Application-aware online power control

Valentin Reis, Argo ECP project @ ANL.
Goal: Design and prototype system-level software for exascale.

- Dealing with the new memory hierarchy,
- Node-level, container-based resource partitioning,
- **Power management across the machine,**
- Support for new HPC workloads (workflows, in-situ, steering)
Changing the role of an operating system

The traditional OS stack manages resources for sharing. We want to also manage resources to optimize the application performance to energy tradeoff.
Changing the role of an operating system

The traditional OS stack manages resources for sharing. We want to also manage resources to optimize the application performance to energy tradeoff.

<table>
<thead>
<tr>
<th>NRM Controls</th>
<th>NRM Measures</th>
<th>NRM Optimizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU throttling</td>
<td>operations per second</td>
<td>runtime</td>
</tr>
<tr>
<td>task pinning</td>
<td>core energy consumption</td>
<td>energy</td>
</tr>
<tr>
<td>core allocations</td>
<td>application performance</td>
<td>. . .</td>
</tr>
<tr>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
</tbody>
</table>

What we are doing now: **control policies**. What should be our optimization objective? Let’s look at the example of RAPL power capping.
Figure: Energy expenditure for some example application runs (sample size 10)
(an) Offline objective: Runtime $r$

**Figure:** runtimes some example application runs (sample size 10)
Offline linearization: $\lambda \times r + (1 - \lambda) \times e$

**Figure:** linearized objective for some example application runs (sample size 10, $\lambda=0.3$)
Online metrics: counters, application progress

Figure: online metrics for LAMMPS (sample size 10)
• similar application-aware resource control work
• underlying measurement/control libraries