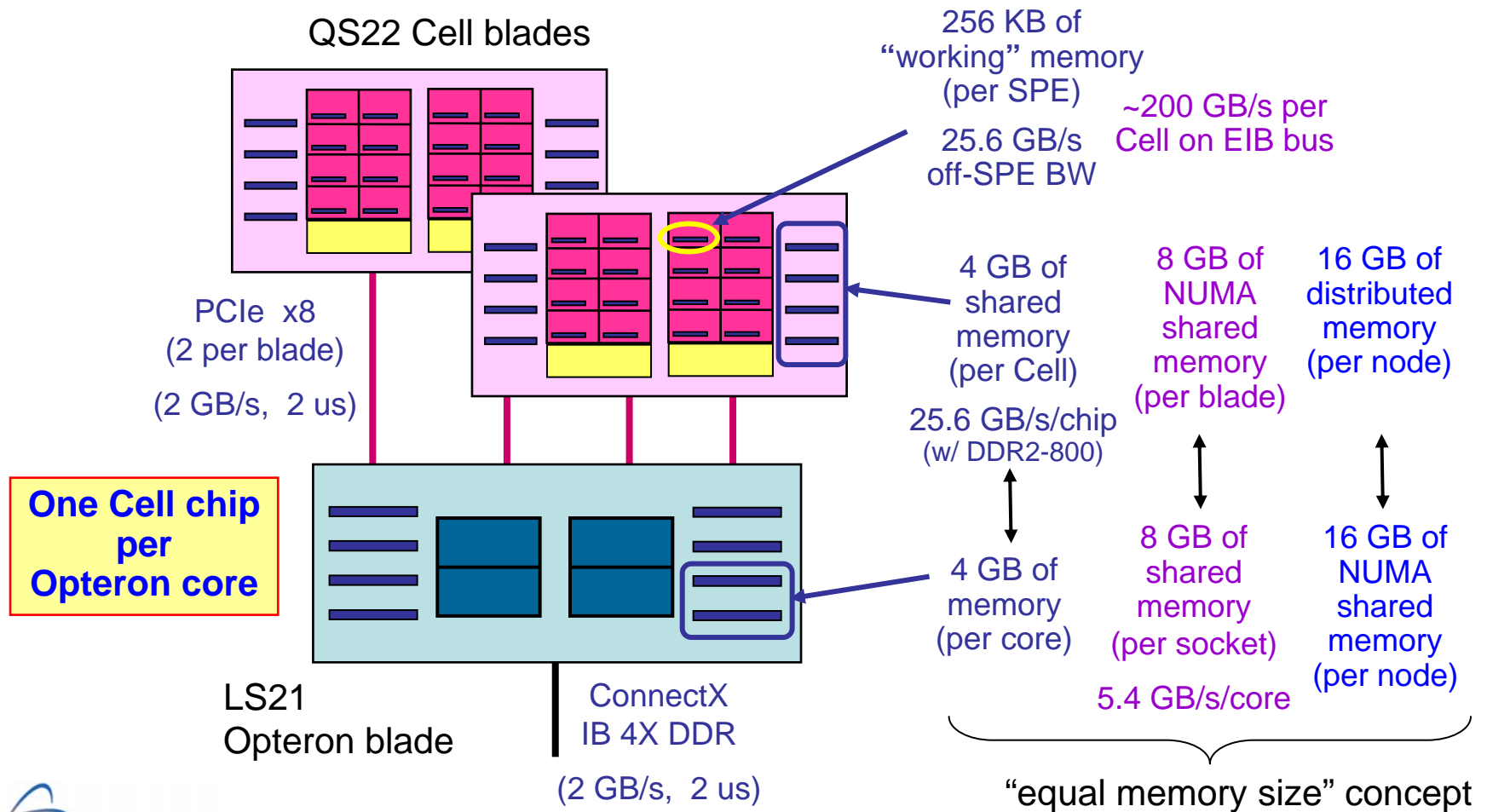
The Cell Broadband Engine (CBE)* is an 8-way heterogeneous parallel processor.

* trademark of Sony Computer Entertainment, Inc.

- Developed by Sony-Toshiba-IBM
 - used in Sony PlayStation 3
- 8 Synergistic Processing Elements (SPEs)
 - 128-bit vector engines
 - 256 kB local memory (LS = Local Store)
 - DMA engine (25.6 GB/s)
 - Chip interconnect (EIB)
 - Run SPE code as POSIX threads (SPMD, MPMD, streaming)
- PowerPC PPE runs Linux OS
- Original Cell chip:
 - 204 GF/s SP, ~15 GF/s DP
 - up to 2GB XDR memory @ 25.6 GB/s
- eDP (enhanced Double-Precision) chip:
 - 204 GF/s SP, 102 GF/s DP
 - up to 16GB DDR2 memory @ 25.6 GB/s (w/ DDR2-800)



Roadrunner nodes have a memory hierarchy.



Cell is a harbinger of the future



Operated by the Los Alamos National Security, LLC for the DOE/NNSA

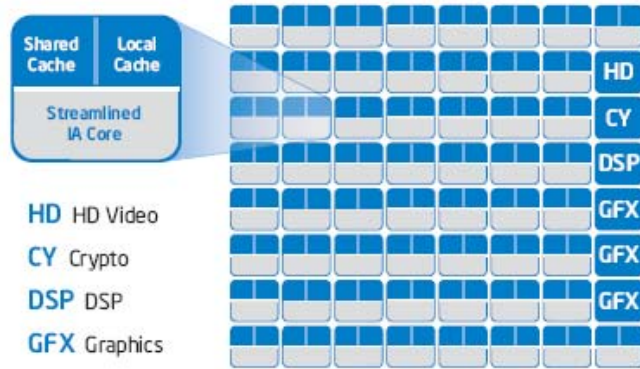


Cell is a harbinger of the future

(or, this idea isn't completely nuts)

Industry presentations show changing trends in processors

Intel's Microprocessor Research Lab

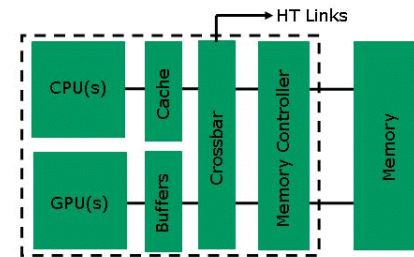


Intel's Visual Computing Group - Larabee



AMD Fusion

The Data Efficiency Benefits of Silicon-Level Integration



Expected Step-Function Improvement in Power/Performance

October 2006 Unleashing the Processing Powerhouse

nVidia G80 - 2006

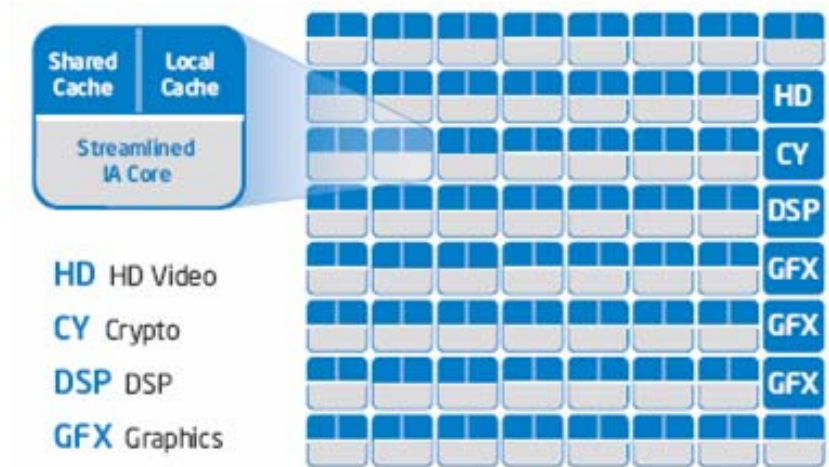


Taken from publicly available information



Intel Research on CPUs & GPUs and integration.

Q&A w/Jerry Bautista director of Intel Microprocessor Research Lab
The Future Of CPUs & GPUs
September 2006 • Vol.6 Issue 9



CPU: Isn't there always going to be more than one processor in a system?

JB: The tera-scale computing project is where people miss the point. *It's not necessarily a homogeneous collection of cores.* The Cell microprocessor has one big core and eight synergistic processing elements; that is already a hybrid. We could have a general-purpose core with fixed function add-ons. [...] **Our version of tera-scale is a hybrid machine.**

Programming



Operated by the Los Alamos National Security, LLC for the DOE/NNSA

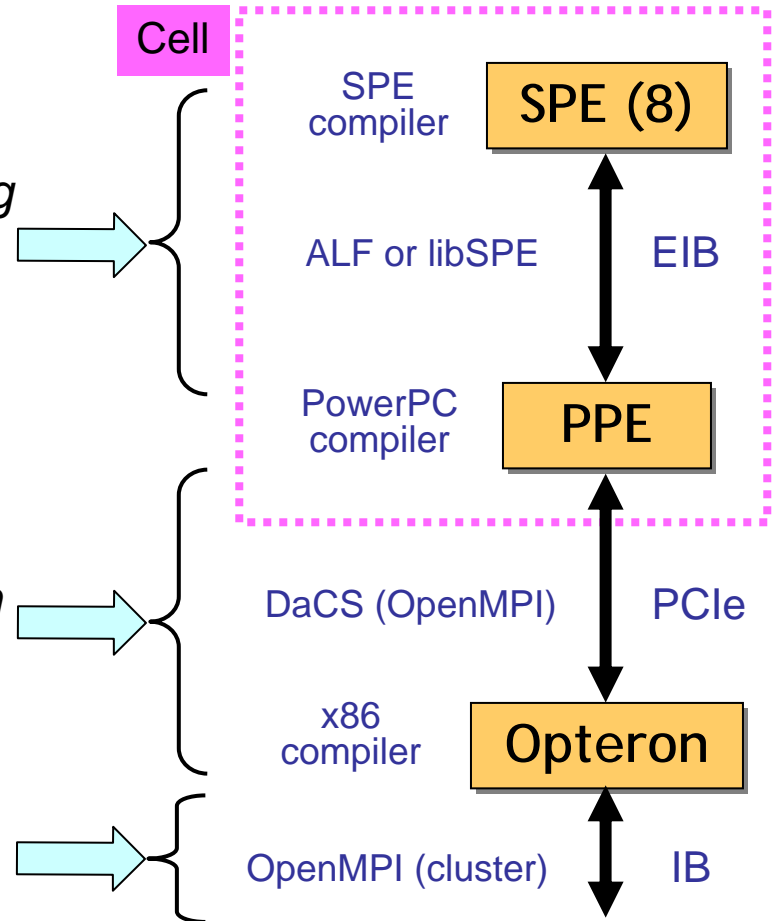


Three types of processors work together.

- parallel computing on Cell
 - *data partitioning & work queue pipelining*
 - *thread management & synchronization*

- remote communication to/from Cell
 - *data communication & synchronization*
 - *process management & synchronization*
 - *computationally-intense offload*

- **MPI remains as the foundation**



Basics of Cell / hybrid programming are straightforward.

- build issues are manageable
 - some used [GNU autoconf](#), some rolled their own solutions
- two compilers
 - [GNU Compiler Collection](#) (gcc)
 - [IBM XL C/C++ for Multicore Acceleration](#) (XLC)
- debugging – not just printf
 - [GNU debugger](#) (gdb)
 - can be used everywhere (Opteron, PPU, and SPU)
 - [Totalview](#) is an option on Opteron and PPU
- performance tools available
- byte-swapping is not an issue

[IBM SDK for Multicore Acceleration](#), includes all tools needed to get started

- all of the above (except Totalview) and more

Typical steps in Cell code development

- study algorithm complexity
- analyze data flow, data layout, data locality
- partition algorithm
- develop PPE control, PPE scalar code
- develop PPE control, partitioned SPE scalar code
 - *communication, synchronization, latency handling*
- vectorize
 - *transform SPE scalar code to SPE SIMD code*
- re-balance computation / data movement
- further optimization
 - *PPE SIMD, system bottle-necks, load balance*

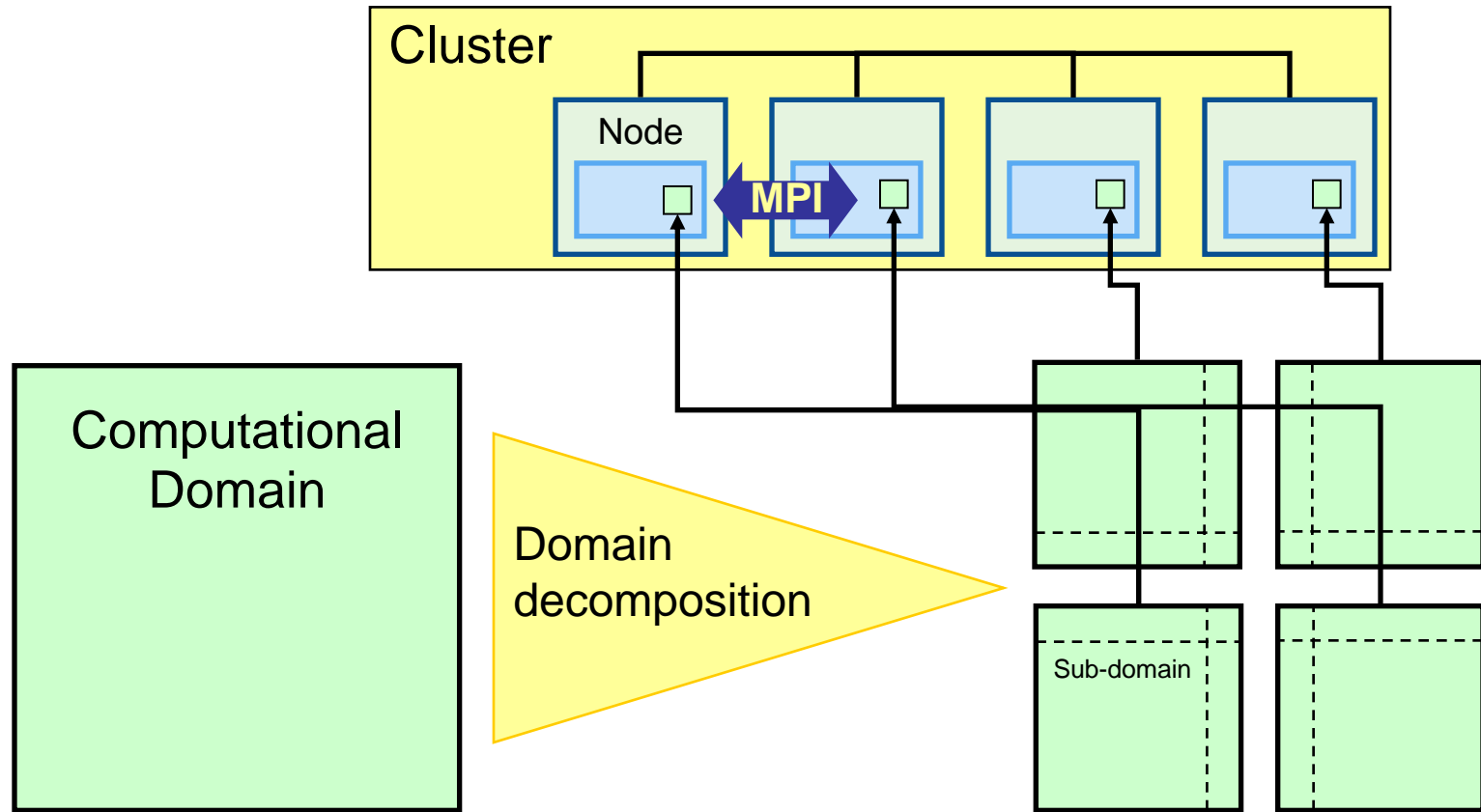
Hybrid programming models



Operated by the Los Alamos National Security, LLC for the DOE/NNSA

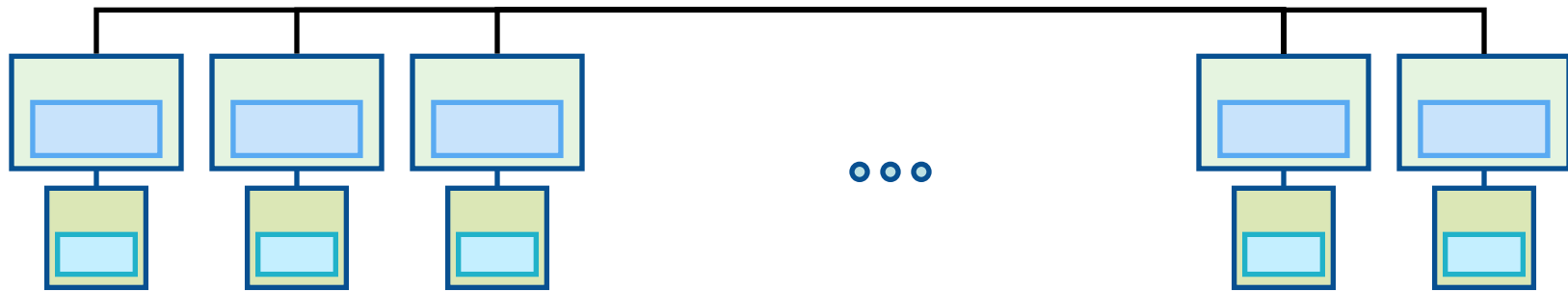


Distributed memory parallelism



Basic models

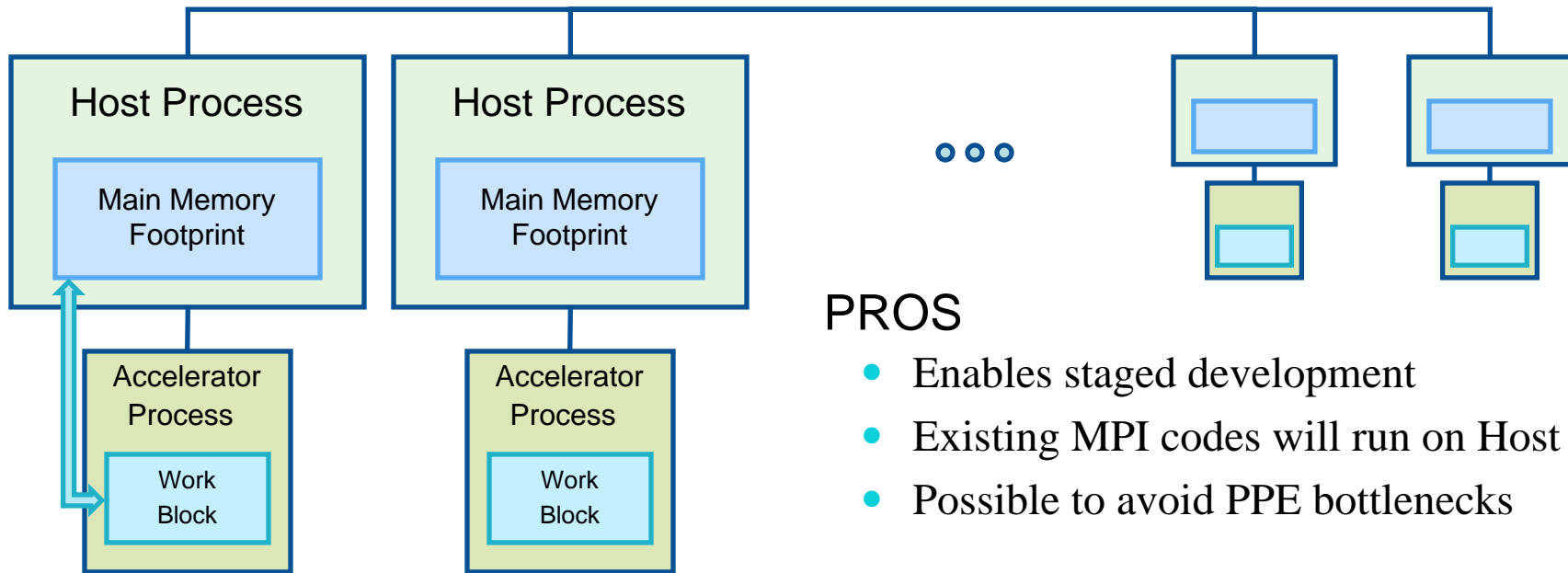
Host-centric model



Accelerator-centric model (inverted memory model)



Host-centric model (function offload)



Synchronous or asynchronous function offload to accelerator

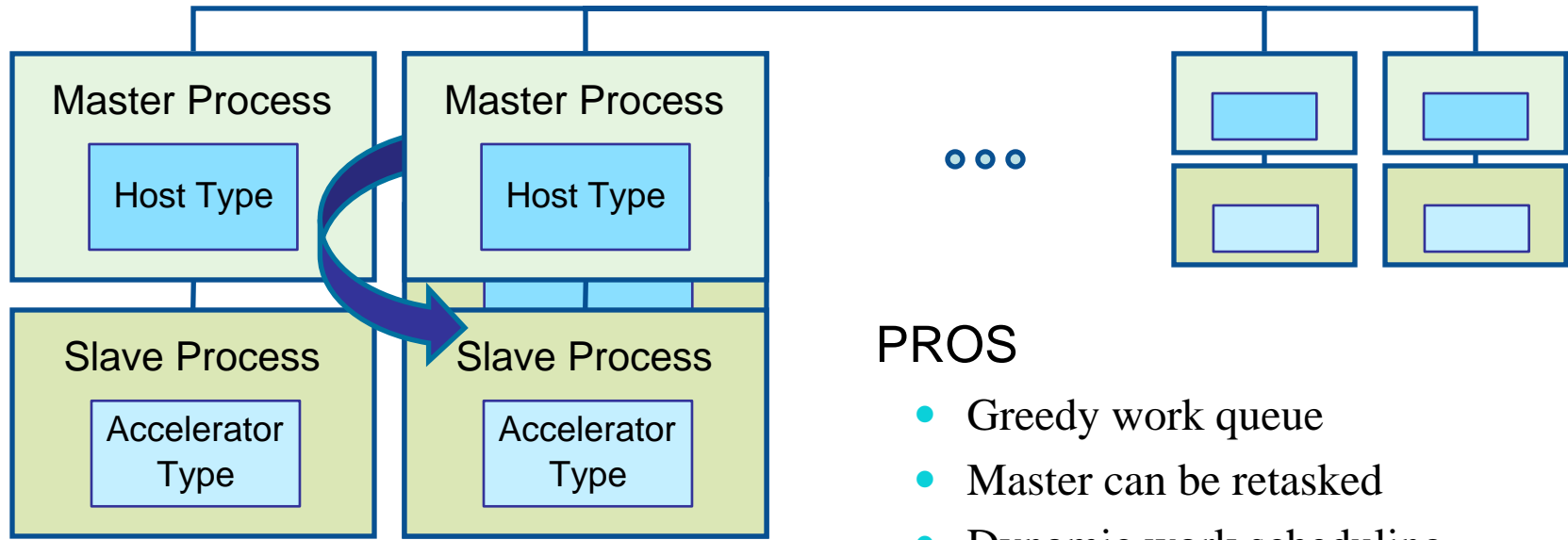
PROS

- Enables staged development
- Existing MPI codes will run on Host
- Possible to avoid PPE bottlenecks

CONS

- Potential data-movement bottleneck

Host-centric model (work stealing)



First step towards true heterogeneous processing

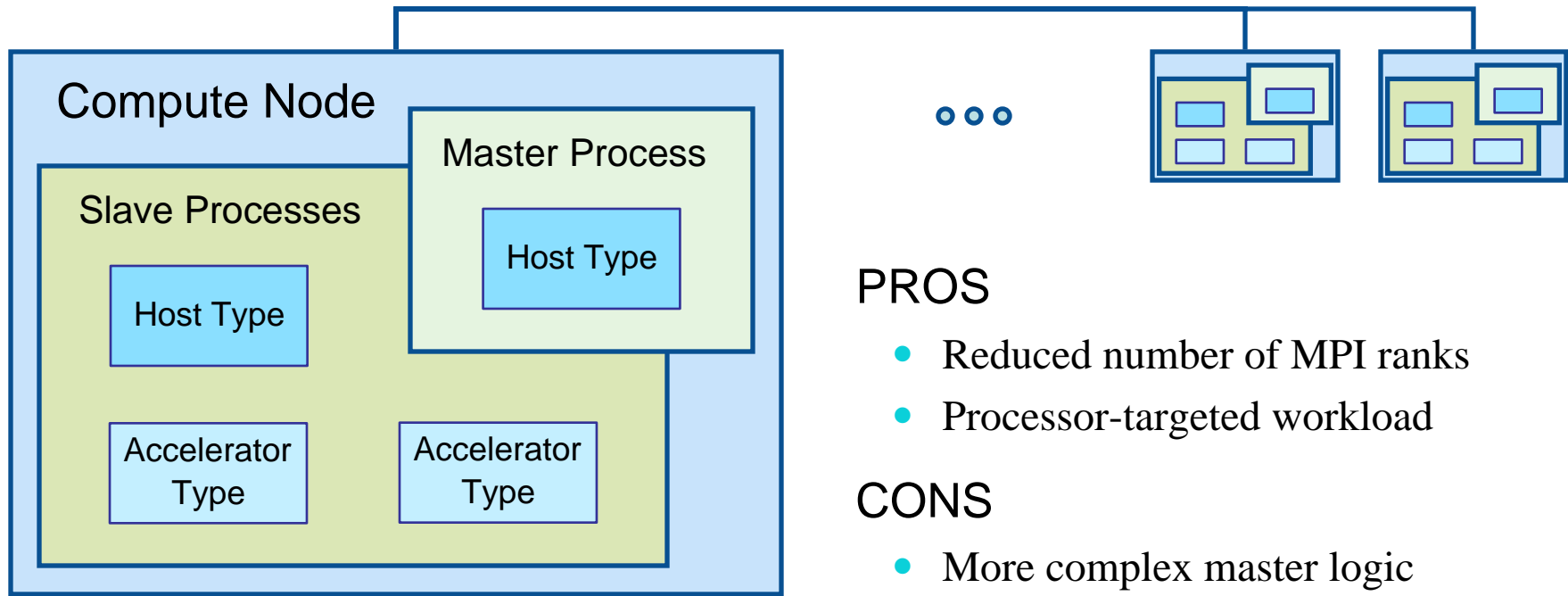
PROS

- Greedy work queue
- Master can be retasked
- Dynamic work scheduling

CONS

- More complex master logic

Control process model



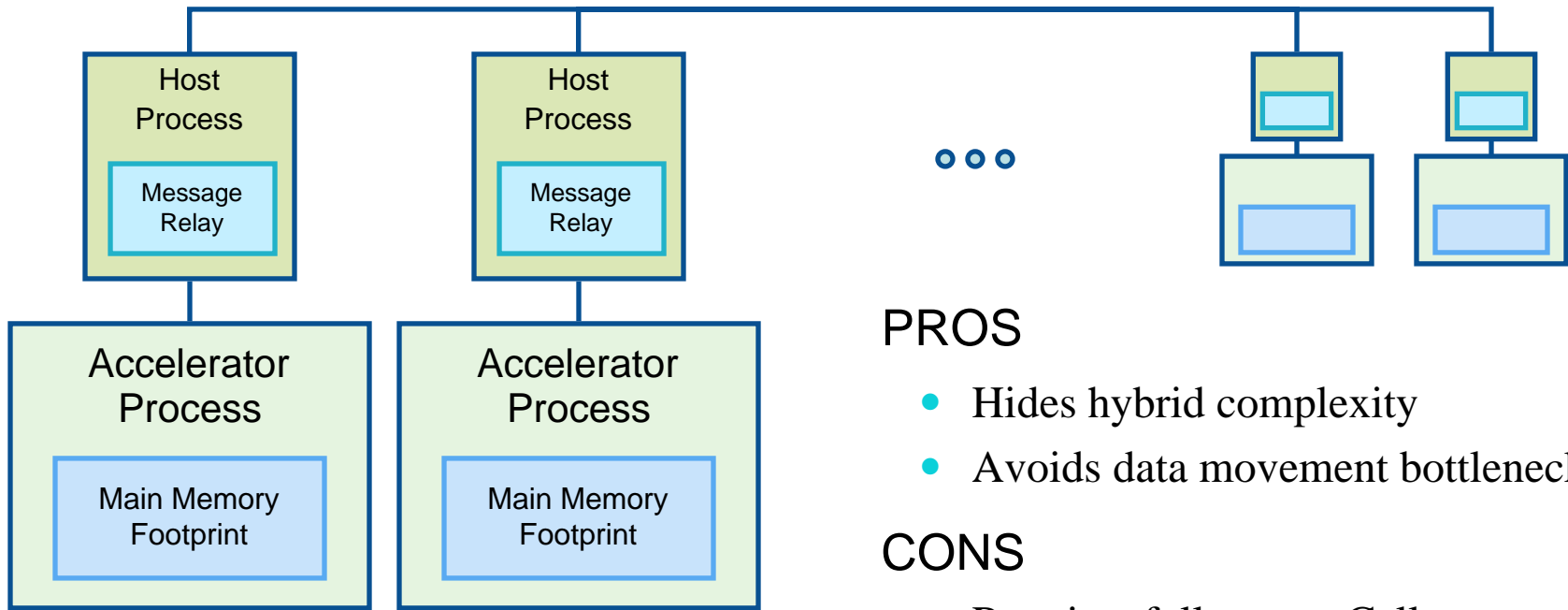
PROS

- Reduced number of MPI ranks
- Processor-targeted workload

CONS

- More complex master logic
- Software support (DaCS/MPI)

Accelerator-centric model



MPI traffic relayed through host

PROS

- Hides hybrid complexity
- Avoids data movement bottleneck

CONS

- Requires full port to Cell
- Potential PPE bottleneck

What is “different” about programming Roadrunner? Will it continue to be different?

- A “rank” in an MPI parallel application becomes a set of processors with varied strengths and unequal access to memories and communication.
 - *This is similar to multi-core, then many-core and the future trends expected in micro-processors as core counts rapidly increase*
- Data layout and SIMD instructions are important.
 - *This is also true on all current processors for good cache and SIMD unit utilization.*
 - *Not dissimilar to previous vector codes, except now short vectors and mainstream.*
- Explicit programmer management of local store.
 - *Focus effort on data locality and computational intensity, which benefit all processors*
 - *Exposes opportunities to overlap communication and computation*
 - *Non-uniform memory hierarchies*

***Roadrunner* embodies many key architectural trends, each in moderation.**

- *Roadrunner* has: multicore, short-vector SIMD, threads, heterogeneous instruction sets, local stores instead of caches, on-chip CPU/memory networks, remote accelerators and cluster computing.
- You can use any of these features as needed, without needing to go to extremes in any one of them.
- *Roadrunner's* scale and flexibility makes it an ideal base from which to explore the changing landscape of HPC
 - *But it also provides immediate benefits.*

Roadrunner offers applications a spectrum of programming models.

- Roadrunner has
 - *~3200 compute nodes, each with 2 dual-core Opteron*
 - *~6400 dual-core Opteron*
 - *~13k Opteron cores*
 - *~13k Cell processors, each with 8 SPEs*
 - *~100k SPEs*
- how many MPI processes?
 - *developer can use model most appropriate for application*

Selected applications



Operated by the Los Alamos National Security, LLC for the DOE/NNSA



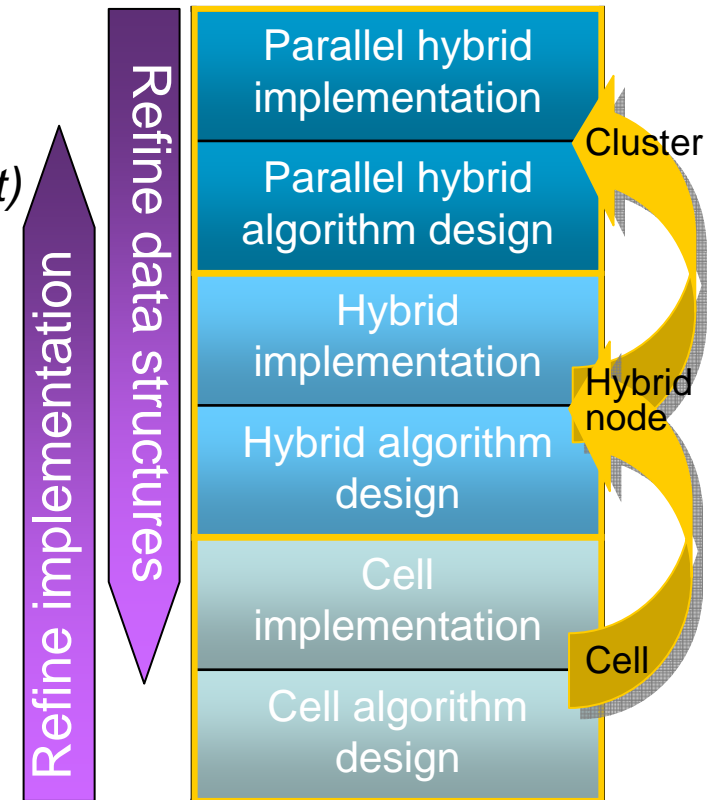
For the Assessment in October 2007, needed to answer the following questions...

Will Roadrunner be applicable to the current and future application workload?

1. Can we can program the Cell processor?
 - *Do sufficient tools (compilers, etc.) exist?*
2. Can we achieve significant performance gains on the Cell processor?
3. Can we program Opteron / Cell hybrid nodes?
4. Can we achieve significant performance gains in hybrid?

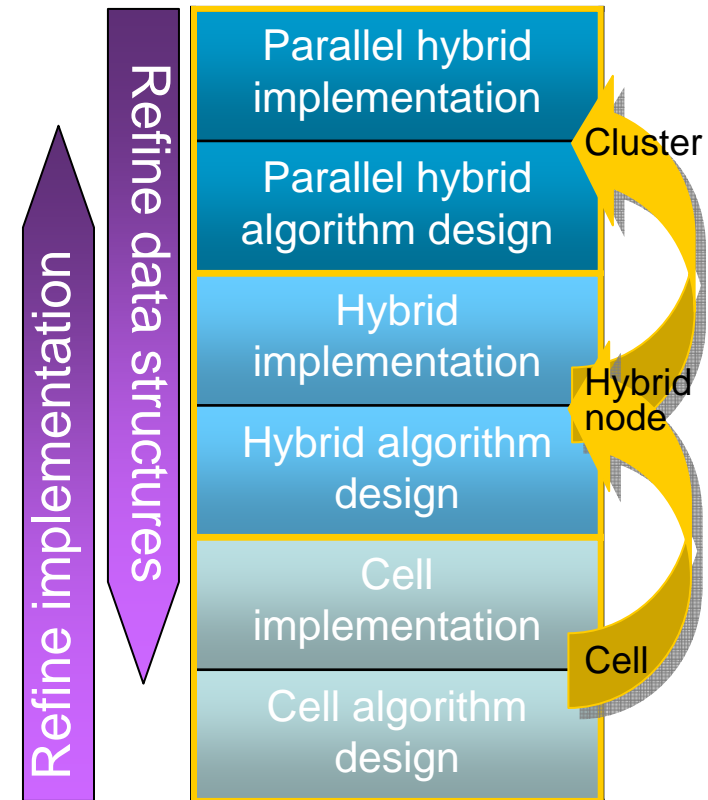
A few key application areas were targeted in FY07

- Transport
 - *PARTISN (Sn neutron transport)*
 - Sweep3D (benchmark code)
 - Sparse solver (PCG)
 - *MILAGRO (IMC thermal radiation transport)*
- Particle methods
 - *VPIC (Particle-In-Cell)*
 - SSE enabled
 - *SPaSM (molecular dynamics)*
 - Data parallel CM-5 implementation
- Eulerian hydro
 - *Direct Numerical Simulation (DNS)*
- Linear algebra
 - *LINPACK*
 - *Krylov subspace methods*
 - Preconditioned Conjugate Gradients (PCG)
 - Generalized Minimal Residuals (GMRES)



Each application team had to address common implementation issues.

- 3 cooperating programs
 - *some could simply be “relays”*
- Cells can't talk to other Cells, but only to it's host Opteron.
- Where do I put “main” logic & memory?
 - *Opteron or Cell?*
- How do I partition computation into streams of fixed-sized tiles or chunks for SPEs?
- How can I use DMA transfers and reusable work buffers in SPE local memory to keep streams flowing?



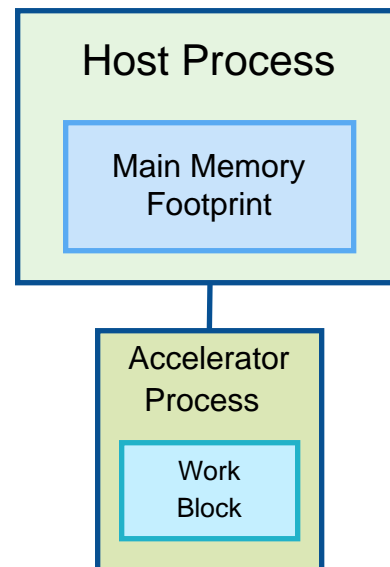
port
vs.
rewrite

Results presented today will focus on four codes.

<i>Code</i>	<i>Description</i>	<i>Type</i>	<i>Language</i>	<i>Lines of code</i>
<i>VPIC</i>	Fully-relativistic, charge-conserving, 3D explicit particle-in-cell code.	full app	C/C++	8.5k
<i>SPaSM</i>	Scalable Parallel Short-range Molecular Dynamics code, orig. developed for the CM-5.	full app	C	34k
<i>Milagro</i>	Parallel, multi-dimensional, object-oriented code for thermal x-ray transport via Implicit Monte Carlo on a variety of meshes.	full app	C++	110k
<i>Sweep3D</i>	Simplified 1-group 3D Cartesian discrete ordinates (Sn) kernel representative of the PARTISN neutron transport code.	kernel	C	2.5k

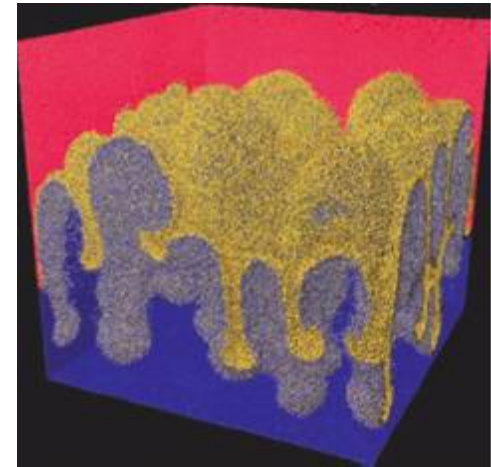
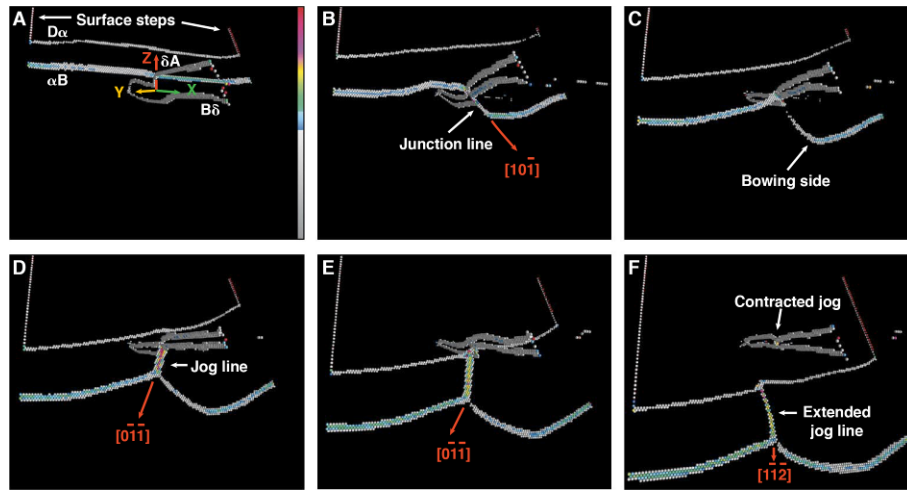
SPaSM

Host-centric, synchronous

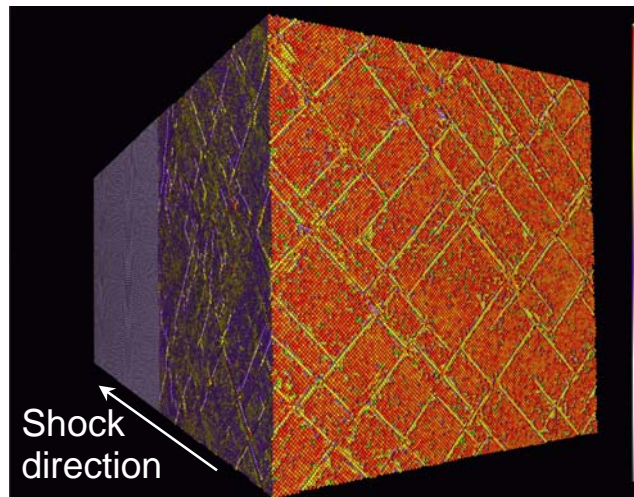


SPaSM simulations have helped to gain insight into fundamental materials and physics processes.

Strength of metals



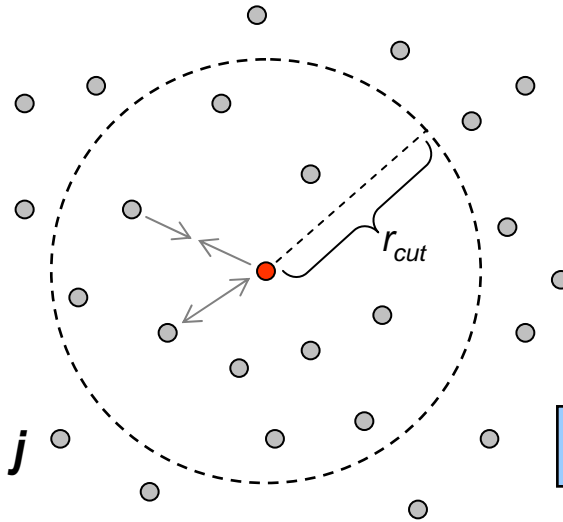
Shock compression of metals



Fluid instabilities and the onset of turbulence

Molecular Dynamics simulate particle interactions in Condensed Matter Physics

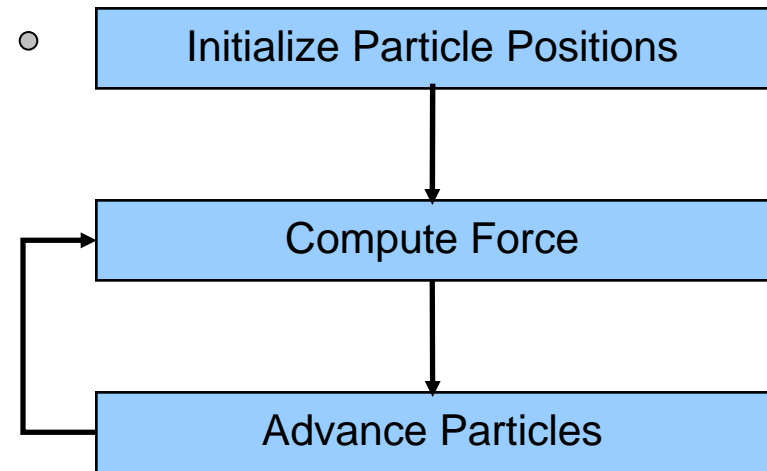
Force calculation



```

foreach particle  $i$ 
  foreach neighbor  $j$ 
    if  $r_{ij} < r_{cut}$ 
       $F_{ij}$  = interactions ( $i,j$ )
    end if
  end foreach
end foreach
  
```

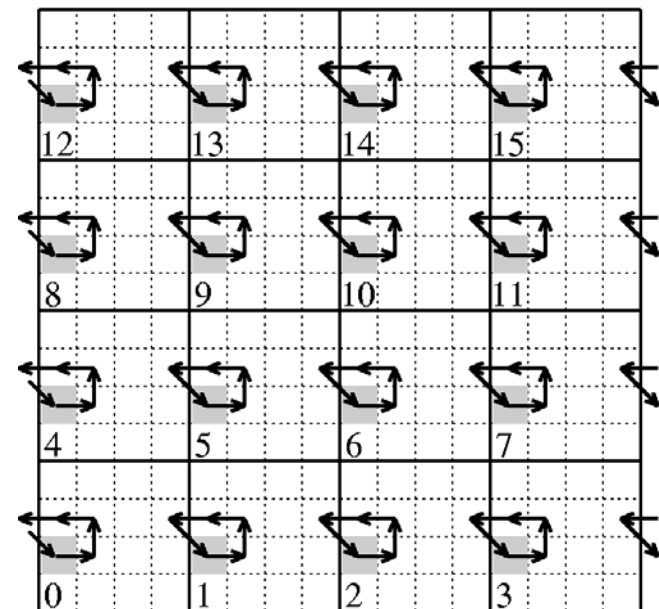
Time Iteration



Original SPaSM implementation

Designed when computation was more expensive than communication

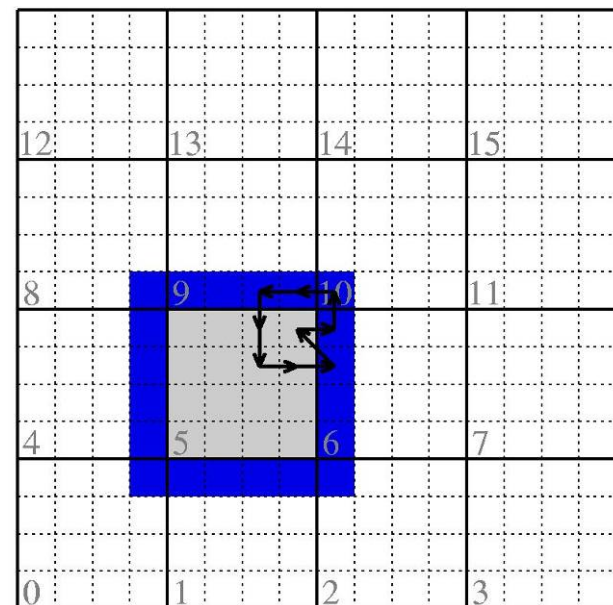
- MPI processes advance through cells in lock-step
- Pair-wise force interactions are symmetric
- MPI send() and recv() calls used every time a remote neighbor is encountered



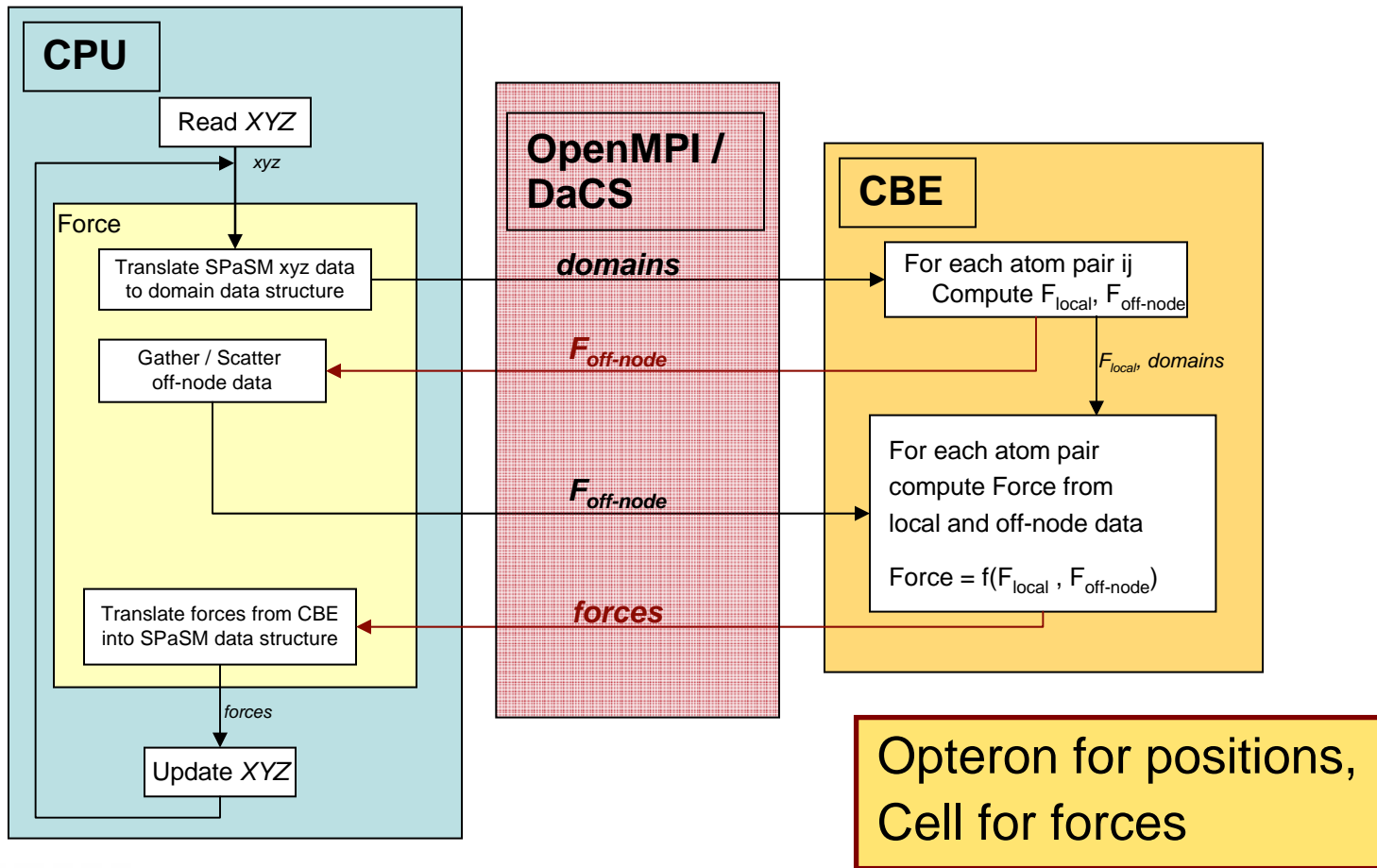
New implementation uses full ghost-cell buffering

Reduces latency with fewer messages and allows for more straightforward thread-level parallelism

- Blue ghost-cell region updated outside of particle interaction loop using MPI calls
- SPE threads can compute force interactions asynchronously without inter-node communication

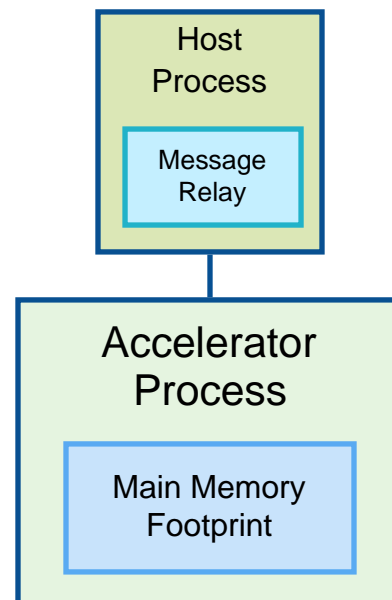


SPaSM hybrid implementation minimizes Cell PPE.



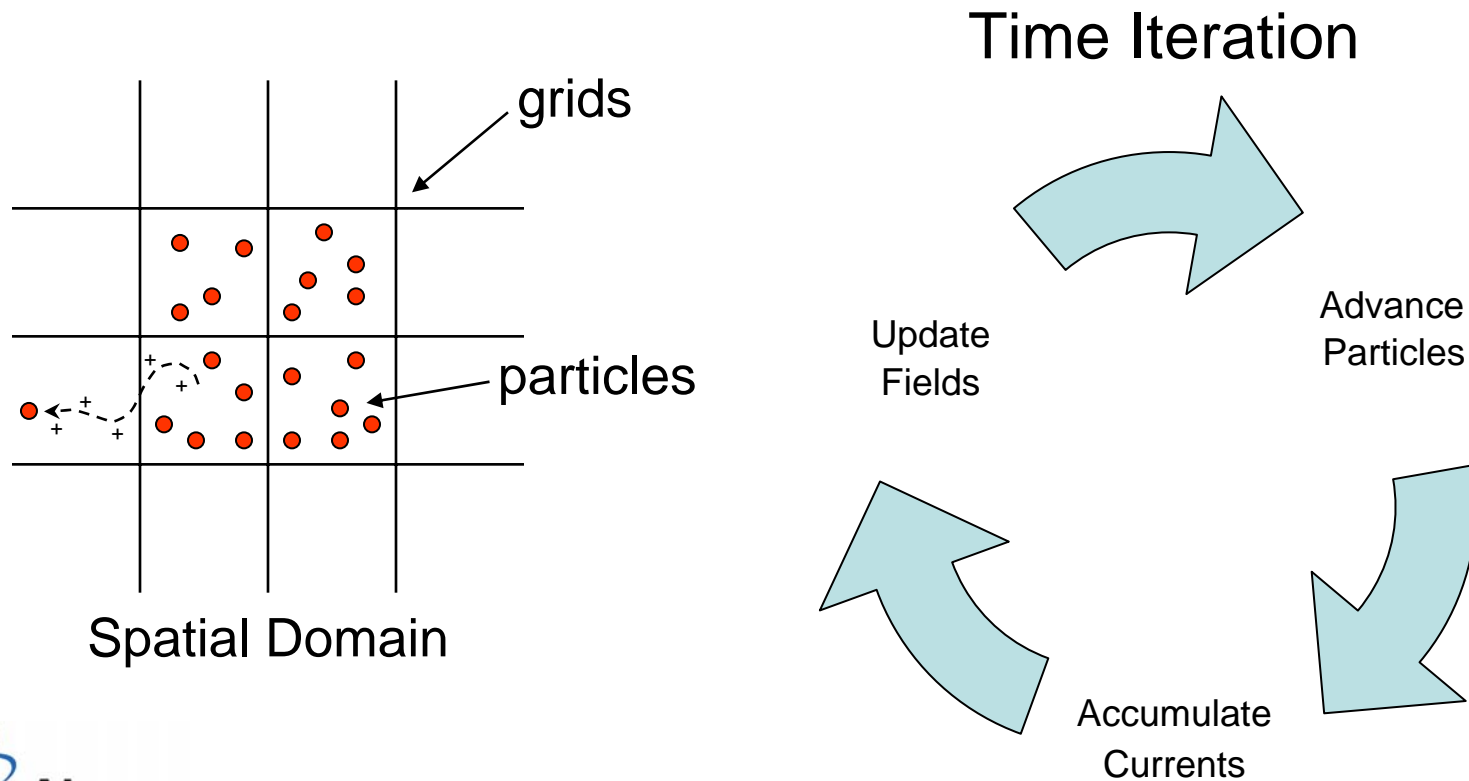
VPIC

Accelerator-centric



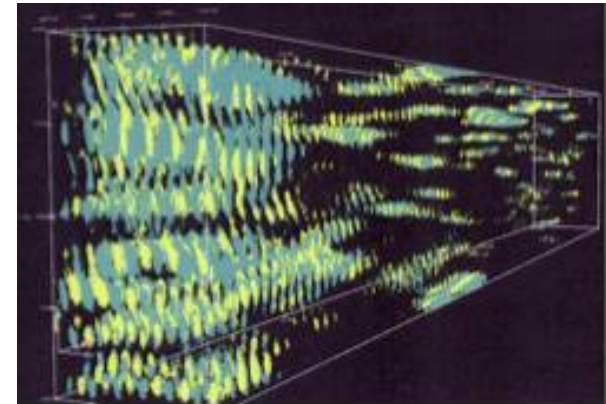
VPIC uses the particle-in-cell method for plasma simulation.

- Frank Harlow published one of the earliest papers on PIC methods in Los Alamos Scientific Library report LAMS-1956



VPIC performs 3D, fully-relativistic, charge-conserving, explicit particle-in-cell plasma simulation.

- ~9k lines of C/C++ (and some Assembly)
- highly-efficient on current hardware
 - *utilizes Streaming SIMD Extension (SSE) instructions*
 - buried in C++ vector class



```
inline v4float operator ++( v4float &a ) {
    v4float b;
    __m128 t;
    float one = 1.;
    t=_mm_load_ss(&one);
    t=_mm_add_ps(a.v,_mm_shuffle_ps(t,t,0));
    a.v = t;
    b.v = t;
    return b;
}
```

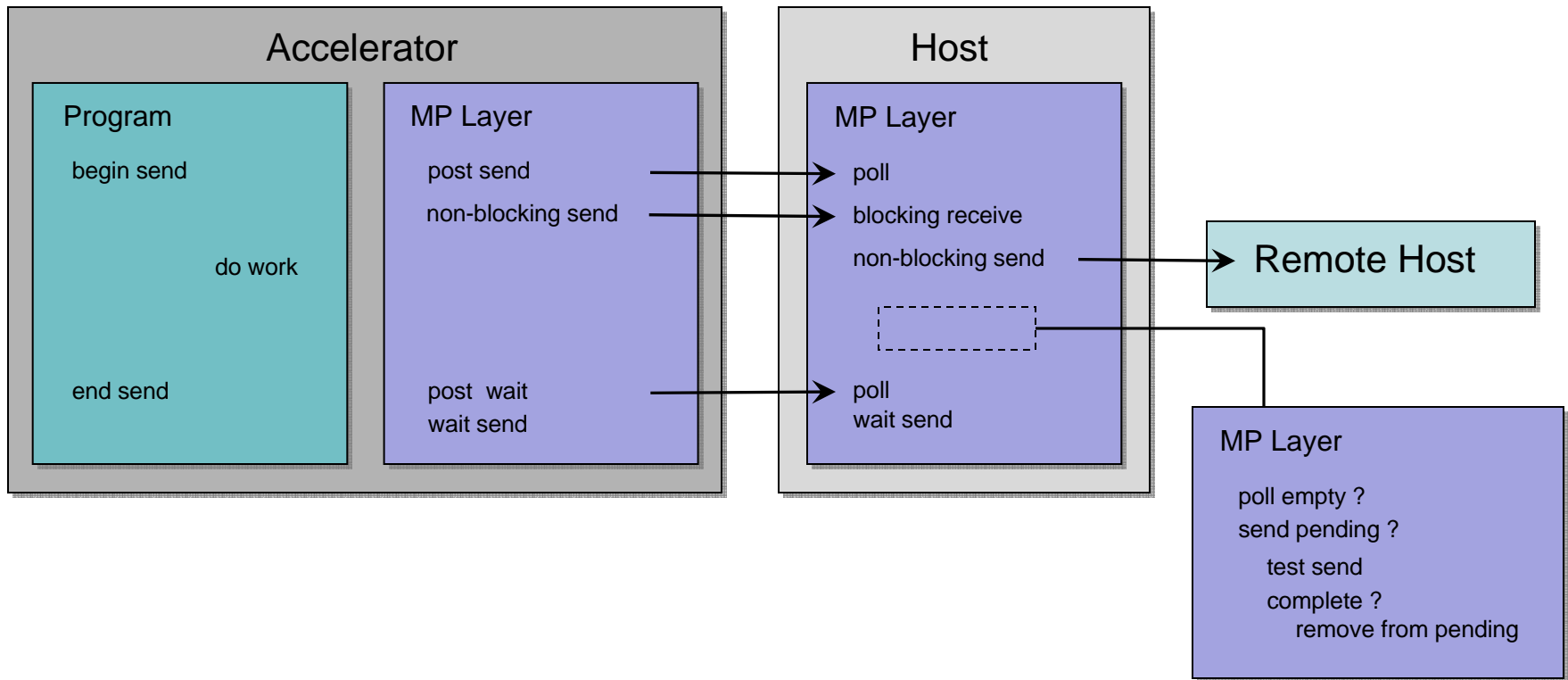
SSE instructions

```
inline v4float operator ++( v4float &a ) {
    vec_float4 a_v = a.v;
    v4float b;
    a_v=spu_add( a_v,spu_splats(1.f) );
    a.v = a_v;
    b.v = a_v;
    return b;
}
```

libSPE SPU intrinsic

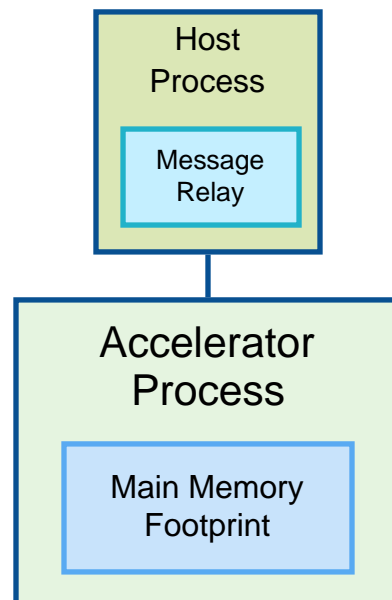
VPIC hybrid implementation uses message-passing relay to minimize Opteron.

- example: non-blocking Send



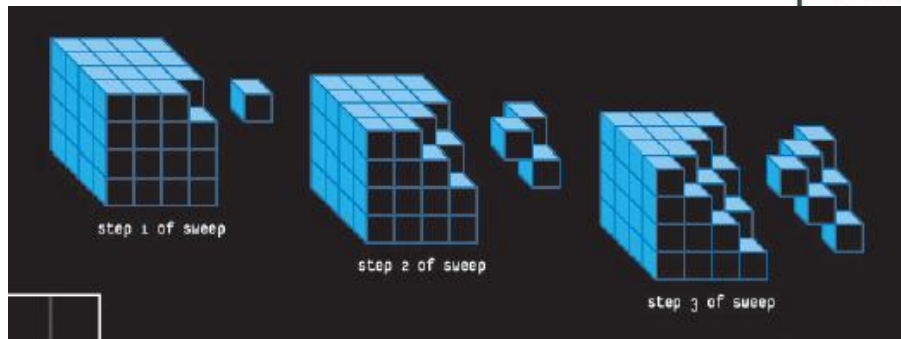
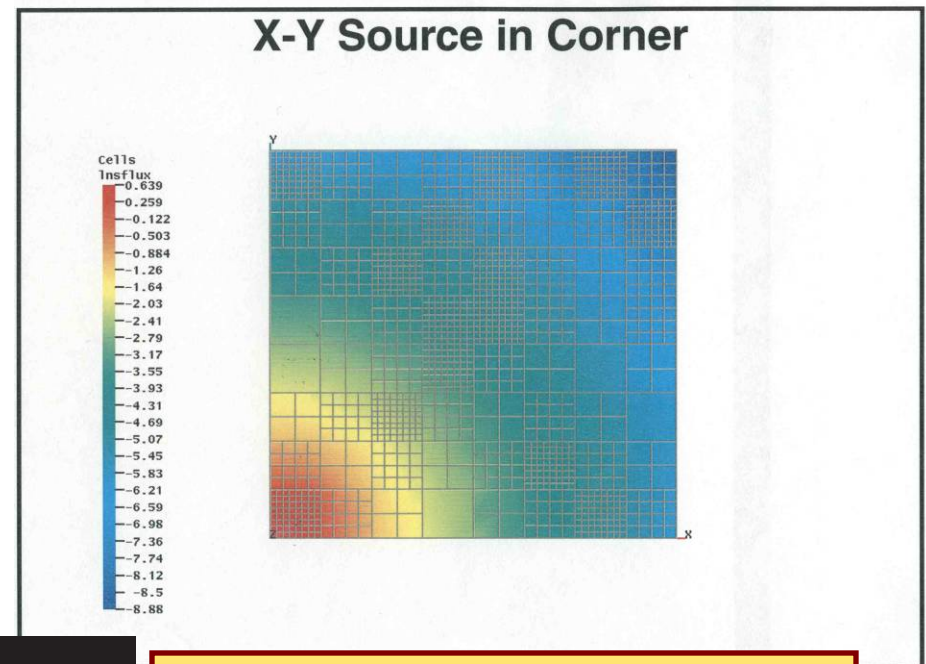
Sweep3D

Accelerator-centric



PARTISN (PARAllel Time-dependent S_N) simulates transport of neutrons and gammas.

- available through RSICC
 - *Radiation Safety Information Computational Center*
- S_N (discrete ordinates)
- multi-dim, multi-geometry
- block Adaptive Mesh Refinement (AMR) Meshes
- parallel



Sweep3D is a kernel representative of the computationally-intensive wavefront algorithm used in PARTISN.

PAL implementation of Sweep3D for Cell

- Cell-centric implementation
 - *standard MPI decomposition with the SPE as the processing element, with all computation performed on SPE*
 - *PPE and Opteron act only as communication relays for SPE's*
- uses PAL “Cell Messaging Layer” (CML) for communications
 - *Micro MPI for the SPE's, open-sourced (GPL license)*
 - *For Details of CML see:*
 - *Scott Pakin. Receiver-initiated Message Passing over RDMA Networks. IPDPS 2008, Miami, FL, April 2008.*
- only application thus far to make extensive use of SPE-SPE communication



Milagro

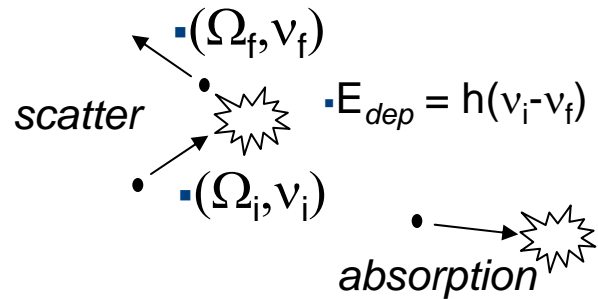
Host-centric (work stealing)

Implicit Monte Carlo simulates the evolution of a radiation field over time

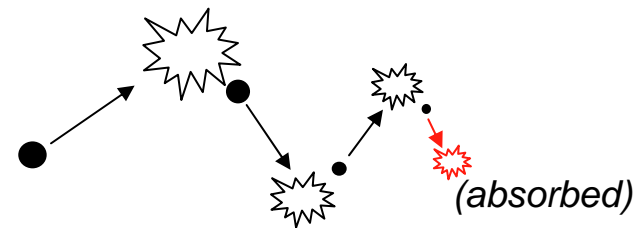
- Particles are
 - Emitted by sources (including blackbody material medium)
 - Tracked through material medium
 - Medium scatters; particles deposit energy and momentum
 - Material properties discretized on mesh: optical, thermodynamic
- Time steps used to linearize radiation-matter interaction
- The particle's track consists of discrete events (terminii):
 - Collision (absorption)
 - Mesh cell boundary (escape)
 - Weight cutoff (statistically insignificant, typ. absorbed next step)
 - Step end (end of time step, recorded in census for next time step)

Monte Carlo events

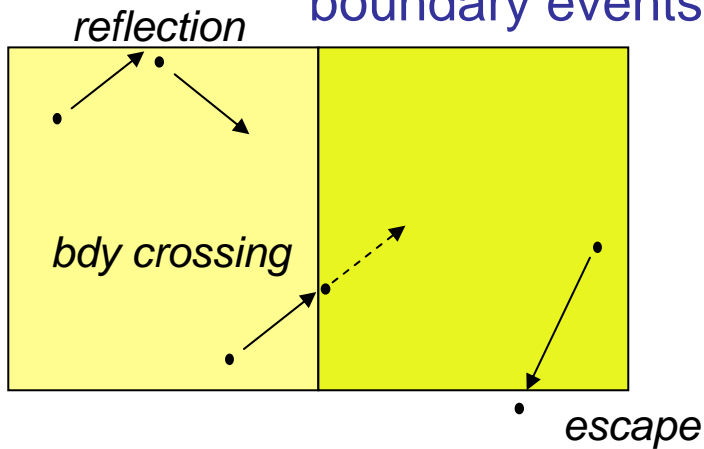
collision events



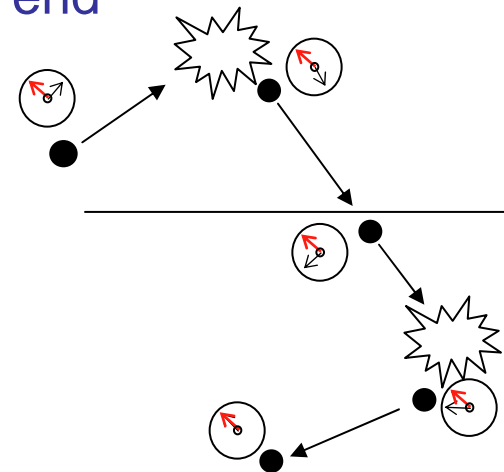
weight cutoff



boundary events



step end

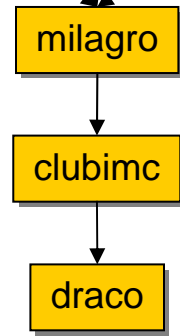
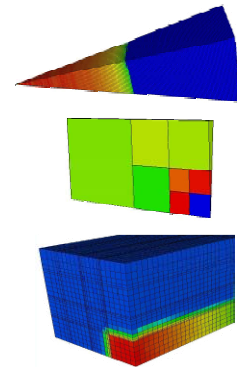


Milagro is a full application to simulate thermal x-ray transport via Implicit Monte Carlo

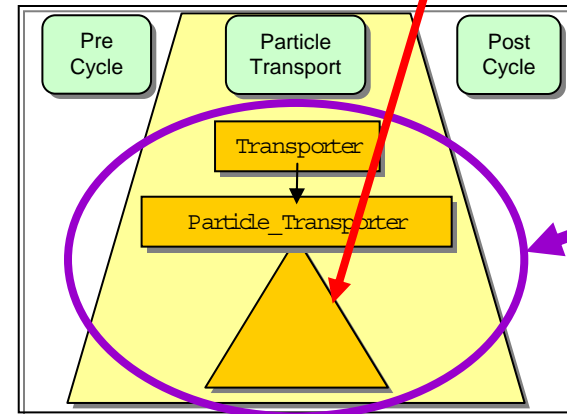
- Multi-dimension, AMR meshes
- Fleck & Cummings time-implicitness
- ~110k lines of object-oriented C++
 - *template parameterization of independent variables (e.g., mesh types)*
- Three layers of libraries.
 - *Changes predominately confined to the middle library layer.*
- Two modes of parallel execution



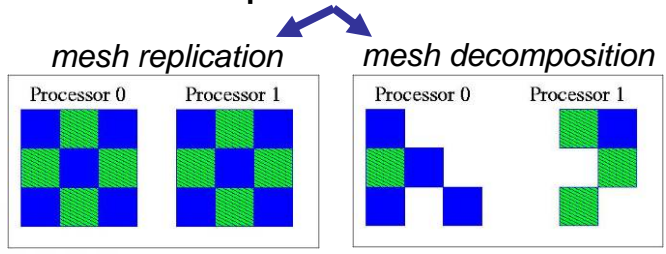
Executables
(by mesh type)



Most changes were here



This part dominates the compute time
~10K lines of C++



Steps in converting Milagro to Roadrunner were orderly

- Modify algorithm to create blocks of particles
- MPI was used to prototype remote offload processing
 - *Familiar parallel computing environment*
 - *Two executables*
- Now test remote code on the PPE
- PPE code is moved to SPE and DMAs are added
 - *get/put/wait asynchronous transfers to/from LS & Cell main memory*
- SPE code is SIMDized
 - *vector types and instructions*
 - *branches converted to conditional merges*

Example of Cell use: distance-to-cell-boundary

- **Original:** nested branches to select face/dimension, then overall closest face.
- Original took ~75 decr ticks (DD2)
- **New:** compute distance to all faces, select closest.
- New takes ~4 decr ticks (DD2)

```

if {omega0 != 0.0}
  if {omega0 > 0.0}    // moving
    towards face2
      min_dist = (x-face2)/omega0
      face = 1
    else
      min_dist = (face1-x)/omega0
      face = 2
    end if

```

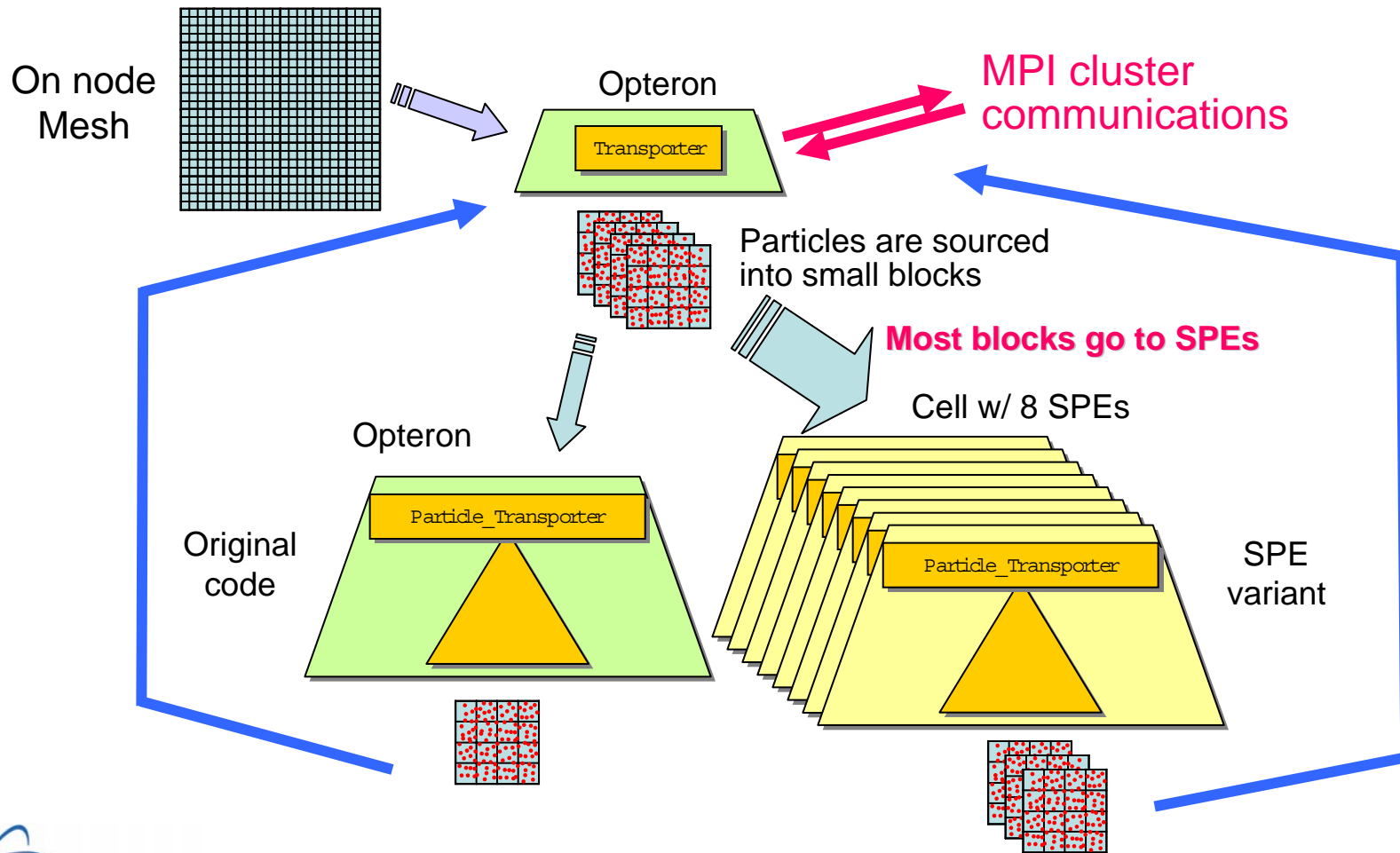
...

```

r0 = spu_splats( x);
o0 = spu_splats( omega0);
L12 = spu_sub( faces12, r0);
// {f1-x, f2-x}
L12 = _divd2( L12, o0);
//{ (f1-x)/omega0, (f2-x)/omega0}
...

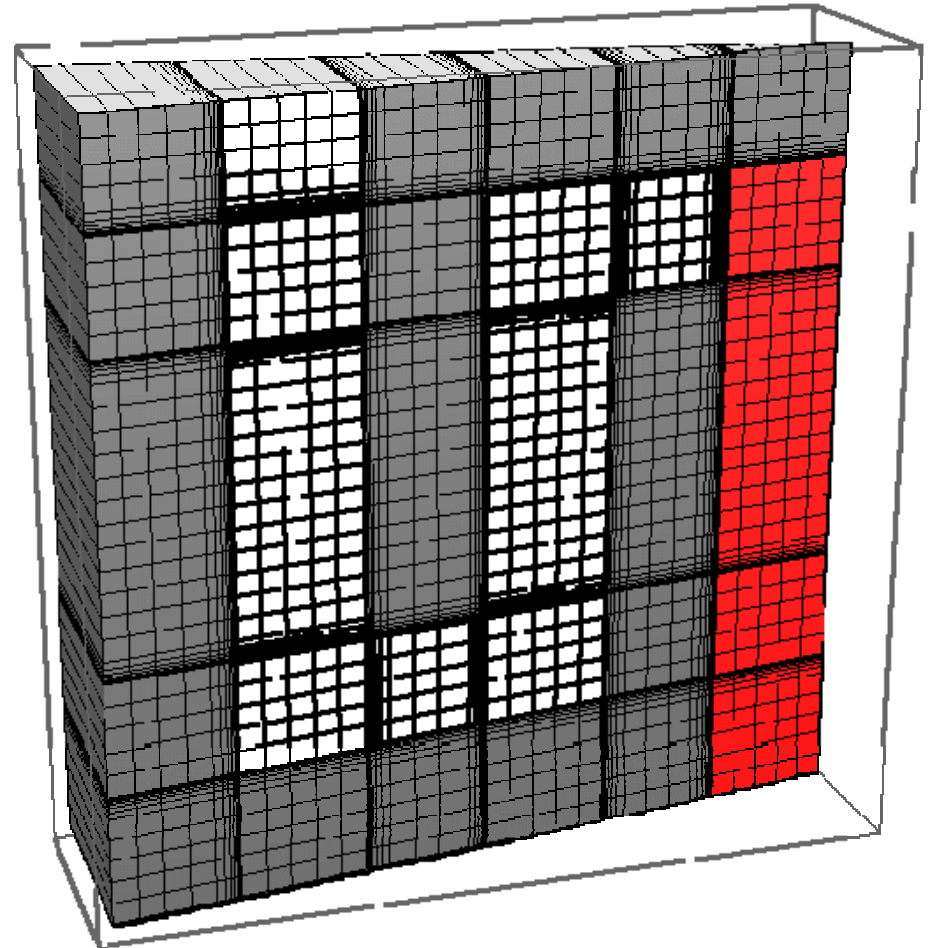
```

Hybrid IMC processing of particle blocks via remote processing on SPEs



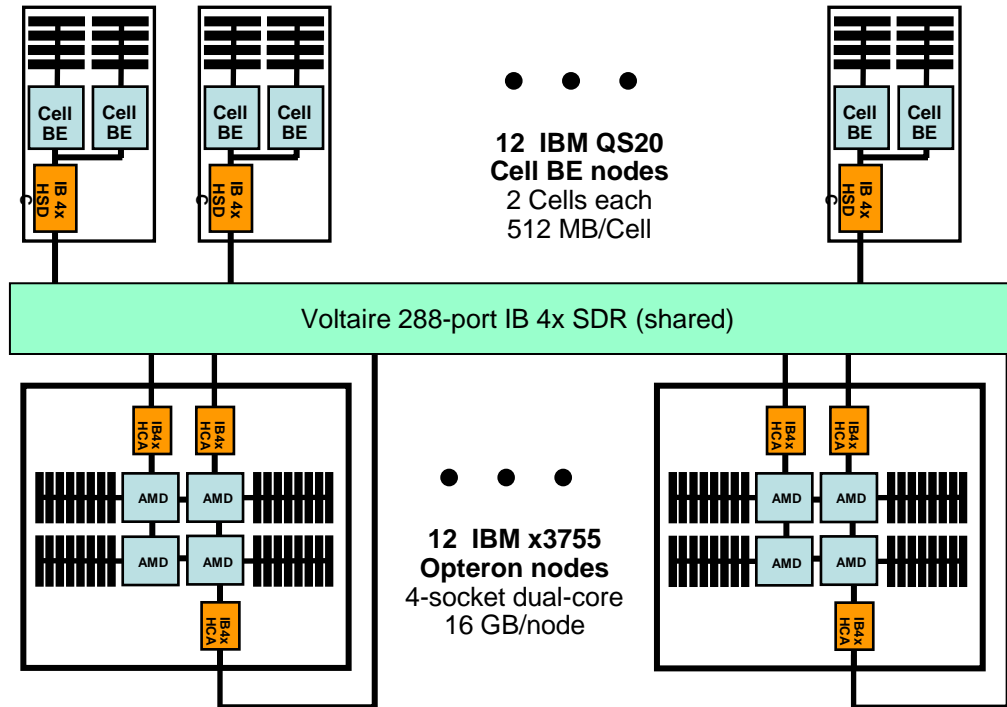
Performance results for all application areas were obtained using realistic / representative problems.

- Example: “double bend” test problem for IMC
 - *RZWedge mesh*
 - regular cells: 0.1 x 0.1 cm
 - ratio-zoned boundary layers
 - 10 degree wedge angle
 - *boundary conditions*
 - outer surface vacuum
 - remainder reflecting
 - *volume radiation source (red)*
 - *5e5 particles/time-step*
 - *optically thick cells (grey)*
 - $\sigma_a = 200/\text{cm}$, 20x20 m.f.p.
 - *optically thin cells (white)*
 - $\sigma_a = 20/\text{cm}$, 2x2 m.f.p.



Prototype hardware was used for applications testing

- InfiniBand-connected current generation Cell-blades



Advanced
Architecture
Initial
System

aka. **AAIS**

(Operational
January 2007)

- limited time on single eDP (IBM PowerXCell 8i) chips since summer

Cell and hybrid speedup results

<i>Application</i>	<i>Type</i>	<i>Class</i>	<i>Cell Only</i> (kernels)		<i>Hybrid</i> (Opteron+Cell)	
			<i>CBE</i>	<i>eDP</i>	<i>CBE+IB</i>	<i>eDP+PCle</i>
<i>SPaSM (10/07)</i>	Science	full app	3x	4.5x	2.5x	>4x
<i>SPaSM (now)</i>			5x	7.5x	4x	>6x
<i>VPIC</i>	Science	full app	9x	9x	6x	>7x
<i>Milagro</i>	IC	full app	5x	6.5x	5x	>6x
<i>Sweep3D</i>	IC	kernel	5x	9x	5x	>5x

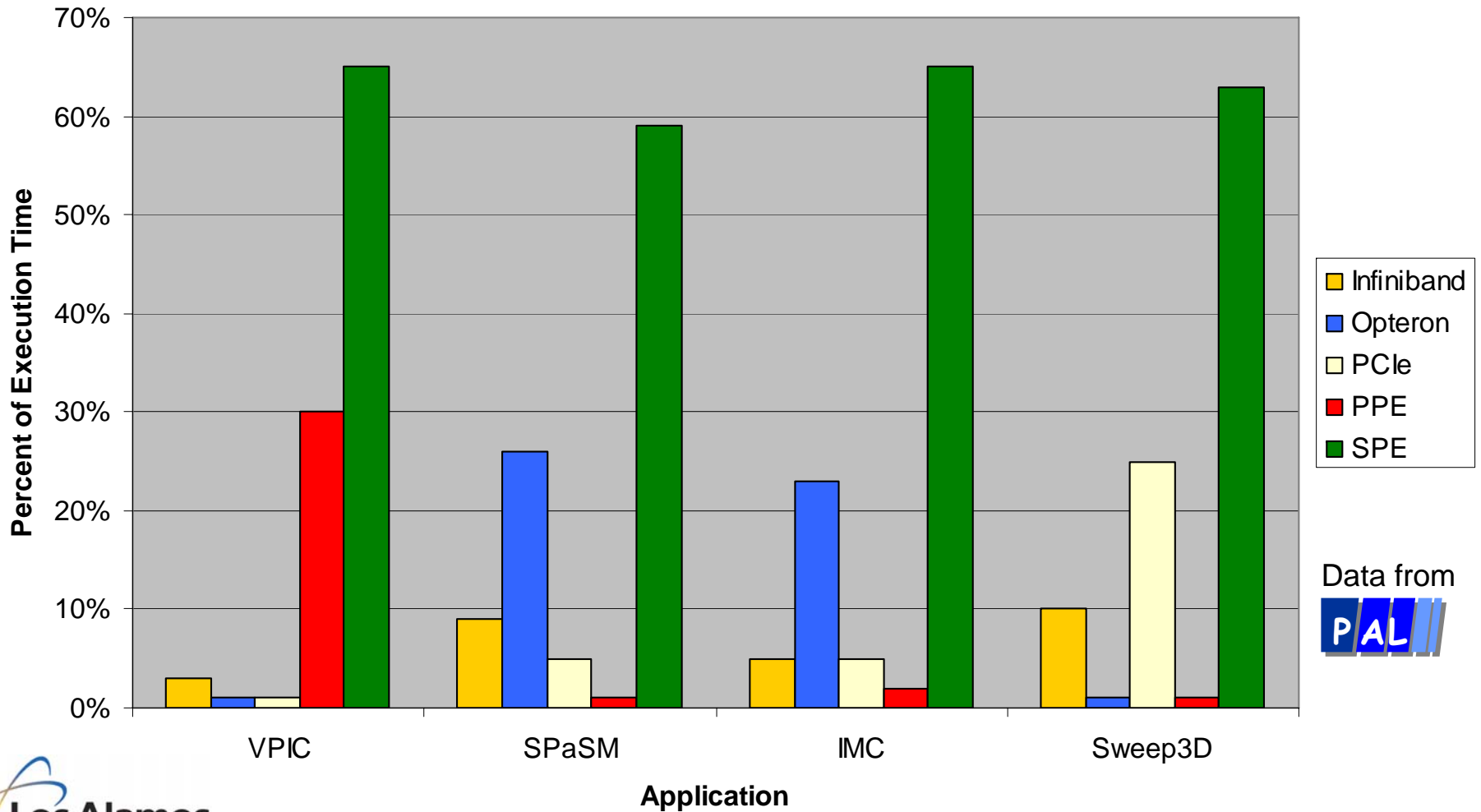
- all comparisons are to a single Opteron core
- parallel behavior unaffected, as will be shown in the scaling results
- first 3 columns are measured, last column is projected

These results were achieved with a relatively modest level of effort.

<i>Code</i>	<i>Class</i>	<i>Language</i>	<i>Lines of code</i>		<i>FY07 FTEs</i>
			<i>Orig.</i>	<i>Modified</i>	
<i>VPIC</i>	full app	C/C++	8.5k	10%	2
<i>SPaSM</i>	full app	C	34k	20%	2
<i>Milagro</i>	full app	C++	110k	30%	2 x 1
<i>Sweep3D</i>	kernel	C	3.5k	50%	2 x 1

- all staff started with little or no knowledge of Cell / hybrid programming
- 2 x 1 denotes separate efforts of roughly 1 FTE each
- most efforts also added code

Roadrunner architecture is flexible - Applications are free to use hardware in most appropriate manner.



Direct Numerical Simulation (DNS) of Turbulence



Operated by the Los Alamos National Security, LLC for the DOE/NNSA



Direct Numerical Simulation

- solution of the Navier-Stokes equations *without any modeled terms*
 - *all scales must be resolved on the computational grid*
 - *turbulence theory depends on separation of scales*
 - *large grids, lots of data storage*
- provides data for closure models
 - *needed for coarse ‘engineering’ simulations*
 - *more detailed information than can be provided by experiments*

Code Overview

- two main elements
 - *derivative calculation*
 - Pade scheme requires tridiagonal solver
 - requires communication
 - *data transpose (serial solver)*
 - *ghost cells, intermediate values (parallel solver)*
 - *update calculation*
 - point-wise, requires no communication
 - easily vectorized for SPU

New hybrid implementation demonstrates remarkable speedup

- serial tests show speedup of 50x compared to the Opteron version
 - *single precision*
- new parallel tridiagonal solver reduces data movement dramatically
 - *latency vs. data volume tradeoff*
- explicit overlapping of compute and communication in final version

Future

Where do we go from here?

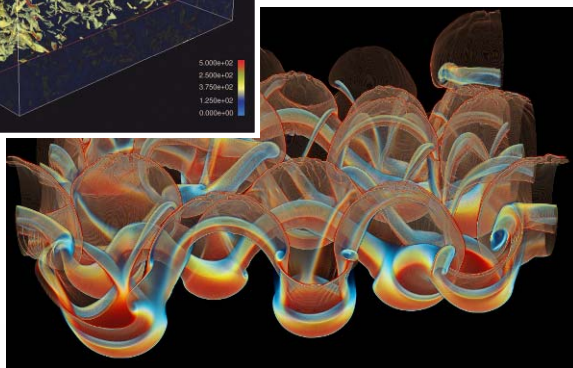
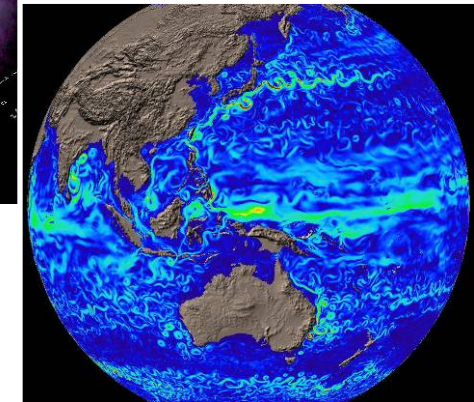
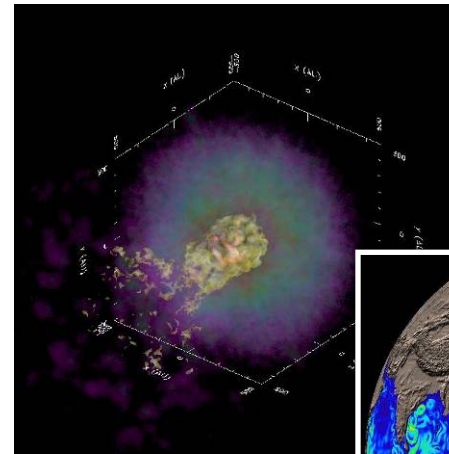
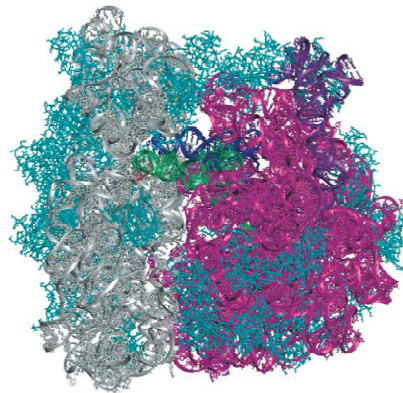
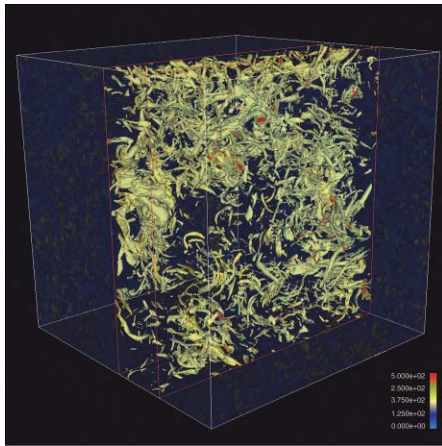
We have demonstrated that the accelerated hybrid model works for applications of interest to NNSA/ASC.

Roadrunner is applicable to the current and future application workload.

1. You can program the Cell processor.
2. You can achieve significant performance gains on the Cell processor.
3. You can program Opteron / Cell hybrid nodes.
4. You can achieve significant performance gains in hybrid.

Efforts will expand into additional areas in FY08 and beyond.

- protein dynamics for drug design
- ocean and climate modeling
- energy research
- astrophysics



- computational fluid dynamics
 - *compressible DNS of turbulence*
 - *existing codes and methods (Rage)*
 - *piecewise-parabolic method (PPM)*
 - *mesh-free / particle methods*

But what now?

- Efforts this year have demonstrated that we can use the machine, but sustainable programming model is unclear.
 - *RR has*
 - ~3200 compute nodes, each with 2 dual-core Opteron
 - ~6400 dual-core Opteron
 - ~13k Opteron cores
 - ~13k Cell procs, each with 8 SPEs
 - ~100k SPEs
 - *how many MPI processes?*

What programming model?

- many to choose from, including...
 - *MPI + MPI or DaCS + threads?*
 - *MPI + MPI or DaCS + ALF?*
 - *MPI + hybrid DaCS?*
 - *MPI + hybrid ALF?*
 - *MPI + threads (OpenMP, TBB, etc.)?*
 - *DARPA/HPCS language (Chapel, Fortress, X10)?*
 - *Partitioned Global Address Space (PGAS) approach (UPC, CoArray Fortran)?*

Are we nearing the end of the “MPI everywhere” era?

- analogous to early days of large-scale parallel
 - *multiple vendor-specific libraries*
 - *PVM*
 - *MPI*
- can't afford to re-write apps multiple times
- community must define...

The Roadrunner Technical Seminar Series

- March 13: "Roadrunner Platform Overview," Ken Koch, CCS-DO
- March 18: "Overview of Applications, Results, and Programming," John Turner, CCS-2
- **March 19: "Overview of Modeling, Performance, and Results," Darren Kerbyson, CCS-1**
- April 10: "Application 1: SPaSM," Sriram Swaminarayan, CCS-2
- April 22: "Application 2: VPIC," Ben Bergen, CCS-2
- April 23: "Application 3: SWEEP3D," Mike Lang, CCS-1
- April 24: "Application 4: Milagro I," Tim Kelley, CCS-2
- May 6: "Application 5: Milagro II," Paul Henning, CCS-2
- May 8: "Application 6: DNS," Jamal Mohd-Yusof, CCS-2
- May 29: "Panel Discussion: Hybrid Computing Programming Models"
- June 3: "Panel Discussion: Future Platforms"

Where to go for more information

- LANL Roadrunner web page and portal
 - <http://www.lanl.gov/roadrunner/>
 - <http://rralgs.lanl.gov/portal>
- Wikipedia entry on Cell processor
 - http://en.wikipedia.org/wiki/Cell_processor
- IBM developerWorks Cell B.E. resource center
 - <http://www-128.ibm.com/developerworks/power/cell/>
- IBM Journal of Research & Development issue devoted to Cell
 - <http://www.research.ibm.com/journal/rd51-5.html>
- IBM developerWorks series on programming the Cell
 - <http://www.ibm.com/developerworks/power/library/pa-linuxps3-1>
 - <http://www.ibm.com/developerworks/power/library/pa-linuxps3-2>
 - <http://www.ibm.com/developerworks/power/library/pa-linuxps3-3>

Where to go for more information (cont.)

- Power.org Cell Developer Corner (links to tons of info)
 - <http://www.power.org/resources/devcorner/cellcorner/>
- Maximizing the power of the Cell Broadband Engine processor: 25 tips to optimal application performance
 - <http://www.ibm.com/developerworks/library/pa-celltips1/>
- Sony Computer Entertainment US Research and Development
 - <http://www.research.scea.com/>
- MIT course on programming the Playstation 3
 - <http://cag.csail.mit.edu/ps3/index.shtml>
- CellPerformance
 - <http://www.cellperformance.com/>
- Beyond3D.com Cell Forum
 - <http://forum.beyond3d.com/forumdisplay.php?f=57>
 - *list of Cell resources*
 - <http://forum.beyond3d.com/showthread.php?t=42626>

Roadrunner at a glance

- **Cluster of 18 Connected Units**
 - 6480 (+432) AMD dual-core Opterons
 - 12,960 IBM Cell eDP accelerators
 - 46.7 (+4.5) Teraflops peak (Opteron)
 - 1.33 Petaflops peak (Cell eDP)
 - 1PF sustained Linpack
- **InfiniBand 4x DDR fabric**
 - 2-stage fat-tree; all-optical cables
 - Full bi-section BW within each CU
 - 384 GB/s (bi-directional)
 - Half bi-section BW among CUs
 - 3.45 TB/s (bi-directional)
 - Non-disruptive expansion to 24 CUs
- **80 TB aggregate memory**
 - 52 TB Opteron
 - 52 TB Cell
- **216 GB/s sustained File System I/O:**
 - 216x2 10G Ethernets to Panasas
- **Fedora Linux (RHEL possible)**
- **SDK for Multicore Acceleration**
 - Cell compilers, libraries, tools
- **xCAT Cluster Management**
 - System-wide GigE network
- **3.9 MW Power:**
 - 0.35 GF/Watt
- **Area:**
 - 296 racks
 - 5500 ft²



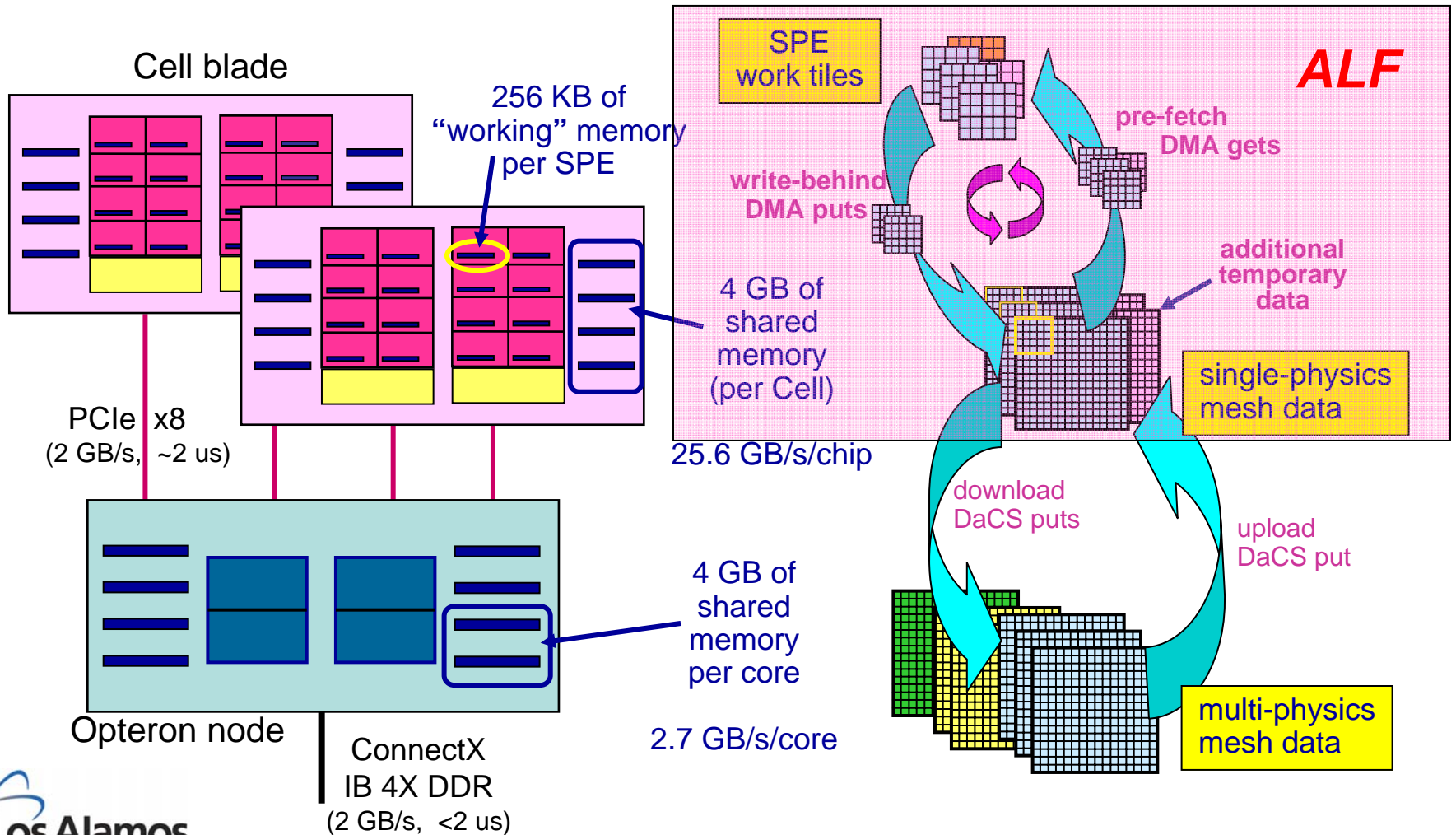
Backup Slides



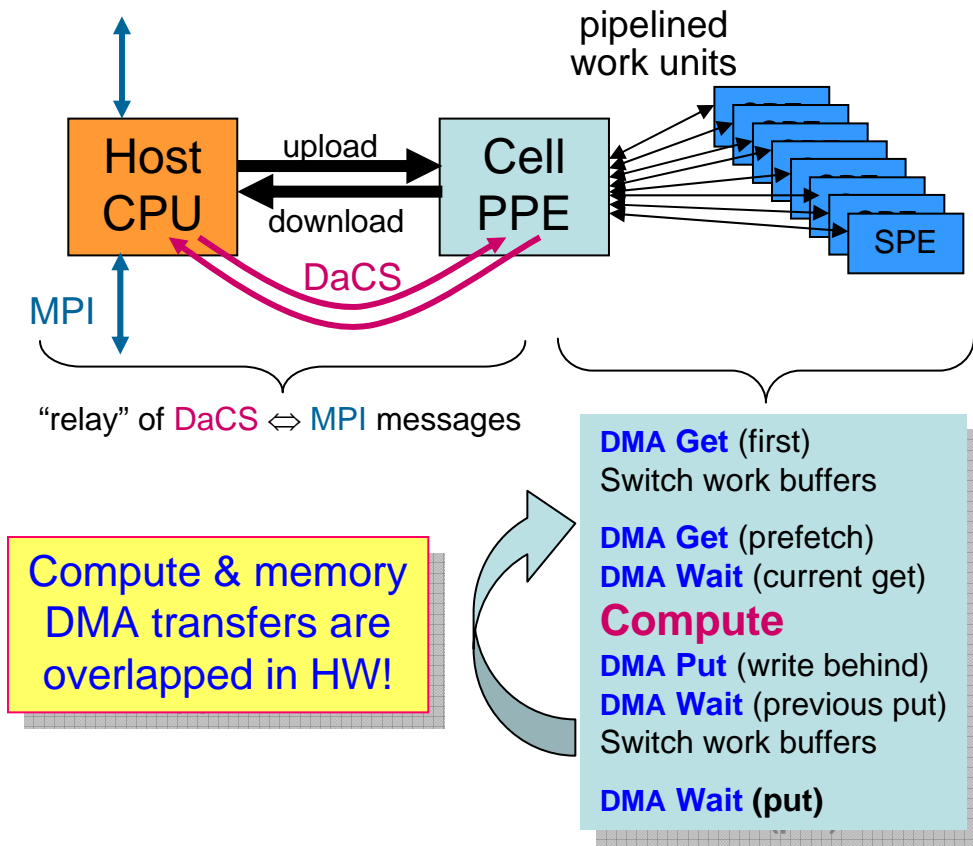
Operated by the Los Alamos National Security, LLC for the DOE/NNSA



Using Roadrunner's memory hierarchy: Today with Hybrid DaCS and Supplemented Tomorrow with ALF



Put it all together



- DMA transfers are simply block memory transfers
 - *HW asynchronous (no SPE stalls)*
 - *DDR2 memory latency and BW performance*

```

DMA Get:
mfc_get( LS_addr, Mem_addr, size, tag, 0, 0);

DMA Put:
mfc_put( Mem_addr, LS_addr, size, tag, 0, 0);

DMA Wait:
mfc_write_tag_mask(1<<tag);
mfc_read_tag_status_all();
    
```