SIMULATION OF EXASCALE NODES THROUGH RUNTIME HARDWARE MONITORING

JOSEPH L. GREATHOUSE, ALEXANDER LYASHEVSKY, MITESH MESWANI, NUWAN JAYASENA, MICHAEL IGNATOWSKI

9/18/2013
CHALLENGES OF EXASCALE NODE RESEARCH
MANY DESIGN DECISIONS
CHALLENGES OF EXASCALE NODE RESEARCH
MANY DESIGN DECISIONS

Heterogeneous Cores

Composition? Size? Speed?
CHALLENGES OF EXASCALE NODE RESEARCH
MANY DESIGN DECISIONS

Heterogeneous Cores
Composition? Size? Speed?

Stacked Memories
CHALLENGES OF EXASCALE NODE RESEARCH
MANY DESIGN DECISIONS

Heterogeneous Cores
Composition? Size? Speed?

Stacked Memories

Thermal Constraints
Power Sharing? Heat dissipation? Sprinting?
CHALLENGES OF EXASCALE NODE RESEARCH

MANY DESIGN DECISIONS

Heterogeneous Cores

- Composition? Size? Speed?

Stacked Memories


Thermal Constraints

- Power Sharing? Heat dissipation? Sprinting?

Software Co-Design

- New algorithms? Data placement? Programming models?
CHALLENGES OF EXASCALE NODE RESEARCH
MANY DESIGN DECISIONS

Heterogeneous Cores

Stacked Memories

Exascale: Huge Design Space to Explore

Power Sharing? Heat dissipation? Sprinting?

New algorithms? Data placement? Programming models?
Power and Thermals on a real heterogeneous processor:

- GPU Pow
- CPU CU0 Pow
- CPU CU1 Pow
- PeakDieTemp

Time (seconds) vs Relative Power and Peak Die Temperature
Power and Thermals on a real heterogeneous processor:

- ~2.5 trillion CPU instructions, ~60 trillion GPU operations
CHALLENGES OF EXASCALE NODE RESEARCH
INTERESTING QUESTION REQUIRE LONG RUNTIMES

Power and Thermals on a real heterogeneous processor:

- ~2.5 trillion CPU instructions, ~60 trillion GPU operations

Exascale Proxy Applications are Large
- Large initialization phases, many long iterations
- Not microbenchmarks
- **Already** reduced inputs and computation from real HPC applications
CHALLENGES OF EXASCALE NODE RESEARCH

INTERESTING QUESTION REQUIRE LONG RUNTIMES

Power and Thermals on a real heterogeneous processor:

- ~2.5 trillion CPU instructions, ~60 trillion GPU operations
- Exascale Proxy Applications are Large
  - Large initialization phases, many long iterations
  - Not microbenchmarks
  - **Already** reduced inputs and computation from real HPC applications
EXISTING SIMULATORS

- Microarchitecture Sims: e.g. gem5, Multi2Sim, MARSSx86, SESC, GPGPU-Sim
  - Excellent for low-level details. We need these!
  - Too slow for design space explorations: \(~60\) trillion operations = 1 year of sim time

- Functional Simulators: e.g. SimNow, Simics, QEMU, etc.
  - Faster than microarchitectural simulators, good for things like access patterns
  - No relation to hardware performance

- High-Level Simulators: e.g. Sniper, Graphite, CPR
  - Break operations down into timing models, e.g. core interactions, pipeline stalls, etc.
  - Faster, easier to parallelize.
  - Runtimes and complexity still constrained by desire to achieve accuracy.
TRADE OFF INDIVIDUAL TEST ACCURACY

Doctors do not start with:

1. [CT Scan Image](http://www.flickr.com/photos/calliope/361953316/ - CC BY 2.0)
2. [Microscope Image](http://www.flickr.com/photos/elsie/7080667207/ - CC BY 2.0)
3. [X-ray Image](http://www.flickr.com/photos/aidan_jones/1438403889/ - CC BY-SA 2.0)
Fast Simulation Using Hardware Monitoring
HIGH-LEVEL SIMULATION METHODOLOGY
MULTI-STAGE PERFORMANCE ESTIMATION PROCESS

Phase 1

- User Application
  - Software Tasks
  - Simulator Software Stack
  - Test System OS
  - Test System APU, CPU, GPU

Phase 2

- Performance, Power, and Thermal Models
- Trace Post-processor
- Performance & Power Estimates

Traces of HW and SW events related to performance and power
BENEFITS OF MULTI-STEP PROCESS
MANY DESIGN SPACE CHANGES ONLY NEED SECOND STEP

Phase 1
User Application
Software Tasks
Simulator Software Stack
Test System OS
Test System APU, CPU, GPU

Multiple Machine Descriptions
HW & SW Traces
Performance, Power, and Thermal Models
Trace Post-processor
Phase 2
Performance & Power Estimates
PERFORMANCE SCALING A SINGLE-THREADED APP

Gather statistics and performance counters about:
- Instructions Committed, stall cycles
- Memory operations, cache misses, etc.
- Power usage

Current Processor

Start Time

End Time

Simulated Processor

Analytical Performance Scaling Model

New Runtime
ANALYTIC PERFORMANCE SCALING

CPU Performance Scaling:
- Find stall events using HW perf. counters. Scale based on new machine parameters.
- Large amount of work in the literature on DVFS performance scaling, interval models.
- Some events can be scaled by fiat:
  "If IPC when not stalled on memory doubled, what would happen to performance?"

GPU Performance Scaling:
- Watch HW perf. counters that indicate work to do, memory usage, and GPU efficiency
- Scale values based on estimations of parameters to test
OTHER ANALYTIC MODELS

- Cache/Memory Access Model
  - Observe memory access traces using binary instrumentation or hardware
  - Send traces through high-level cache simulation system
  - Find knees in the curve, feed this back to CPU/GPU performance scaling model

- Power and thermal models
  - Power can be correlated to hardware events or directly measured
  - Scaled to future technology points
  - Any number of thermal models will work at this point

- Thermal and power models can feed into control algorithms that change system performance
  - This is another HW/SW co-design point. Fast operation is essential.
SCALING PARALLEL SEGMENTS REQUIRES MORE INFO

Example when everything except CPU1 gets faster:
SCALING PARALLEL SEGMENTS REQUIRES MORE INFO

Example when everything except CPU1 gets faster:

- **CPU0**: Performance Difference
- **CPU1**: Performance Difference
- **GPU**: Performance Difference
Scaling parallel segments requires more info:

Example when everything except CPU1 gets faster:
SCALING PARALLEL SEGMENTS REQUIRES MORE INFO

Example when everything except CPU1 gets faster:

Performance Difference
MUST RECONSTRUCT CRITICAL PATHS

- Gather program-level relationship between individually scaled segments
- Use these happens-before relationships to build a legal execution order

CPU0

1

2

3

4

5

6

7

CPU1

GPU

①

②

③

④

⑤

⑥

⑦

② happens after ①
③ happens after ①
④ happens after ③
⑤ happens after ③
⑥ happens after ④
⑦ happens after ②
⑥ happens after ⑤

①, ②, ⑦ run on CPU
③, ④, ⑥ run on CPU
⑤ runs on GPU

- Gather ordering from library calls like pthread_create() and clWaitForEvents()
- Can also split segments based on program phases
CONCLUSION

Exascale node design space is huge

Trade off some accuracy for faster simulation

Use analytic models based on information from existing hardware
Research Related Questions
MODSIM RELATED QUESTIONS

△ Major Contributions:
- A fast, high-level performance, power, and thermal analysis infrastructure
- Enables large design space exploration and HW/SW co-design with good feedback

△ Limitations:
- Trace-based simulation has known limitations w/r/t multiple paths of execution, wrong-path operations, etc.
- It can be difficult and slow to model something if your hardware can’t measure values that are correlated to it.

△ Bigger Picture:
- Node-level performance model for datacenter/cluster performance modeling
- First pass model for APU power sharing algorithms.
- Exascale application co-design
- Complementary work to broad projects like SST
What is the one thing that would make it easier to leverage the results of other projects to further your own research

- Theoretical bounds and analytic reasoning behind performance numbers. Even “good enough” guesses may help, vs. only giving the output of a simulator

What are important thing to address in future work?

- Better analytic scaling models. There are a lot in the literature, but many rely on simulation to propose new hardware that would gather correct statistics.

- It would be great if open source performance monitoring software were better funded, had more people, etc.
The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION

© 2013 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names are for informational purposes only and may be trademarks of their respective owners.
Backup
Is 3D-Stacked Memory Beneficial for Application X?

- Baseline Performance
- Bandwidth Difference
- Latency Difference
- Thermal Model
- Core Changes due to Heat
Is 3D-Stacked Memory Beneficial for Application X?

Baseline Performance
Bandwidth Difference
Latency Difference
Thermal Model
Core Changes due to Heat

Simulation Run(s)

Performance results based on design point(s)
Is 3D-Stacked Memory Beneficial for Application X?

Baseline Performance
Bandwidth Difference
Latency Difference
Thermal Model
Core Changes due to Heat

Simulation Run(s)

Performance results based on design point(s)

Redesign
AN EXASCALE NODE EXAMPLE QUESTION

Is 3D-Stacked Memory Beneficial for Application X?

Baseline Performance
Bandwidth Difference
Latency Difference
Thermal Model
Core Changes due to Heat

Simulation Run(s)

Performance results based on design point(s)

Redesign

Modify software to better utilize new hardware
Is 3D-Stacked Memory Beneficial for Application X?

Baseline Performance
Bandwidth Difference
Latency Difference
Thermal Model
Core Changes due to Heat

Simulation Run(s)

Performance results based on design point(s)

Redesign

Modify software to better utilize new hardware