Experiences with Numerical Codes on the Cell Broadband Engine Architecture

M. Stürmer, D. Ritter, H. Köstler, and U. Rüde

System Simulation Group
Friedrich-Alexander-University Erlangen-Nuremberg

HipHaC 2008

November 8, 2008
The Cell Processor

Computational Fluid Dynamics

Image Processing

Molecular Dynamics

Conclusions
The Cell/B.E. and PowerXCell8i

Element Interconnect Bus

- ring bus
- aggregated transfer of 204.8 GB/s
The Cell/B.E. and PowerXCell8i

Memory Interface Controller

- 25.6 GB max. bandwidth
- Rambus XDR (Cell/B.E.) or DDR2 (PowerXCell8i) memory
The Cell/B.E. and PowerXCell8i

PowerPC Processor Element
- PowerPC core
- control execution and privileged tasks
- slow computation, low bandwidth
The Cell/B.E. and PowerXCell8i

Broadband Engine Interface

- non-coherent connection to I/O-devices
- coherent connection to another processor
The Cell/B.E. and PowerXCell8i

Synergistic Processor Element
- the **Synergistic Execution Unit** is a SIMD-only compute core
- operating on 256 kB of dedicated **Local Storage** and using its
- **Memory Flow Controller** for data transfer and communication
The Lattice Boltzmann Method

- fluid flow approximated by interaction of fictive particles
- domain divided into lattice cells (squares / cubes)
- two actions per time step
  - **streaming** data exchange with neighboring cells, but no computation
  - **collision** local computation
Optimizations

- memory layout
  - patch concept ($8 \times 8 \times 8$ lattice cells)
  - explicit copies of patch surfaces instead of “ghost layers”
  - carefully aligned

- optimized kernels
  - SIMD-vectorized
  - “select bit” instructions instead of conditional branches
  - unrolling, register blocking etc.

- ccNUMA-aware parallelization
Performance comparison: Straight-forward implementation in C against SPU-kernel for a $8^3$ patch. Tests run from cache or Local Storage without DMAs, respectively.
Performance comparison: Straight-forward implementation in C against SPU-kernel for a $8^3$ patch. Tests run from cache or Local Storage without DMAs, respectively.
Application Performance Results

Performance comparison: LBM by *LB-DC*\(^1\) on a x86_64 cluster node\(^2\) against Cell-optimized code for a real aneurysm geometry.

\(^1\)Lattice Boltzmann Development Consortium
\(^2\)2-socket Core2 Duo Xeon Woodcrest, Woody@RRZE
pdevc: a PDE-based video codec

1. original image
2. landmark selection
3. reconstruction
   (solution of PDE)
Reconstruction using Homogenous Diffusion, Nonlinear Isotropic Diffusion, and Nonlinear Anisotropic Diffusion using 65 $\omega$-RBGS or 130 $\omega$-Jacobi iterations (NAD).
FAS-Multigrid for Open-Boundary Poisson Problem in 3D

\[ \Delta \Phi(x) = f(x), \ x \in \mathbb{R}^3 \]

with \( \Phi(x) \to 0 \) for \( \|x\| \to \infty \)

coarsening increases size of \( \Omega \)

complex treatment of interfaces
Performance Results

Performance of the $\omega$-Jacobi smoother on one half of a QS20.
What is different when programming the CBEA

huge potential, but programming is more demanding than on common multicore architectures

Parallelization
sub-tasks must match the abilities of a certain unit

SIMD
has become default case, scalar operations are very expensive

Alignment
more restrictions, more influence on performance

Local Storage and DMA
“cache blocking and prefetching” for all data
Acknowledgements

- Jülich Supercomputing Centre (JSC)
- Regionales Rechenzentrum Erlangen, especially Dr. Thomas Zeiser
- Jan Götz and Christoph Freundl (LSS)

Thank you very much for your attention!