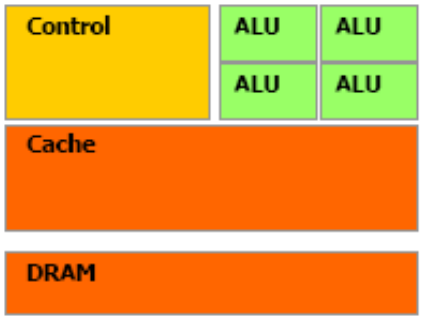
OpenMPC: OpenMP Extended for CUDA

Presented by

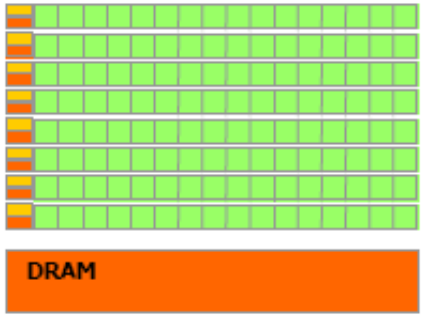
Seyong Lee
Jeffrey Vetter
Future Technologies Group
Oak Ridge National Laboratory

Rudolf Eigenmann
School of ECE, Purdue University

Tradeoff between Performance and Programmability in Many-Core Processors



(a) CPU



(b) GPU

Courtesy: NVIDIA

- GPU provides more computing power but worse programmability than CPU.
- GPU architectures are optimized for stream computations.
- General-Purpose GPUs (GPGPUs) provide better programmability for general applications.
  - CUDA programming model is more user-friendly than previous approaches, but still complex and error-prone.

OpenMPC (OpenMP extended for CUDA)

- OpenMPC = OpenMP + a new set of directives and environment variables for CUDA
- OpenMPC provides
  - A high level abstraction of the CUDA programming model (Programmability)
  - An easy tuning environment to generate CUDA programs in many optimization variants (Tunability)

OpenMPC Approach

- Use OpenMP for easier programming on CUDA-based GPGPUs.
- Provide various compile-time optimizations for performance.
- Extend OpenMP to allow fine-grained control of CUDA-related parameters and optimizations.

OpenMPC: Directive Extension and Environment Variables

- **OpenMPC Directive Format**

  #pragma cuda gpurun [clause [,] clause]…]

  #pragma cuda cpurun [clause [,] clause]…]

  #pragma cuda nogpurun

  #pragma cuda ainfo procname(pname) kernelid(kID)

- **OpenMPC Environment Variables**
  - **Control the program-level behavior of various optimizations or execution configurations for an output CUDA program.**

OpenMPC Code Example

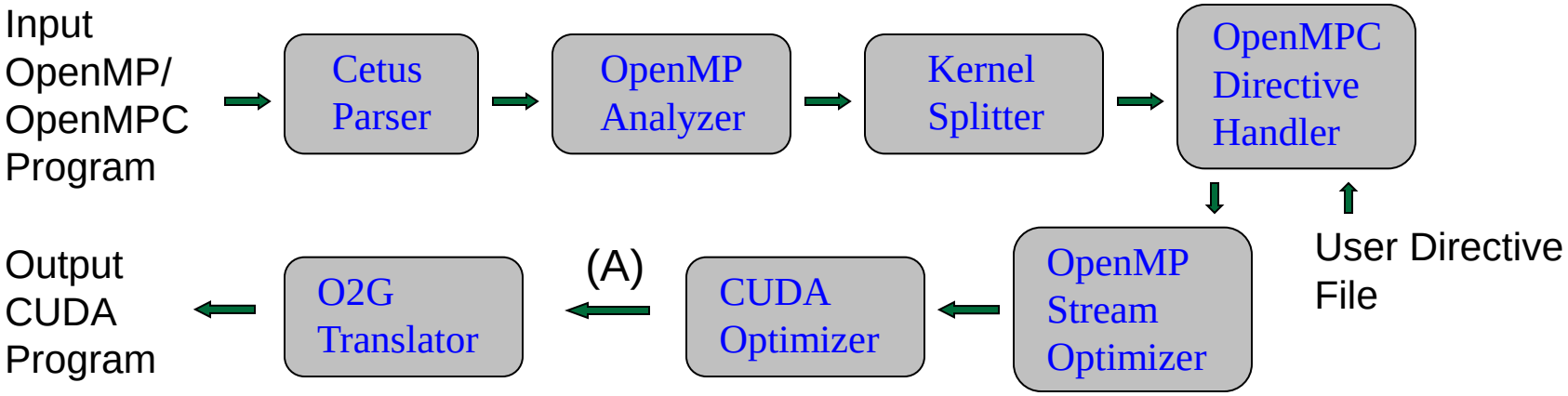
```
#pragma omp parallel shared(firstcol, lastcol, x, z) private(j) reduction(+: norm_temp11, norm_temp12)
#pragma cuda ainfo kernelid(1) procname(main)
#pragma cuda gpurun noc2gmemtr(x, z)  nocudamalloc(x, z) nocudafree(firstcol, lastcol, x, z)
#pragma cuda gpurun nog2cmemtr(firstcol, lastcol, x, z) sharedRO(firstcol, lastcol) texture(z)
{
   #pragma omp for private(j) nowait
   for (j=1; j<=((lastcol-firstcol)+1); j ++ ) {
      norm_temp11=(norm_temp11+(x[j]*z[j]));
      norm_temp12=(norm_temp12+(z[j]*z[j]));
   }
}
```

OpenMP directives:
inserted by a programmer

OpenMPC directives:
inserted by the OpenMPC compiler, but the programmer can alter them for fine tuning.

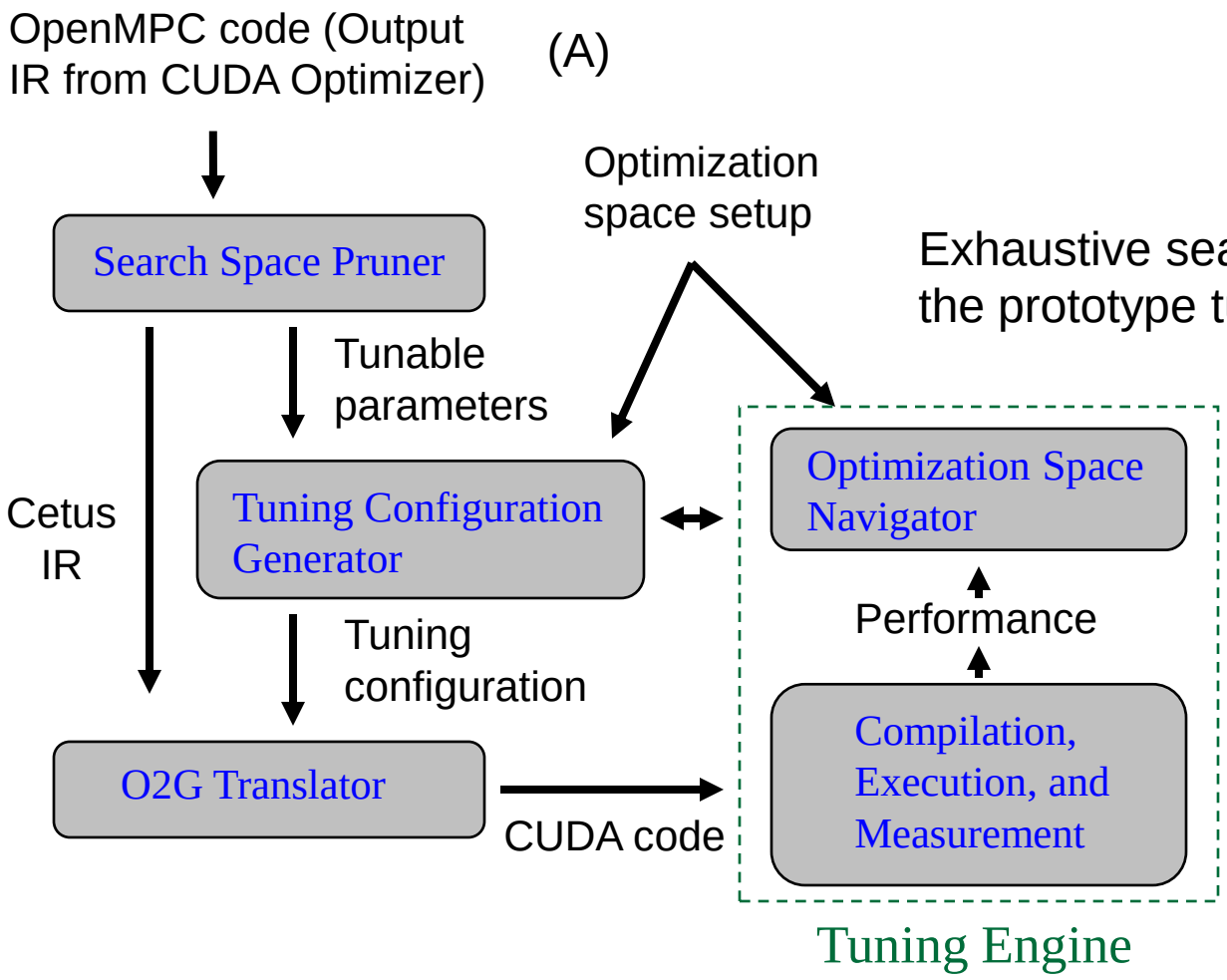
OpenMPC Compilation System

- **Overall Compilation Flow**



For automatic tuning, additional passes are invoked between *CUDA Optimizer* and *O2G Translator*, marked as (A) in the figure.

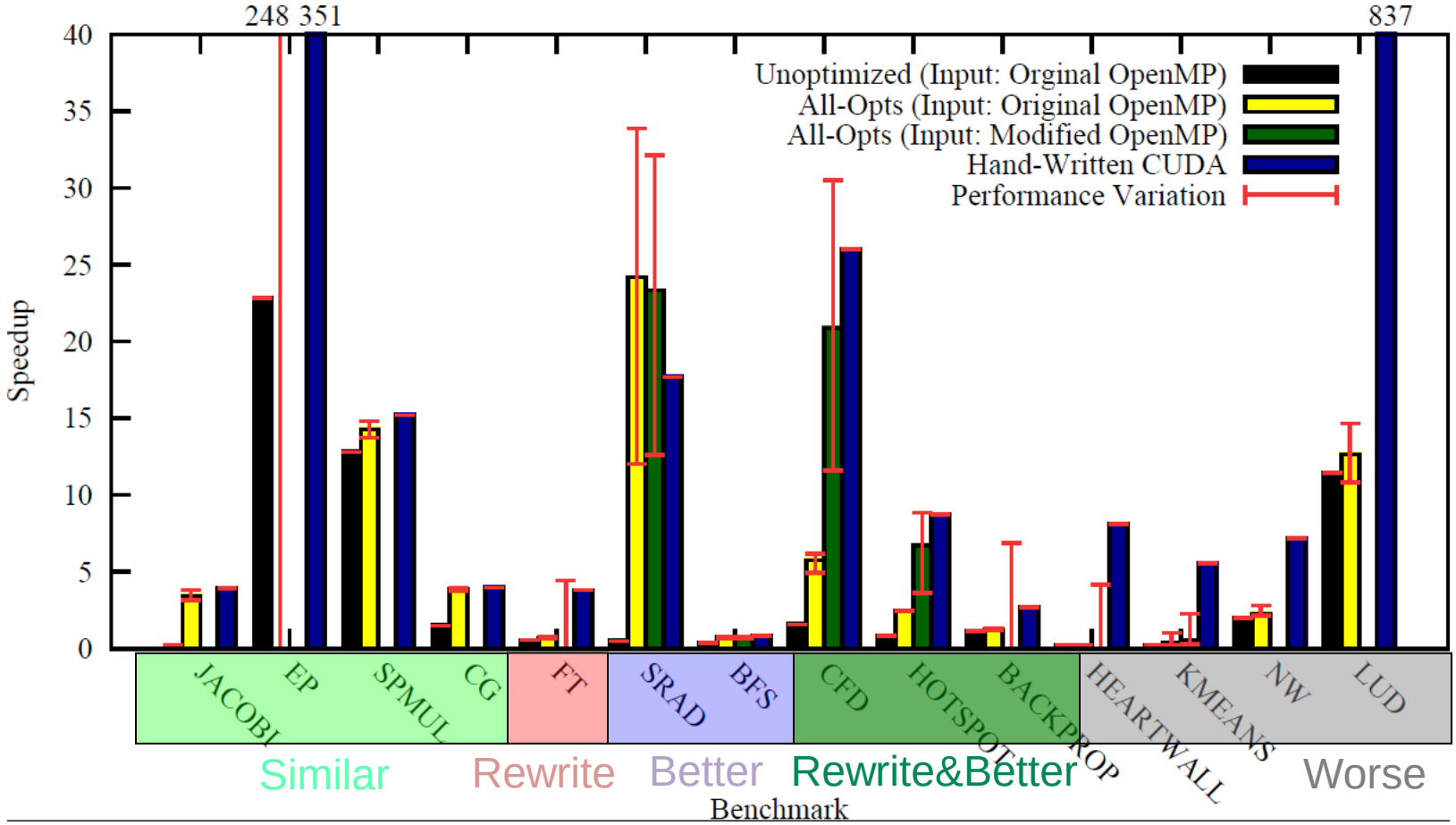
OpenMPC Tuning Framework



Exhaustive search was used in the prototype tuning system.

Programmers can replace the tuning engine with any custom engine.

Performance of OpenMP Programs on CUDA



- **Speedups are over serial on the CPU, when the largest available input data were used.**
- **Experimental Platform**: **CPU**: Dual-Core AMD Opteron at 3 GHz **GPU**: NVIDIA Quadro FX 5600 with 16 multiprocessors at 1.35GHz

Overall Tuning Performance

- **OpenMPC Performance Summary**

Translator Input	Performance Improvement over All-Opt Versions			Relative Performance over Manual Versions		
	MIN	MAX	AVG	MIN	MAX	AVG
Orig. OpenMP	1	4.23	1.19	0.02 (0.03)	1.92 (1.92)	0.5 (0.58)
Mod. OpenMP	1	7.71	1.24	0.02 (0.33)	2.68 (2.68)	0.75 (0.92)

In A(B) format, B refers the performance when the results of LUD are excluded.

- **Optimization Search Space Reduction by the Built-in Pruner**
  - **98.7% on average for program-level tuning**

References

Seyong Lee and Rudolf Eigenmann, OpenMPC: Extended OpenMP Programming and Tuning for GPUs, SC10: Proceedings of the 2010 ACM/IEEE conference on Supercomputing (Best Student Paper Award), November 2010

Seyong Lee, Seung-Jai Min, and Rudolf Eigenmann, OpenMP to GPGPU: A Compiler Framework for Automatic Translation and Optimization, Symposium on Principles and Practice of Parallel Programming (PPoPP09), February 2009

Contacts

Seyong Lee

Future Technologies Group
Computer Science and Mathematics Division
Oak Ridge National Laboratory
(865) 576-3869
lees2@ornl.gov

Rudolf Eigenmann

School of ECE, Purdue University
eigenman@purdue.edu