

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/267419442>

Roadrunner: Heterogeneous Petascale Computing for Predictive Simulation

Article · February 2007

CITATIONS

4

READS

47

1 author:



John Alexander Turner

Oak Ridge National Laboratory

44 PUBLICATIONS 314 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Consortium for Advanced Simulation of Light-Water Reactors (CASL) [View project](#)



Solidification Microstructure Control in Electron Beam Additive Manufacturing [View project](#)



Roadrunner: Heterogeneous Petascale Computing for Predictive Simulation

(Los Alamos Unclassified Report LA-UR-07-1037)

Advanced Simulation & Computing (ASC)

2007 Principal Investigator's Meeting

Las Vegas, NV (2/20-22/2007)

John A. Turner

Group Leader, Computational Physics and Methods Group ([CCS-2](#))

Computer, Computational, and Statistical Sciences Division ([CCS](#))

turner@lanl.gov

The Computing Landscape is Changing (again)



highly multi-core: 4, 8, 16, 32, ...

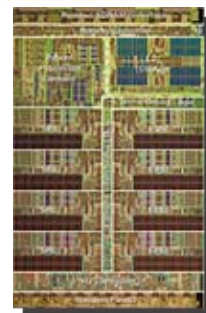
- CPUs, [GPUs](#), [Cell](#)
- [distributed memory](#) at core level

co-processors

- [GPUs](#), [Cell](#), [ClearSpeed](#), [FPGAs](#)

heterogeneous architectures

- within processor itself (e.g. [Cell](#))
- at the board level (e.g. [AMD's Torrenza](#))
- on the same bus (e.g. CPU+[GPUs](#), Intel's [Geneseo](#))
- within a cluster (e.g. [Roadrunner](#))



Cell

+



CPU

+



GPU

= ?

Jan. 2007 IDC Study: “Next Phase in HPC”



What Will Be The Next Phase in HPC, and Will It Require New Ways of Looking at HPC Systems?

- key findings:
 - *individual processor core speeds relatively flat*
 - *bandwidth per socket will grow slowly, but cores per socket will increase at to Moore's law (doubling every 18-24 mo.)*
 - “inverse Moore's law” for bandwidth per core
 - *new ways of dealing with parallelism will be required*
 - *must focus more heavily on bandwidth (flow of data) and less on processor*

What's driving the move to multi-core?

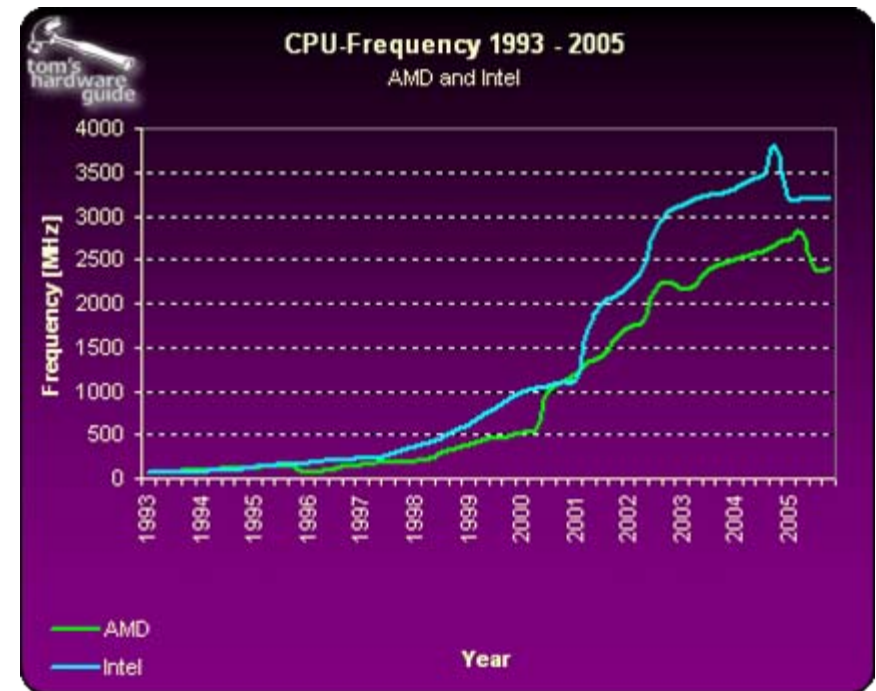


CPU speeds are stagnating

- diminishing returns from deeper pipelines
- multi-core increases spatial efficiency, constrains processor complexity

power considerations

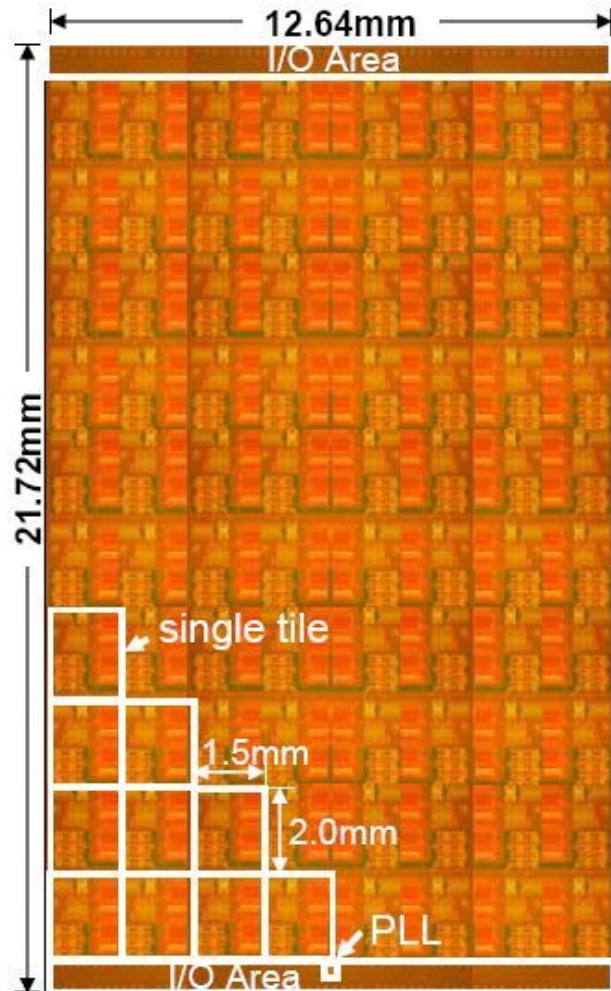
- multi-core yields improved performance per watt



"doubling happens... until it doesn't"



Intel 80-core Prototype

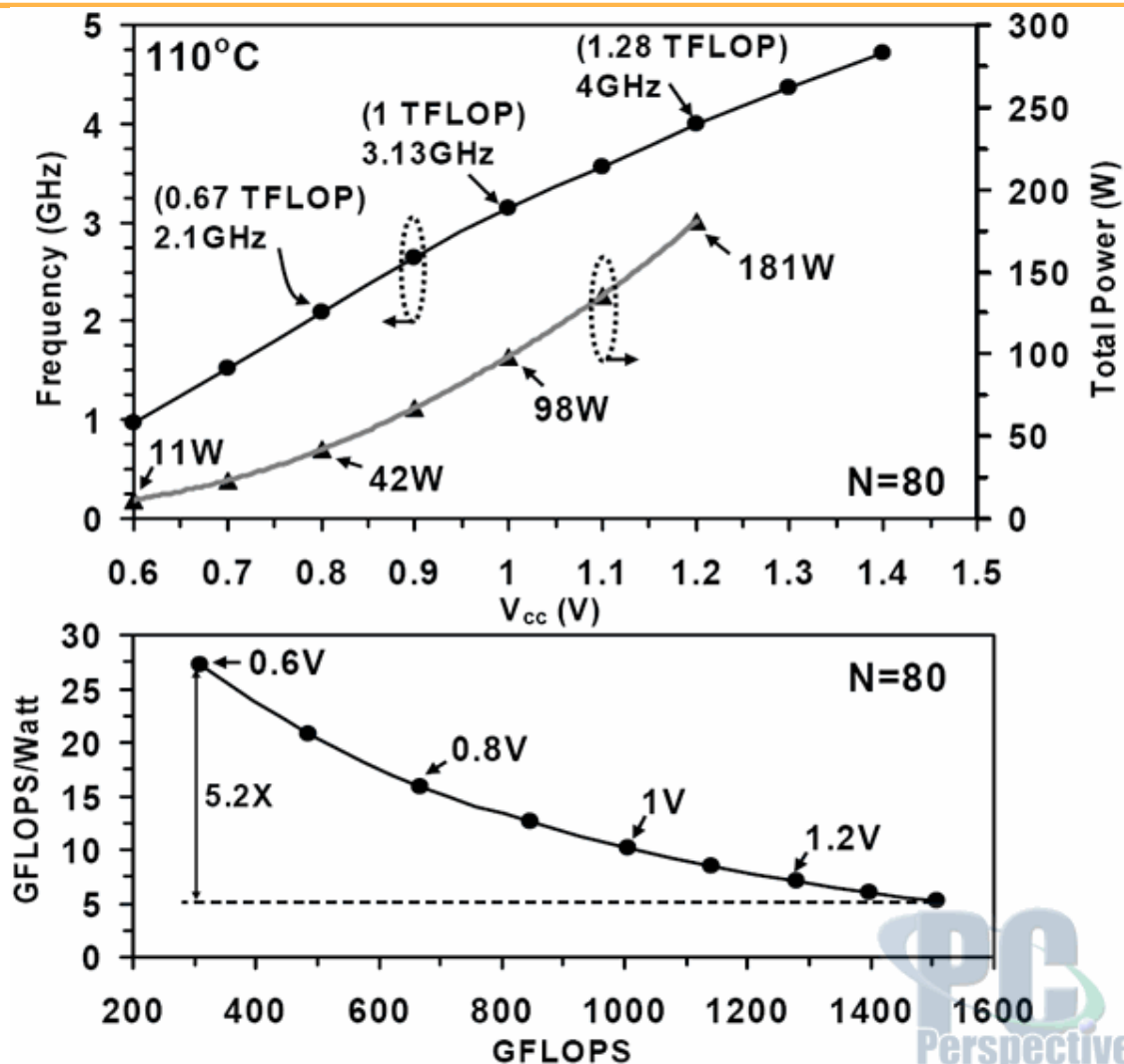


shown at *Fall 2006 Intel Dev. Forum*

- more details at [PC Perspective](#)
- Polaris, 1.8 TF/s aggregate performance
 - *10x8 2D mesh, 4 GHz*
- additional level of memory hierarchy
 - *each core has small local store*
- non-uniform off-chip access
 - *only edge cores communicate*

| | |
|--------------|--|
| Technology | 65nm CMOS Process |
| Interconnect | 1 poly, 8 metal (Cu) |
| Transistors | 100 Million |
| Die Area | 275mm ² |
| Tile area | 3mm ² |
| Package | 1248 pin LGA, 14 layers, 343 signal pins |

Performance per watt improvement...





Multi-core Challenges

exacerbates the imbalance between processing and memory access speeds

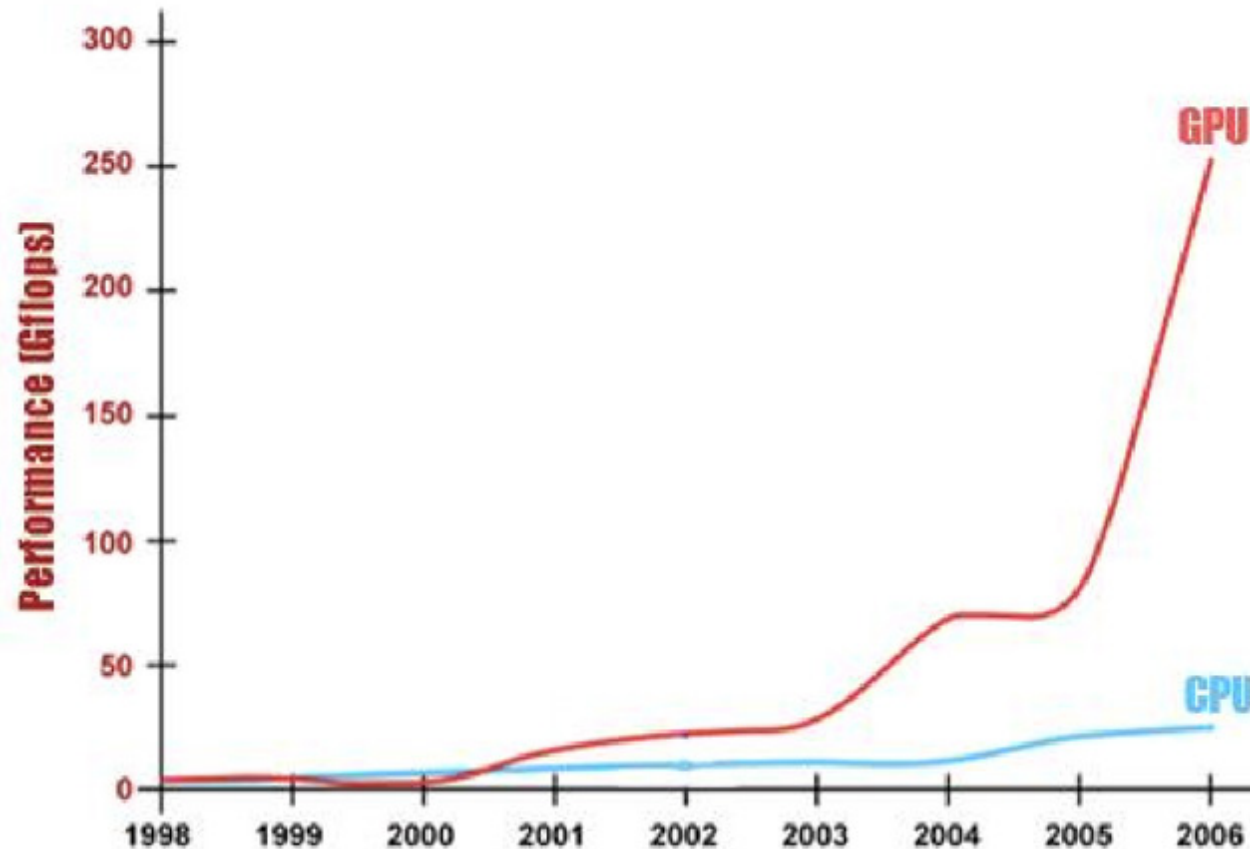
- not like large SMPs
- all systems start to look like attached processors
 - *high latency, low relative bandwidth to main memory*



must identify much more parallelism in apps

- not just thousands of processes - now thousands of threads within nodes
 - *the era of “relentless multithreading” is upon us*

Video Cards (GPUs) as Compute Engines





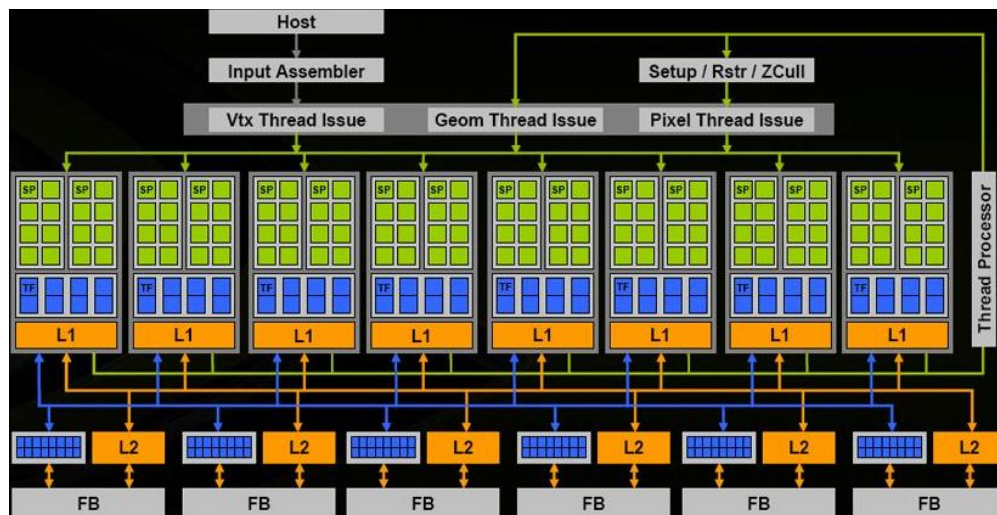
GPGPU Developments

NVIDIA G80 architecture

- 681 million transistors, 128 stream processors
 - *Intel Core 2 Extreme Quad Core (Kentsfield) has ~582 million*
 - *supports "thousands" of threads in flight*
- more support for general programming (branching, etc.)
- simultaneously announced CUDA SDK
 - *treat GPU as pure compute engine - no graphics layer*

"GPU"s with no video out

- pure compute engine



Roadrunner is a Critical Supercomputer Asset



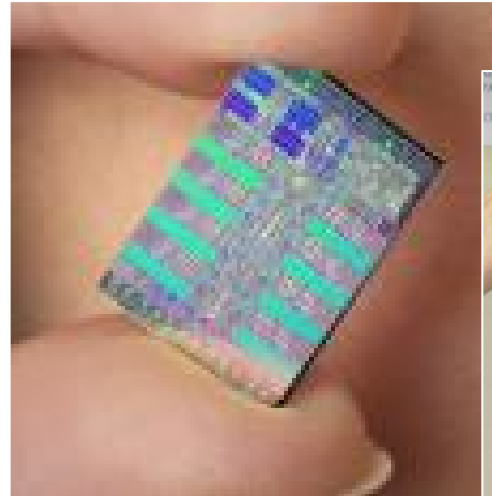
Contract Awarded to **IBM**
September 8, 2006

*Critical component of
stockpile stewardship*

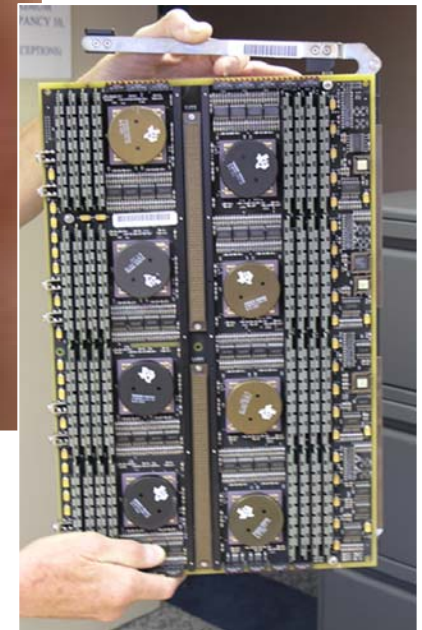
- Initial system supports near-term mission deliverables
- Hybrid final system achieves PetaFlops level of performance

*Accelerated vision of the
future*

- Faster computation, not more processors



Cell processor
(2007, ~100 GF)



CM-5 board (1994, 1 GF)



Roadrunner Goals

Provide a large “capacity-mode” computing resource for LANL weapons simulations

- 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

- Capable of supporting future LANL weapons physics and system design workloads
- Capable of achieving a sustained PetaFlop



Roadrunner Phases

Phase 1 (Now)

- 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 (Summer '07)

- Improved Cell Blades & Cell software on 6 more nodes of AAIS
- Supports pre-Phase 3 assessment

Phase 3 (FY08)

- Populate entire classified system with Cell Blades
- Achieve a sustained 1 PetaFlop Linpack
- *Contract Option*

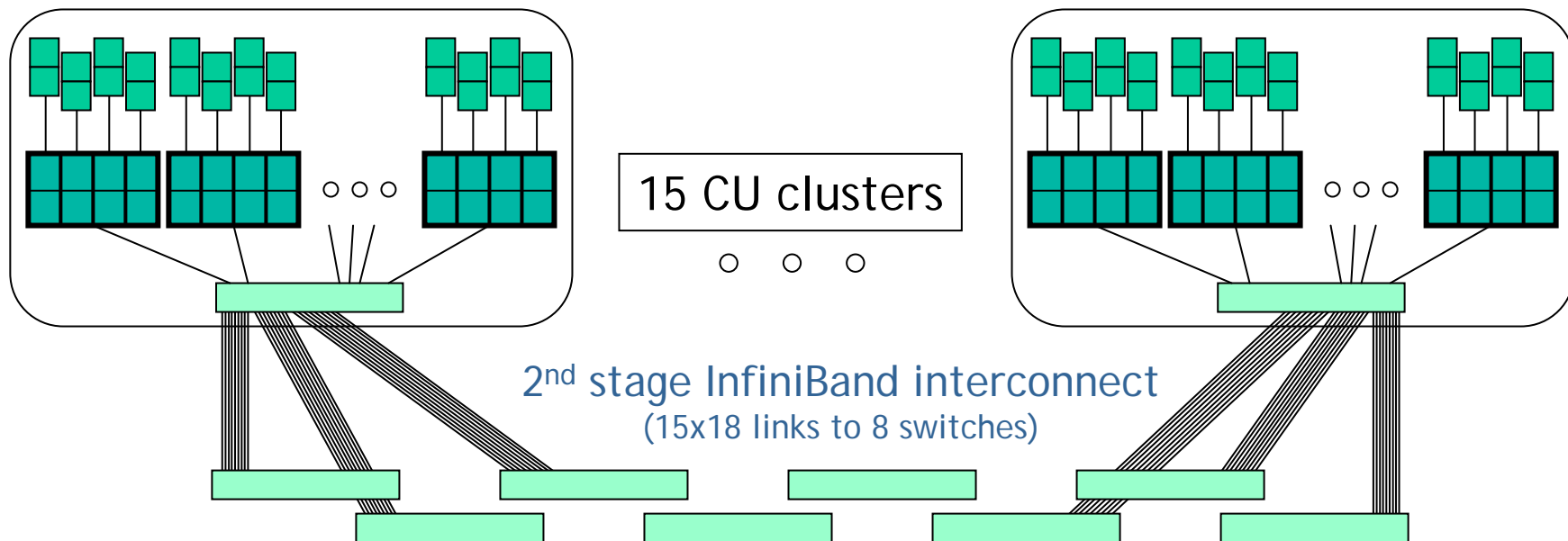
Roadrunner Final System:

8,640 dual-core Oploterons + 16,560 Cells



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

1 Oploteron core \Leftrightarrow 1 Cell processor



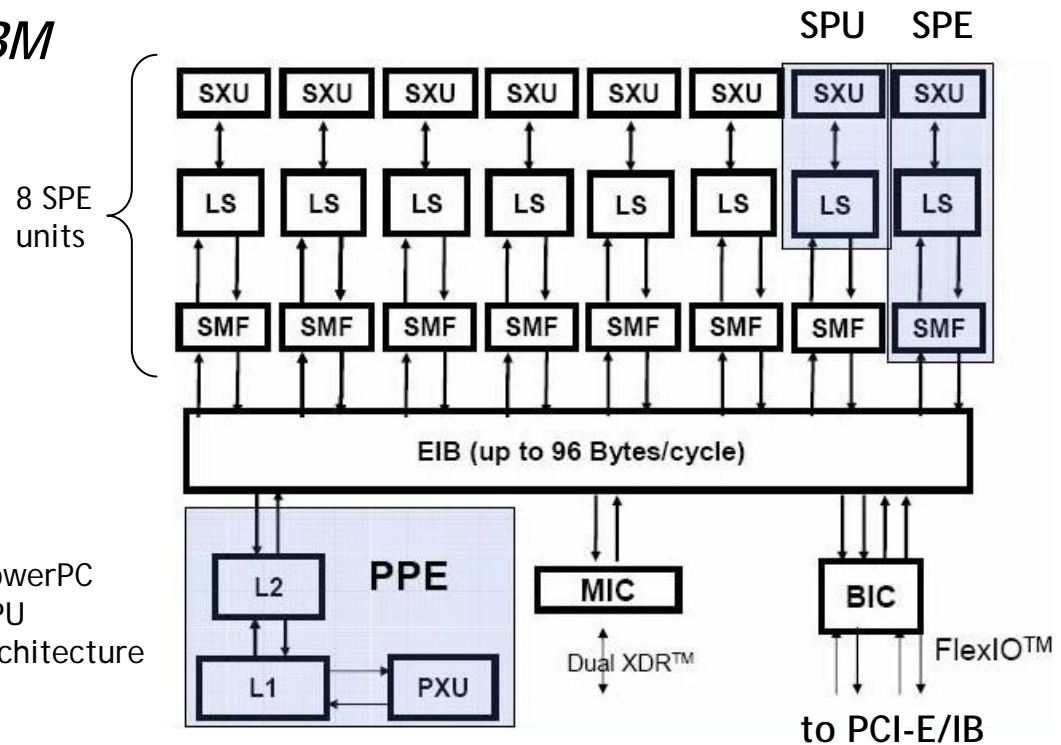
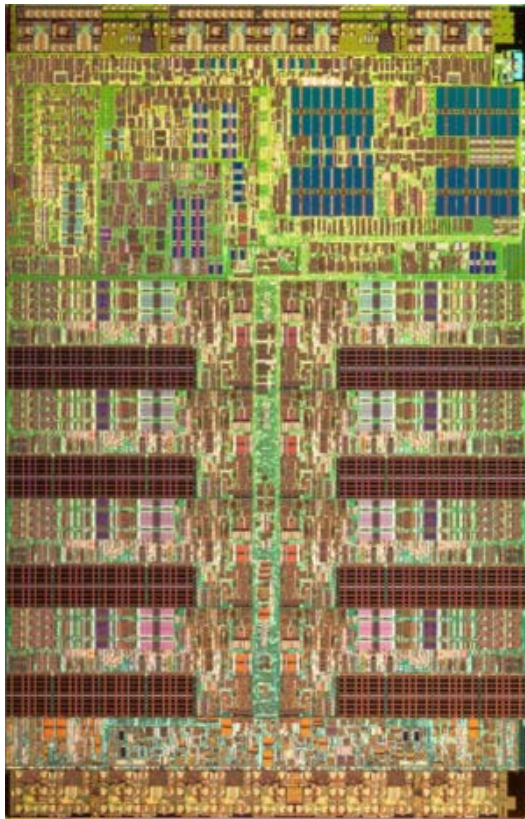
76 TeraFlop/s Oploterons + ~1.7 PetaFlop/s Cell

Cell Broadband Engine (CBE): an 8-way heterogeneous parallel processor



developed by Sony-Toshiba-IBM

- used in [Sony PlayStation 3](#)

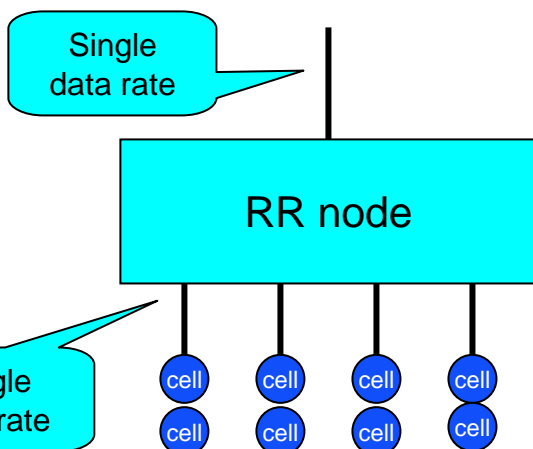


8 Synergistic Processing Elements (SPEs)

- 128-byte vector engines
- 256 kB local memory w/DMA engine
- operate together (SPMD) or independently (MPMD)
- currently 200 GF/s single-precision, 15 GF/s DP

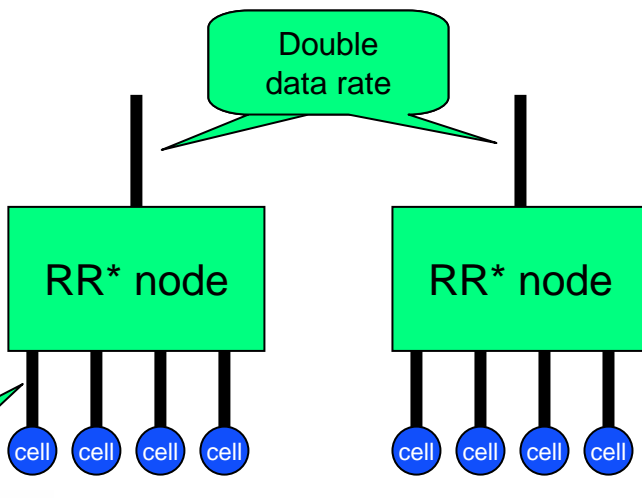


Improved Roadrunner



Keep current base system

- 70 TF capacity resource in secure
- fully available for stockpile stewardship
- no restabilization after cells arrive



Next generation PetaFlop system on same schedule

- based on existing technology
- better performance
- PetaFlop run 2 months early
- possible "science runs" in open



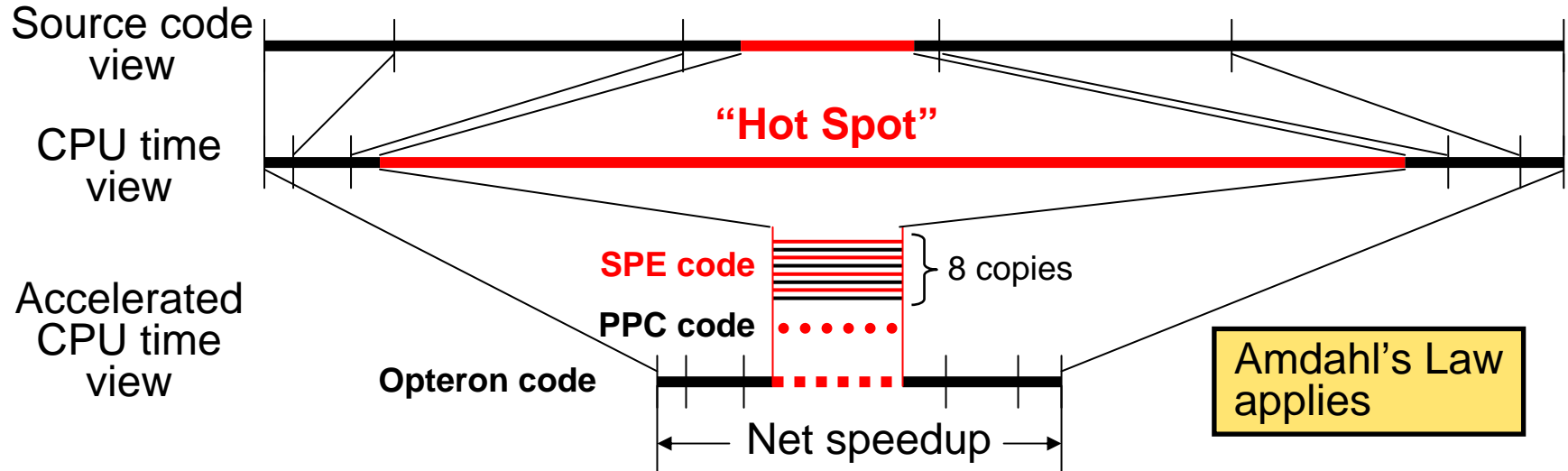
Hybrid Programming

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 Local Store (LS), transferring data blocks in/out as necessary*
 - *Performs vector computations from its LS data*

Each code is compiled separately (currently)

Identify Computationally-Intensive Physics for Cell



- no compiler switches to “just use the Cells”
 - *not even a single compiler - 3 of them*
- currently, code developer must decompose application and create cooperative program pieces
- tools are an issue



Hybrid Programming Env. Under Development With IBM

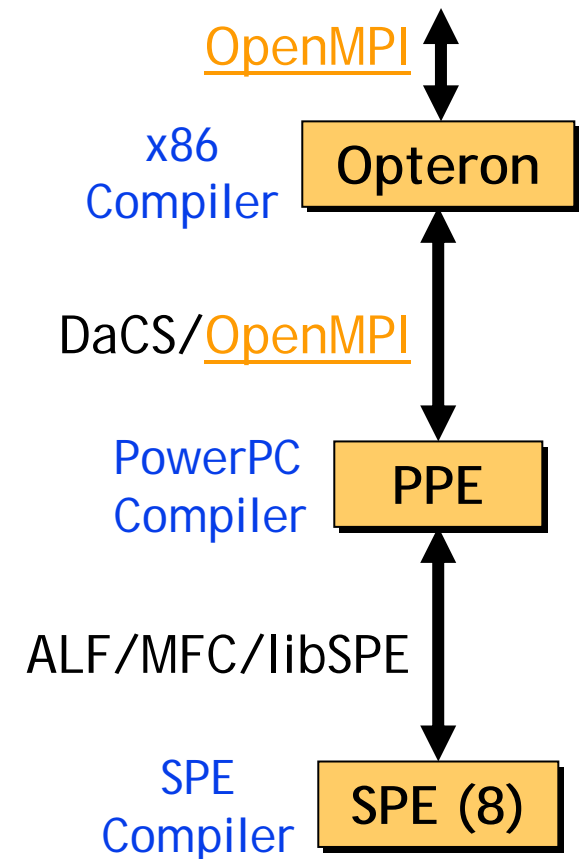


Computational Library

- Data partitioning
- Task & work queue management
- Process management
- Error handling

Communication Library

- Data movement & synchronization
- Process management & synchronization
- Topology description
- Error handling
- First implementation may be OpenMPI



Advanced Hybrid Eco-System



higher level tools

- Cell compilers for data parallel, streaming, work blocks, etc. undergoing rapid development
 - *Scout* (LANL), *Sequoia* (Stanford), *RapidMind* (commercial), *PeakStream* (commercial)
 - *game-oriented Cell development tools*
- more expected in the future



The bright side...

“Big Iron” sometimes inspires algorithmic advances

- hard for computational physicists to admit...
 - *Krylov methods as iterative solvers enabled by vector in 70s*
- we'd like to think it was a “push” instead of a “pull”, but
 - *computational physicists often don't think “outside of the box” without the lure of a new, bigger, shinier box*

Petascale systems will serve as catalyst for next leap(s) forward

will be as painful as previous architectural shifts

- *vector, massively-parallel, cache-based clusters, etc.*

Heterogeneous Manycore Architectures Are Here



we have been pursuing heterogeneous computing for several years

- results thus far ([GPU](#), [FPGA](#), [Cell](#)) are encouraging
- [Roadrunner](#) is simply the first large-scale example

focus on applications of interest

- 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”

Dealing with the Processor / Bandwidth Imbalance



“better, not just faster”

- high-order methods
 - *more computation per word of memory moved*
 - *more accurate answer in less elapsed time*
- more rigorous multiscale treatments
 - *e.g. simultaneous subgrid simulations*
- integrated uncertainty quantification / sensitivity analysis
- ensemble calculations
 - *compute set of values / cells / particles*
- rather than compute properties a priori, store in tables, and interpolate, compute on-the-fly
- coupled physics: rigorous nonlinear consistency
- different problem decompositions

*long-term, must (re-)design algorithms for memory
locality and latency tolerance*

Radiative Heat Transfer on GPUs



original approach - project onto hemisphere

- hemispheric projection inefficient
- straight lines map to curves
- req. intricate tessellation

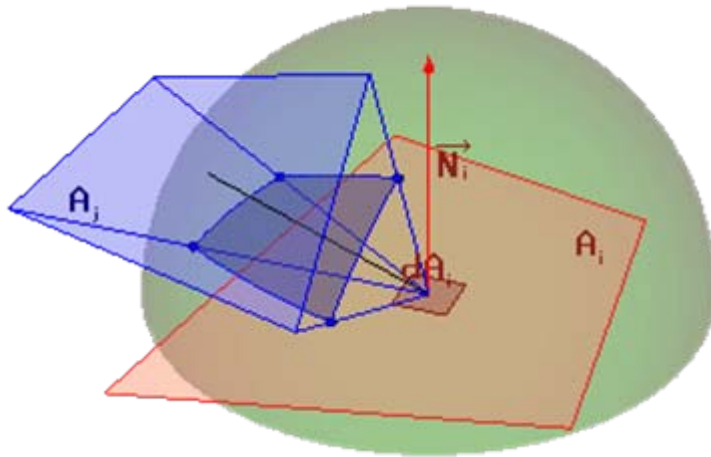


Image credit: <http://raphaello.univ-fcomte.fr/IG/Radiosite/Radiosite.htm>

current "standard" algorithm is hemi-cube

- developed in 1985 for graphics (radiosity)
- project onto faces of tessellated cube

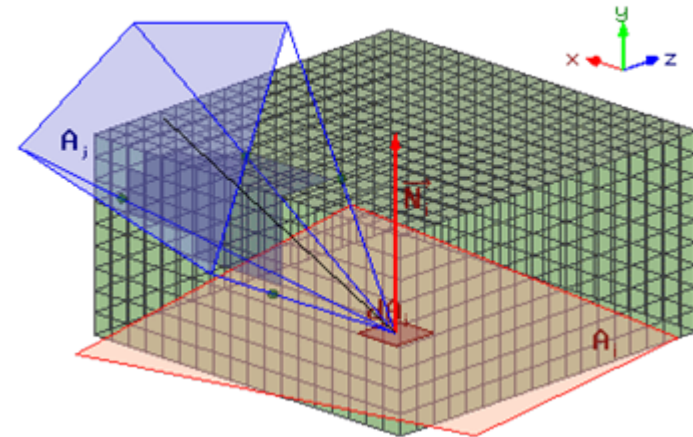


Image credit: <http://raphaello.univ-fcomte.fr/IG/Radiosite/Radiosite.htm> &

GPUs are hardware-accelerated for 3D projections

- insight led to improved algorithm
- one projection rather than five, built-in adaptivity

Radiative Heat Transfer on GPU



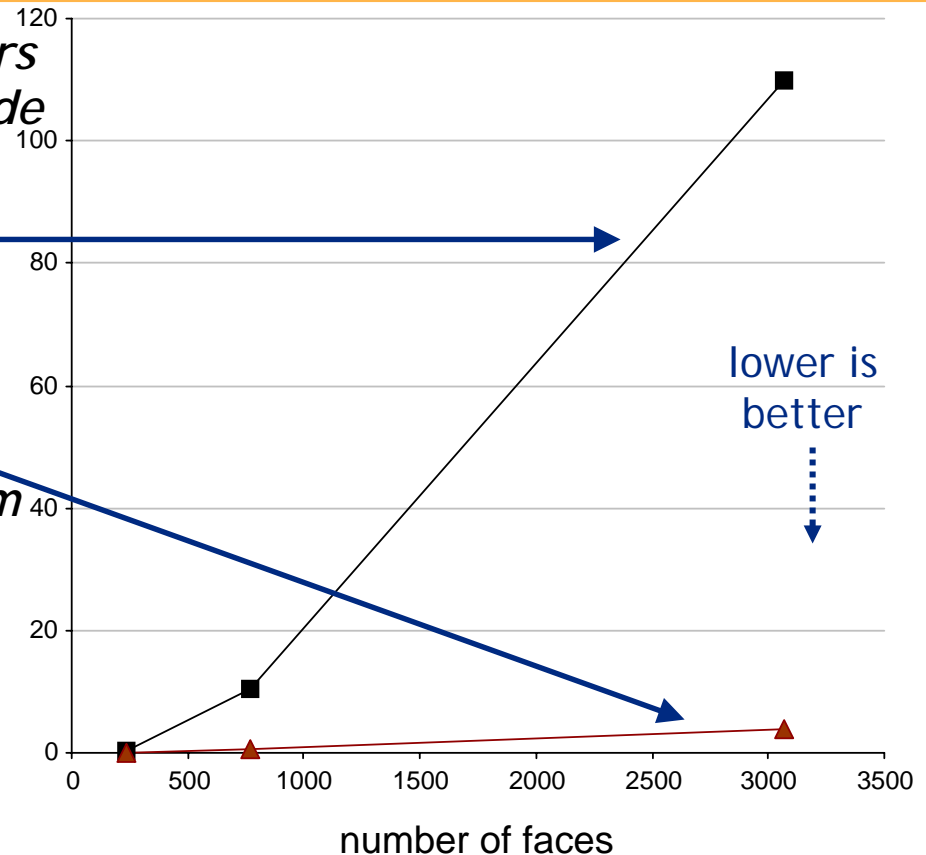
*time (seconds) to compute viewfactors
within Truchas casting simulation code*

- hemi-cube
 - 3.4 GHz 64-bit Xeon
 - Chaparral (from Sandia Nat. Lab.)
- plane projection
 - NVIDIA Quadro FX 1400 GPU

GPU implementation of new algorithm
30x faster

- including data transfer
- can now consider re-computing
viewfactors during fill!

parallel execution on cluster



bandwidth/latency limitations can be overcome

- identify computationally-intensive chunks
- match algorithms to hardware



Roadrunner Advanced Algorithms & Assessment Team



continuation of “swat team” effort initiated in Spring 2006

- gain early experience with Cell
- focus on apps of interest to ASC
- inform Roadrunner architecture decisions

two primary goals for FY07

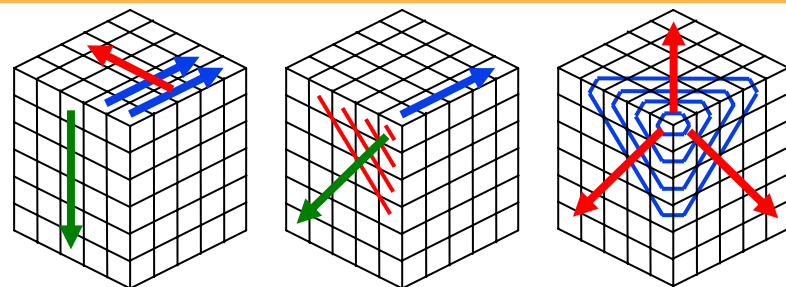
- develop predictive models for LINPACK performance on final system
 - *follow-on to performance modeling efforts for Q, etc.*
 - *track IBM’s LINPACK implementation*
- develop advanced Cell/hybrid algorithms
 - *assess potential performance of applications on final system*
 - *prepare for accelerated science apps in FY08, and later for multi-physics applications*

Initial Cell Results are Encouraging



Transport

- neutron transport via S_n ([PARTISN](#))
 - [Sweep3D](#) - 5x speedup on [Cell](#)
 - sparse linear solver (PCG)
- radiation transport via implicit [Monte Carlo](#) ([MILAGRO](#))
 - 10x speedup for opacity calculation on [Cell](#)

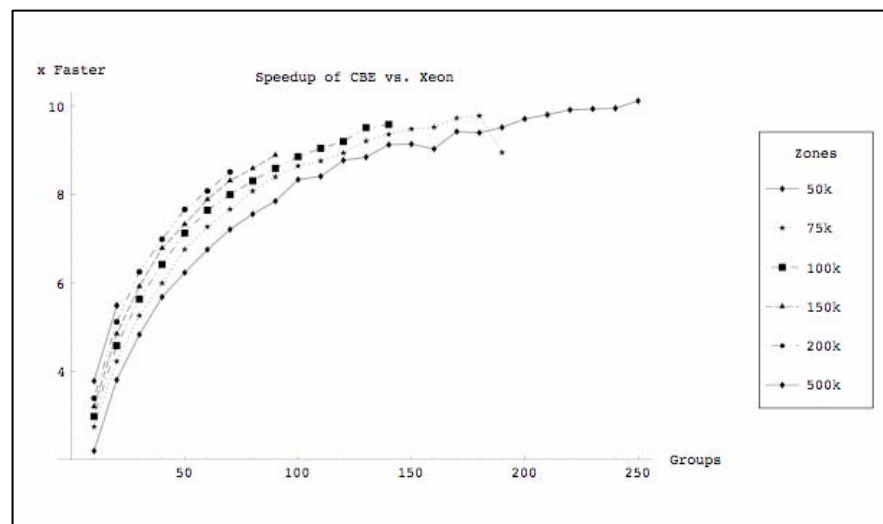


Particle methods

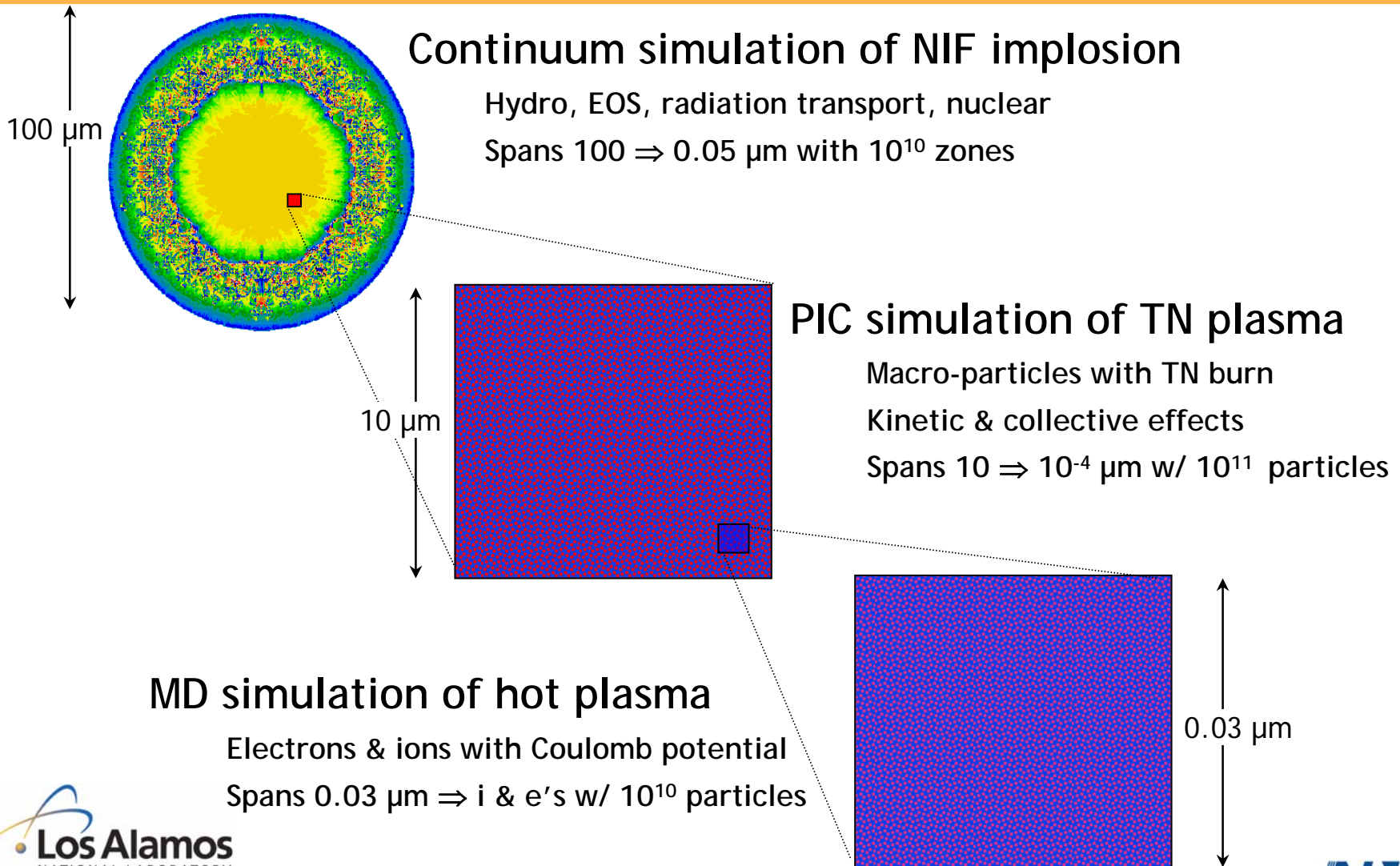
- [Molecular Dynamics](#) (e.g. SPaSM)
 - 7x speedup on [Cell](#)
- [Particle-in-cell](#) (plasma)

Fluid dynamics

- compressible Eulerian hydro
- compressible DNS of turbulence
- advanced methods
 - mesh-free / particle methods



Multi-scale validation of NIF implosion





Roadrunner Represents the Future of Computing



View initial RR design as simply “rev. 0” of large-scale many-core hybrid computing.

- rev. 1 – processors on boards in PCI-E slots
- rev. 2 – processors directly on motherboards
- rev. 3 – different processors on die
- ...

Develop algorithms and software tools for many-core heterogeneous computing – not just RR.