



# Linux for the Cell Broadband Engine

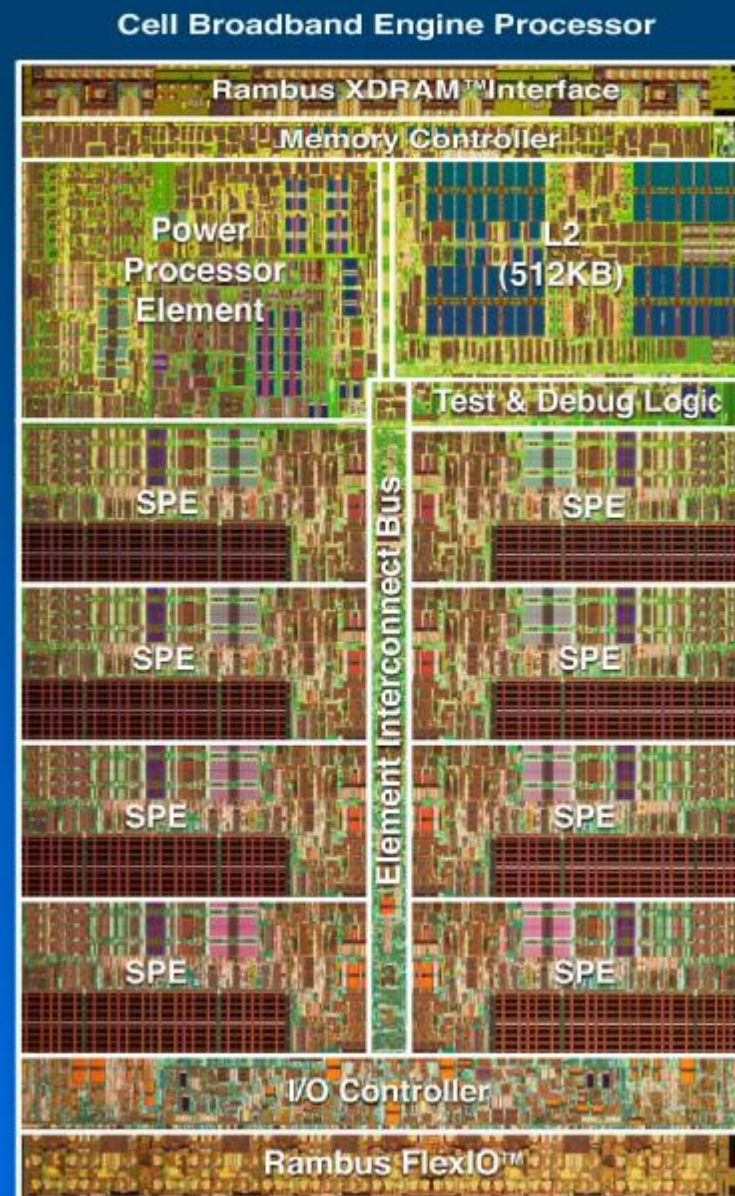
Roland Seiffert  
IBM Linux Technology Center

## Cell/B.E. Processor

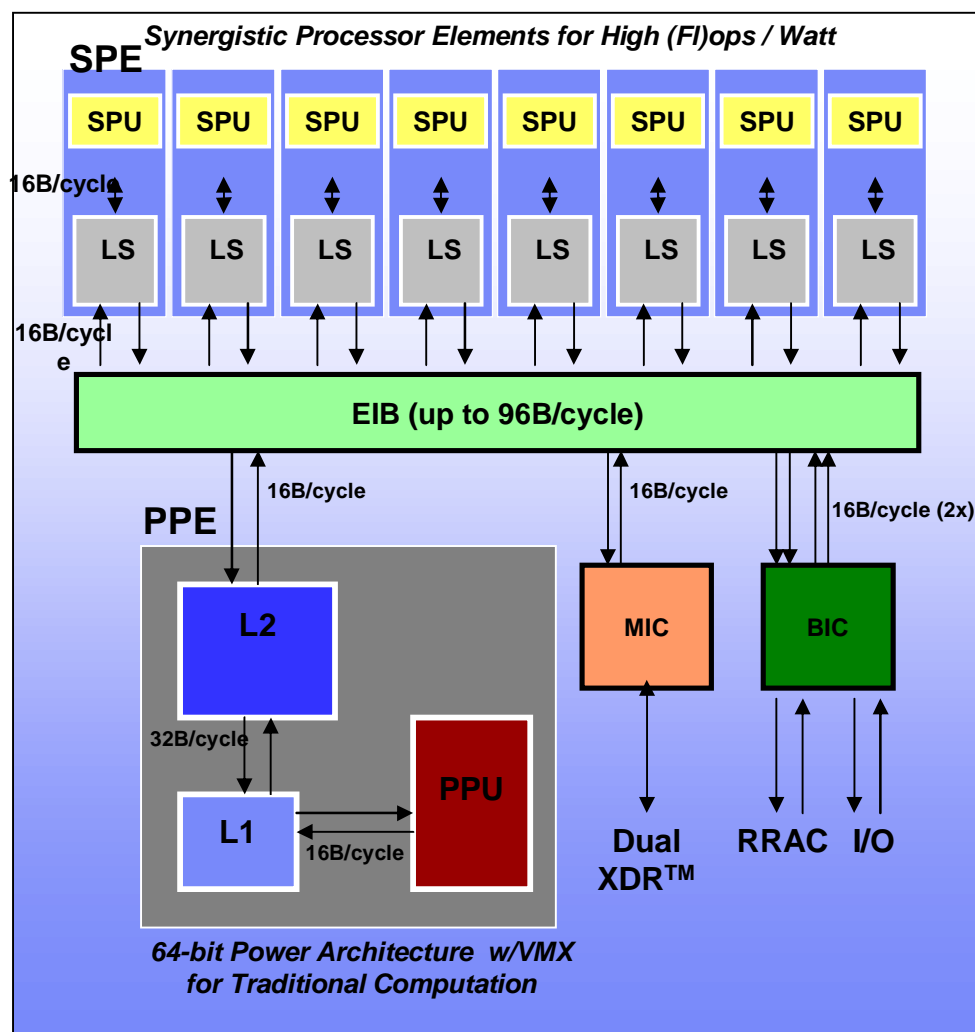
- § ~241M transistors
- § ~235mm<sup>2</sup>
- § Top frequency in lab >4GHz
- § 9 cores, 10 threads
- § > 256 GFlops (SP) @4GHz
- § > 26 GFlops (DP) @4GHz
- § Up to 25.6GB/s memory B/W
- § Up to 75 GB/s I/O B/W

### Heterogeneous multicore architecture

- 1 Power Processor Element
  - Control tasks
- 8 Synergistic Processor Elements
  - Compute-/Data-intensive tasks

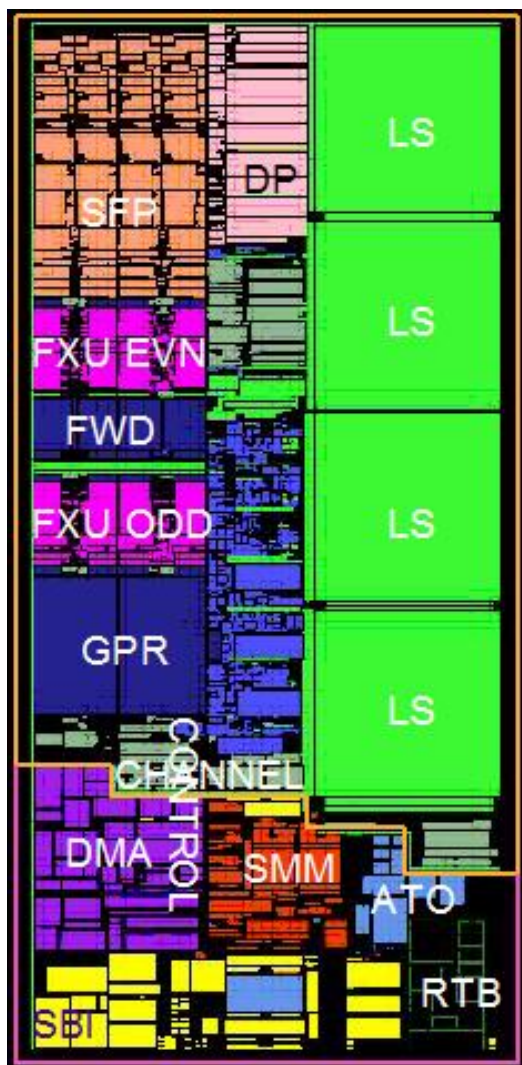


# Cell/B.E. - A Multi-Core System-on-Chip



- q Power Processor Element
  - q Control tasks
- q Synergistic Processor Element
  - q Data-intensive tasks
- q Memory Interface Controller
  - q Rambus XDR memory
- q Bus Interface Controller
  - q Rambus FlexIO
- q Element Interconnect Bus
  - q Data movement

## SPE Highlights



- q User-mode (application) architecture
  - q No translation/protection within SPU
- q 256 KB local store
  - q Combined I & D (not a cache!)
- q SIMD dataflow
  - q Graphics SP-Float
  - q IEEE DP-Float
  - q Rich set of integer operations
- q Unified Register File
  - q 128 entry x 128 bit
  - q No register renaming
- q Direct Program Control
  - q DMA, list DMA
  - q Branch hint

The Cell BE Processor and Architecture is a  
**Breakthrough Architectural Innovation** in Chip Design

**OK. Now what?**

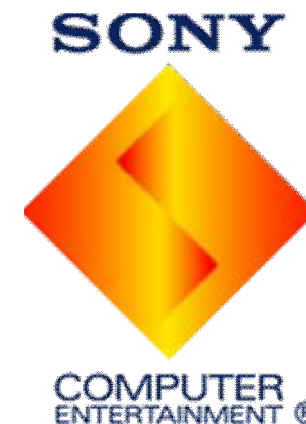
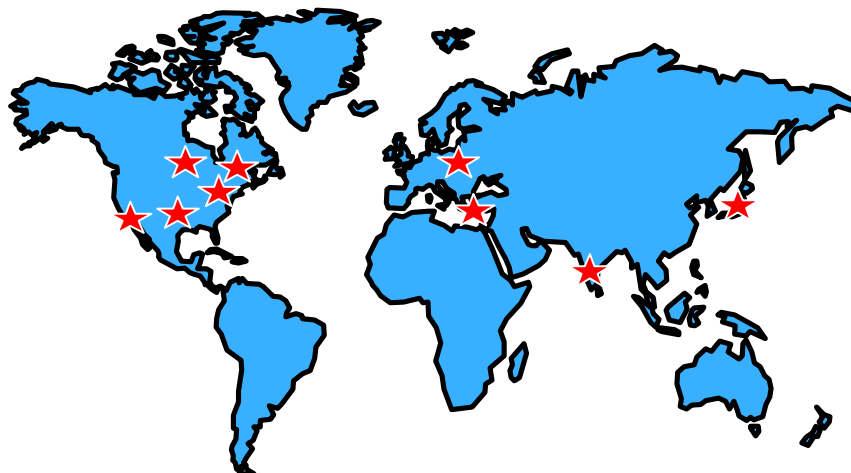
**What is it good for?  
How can I get it to do that?  
Am I on my own?**

# The Cell BE Processor and Architecture is a **Breakthrough Architectural Innovation** in Chip Design

- § **SW Challenge #1:** The breakthrough capabilities of Cell BE must be made fully available to application developers
  - Standardized (low-level) APIs, libraries, compilers, debuggers, ...
- § **SW Challenge #2:** New programming models and corresponding high-level APIs are required to allow for easy exploitation of the Cell BE capabilities
  - Open community collaboration in Research & Development to drive Cell BE exploitation – build new Cell-focused communities
- § **SW Challenge #3:** Keep existing Linux environment standard, while enabling breakthrough exploitation by applications
  - Leverage the existing Linux and Linux on POWER ecosystems as a base for the Cell BE operating environment
- § **SW Challenge #4:** Rapidly enable new communities and end-to-end solutions based on Cell BE systems
  - Interweave the existing and new communities into a “Cell-society” that embraces both standards and envelope-pushing
  - (Initial) focus on application segments with well-understood, high “Cell affinity” to create a success story and further enable community expansion

## Cell/B.E. History

- § IBM, SCEI/Sony, Toshiba Alliance formed in 2000
- § S/T/I Design Center opened in March 2001 in Austin, TX
- § Hardware designed in parallel with software, Linux
- § **Linux used for bringup / test throughout dev't cycle**
- § February 7, 2005: First external technical disclosures on Cell BE
- § **April 26, 2005: First Linux patches for Cell BE disclosed**
- § May 2005: IBM Cell BE-blade prototype running Linux demonstrated at E3
- § August 25, 2005: Release of technical documentation
- § **September 2005: Linux kernel 2.6.13 enables Cell BE platform**
- § November 9, 2005: SDK 1.0 Released
- § **March 20, 2006: Linux kernel 2.6.16 released with official support for Cell BE**
- § November 11, 2006: Playstation 3 availability with Linux support



# Sony/Toshiba/IBM Cell/B.E. Open Source Cooperation

- § Sony/SCE, Toshiba and IBM share a common vision on Cell BE Architecture and its potential in many application areas
- § S/T/I agreed to cooperate closely to enable an active ecosystem for a broad usage of Cell BE systems
- § The goal is to provide a single “Common Linux” for all systems using the Cell BE processor – including IBM’s QS21, Mercury’s Cell-based Blade, Toshiba’s Reference Set, SCE’s PlayStation 3, and all others to come.
- § The commitment is to develop this platform as part of the Open Source communities and achieve mainstream integration of the new Cell BE platform.
- § This is \*not\* an exclusive club. We...
  - ...work with and within the existing communities wherever possible
  - ...actively encourage participation by others in these efforts

## Sony/Toshiba/IBM Common Linux Workgroup

### Common Linux WG Strategy & Focus

#### § Ensure a common and attractive “Cell/B.E. Architecture runtime platform”

- The WG works on enhancing the Linux kernel and core runtime libraries with new and/or improved support in all Cell-related areas
- The WG drives the core runtime API specifications, e.g. libspe2

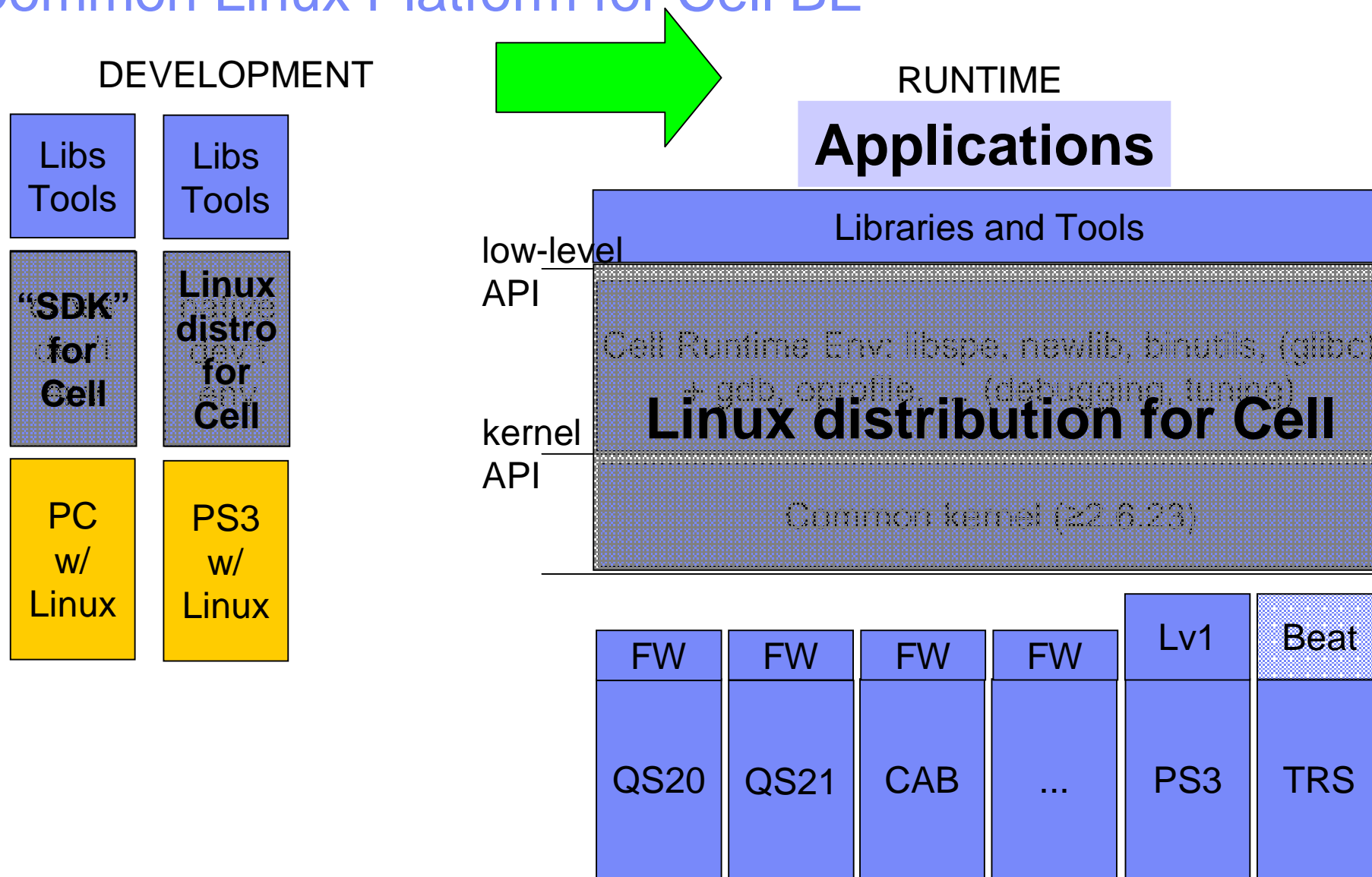
#### § Work on runtime support for extensions of the Cell/B.E. Architecture

- In 2008, we will start extending the platform to cover “hybrid models”

#### § The WG is fully dedicated to Open Source Software

- The primary goal is to integrate the Cell/B.E. Architecture into and interweave it with existing communities

# Common Linux Platform for Cell BE



## Linux for Cell/B.E. Distributions

### ***Commercial distributions***

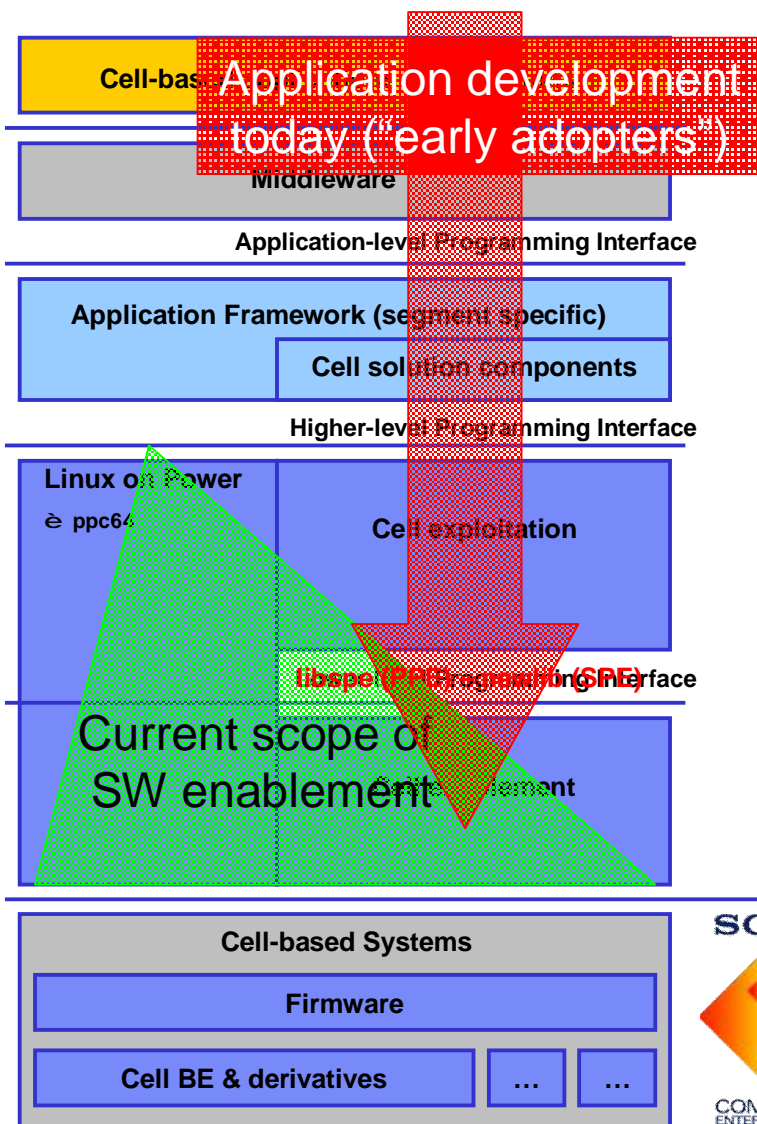
- § Red Hat Enterprise Linux 5.1 (2.6.18-based)
  - IBM Blade Server QS21
- § Yellow Dog Linux 6.0 (2.6.22-based)
  - IBM Blade Server QS21, Sony Playstation3

### ***Community Distributions***

- § Fedora 8
- § OpenSUSE 10.3
- § Ubuntu 7.10
- § Gentoo, Debian, ...

# Cell/B.E. Core Programming Model Considerations

# Cell/B.E. Software Platform



## Commercial and Open Source Exploitation of Cell

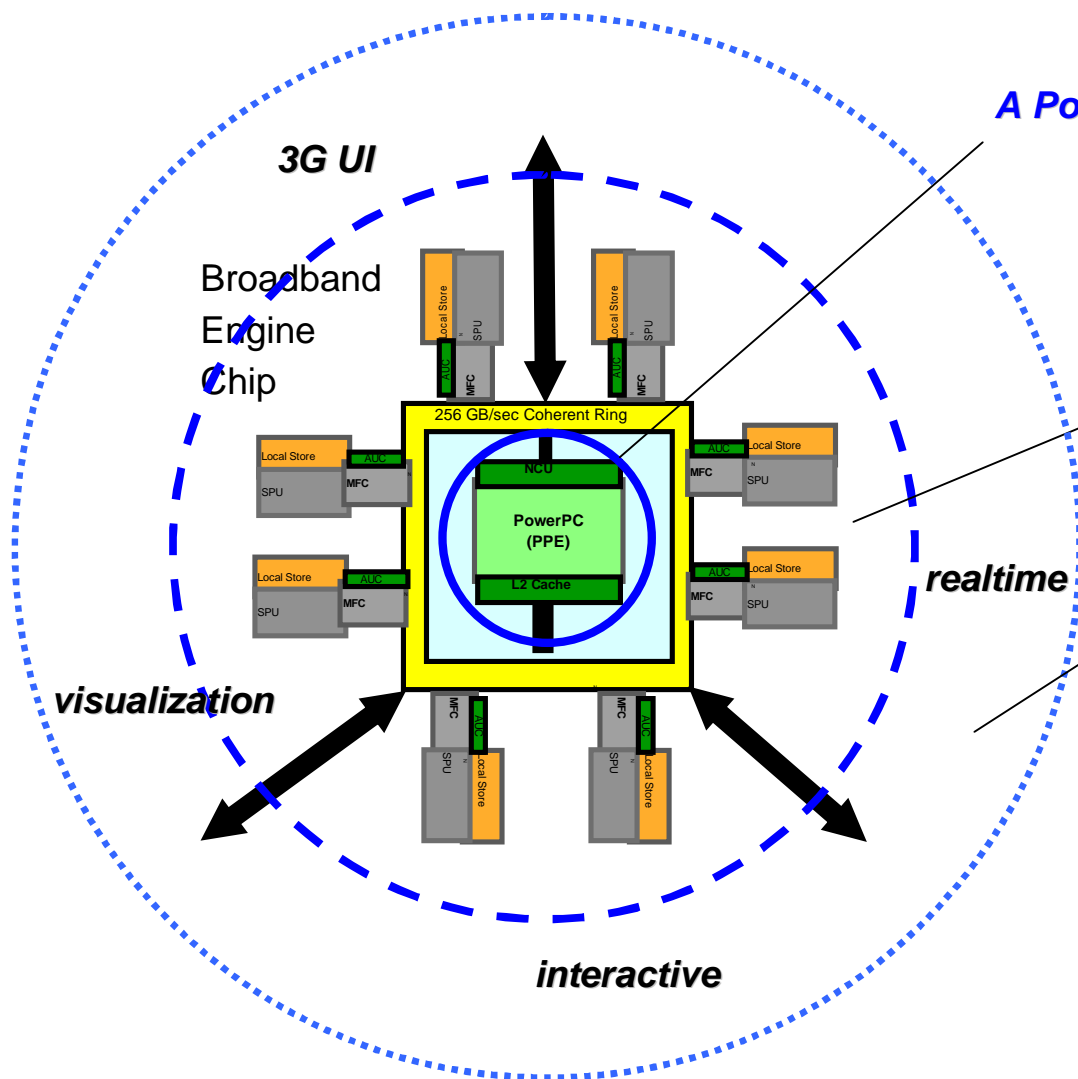
**Provide standard application platforms for Cell**  
 § Middleware and frameworks provide architecture-specific components and hide Cell-specifics from application developer

**Make Cell easier to program**  
 § Hide complexity in critical libraries  
 § Compiler support for standard tasks, e.g., overlays, global data access, SW-managed cache, auto vectorization, auto parallelization, ...  
 § Smart tooling

**Provide access to full Cell capabilities**  
 § Reflects the "exotic platform" and is hard to program  
 § Challenging to exploit, e.g., SPE's limited local memory (256 KB) – need to DMA data and code fragments back and forth; Multi-level parallelism – 8 SPEs, 128-bit wide; SIMD units in each SPE; ...



# Cell/B.E. from a Software Perspective



## A PowerPC processor

- § known architecture
- § known programming model
- § known SW stack/tool chain
- § supported by Linux on Power ecosystem

with

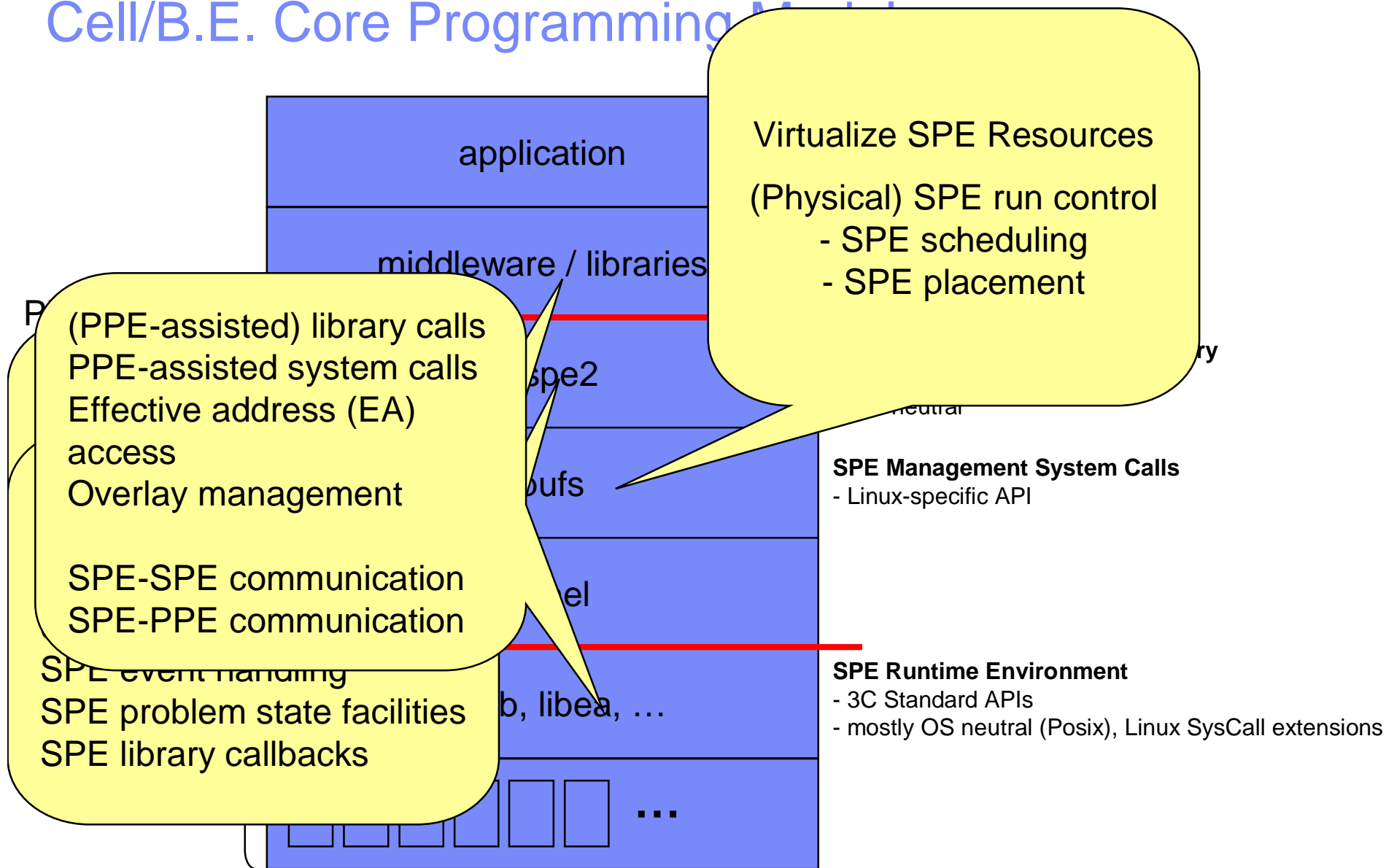
- § breakthrough new capabilities
- § radically new application structure
- § new programming models
- § accessible via support in Linux

realtime

Everything available for PowerPC just works... That's a great start!

True exploitation of the Cell BE performance potential may still be a significant challenge.

# Cell/B.E. Core Programming Model



## “Hybrid Threads”

### § PPE provides the “infrastructure”

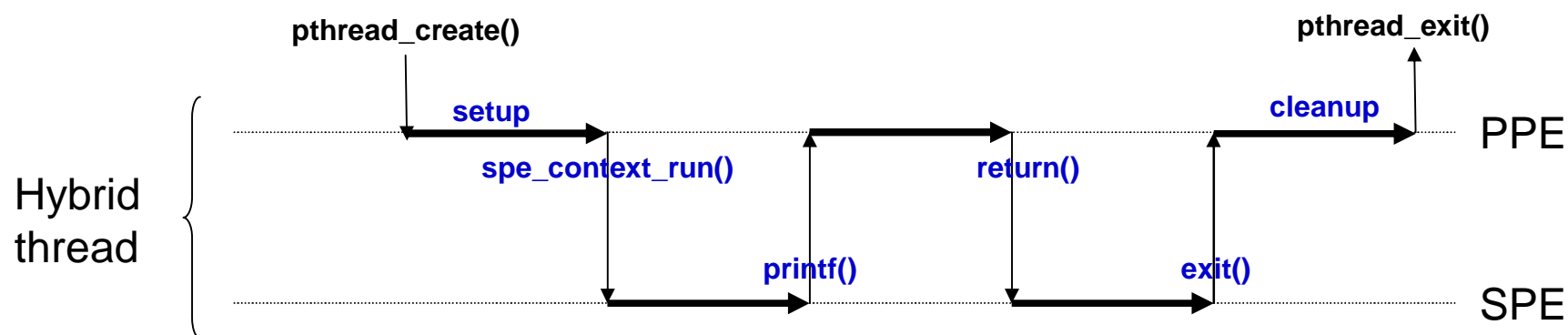
- OS kernel, device drivers,...
- Overall application logic (“orchestration”)

### § SPE provides the “compute power”

- Accelerators – for application and/or OS functions

### § Key design: the “hybrid thread”

- A (regular) OS thread started on the PPE that may use one (or more) SPEs
- Execution flip-flops between PPE and SPE as needed, e.g., setup code on PPE, then computation on SPE, execution of library/system calls on PPE, more computation on SPE, ...

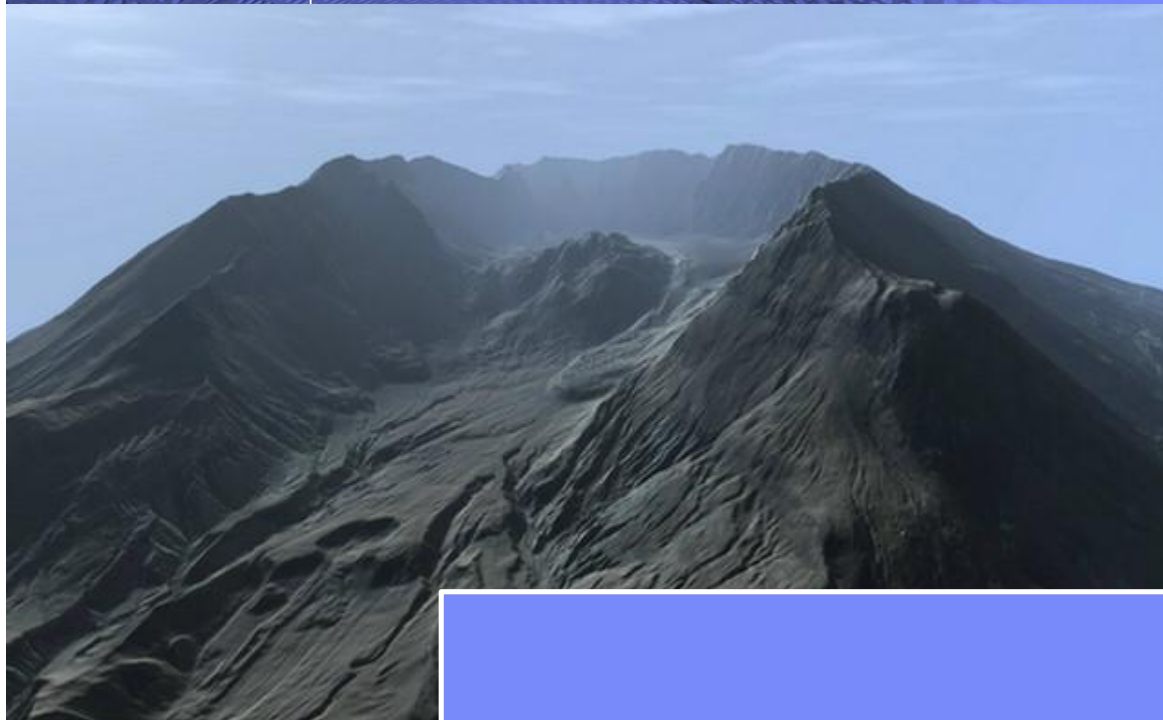


#### Notes:

- 1) Multi-threaded (parallel) applications use multiple hybrid threads to use multiple SPEs
- 2) A single hybrid thread may manage multiple SPE contexts – but only one can be running at any given point in time

# SUMMARY

- § Cell has been / is / will be a success story for ***collaborative innovation***
  - from chip development to business development
  - finding new ways to use collaboration to drive core technology evolution, adoption and ecosystems is a critical new demand placed on technologists and successful businesses
- § Open Source software forms a critical element in this
  - but is not enough...
- § Linux is *great!*
  - great technology – mature, but open for change
- § The Linux community is *fantastic!*
  - you'll always find the enthusiasts you need to get something new and big started and moving fast



Thank you!

