

Introduction to

Cell Broadband Engine Architecture Processor

Amir Khorsandi

Spring 2012

Agenda

- **History**
- Motivation
- Architecture
- Software Environment
- Power of Parallel Processing
- Conclusion

History

SONY



IBM



TOSHIBA



History

- IBM, SCEI/Sony, Toshiba Alliance formed in 2000
- Design Center opened in March 2001 Based in Austin, Texas
- February 7, 2005: First technical disclosures
- May 16, 2005: First public demonstrations at E3
- August 25, 2005: Release of technical documentation

Agenda

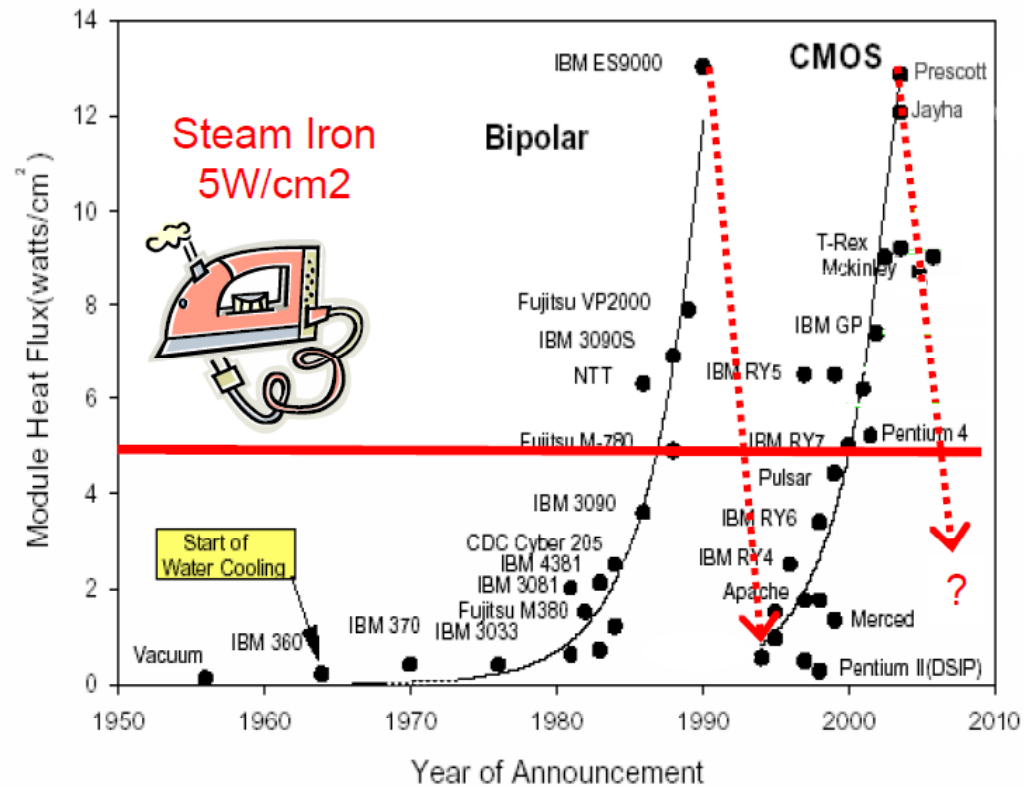
- History
- **Motivation**
- Architecture
- Software Environment
- Power of Parallel Processing
- Conclusion

Motivation

- Limiters to Processor Performance
 - Power wall
 - Memory wall
 - Frequency wall

Motivation

- Power wall



Motivation

- Memory wall
 - Main memory now nearly 1000 cycles from the processor
 - Situation worse with (on-chip) SMP
 - Memory latency penalties drive inefficiency in the design
 - Expensive and sophisticated hardware to try and deal with it
 - Programmers that try to gain control of cache content, but are hindered by the hardware mechanisms
 - Latency induced bandwidth limitations
 - Much of the bandwidth to memory in systems can only be used speculatively
 - Diminishing returns from added bandwidth on traditional systems

Motivation

- Frequency wall
 - Increasing frequencies and deeper pipelines have reached diminishing returns on performance
 - Returns negative if power is taken into account
 - Results of studies depend on issue width of processor
 - The wider the processor the slower it wants to be
 - Simultaneous Multithreading helps to use issue slots efficiently
 - Results depend on number of architected registers and workload
 - More registers tolerate deeper pipeline
 - Fewer random branches in application tolerates deeper pipelines

Motivation

- Microprocessor Efficiency
 - Gelsinger's law
 - 1.4x more performance for 2x more transistors
 - Hofstee's corollary
 - $1/1.4x$ efficiency loss in every generation
 - • Examples: Cache size, OoO, Superscalar, etc. etc.

Agenda

- History
- Motivation
- **Architecture**
- Software Environment
- Power of Parallel Processing
- Conclusion

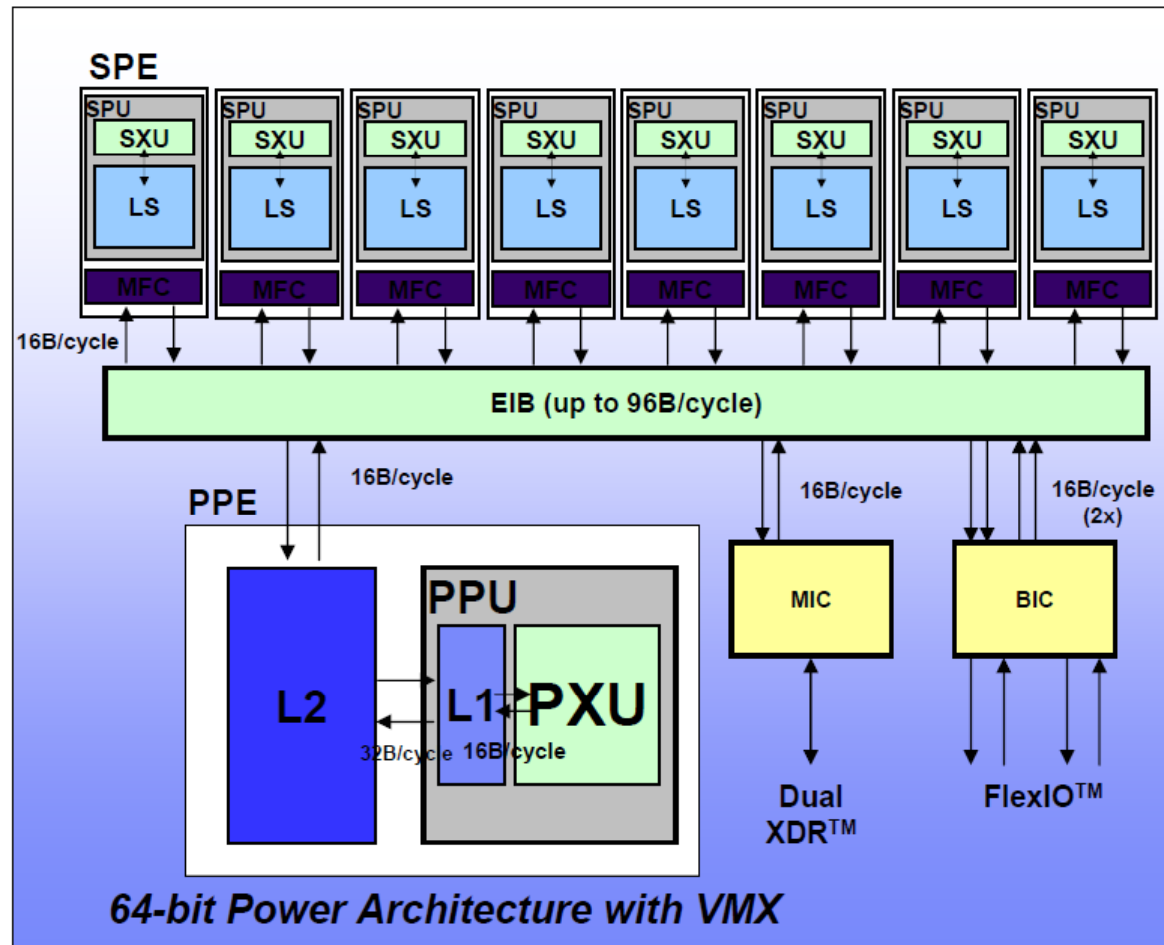
Architecture

- Patented by Ken Kuturagi (Sony) in 1999.
- Consists of:
 - Software Cell: a program with the associated data
 - Hardware Cell: an execution unit with the capability to execute a software cell

Architecture

- 3.2 GHz Cell Chip Highlights
 - 241M transistors
 - 235 mm²
 - 9 cores, 10 threads
 - >200 GFlops (SP)
 - >20 GFlops (DP)
 - Up to 25 GBps memory BW
 - Up to 75 GBps I/O BW
 - >300 GBps interconnect medium
 - >4Ghz frequency (observed in lab)

Architecture



Architecture

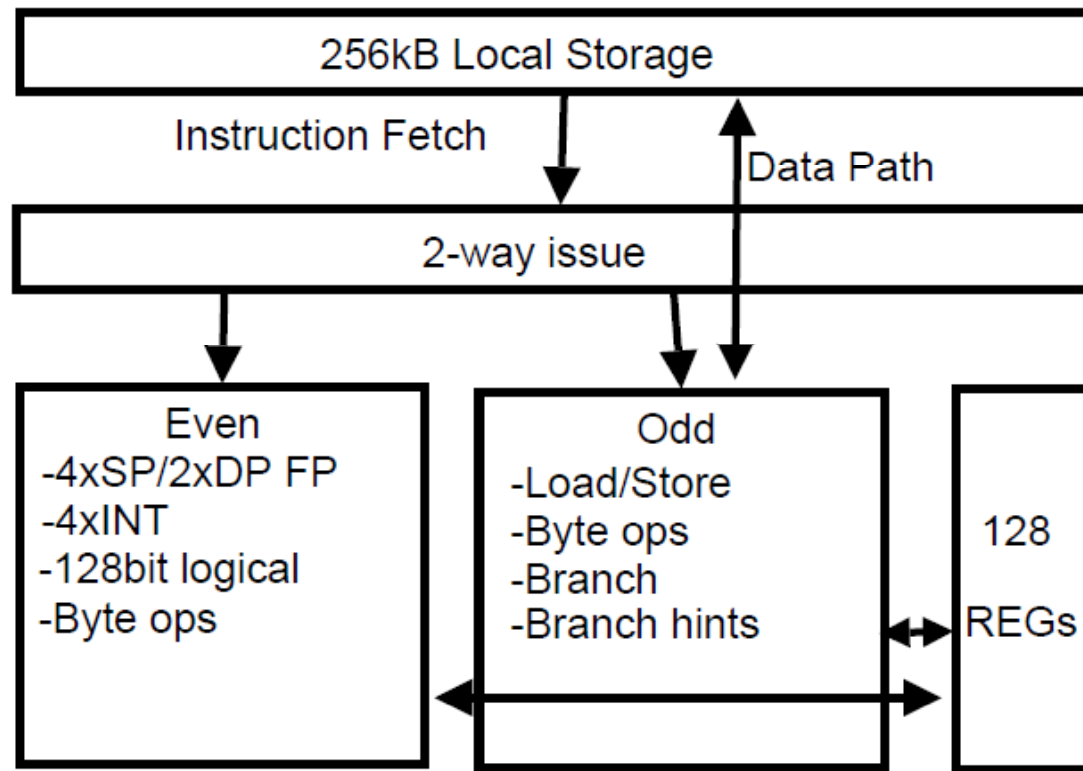
- Power Processing Element (PPE)
 - General purpose, 64-bit RISC processor (PowerPC AS 2.0.2)
 - 2-Way hardware multithreaded
 - L1 : 32KB I ; 32KB D
 - L2 : 512KB
 - Coherent load / store
 - VMX-32
 - Realtime Controls

Architecture

- Synergistic Processor Element (SPE)
 - RISC vector processor with fixed length instruction words of 32-bit
 - No branch prediction or scheduling logic
 - Issues two instructions per cycle:
 - one SIMD computation operation
 - one memory access operation
 - In order execution
 - 128-bit compound data

Architecture

- Synergistic Processor Element (SPE)



Architecture

- Element Interconnect Bus (EIB)
 - Central communication channel
 - Consists of four 128 bit wide concentric rings
 - Moves 96 bytes per cycle and is optimized for 1024 bit data blocks
 - Buffered point to point communication to transfer the data
 - Additional nodes (e.g. SPEs) only affect the maximal latency of the ring
 - A hardware guaranteed bandwidth of $1/\text{numDevices}$ for each node

Architecture

- Memory Interface Controller (MIC)
 - Connects the EIB to the main DRAM memory
 - XDR memory with a bandwidth of 25.2 GB/s
 - Virtual memory translation to the PPE and the SPEs
 - The memory itself is not cached

Architecture

- The I/O Interconnect – FlexIO
 - Connects the Cell processor (the EIB) to the external world
 - 12 uni-directional byte-lanes which are 96 wires
 - Each lane may transport up to 6.4GB/s
 - 76.8 GB accumulated bandwidth
 - 7 lanes are outgoing (44.8 GB/s) and 5 lanes incoming (32 GB/s)
 - Two cell processors can be connected glueless

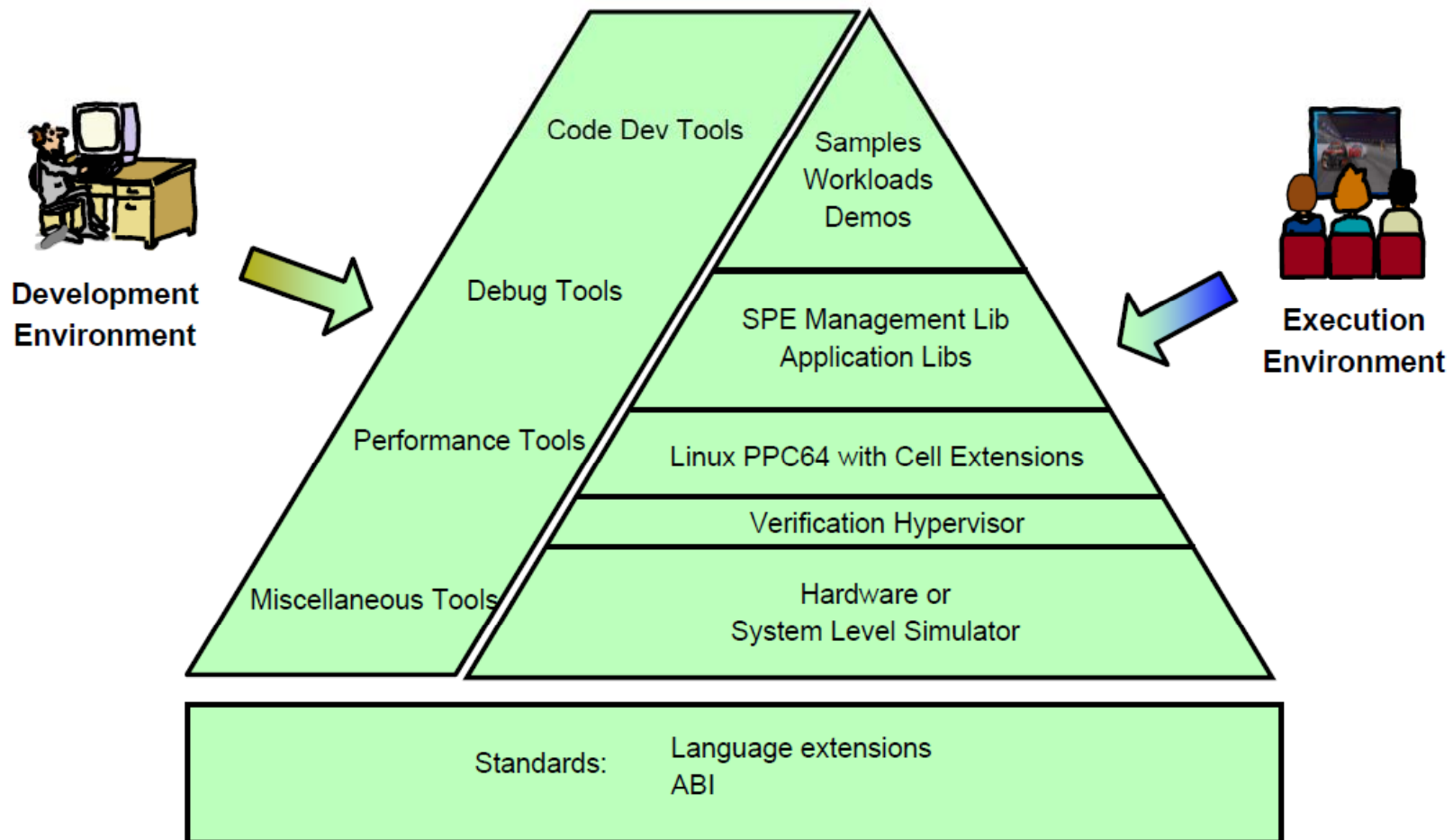
Agenda

- History
- Motivation
- Architecture
- **Software Environment**
- Power of Parallel Processing
- Conclusion

Software Environment

- No abstraction layer between an external ISA and the internal core (cmp. x86)
- RISC design moves the effort to generate optimal code up, to the programmer or compiler
- The SPEs are programmed in a direct manner
- The task distribution and allocation of SPEs is fully done in software
- The Local Storage could be used as a cache, but has to be managed by the software

Software Environment



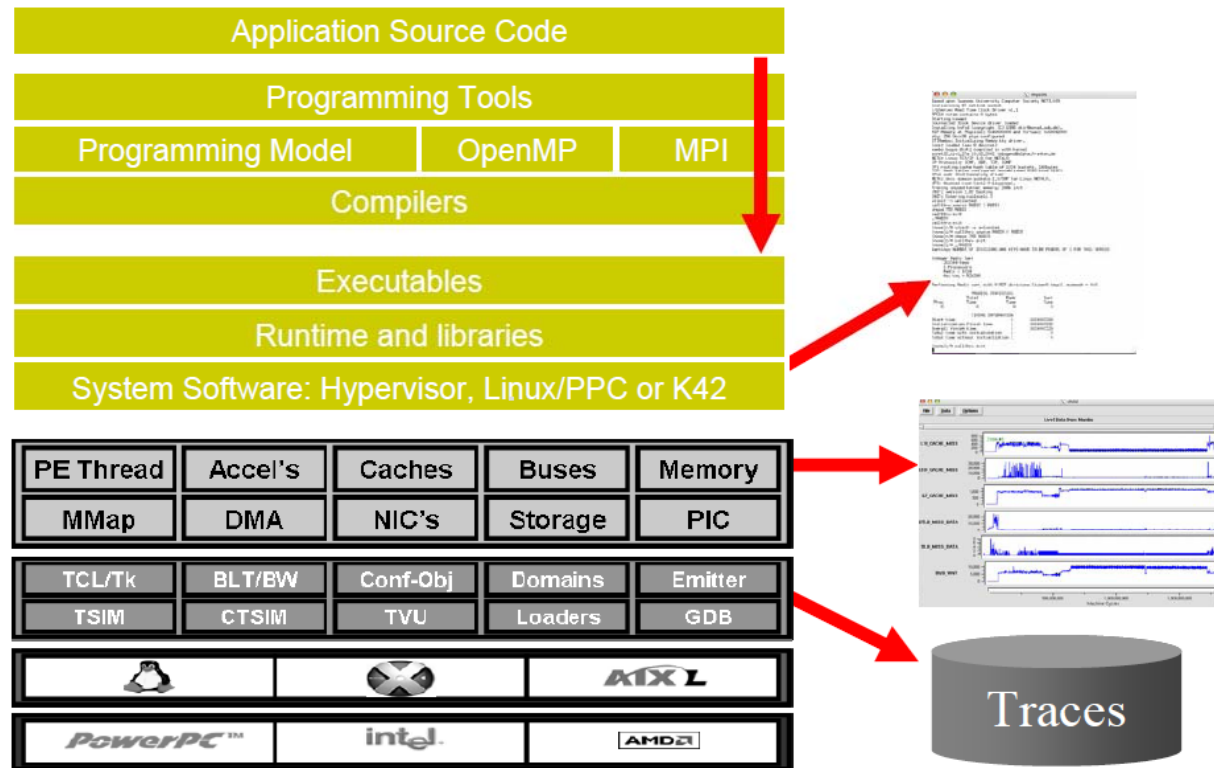
Software Environment

- Cell BE – full system simulator
 - Uni-Cell and multi-Cell simulation
 - User Interfaces – TCL and GUI
 - Cycle accurate SPU simulation (pipeline mode)
 - Emitter facility for tracing and viewing simulation events

Software Environment

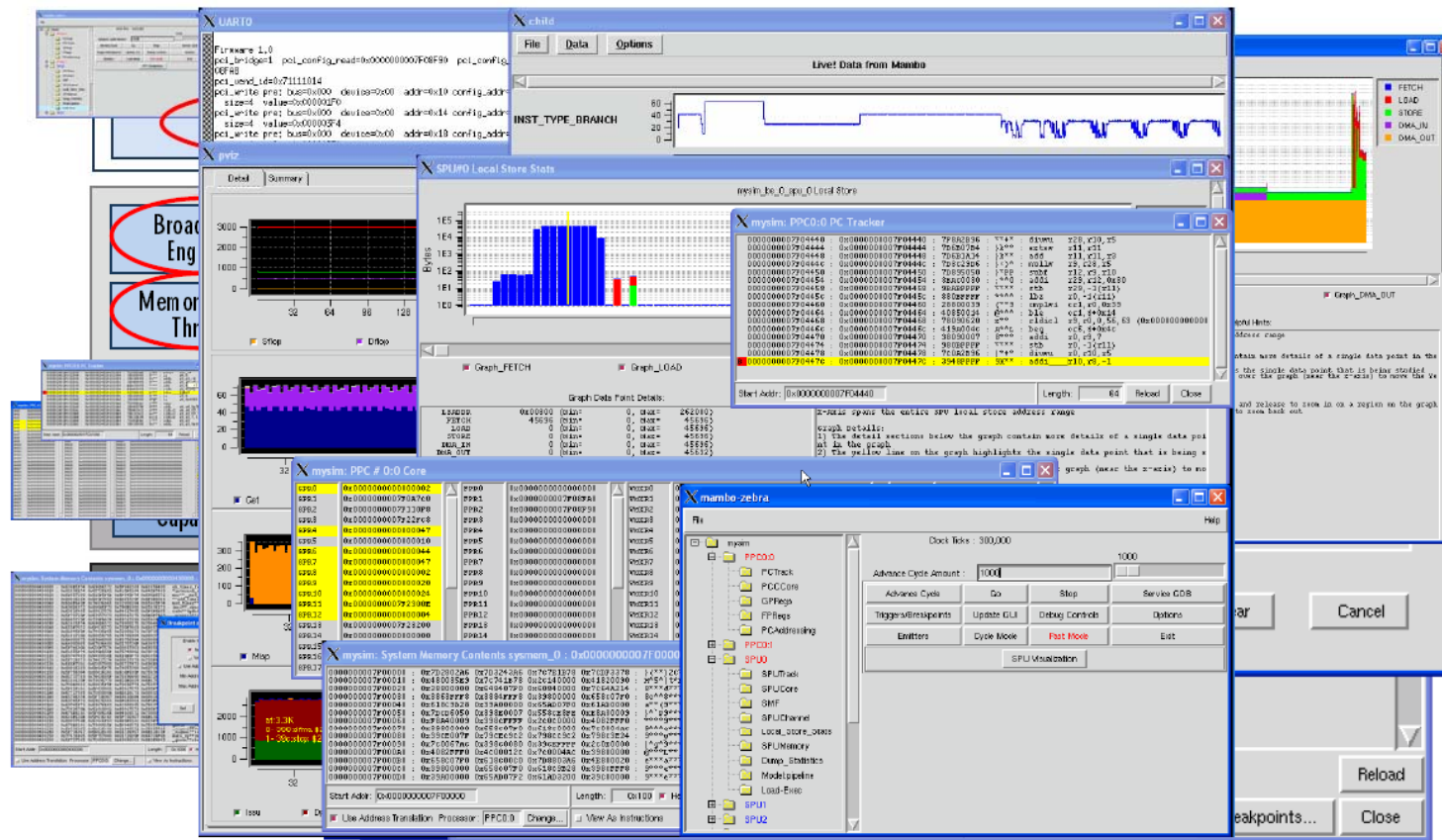
- SW Stack in Simulation

*CellSim:
Simulation of
hardware*



Software Environment

Cell Simulator Debugging Environment

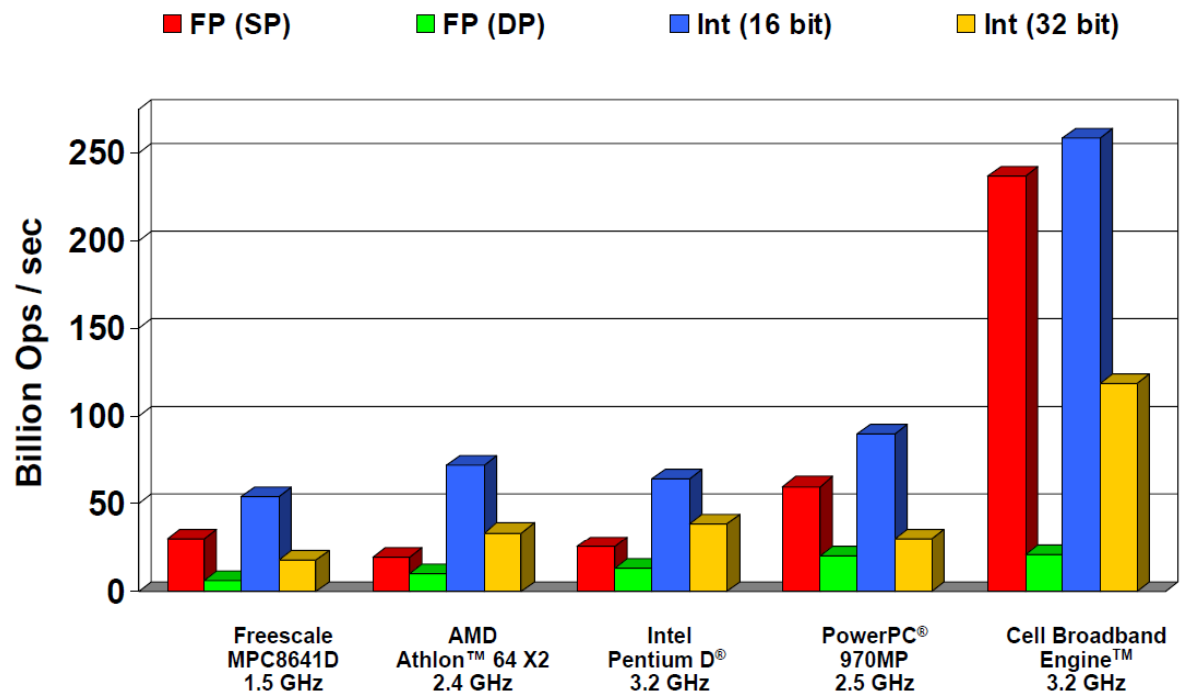


Agenda

- History
- Motivation
- Architecture
- Software Environment
- **Power of Parallel Processing**
- Conclusion

Power of Parallel Processing

- High performance can be achieved with a single cell
- We can develop it to gain more



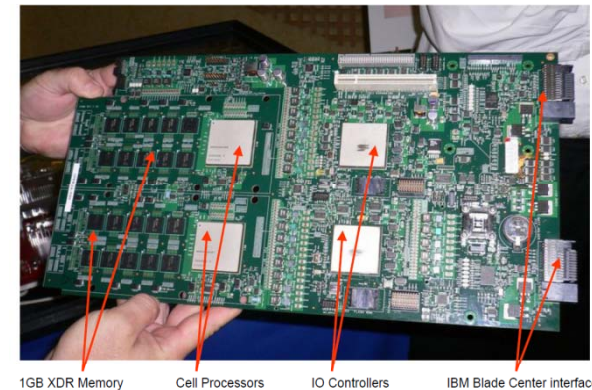
Power of Parallel Processing



Scene	ERW6	Conference	VW Beetle
ray casting, no shading			
2.4GHz x86	28.1	8.7	7.7
2.4GHz SPE	30.1 (+7%)	7.8 (-12%)	7.0 (-10%)
Single-Cell	231.4 (8.2x)	57.2 (6.5x)	51.2 (6.6x)
Dual-Cell	430.1 (15.3x)	108.9 (12.5x)	91.4 (11.8x)
PS3-Cell	270.0 (9.6x)	66.7 (7.6x)	59.7 (7.7x)
ray casting, simple shading			
2.4GHz x86	15.3	6.7	6.6
2.4GHz SPE	14.9 (-3%)	5.1 (-23%)	3.5 (-47%)
Single-Cell	116.3 (7.6x)	38.7 (5.7x)	27.1 (4.1x)
Dual-Cell	222.4 (14.5x)	73.7 (11x)	47.1 (7.1x)
PS3-Cell	135.6 (8.9x)	45.2 (6.7x)	31.6 (4.8x)
ray casting, shading&shadows			
2.4GHz x86	7.2	3.0	2.5
2.4GHz SPE	7.4 (+3%)	2.6 (-13%)	1.9 (-24%)
Single-Cell	58.1 (8x)	20 (6.6x)	16.2 (6.4x)
Dual-Cell	110.9 (15.4x)	37.3 (12.4x)	30.6 (12.2x)
PS3-Cell	67.8 (9.4x)	23.2 (7.7x)	18.9 (7.5x)

Power of Parallel Processing

- It is possible to gain more, Cell Blade
 - Blade
 - Two Cell BE Processors
 - 1GB XDRAM
 - BladeCenter Interface (Based on IBM JS20)
 - Chassis
 - Standard IBM BladeCenter form factor with:
 - 7 Blades (for 2 slots each) with full performance
 - 2 switches (1Gb Ethernet) with 4 external ports each
 - Updated Management Module Firmware.
 - External Infiniband Switches with optional FC ports
 - Typical Configuration (available today from E&TS)
 - eServer 25U Rack
 - 7U Chassis with Cell BE Blades, OpenPower 710
 - Nortel GbE switch
 - GCC C/C++ (Barcelona) or XLC Compiler for Cell (alphaworks)
 - SDK Kit on <http://www-128.ibm.com/developerworks/power/cell/>



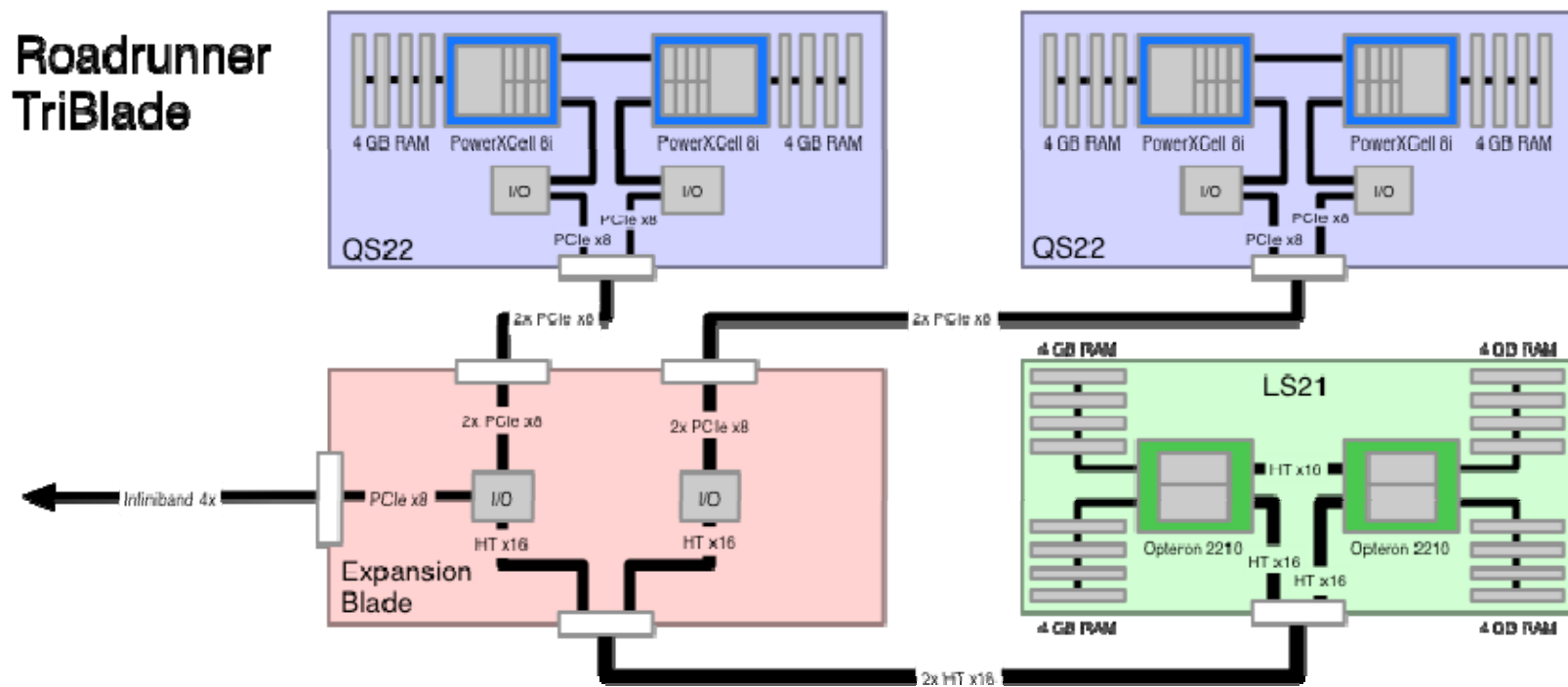
Power of Parallel Processing

- Even more, IBM Roadrunner
 - Currently the world's tenth fastest computer
 - US\$133-million Roadrunner is designed for a peak performance of 1.7 petaflops
 - In November 2008, it reached a top performance of 1.456 petaflops
 - It is also the fourth-most energy-efficient supercomputer in the world



Power of Parallel Processing

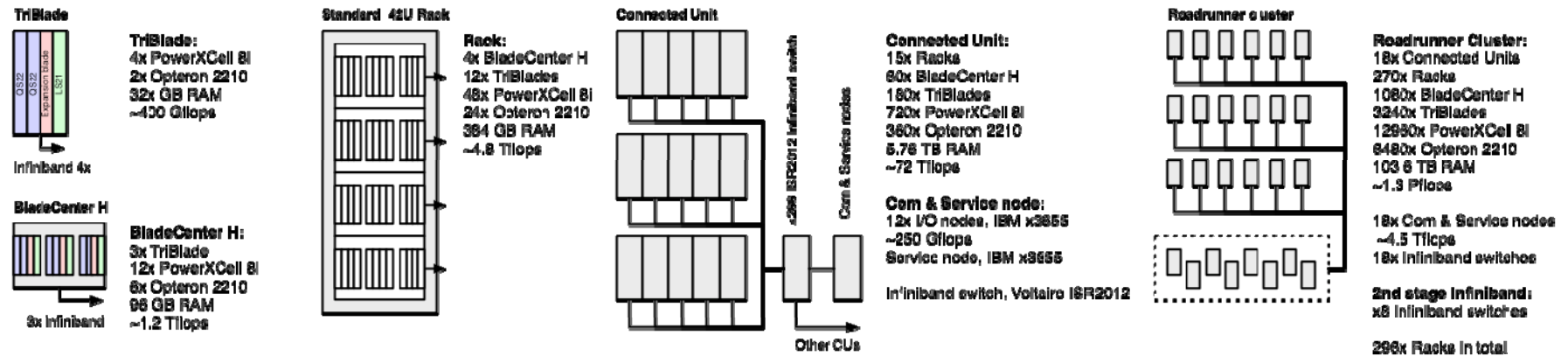
- IBM Roadrunner Triblade Module



Power of Parallel Processing

- IBM Roadrunner Cluster

Roadrunner, tiered architecture



Agenda

- History
- Motivation
- Architecture
- Software Environment
- Power of Parallel Processing
- **Conclusion**

Conclusion

- Powerful architecture for attacking
 - Power wall
 - Memory wall
 - Frequency wall
- High potential for parallel processing
- Develop needs Expertise

References

- Cell Broadband Engine Architecture (Version 1.01)
 - Sony Corporation, October 2006
- The Cell Processor - A short Introduction –
 - Torsten Hoefler, November 2005
- Introduction to the Cell Processor
 - Dr. Michael Perrone (IBM-MIT), 2007
- Power Efficient Processor Design and the Cell Processor
 - Dr.H. Peter Hofstee (IBM), 2005
- Cell Architecture
 - IBM Corporation, 2005
- The Cell Processor Architecture & Issues
 - IBM Corporation, 2005
- Cell Broadband Engine Architecture Processor
 - Ryan Layer, Ben Kreuter, Michelle McDaniel, Carrie Ruppar
- Ray Tracing on the Cell Processor
 - Carsten Benthin, Ingo Wald, Michael Scherbaum, Heiko Friedrich
- http://en.wikipedia.org/wiki/IBM_Roadrunner

Thanks



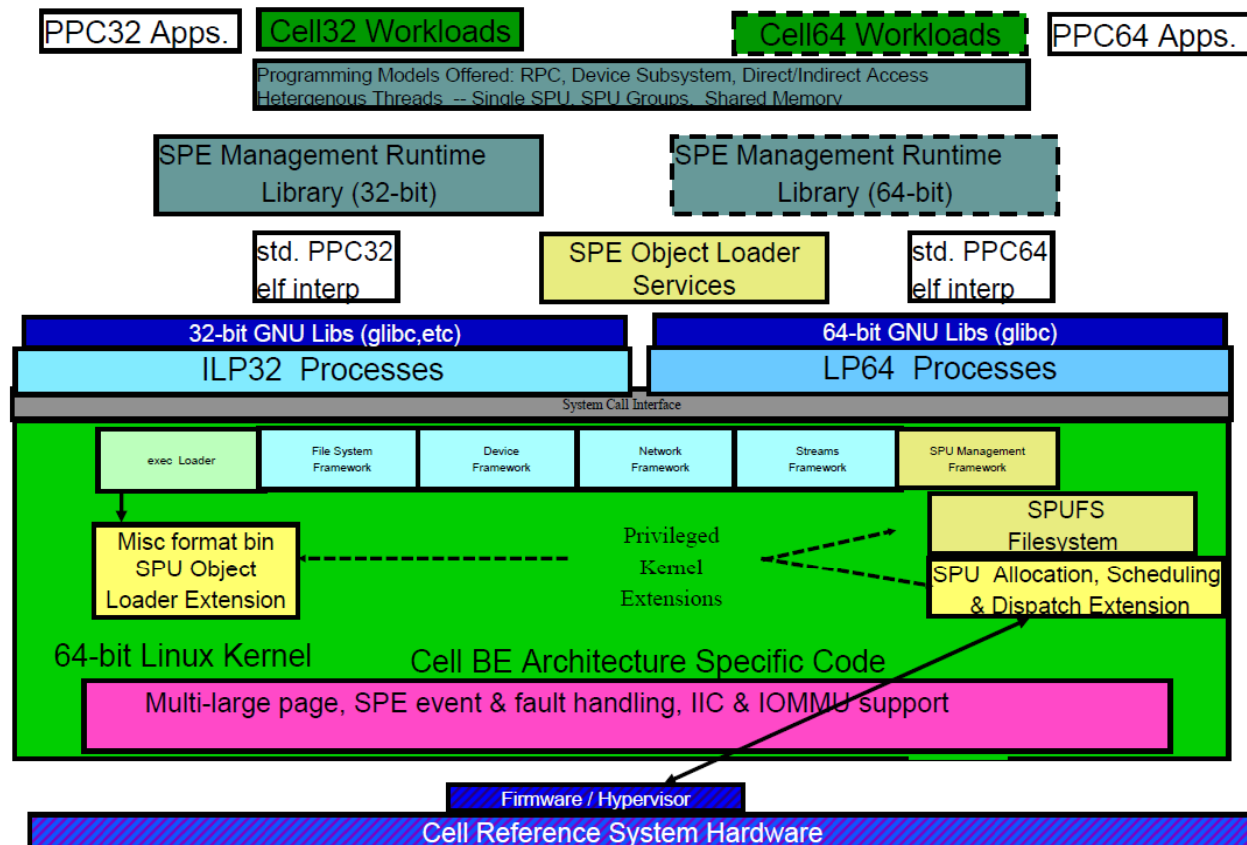
Backup Slides

Software Environment

- Linux on Cell BE
 - Provided as patched to the 2.6.15 PPC64 Kernel
 - Added heterogeneous lwp/thread model
 - SPE thread API created (similar to pthreads library)
 - User mode direct and indirect SPE access models
 - Full pre-emptive SPE context management
 - spe_ptrace() added for gdb support
 - spe_schedule() for thread to physical SPE assignment
 - currently FIFO – run to completion
 - SPE threads share address space with parent PPE process (through DMA)
 - Demand paging for SPE accesses
 - Shared hardware page table with PPE
 - PPE proxy thread allocated for each SPE thread to:
 - Provide a single namespace for both PPE and SPE threads
 - Assist in SPE initiated C99 and POSIX-1 library services
 - SPE Error, Event and Signal handling directed to parent PPE thread
 - SPE elf objects wrapped into PPE shared objects with extended gld
 - All patches for Cell in architecture dependent layer (subtree of PPC64)

Software Environment

- Linux on Cell BE



Software Environment

- SPE Management Library
 - SPEs are exposed as threads
 - SPE thread model interface is similar to POSIX threads.
 - SPE thread consists of the local store, register file, program counter, and MFC-DMA queue Associated with a single Linux task
 - Features include:
 - Threads - create, groups, wait, kill, set affinity, set context
 - Thread Queries - get local store pointer, get problem state area pointer, get affinity, get context
 - Groups - create, set group defaults, destroy, memory map/unmap, madvise
 - Group Queries - get priority, get policy, get threads, get max threads per group, get events
 - SPE image files - opening and closing
 - SPE Executable
 - Standalone SPE program managed by a PPE executive
 - Executive responsible for loading and executing SPE program
 - It also services assisted requests for I/O (eg, fopen, fwrite, fprintf) and memory requests (eg, mmap, shmat, ...)

Software Environment

- Optimized SPE and Multimedia Extension Libraries
 - Standard SPE C library subset
 - optimized SPE C99 functions including stdlib c lib, math and etc.
 - subset of POSIX.1 Functions – PPE assisted
 - Audio resample - resampling audio signals
 - FFT - 1D and 2D fft functions
 - gmath - mathematic functions optimized for gaming environment
 - image - convolution functions
 - intrinsics - generic intrinsic conversion functions
 - large-matrix - functions performing large matrix operations
 - matrix - basic matrix operations
 - mpm - multi-precision math functions
 - noise - noise generation functions
 - oscillator - basic sound generation functions
 - sim – simulator only function including print, profile checkpoint, socket I/O, etc ...
 - surface - a set of bezier curve and surface functions
 - sync - synchronization library
 - vector - vector operation functions