Announcements

• Project updates, related work.

• HW#8 was due

• Big Data: Last HW not big data sadly, maybe power
  Don’t have gigabytes of data, let alone petabytes

• How to get your exabyte of data to the cloud?
  https://aws.amazon.com/snowmobile/

• Midterm in a week. Will go over next time.
Cluster Computing Power

Can spend a whole class (i.e. ECE571) discussing where power goes in a modern computing system.
Cluster Computing Power

Why is low-power super-computing important?
Green500

- Green 500 list
- Push for more accurate power reporting in the Top500 list
- Top 5, Nov 2016

1. NVIDIA DGX-1 Xeon/Tesla 350kW, 9.462 GFLOPs/W
2. (#8) Swiss Piz Daint Cray XC50 Xeon/Tesla 1.3MW, 7.453 GFLOPs/W
3. Riken ZettaScaler Xeon/PEZY-SCnp  
   2 ARM Cores/1024 RISC Cores, 1.5TFLOPs  
   150kW, 6.7GFLOPs/W  
4. (#1) Sunway TaihuLight, Sunway, 15MW,  
   6.0GFLOPs/W  
5. Fujitsu PRIMERGY, Xeon Phi, 77kW, 5.8GFLOPs/W
Pi-cluster Power

If we had more time I would have had you read *A Raspberry Pi Cluster Instrumented for Fine-Grained Power Measurement* by Cloutier, Paradis, and Weaver.

- Push for low power computers! Maybe even ARM!
- Low power, but floating-point so-so. Even worse is I/O (networking)
- Finally getting close
<table>
<thead>
<tr>
<th>Machine</th>
<th>N</th>
<th>GFLOPS</th>
<th>Idle</th>
<th>Active</th>
<th>GFLOPS/W</th>
<th>GFLOPS/$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi 2B</td>
<td>10,000</td>
<td>1.47</td>
<td>1.8</td>
<td>3.4</td>
<td>0.432</td>
<td>42.0</td>
</tr>
<tr>
<td>Pi 3B</td>
<td>10,000</td>
<td>3.7/6.4</td>
<td>1.8</td>
<td>4.4</td>
<td>0.844</td>
<td>106</td>
</tr>
<tr>
<td>Jetson TX-1</td>
<td>20,000</td>
<td>16.0</td>
<td>2.1</td>
<td>13.4</td>
<td>1.20</td>
<td>26.7</td>
</tr>
<tr>
<td>16x Haswell-EP</td>
<td>80,000</td>
<td>428</td>
<td>58.7</td>
<td>201</td>
<td>2.13</td>
<td>107</td>
</tr>
<tr>
<td>24xpi-cluster</td>
<td>48,000</td>
<td>15.5</td>
<td>71.3</td>
<td>93.1</td>
<td>0.166</td>
<td>7.75</td>
</tr>
</tbody>
</table>

- Per-node power measurement
- Network I/O big problem
- Why not OrangePi? (UTK)
- What if we could use the Pi GPU? No OpenCL, but people have reverse engineered part, also QPU?
SuperComputer Power

- Cooling
- DVFS
- Power-capping
- Up to 12% spent by the interconnect
  Pi cluster, 90W, 20W or so is the ethernet switch
Fujitsu K Computer, 2012

- [https://www.extremetech.com/extreme/120071-how-the-worlds-fastest-supercomputer-fujitsus-k-saves-on-power-and-money](https://www.extremetech.com/extreme/120071-how-the-worlds-fastest-supercomputer-fujitsus-k-saves-on-power-and-money)

- Fine tune voltage for each CPU (variation in production). Save 7W/CPU (one Megawatt total)

- Watercooled
Titan Supercomputer, 2012


- Was an upgrade, had to install 18,688 CPUs and GPUs manually

- 480V input to cabinets (rather than 208V) to reduce cable thickness

- 9MW, building gets 25MW
• Not big enough UPS for whole machine, flywheel UPS to keep I/O nodes up until diesel kicks in

• Cabinets are air cooled, but air is water-cooled first
Power-Capping

Power Capping: a Prelude to Power Shifting by Lefurgy Wang, and Ware

- Traditionally you have to design for the “worst-case” thermal and power behavior
- Often this will leave some resources underutilized “over-provisioned”
- Power-capping – let you design cheaper power/thermal setup, and if the CPU detects it is getting too hot/too much power automatically slows things down
Measuring Power and Energy

- Sense resistor or Hall Effect sensor gives you the current
- Sense resistor is small resistor. Measure voltage drop. Current $V=IR$ Ohm’s Law, so $V/R=I$
- Voltage drops are often small (why?) so you made need to amplify with instrumentation amplifier
- Then you need to measure with A/D converter
- $P = IV$ and you know the voltage
- How to get Energy from Power?
Definitions

People often say Power when they mean Energy

- Dynamic Power – only consumed while computing
- Static Power – consumed all the time. Sets the lower limit of optimization
Units

- Energy – Joules, kWh (3.6MJ), Therm (105.5MJ), 1 Ton TNT (4.2GJ), eV \(1.6 \times 10^{-19}\) J, BTU (1055 J), horsepower-hour (2.68 MJ), calorie (4.184 J)
- Power – Energy/Time – Watts (1 J/s), Horsepower (746W), Ton of Refrigeration (12,000 Btu/h)
- Volt-Amps (for A/C) – same units as Watts, but not same thing
- Charge – mAh (batteries) – need voltage to convert to Energy
CPU Power and Energy
CMOS Dynamic Power

- $P = C \Delta V V_{dd} \alpha f$
  Charging and discharging capacitors big factor $(C \Delta V V_{dd})$ from $V_{dd}$ to ground
  $\alpha$ is activity factor, transitions per clock cycle
  $f$ is frequency

- $\alpha$ often approximated as $\frac{1}{2}$, $\Delta V V_{dd}$ as $V_{dd}^2$ leading to
  $P \approx \frac{1}{2} C V_{dd}^2 f$

- Some pass-through loss ($V$ momentarily shorted)
CMOS Dynamic Power Reduction

How can you reduce Dynamic Power?

- Reduce $C$ – scaling
- Reduce $V_{dd}$ – eventually hit transistor limit
- Reduce $\alpha$ (design level)
- Reduce $f$ – makes processor slower
CMOS Static Power

• Leakage Current – bigger issue as scaling smaller. Forecast at one point to be 20-50% of all chip power before mitigations were taken.

• Various kinds of leakage (Substrate, Gate, etc)

• Linear with Voltage: $P_{\text{static}} = I_{\text{leakage}}V_{dd}$
Leakage Mitigation

- SOI – Silicon on Insulator (AMD, IBM but not Intel)
- High-k dielectric – instead of SO2 use some other material for gate oxide (Hafnium)
- Transistor sizing – make only critical transistors fast; non-critical can be made slower and less leakage prone
- Body-biasing
- Sleep transistors
Total Energy

- $E_{tot} = [P_{dynamic} + P_{static}]t$

- $E_{tot} = [(C_{tot}V_{dd}^2 \alpha f) + (N_{tot}I_{leakage}V_{dd})]t$
Delay

\[ T_d = \frac{C_L V_{dd}}{\mu C_{ox} \left( \frac{W}{L} \right) (V_{dd} - V_t)} \]

- Simplifies to \( f_{MAX} \sim \frac{(V_{dd} - V_t)^2}{V_{dd}} \)
- If you lower \( f \), you can lower \( V_{dd} \)
Thermal Issues

- Temperature and Heat Dissipation are closely related to Power
- If thermal issues, need heatsinks, fans, cooling
Metrics to Optimize

- Power
- Energy
- MIPS/W, FLOPS/W (don't handle quadratic $V$ well)
- $Energy \times Delay$
- $Energy \times Delay^2$
Power Optimization

- Does not take into account time. Lowering power does no good if it increases runtime.
Energy Optimization

- Lowering energy can affect time too, as parts can run slower at lower voltages
Energy Optimization

Which is better?

5W
1W
50J
10 20 30 40 50

10 20 30 40 50
Energy Delay – Watt/t*t

- Horowitz, Indermaur, Gonzalez (Low Power Electronics, 1994)
- Need to account for delay, so that lowering Energy does not made delay (time) worse
- Voltage Scaling – in general scaling low makes transistors slower
- Transistor Sizing – reduces Capacitance, also makes transistors slower
• Technology Scaling – reduces V and power.

• Transition Reduction – better logic design, have fewer transitions
  Get rid of clocks? Asynchronous? Clock-gating?
Which is better?

- **200W**
  - $E = 200\text{J}$
  - $ED = 200\text{Js}$
  - $EDD = 200\text{Jss}$

- **50W**

- **1**

- **2**

- **200W**
  - $E = 100\text{J}$
  - $ED = 200\text{Js}$
  - $EDD = 400\text{Jss}$

- **50W**

- **1**

- **2**
Energy Delay Squared – \( E \times t^2 \)

- Martin, Nyström, Pénzes – Power Aware Computing, 2002

- Independent of Voltage in CMOS

- ED can be misleading
  \[ E_a = 2E_b, \ t_a = \frac{t_b}{2} \]
  Reduce voltage by half, \( E_a = \frac{E_a}{4}, \ t_a = 2t_a, \ E_a = \frac{E_b}{2}, \ t_a = t_b \)
• Can have arbitrary large number of delay terms in Energy product, squared seems to be good enough
Energy-Delay Product Redux

Roughly based on data from “Energy-Delay Tradeoffs in CMOS Multipliers” by Brown et al.
## Raw Data

<table>
<thead>
<tr>
<th>Delay</th>
<th>Energy</th>
<th>$ED$</th>
<th>$ED^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>130</td>
<td>390</td>
<td>1170</td>
</tr>
<tr>
<td>3.5</td>
<td>100</td>
<td>350</td>
<td>1225</td>
</tr>
<tr>
<td>3.8</td>
<td>85</td>
<td>323</td>
<td>1227</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td><strong>300</strong></td>
<td>1200</td>
</tr>
<tr>
<td>4.5</td>
<td>70</td>
<td>315</td>
<td>1418</td>
</tr>
<tr>
<td>5</td>
<td>65</td>
<td>325</td>
<td>1625</td>
</tr>
<tr>
<td>5.5</td>
<td>58</td>
<td>319</td>
<td>1755</td>
</tr>
<tr>
<td>6</td>
<td>55</td>
<td>330</td>
<td>1980</td>
</tr>
<tr>
<td>6.5</td>
<td><strong>50</strong></td>
<td>390</td>
<td>2535</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>400</td>
<td>3200</td>
</tr>
</tbody>
</table>
Other Metrics

- *Energy – Delay*^n^ – choose appropriate factor

- *Energy – Delay – Area*^2^ – takes into account cost (die area) [McPAT]

- Power-Delay – units of Energy – used to measure switching

- Energy Delay Diagram – [SWEEP]
Measuring Power and Energy
Why?

- New, massive, HPC machines use impressive amounts of power
- When you have 100k+ cores, saving a few Joules per core quickly adds up
- To improve power/energy draw, you need some way of measuring it
Energy/Power Measurement is Already Possible

Three common ways of doing this:

• Hand-instrumenting a system by tapping all power inputs to CPU, memory, disk, etc., and using a data logger
• Using a pass-through power meter that you plug your server into. Often these will log over USB
• Estimating power/energy with a software model based on system behavior