Communication Analysis of the Cell Broadband Engine Processor

Fabrizio Petrini
Pacific Northwest National Laboratory
fabrizio.petrini@pnl.gov

Michael Perrone
IBM TJ Watson
mpp@us.ibm.com

Michael Kistler and Gordon Fossum
IBM Austin Research Laboratory
mkistler@us.ibm.com, fossum@us.ibm.com

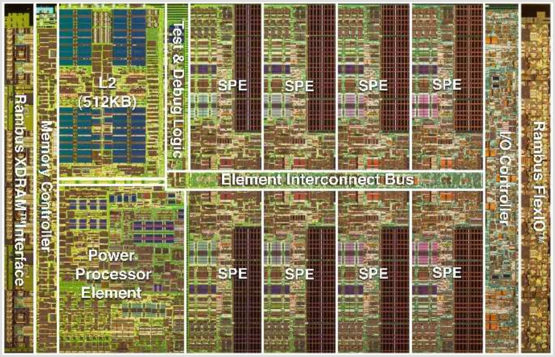
Battelle

EDGE Workshop, UNC, May 2006

Pacific Northwest National Laboratory
Operated by Battelle for the U.S. Department of Energy

The Charm of the IBM Cell Broadband Engine

- Extraordinary processing power
  - 8 independent processing units (SPEs)
  - One control processor
    - A traditional 64-bit PowerPC
- At 3.2 Ghz the Cell
  - a peak performance of 204.8 Gflops/second (single precision)
  - 14.64 Gflops/second (double precision)



Communication Performance

- Internal bus (Element Interconnect Bus EIB) with peak performance of 204.8 Gbytes/second
- Memory Bandwidth 25.6 Gbytes/second
- Impressive I/O bandwidth
  - 25 Gbytes/second inbound
  - 35 Gbytes/second outbound
- Many outstanding memory requests
  - Up to 128, typical of multi-threaded processors

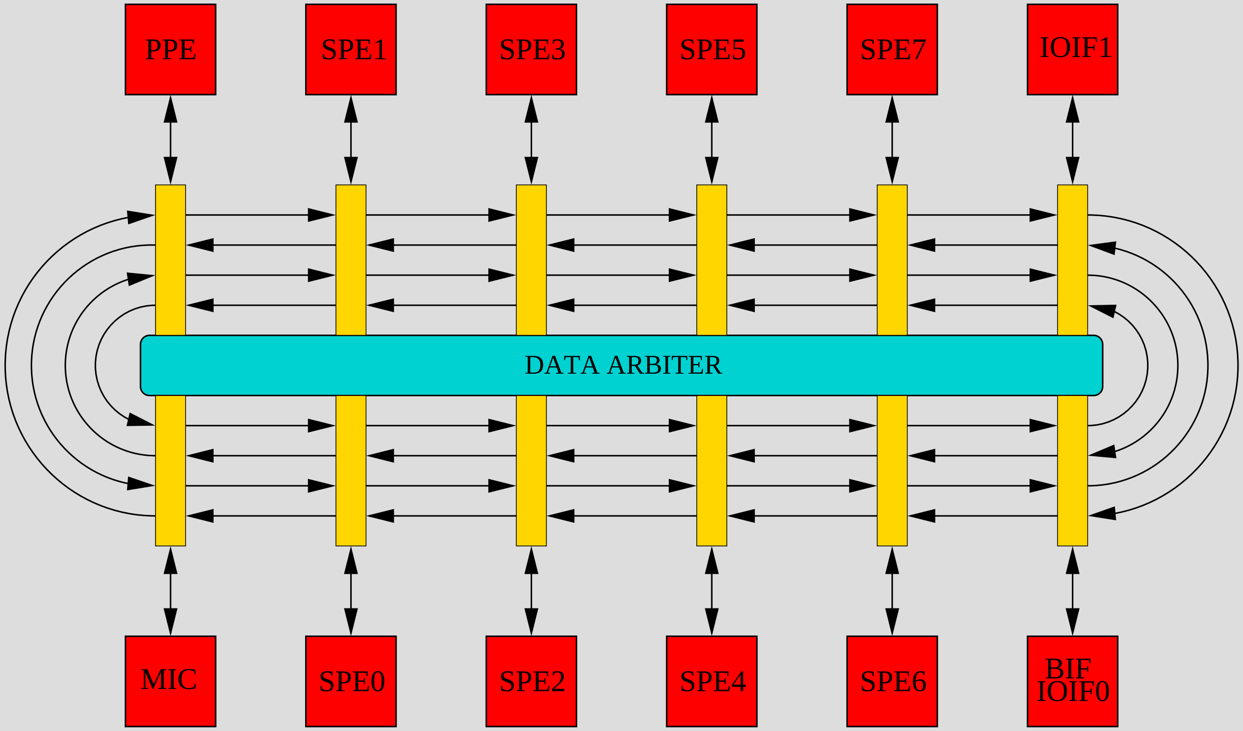
Moving the Spotlight from Processor Performance to Communication Performance

- Traditionally the focus is on (raw) processor performance
- Emphasis is now shifting towards communication performance
- Lots of (peak) processing power *inside* a chip (approaching Teraflops/sec)
  - Small fraction is delivered to applications
- Lots of (peak) aggregate communication bandwidth *inside* the chip (approaching Terabytes/sec)
  - But processing units do not interact frequently
- Small on chip local memories
  - Little data reuse
- Main memory bandwidth is the **primary** bottleneck
- And then I/O and network bandwidth

Dangerous Connection Between Memory and Network Performance and Programmability

- Programming model is already a critical issue
  - And it is going to get worse
- Low data-reuse increases the algorithmic complexity
- Memory and Network bandwidth are key to achieve performance and simplify the programming model
- Multi-core Uni-bus ☺

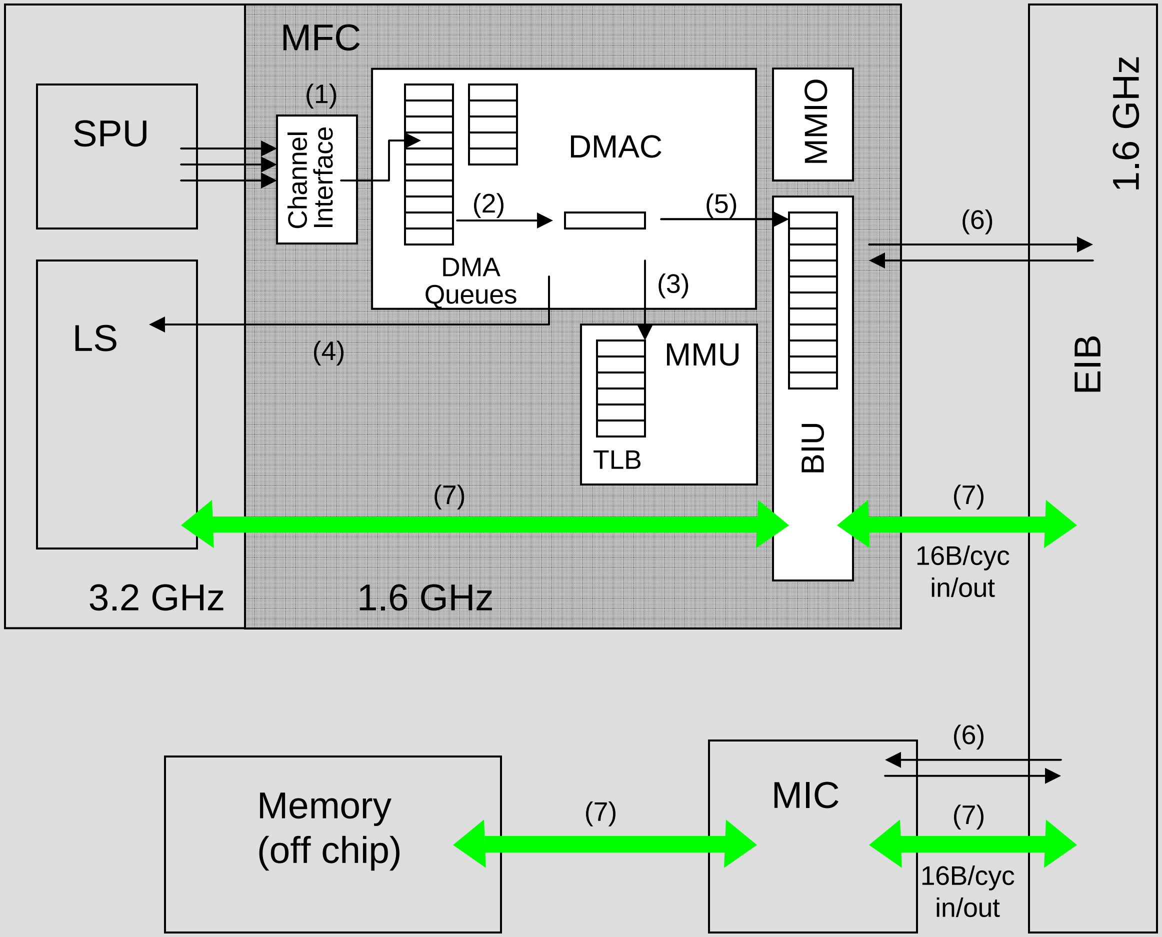
Internal Structure of the Cell BE



Cell BE Communication Architecture

- SPUs can only access programs and data in their local storage
- SPEs have a DMA controller that performs transfers between local stores, main memory and I/O
- SPUs can post list a list of DMAs
- SPUs can also use mailboxes and signals to perform basic synchronizations
- More complex synchronization mechanisms can support atomic operations
- All resources can be memory mapped

SPE Internal Architecture



Battelle

Pacific Northwest National Laboratory
U.S. Department of Energy

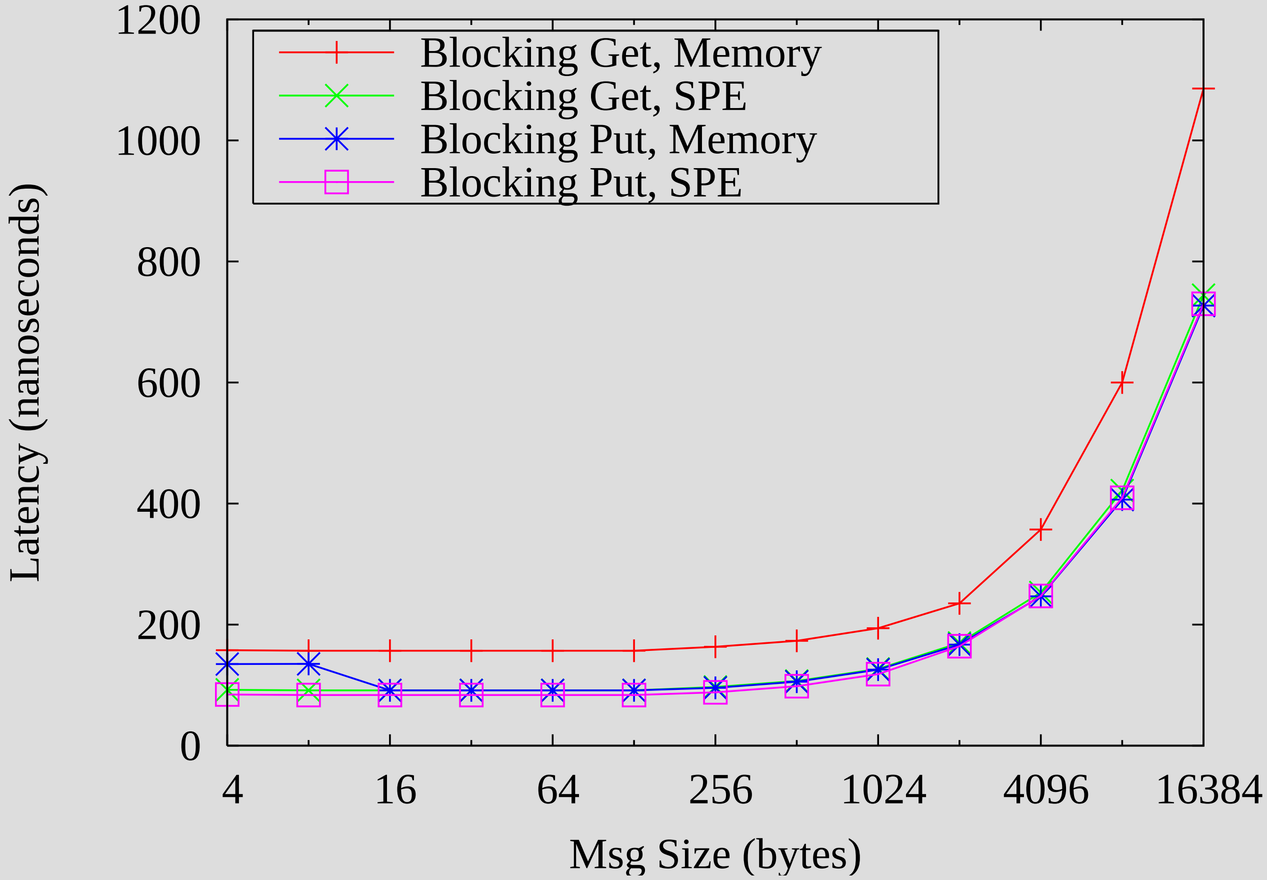
Basic Latencies (3.2 Ghz)

LATENCY COMPONENT	**CYCLES**	**NANOSECONDS**
DMA issue	10	3.125
DMA to EIB	30	9.375
List Element Fetch	10	3.125
Coherence Protocol	100	31.25
Data Transfer for inter-SPE put	140	43.75
TOTAL	290	90.61

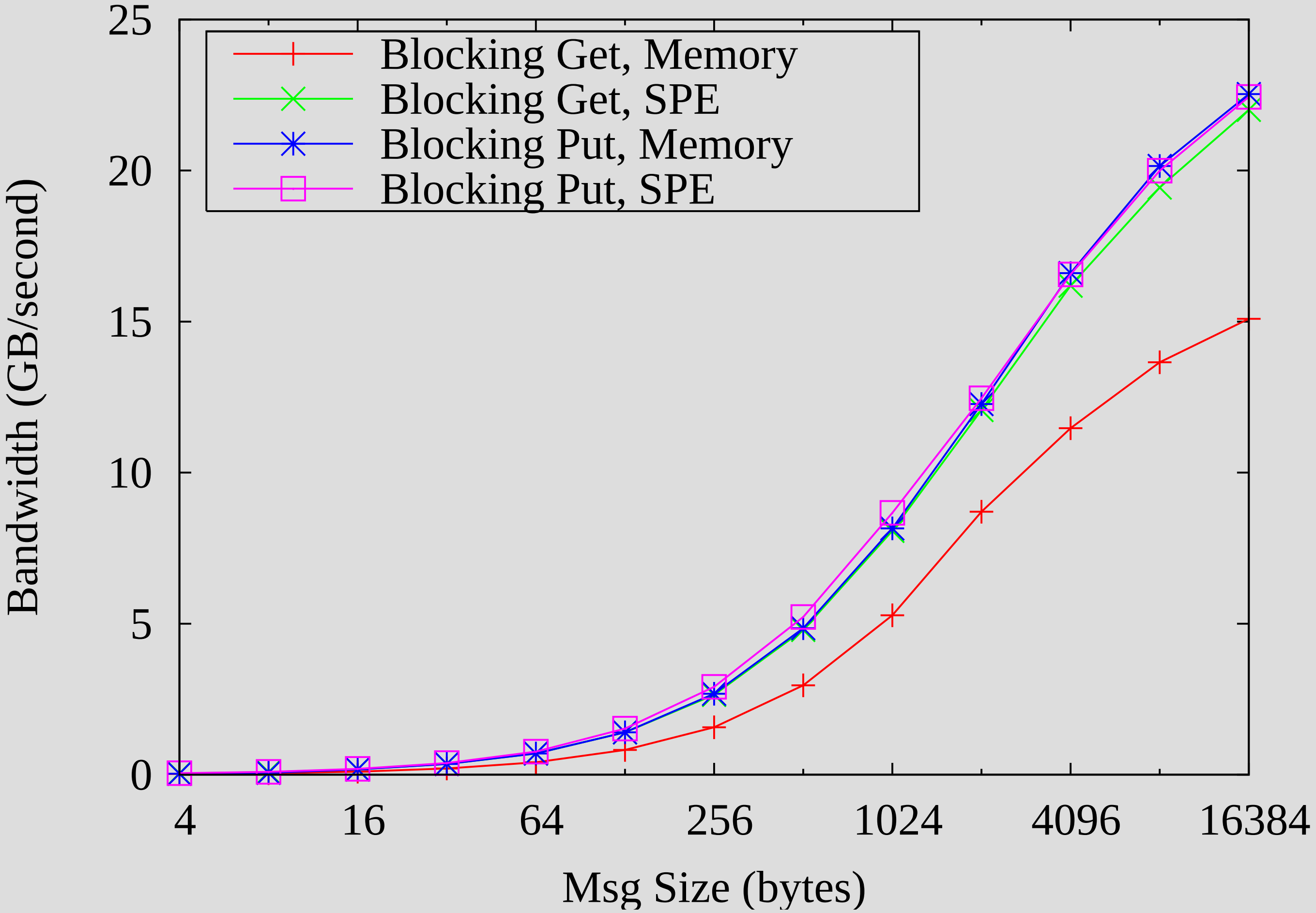
Is this is a processor or a supercomputer on a chip?

- Striking similarities with high-performance networks for supercomputers
  - E.g., Quadrics Elan4
- DMAs overlap computation and communication
- Similar programming model
- Similar synchronization algorithms
  - Barriers, allreduces, scatter & gather
- We can adopt the same techniques that we already use in high-performance clusters and supercomputers!

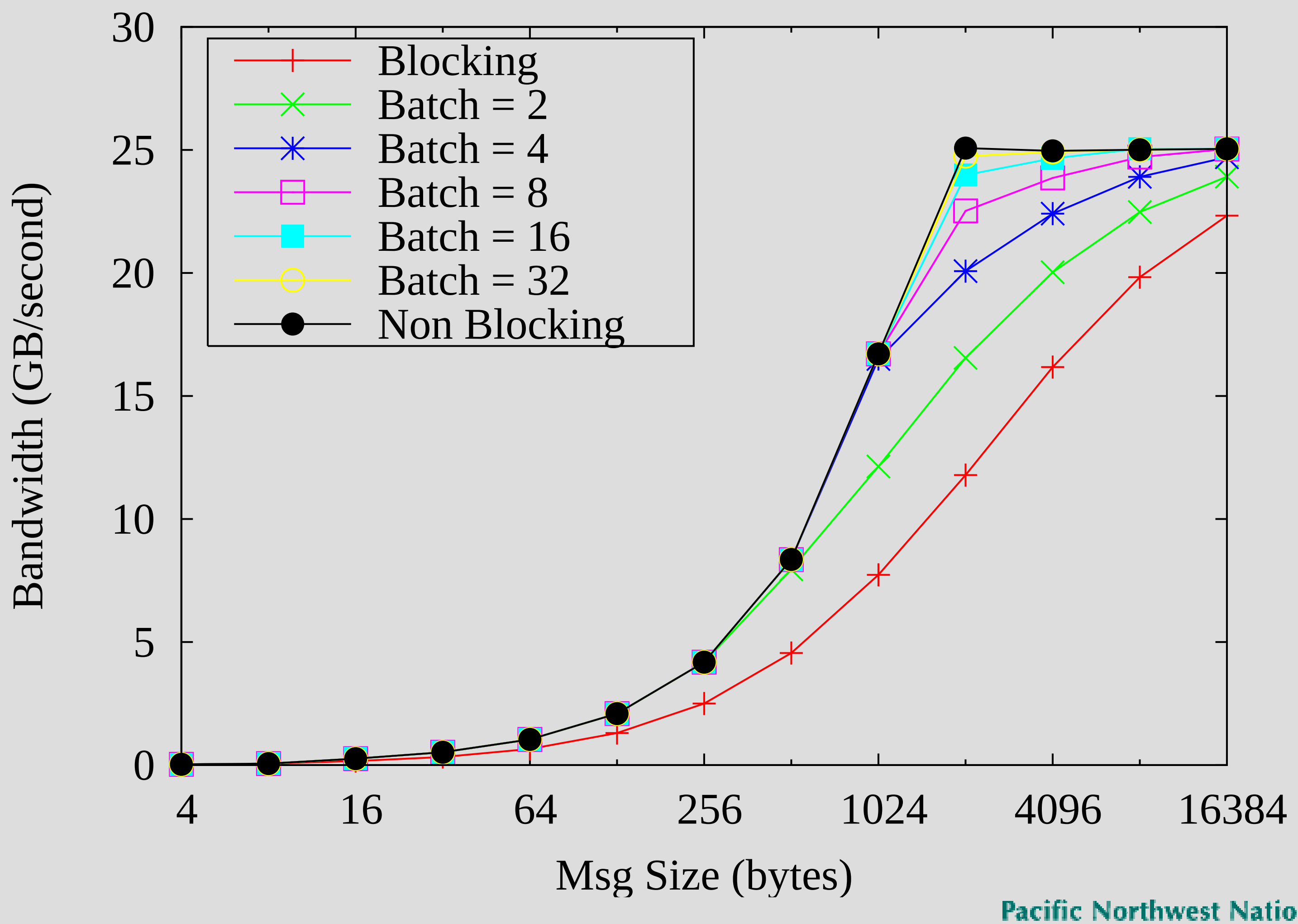
DMA Latency


Blocking Get, Memory
Blocking Get, SPE
Blocking Put, Memory
Blocking Put, SPE
Latency (nanoseconds)
1200
1000
800
600
400
200
0
4
16
64
256
1024
4096
16384
Msg Size (bytes)

DMA Bandwidth

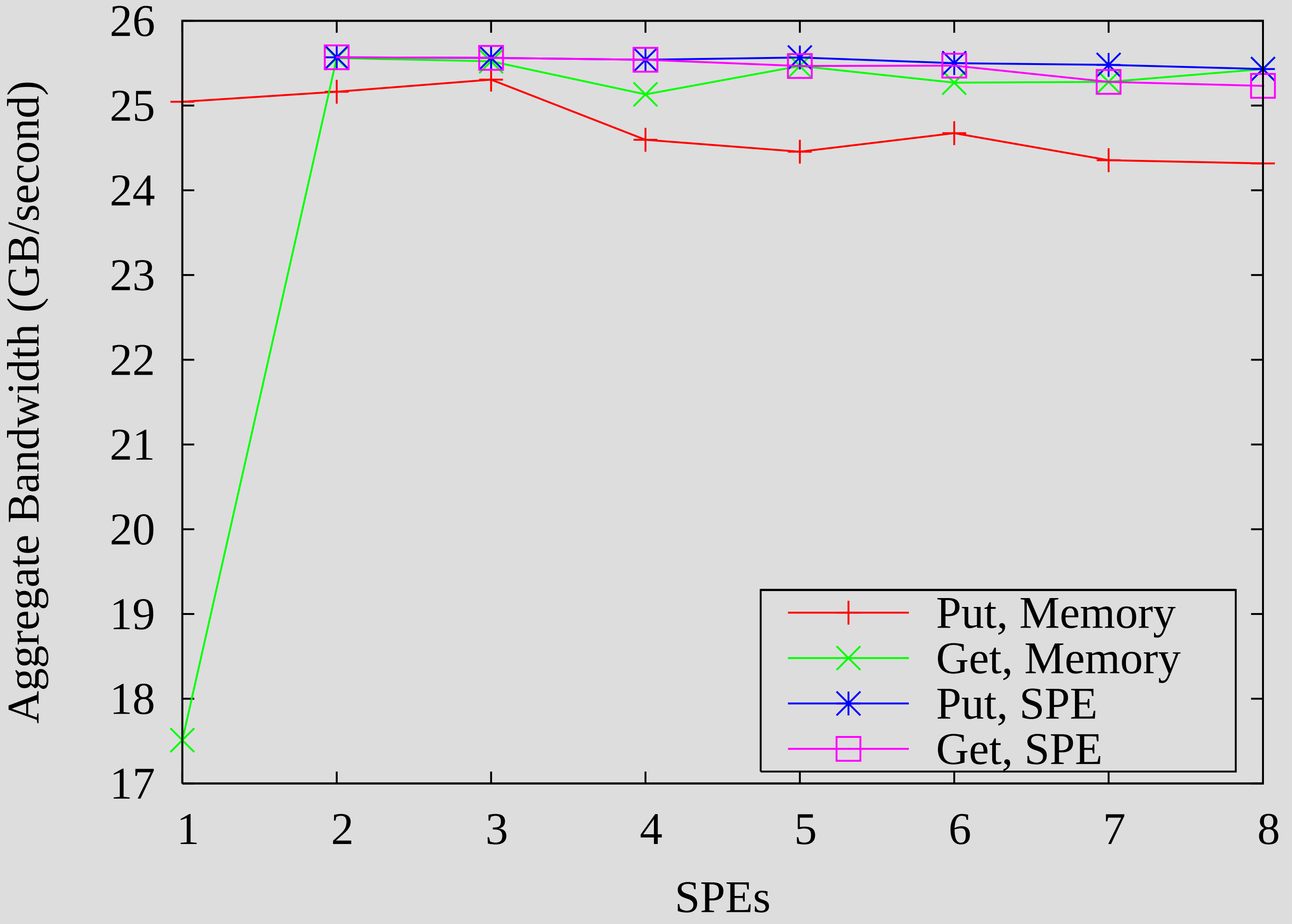


DMA batches (put)

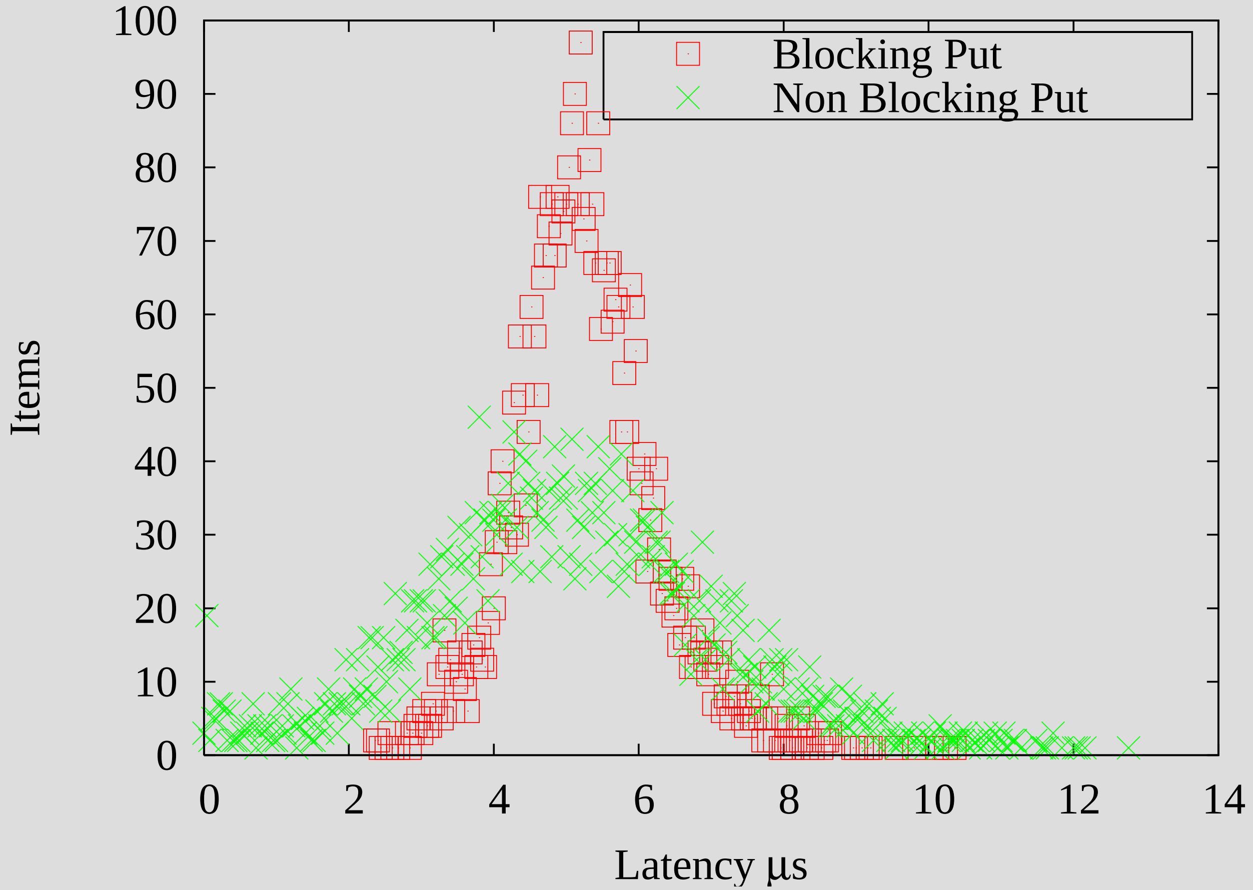


Battelle

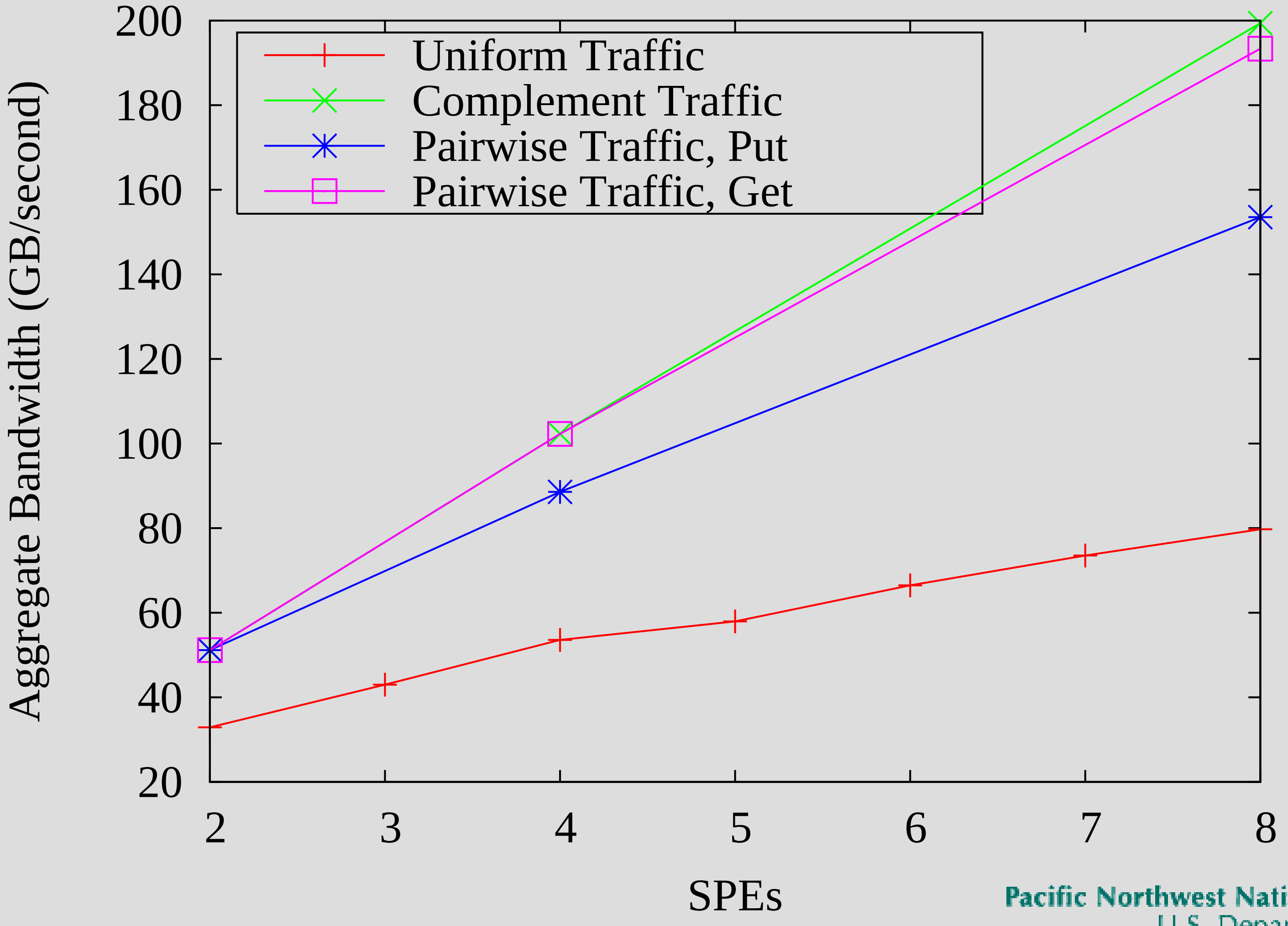
Hot Spot


Aggregate Bandwidth (GB/second)
26
25
24
23
22
21
20
19
18
17
1
2
3
4
5
6
7
8
SPEs
Put, Memory
Get, Memory
Put, SPE
Get, SPE

Latency Distribution under Hot-Spot



Aggregate Behavior



Battelle

Pacific Northwest National Laboratory
U.S. Department of Energy

Putting the Pieces Back Together

- We have discussed the “raw” communication capability of the network
- We now try to see how we can parallelize scientific application on the Cell BE
  - A point in a large design space
- Sweep3D: a well known scientific application
- A case study to provide insight on the various aspects of the Cell BE
  - Parallelization strategies, nature of parallelism, actual computation and communication performance

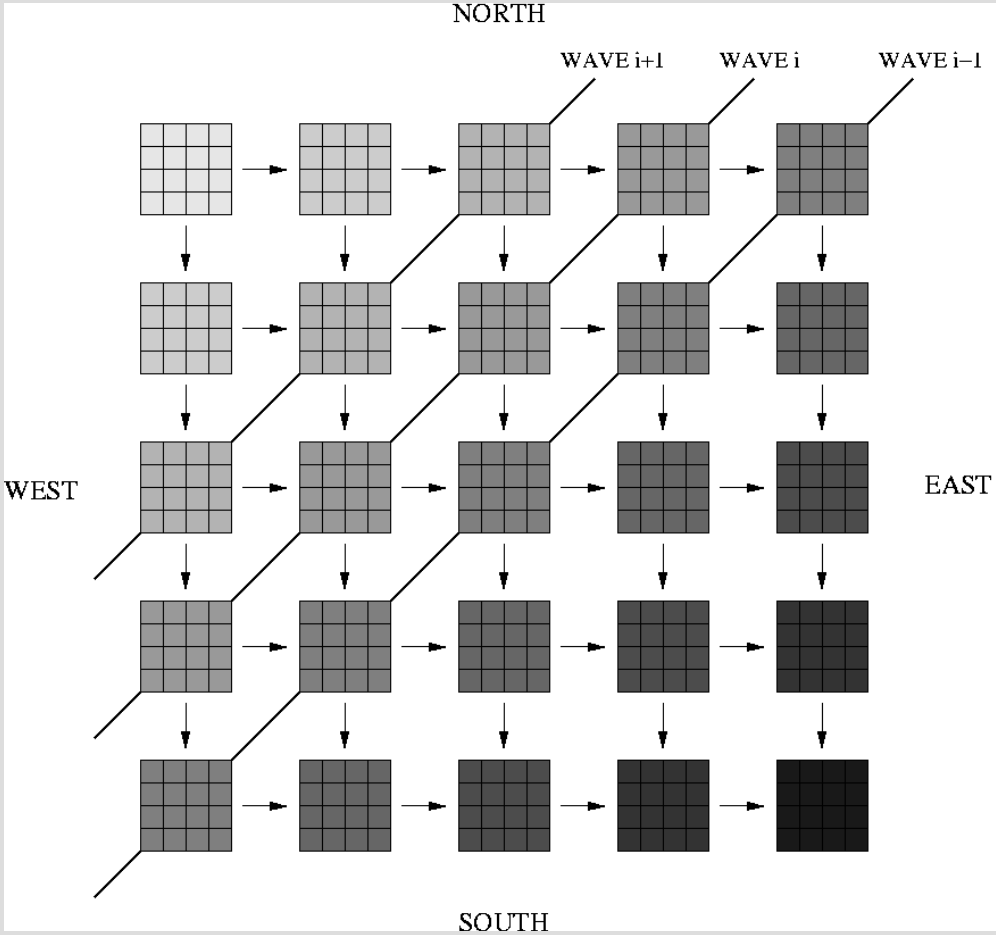
Challenges

- Initial excitement in the scientific community, but concerns about the
  - *Actual* fraction of performance that can be achieved with real applications
  - Complexity of developing new applications
  - Complexity of developing new parallelizing compilers
  - Whether there is clear migration path for existing legacy software, written using MPI, Shared memory programming libraries (Global Arrays, UPC, Cray Shmem, etc.)

Sweep3D

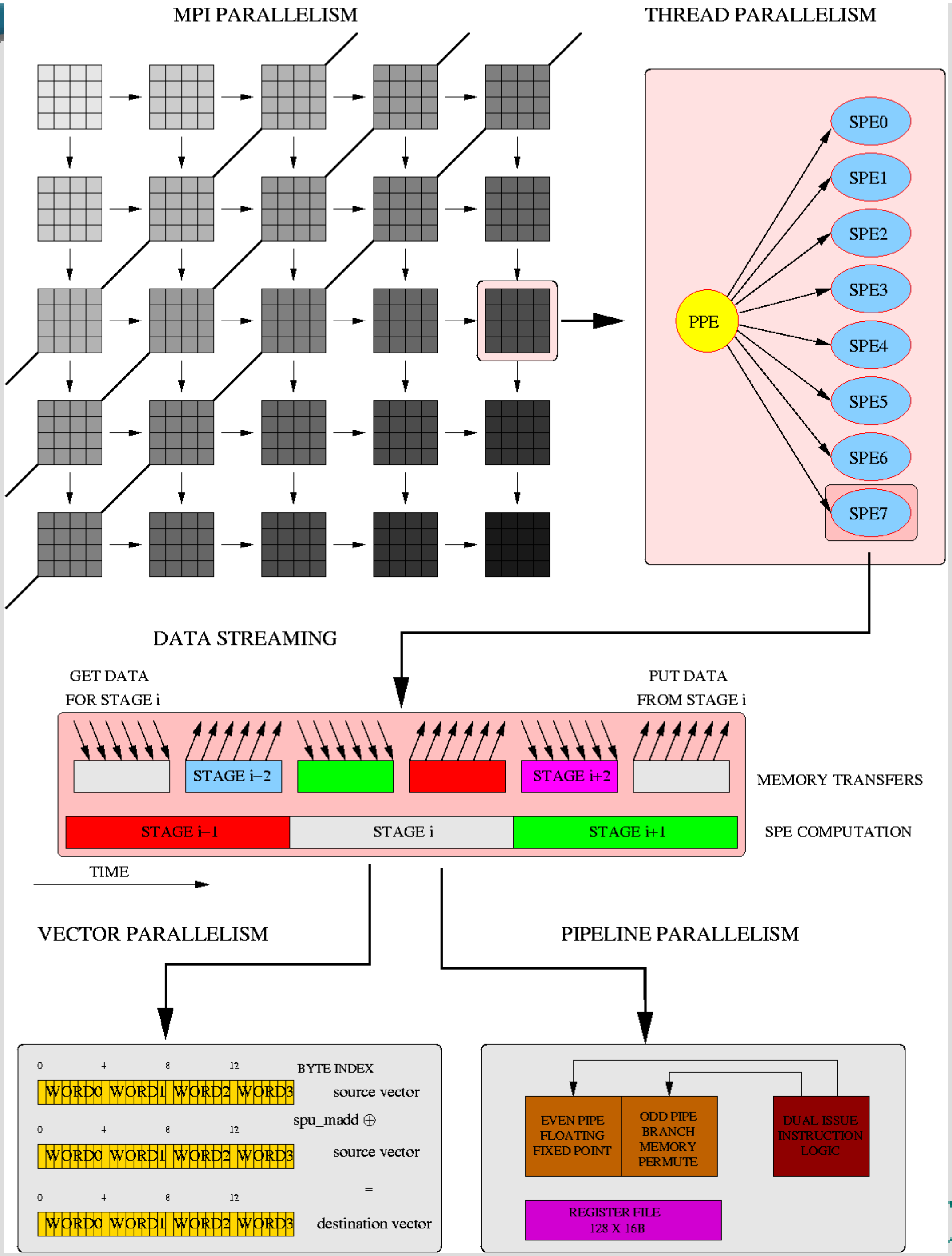
- Application kernel representative of the ASC workload
  - Considerable number of cycles on ASC machines
  - Relevant for a number of national security applications at PNNL
  - It solves 1-group time-independent discrete ordinates three-dimensional neutron transport problem

Sweep3D: data mapping and communication pattern

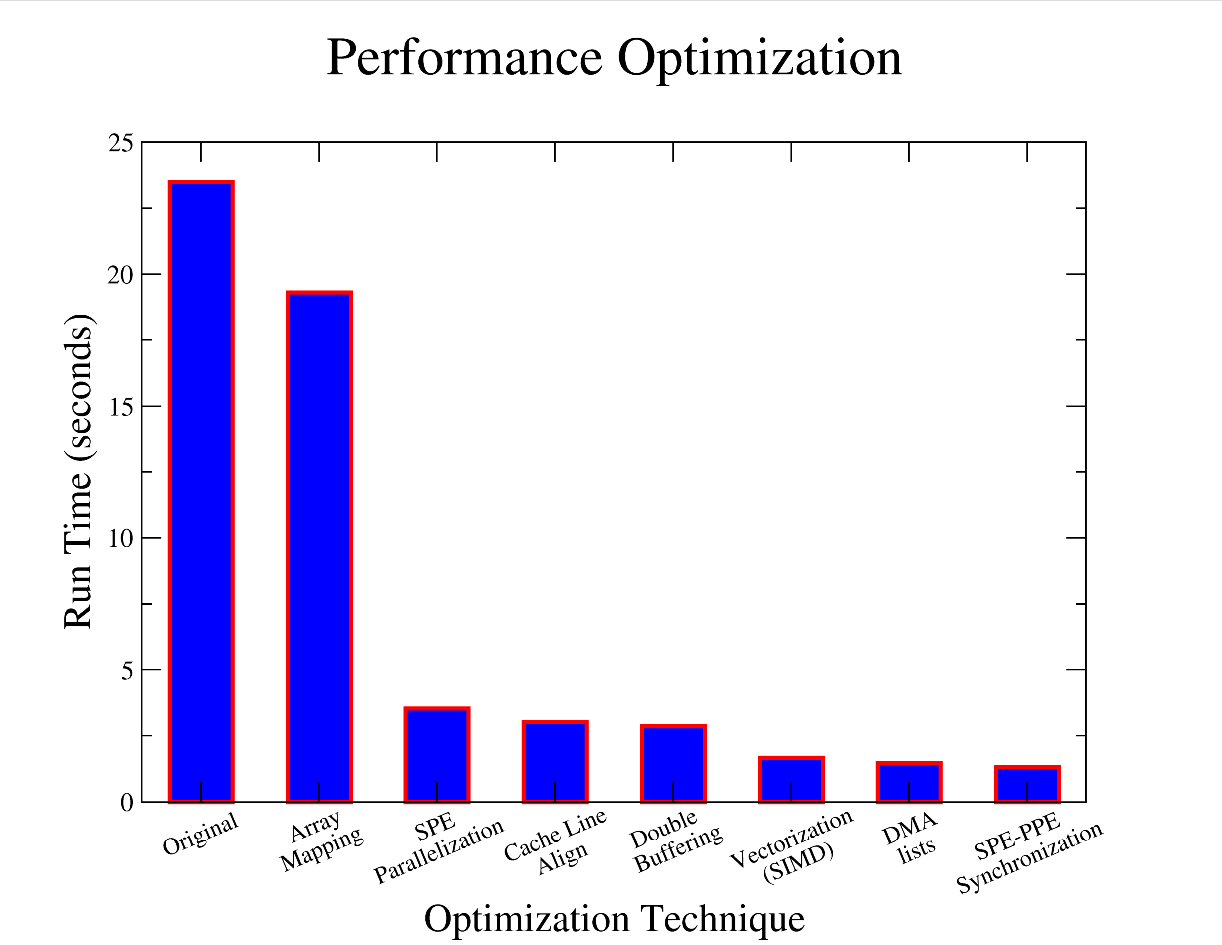


Parallelization Strategy

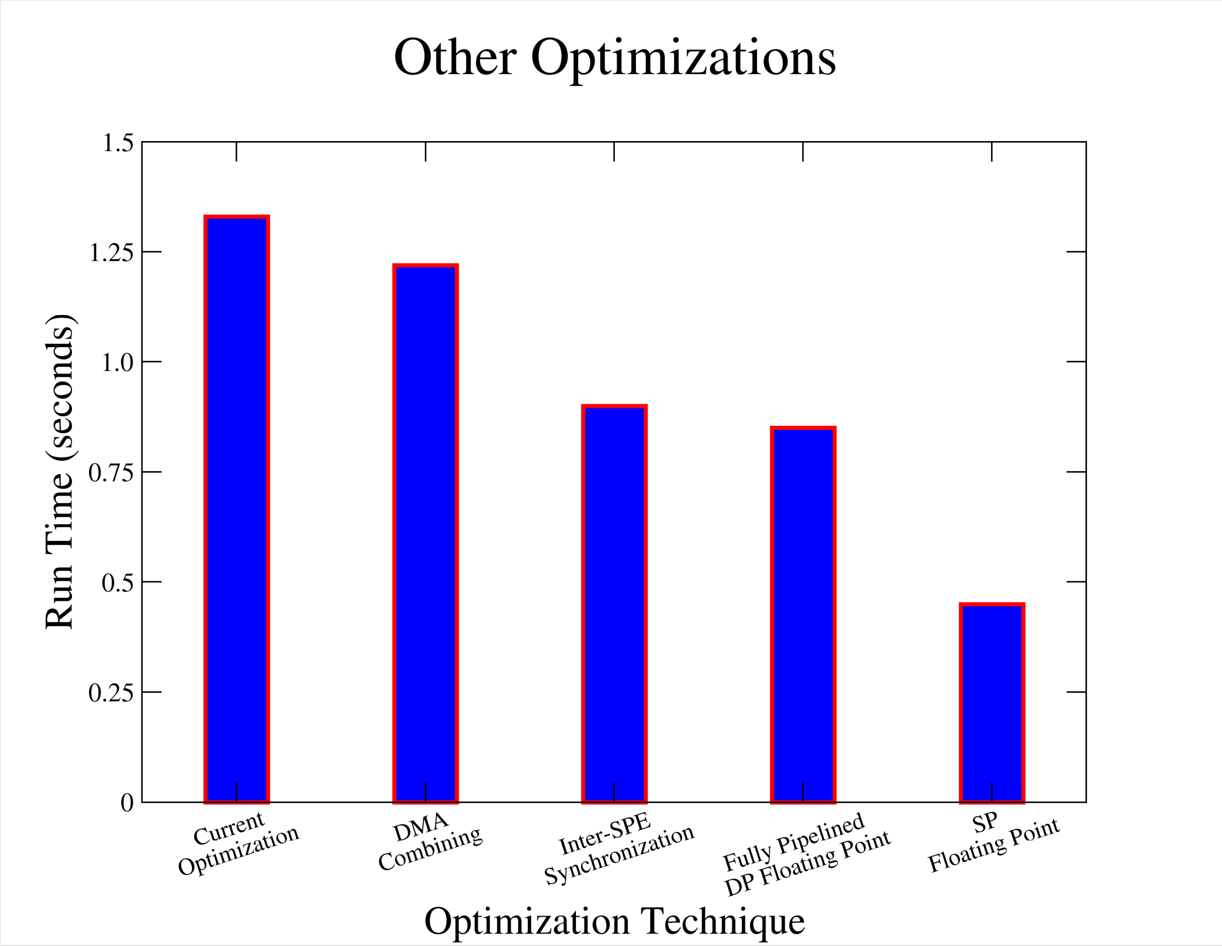
- Process level parallelism
  - We keep the existing MPI parallelization, to guarantee seamless migration path of existing software
- Thread-level parallelism
  - Take advantage of loop independency
- Data-streaming parallelism
  - Data orchestration algorithms
- Vector parallelism
  - To exploit vector units
- Pipeline parallelism
  - Even-odd pipe optimizations


MPI PARALLELISM
THREAD PARALLELISM
PPE
SPE0
SPE1
SPE2
SPE3
SPE4
SPE5
SPE6
SPE7
DATA STREAMING
GET DATA FOR STAGE i
PUT DATA FROM STAGE i
STAGE i−2
STAGE i+2
MEMORY TRANSFERS
STAGE i−1
STAGE i
STAGE i+1
SPE COMPUTATION
TIME
VECTOR PARALLELISM
PIPELINE PARALLELISM
0
4
8
12
BYTE INDEX
WORD0 WORD1 WORD2 WORD3
source vector
spu_madd ⊕
0
4
8
12
WORD0 WORD1 WORD2 WORD3
source vector
=
0
4
8
12
WORD0 WORD1 WORD2 WORD3
destination vector
EVEN PIPE FLOATING FIXED POINT
ODD PIPE BRANCH MEMORY PERMUTE
DUAL ISSUE INSTRUCTION LOGIC
REGISTER FILE 128 X 16B

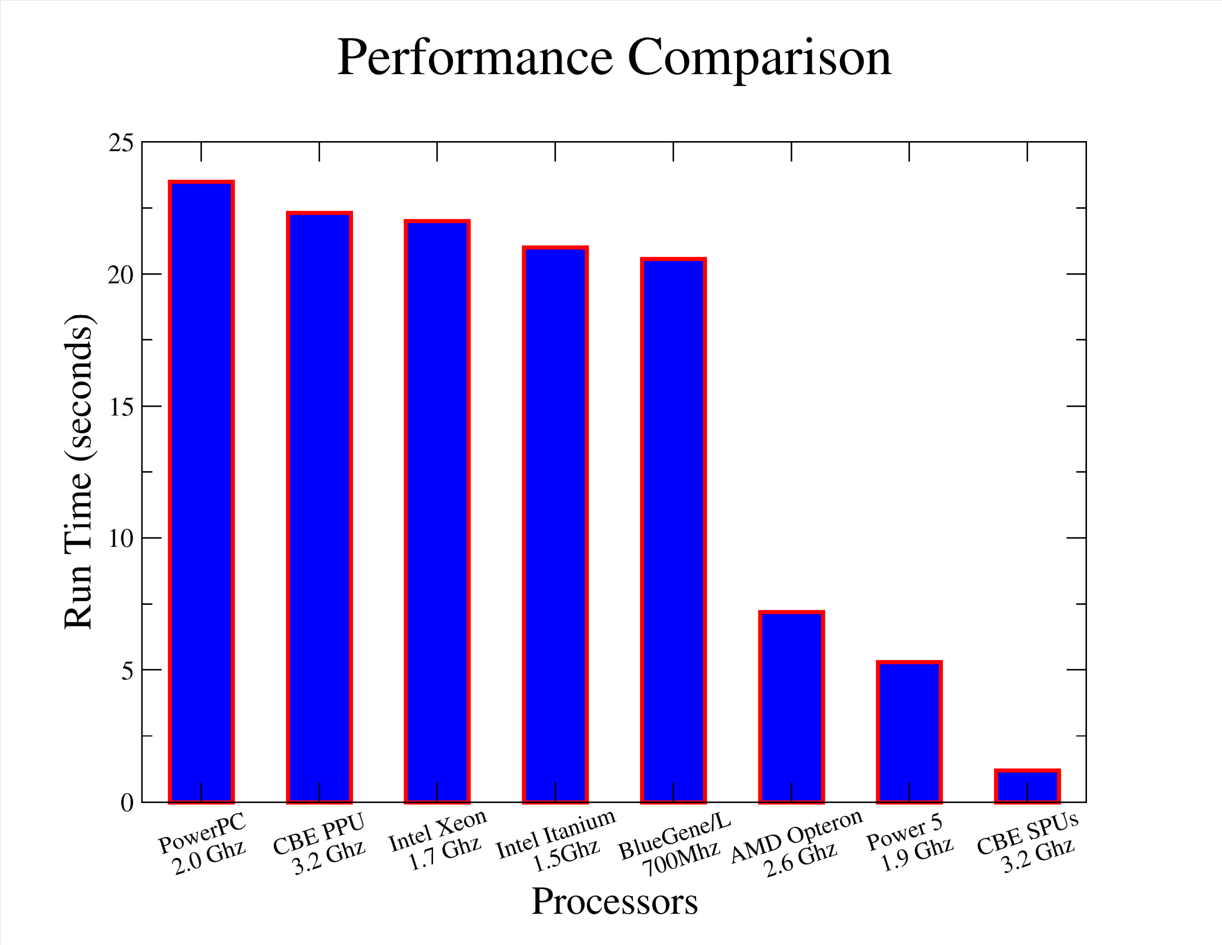
An arsenal of tools/techniques and optimizations



Work in progress



How does it compare with other processors?



Multicore surprises

- High sustained floating point performance
  - 64% in double precision (9 Gflops), 25% in single (50 Gflops)
  - Typical values of actual performance for Sweep3D are 5-10%
- Memory bound
  - The real problem, is data movement, not floating point performance
- Outstanding Power Efficiency
  - 2-4 times faster than BlueGene/L, the most power efficient computer at the moment (conservative estimate)

Conclusions

- Papers available at the following URLs
  - Cell Multiprocessor Interconnection Network: Built for Speed, *IEEE Micro, May/June 2006*
    - http://hpc.pnl.gov/people/fabrizio/ieeemicro-cell.pdf
  - Multicore Surprises: Lesson Learned from Optimizing Sweep3D on the Cell Broadband Engine, *Submitted for publication*
    - http://hpc.pnl.gov/people/fabrizio/sweep3d-cell.pdf