Productive Programing Systems for Contemporary Heterogeneous Systems

Jeffrey S. Vetter

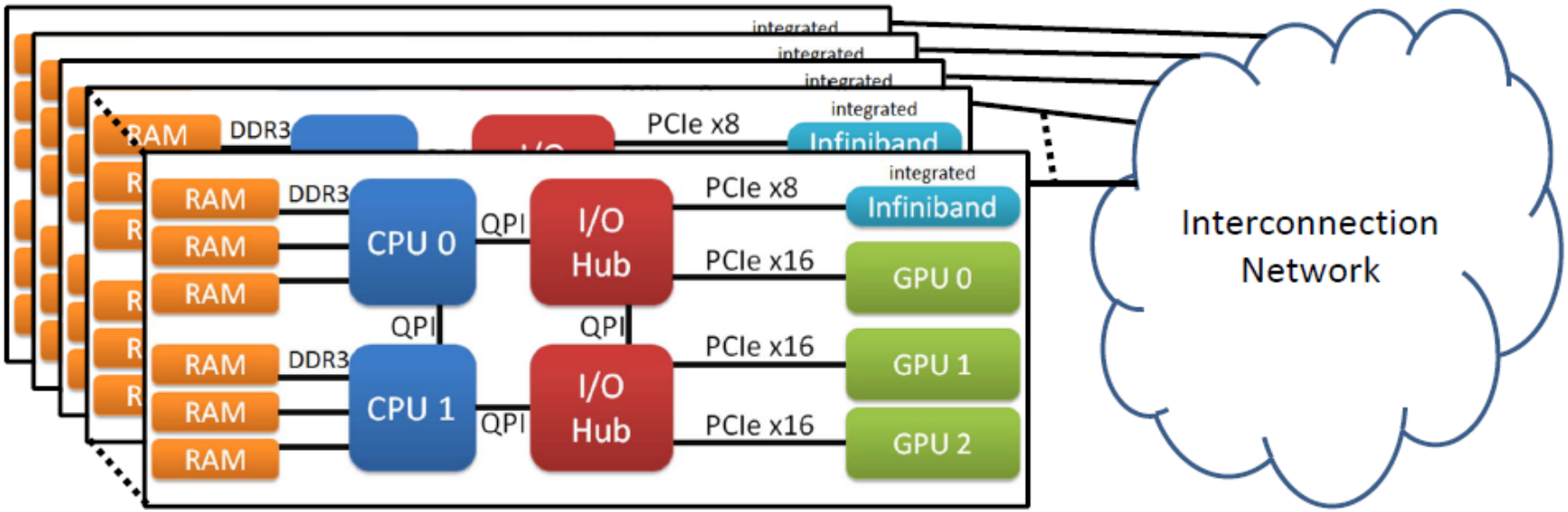
Presented to
Salishan Conference on High-Speed Computing
Gleneden Beach, Oregon
23 Apr 2014

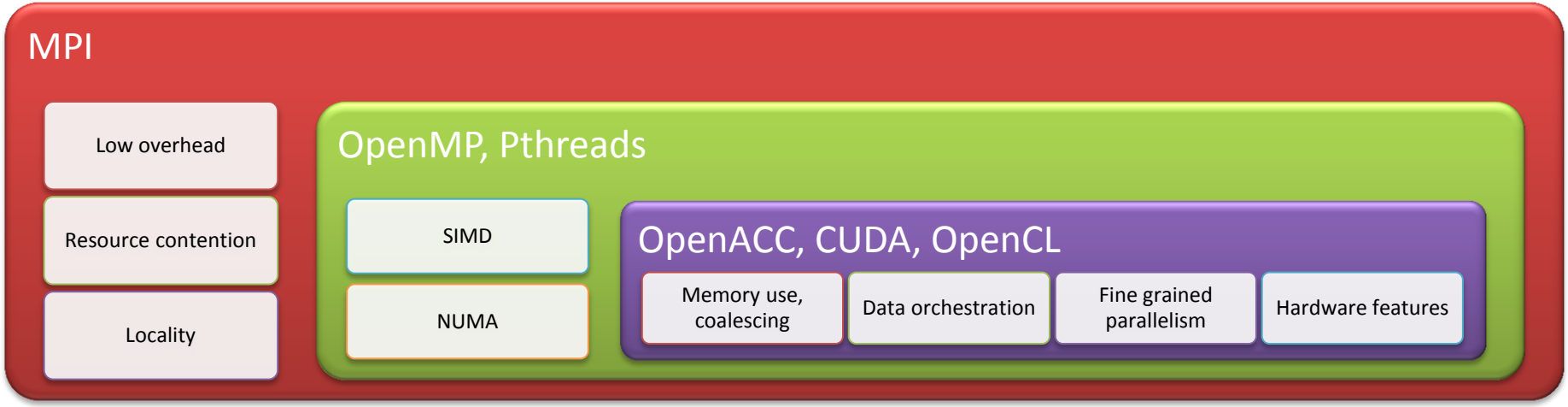
OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Georgia Tech College of Computing
Computational Science and Engineering

http://ft.ornl.gov ♦ vetter@computer.org

Applications must use a mix of programming models for these architectures





Realizing performance portability across heterogeneous architectures

- Heterogeneous architectures will continue to change over the next decade.
- Where are we today?
- How can we develop a 'write once, run anywhere efficiently' application?

Table 1: Comparison of Heterogeneous Architectures

Property	CUDA	GCN	MIC
Programming models	CUDA, OpenCL	OpenCL, C++ AMP	OpenCL, Cilk, TBB, LEO OpenMP
Thread Scheduling	Hardware	Hardware	Software
User Managed Cache	Yes	Yes	No
Global Synchronization	No	No	Yes
L2 Cache Type	Shared	Private per core	Private per core
L2 Total Size	upto 1.5MB	upto 0.5MB	25MB
L2 Line-size	128	64	64
L1 Data Cache	Read-only + Read-write	Read-only	Read-write
Native Mode	No	No	Yes

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

OpenARC Compiler and Runtime – Aiming for Portable Performance

"Write one program and run efficiently anywhere"

- OpenARC: Open Accelerator Research Compiler
  - Open-Sourced, High-Level Intermediate Representation (HIR)-Based, Extensible Compiler Framework.
    - Perform source-to-source translation from OpenACC C to target accelerator models.
      - Support full features of OpenACC V1.0 (+ array reductions and function calls)
      - Support both CUDA and OpenCL as target accelerator models
      - Supports OpenMP3
  - Provide common runtime APIs for various back-ends
  - Can be used as a research framework for various study on directive-based accelerator computing.
    - Built on top of Cetus compiler framework, equipped with various advanced analysis/transformation passes and built-in tuning tools.
    - OpenARC's IR provides an AST-like syntactic view of the source program, easy to understand, access, and transform the input program.
  - Building common high level IR that includes constructs for parallelism, data movement, etc

S. Lee and J.S. Vetter, "OpenARC: Open Accelerator Research Compiler for Directive-Based, Efficient Heterogeneous Computing," in *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC). Vancouver: ACM, 2014*

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

OpenARC translation and runtime using HeteroIR

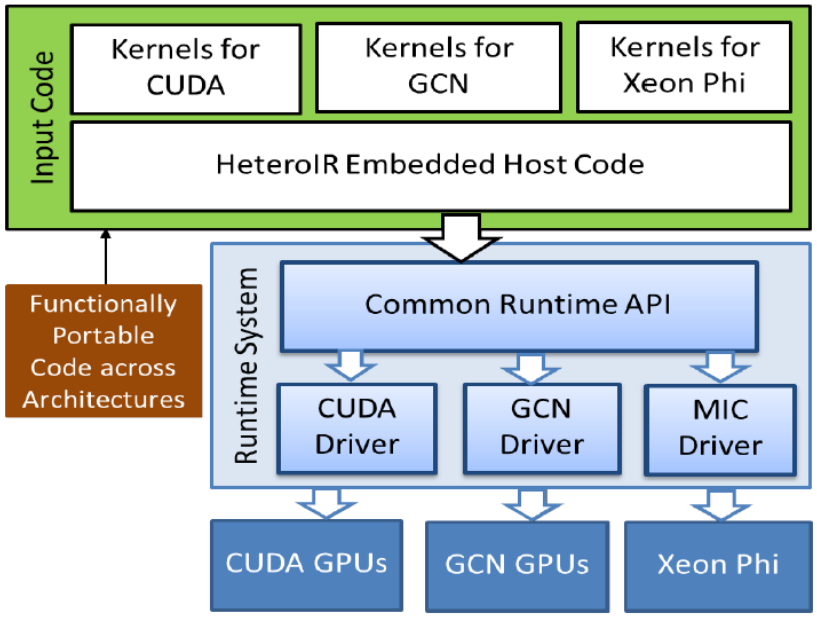


Fig. 3: HeteroIR Runtime System

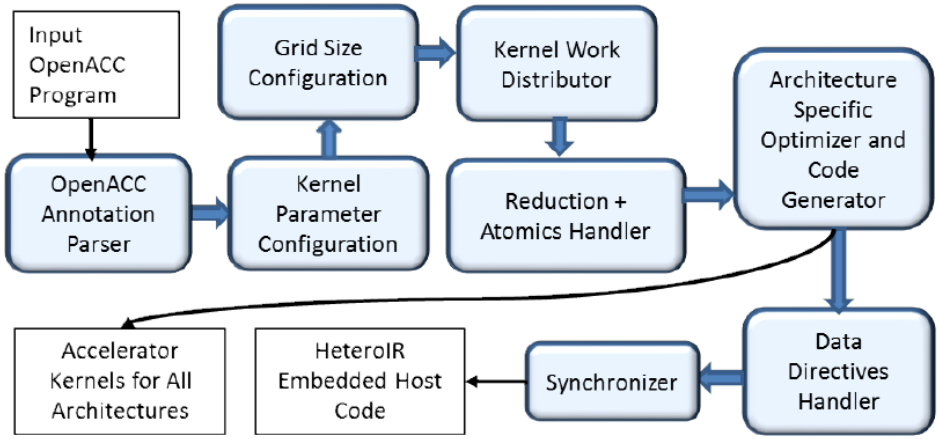


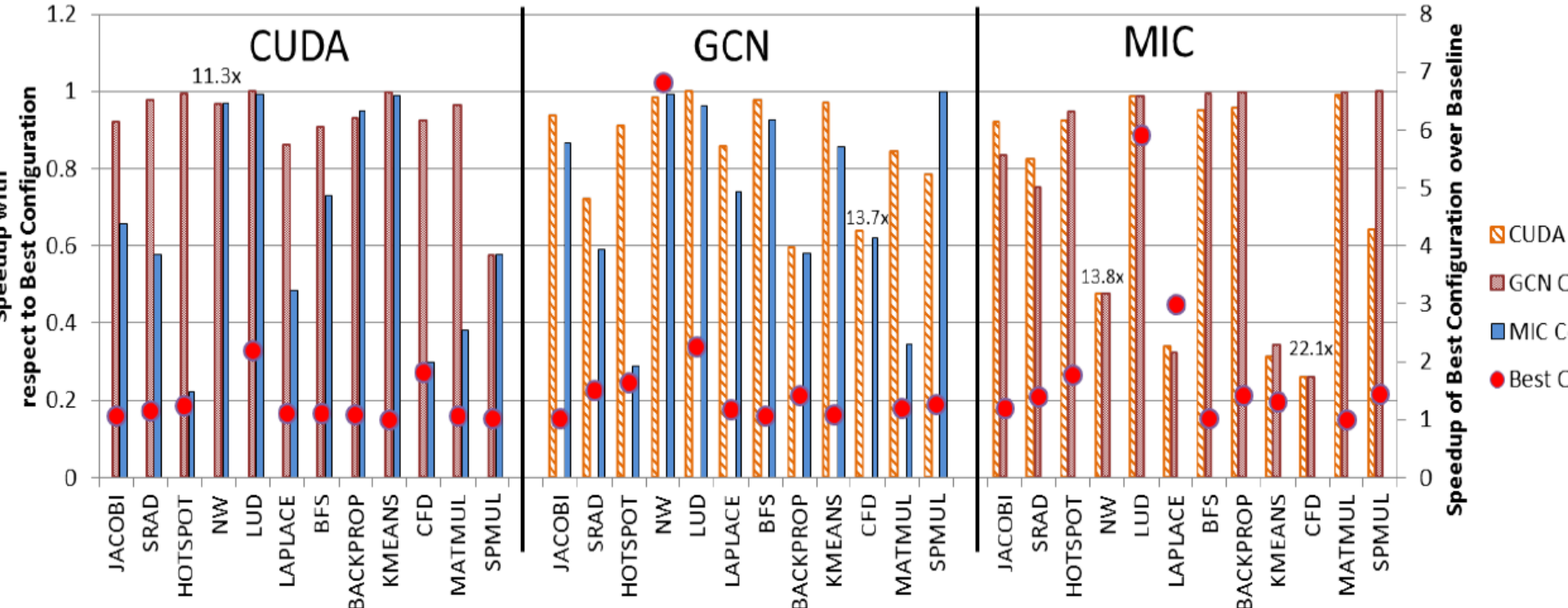
Fig. 1: OpenARC Translation Process

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

Performance Portability is critical and challenging

Best Program version of	Executed on: CUDA	GCN	MIC
CUDA	100	84	65
GCN	91	100	67
MIC	58	68	100

- One 'best configuration' on other architectures
- Major differences
  - Parallelism arrangement
  - Device-specific memory
  - Other arch optimizations



Automating selection of optimizations based on machine model

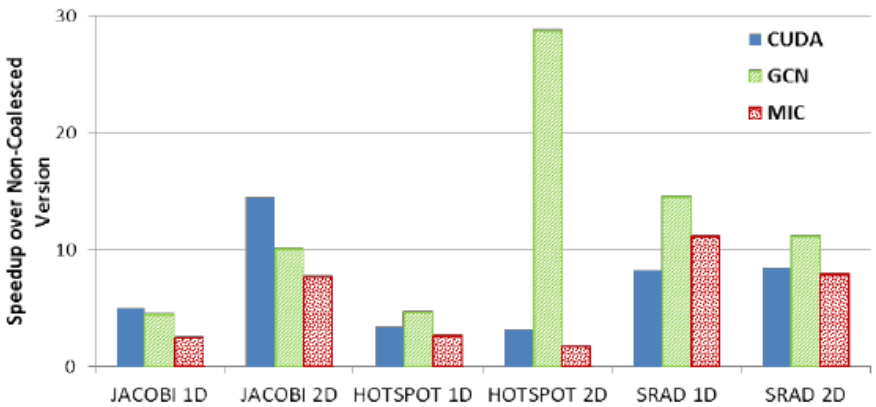


Figure 5: Memory Coalescing Benefits on Different Architectures : MIC is impacted the least by the non-coalesced accesses

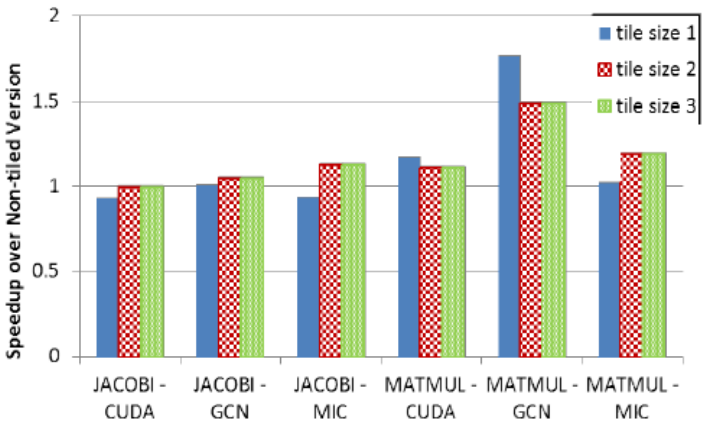


Figure 7: Impact of Tiling Transformation : *MATMUL* shows higher benefits than *JACOBI* owing to more contiguous accesses

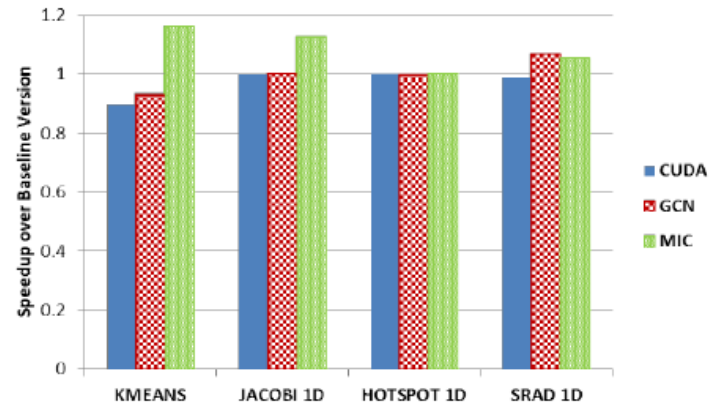


Figure 9: Effects of Loop Unrolling - MIC shows benefits on unrolling

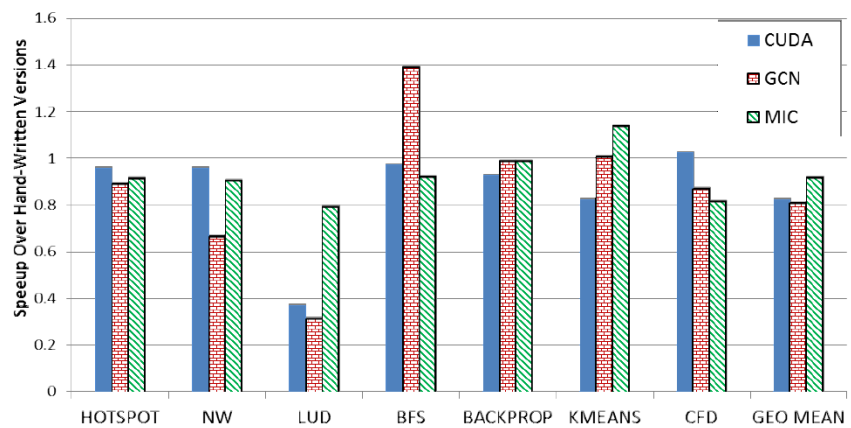


Fig. 11: Comparison of hand-written CUDA/OpenCL programs against auto-tuned OpenARC code versions : Tuned OpenACC programs perform reasonably well against hand-written codes

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

Parameterized application performance models for runtime optimization

Informing Runtime Optimization

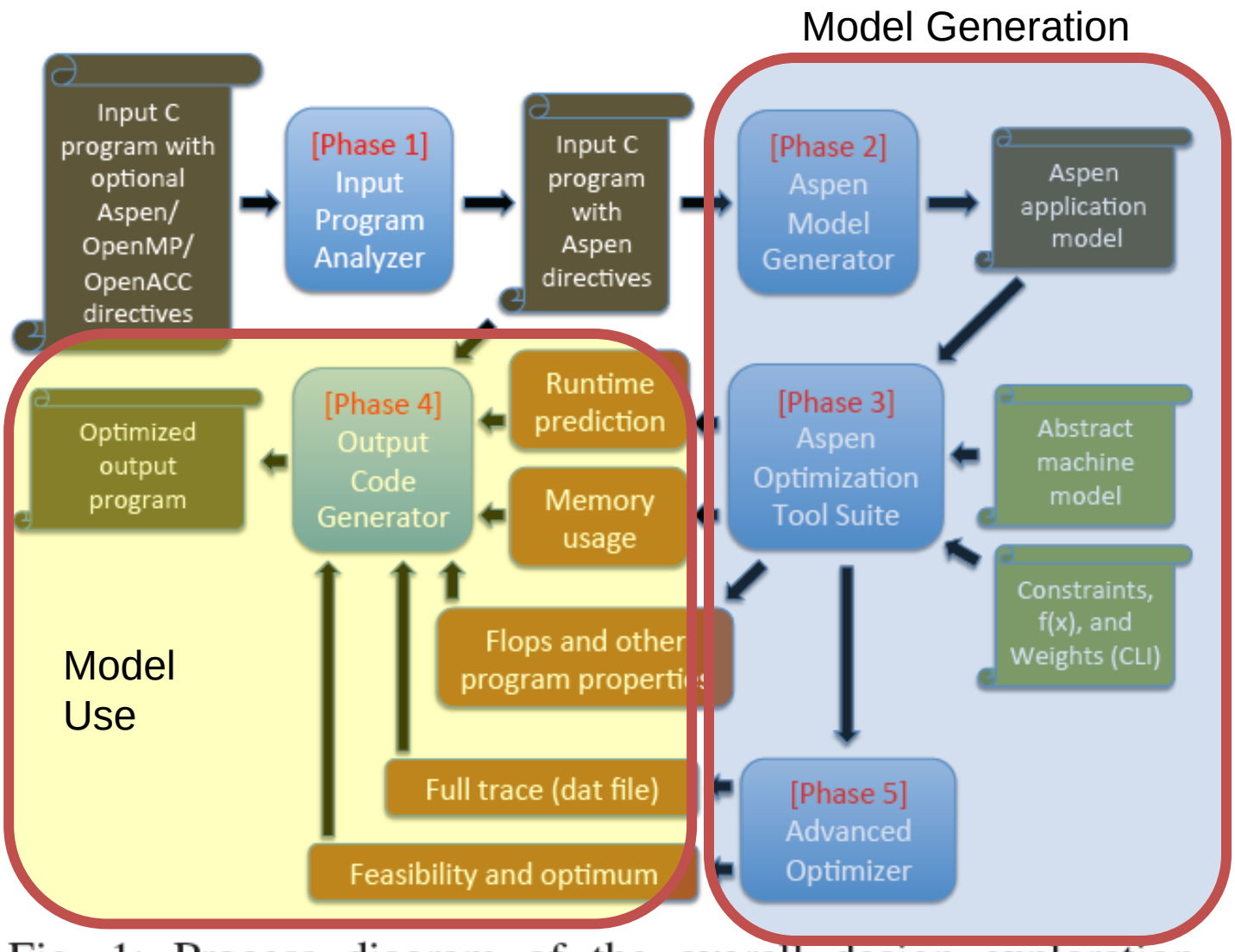


Fig. 1: Process diagram of the overall design exploration workflow.

Aspen Performance Models Drive Runtime Decisions

Listing 5: Input MM code modified to selectively offload OpenACC compute regions to GPUs based on the Aspen prediction

```
extern int HI_aspenpredict(double N); int N = 1024;
void matmul(float * A, float * B, float * C) { int i,j,k;
#pragma acc kernels loop if(HI_aspenpredict((double)N)) \
 gang copyin(N, B[0:N*N], C[0:N*N]) copyout(A[0:N*N])
  for (i=0; i<N; i ++ ) {
#pragma acc  loop worker
    for (j=0; j<N; j ++ ) { ... }
  }; return ;
} //end of matmul()
int main() { ... }
```


OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

LULESH – runtime optimizations

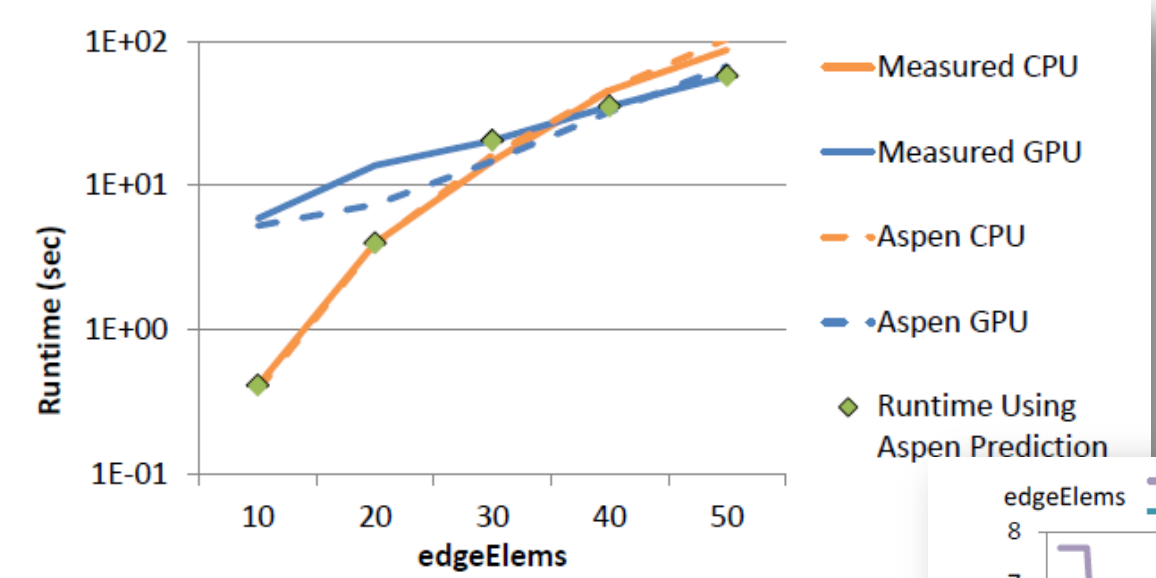


Fig. 7: Measured and predicted runtime of the entir program on CPU and GPU, including measured run the automatically predicted optimal target device a

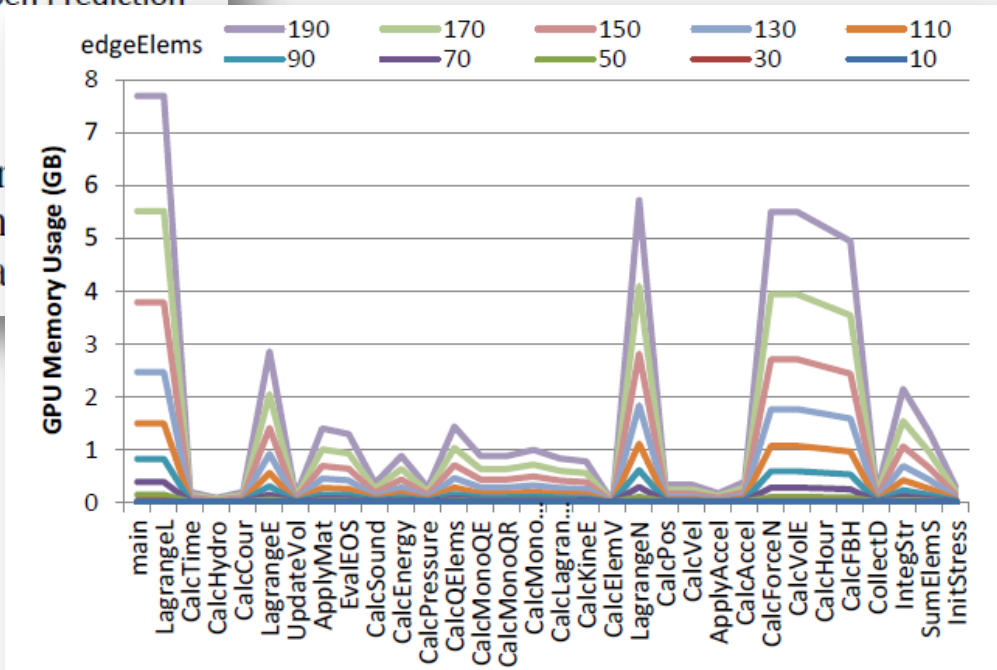


Fig. 8: GPU Memory Usage of each Function in LULESH, where the memory usage of a function is inclusive; value for a parent function includes data accessed by its child functions in the call graph.

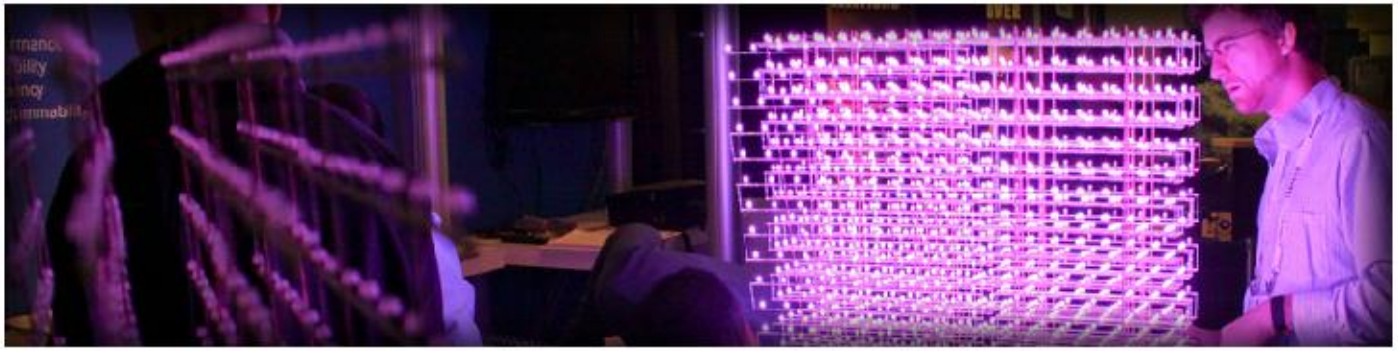
Summary

- Goals
  - Performance portability
  - Write once, run anywhere efficiently
- Activities
  - OpenARC portable compiler and runtime
  - HeteroIR allows mapping to various architectures
  - autoAspen allows performance modeling and runtime optimization

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY

Bonus

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE U.S. DEPARTMENT OF ENERGY



Emerging Technologies

Submissions due July 31, 2014

In the Emerging Technologies Track of the Technical Program, SC14 will provide a showcase for high-risk, high-reward hardware and software technologies that may significantly change the world of HPC in the next ten years. For example, technologies like reconfigurable computing, new SoC designs, alternative programming systems, and novel cooling techniques may offer near-term benefits, while new device technologies, like carbon nanotubes, non-volatile memory, quantum computing, and chip-level optical interconnects offer potentially paradigm-changing benefits over the long- term.

The SC14 Emerging Technologies Committee will select these technologies from submissions, based on a peer-review process with a single round of reviews. Successful projects will advocate future technologies with the potential to influence computing and society as a whole. Winners will receive free exhibit floor space in a high visibility location; this location will allow attendees the opportunity to witness technology demonstrations, see presentations, and hold in-depth technical discussions.

We invite submissions from industry, academia, and government researchers for this Emerging Technologies Exhibit Showcase. Submissions will be evaluated on several criteria including potential impact on performance, productivity, power, reliability, cost of HPC systems, and novelty. Submissions with exceptional merit may be offered the opportunity to present their work at an Emerging Technology Theatre on the exhibit floor. Submissions are accepted through the submission website (listed below). Each submission should describe the project scope in detail and provide references to external resources (newspaper articles, existing installations, webpages etc.).

Submissions are accepted through the submission website. Each submission should describe the project scope in detail and provide references to external resources (newspaper articles, webpages etc.). Additional materials, such as prototype photos are very welcome. Each project also needs to describe how much space it would require and how it would use the space provided (include any additional requirements such as A/V equipment or network connectivity). Projects will be selected based on a peer-review process with a single round of reviews.

http://sc14.supercomputing.org/program/emerging-technologies

SCHEDULE AND SUBMISSION PROCEDURE:
* Submission deadline: July 31, 2014
* Notification of acceptance: August 31, 2014

July 31st