Energy efficiency: A down-to-earth perspective

Georg Hager
Erlangen Regional Computing Center (RRZE)
University of Erlangen-Nuremberg
Erlangen, Germany

Cool Supercomputing BoF @ SC12, Nov 14, 2012
Assumptions & naive conclusions

1. The **lifetime** of a typical machine is **constant** (4-6 years)
2. **Energy** costs account for a **significant** fraction of TCO (especially in Europe)
3. **Machines** are almost **100%** utilized
4. **Domain scientists** have no idea about
   - Performance optimization
   - Connection between performance and power bill for *their* jobs

Straightforward conclusions:

- Install automatic mechanisms to **automagically clock down** CPUs in apps not sensitive to clock speed
- Use “**application slack**” to clock down/power down individual cores
My point of view: Get the low-hanging fruits first!

Application optimization is the first and easiest way to save energy.
**Example:**
*A medical image reconstruction code on Sandy Bridge*

Sandy Bridge EP (8 cores, 2.7 GHz base freq.)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8 cores, plain C</td>
<td>90.43</td>
<td>90</td>
<td>8110</td>
</tr>
<tr>
<td>8 cores, SSE</td>
<td>29.63</td>
<td>93</td>
<td>2750</td>
</tr>
<tr>
<td>8 cores (SMT), SSE</td>
<td>22.61</td>
<td>102</td>
<td>2300</td>
</tr>
<tr>
<td>8 cores (SMT), AVX</td>
<td>18.42</td>
<td>111</td>
<td>2040</td>
</tr>
</tbody>
</table>

Faster code $\Rightarrow$ less energy
Get the low-hanging fruits first!

Load imbalance is better removed from the start

Energy to solution:

\[ E_{slack} \approx T \cdot (W_{static} + W_{dyn}) = T \cdot W_{static} + T \cdot W_{dyn} \]

Energy to solution:

\[ E_{opt} \approx \frac{T}{2} \cdot (W_{static} + 2 \cdot W_{dyn}) = \frac{T}{2} \cdot W_{static} + T \cdot W_{dyn} \]
Second-order effects

Performance and energy to solution of LBM solver

- Optimize first, then clock down!

Race-to-idle regime

Saturation point
Conclusions

Low-hanging fruits for power efficiency

- Think about a “Science per Joule” metric
- Remove load imbalance for better resource utilization
- Single-core (and then parallel) optimization for
  - Shorter time to solution
  - Earlier in-socket saturation
- Train application programmers to get the fallen fruits themselves (zeroth order)!

And then, if there’s time, think about the third order:

- Power capping
- Efficient power distribution
- Automatic, profile-guided DVFS
- ...