A Transport Kernel on the Cell Broadband Engine

Paul Henning

Los Alamos National Laboratory
Cell Chip Overview

- **Cell Broadband Engine™** *(Cell BE)*
  - Developed under Sony-Toshiba-IBM efforts
  - Current Cell chip is used in the Sony PlayStation 3
- An 8-way heterogeneous parallel engine

Each of the 8 SPEs are 128 byte (e.g. 2-way DP-FP) vector engines w/ 256KB of Local Store (LS) memory & a DMA engine.

They can operate together or independently (SPMD or MPMD).

~153 GF/s single precision (2.4 GHz chip)
~ 11 GF/s double precision (2.4 GHz chip)
Cell Chip Overview - Cont.

• Each SPU has 256KiB of “local store” (LS).

• Data/Instructions transferred between LS and main memory with DMA calls.
  – EIB transfers at 25.6 GiB/s.
  – Maximum of 16 KiB per transfer.
  – Data alignment critical for performance.

• SPU and PPE have different instruction sets.
  – SPU is vector-only.
  – SPU executable is packaged in the PPE executable and DMA’d to SPU LS.
Terminology: Radiation Spectrum

- A blackbody emits radiation with a frequency spectrum that is characterized by the Planck distribution function.
- The “gray approximation” averages the emission spectrum.
- The “multigroup approximation” divides the spectrum into equally spaced bins or frequency groups.
Normalized Planck Distribution
MG Opacity Application Overview

- Determine zone opacities for multigroup thermal radiation transport on a mesh.
- Calculates definite integral of Planck distribution within each group interval and combines to form zone opacities.
- Outer loop on number of zones (NZ).
- Inner loop on number of groups (NG).
- 
  \((3+2\times NG)\times NZ+NG+6\) read-only input doubles.
- 
  \((4+NG)\times NZ\) output doubles.
Kernel Overview

\[ b_g = \frac{15}{\pi^4} \int_{x_g}^{x_{g+1}} \frac{x'^3}{e^{x'} - 1} \, dx' \]

- Approximated by taking minimum of
  - \( O(x^{22}) \) Taylor series (~23 flops)
  - ten-level polylogarithmic series (~306 flops)
- Needs \( NG \times NZ \) calls to exponential function.
- Each zone computed independently.
Polylogarithmic Approximation

\[ \Pi_L(x) = 1 + \frac{15}{\pi^4} \left[ -x^3 \sum_{l=1}^{L} \frac{e^{-lx}}{l} - 3x^2 \sum_{l=1}^{L} \frac{e^{-lx}}{l^2} - 6x \sum_{l=1}^{L} \frac{e^{-lx}}{l^3} - 6 \sum_{l=1}^{L} \frac{e^{-lx}}{l^4} \right] \]

Implementation Notes

• Rewrite of existing production kernel
  – Matches results of original.

• Double precision (10.9 GF for a 2.4 GHz Cell)

• C++, no inheritance or containers on SPU.

• “Kitchen-sink” SPU implementation, using:
  – SPU intrinsics and vector types,
  – manual loop unrolling, and
  – double buffered DMAs.
Vectorized Kernel Performance

2.1 GHz CBE vs. 2.8 GHz Xeon
Scalar Kernel Performance

2.1 GHz CBE vs. 2.8 GHz Xeon
Observations

- SPE performance heavily dependent on multi-buffering and vector instruction mix.

- Multi-buffering issues
  - Definitely need communication overlapping computation.
  - Need to tune data transfer sizes and frequency to reduce EIB and main memory contention.

- Instruction issues
  - Compiler conversion from scalar-to-vector is expensive.
  - Need mix of instructions to keep both pipelines busy.
  - Short function calls are bad.

- Static scheduling great to work with!
DMA Performance Tuning

Time required to complete a large DAXPY varying the number of SPEs (1-6,8) and the number of communication buffers (1-6).
Heterogeneous Computing Issues

• Want a low-level, C language interface for programming all common short-vector instruction sets (AltiVec, SSEn, SPU).

• Algorithm design needs to focus on minimizing data motion and maximizing flop rates.
  – Recompute.
  – Use more expensive methods on the same data.
  – Use single or mixed precision.

• Data structures need to be vector and communication friendly.
  – Data encapsulation/OO is bad!?!? SOA vs. AOS.
  – Pointer trees are bad.
  – Multidimensional arrays (esp. ragged right) are not easy.
Questions?