Chemical Transport Models on Accelerator Architectures

John C. Linford
Dr. Adrian Sandu

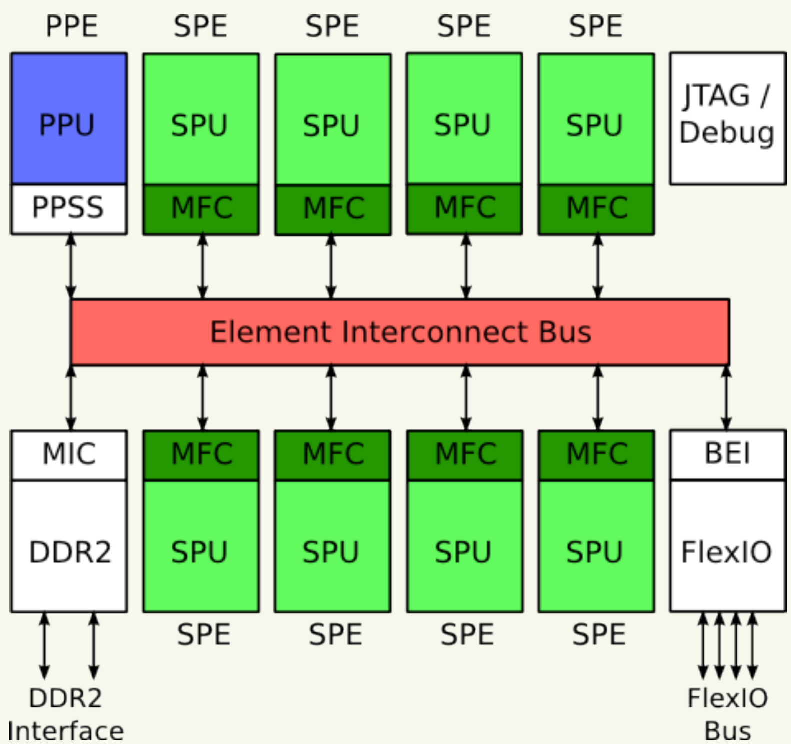
CSL

Virginia Tech

Presentation Outline

- Features of the Cell Broadband Engine Architecture
  - Multiple layers of heterogeneous parallelism
  - Cores operate on data in separate, distinct address spaces
- FIXEDGRID: A prototypical air quality model
  - Solves the upwind-biased advection/diffusion equations
- Chemical transport modeling on the CBEA
  - Two-dimensional
  - Three-dimensional

The Cell Broadband Engine Architecture



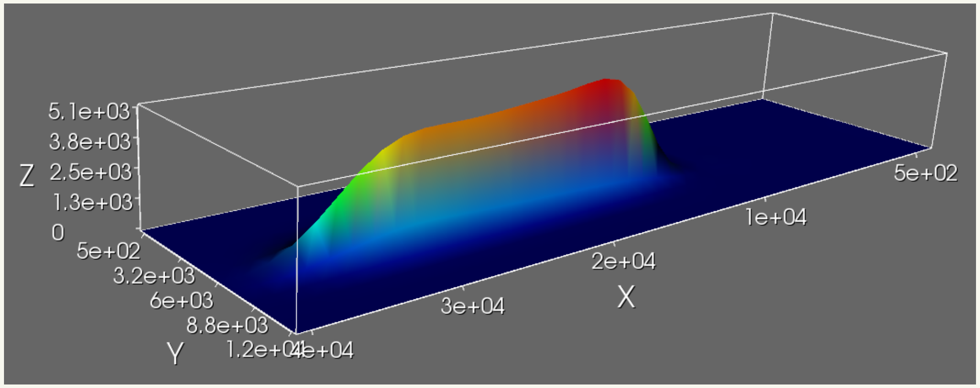
- 1 × Power Processing Elem.
  - 64-bit PowerPC + Vector/SIMD
- 8 × Synergistic Processing Elem.
  - 128-bit SIMD processor
  - 256KB local storage
  - Cannot access main memory directly
  - Uses DMA to copy data between main memory and local storage
- Memory Flow Controller
  - Asynchronous DMA between main-memory and local storage
- Element Interconnect Bus
  - Circuit-switched ring topology
  - 204.8GB/second peak bandwidth
- PowerXCell 8i: 102.4 gigaFLOPS
  - Double precision

Presentation Outline

- Features of the Cell Broadband Engine Architecture
  - Multiple layers of heterogeneous parallelism
  - Cores operate on data in separate, distinct address spaces
- FIXEDGRID: A prototypical air quality model
  - Solves the upwind-biased advection/diffusion equations
- Chemical transport modeling on the CBEA
  - Two-dimensional
  - Three-dimensional

FIXEDGRID: A prototypical atmospheric model for emerging multi-core research

- Experimental platform for modeling on multi-core
  - Isolate model components for detailed profiling
  - Test new concepts without rewriting a large production code
  - 100% standard C
- Simple domains, realistic processes
  - Properly-formatted real-world data can be used
- Has been ported to many platforms
  - Serial / SSE
  - OpenMP
  - CBEA
  - Sequoia
  - NVIDIA CUDA



Fixedgrid uses finite differences to discretize the upwind-biased transport-balance equations

Transport

$$\frac{\partial c}{\partial t} = -u\frac{\partial c}{\partial x} - v\frac{\partial c}{\partial y} - w\frac{\partial c}{\partial z} + \frac{1}{\rho}\frac{\partial}{\partial x}\left(\rho K \frac{\partial c}{\partial x}\right) + \frac{1}{\rho}\frac{\partial}{\partial y}\left(\rho K \frac{\partial c}{\partial y}\right) + \frac{1}{\rho}\frac{\partial}{\partial z}\left(\rho K \frac{\partial c}{\partial z}\right)$$

Advection

$$-u\frac{\partial c}{\partial x} = \begin{cases} u_i(-c_{i-2} + 6c_{i-1} - 3c_i - 2c_{i+1})/(6\Delta x), & u_i \geq 0 \\ u_i(2c_{i-1} + 3c_i - 6c_{i+1} + c_{i+2})/(6\Delta x), & u_i < 0 \end{cases}$$

Diffusion

$$\frac{1}{\rho}\frac{\partial}{\partial x}\left(\rho K \frac{\partial c}{\partial x}\right) = \frac{(\rho_{i+1}K_{i+1} + \rho_i K_i)(c_{i+1} - c_i) - (\rho_i K_i + \rho_{i-1}K_{i-1})(c_i - c_{i-1})}{2\rho_i \Delta x^2}$$

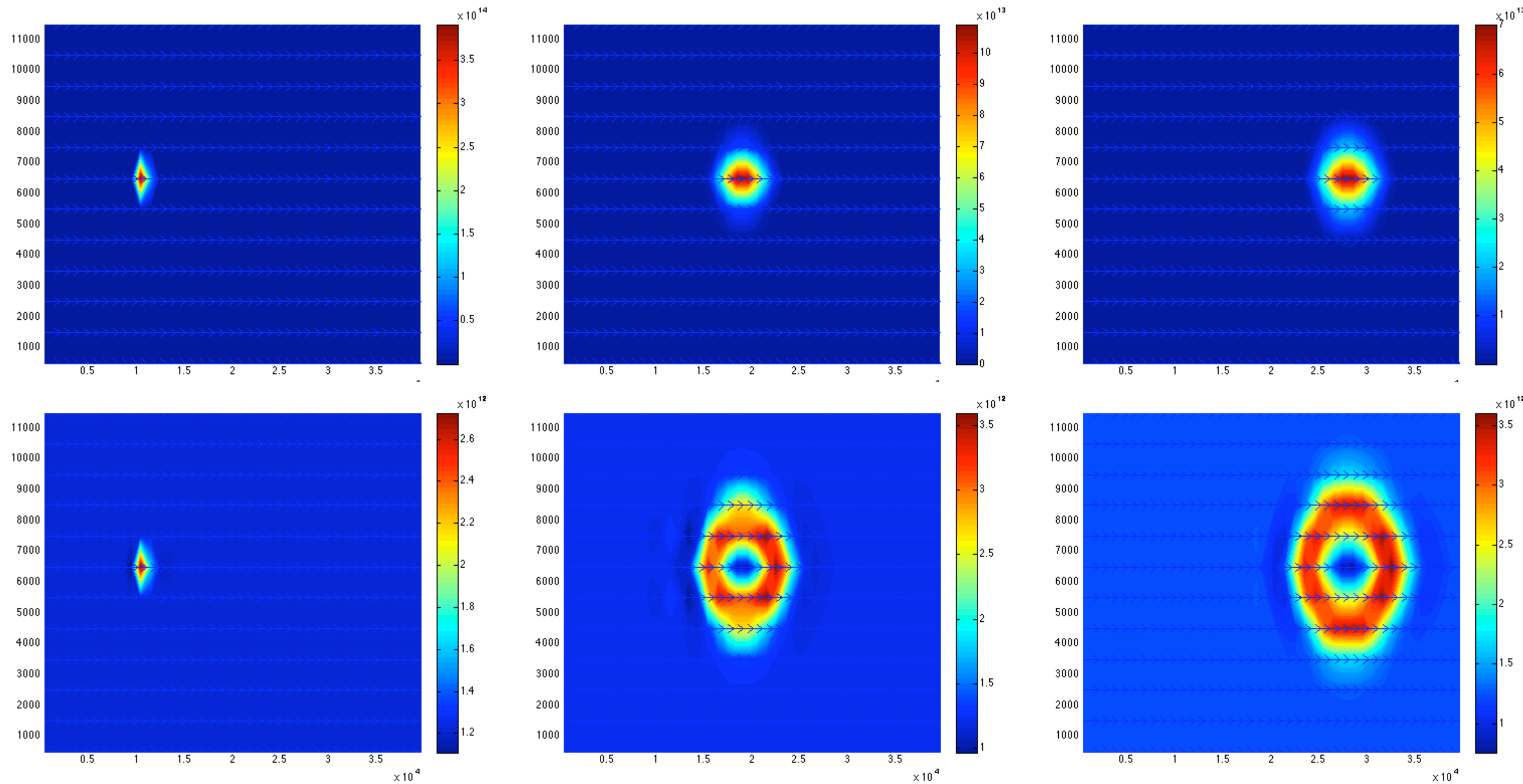
Convection

$$-w\frac{\partial c}{\partial z} = \begin{cases} -w\dfrac{c_k - c_{k-1}}{z_k - z_{k-1}}, & w_k \geq 0 \\ -w\dfrac{c_{k+1} - c_k}{z_{k+1} - z_k}, & w_k < 0 \end{cases}$$

A. Sandu, D.N. Daescu, G.R. Carmichael, and T. Chai. *Adjoint Sensitivity Analysis of Regional Air Quality Models*. J. Comp. Phys., Vol. 204, p. 222-252, 2005.

O_3 (top) and NO_2 (bottom) from FIXEDGRID

t = 0 minutes | t = 30 minutes | t = 60 minutes

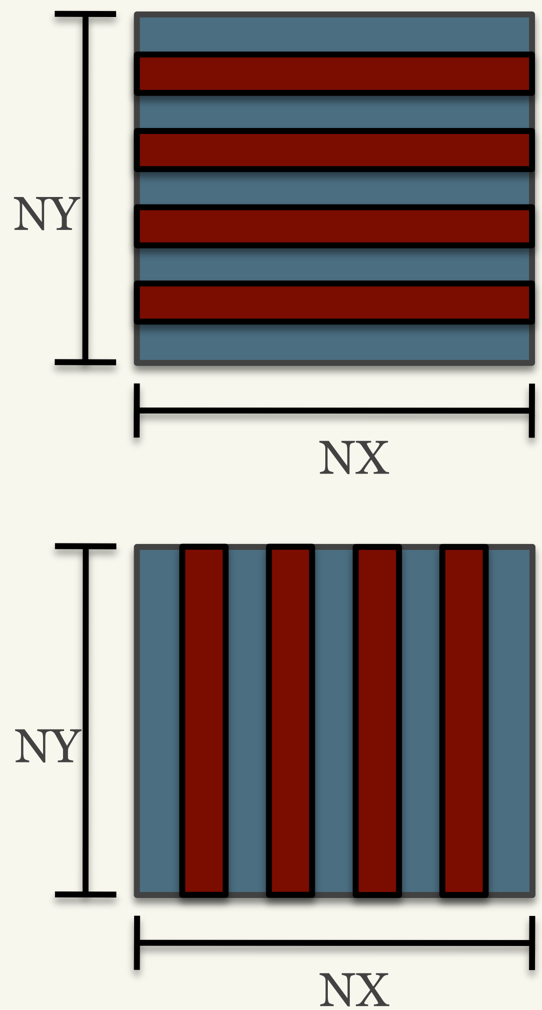


Presentation Outline

- Features of the Cell Broadband Engine Architecture
  - Multiple layers of heterogeneous parallelism
  - Cores operate on data in separate, distinct address spaces
- FIXEDGRID: A prototypical air quality model
  - Solves the upwind-biased advection/diffusion equations
- Chemical transport modeling on the CBEA
  - Two-dimensional
  - Three-dimensional

Operator splitting parallelizes advection / diffusion kernel in rows and columns

- Compared function offload approaches for 2D transport on the CBEA and shared-memory multi-core.
- Demonstrated an effective method for scalable random access to matrix column data using DMA lists.
- Identified potential compiler research areas.



John C. Linford and Adrian Sandu. *Optimizing large scale chemical transport models for multicore platforms*. In Proceedings of the 2008 Spring Simulation Multiconference (SpringSim '08), Ottawa, Canada, April 2008.

Three versions of the function offload approach were compared

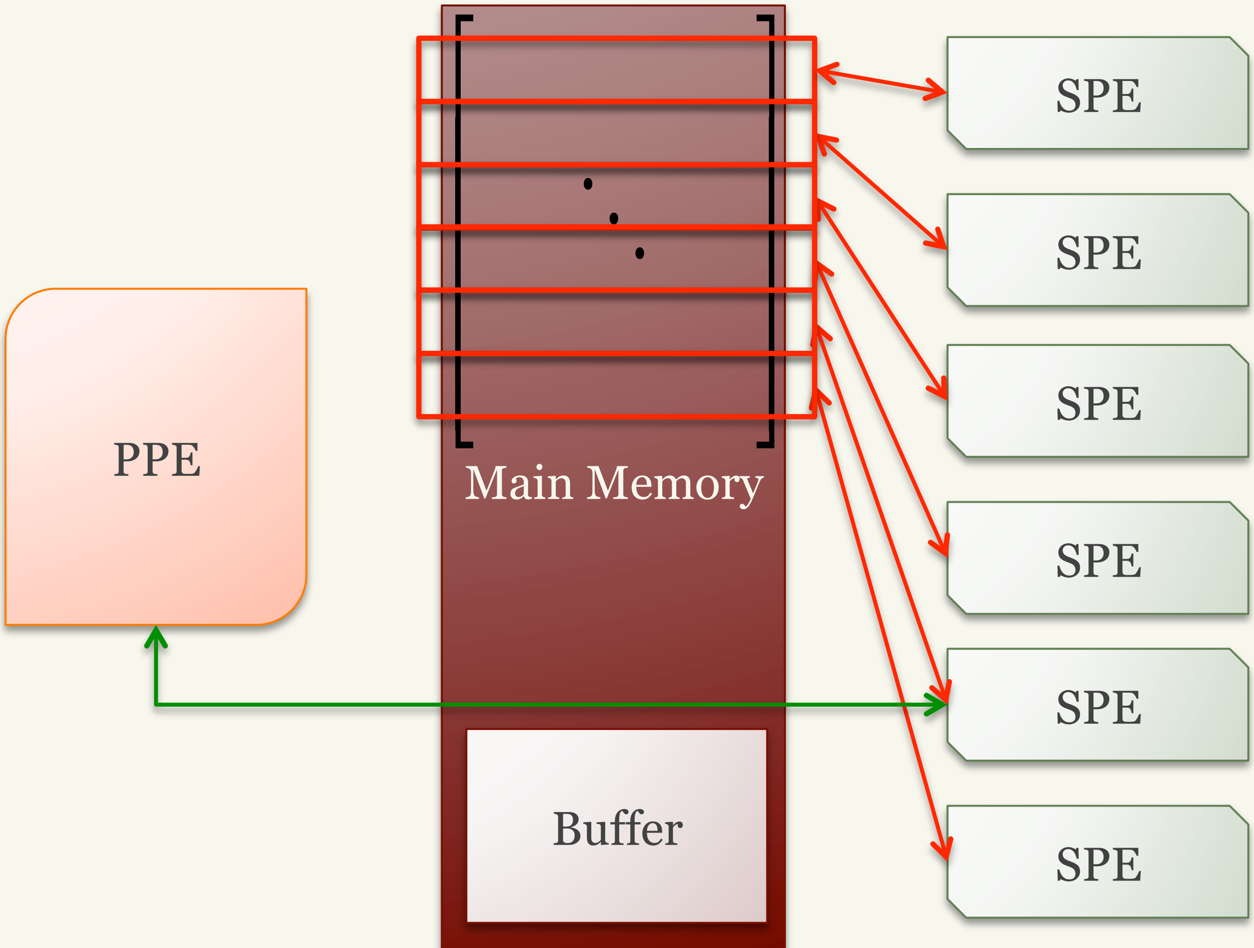
Ver. 1: Naïve function offload

- Offloaded the main computational cores to SPEs
  - `discretize()`
  - `advec_diff()`
- Implemented generic data types for passing arguments to the SPUs
- Implemented tiny 32-bit communication library for PPU / SPU communication
  - Based on mailbox registers
  - Send/receive
  - Broadcast
  - Synchronize

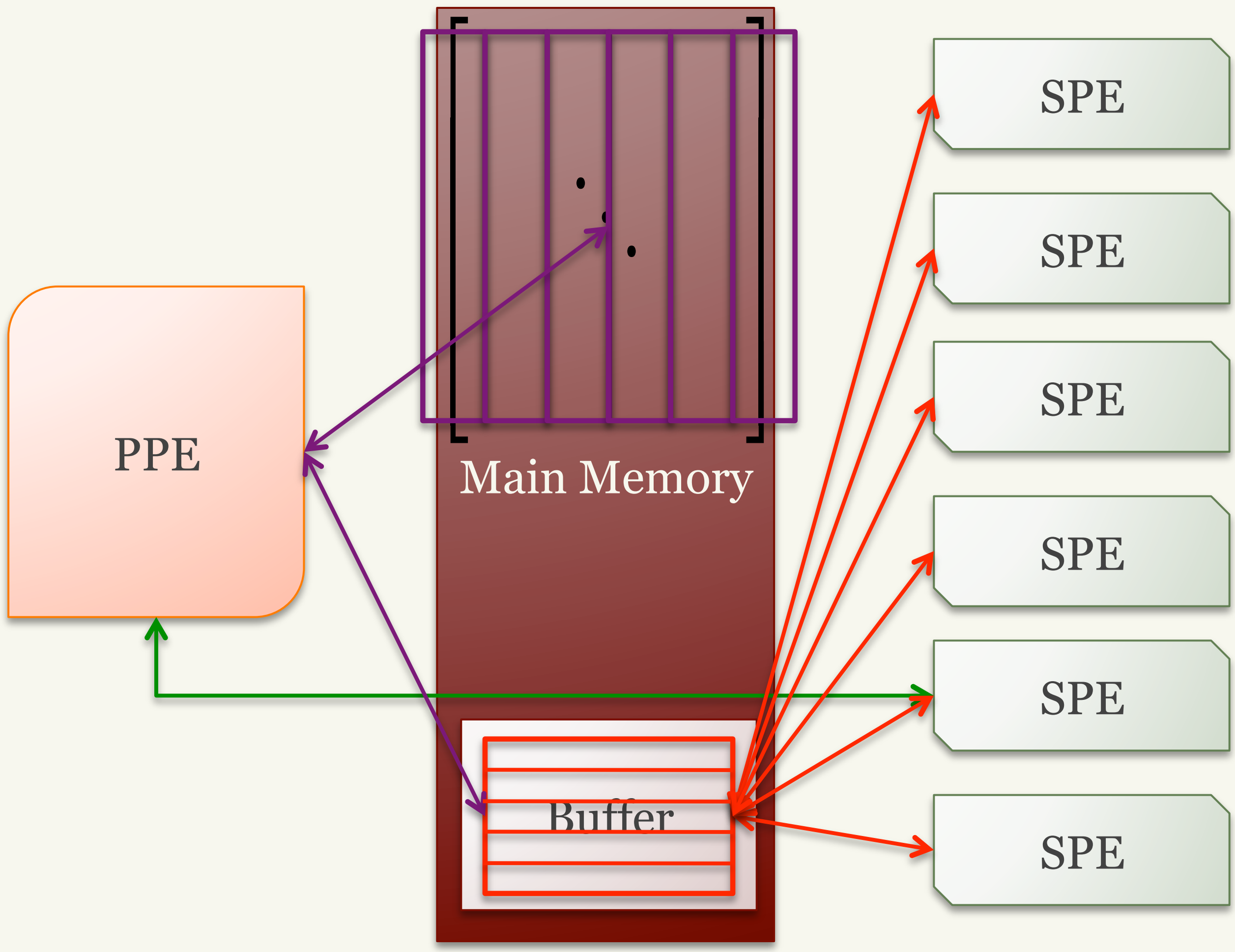
Ver. 2: SPE optimized offload

- Double-buffered DMA
  - 23% runtime reduction
- Vectorized functions with SPU intrinsics and compiler flags
  - 18% runtime reduction
- Unrolled loops and used branch prediction intrinsics
  - 8% runtime reduction
- **~2**× faster. An *optimized compiler* should be developed

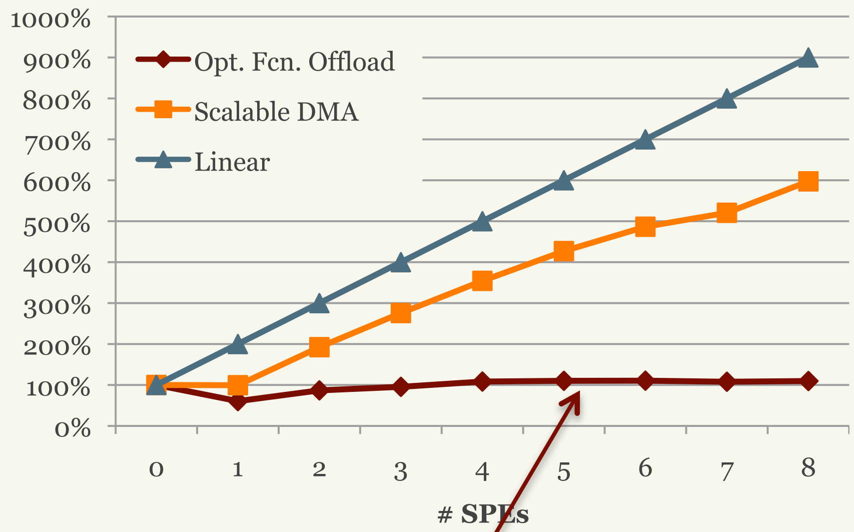
The SPEs copy matrix rows directly from main memory to LS via DMA



The PPU buffers transposed columns in main memory so SPEs can DMA to LS

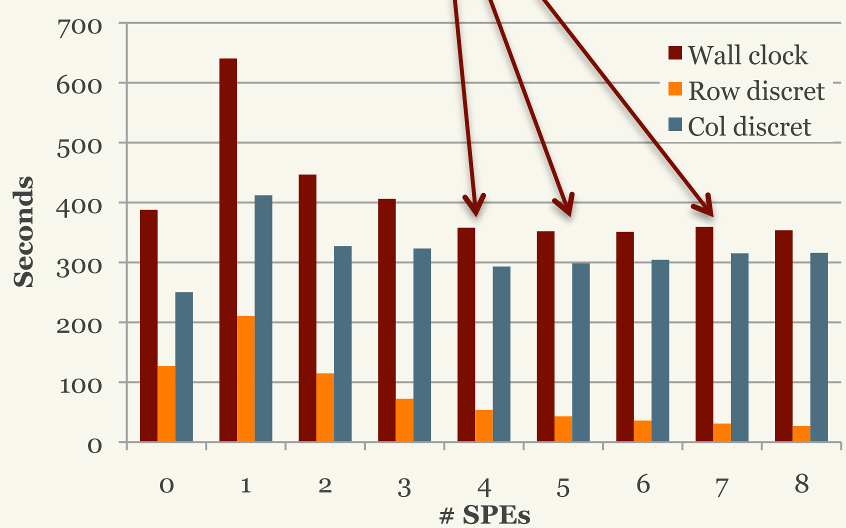


Scalability is severely limited by PPU matrix transpose bottleneck



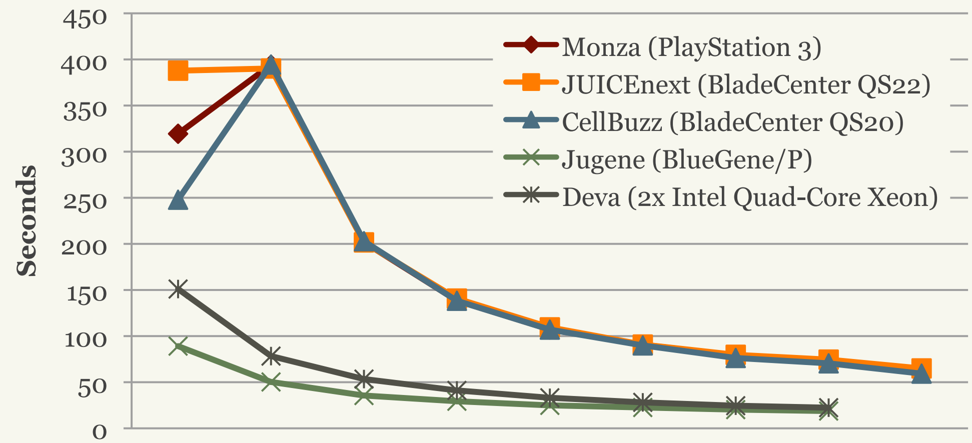
Scalability is limited by PPU bottleneck

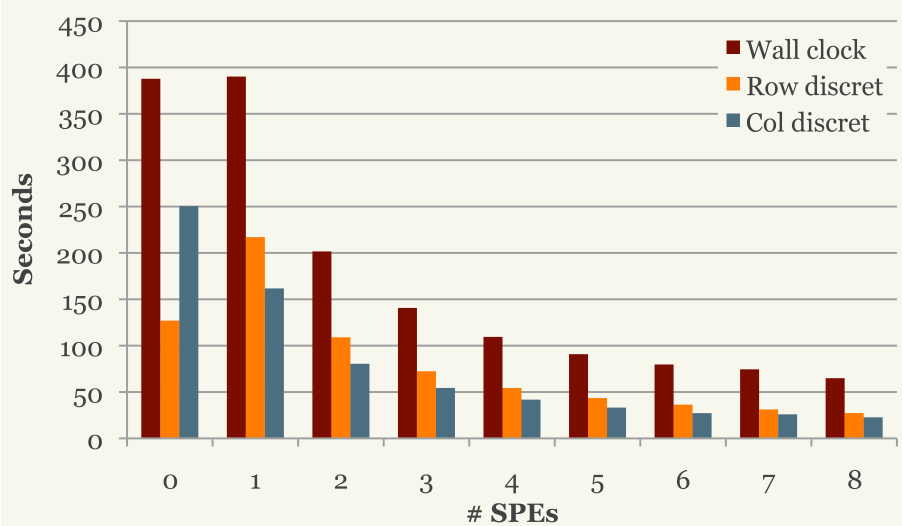
Wall clock bounded by column discretization

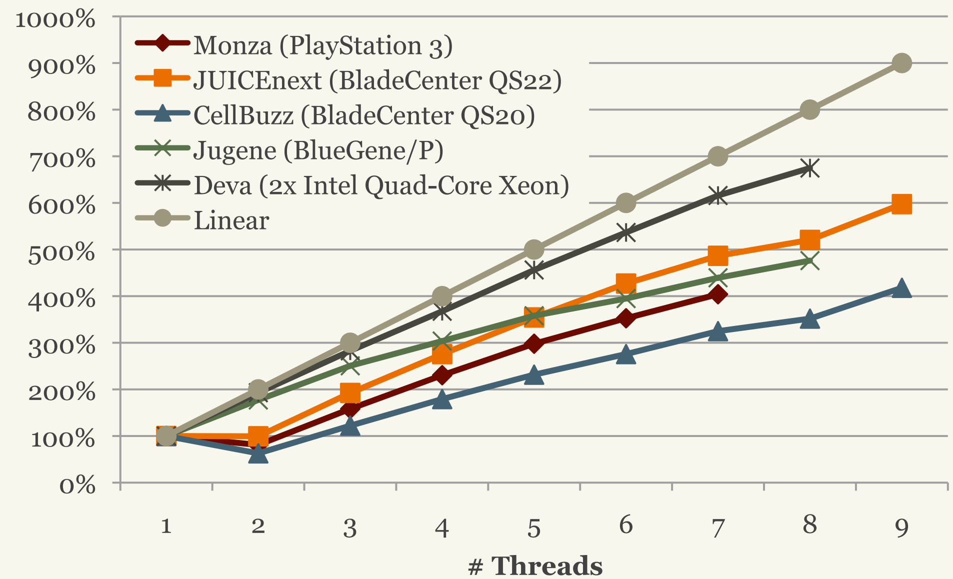


Version 3: Use DMA lists for column data transfer and alleviate scalability bottleneck

- Use DMA lists to transfer incontiguous data
- 16 byte minimum transfer size, so transfer and process two columns at once
- Columns are interleaved in SPE local storage and must be separated before processing

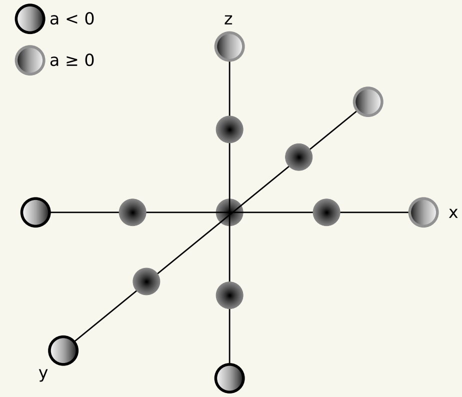


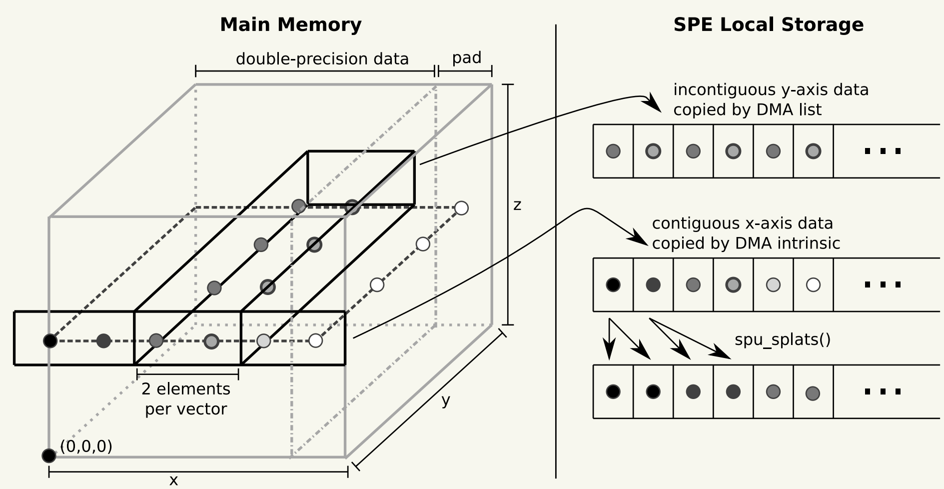




Vector Stream Processing in 3D transport uses every layer of heterogeneous parallelism

- Combined the stream processing features of CBEA with the SPE SIMD ISA
- Streams are formed by combining DMA lists with a triple-buffering scheme.
  - Asynchronous comm. enables simultaneous read, write, and compute on a single SPE
- Used MFC intrinsics and the proxy command queue to avoid mailbox registers and improve throughput

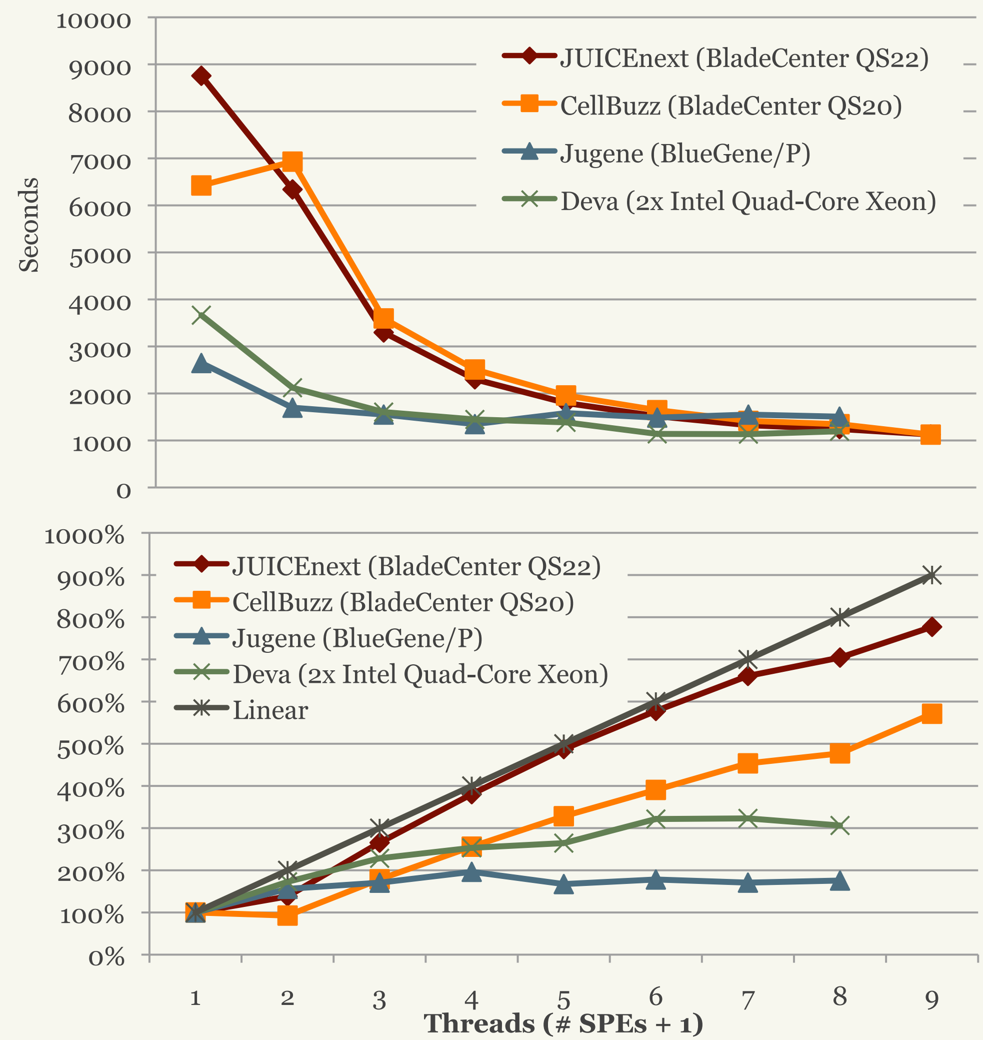




- John C. Linford and Adrian Sandu. *Vector stream processing for effective application of heterogeneous parallelism*. 24th Annual ACM Symposium on Applied Computing (SAC'09), Honolulu, HI, March 8–12 2009.

The vector stream processing approach achieves the best performance

```
/* Start buffer 0 transfer */
fetch_x_buffer(0, 0);

/* Start buffer 1 transfer */
fetch_x_buffer(1, NX_ALIGNED_SIZE);

/* Process buffer 0 */
transport_buffer(0, size, dt);

/* Loop over rows in this block */
for(i=0; i<block-2; i++)
{
  w = i % 3;
  p = (i+1) % 3;
  f = (i+2) % 3;

  /* Write buffer w back to main memory (nonblocking) */
  write_x_buffer(w, i*NX_ALIGNED_SIZE);

  /* Start buffer f transfer (nonblocking) */
  fetch_x_buffer(f, (i+2)*NX_ALIGNED_SIZE);

  /* Process buffer p */
  transport_buffer(p, size, dt);
}

/* Discretize final row */
w = i % 3;
p = (i+1) % 3;

/* Write buffer w back to main memory (nonblocking) */
write_x_buffer(w, i*NX_ALIGNED_SIZE);

/* Process buffer p */
transport_buffer(p, size, dt);

/* Write buffer p back to main memory (nonblocking) */
write_x_buffer(p, (i+1)*NX_ALIGNED_SIZE);

/* Make sure DMA is complete before we exit */
mfc_write_tag_mask( (1<<w) | (1<<p) );
spu_mfcstat(MFC_TAG_UPDATE_ALL);
```



?