

Exploration of multiple layers of data parallelism for clusters of asymmetric multi-core processors

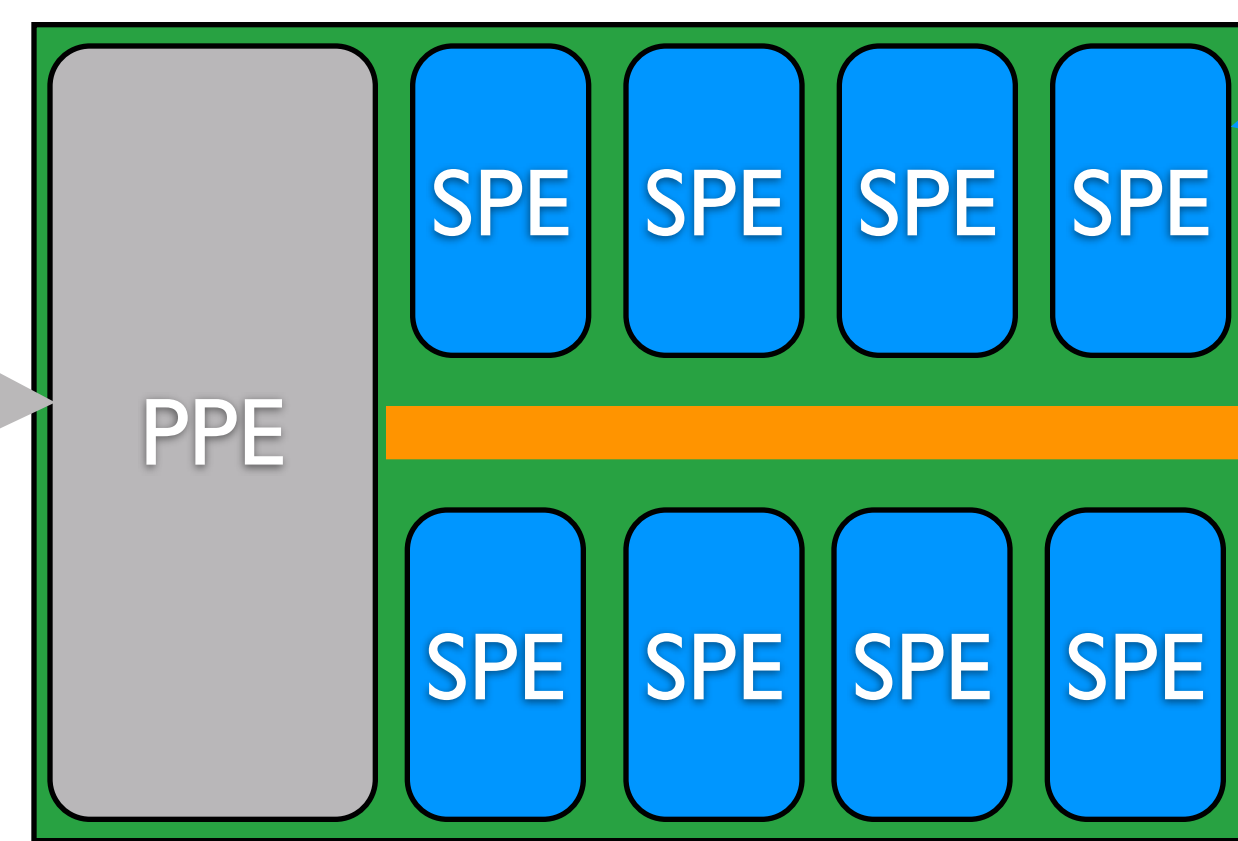
Benjamin Rose, Dimitris Nikolopoulos, David Lowenthal
 {bar234,dsn}@cs.vt.edu, dkl@cs.arizona.edu

An Asymmetric Multi-Core Processor (Cell Broadband Engine)



Traditional Core

- Larger general purpose core
- Uses an established ISA
- Similar to most of today's processors
- Runs the operating system
- Multiple cache layers that are managed by hardware
- In the Cell Broadband Engine, this is the PPE (Power Processing Element)



Accelerator Cores

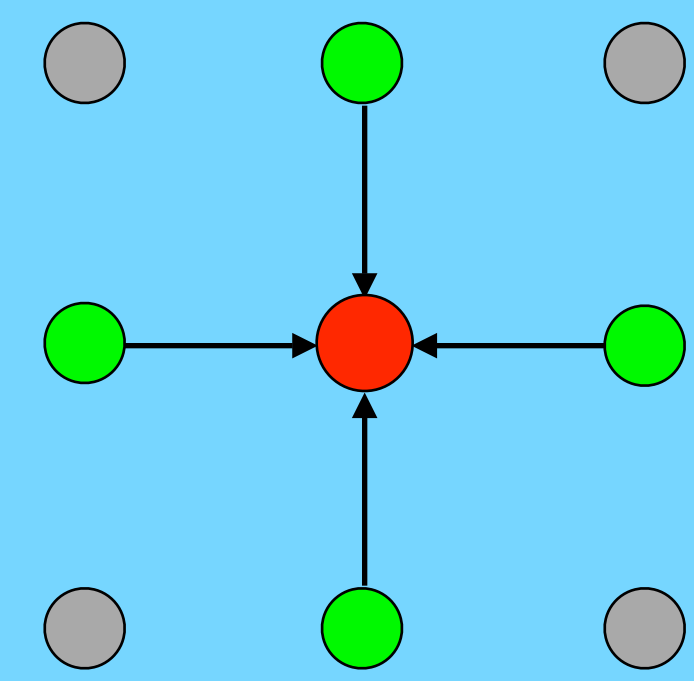
- Smaller computational cores
- Different ISA than the host core
- Programmer managed cache
- Support SIMD operations
- In the Cell Broadband Engine, this is the SPE (Synergistic Processing Element)

Interconnect

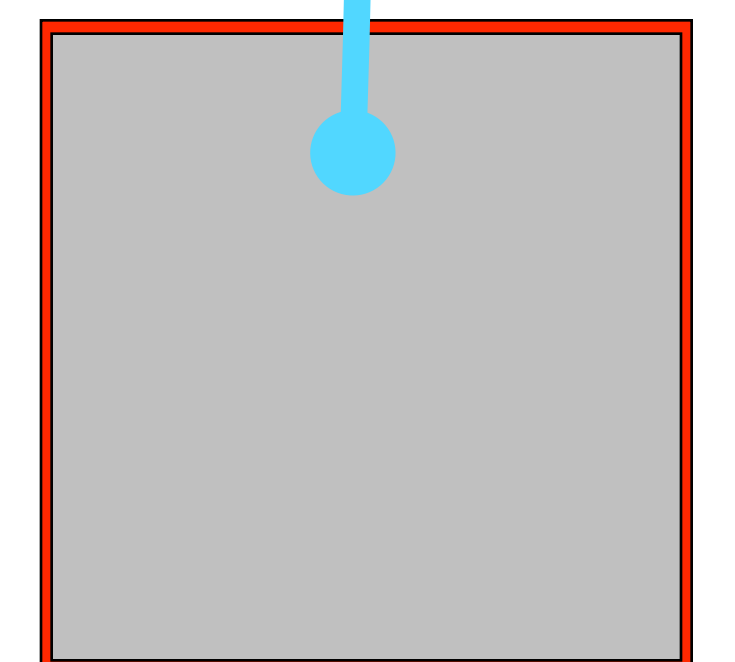
- Very fast multi-lane bus
- Connects all elements in the chip to each other and to external components (memory, I/O, etc.)

Jacobi Stencil Code

- Data set is a grid of points.
- During each iteration, each point is updated to the average of all four of its neighbors.
- Iterates until the residual becomes very small or a set number of iterations



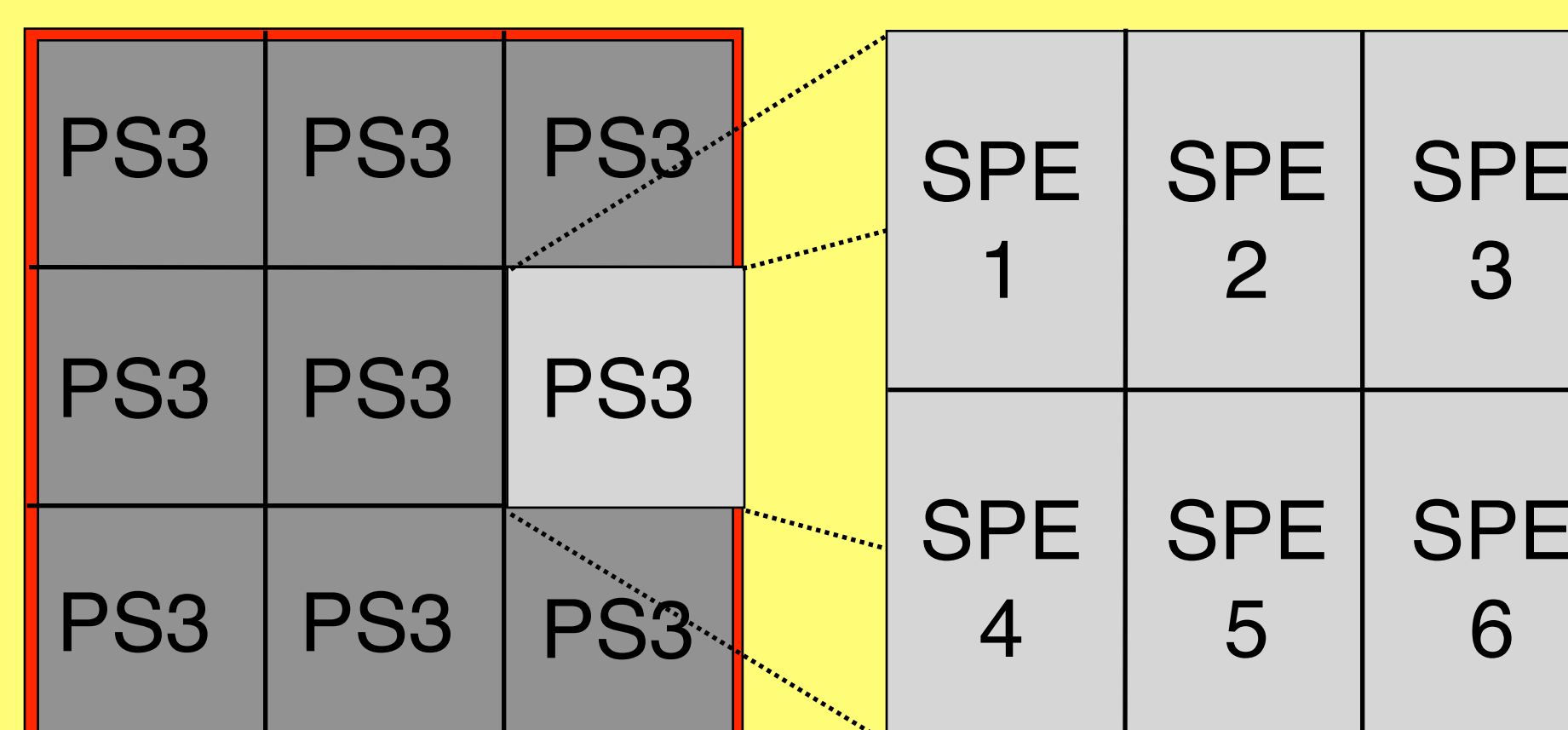
- Data set initialized to all zeros
- Border values (in red) initialized to one
- Two copies of the data set: one contains the input value and the other contains the output
- Roles of the two sets switch between iterations



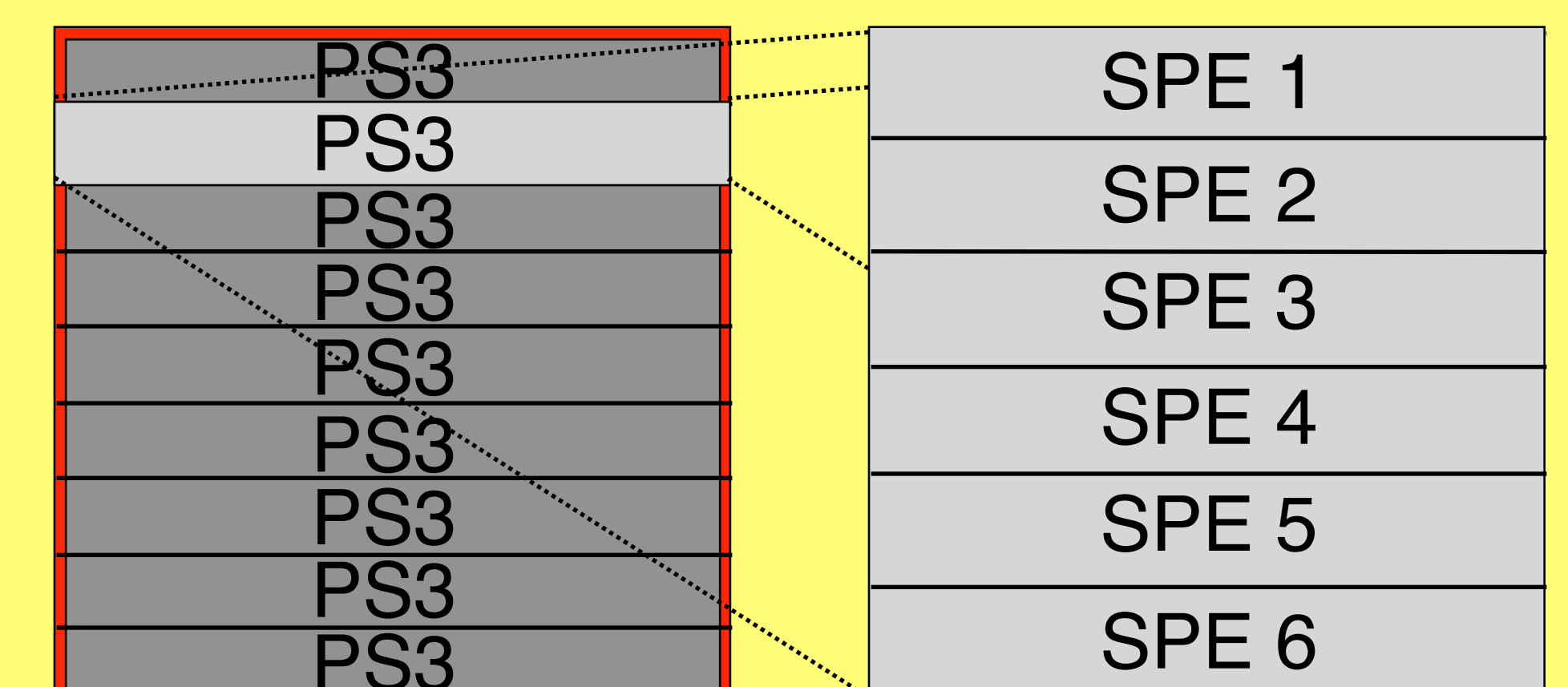
Data Distributions

We use a cluster of Sony PlayStation 3s for our data distribution exploration. Each PlayStation 3 has a Cell Broadband Engine with 6 available SPE accelerators. Here we layout the different distributions using 9 PlayStation 3s.

There are 4 possible combinations: BS-BS, BS-BB, BB-BS, BB-BB (The first two letters represent the cluster data distribution, the second two represent the SPE data distribution)



An example of BB-BB: BLOCK,BLOCK between the PS3s and BLOCK,BLOCK between the SPEs in each PS3



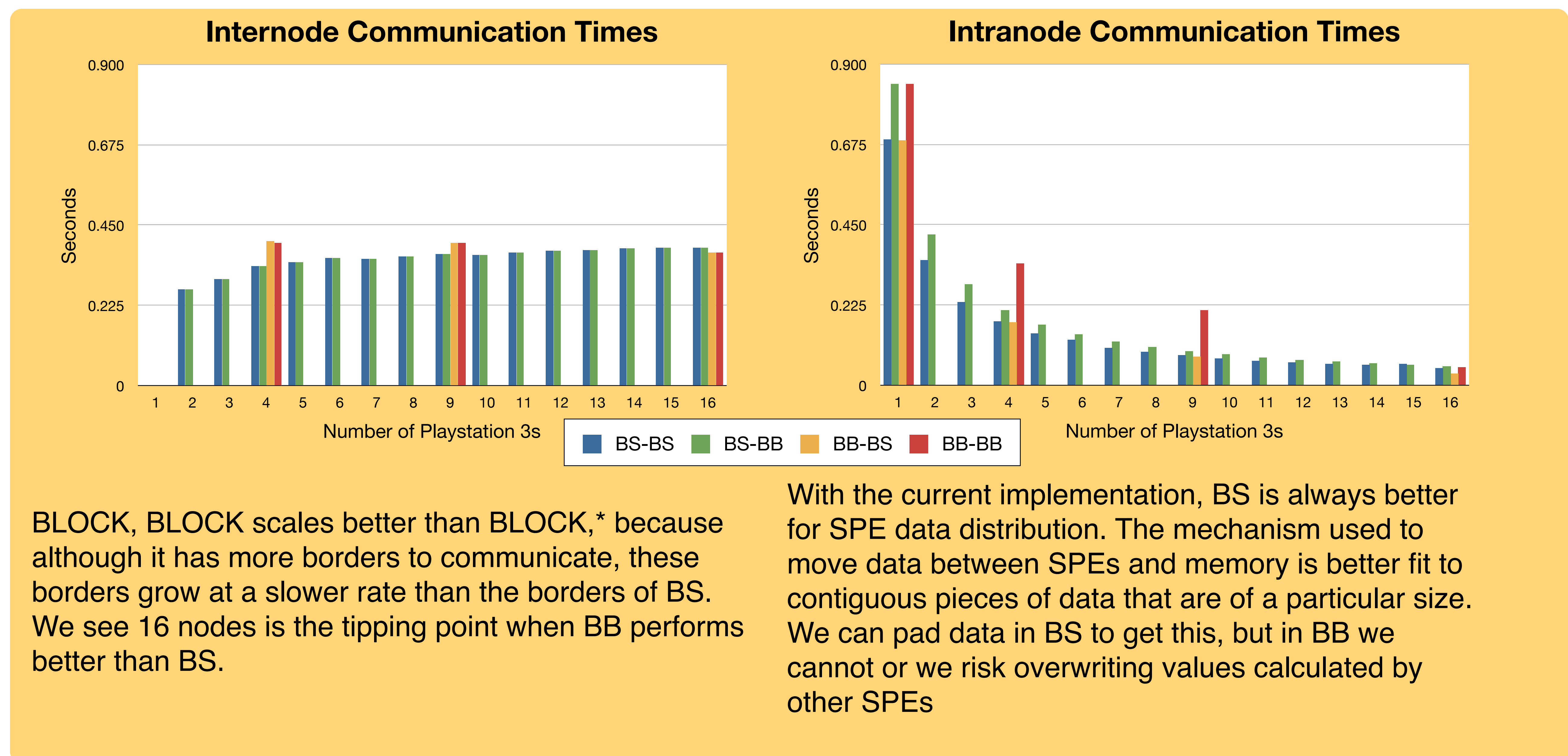
An example of BS-BS: BLOCK,* between the PS3s and BLOCK,* between the SPEs in each PS3

Observations

Since computation time is approximately constant across all configurations, the communication is the focus of our experiments.

As the results show, communication schemes produce different results at different data sizes. Eventually we'd like to perform static/dynamic analysis of programs to determine the best data distribution combination.

This is still very much a work in progress. Further work needs to be done on the SPE BB implementation to remove the intranode communication spikes seen in the BB-BB trial.



BLOCK, BLOCK scales better than BLOCK,* because although it has more borders to communicate, these borders grow at a slower rate than the borders of BS. We see 16 nodes is the tipping point when BB performs better than BS.

With the current implementation, BS is always better for SPE data distribution. The mechanism used to move data between SPEs and memory is better fit to contiguous pieces of data that are of a particular size. We can pad data in BS to get this, but in BB we cannot or we risk overwriting values calculated by other SPEs