IBM HPC DIRECTIONS

Dr Don Grice

ECMWF Workshop November, 2008

Agenda

- What Technology Trends Mean to Applications
- Critical Issues for getting beyond a PF
- Overview of the Roadrunner Project
- Overview of IBM HPC Roadmap
- Application Structure Research Questions

Introduction

Petascale Computing is enabling new applications that would have been unimaginable just a few short years ago.

It is likely to do to science and business what the Internet did for information retrieval.

Technologically - What's Changing?
The Rate of Frequency Improvement is Slowing!

- Moore's Law (Frequency improvement) is a simplistic subset of Scaling
  - Circuit Density will continue to increase

BUT…

- Rate of Frequency Improvement is slowing
  - Leakage/Standby Power is increasing

- How do we adjust to the change in the rate of Frequency Improvement?
  - How do we deal with the Power and Power Density issues?

Technology Trends and Challenges

There are 3 major challenges on the road to and beyond Petascale computing:

- **Supplying and paying for the power required to run the machines which can be equivalent to a small town!**
- **Reliability, Availability, and Serviceability because of the law of large numbers and Soft Errors**
- **Finding efficient ways to program machines that will support MILLIONS of simultaneous processes!**

Technology Trends and Challenges

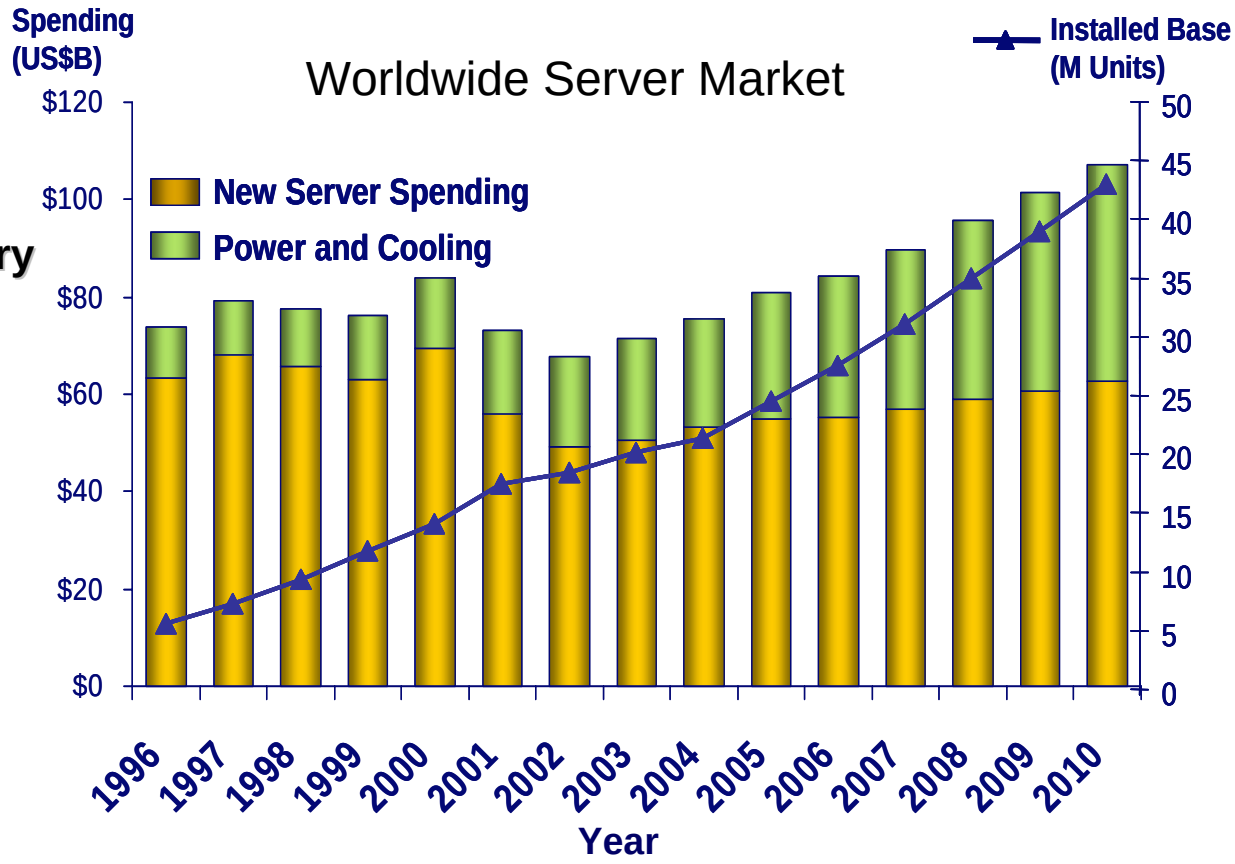
Utility Power and Cooling

The Green Catalyst; Worldwide IT spending trend

Power and cooling spending will exceed new server spending

2000 Raw processing "horsepower" is the primary goal, while the infrastructure to support it is assumed ready

2006 – Raw processing "horsepower" is a given, but the infrastructure to support deployment is a limiting factor



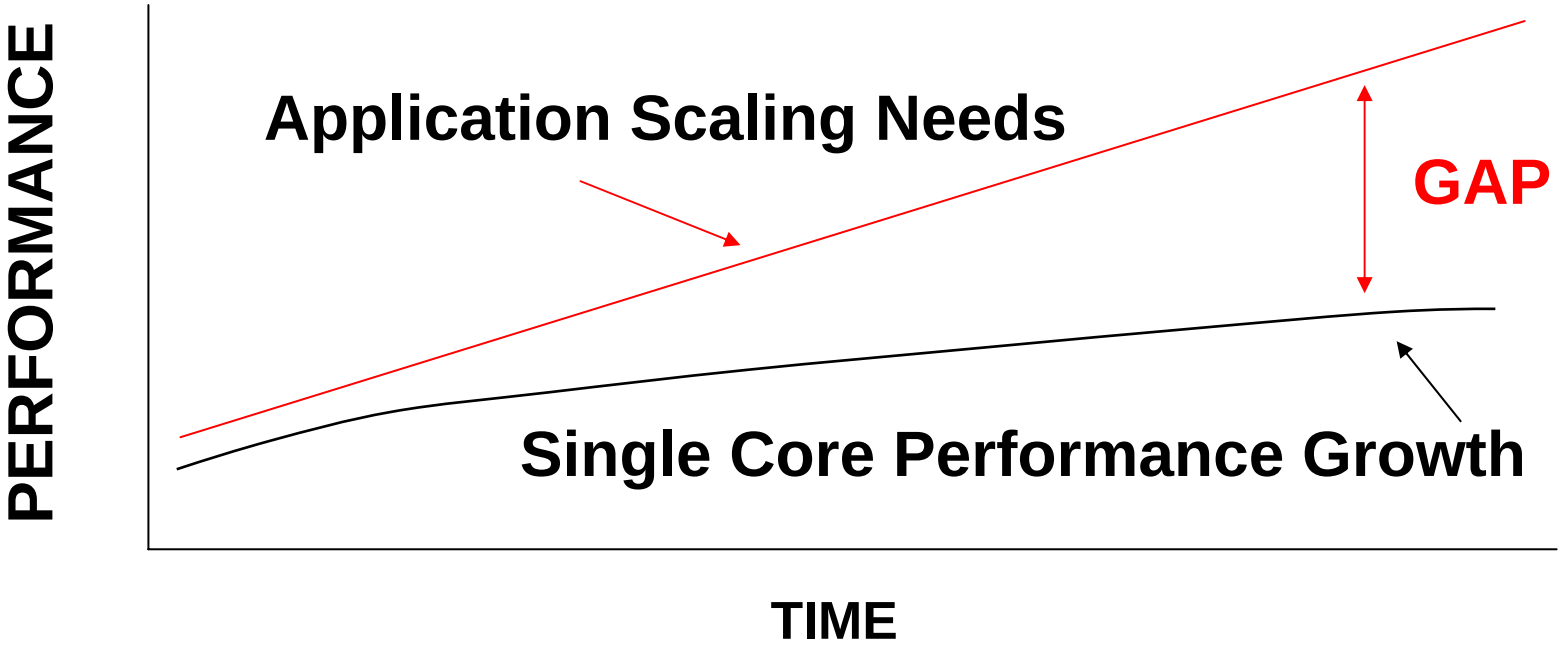
Heterogeneous Cores Can Reduce Energy Consumption (And Reduce the Total number of Cores Needed?)

- Can require Application work to utilize the mixed cores
- Languages, Tools and Libraries can minimize the impact of complexity

Design for Availability and Serviceability

- HW Failures will happen
- Machines need to be designed to keep them from effecting user applications as much as possible
- Repairing failures quickly and easily is critical
- Application Impact:
  - With enough circuits error rates will not be ‘zero’
  - Need algorithms that are fault tolerant
    - Compute more than once – or more than one way?
    - Internal Consistency checking?

Technology Trends and Challenges
Programmer Productivity



Key Problem: Frequency Improvements Do Not Match App Needs

Increasing Burden On The Application Design

Objective: Provide Tools to allow Scientists to Bridge the Gap

ROADRUNNER

Roadrunner: Science, Cell and a Petaflop/s

International Supercomputing Conference

Special Roadrunner Session, 6-18-2008

Andy White

Los Alamos

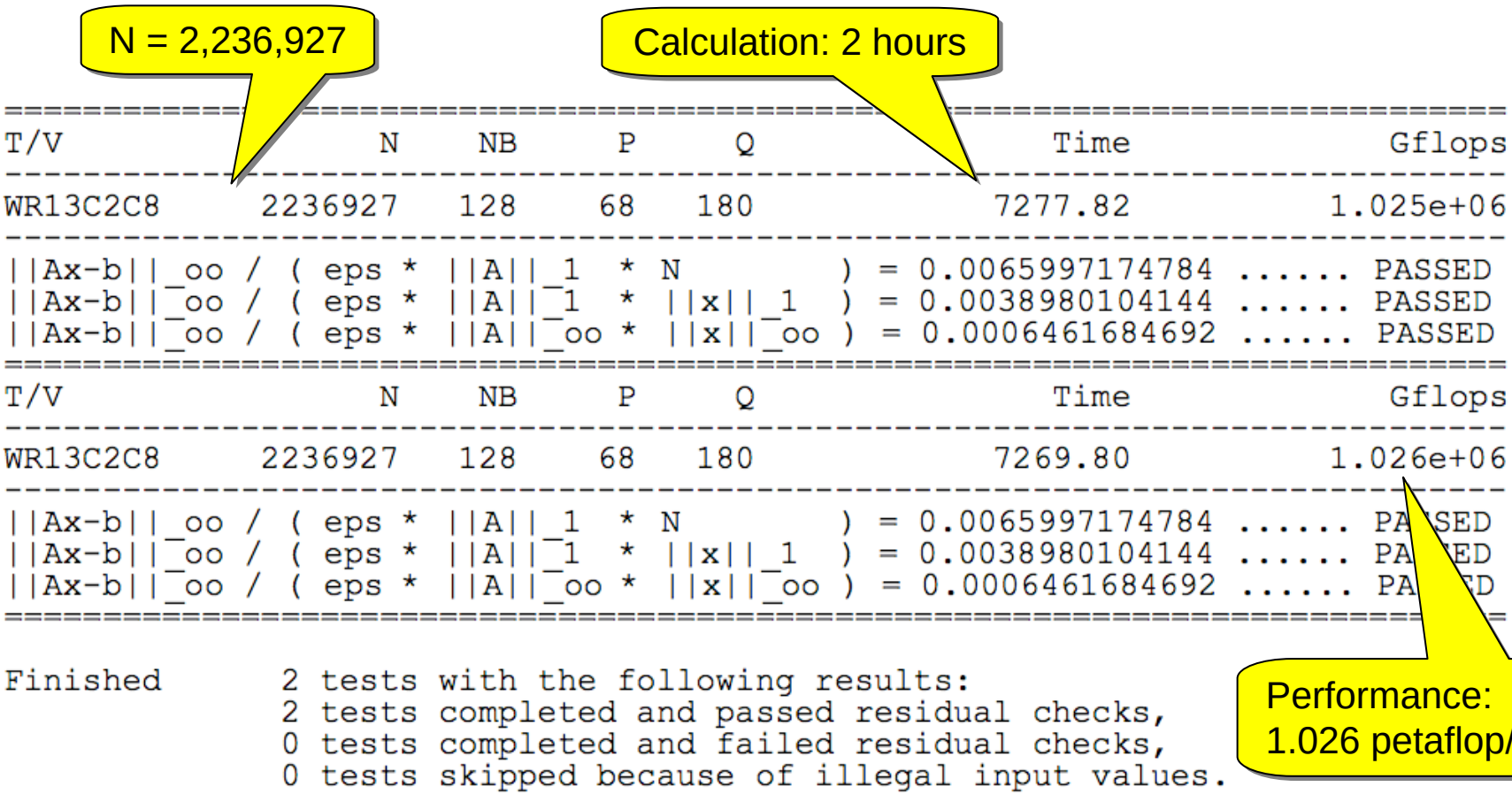
Don Grice

IBM

The partnership between Los Alamos and IBM made Roadrunner possible.

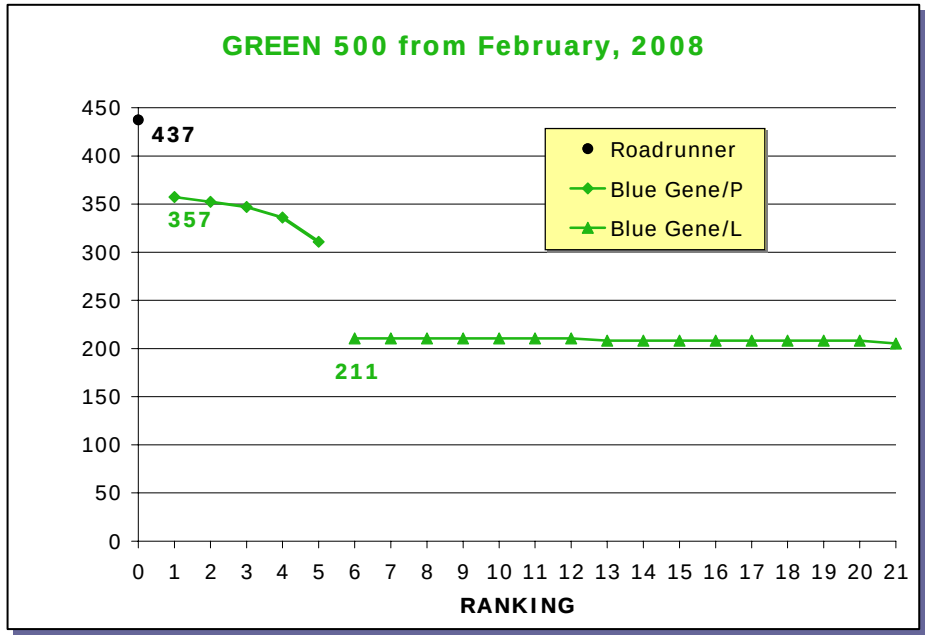
- Los Alamos began working with IBM in 2002 on the possibilities of the Cell processor
- Roadrunner was selected via a competitive procurement (2006) for a petascale supercomputer
- Over the last year we have proven that Roadrunner has great potential.
- Beginning May 23, we have begun realize that potential.

Roadrunner first achieved a petaflop/s at 3:30 am, Monday, May 26.



Achieving a petaflop/s in less than 3 days demonstrates the stability of the Roadrunner system.

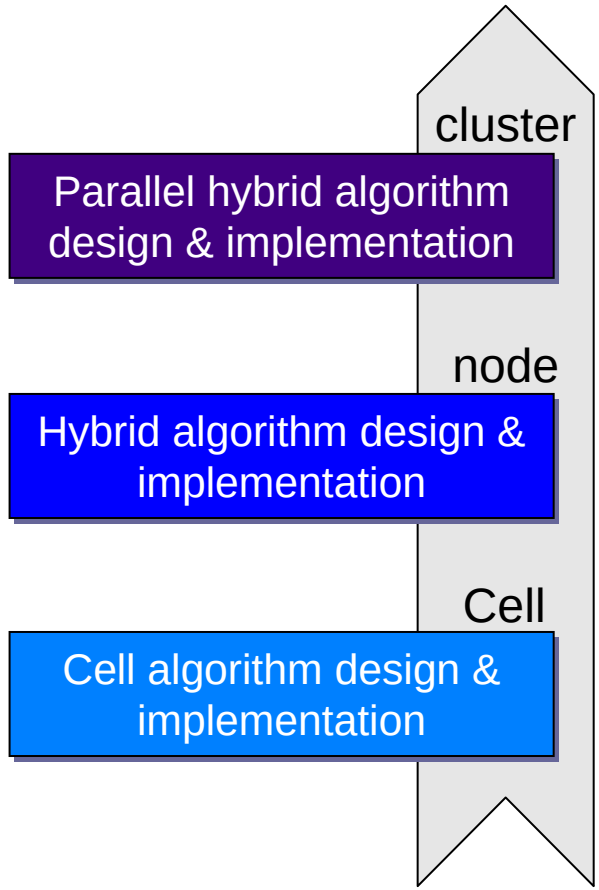
- **Full system available**
  - 8:30 am, Friday, May 23
- **Full system job launch tests begin**
  - 3:00 pm, Friday, May 23
- **First full system LINPACK launch**
  - 8:30 pm, Friday, May 23
  - Node failure after running an hour
- **Successful LINPACK runs**
  - 5:45 pm, Saturday, May 24 (879 TF/s)
  - 2:45 pm, Sunday, May 25 (945 TF/s)
  - 1:10 am, Monday, May 26 (997 TF/s)
  - 3:30 am, Monday, May 26 (Petaflop/s)



Roadrunner is also very energy efficient, 437 MF/s per watt.

We have focused on important application codes.

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



NASA Study: Cell Processor Shows Promise for Climate Modeling

Solar Radiation Component of GEOS-5

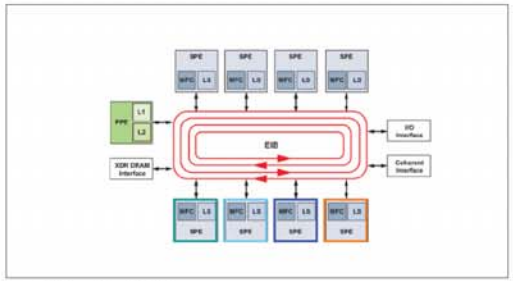
Impacts of the IBM Cell Processor on Supporting Climate Models

Shujia Zhou[1,2], Daniel Duffy[1,3], Tom Clune[1], Max Suarez[1], Samuel Williams[4], Milt Halem[5]

Introduction:

NASA is interested in the potential performance and cost benefits of adapting some science applications to emerging nontraditional processors such as the IBM Cell Broadband Engine System (hereafter referred to as "Cell"). The Cell is a multi-core system with the capability of increasing performance by one to two orders of magnitude over traditional processors. However, the Cell's characteristics, 256K byte local memory in a single SPE (Synergistic Processing Element) as well as the new low-level communication mechanism, make it very challenging to port a full application such as the NASA Goddard Earth Observing System Model, Version 5 (GEOS-5). Like other climate and weather models, GEOS-5 consists of dynamics and column physics. To avoid the complexity of porting a full application in this feasibility study, we selected a single component of GEOS-5, namely the solar radiation component, which has been used in various climate and weather models.

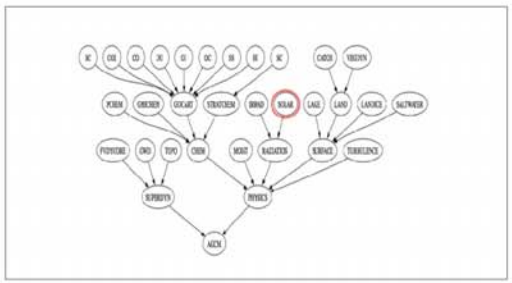
CELL Chip:



- 205 single-precision GFLOPS
- High-speed data ring (EIB) with a sustained bandwidth of 205 GB/s
- 25.6 GB/s processor-to-memory bandwidth
- 256 KB local store at SPE

NASA GEOS-5 Atmospheric Model:

The GEOS-5 atmospheric model consists of a large number of interacting physical components. Although not large in size (~2000 line Fortran code), the solar radiation component, the ultimate source of energy for the Earth's climate, is one of the most computationally-intensive components. Along with the infrared radiation component, it consumes at least 20% of the computing time of the atmospheric model. The requirements of the solar radiation component in loading and storing data from and to main memory are also small relative to the computational load. These factors make the solar radiation component well-suited for one of Cell's characteristics: a high ratio of computation to data transfer capabilities. In addition, this component consists of calculations in each column (along the vertical direction, on a grid point), which are independent of other columns, so-called "embarrassingly parallel." This greatly simplified the coding involving the Cell communication mechanism.

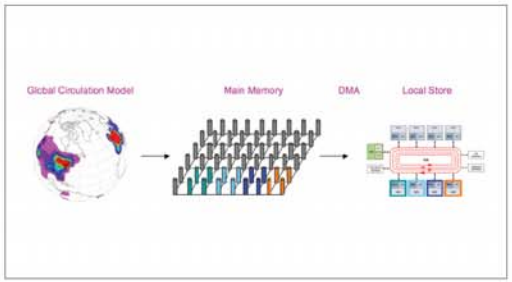


Translate the Code from Fortran to C and Port It to Cell:

We converted the ~2000-line solar radiation code from Fortran to C (due to a Fortran compiler being unavailable for the SPE as of summer 2007). Converting the code includes three steps: (1) translating the code from Fortran to C, (2) inserting library calls to transfer data between the PPE and SPE with DMA (Direct Memory Access), and (3) vectorizing the most computationally-intensive function (currently a manual process not performed by the compiler). We found that 16-byte alignment and direct management of memory address (mapping the local multi-dimensional array index in SPEs to the global array index in main memory) required considerable time and attention to implement, since there are 27 various shapes of 1D, 2D, and 3D arrays involving data transfer between PPE and SPEs through DMA. However, we believe that conceptually these modifications are not difficult for a user who knows C and MPI. Steps (1) and (3) took ~3 weeks each. However, the newly released IBM XL V11.1 Fortran compiler should make steps (1) and (3) unnecessary. At this time, we do not have access to this Fortran compiler and therefore have not evaluated its performance, particularly regarding autovectorization. Hence, step (3) might still be necessary for further performance improvement. Step (2) is unaffected by the availability of foreseeable Fortran compilers. However, a Fortran compiler will require writing a wrapper to use DMA, since the DMA's API is in C.

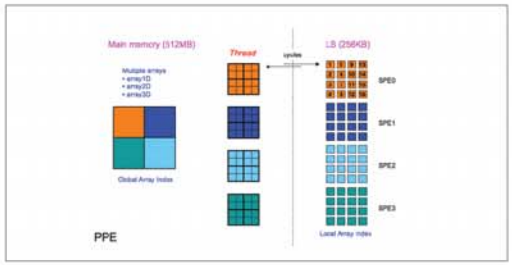
Architecture:

Decomposing the solar radiation code for SPEs



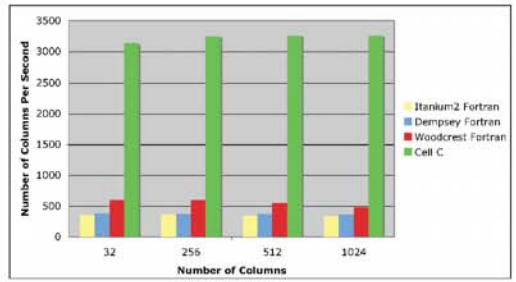
Processing four columns, each cycle using one SPE, is the optimal case

Flow Diagram of Data Transfer via DMA (4 SPEs):



Performance Evaluation of C-Version Cell Code vs. Original Baseline in Fortran:

- Since the current Cell processor in an IBM BladeCenter QS20 has a deficiency in processing double-precision floating point operations, we will only present results for the single-precision C-version solar radiation code. Our detailed memory analysis reveals that each SPE can contain four entire columns of data in local memory simultaneously, allowing us to further optimize the performance by vectorizing (SIMDizing) two computationally-intensive functions across these four independent columns. In comparison with the best baseline results (the single-precision, Fortran-version code), we found that in the case of 1024 columns per Intel core, a Cell with eight SPEs is 6.76x faster than Intel Xeon Woodcrest (2.66GHz, four floating point operations per cycle), 8.91x faster than Intel Xeon Dempsey (3.2 GHz, two floating point operations per cycle), and 9.85x faster than Intel Itanium2 (1.5 GHz, four floating point operations per cycle), respectively.
- We note that the C-version runs measurably slower than the Fortran baseline by factors of 1.46x, 2.56x, and 5.81x on Woodcrest, Dempsey, and Itanium2, respectively. Here we assume that the Cell performance is not hampered by our language conversion, and therefore will base our performance comparisons against the Fortran baseline. (Note: the O3 option is used for the compilation on all the processors reported here.)
- Beyond the Cell's powerful computational capability, we believe there are two essential factors for dramatic performance improvement over cache-based Intel processors: (1) fitting all computation-support data such as look-up tables into the SPE's local store, and (2) explicitly transferring data between SPEs and main memory via DMA. Once the data is in the local store, there are no effects from cache and TLB (Translation Lookaside Buffer), and DMA is a small fraction of total time.
- The more than 6x improvement on the solar radiation component is certainly very encouraging since the other column physics components such as infrared radiation and moisture have a similar code structure. That means ~50% of the total computational load for the model can expect to obtain significant performance benefits on the Cell.



Summary:

- We found that the IBM Cell technology clearly provides a new way of dramatically improving the performance of climate and weather applications.
- Identifying the appropriate algorithms is the key to lowering the porting cost. In particular, we found that NASA GEOS-5 column physics components can improve their performance with Cell technology.

Acknowledgments:

We would like to thank Carlos Cruz and Bruce Van Aartsen for translating some of the code from Fortran to C. Further, we would like to thank Tsengdar Lee (NASA High End Computing Program) for providing funding, Phil Webster for project initiation, Mike Seablom for his inspiration and helpful discussion, John Shalf for sharing his insight on the IBM Cell technology, the NASA Center for Computational Sciences for installing the IBM Cell Simulator for code development, the Dice Project for training support, and the UMBC Multicore Computational Center for providing access to an IBM BladeCenter QS20.

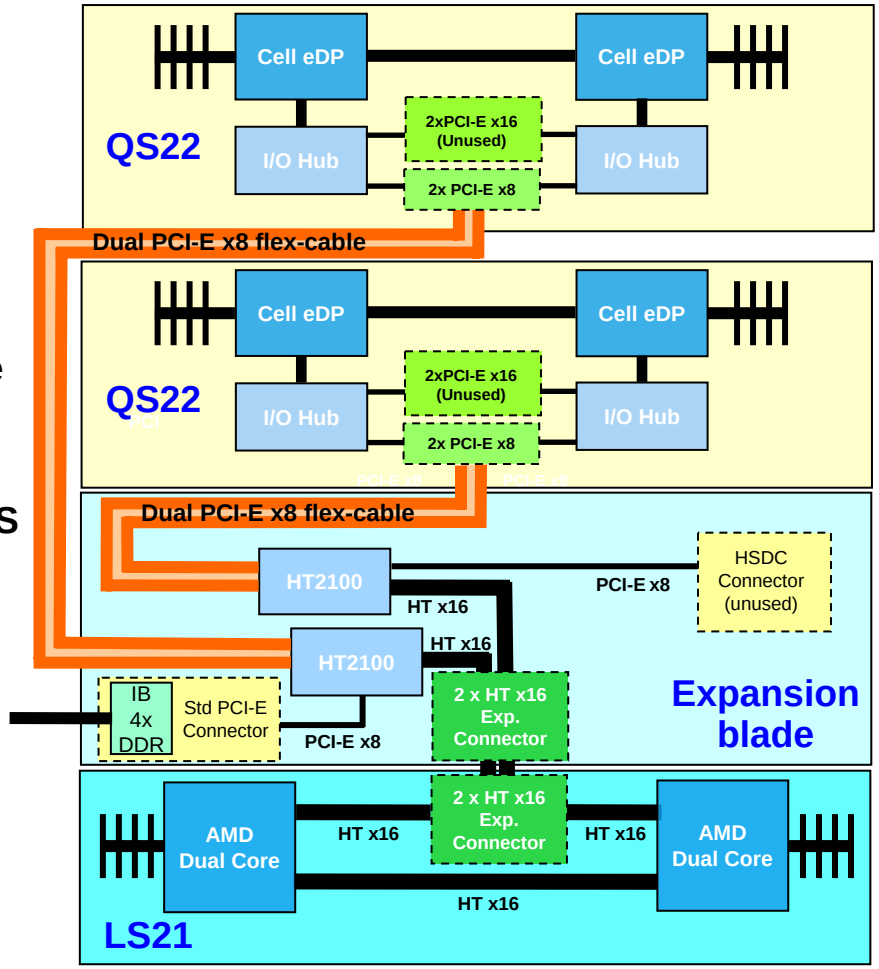
[1] NASA Goddard Space Flight Center, [2] Northrop Grumman Corporation, [3] Computer Sciences Corporation, [4] University of California, Berkeley, [5] University of Maryland, Baltimore County

Roadrunner at a glance

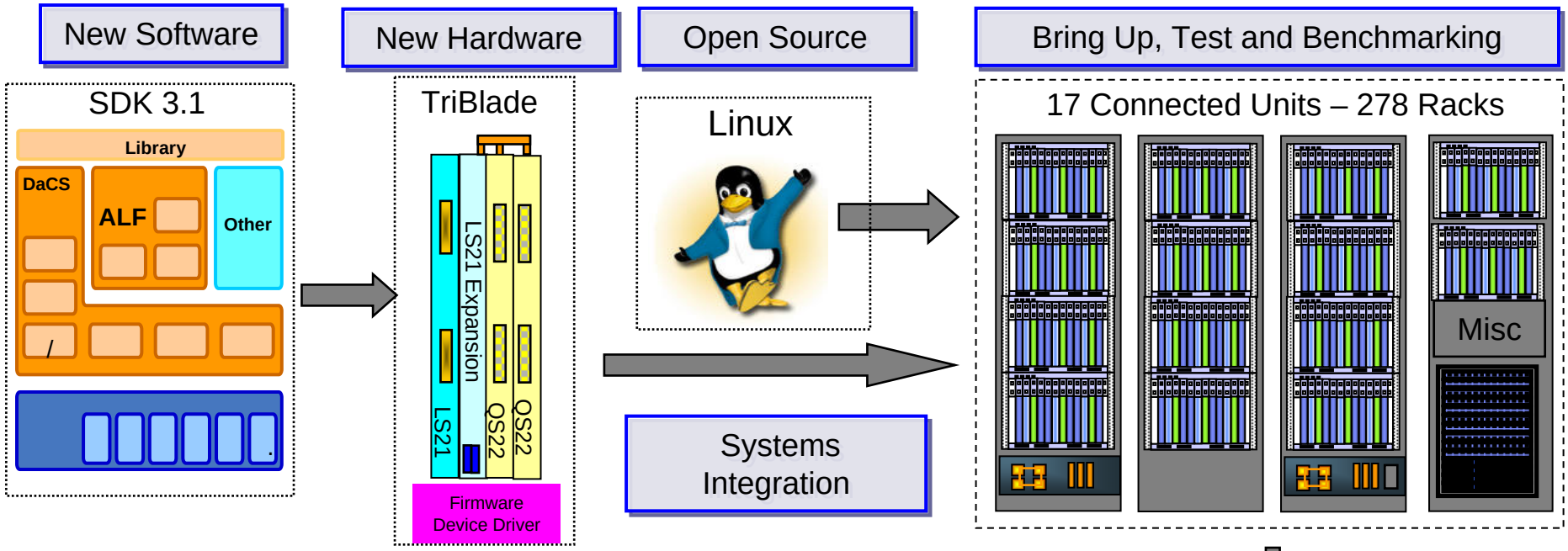
- **Cluster of 17 Connected Units**
  - 12,240 IBM PowerXCell 8i accelerators
  - 6,120 AMD dual-core Opterons (comp)
  - 408 AMD dual-core Opterons (I/O)
  - 34 AMD dual-core Opterons (man)
  - 1.332 Petaflop/s peak (PowerXCell)
  - 44 Teraflop/s peak (Opteron-comp)
  - 1.026 Petaflop/s sustained Linpack
- **InfiniBand 4x DDR fabric**
  - 2-stage fat-tree; all-optical cables
  - Full CU bi-section bi-directional BW
    - 384 GB/s (CU)
    - 3.3 TB/s (system)
  - Non-disruptive expansion to 24 CUs
- **98 TB aggregate memory**
  - 49 TB Opteron
  - 49 TB Cell
- **408 GB/s peak File System I/O:**
  - 204x2 10G Ethernets to Panasas
- **RHEL & Fedora Linux**
- **SDK for Multicore Acceleration**
- **xCAT Cluster Management**
  - System-wide GigEnet network
- **2.35 MW Power (Linpack):**
  - 437 Megaflop/s per Watt
- **Other:**
  - 278 racks
  - 5200 ft^2
  - 500,000 lbs.
  - 55 miles of IB cables

A Roadrunner Triblade node integrates Cell and Opteron blades

- **QS22 is the newly announced IBM Cell blade containing two new enhanced double-precision (eDP/PowerXCell™) Cell chips**
- **Expansion blade connects two QS22 via four PCI-e x8 links to LS21 & provides the node's ConnectX IB 4X DDR cluster attachment**
- **LS21 is an IBM dual-socket Opteron blade**
- **4-wide IBM BladeCenter packaging**
- **Roadrunner Triblades are completely diskless and run from RAM disks with NFS & Panasas only to the LS21**
- **Node design points:**
  - One Cell chip per Opteron core
  - ~400 GF/s double-precision & ~800 GF/s single-precision
  - 16 GB Cell memory & 16 GB Opteron memory



Roadrunner Overview – Building Blocks



- A New Programming Model extended from standard, cluster computing
- Hybrid and Heterogeneous HW
- Built around BladeCenter and Industry IB-DDR

Next stop, exaflop?

www.lanl.gov/roadrunner

www.lanl.gov/news

www-03.ibm.com/press/us/en/pressrelease/24405.wss

www.ibm.com/deepcomputing

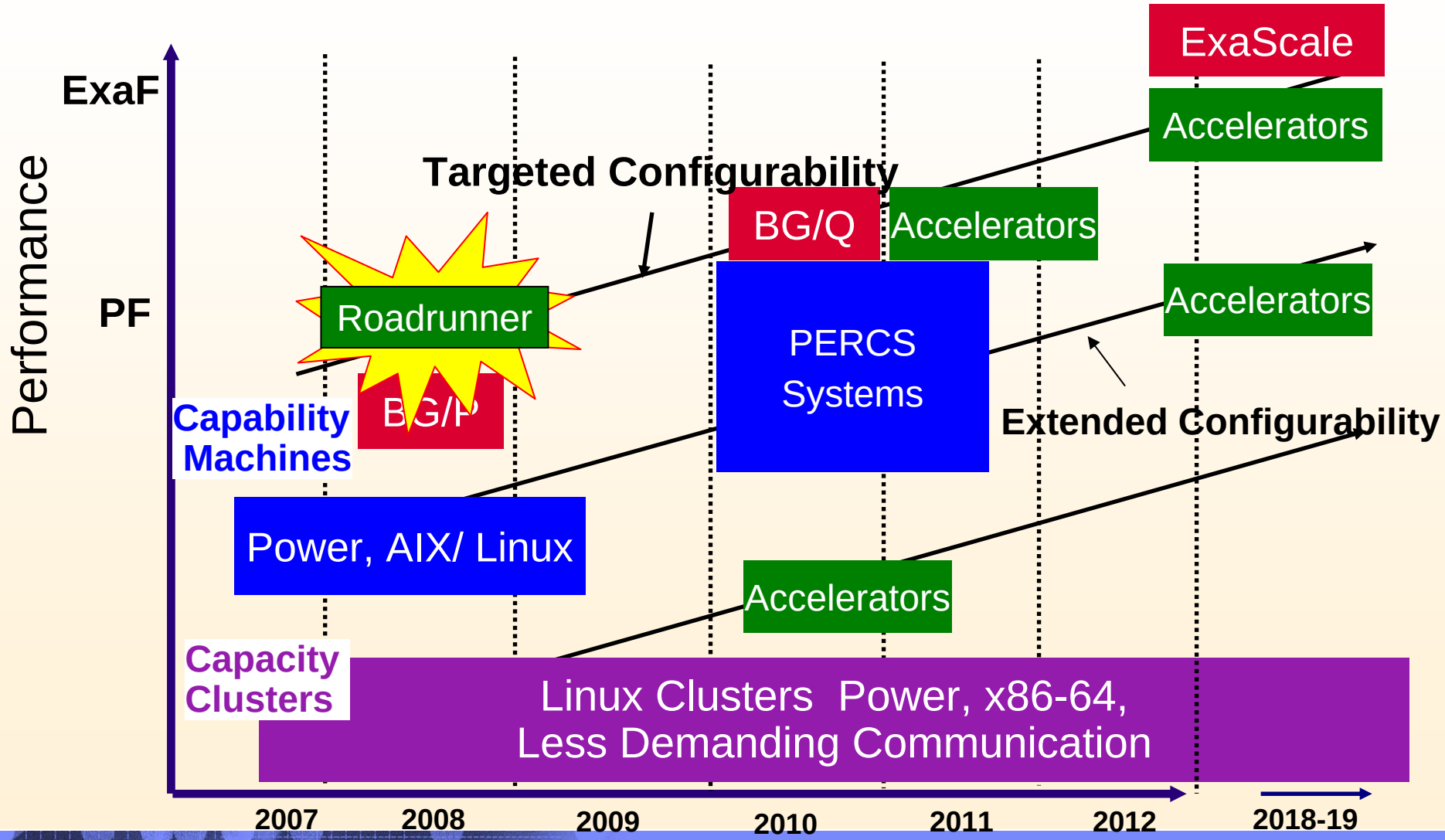
IBM Ultra Scale Approaches

- Blue Gene – Maximize Flops Per Watt
  with Homogeneous Cores
  by reducing Single Thread Performance

- Power/PERCS – Maximize Single Thread Performance
  with Homogeneous Cores

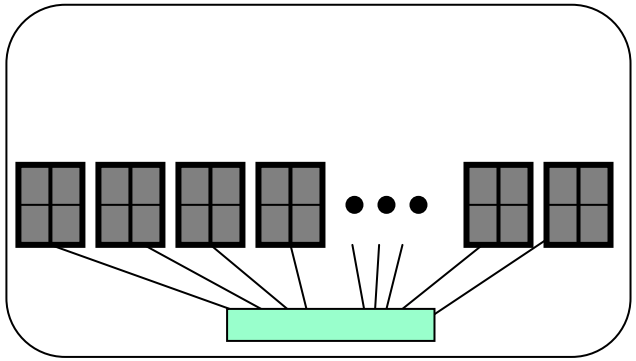
- Roadrunner – Use Heterogeneous Cores and
  an Accelerator Software Model
  to Maximize Flops Per Watt
  and keep High Single Thread Performance

HPC Cluster Directions

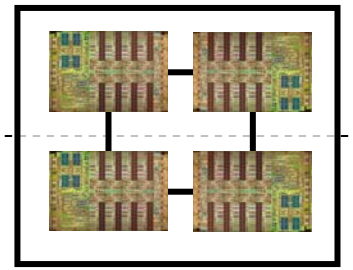


Roadrunner anticipates the future of supercomputing.

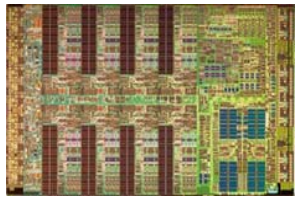
cluster



node



socket



Message Passing

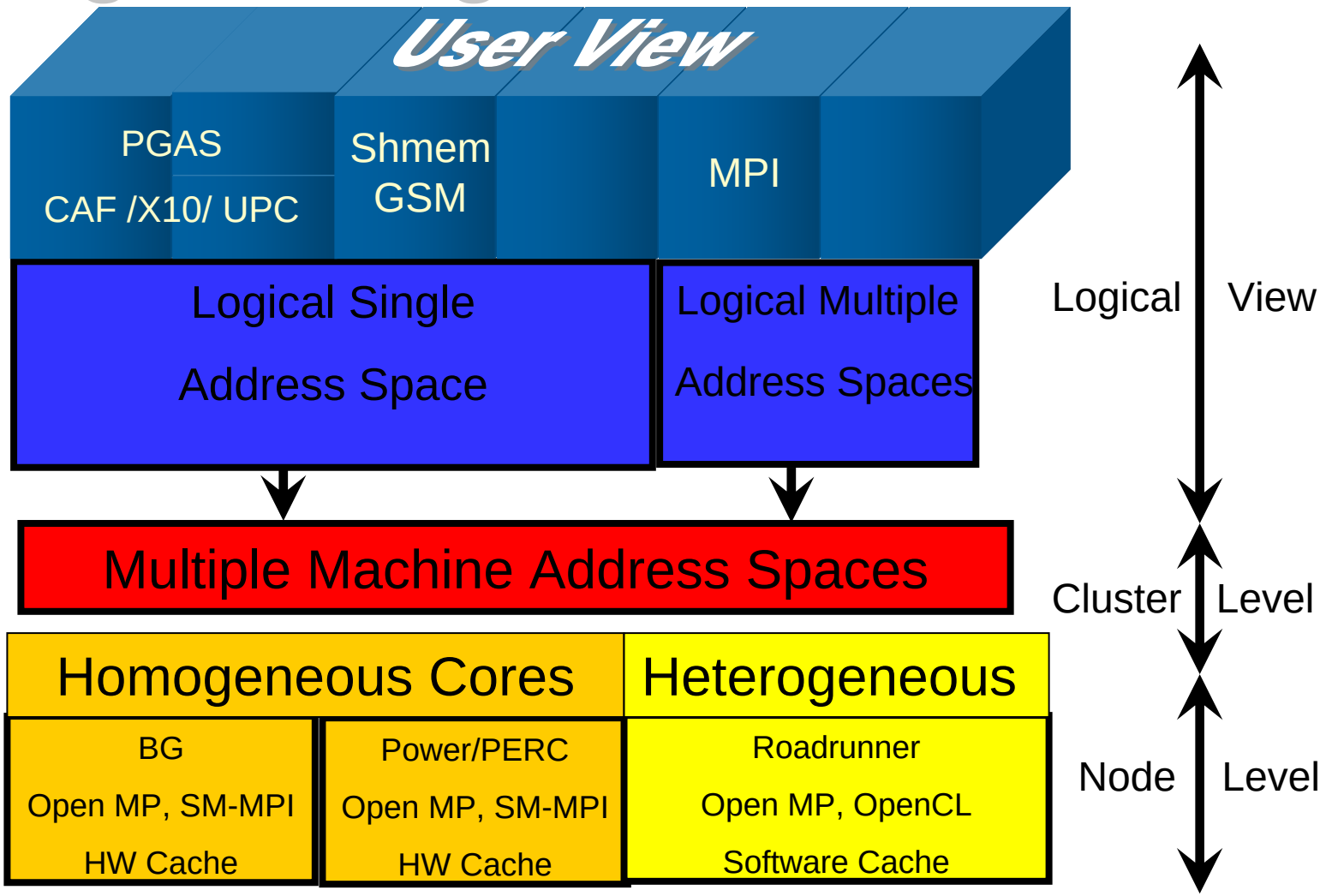
Not Message Passing

Hybrid & many core technologies will require new approaches:

DaCS/ALF, libSPE,

OpenMP, PGAS, …

Programming Models: Architecture



Things to Think About
Algorithm Changes for New System Limits?

- Computing will be ‘free’ but Pin BW will be expensive
  - Memory BW
  - Communication BW
- Need to restructure algorithms around the new limits
- Roadrunner Linpack used Redundant computing to reduce Communication BW
- DGEMM was restructured to minimize Memory BW
- Do we need a new FFT?

THANK YOU