Cell, Multi-core Programming, and Applications

Alex Chunghen Chow
Cell architecture

- **1 PPE**
  - VMX unit
  - L1, L2 caches
  - 2 way SMT

- **8 SPEs**
  - 128-bit SIMD instruction set
  - Register file – 128x128-bit
  - Local store – 256KB
  - Instruction execution latency

- **EIB + PPE L1/L2 + SPE MFCs**
  - Bus bandwidth
  - DMA latency
  - Memory address-ability

- **System memory**
  - Bandwidth 25.6GB/s
Cell/B.E. uses ½ the space & power vs traditional approaches

**Example Dual Core**
- 349mm², 3.4 GHz @ 150W
- 2 Cores, ~27.2 SP GFlops
- 65nm

**Example Quad Core**
- 214 mm², 3 GHz @ 130W
- 4 Cores, ~48 SP GFlops
- 45nm

**Cell/B.E.**
- 175 mm², 3.2 GHz
- 9 Cores, ~230 SP GFlops
- 65nm

On any traditional processor, shown ratio of cores to cache, prediction, & related items illustrated here remains at ~50% of area the chip area.

Intel’s x86 Quad Core processors are Dual Chip Modules (DCMs), 2 of these processor stacked vertically & packaged together.
Cell/B.E.‘s Heterogeneous Core architecture allows it to cover a greater range
General Software Programming Direction(s)

- Adopting open standards to achieve higher performance and still retain new multi-core software investments
  - OpenMP (Open Multi-processing)
    - A language/compiler directives to facilitate a compiler in generating parallel programs
  - OpenCL (Open Computing Language)
    - A language and environment for developing and running data and task parallel programs across heterogeneous cores

- A path to migrate legacy serial / parallel / distributed applications
  - Support legacy programming and execution environments / libraries - e.g.
    - Standard languages: C, C++, Fortran
    - Standard operating environment
      - POSIX (Portable Operating System Interface)
        - e.g. standardized pthread multi-threading support
    - MPI (Message Passing Interface)
      - Explicit communication library for distributed executing programs