LogCA: A High-Level Performance Model for Hardware Accelerators

Muhammad Shoaib Bin Altaf*
David A. Wood
University of Wisconsin-Madison

*Now at AMD Research, Austin TX
Executive Summary

• Accelerators do not always perform as expected
• Crucial for programmers and architects to understand the factors which affect performance
• Simple analytical models beneficial early in the design stage
• Our proposal: LogCA
  – High-level performance model
  – Help identify design bottlenecks and possible optimizations
• Validation across variety of on-chip and off-chip accelerators
• Two retrospective case studies demonstrate the usefulness of the model
Outline

• Motivation
• LogCA
• Results
• Conclusion
Why Need a Model?

“An accelerator is a separate architectural substructure ... that is architected using a different set of objectives than the base processor, ..., the accelerator is tuned to provide higher performance ... than with the general-purpose base hardware.”
Why a Model?

Encryption algorithm on UltraSPARC T2

- Host outperforms
- Accelerator outperforms
- Break-even point

Better

Time (ms)

Block Size (Bytes)

Amdahl’s Law for Accelerators
Why a Model?

![Graph showing speedup vs. offloaded data (Bytes)](image)

Running the same kernel, accelerators can have different break-even points
Outline

• Motivation
• LogCA
• Results
• Conclusion
The Performance Model

- Inspired by LogP [CACM 1996]
- Abstract accelerator using five parameters
  - \( L \) Latency: Cycles to move data
  - \( o \) Overhead: Setup cost
  - \( g \) Granularity: Size of the off-loaded data
  - \( C \) Computational index: Amount of work done per byte of data
  - \( A \) Acceleration: Speedup ignoring overheads
- Sixth parameter \( \beta \) generalizes to kernels with non-linear complexity
The Performance Model

- Execution w/o an accelerator
  \[ T_0(g) = C_0(g) \]
- Execution with one accelerator
  \[ T_1(g) = o_1(g) + L_1(g) + C_1(g) \]
Granularity independent latency

- Captures the effect of granularity on speedup
- Speedup bounded by acceleration
  \[ \lim_{g \to \infty} \text{Speedup} (g) = A \]
- Overheads dominate at smaller granularities
  \[ \text{Speedup}(g)_{g=1} = \frac{c}{o + L} + \frac{c_A}{o + L} < \frac{c}{o + L} \]

Amdahl’s law for Accelerators
Performance Metrics

• Right amount of off-loaded data?
• Inspired from vector machine metrics $N_v$, $N_\frac{1}{2}$

  • $g_1$: Granularity for a speedup of 1
    – $g_1$ is essentially independent of acceleration
    – Identify complexity of the interface

  • $g_A$: Granularity for a speedup of $\frac{A}{2}$
    – Increasing $A$ also increases $\frac{g_A}{2}$
Granularity dependent latency

- Speedup bounded by computational intensity $C/L$
  \[ \lim_{g \to \infty} Speedup(g) < \frac{C}{L} \]  \textit{(linear algorithms)}

- Speedup for sub-linear algorithms asymptotically decreases with the increase in granularity
Granularity dependent latency

- Computational intensity must be greater than 1 to achieve any speedup.
- Computational intensity should be greater than peak performance to achieve A/2.

Performance metrics help programmers early in the design cycle.
Bottleneck Analysis using LogCA

- 10X change in parameter ➔ 20% performance gain
- Helps focus on performance bottlenecks
Outline

• Motivation
• LogCA
• Results
• Conclusion
Experimental Methodology

• Fixed-function and general-purpose accelerators
  – Cryptographic accelerators on SPARC architectures
  – Discrete and integrated GPUs

• Kernels with varying complexities
  – Encryption, Hashing, Matrix Multiplication, FFT, Search, Radix Sort

• Retrospective case studies
  – Cryptographic interface in SPARC architectures
  – Memory interface in GPUs
Case Study I
Cryptographic Interface in the SPARC Architecture

PCle Crypto Accelerator

UltraSPARC T2

SPARC T3

SPARC T4 engine

SPARC T4 instructions
Conclusion

• Simple models effective in predicting performance of accelerators
• Proposed a high-level performance model for hardware accelerators
• These models help programmers and architects visually identify bottlenecks and suggest optimizations
• Performance metrics for programmers in deciding the right amount of offloaded data
• Limitations include inability to model resource contention, caches, and irregular memory access patterns
Questions?

Source: http://www.medarcade.com/