Practicalities of Design Space Exploration with gem5 and McPAT

by Erik Tomusk

Email: E.Tomusk@sms.ed.ac.uk
Homepage: http://homepages.inf.ed.ac.uk/s1151659/

Advisors: Michael O’Boyle
Christophe Dubach
John Goodacre (ARM)

part of “Low Power Ecosystem - gem5 Practical User Experience”
at HiPEAC CSW, Ghent, 18 October 2012
Current Research: Weak Heterogeneity

- Simplest form of heterogeneity
- Can small microarchitectural variations be beneficial?

Heterogeneous computing can be more power efficient. gem5 and McPAT allow fine-grained parameter tweaks. Design space exploration requires 1000s of simulations.
Current Research: Weak Heterogeneity

- Simplest form of heterogeneity
- Can small microarchitectural variations be beneficial?

Heterogeneous computing can be more power efficient. gem5 and McPAT allow fine-grained parameter tweaks. Design space exploration requires 1000s of simulations.
Current Research: Weak Heterogeneity

• Simplest form of heterogeneity
• Can small microarchitectural variations be beneficial?

Heterogeneous computing can be more power efficient. gem5 and McPAT allow fine-grained parameter tweaks. Design space exploration requires 1000s of simulations.
Heterogeneous computing can be more power efficient. gem5 and McPAT allow fine-grained parameter tweaks. Design space exploration requires 1000s of simulations.
Current Research: Weak Heterogeneity

- Simplest form of heterogeneity
- Can small microarchitectural variations be beneficial?
- Less potential for power & performance gains
- Less potential for catastrophic problems
- Perfect fit for gem5 & McPAT

Heterogeneous computing can be more power efficient. gem5 and McPAT allow fine-grained parameter tweaks. Design space exploration requires 1000s of simulations.
gem5 Workflow—Running on a Server Farm

Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
gem5 Workflow—Running on a Server Farm

**Python Overrides**

```python
system.cpu[0].choiceCtrBits=1
system.cpu[0].globalCtrBits=1
...```

**Simulation List**
- Run1
- Run2
- ...

**CMD Options**

**gem5 Simulation**

**Configurations**
- Benchmarks
- HW parameters
- Simulation options

Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
gem5 Workflow—Running on a Server Farm

Python Overrides
system.cpu[0].choiceCtrBits=1
system.cpu[0].globalCtrBits=1
...

Simulation List
• Run1
• Run2
• ...

CMD Options

gem5 Simulation

config.ini
stats.txt

Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
gem5 Workflow—Running on a Server Farm

Python Overrides
system.cpu[0].choiceCtrBits=1
system.cpu[0].globalCtrBits=1
...

Simulation List
• Run1
• Run2
• ...

CMD Options

gem5 Simulation

config.ini
stats.txt

Analysis

Override some parameters before Python hierarchy moves to gem5 executable.
Auto-generate everything from the configuration table.
McPAT Workflow

**McPAT XML**

```xml
...<param name="clock_rate"
    value="{int(1e-6/config.system.cpu.clock * stats.sim_freq)}"/>
<stat name="total_cycles"
    value="{stats.system.cpu.numCycles}"/>
...```

*Original gem5->McPAT tool: Andrew Rice ([https://www.cl.cam.ac.uk/~acr31/sicsa/](https://www.cl.cam.ac.uk/~acr31/sicsa/))

McPAT isn’t run on farm since it runs in ~15–20 sec.
McPAT Workflow

stats.txt

config.ini

McPAT XML

...<param name="clock_rate"
value="{int(1e-6/config.system.cpu.clock
* stats.sim_freq)}"/>
<stat name="total_cycles"
value="{stats.system.cpu.numCycles}"/>
...

*Original gem5->McPAT tool: Andrew Rice (https://www.cl.cam.ac.uk/~acr31/sicsa/)

McPAT isn’t run on farm since it runs in ~15–20 sec.
McPAT Workflow

**McPAT XML**

```xml
... 
<param name="clock_rate" 
    value="\{\mathtt{int(1e-6/config.system.cpu.clock} 
    * \mathtt{stats.sim_freq)}\}"/>
<stat name="total_cycles" 
    value="\{\mathtt{stats.system.cpu.numCycles}\}"/>
... 
```

*Original gem5->McPAT tool: Andrew Rice ([https://www.cl.cam.ac.uk/~acr31/sicsa/](https://www.cl.cam.ac.uk/~acr31/sicsa/))

McPAT isn’t run on farm since it runs in ~15–20 sec.
McPAT Workflow

McPAT XML

```
<param name="clock_rate"
    value="{int(1e-6/config.system.cpu.clock
                * stats.sim_freq)}"/>
<stat name="total_cycles"
    value="{stats.system.cpu.numCycles}"/>
```

stats.txt

config.ini

*Original gem5->McPAT tool: Andrew Rice (https://www.cl.cam.ac.uk/~acr31/sicsa/)

McPAT isn’t run on farm since it runs in ~15–20 sec.
McPAT Workflow

```
<param name="clock_rate" value="{int(1e-6/config.system.cpu.clock * stats.sim_freq)}"/>
<stat name="total_cycles" value="{stats.system.cpu.numCycles}"/>
```

*Original gem5->McPAT tool: Andrew Rice (https://www.cl.cam.ac.uk/~acr31/sicsa/)

McPAT isn’t run on farm since it runs in ~15–20 sec.
gem5 Experience

Server farm sends signal short time before time is up.
gem5 Experience

Heisenbugs

- Sometimes simulations fail immediately on start (1-2%)
- Possibly an artifact of the server farm
- Immediately rerunning on same machine works

Server farm sends signal short time before time is up.
gem5 Experience

Heisenbugs

• Sometimes simulations fail immediately on start (1-2%)
• Possibly an artifact of the server farm
• Immediately rerunning on same machine works

Syndrome 1

```
stdout:
Exiting @ tick 0 because user interrupt received
```

Server farm sends signal short time before time is up.
gem5 Experience

**Heisenbugs**

- Sometimes simulations fail immediately on start (1-2%)
- Possibly an artifact of the server farm
- Immediately rerunning on same machine works

**Syndrome 1**

stdout:
Exiting @ tick 0 because user interrupt received

**Syndrome 2**

stderr:
panic: ListenSocket(listen): listen() failed!

Server farm sends signal short time before time is up.
gem5 Experience

Heisenbugs

- Sometimes simulations fail immediately on start (1-2%)
- Possibly an artifact of the server farm
- Immediately rerunning on same machine works

Fixed-Time Simulations

- Maximize server farm usage
- Simulations are killed and data is lost when server slot runs out

Syndrome 1

stdout:
Exiting @ tick 0 because user interrupt received

Syndrome 2

stderr:
panic: ListenSocket(listen): listen() failed!

Server farm sends signal short time before time is up.
gem5 Experience

Heisenbugs

• Sometimes simulations fail immediately on start (1-2%)
• Possibly an artifact of the server farm
• Immediately rerunning on same machine works

Fixed-Time Simulations

• Maximize server farm usage
• Simulations are killed and data is lost when server slot runs out
• gem5 dumps stats.txt and exits on SIGINT

Syndrome 1
stdout:
Exiting @ tick 0 because user interrupt received

Syndrome 2
stderr:
panic: ListenSocket(listen): listen() failed!

Server farm sends signal short time before time is up.
gem5 Experience

Heisenbugs

• Sometimes simulations fail immediately on start (1-2%)
• Possibly an artifact of the server farm
• Immediately rerunning on same machine works

Fixed-Time Simulations

• Maximize server farm usage
• Simulations are killed and data is lost when server slot runs out
• gem5 dumps stats.txt and exits on SIGINT

Syndrome 1
stdout:
Exiting @ tick 0 because user interrupt received

Syndrome 2
stderr:
panic: ListenSocket(listen): listen() failed!

Server farm sends signal short time before time is up.
gem5 Experience

Heisenbugs

• Sometimes simulations fail immediately on start (1-2%)
• Possibly an artifact of the server farm
• Immediately rerunning on same machine works

Fixed-Time Simulations

• Maximize server farm usage
• Simulations are killed and data is lost when server slot runs out
• gem5 dumps stats.txt and exits on SIGINT

Syndrome 1
stdout:
Exiting @ tick 0 because user interrupt received

Syndrome 2
stderr:
panic: ListenSocket(listen): listen() failed!

Varying simulated time leads to noisy results

Server farm sends signal short time before time is up.
gem5 performs as expected.
Noise would be removed if simulations run for same number of instructions.
Execution time decreases less than linearly with frequency increase.
gem5 performs as expected. Noise would be removed if simulations run for same number of instructions. Execution time decreases less than linearly with frequency increase.
gem5 performs as expected.
Noise would be removed if simulations run for same number of instructions.
Execution time decreases less than linearly with frequency increase.
gem5 performs as expected.
Noise would be removed if simulations run for same number of instructions.
Execution time decreases less than linearly with frequency increase.
McPAT Experience: Power Calculation

\[ P = a \times f \times C \times V^2 \]

McPAT does not directly use a standard power equation.
McPAT Experience: Power Calculation

McPAT does not directly use a standard power equation.
McPAT Experience: Power Calculation

McPAT does not directly use a standard power equation.
McPAT Experience: Power Calculation

• Core is divided into CACTI blocks
  – Non-CACTI blocks are negligible
McPAT Experience: Power Calculation

- Core is divided into CACTI blocks
  - Non-CACTI blocks are negligible
- Each block is solved to meet timing while minimizing area
McPAT Experience: Power Calculation

- Core is divided into CACTI blocks
  - Non-CACTI blocks are negligible
- Each block is solved to meet timing while minimizing area
- Solution:
  - Read/write energy costs (Joules)
McPAT Experience: Power Calculation

- Core is divided into CACTI blocks
  - Non-CACTI blocks are negligible
- Each block is solved to meet timing while minimizing area
- Solution:
  - Read/write energy costs (Joules)
  - Area (mm$^2$)
McPAT Experience: Power Calculation

- Core is divided into CACTI blocks
  - Non-CACTI blocks are negligible
- Each block is solved to meet timing while minimizing area
- Solution:
  - Read/write energy costs (Joules)
  - Area (mm$^2$)
  - Leakage power (Watts)
McPAT Experience: Power Calculation

- Core is divided into CACTI blocks
  - Non-CACTI blocks are negligible
- Each block is solved to meet timing while minimizing area
- Solution:
  - Read/write energy costs (Joules)
  - Area (mm$^2$)
  - Leakage power (Watts)
- NoC power calculated separately
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + \left( E_R \times \text{reads} + E_W \times \text{writes} \right) / \text{time} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + \left( E_R \ast \text{reads} + E_W \ast \text{writes} \right) / \text{time} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + (E_R \times \text{reads} + E_W \times \text{writes}) / \text{time} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + (E_R \times \text{reads} + E_W \times \text{writes}) / \text{time} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + (E_R \times \text{reads} + E_W \times \text{writes}) / \text{time} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + (E_R \times \text{reads} + E_W \times \text{writes}) / \text{time} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + \frac{(E_R \times \text{reads} + E_W \times \text{writes})}{\text{time}} \]

Energy

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
McPAT Experience: Power Calculation

\[ P = \sum \text{leakage} + \frac{(E_R \times \text{reads} + E_W \times \text{writes})}{\text{time}} \]

Dynamic power is inversely proportional to execution time. Mathematically correct, but not completely realistic.
Power does not increase superlinearly with frequency as in the real world.
As frequency increases, timing requirements become tighter. When timing can’t be met, CACTI’s solutions suddenly become more energy costly.

Since time does not decrease linearly with frequency and McPAT bases dynamic power only on energy and frequency, power does not scale as expected.
As frequency increases, timing requirements become tighter. When timing can’t be met, CACTI’s solutions suddenly become more energy costly. Since time does not decrease linearly with frequency and McPAT bases dynamic power only on energy and frequency, power does not scale as expected.
As frequency increases, timing requirements become tighter. When timing can’t be met, CACTI’s solutions suddenly become more energy costly. Since time does not decrease linearly with frequency and McPAT bases dynamic power only on energy and frequency, power does not scale as expected.

- Large increases from CACTI segmentation
- Small power increase because small time increase
McPAT Experience: Frequency

As frequency increases, timing requirements become tighter. When timing can’t be met, CACTI’s solutions suddenly become more energy costly. Since time does not decrease linearly with frequency and McPAT bases dynamic power only on energy and frequency, power does not scale as expected.

- Large increases from CACTI segmentation
- Small power increase because small time increase
- McPAT ignores dynamic power for inactive blocks—no clock tree
As frequency increases, timing requirements become tighter. When timing can’t be met, CACTI’s solutions suddenly become more energy costly. Since time does not decrease linearly with frequency and McPAT bases dynamic power only on energy and frequency, power does not scale as expected.
McPAT Experience: Power Breakdown
McPAT Experience: Power Breakdown

**Power Breakdown, SPEC, LOP Process**

Normalized Power

- Runtime Dynamic
- Subthreshold Leakage
- Gate Leakage

**Power Breakdown, SPEC, HP Process**

Normalized Power

- Runtime Dynamic
- Subthreshold Leakage
- Gate Leakage

Benchmark
McPAT Experience: Power Breakdown

- Static/dynamic power breakdown seems reasonable
  - Leakage slightly high for high performance process?
  - Due to missing quiescent dynamic power component?
McPAT Experience: Power Breakdown

- Static/dynamic power breakdown seems reasonable
  - Leakage slightly high for high performance process?
  - Due to missing quiescent dynamic power component?
- What about peak dynamic power?
  - Up to ~30x runtime dynamic power—unreasonable
  - Probably based on maximum usage of every CACTI block all the time
Future Work
Future Work

• Implement a clock tree model/heuristic
• Find better ways of configuring McPAT
• Open to collaborating on McPAT, gem5, etc.
Conclusions

• gem5 is a great tool for design space exploration
Conclusions

• gem5 is a great tool for design space exploration

• McPAT can complement gem5, but
  — McPAT is not as intuitive to configure as gem5
  — McPAT’s reliance on CACTI creates some quirks
Conclusions

• gem5 is a great tool for design space exploration
• McPAT can complement gem5, but
  — McPAT is not as intuitive to configure as gem5
  — McPAT’s reliance on CACTI creates some quirks
  — *Difficult to ensure gem5 and McPAT model the same thing*
Conclusions

• gem5 is a great tool for design space exploration
• McPAT can complement gem5, but
  – McPAT is not as intuitive to configure as gem5
  – McPAT’s reliance on CACTI creates some quirks
  – *Difficult to ensure gem5 and McPAT model the same thing*
• gem5+McPAT are the best free tools for design space exploration
Conclusions

• gem5 is a great tool for design space exploration
• McPAT can complement gem5, but
  — McPAT is not as intuitive to configure as gem5
  — McPAT’s reliance on CACTI creates some quirks
    — *Difficult to ensure gem5 and McPAT model the same thing*
• gem5+McPAT are the best free tools for design space exploration

Questions?