A Roofline Model of Energy

Jee Whan Choi
Georgia Institute of Technology

Richard Vuduc
Georgia Institute of Technology

Introduction

- Energy-based analogue of the time-based roofline model.
- Directly connects properties of an algorithm with architectural time and energy costs.
- Contributions:
  - Energy cost model.
  - Energy “arch line.”
  - Energy-balance.
  - Time-energy balance gap.
- Work-communication trade-off.
- Experimental results.

Target Systems

- PowerMon 2 intercepts and measures DC voltage and current in time while the new algorithm is compute-bound.
- Baseline and new algorithm are compute-bound in time.

Tools

- PowerMon 2 intercepts and measures DC voltage and current.
- PICe Interposer intercepts signals coming from the motherboard’s PCIe connector.
- Highly optimized benchmarks achieve high % of peak performance.
- Model instantiated using linear regression to find \( E_{\text{flop}} \), \( E_{\text{mem}} \), and \( \pi_{\text{flop}} \).

Energy Cost Model

- \( T = \max\left( W \tau_{\text{flop}}, Q\tau_{\text{mem}} \right) \)
- \( E = W\epsilon_{\text{flop}} + Q\epsilon_{\text{mem}} + \pi_{\text{flop}}T \)
- \( P = \frac{\tau_{\text{flop}}}{\eta_{\text{flop}}} \left[ \min\left(I, B_c\right) - \frac{\hat{B}_t(I)}{B_c} + \frac{\hat{B}_t(I)}{\max\left(I, B_c\right)} \right] \)

Architectures and Models

- Cortex-A9 TI OMAP 4460
- Kepler GTX 680
- Ivy Bridge i3-3217U

Peak performance

- Single (Double) GFLOPs
- Peak memory bandwidth GB/s
- TDP Watts

Future Work

- Can we save energy by trading off work (FLOPs) for communication?
  - Assume a baseline algorithm that consumes energy of \( E_{\text{f},1} \).
  - A “new” algorithm that reduces communication by a factor of \( m \) in exchange for increasing work by a factor of \( f \) consumes \( E_{\text{f},m} \).

- We define a new energy-efficiency metric “Greenup” as
  \[ \Delta E = \frac{E_{\text{f},1}}{E_{\text{f},m}} \]

- We can now compare three cases in relating speedup \( \Delta T \) and greenup \( \Delta E \):

  **Case 1**: Baseline and new algorithm are memory-bound in time
  \[ I < B_c \]
  \[ \frac{1}{f} < \Delta T = m \]
  \[ \frac{1}{f} < \Delta E < \frac{1}{f} \frac{B_c}{B_c} \]

  **Case 2**: Baseline is memory-bound in time while the new algorithm is compute-bound
  \[ I < B_c \]
  \[ \frac{1}{f} \Delta T = \frac{1}{f} \frac{B_c}{B_c} < \Delta E < \frac{1}{f} \frac{B_c}{B_c} \]

  **Case 3**: Baseline and new algorithm are compute-bound in time
  \[ B_c < I \]
  \[ \frac{1}{f} < \Delta T < \frac{1}{f} \]
  \[ \frac{1}{f} \frac{B_c}{B_c} < \Delta E < \frac{1}{f} \frac{B_c}{B_c} \]

- Visualized:
  - Each point corresponds to \( (\Delta T, \Delta E) \) pair
  - Indicates that improving time while incurring a loss in energy-efficiency is unlikely
  - Converse is possible, particularly in case 2 and case 3.

- Test our theory on real applications
- Fast Multipole Method (FMM)
- Convolution (FFT vs. stencil)
- 2D vs. 2.5D vs. 3D matrix multiplication
- Quantum Chromodynamics (QCD)