Accelerating Forward Modeling by CPU,GPU and MIC

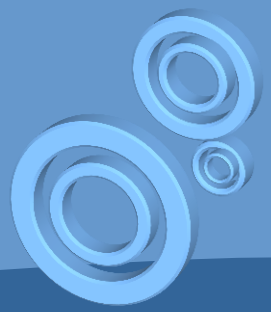
You, Yang and Fu, Haohuan
you-y12@mails.tsinghua.edu.cn
Department of Computer Science & Technology
Tsinghua University, Beijing, China

Outline

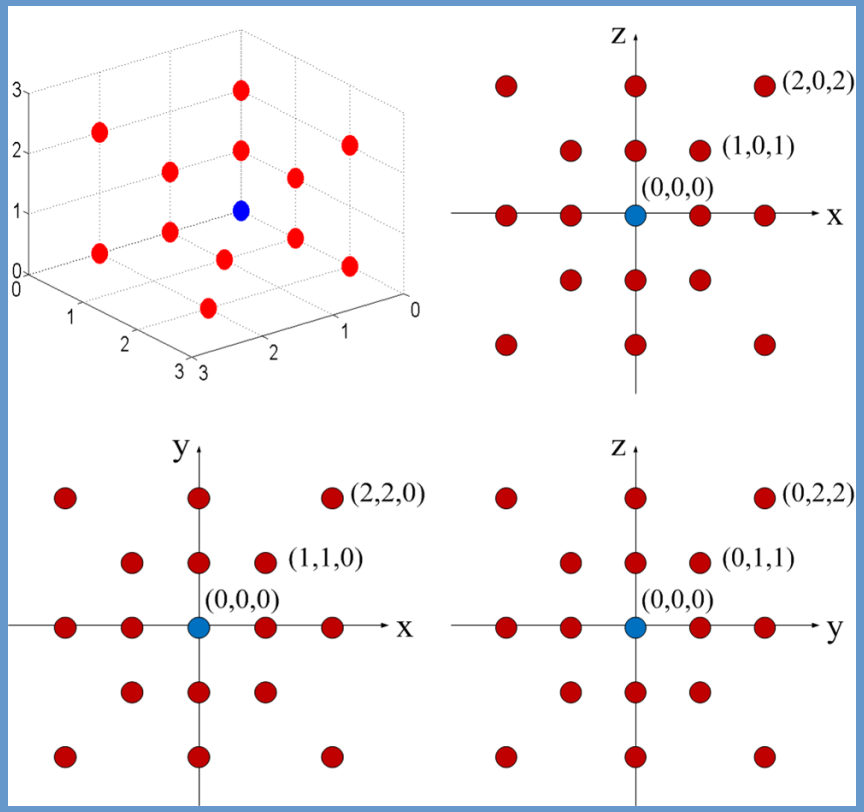
1. *Stencil*
2. *Architectures*
3. *Optimization Techniques*
4. *Experimental results*

Introduction

- Forward Modeling
  - Oil and Gas Exploration
- Iterative Stencil Loops
  - 99% of the time
- Lax-Wendroff Correction (LWC)
  - Dablain, 1986



LWC stencil



- Memory Access
  - (36+1+1)*3=114
- Operations
  - 228
- Flop to Byte Ratio
  - 0.25 ~ 9.5 (double)
  - 0.5 ~ 19 (single)

Outline

1. *Stencil*
2. *Architectures*
3. *Optimization Techniques*
4. *Experimental results*

Architectures

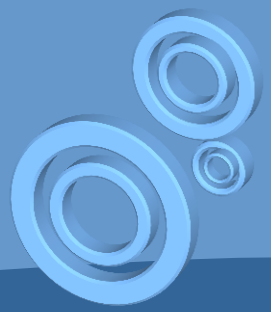
Architectures	Peak Performance (Double Precision)	Flop to Byte Ratio (Double Precision)
Intel Sandy Bridge CPU	256 Gflops	3.76
Intel Knights Corner MIC	1010 Gflops	6.35
Nvidia Fermi C2070 GPU	515 Gflops	5.31
Nvidia Kepler K20x GPU	1320 Gflops	7.02

Outline

1. *Stencil*
2. *Architectures*
3. *Optimization Techniques*
4. *Experimental results*

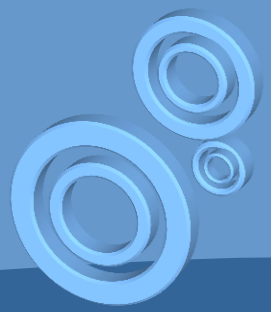
General techniques

- On-chip data reuse
  - algorithmic FBR < architectural FBR
  - 0.50 (stencil) < 21 (Kepler)
- Remove high-latency operations
  - e.g. divisions
- Get rid of branches
  - avoid discontinuities in SIMD/SIMT
- Continuous memory access
  - AOS to SOA



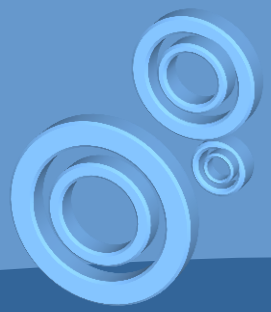
GPU: General Strategies

- One thread One point
  - High parallelism
- One thread One line
  - on-chip data reuse
- Common tricks
  - best L1/smem configuration
  - minimize PCIe communication
  - right block size
    - warp size
    - shared memory bank size



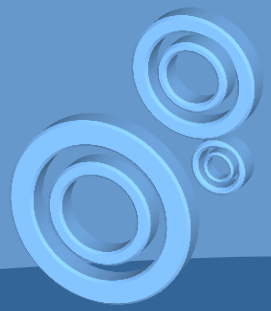
CPU/MIC: General Strategies

- Task parallelism
  - Inherent parallelism
- Data parallelism
  - SIMD
- On-chip data reuse
  - Cache blocking
- Offload and Native



CPU/MIC: Additional techniques

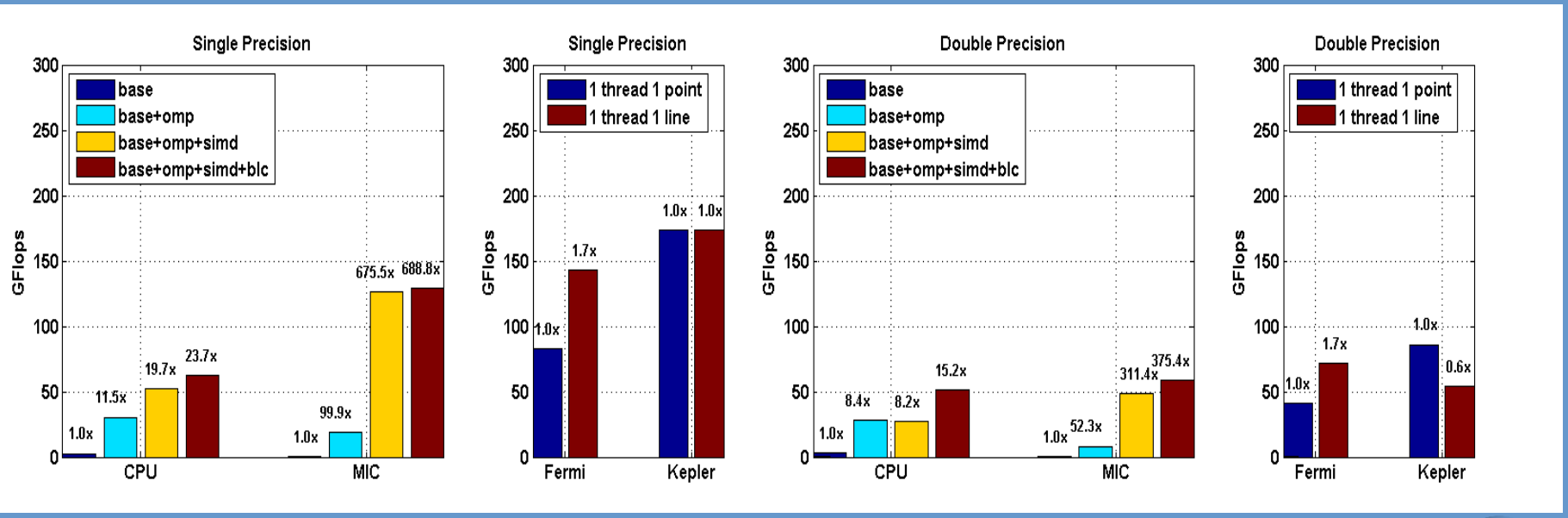
- Prefetch
  - memory to cache
- Schedule
  - thread and iteration
- Cilk array notation
  - replace compiler hints
- Cilk Plus
  - replace OpenMP



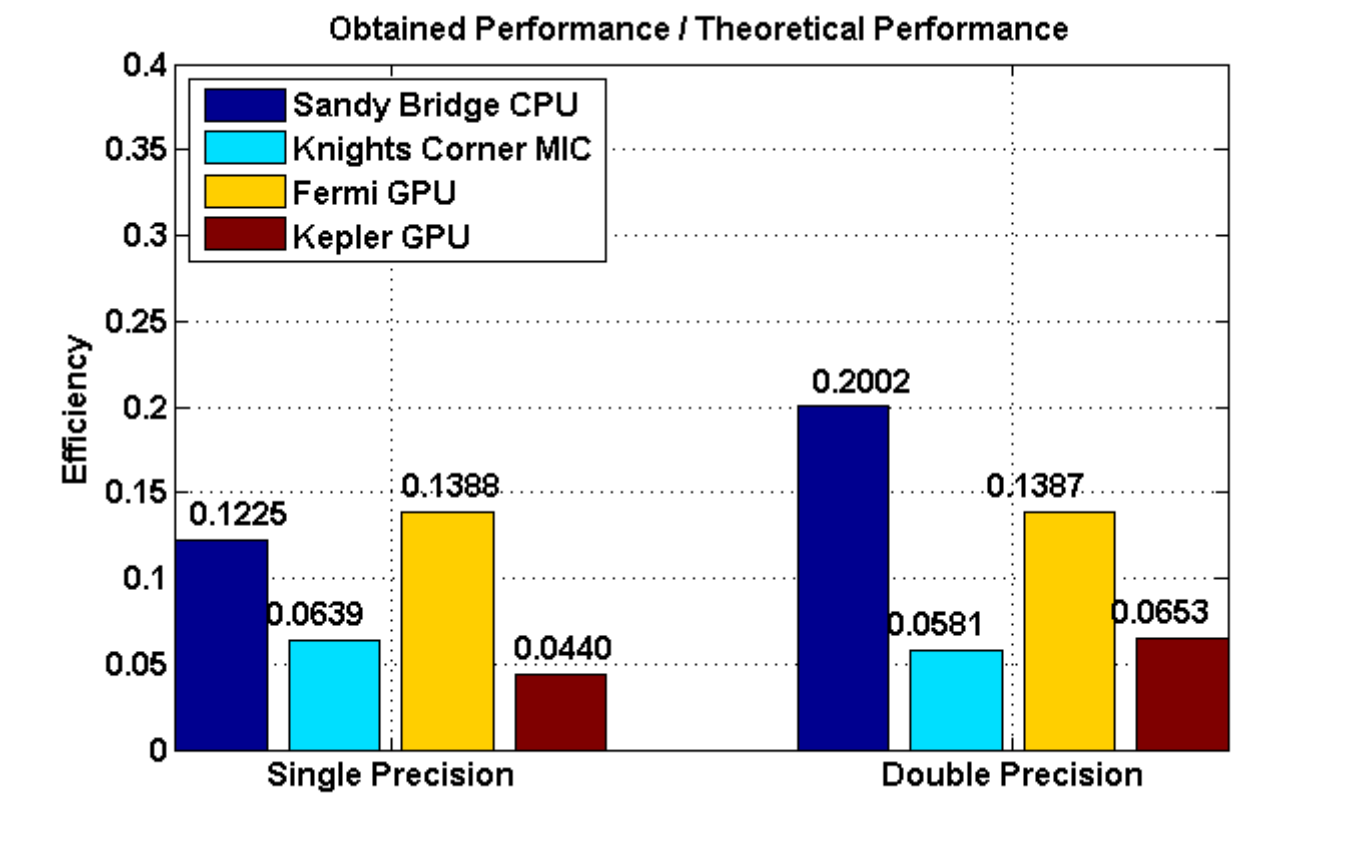
Outline

1. *Stencil*
2. *Architectures*
3. *Optimization Techniques*
4. *Experimental results*

Experimental results

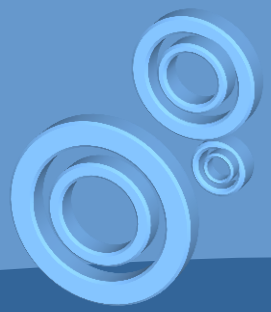


Performance Efficiency



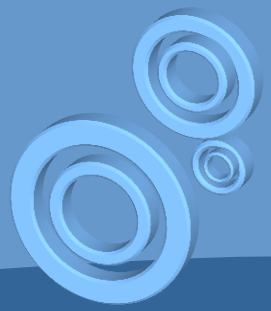
Knights Corner vs. Kepler

- User-controlled SM > Compiler-controlled L1
- Programmabilty
  - CUDA vs. Knights Corner Instructions
  - OpenACC vs. OpenMP/Cilk
- Performance/Effort (double precision)
  - Hard to evaluate
- Power Efficiency (whole workstation)
  - MIC: 0.2770 Gflops/watt (single precision)
  - GPU: 0.4133 Gflops/watt (single precision)



Future work

- Support Vector Machine
  - data mining
- Many-core and Multi-core architectures
- Welcome for discussion



Thank You

You, Yang and Fu, Haohuan
you-y12@mails.tsinghua.edu.cn
Department of Computer Science& Technology
Tsinghua University, Beijing, China