Unleashing the power of Cell/B.E. for HPC Applications

Virat Agarwal

Multicore Research

Georgia Tech College of Computing
Computational Science and Engineering
Graph Combinatorial problems
(List ranking, Irregular Algorithms)

Financial Modeling/Monte Carlo
(Options Pricing, CDO pricing, Risk Analytics)

Signal Processing/Data Compression
(FFTs, DFTs, Zlib)

Streaming technologies
(Text Search)
• We have thoroughly enjoyed getting our hands dirty.
  – SDK/Compilers evolved … => 1.0 … => 2.0 … => 3.1 …
  – Performance tools: Simulator, Counters, Assembly Visualizer

• ‘Architecture people’ like to talk about *heterogeneity*.

• We understand the necessity for programming models/frameworks.

  ![heterogeneity diagram]

  *heterogeneity*

  reduce It ! OR/AND hide It !
Cell Messaging Layer
(hides heterogeneity by making the system look like a homogeneous cluster of vector units)

IBM
XL C/C++ for Multicore Acceleration
(single source compiler, improves programmability, and opens the architecture for widely adopted framework)

RapidMind
(provides an easy API to build portable high performance applications for multicore processors)

IBM
ALF/DACS
(hides heterogeneity for programmers by making x86 as host and SPU as accelerators)

Others: CellSs (BSC), Sequoia (Stanford), Charm++ (UIUC), CorePy (IU)
Cell Messaging Layer
(hides heterogeneity by making the system look like a homogenous cluster of vector units)

**P** : Programmability

**P** : Performance

**P** : Predictability

**P** : Portability

Others: C CellISs (BSC), Sequoia (Stanford), Charm++ (UIUC), CorePy (IU)