Announcements:

- HW3
- Project
- Control
Application in Computation

Translate the bank analogy to computers:

\[ P \leq \min(P_{\text{peak}}, I \cdot b) \]

Which parts of this are task-dependent?

\[ \sim P_{\text{peak}} \quad \sim I \quad \sim b \]

Hager et al. ‘17
A Graphical Representation: 'Roofline'

Plot (often log-log, but not necessarily):

- X-Axis: Intensity
- Y-Axis: Performance

What does our inequality correspond to graphically?

\[ P \leq \min(P_{\text{peak}}, I \cdot b) \]

What does the shaded area mean?

'In': attainable
'Out': not attainable
Example: Vector Addition

```c
float r, s, a[N];
for (i=0; i<N; ++i)
    a[i] = r + s * a[i];
```

Find the parameters and make a prediction.

\[
M
\]

\[
\begin{align*}
b &= 10 \text{ GB/s} \\
P_{\text{peak}} &= 70 \text{ GF/s} \\
\frac{1}{B} &= \frac{\frac{2 \text{ flops}}{8 \text{ bytes}}}{B} = \frac{1}{q} \frac{1}{B}
\end{align*}
\]

Performance

\[
(\text{not to scale})
\]

\[
70 \text{ GF/s}
\]

\[
P \leq \min \left( T_0, P_{\text{peak}} \right)
\]
Refining the Model

- $P_{\text{max}}$: Applicable peak performance of a loop, assuming that data comes from the fastest data path (this is not necessarily $P_{\text{peak}}$)
- Computational intensity ("work" per byte transferred) over the slowest data path utilized
- $b$: Applicable peak bandwidth of the slowest data path utilized

Hager et al. ‘17
Calibrating the Model: Bandwidth

Typically done with the STREAM benchmark.
Four parts: Copy, Scale, Add, Triad $a[i] = b[i] + s \cdot c[i]$

Do the four measurements matter?

**No, weak attempt at modeling I**

Any pitfalls?

**Non-temporal stores**

McCalpin: STREAM
Calibrating the Model: Peak Throughput

Name aspects that should/could be factored in when determining peak performance:

- # execution units
- dependence
- pipeline depth

# cycles / loop trip
Practical Tool: IACA

**Question:** Where to obtain an estimate of $P_{\text{max}}$?

**Demo:** perf/Forming Architectural Performance Expectations

**Questions:**

- What does IACA do about memory access? / the memory hierarchy?
An Example: Exploring Titan V Limits

- Memory bandwidth: 652 GB/s theoretical, 540 GB/s achievable
- Scratchpad / L1 throughput:
  80 (cores) × 32 (simd width) × 4 (word bytes) × 1.2 (base clock) \(\sim= 12.288\) TB/s
- Theoretical peak flops of 6.9 TFLOPS/s [Wikipedia]

Warburton ‘18
Rooflines: Assumptions

What assumptions are built into the roofline model?

- Steady-state
- Latency effects excluded

Important to remember:

- It is what you make of it—the better your calibration, the more info you get
- But: Calibrating on experimental data loses predictive power (e.g. SPMV)
Modeling Parallel Speedup: A ‘Universal’ Scaling Law

Develop a model of throughput $X(N)$ for a given load $N$, assuming execution resources scale with $N$.

$$X(N) = \frac{NY}{1 + \alpha (N-1) + \beta N(N-1)}$$

- $\alpha$: models queuing
- $\beta$: models incoherence (all-to-all)
- $\gamma$: perfect scalability

[Gunther ‘93]
Outline

Introduction

Machine Abstractions

Performance: Expectation, Experiment, Observation
  Forming Expectations of Performance
  Timing Experiments and Potential Issues
  Profiling and Observable Quantities
  Practical Tools: perf, toplevel, likwid

Performance-Oriented Languages and Abstractions

Polyhedral Representation and Transformation

Code Generation and Just-in-Time Compilation
Timing Experiments: Pitfalls

What are potential issues in timing experiments? (What can you do about them?)

- Compiler opt.
- Overheads
- Timing noise
  - Other people
  - Human resolution
  - Chunk size sufficiently big
  - Know your clock sources
    - RTC, monolothic, process, user, TSC (nominal, core clock)
Timing Experiments: Pitfalls (part 2)

What are potential issues in timing experiments? (What can you do about them?)
Combining Multiple Measurements

How can one combine multiple performance measurements? (e.g. "average speedup"?)

Example: Which computer should you buy?

<table>
<thead>
<tr>
<th>Execution time [s]</th>
<th>Computer A</th>
<th>Computer B</th>
<th>Computer C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program 1</td>
<td>1</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>Program 2</td>
<td>1000</td>
<td>100</td>
<td>20</td>
</tr>
</tbody>
</table>

$$\text{with } \frac{1}{n} \sum_{i=1}^{n} a_i$$

$$\text{good: } \sqrt{a_1 \cdots a_n} \leftarrow \text{ranking independent of scale}$$

\[ \text{good-bad: not necessarily pick a weighted} \]