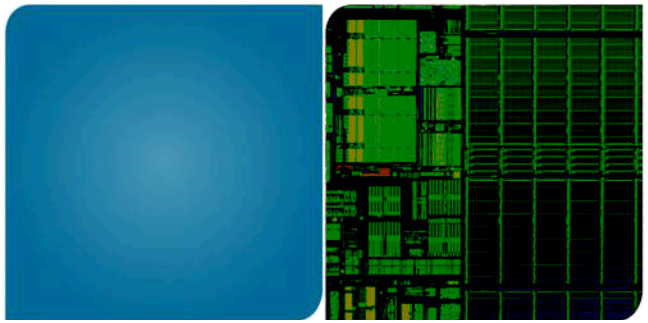
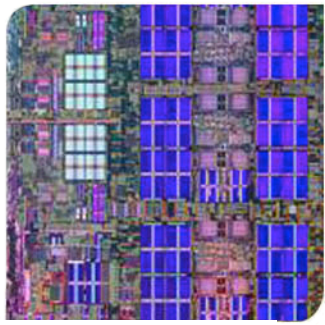




Integrated, Application-Level, Performance-Energy Modeling for Heterogeneous Architectures

PIs:Sudha Yalamanchili, **Hyesoon Kim,**

Students: Eric Anger, Prasun Gera,

Nagesh B. Lakshminarayana

Collaborators: Jeremiah J. Wilke, Patrick S McCormick,

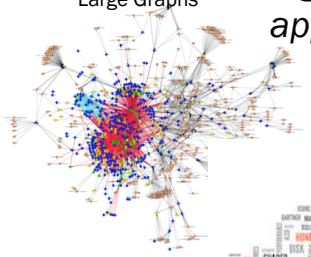

Sandia National Laboratories

Georgia Tech

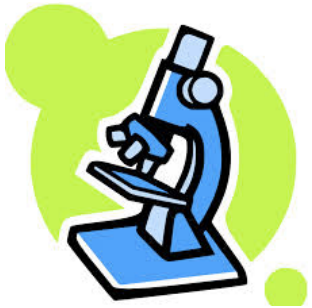
Los Alamos NATIONAL LABORATORY EST. 1943

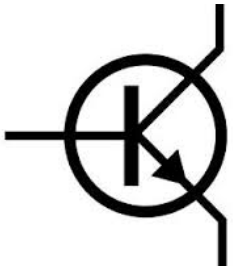
Goals

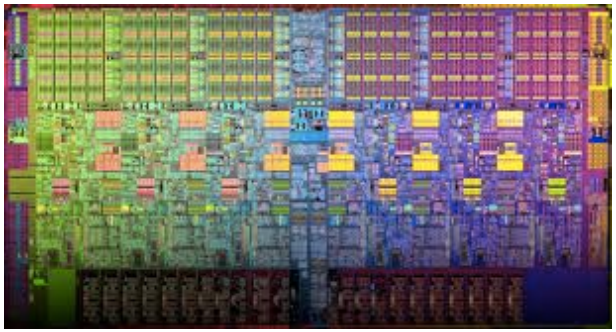
Large Graphs



New generation of applications







Needs: Fast simulation/profiling & Understanding high-level behaviors

Sandia National Laboratories

Los Alamos NATIONAL LABORATORY EST. 1943

Georgia Tech

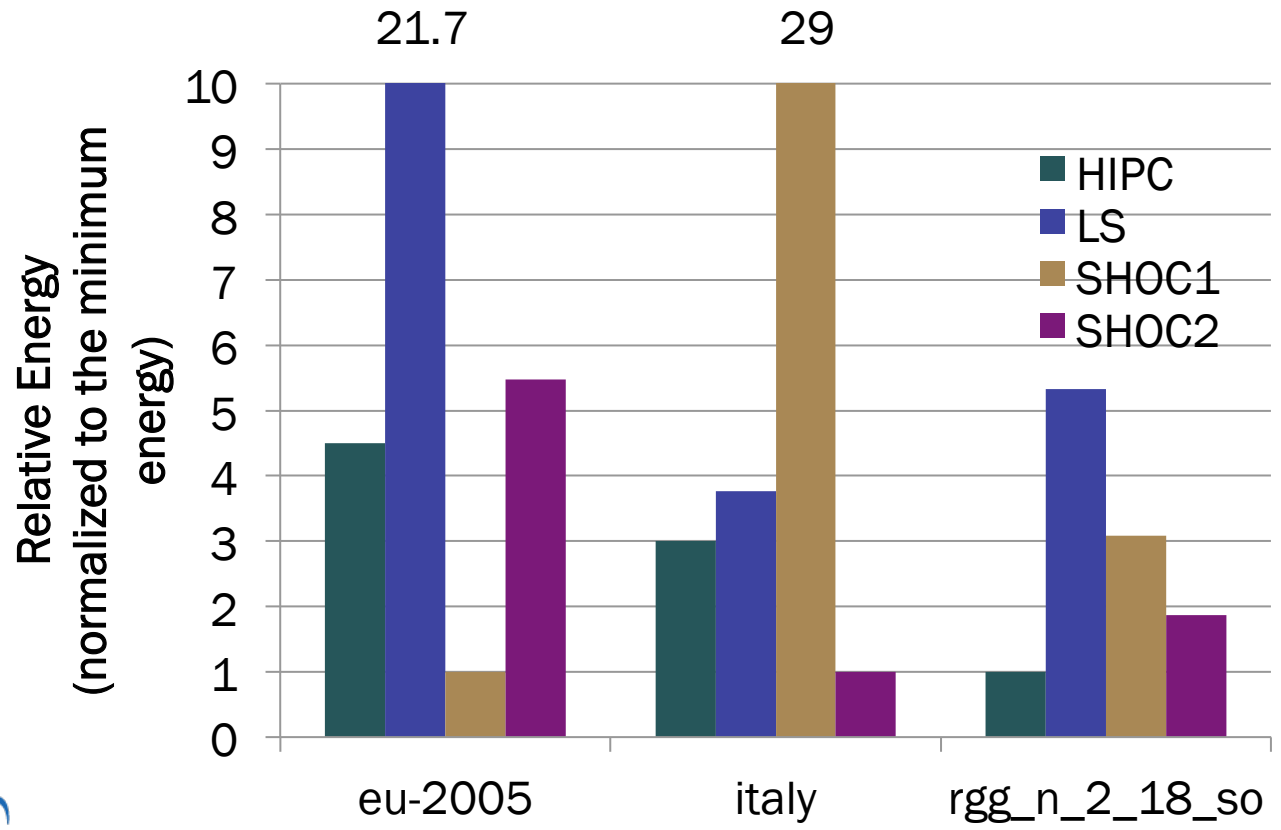
comparch

For whom?

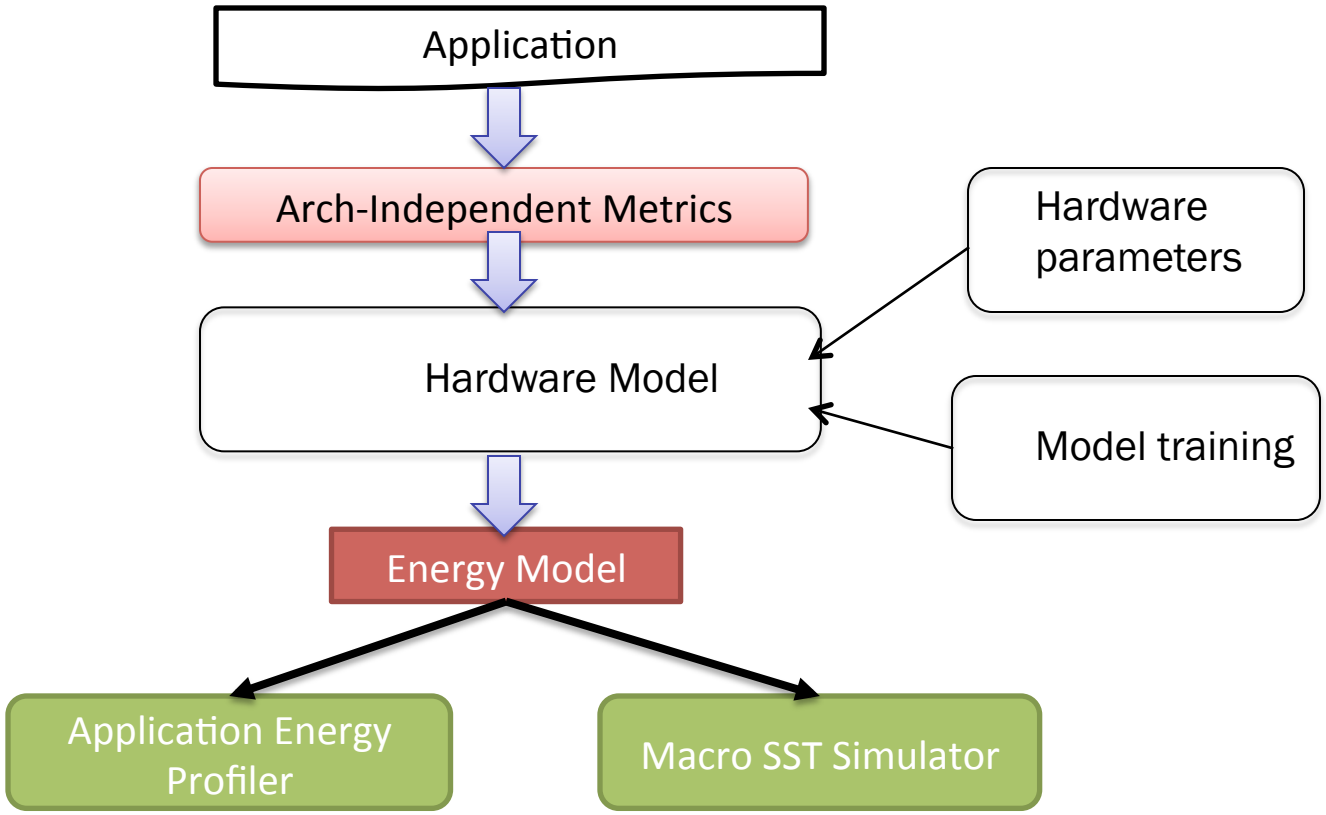
- Application developers
  - Application optimizations for different architectures
  - Algorithm selections
- Hardware developers
  - Architecture parameter decisions
  - Large scale hardware developers

Motivation

- NVIDIA GPU-K40
- BFS algorithm: different implementations, different inputs



Proposed Framework



- Fast and scalable simulation
- Application optimization guide

Sandia National Laboratories
Los Alamos NATIONAL LABORATORY EST. 1943
Georgia Tech
comparch

Application Level Energy Profiling

- Function level energy profiling
  - collect time and energy per function boundaries

Function Name	Time(s)	Energy(J)	Power Avg(w)	# Calls	%Time	%Energy
HPC_sparsemv(HPC_Sparse_Matrix_STRUCT*, double const*, double*)	33.27	917.11	27.56	150	79.58	80.82
waxpby(int, double, double const*, double, double const*, double*)	4.98	132.10	26.51	449	11.92	11.64
ddot(int, double const*, double const*, double*, double&)	2.46	62.17	25.28	298	5.88	5.48
generate_matrix(int, int, int, HPC_Sparse_Matrix_STRUCT**, double**, double**, double**)	1.08	23.21	21.54	1	2.58	2.05
_GLOBAL__sub_I_main	0.01	0.00	0.00	1	0.02	0.00
HPCCG(HPC_Sparse_Matrix_STRUCT*, double const*, double*, int, double, int&, double&, double*)	0.01	0.18	23.45	1	0.02	0.02
SUM	41.81	1134.77				

- Instruction level profiling
  - synchronizations, parallelism?



Profiling Mechanisms

- Profiling with Byfl*
  - From LANL
  - To collect hardware independent metrics
  - To help application developers
  - Instrumenting code in LLVM's immediate representation
  - Profiled information: All IR level information
    - Low-level primitives (barriers, synchronization information), computation per memory bytes etc.
- Profiling for fast and scalable hardware simulation
  - Application skeleton (more detail in later)
- Profiling for application understanding

Scott Pakin, Patrick McCormick, "Hardware-independent application characterization," IISWC 2013

https://github.com/losalamos/Byfl

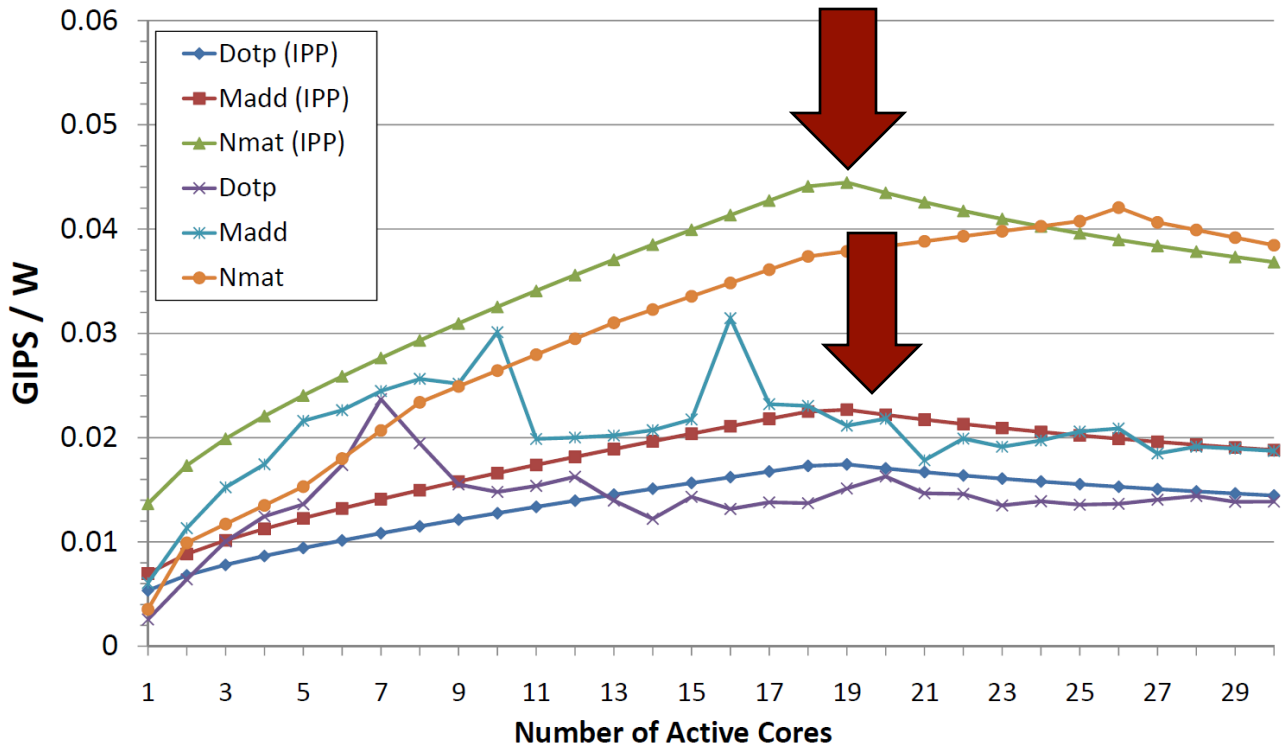


Why Architecture Independent Metrics?

- To get high-level information
  - Synchronization overhead?
  - # of data accesses
  - Data movements
  - Leading to more software level optimization decisions.
- Separate the hardware dependent overhead and software caused overhead

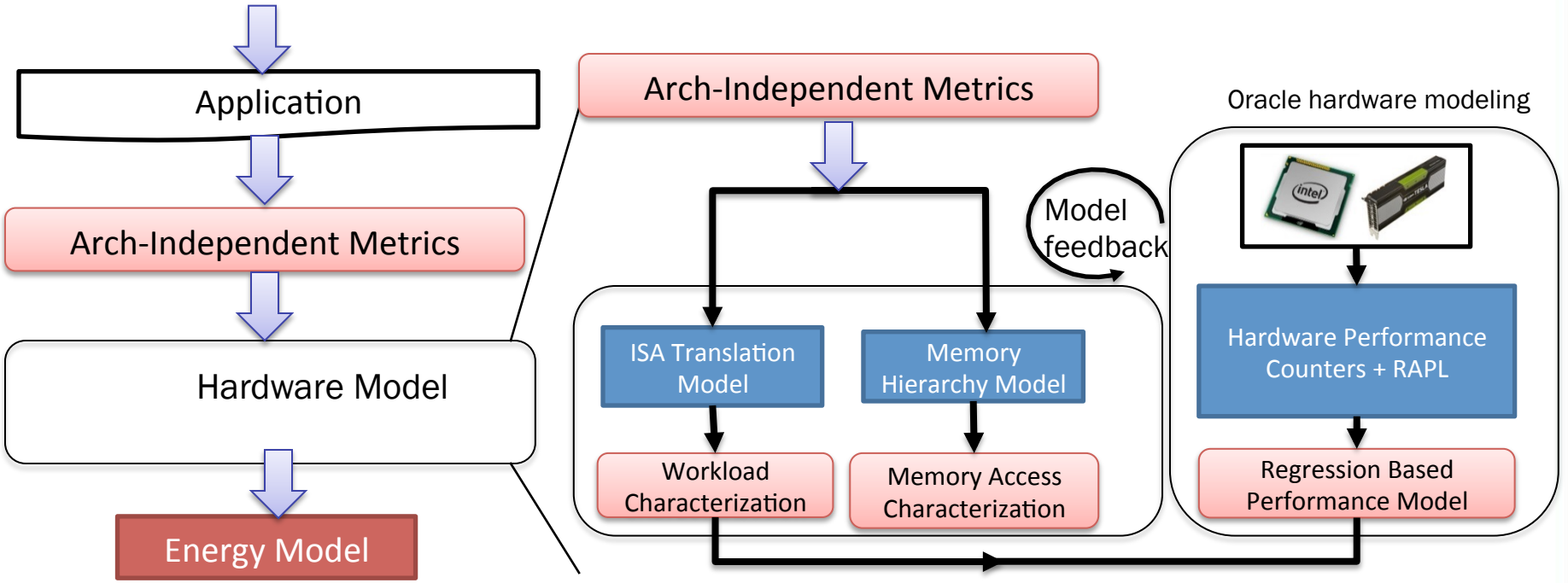
Eg) Power Efficiency and TLP

peak power efficient point



source: An Integrated GPU Power and Performance Model, ISCA'10

Hardware Modeling



Sandia National Laboratories

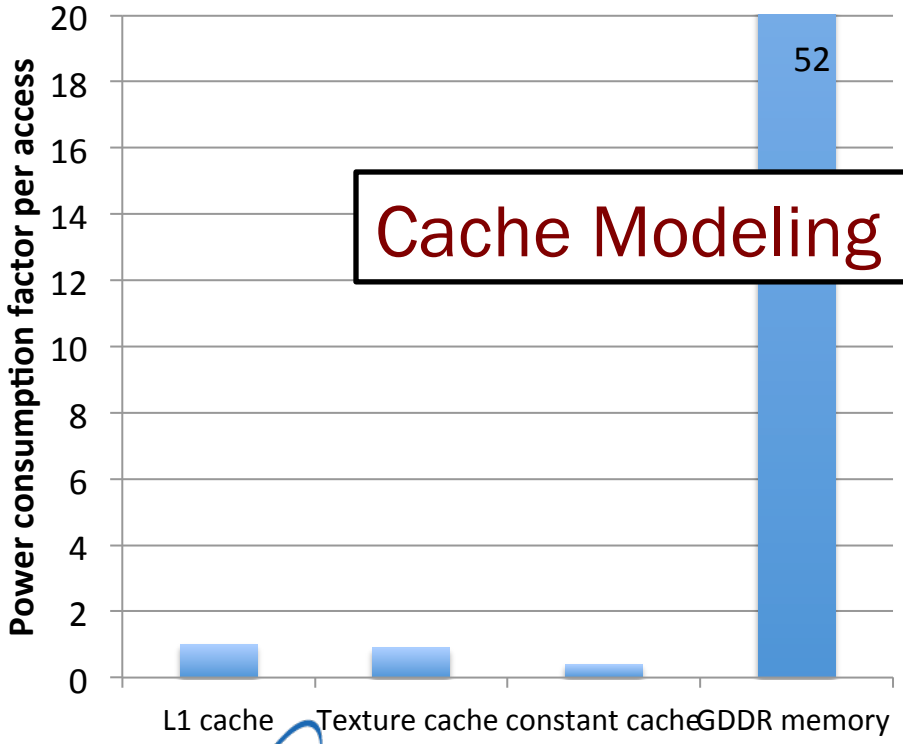
Los Alamos NATIONAL LABORATORY EST. 1943

Georgia Tech

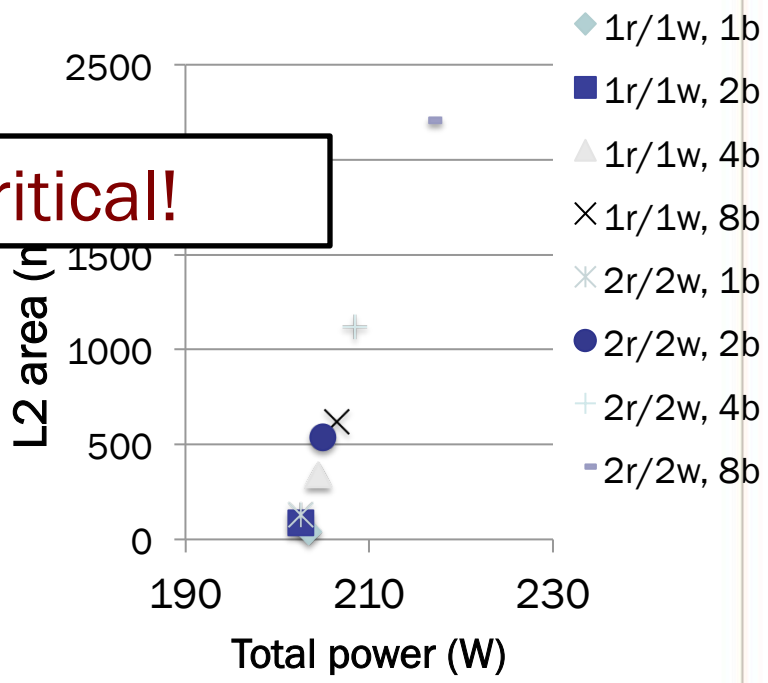
comparch

Power modeling with Application metrics

- Now, we need to model architecture components
- Not all memory instructions are equal!!!



Cache Modeling is critical!



source: An Integrated GPU Power and Performance Model, ISCA'10

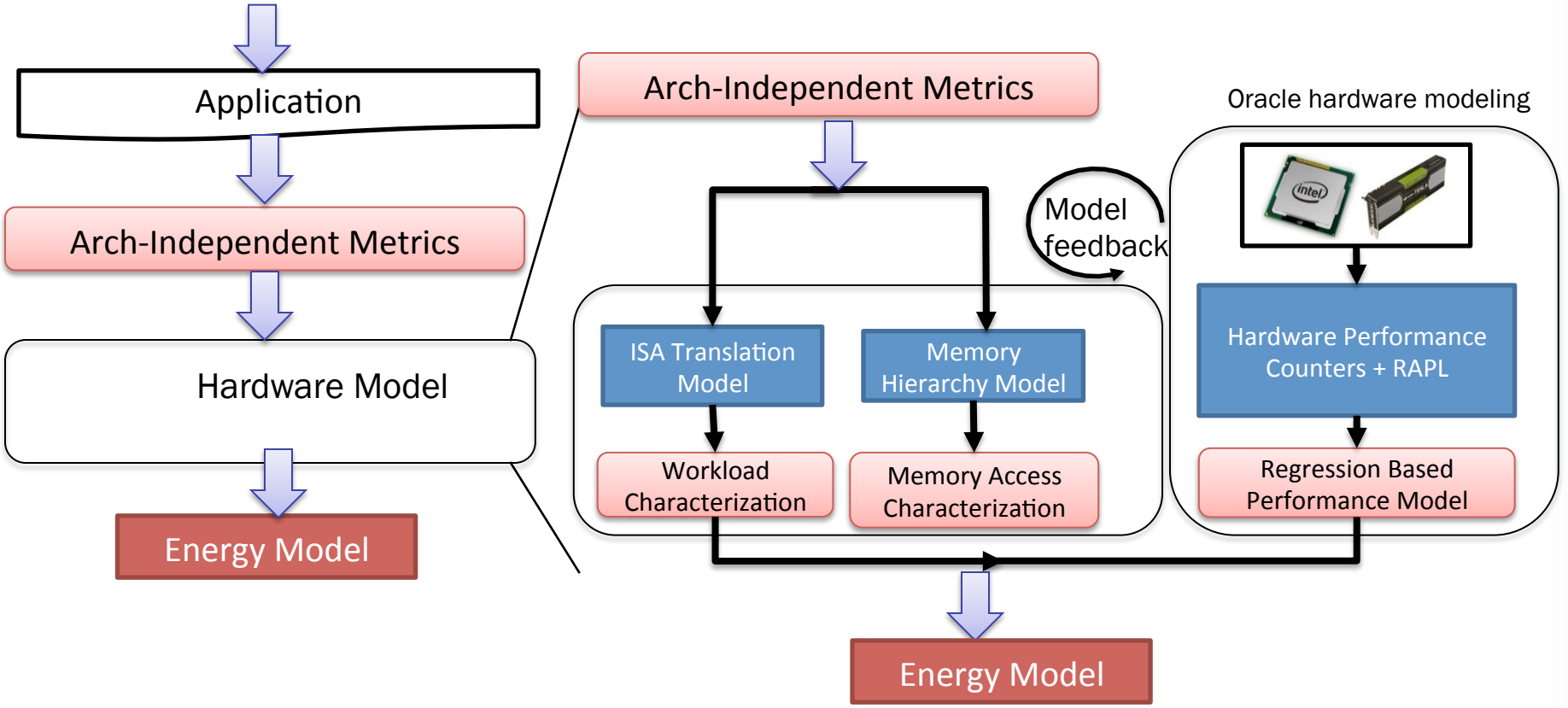
Sandia National Laboratories

Los Alamos NATIONAL LABORATORY EST. 1943

Georgia Tech

comparch

Hardware Model Stage





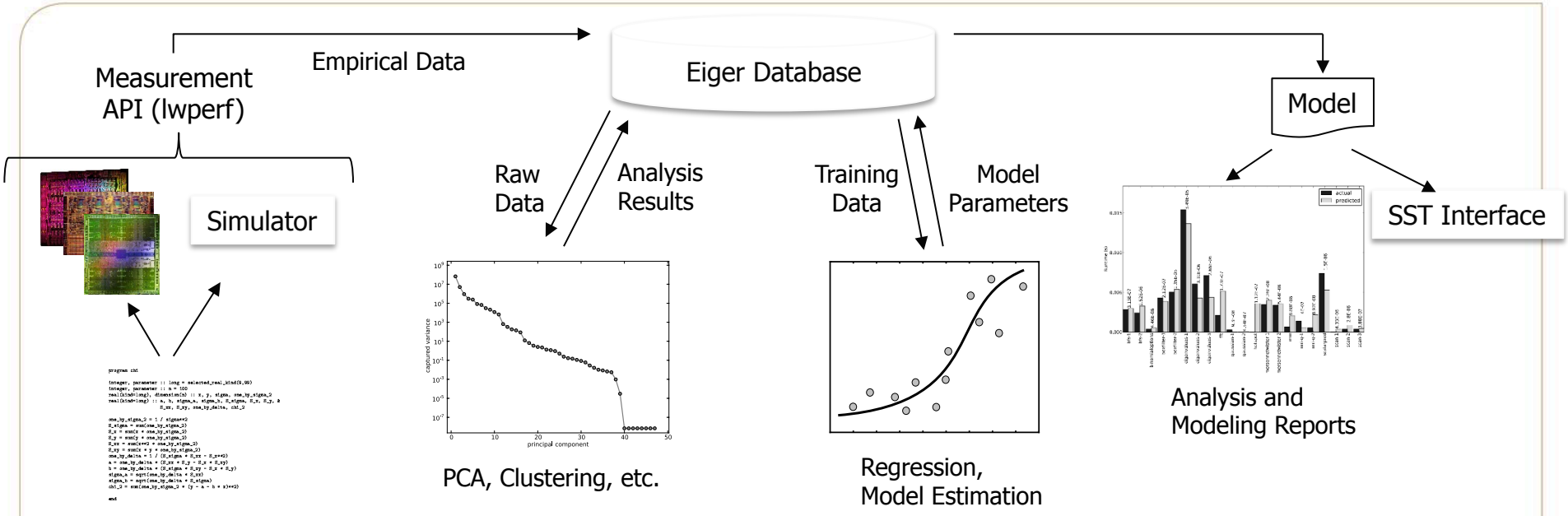
Training Power Model

- Collect power numbers using RAPL
  - Read hardware performance counters and memory power consumption values
  - Integrate RAPL calls from LLVM
- Regression based power modeling
  - Eiger

RAPL, https://01.org/blogs/tlcounts/2014/running-average-power-limit-%E2%80%93-rapl

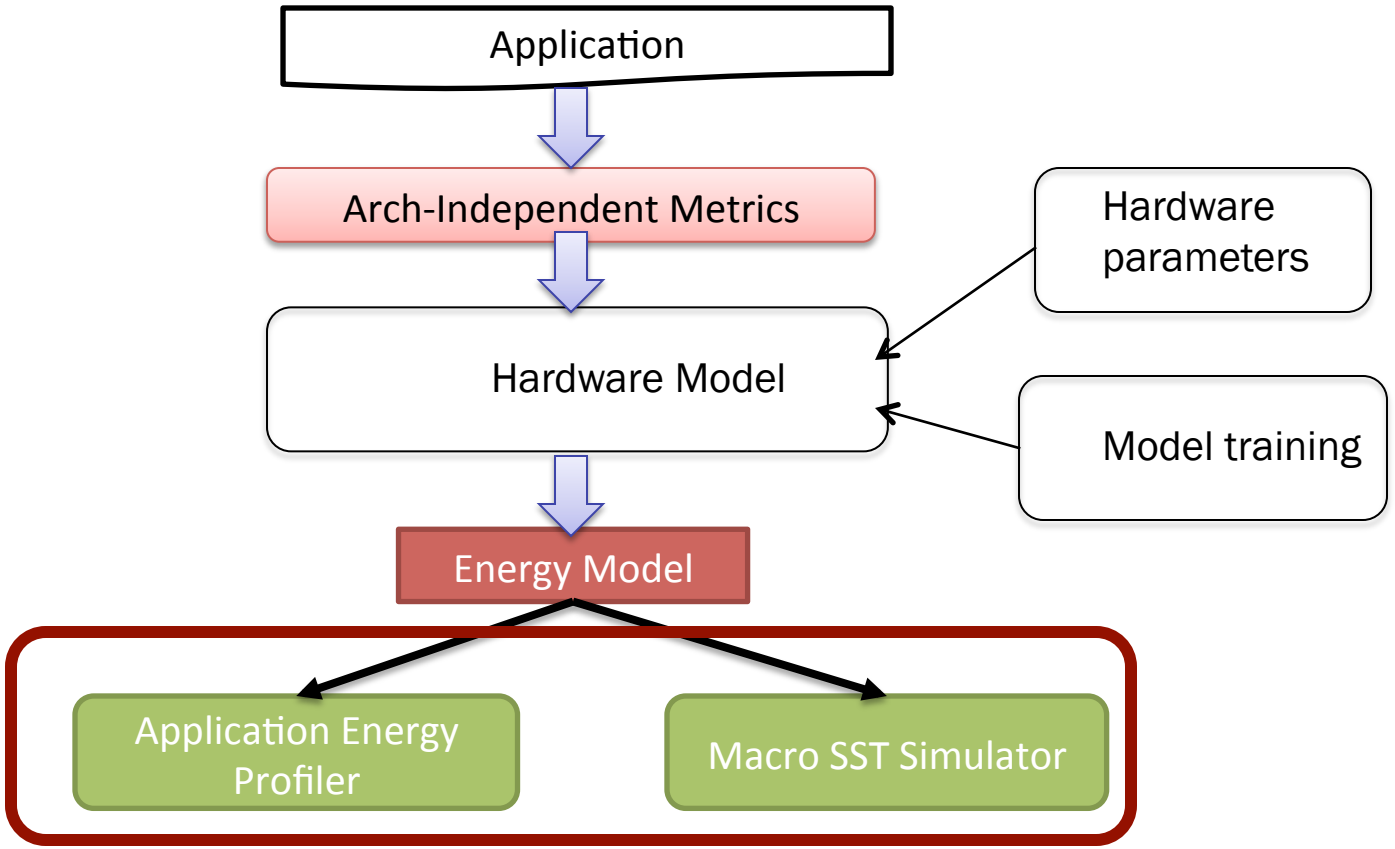
Andrew Kerr, Eric Anger, Gilber Hendry, and Sudhakar Yalamanchili. “Eiger: A framework for the automated synthesis of statistical performance models”, In High Performance Computing, 2012.

Eiger Framework



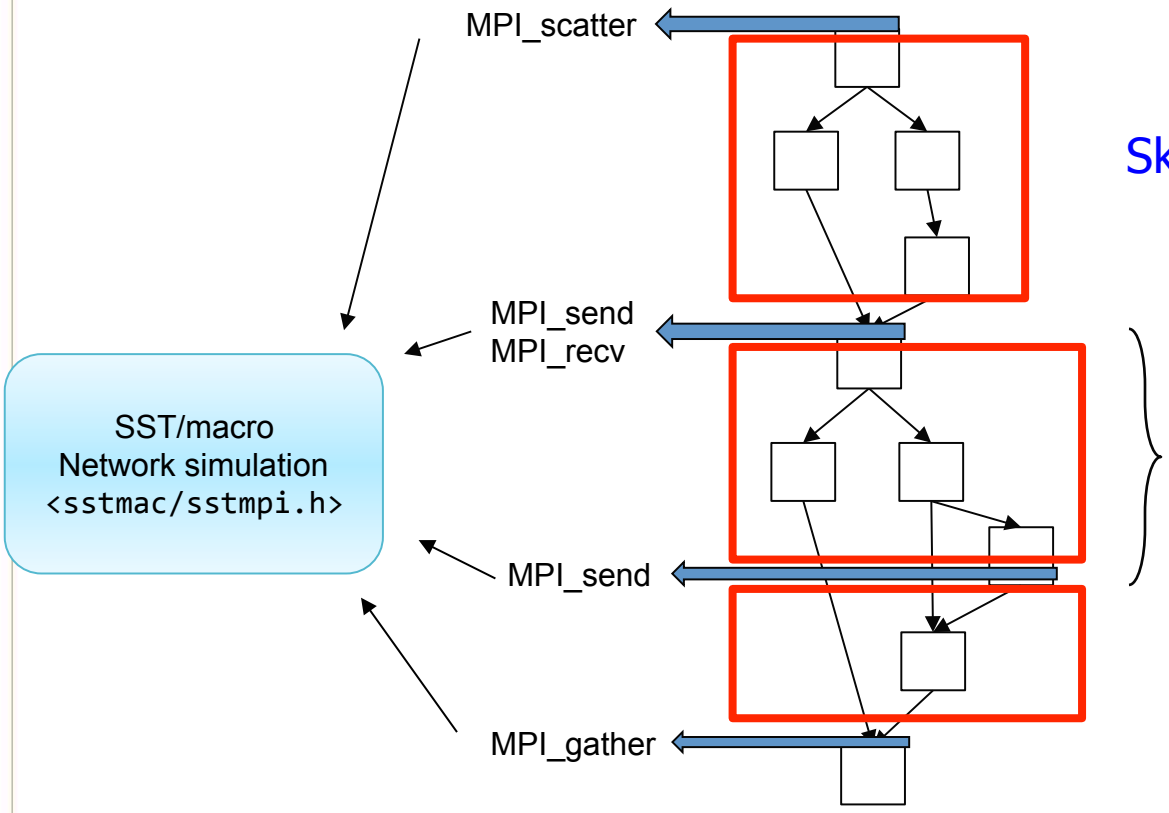
- Manipulations of data → analyze relationships
  - Aid in analysis and verification of model behavior
- Ease model exploration
- Extensible to new modeling techniques

Framework


Application
Arch-Independent Metrics
Hardware parameters
Hardware Model
Model training
Energy Model
Application Energy Profiler
Macro SST Simulator

Application Skeletons

Simplified code which (approximately) reproduces some behavior of interest for a full application



Example - Communication Skeletons: Capture only control flow and communication

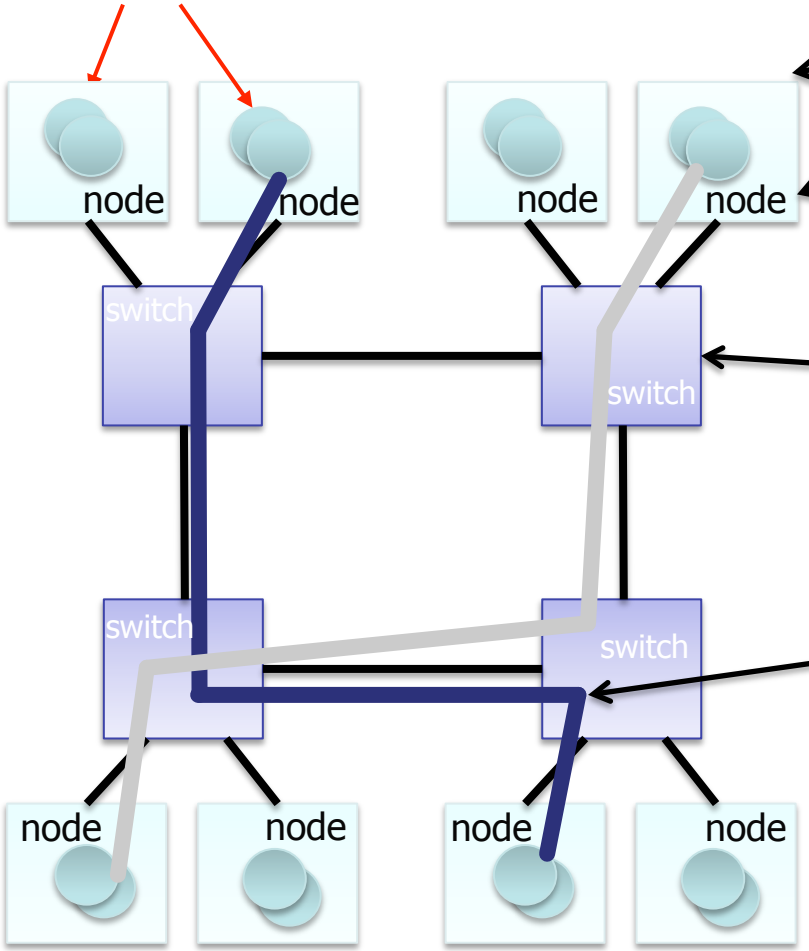
Replace with a model of execution time

will expand for energy

Eiger Framework

Macro-scale Simulation Structure*

Instances of application skeletons



Region-specific energy models

Nodes Model:
- Multithreading
- Accelerators
- NIC effects/contention

Network switches model:
- Packet arbitration
- Adaptive Routing
- Queuing/buffering

Messages modeled as:
- Flows
- Packets
- Packet trains

System-level estimates

*SST, SNL



Application-level Energy Profiler

- Source code level energy profiler
  - Function level, instruction level
- Future work will construct more high-level analysis
  - e.g.) analysis of TLP, synchronization overhead, data movement

Sandia National Laboratories
Los Alamos NATIONAL LABORATORY EST. 1943
Georgia Tech
comparch



Future Work

- More coding, modeling, ….
  - all model components need to be improved/ integrated
  - Validations
  - Large scale simulations
- More high-level application level energy models

Sandia National Laboratories
Los Alamos NATIONAL LABORATORY EST. 1943
Georgia Tech
comparch



Summary Q&A-I

- Major contributions of the work
  - Integrated performance/power model starting from application level
- What are the gaps in the research area
  - Providing feedback to high-level applications from low-level application modeling
- What major opportunities
  - Frame can provide tools for application developers to optimize their applications and hardware developers to simulate large scale applications



Summary-Q&A II

- What is the one thing that would make it easier/possible to leverage/use the results of other projects to further your own research
  - Faster cache modeling
- What would you like to most see solved/addressed other than what they are working on?
  - Low-overhead DRAM (memory technology) specific models
  - Hardware performance counters to measure detailed DRAM access behaviors



THANK YOU!

Sandia National Laboratories

Los Alamos NATIONAL LABORATORY EST. 1943

Georgia Tech

comparch