Sony Computer Entertainment US Research and Development
http://www.research.scea.com

Beyond the GFLOPS

Dominic Mallinson

Vice President, US R & D
Sony Computer Entertainment Inc.

"Why not go out on a limb?
That's where the fruit is."

(Will Rogers, cowboy, actor, philanthropist)

The Cell Broadband Engine (Cell/B.E.) Processor

The Cell/B.E. Processor

- Leading the industry in heterogeneous multi-core
- 200+ GFLOPS high performance computing
- But what lies *beyond the GFLOPS* statistics ?
  - Why does an application need Cell/B.E.'s power ?
  - How can we make Cell/B.E.'s performance more accessible ?
  - What part do you and the Cell/B.E.'s software community play ?

Why does SCE need Cell/B.E. performance ?

Games and Virtual World

- GBytes of data streaming through the CPU in real-time
- 100s of animating 3D characters on screen
- True HD 3D Graphics with millions of vertices visible
- Complex Artificial Intelligence techniques
- Physical Simulation, cloth, fluids, soft and rigid bodies
- Real-time spatial audio processing and encode
- Millions of simultaneous users
- Potential for client and server to use Cell/B.E. processor

Demo Time

Media Processing

- Blu-ray movie playback
  - 1080p video decode in AVC, VC1 or MPEG2
  - Simultaneous 480p “picture in picture” decode
  - 7.1 multi-channel audio decode and mixing
  - … and a Java™ VM
- Remote Play function of PLAYSTATION®3 (PS3™)
  - Realtime AV encoding and streaming to a PlayStation®Portable
- Multi-person AV Chat
  - 1 encode plus up to 5 decodes, AEC noise reduction

Folding@home™ on PS3

- A distributed computing project from Stanford University
  - Research into protein misfolding to help understand and find treatments for diseases such as Alzheimer's and cancer.
- PS3 Client launched in March 2007
  - Over 250,000 unique PS3 users in the first month
- 488 TFLOPS (Stanford metrics from June 14th 2007)
  - 26,961 Active Cell/B.E. CPUs
- More than doubled previous PC/GPU contributions
- DEMO

Accessing the power of Cell/B.E.

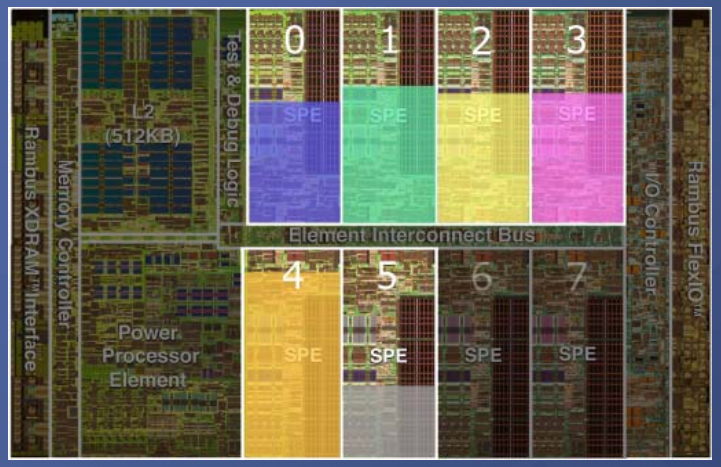
Accessing the power of Cell/B.E.

- The Cell/B.E. is designed for performance
  - Maximum performance requires complex software
  - The upper quartile of engineers already achieve it
  - The lower quartile *currently* cannot
- Research and Industry must bridge this gap
  - Many programming models are emerging
- How does SCE tackle this problem ?

SCE's SPURS Environment

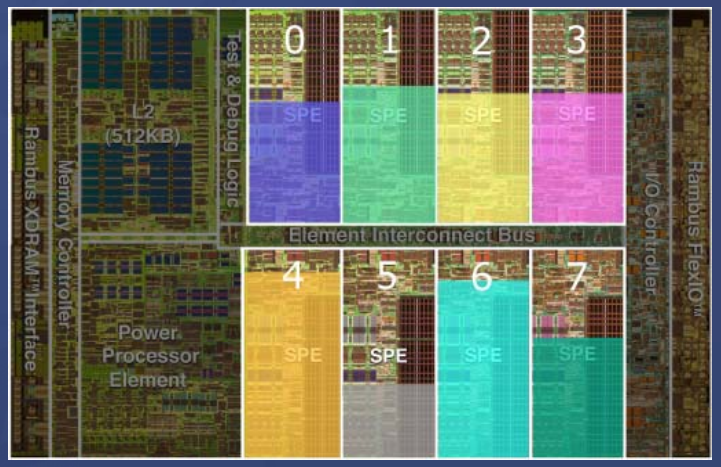
- A flexible, cooperative SPE management layer
- SPE-centric scheduling (minimal PPU overhead)
- Low or zero context switch overhead
- Application control for scheduling priorities
- Supports sharing SPE with 3rd party middleware
- Built on top of OS SPE Threads
- Policy manager allows multiple models

Duck Demo SPE Usage



Old Code – no machine vision – 6 SPEs

- SPE0 – Surface water physics
- SPE1 – Splash physics
- SPE2 – Boat 1 physics
- SPE3 – Boat 2 physics
- SPE4 – Collision physics
- SPE5 – Graphics



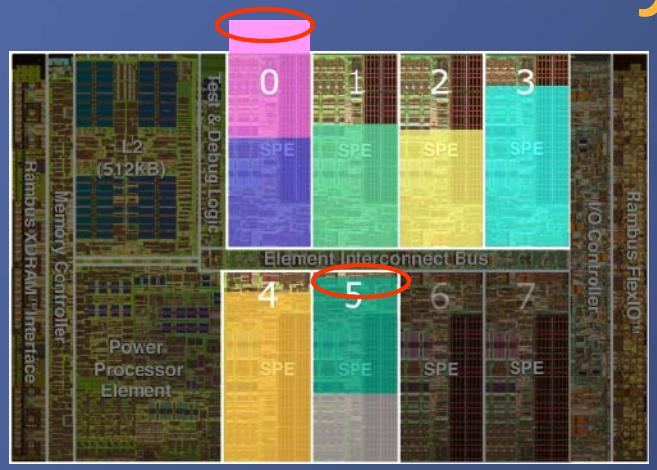
Old Code - machine vision – 8 SPEs

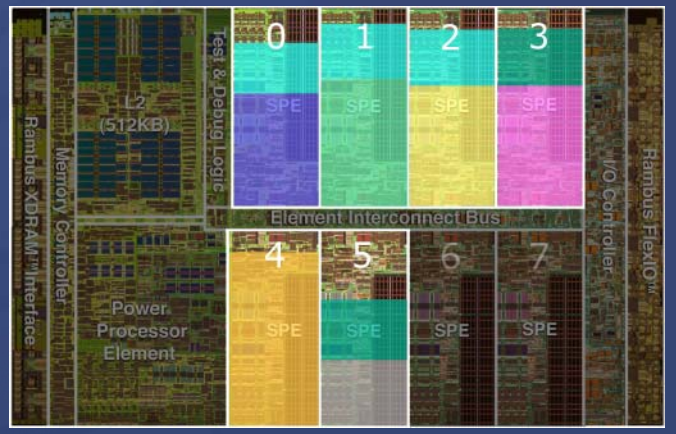
- SPE0-SPE5 UNCHANGED

Added machine vision, particle water

- SPE6 – Particle water physics
- SPE7 – Machine vision

Goal: Everything on 6 SPEs





Naïve use of SPURS

- Just try to move work around
- Water + Boat 2 is over time
- Graphics + Machine vision
  - Fits but no room to flex

Refactor with SPURS

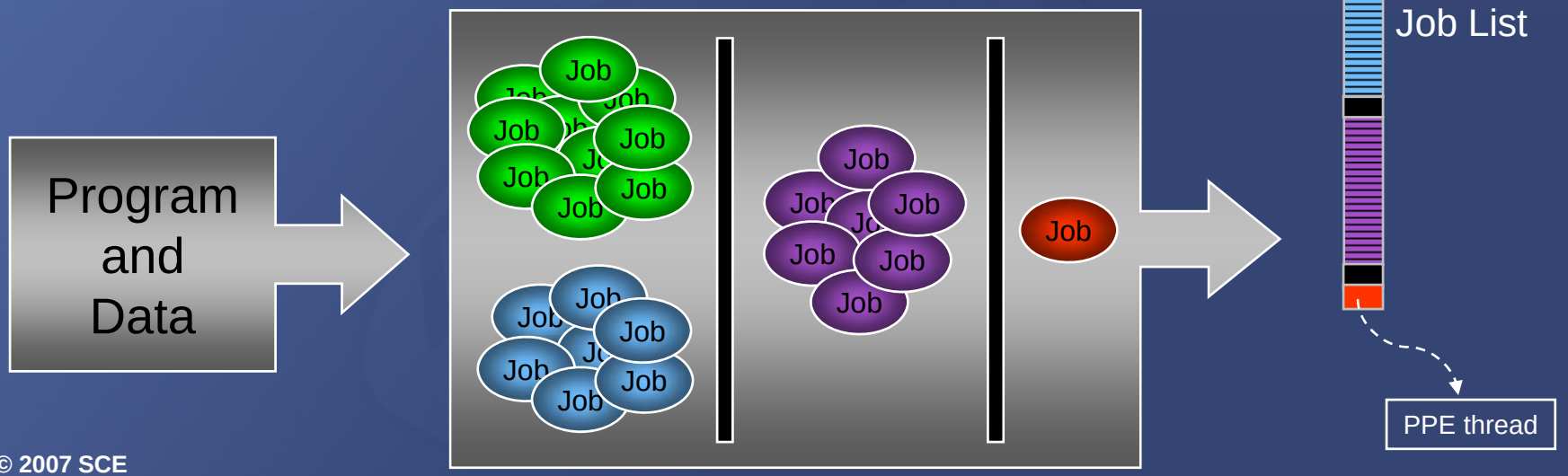
- Refactor machine vision
- Refactor particle water
- Use SPURS to share SPEs
- Room to ‘breath’

SCE's SPURS Environment

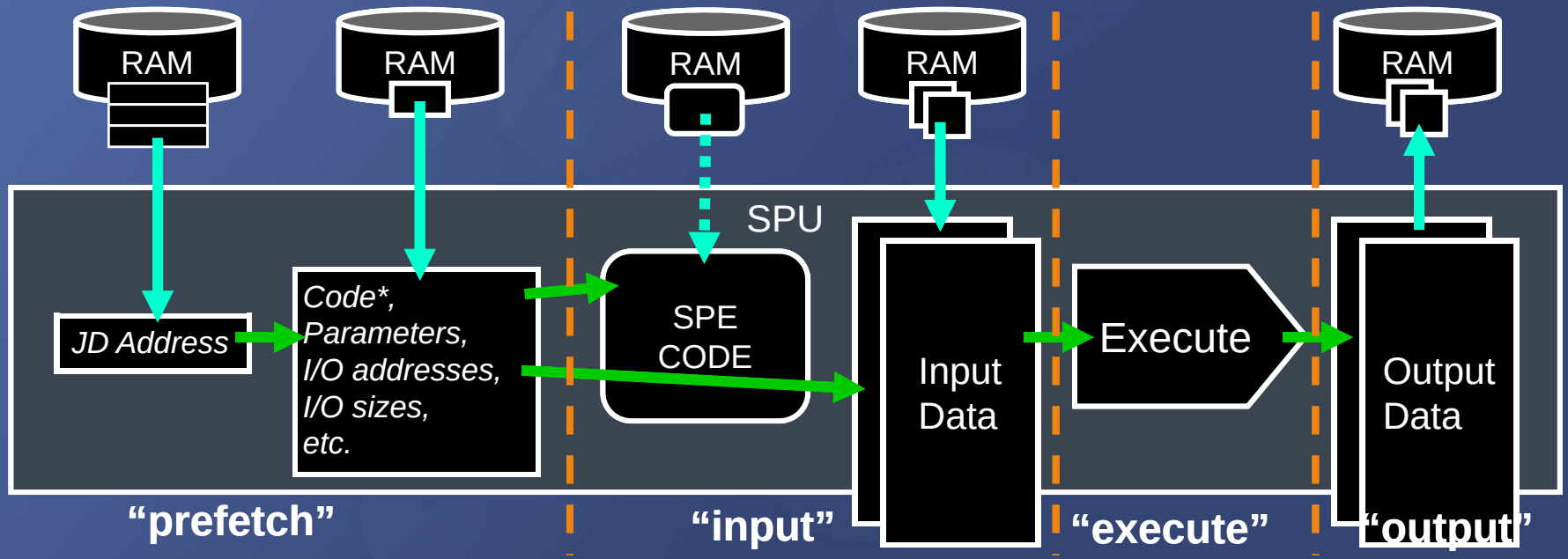
- The "Tasks" policy module
  - Similar to threads but cooperative scheduling
  - SPE's pull tasks from a shared memory pool
  - Good for mid to high complexity programs
- The "Jobs" policy module
  - Stateless execution kernels (specify all input/output)
  - SPE's pull from a shared queue of jobs
  - Good for low to mid complexity programs
  - Ideal for stream processing

Job Streaming

- Divide a program and data into pieces (called Jobs)
- Define dependencies between groups of jobs
- Build Job Lists
- SPEs grab Jobs and execute them in parallel



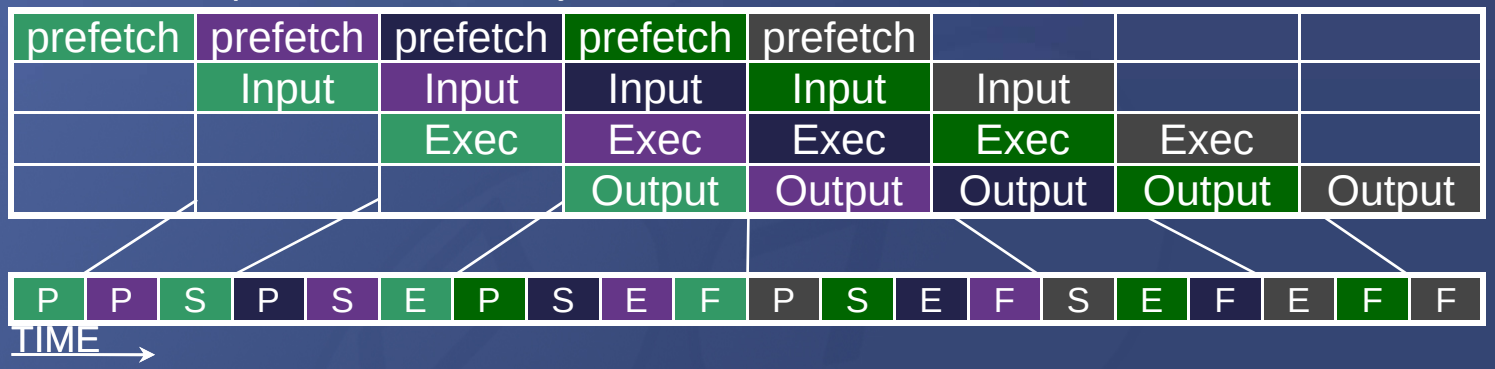
Job Streaming Pipeline



Multi-Buffering

- Job stages are interleaved so that DMA memory transfers will be in progress during job execution.

Each color represents a different job.



- Potentially, there is no stalling for memory transfers!

SCE's SPURS Environment

- SPURS solves part of the problem
  - Allows effective sharing of the SPE resources
  - Simplifies the programming and synchronization
- But it still doesn't bridge the gap
- We need higher level models which provide…
  - Automatic DMA for large code and data on SPE
  - Parallel programming abstractions
  - Scalable synchronization methods
  - Full debug and performance analysis

The Cell/B.E. Software Community

The Importance of the CoC

- The Center of Competence is a focal point
  - To bring together researchers and industry
  - To help develop optimized ‘standard’ libraries for Cell/B.E
  - Research new programming languages/models
  - Research new compiler techniques
  - General multi-core / parallel programming research
  - Dealing with distributed memory hierarchies
  - Research scalability of synchronization methods
  - Develop tools that can help visualize parallel software

Industry Support

- Terra Soft Solutions – Yellow Dog Linux for PS3
- Mercury Systems
- RapidMind
- Cmpware, Inc.
- Reservoir Labs
- Gedae
- allinea

Concluding Thoughts

Concluding Thoughts

- The Cell/B.E. has amazing performance
- Its available now in consumer and HPC markets

- *We need* more software targeting Cell/B.E.
- *We need* Cell/B.E.'s power to be more accessible
- *We need* more research into Cell/B.E. and multi-core

We need YOU to help us go..

Beyond the GFLOPS