Understanding the Future of Energy Efficiency in Multi-Module GPUs

Akhil Arunkumar*, Evgeny Bolotin#, David Nellans#, Carole-Jean Wu*

*Arizona State University, #NVIDIA
Multi-Module GPUs

Monolithic GPU

On-package Integration

On-board Integration

Hybrid Integration

**On-package Integration**
Utilize organic package / interposer
- Arunkumar et al., ISCA ‘17
- Vijayaraghavan et al., HPCA ‘17

**On-board Integration**
Utilize PC board
- Milic et al., ISCA ‘17
- NVIDIA DGX, HGX

**Hybrid Integration**
Utilize package and PC board
- Dally et al., VLSI ’18

Prior works have focused only on the performance aspect.
Energy Cost of Multi-Module Scaling

- Energy cost per task could double!
  - 32 GPMs integrated on-board consumes 2x the energy of 1 GPM 😞

- *What are the energy efficiency limitations?*

- *Where are the bottlenecks?*
Outline

• Introduction and background

• GPU energy estimation framework – GPUJoule

• Energy efficiency scaling metric – EDPSE

• Energy efficiency trends in future multi-module GPUs

• Conclusion
GPU Energy Estimation – Prior Work

• **Bottom-up GPU energy estimation**[1][2][3]:
  - Estimate energy cost of each microarchitectural component
  - Hard to keep current as GPUs evolve

• **Top-down instruction-based energy estimation**[4][5][6]:
  - Estimate energy cost of instruction operations executed
  - Flexible and agile as microarchitecture evolves

Top-down energy model is well suited for GPUs

---

Our Contribution: The GPUJoule Framework

• Key Idea:
  • Estimate the energy-per-instruction (EPI) for each compute instruction type
  • Estimate the energy-per-transaction (EPT) for each memory transaction type
  • GPU-Energy (per-application):
    \[ = \sum (N_i \times EPI_i) + \sum (Txn_j \times EPT_j) + \text{idle_energy} \]

Energy to execute compute instructions

Energy to execute data movement instructions
**GPUJoule Energy Modeling Methodology**

**Microbenchmarks**
- Stress
- Compute and Memory Insts

**GPUJoule Energy Model**
\[
\text{Est. Energy} = \sum(N_i \times EPI_i) + \sum(Txn_j \times EPT_j) + \text{idle_energy}
\]

**uBench Validation**
\[
\text{Error} = (\text{Si}) \text{ Measured Energy} - \text{Est. Energy}
\]

**Compute Instruction Microbenchmarks**
For \( i = 0 \) to \( i < \text{num_iterations} \) do:

```c
__asm_volatile ("fma.rn.f32 %r3, %r1, %r3, %r2;"
...)
```

**Memory Instruction Microbenchmarks**
For \( i = 0 \) to \( i < \text{num_iterations} \) do:

```c
ptr = (void **)(&array[index])
ptr = (void**)(*ptr)
```
GPUJoule Validation

- **GPU platform**
  - Nvidia Tesla K40 GPU
    - 15 SMs, 16 – 48 KB L1 cache,
    - 1.5 MB L2 cache, 12 GB, 280 GB/s GDDR5 Memory
    - On-board power sensors for power measurement

- **Workloads**
  - Validation microbenchmarks ➔ compute instruction + data movement operations
  - Real GPU applications from Rodinia, CORAL & Stream suites
Tesla K40 Energy Characteristics

<table>
<thead>
<tr>
<th>Inst or Op</th>
<th>EPI (nJ)</th>
<th>EPT (pJ/bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DADD, FFMA</td>
<td>0.15, 0.05</td>
<td>-</td>
</tr>
<tr>
<td>IADD, IMAD</td>
<td>0.07, 0.15</td>
<td>-</td>
</tr>
<tr>
<td>LOG2, SINE</td>
<td>0.03, 0.10</td>
<td>-</td>
</tr>
<tr>
<td>Shd Mem -&gt; Reg, L1 -&gt; Reg</td>
<td>-</td>
<td>5.32, 5.85</td>
</tr>
<tr>
<td>L2 -&gt; L1</td>
<td>-</td>
<td>15.48</td>
</tr>
<tr>
<td>DRAM -&gt; L2</td>
<td>-</td>
<td>30.55</td>
</tr>
</tbody>
</table>

- EPI influenced by bit width, and functional unit
- EPT influenced by the level of memory hierarchy
  - DRAM -> Register costs 9x more than L1 -> Register
  - DRAM -> Register costs 80x more than floating point compute
GPUJoule Accuracy

98% Accuracy

90% Accuracy
Outline

• Introduction and background
• GPU energy estimation framework – GPUJoule
• Energy efficiency scaling metric – EDPSE
• Energy efficiency trends in future multi-module GPUs
• Conclusion
Quantifying Energy Efficiency: EDP Scaling Efficiency

- EDP and ED^2 well suited for comparing systems with similar resources
- For strong scaled systems: Energy-Delay-Product Scaling Efficiency (EDPSE)
  \[ EDPSE = \frac{EDP_1}{N} \times \frac{1}{EDP_N} \]
- Evaluates performance, energy costs, and resource scaling together
- Systems can be expected to achieve an EDPSE threshold in the future
  - 50% EDPSE ➔ “Energy efficiency scales to 50% of the ideal with strong scaling”
Outline

• Introduction and background
• GPU energy estimation framework – GPUJoule
• Energy efficiency scaling metric – EDPSE
• Energy efficiency trends in future multi-module GPUs
• Conclusion
Methodology

• Performance Simulations:
  • Model GPUs with 1 – 32 GPU modules
  • Distributed CTA scheduling, first touch page placement, ring interconnect

<table>
<thead>
<tr>
<th>BW Config Name</th>
<th>I/O BW</th>
<th>DRAM BW</th>
<th>I/O to DRAM BW Ratio</th>
<th>Integration Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x-BW</td>
<td>128 GB/s</td>
<td>256 GB/s</td>
<td>1:2</td>
<td>On-Board</td>
</tr>
<tr>
<td>2x-BW</td>
<td>256 GB/s</td>
<td>256 GB/s</td>
<td>1:1</td>
<td>On-Package</td>
</tr>
<tr>
<td>4x-BW</td>
<td>512 GB/s</td>
<td>256 GB/s</td>
<td>2:1</td>
<td>On-Package</td>
</tr>
</tbody>
</table>

• Energy Modeling:
  • EPI and EPT values from GPUJoule
  • Augmented with HBM Memory & Inter-GPM data movement energy costs

<table>
<thead>
<tr>
<th>Energy Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBM DRAM -&gt; L2 Cache[1]</td>
</tr>
<tr>
<td>On-Package Inter-GPM[2]</td>
</tr>
<tr>
<td>On-Board Inter-GPM[3]</td>
</tr>
</tbody>
</table>

EDP Scaling Efficiency of Future GPUs

- EDPSE reduces drastically with increase in GPMs
- Multi-Module GPUs face energy efficiency limitations at scale
Diminishing Trend in Energy Efficiency Scaling

- Speedup reduces as number of modules increase
- Energy cost increases as number of modules increase

NUMA-effects lead to performance loss and energy increase
On-package integration and constant energy amortization

- Multi-module GPUs suffer from high constant energy overheads
  - VRMs, power delivery network, system I/O etc.

- On-package integration allows amortization of these overheads

Higher link BW and tighter integration yields better energy efficiency scaling
Speedup & Energy Consumption

- Speedup is dependent on bandwidth
- Energy consumption drops with speedup

- Only increasing GPMs might not help
  - 16-GPM with 2xBW has same performance as 32-GPM with 1xBW
  - Consumes only half the energy!

- Path to an efficient 32-GPM GPU
  - Increase bandwidth to 4x-BW.
  - Utilize on-package integration
  - Reduce energy consumption by 45%
Conclusions

- Developed GPUJoule Instruction level GPU energy estimation framework
  - Achieves 90% accuracy compared to real silicon energy measurements
  - Open sourced at github.com/akhilarunkumar/GPUJoule_release
- Identify key energy efficiency trends in future GPUs
  - Energy efficiency scaling reduces as number of modules increase
  - NUMA effects lead to suboptimal performance and energy consumption
  - Inter-module bandwidth and tighter integration of components (on package integration) lead to higher energy efficiency
Understanding the Future of Energy Efficiency in Multi-Module GPUs

Thank you

Akhil Arunkumar*, Evgeny Bolotin#, David Nellans#, Carole-Jean Wu*
*Arizona State University, #NVIDIA
Impact of On-Board Switch

![Bar chart showing EDPSE (%) for different on-board integration levels (2-GPM, 4-GPM, 8-GPM, 16-GPM, 32-GPM) with three types of switch connection (Ring (1x-BW), Switch (1x-BW), Switch (2x-BW)).]