SLA-Based Scheduling of Bag-of-Tasks Applications on Power-Aware Cluster Systems

Kyong Hoon KIM, Wan Yeon LEE, Jong KIM and Rajkumar BUYYA
Outline

• Introduction
• SLA-Based Power-Aware Cluster Systems
• EDF-Based DVS Scheduling
• Proportional Share-Based DVS Scheduling
• Simulation Results
• Conclusion
Introduction

• Service Level Agreements (SLAs) are obligations like resource usage, price, quality-of-service between consumers and producers.

• Efficient power management of cluster systems becomes important issue not only for reducing the operational cost but also for system reliability.

• Recent processors support DVS with multiple operating points that can run at low voltage or frequency to reduce the power consumption.

• In this paper, we consider deadline as QoS metric in SLAs of users’ applications and focus on the problem of reducing energy consumption of applications with SLAs.
SLA-Based Power-Aware Cluster Systems (1/6)

- Power Model

\[
E = E_{\text{dynamic}} + E_{\text{static}}
\]

\[
E_{\text{dynamic}} = k_1 V_{dd}^2 N_{\text{cycl}} = k_1 V^2 L
\]

\[
E_{\text{static}} = k_2 E_{\text{dynamic}}
\]

\[
E = (k_1 V^2 L) + k_2 (k_1 V^2 L)
\]

\[
E = (1 + k_2)k_1 V^2 L = \alpha V^2 L
\]
Execution Time Model

\[ T(f_i) = T(f_{max}) \times \left[ \beta \left( \frac{f_{max}}{f_i} - 1 \right) + 1 \right] \]

\[ S_{i,j} = \frac{T(f_i)}{T(f_{max})} = \frac{1}{\beta_j \left( \frac{f_{max}}{f_i} - 1 \right) + 1} \]
SLA-Based Power-Aware Cluster Systems (3/6)

- Job Model

\[ J = (p, \{l_1, l_2, \ldots, l_p\}, \beta, d) \]

- \( p \): Number of sub-tasks
- \( l \): Number of instruction of i-th task in Million Instructions
- \( \beta \): CPU-boundness parameter
- \( d \): Deadline
SLA-Based Power-Aware Cluster Systems (4/6)

• Job Admission Control

Fig. 1  Resource allocation framework.

1  Job submission
2  Schedulability test & Energy estimation
3  Acknowledgement of schedulability and energy amount
4  Selection of PEs
Algorithm `SLA_based_Job_Admission (J)`
/* - J = (p, \{l_1, \cdots, l_p\}, \beta, d) : a new job  
 - N : the number of processing elements  
*/
1: for i from 1 to p do
2:   \(PE_{alloc} \leftarrow null\);
3:   energy_{min} \leftarrow \text{MAX\_VALUE};
4:   for k from 1 to N do
5:     if (scheduled \(PE_k, l_i, d\) == true) then
6:       energy_{k} \leftarrow \text{energy\_estimate} \(PE_k, l_i, \beta, d\);
7:     if energy_{k} < energy_{min} then
8:       energy_{min} \leftarrow energy_{k};
9:       PE_{alloc} \leftarrow PE_k;
10:   endif
11:   endif
12: endfor
13: if PE_{alloc} \neq null then
14:   Allocate the i-th task of J to PE_{alloc}.
15: else
16:   Cancel all jobs of J.
17:   return reject;
18: endelse
19: endfor
20: return accept;

Algorithm `schedulable_EDF (PE_k, e, d)`
/* - e : the execution time of a task at frequency \(f_{\text{max}}\)  
 - d : the deadline of a task  
*/
1: \(T_{k'} \leftarrow T_k \cup \{(e, d)\};\)
2: Sort \(T_{k'}\) in the order of deadline.
3: for i from 1 to \(n_k + 1\) do
4:   \(u_{k',i} \leftarrow \frac{\sum_{j=1}^{i} e_{k',j}}{d_{k',i}};\)
5:   if \(u_{k',i} > 1\) then return false;
6: return true;

\[ u_{k',4} = \frac{1 + 2 + 2 + 6}{10} = 1.1 \]
SLA-Based Power-Aware Cluster Systems (6/6)

Algorithm energy_estimate_EDF \((PE_k, e, \beta, d)\)
/* - e : the execution time of a task at frequency \(f_{max}\)
 - d : the deadline of a task */
Construct the relative speed levels of the task using \(\beta\).
\[E_{current} \leftarrow energy\_consumption (T_k, n_k)\];
\[T_k' \leftarrow T_k \cup \{(e, d)\}\];
\[E_{new} \leftarrow energy\_consumption (T_k', n_k+1)\];
return \((E_{new} - E_{current})\);

function energy\_consumption \((T, n)\)
/* - T : a task set
 - n : the number of tasks
 - t\_current : the current time */
Energy \(\leftarrow 0\);
time \(\leftarrow t\_current\);
for \(i\) from 1 to \(n\) do
  for \(j\) from \(i\) to \(n\) do
    \(u_j \leftarrow \sum_{k=1}^{j} e_k / d_j\);
    \(\hat{s} \leftarrow \max_{j=i}^{n} \{u_j\}\);
    \(v \leftarrow \min_{j=1}^{m} \{V_j | S_{j,i} \geq \hat{s}\}\);
    \(s \leftarrow \min_{j=1}^{m} \{S_{j,i} | S_{j,i} \geq \hat{s}\}\);
    Energy \(\leftarrow\) Energy + \(\alpha v^2 l_i\);
    time \(\leftarrow\) time + \(e_i / s\);
  endfor
  for \(j\) from \(i\) to \(n\) do
    \(d_j \leftarrow d_j - e_i / s\);
endfor
return Energy;

\( (1,4) \quad (2,6) \quad (2,10) \quad (3,10) \)

\[u_1 = \frac{1}{4} \quad u_2 = \frac{3}{6} \quad u_3 = \frac{5}{10} \quad u_4 = \frac{8}{10} \]

\(\hat{s} = \frac{8}{10}\) \((v, s) = (1.3v, 0.8)\)

\[Energy += \alpha \times (1.3)^2 \times \left(\frac{1}{0.8} \times 10^6\right)\]

\(\frac{2}{6 - 1.25} \quad \frac{4}{10 - 1.25} \quad \frac{7}{10 - 1.25}\)

\(\hat{s} = \frac{7}{8.75} = 0.8\) \((v, s) = (1.1v, 0.8)\)

\[Energy += \alpha \times (1.1)^2 \times \left(\frac{2}{0.8} \times 10^6\right)\]

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Frequency</th>
<th>Time</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9 V</td>
<td>0.8 GHz</td>
<td>0.4</td>
<td>0.7</td>
</tr>
<tr>
<td>1.1 V</td>
<td>1.2 GHz</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>1.3 V</td>
<td>1.6 GHz</td>
<td>0.8</td>
<td>0.9</td>
</tr>
<tr>
<td>1.5 V</td>
<td>2.0 GHz</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>
EDF-Based DVS Scheduling (1/2)

- Task set in k-th PE

\[ T_k = \{ \tau_{k,i}(e_{k,i}, d_{k,i}) | i = 1, \ldots, n_k \} \]

- \( e_{k,i} \): Remaining execution time at \( f_{max} \)
- \( d_{k,i} \): Relative deadline

- \( T_k \) is sorted by the deadline ascending order. The scheduler always executes the earliest-deadline task in the queue.

- Utilization

\[ u_{k,i} = \frac{\sum_{j=1}^{i} e_{k,j}}{d_{k,i}} \]

\[ \tilde{s}_k = \max_{i=1}^{n_k} \{ u_{k,i} \} \]

\[ v_k = \min_{i=1}^{m} \{ V_i | S_{i,1} \geq \tilde{s}_k \} \]

<table>
<thead>
<tr>
<th>Table 1</th>
<th>An example of energy model.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage ((V_i))</td>
<td>Frequency ((f_i))</td>
</tr>
<tr>
<td>(S_{i,1})</td>
<td>(S_{i,2})</td>
</tr>
<tr>
<td>0.9 V</td>
<td>0.8 GHz</td>
</tr>
<tr>
<td>1.1 V</td>
<td>1.2 GHz</td>
</tr>
<tr>
<td>1.3 V</td>
<td>1.6 GHz</td>
</tr>
<tr>
<td>1.5 V</td>
<td>2.0 GHz</td>
</tr>
</tbody>
</table>
EDF-Based DVS Scheduling (2/2)

• Example

\[ T_k = \{ \tau_{k,1}(1,4), \tau_{k,2}(2,6), \tau_{k,3}(2,10) \} \]

- Time 0:
  - \( u_{k,1} = \frac{1}{4}, u_{k,2} = \frac{1+2}{6}, u_{k,3} = \frac{1+2+2}{10} \)
  - \( \tilde{s}_k = 0.5, V_j|S_{i,1} = 1.1V|0.6 \)
  - \( T = 1/S_{i,1} = 1.67s \)

- Time 1.67:
  - \( u_{k,2} = \frac{2}{6-1.67} = 0.46, u_{k,3} = \frac{2+2}{10-1.67} = 0.48 \)
  - \( \tilde{s}_k = 0.48, V_j|S_{i,2} = 0.9V|0.7 \)
  - \( T = 2/S_{i,2} = 2.85s \)

- Time 4.52:
  - \( u_{k,3} = \frac{2}{10-4.52} \)
  - \( \tilde{s}_k = 0.36, V_j|S_{i,3} = 0.9V|0.55 \)
  - \( T = 2/S_{i,3} = 3.64s \)

Fig. 3 An example of DVS-based EDF scheduling.
Proportional Share-Based DVS Scheduling (1/3)

- Each task requires $e_{k,i}/d_{k,i}$ utilization
- The all required utilization is $\sum \frac{e_{k,i}}{d_{k,i}}$

\[- share_{k,i} = \frac{e_{k,i}}{d_{k,i}} / \sum_{l=1}^{n_k} \frac{e_{k,l}}{d_{k,l}}\]

\[- v_k = \max_{i=1}^{n_k} \min_{j=1}^{m} \left\{ V_j \mid S_{j,i} \geq \sum_{l=1}^{n_k} \frac{e_{k,l}}{d_{k,l}} \right\}\]

<table>
<thead>
<tr>
<th>Voltage ($V_i$)</th>
<th>Frequency ($f_i$)</th>
<th>Relative Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$S_{i,1}$</td>
</tr>
<tr>
<td>0.9 V</td>
<td>0.8 GHz</td>
<td>0.4</td>
</tr>
<tr>
<td>1.1 V</td>
<td>1.2 GHz</td>
<td>0.6</td>
</tr>
<tr>
<td>1.3 V</td>
<td>1.6 GHz</td>
<td>0.8</td>
</tr>
<tr>
<td>1.5 V</td>
<td>2.0 GHz</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Proportional Share-Based DVS Scheduling (2/3)

- Example

\[ T_k = \{ \tau_{k,1}(1,4), \tau_{k,2}(2,6), \tau_{k,3}(2,10) \} \]

- Time 0:
  - \( u_{k,1} = \frac{1}{4}, u_{k,2} = \frac{2}{6}, u_{k,3} = \frac{2}{10} \)
  - \( \Sigma u = 0.78, V_j|S_{i,1} = 1.3V|0.8 \)
  - \( T = 3.92s \)

- Time 3.92:
  - \( u_{k,2} = \frac{0.5}{6-3.92}, u_{k,3} = \frac{1.15}{10-3.92} \)
  - \( \Sigma u = 0.43, V_j|S_{i,2} = 0.9V|0.7 \)
  - \( T = 1.28s \)

- Time 5.19:
  - \( u_{k,3} = \frac{0.84}{6.08-1.28} \)
  - \( \Sigma u = 0.17, V_j|S_{i,3} = 0.9V|0.55 \)
  - \( T = 1.53s \)

Fig. 6 An example of DVS-based proportional share scheduling.
Proportional Share-Based DVS Scheduling (3/3)

<table>
<thead>
<tr>
<th>Task</th>
<th>Task name</th>
<th>Execution time</th>
<th>Deadline</th>
<th>( u_{k,i} )</th>
<th>Round 1</th>
<th>( \Sigma u )</th>
<th>( S_{k,i} )</th>
<th>( share_{k,i} )</th>
<th>Task i relative speed</th>
<th>T</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
<td>4</td>
<td>0.25</td>
<td></td>
<td>0.78</td>
<td>0.80</td>
<td>0.32</td>
<td>0.26</td>
<td><strong>3.92</strong></td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>6</td>
<td>0.33</td>
<td></td>
<td></td>
<td>0.90</td>
<td>0.43</td>
<td>0.38</td>
<td><strong>5.22</strong></td>
<td>75%</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>2</td>
<td>10</td>
<td>0.20</td>
<td></td>
<td></td>
<td>0.85</td>
<td>0.26</td>
<td>0.22</td>
<td><strong>9.22</strong></td>
<td>43%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remaining Execution time</th>
<th>Remaining Deadline</th>
<th>( u_{k,i} )</th>
<th>Round 2</th>
<th>( \Sigma u )</th>
<th>( S_{k,i} )</th>
<th>( share_{k,i} )</th>
<th>Task i relative speed</th>
<th>T</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.50</td>
<td>2.08</td>
<td>0.24</td>
<td></td>
<td>0.43</td>
<td>0.70</td>
<td>0.56</td>
<td><strong>0.39</strong></td>
<td><strong>1.28</strong></td>
</tr>
<tr>
<td>3</td>
<td>1.15</td>
<td>6.08</td>
<td>0.19</td>
<td></td>
<td></td>
<td>0.55</td>
<td>0.44</td>
<td><strong>0.24</strong></td>
<td><strong>4.75</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remaining Execution time</th>
<th>Remaining Deadline</th>
<th>( u_{k,i} )</th>
<th>Round 3</th>
<th>( \Sigma u )</th>
<th>( S_{k,i} )</th>
<th>( share_{k,i} )</th>
<th>Task i relative speed</th>
<th>T</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.84</td>
<td>4.81</td>
<td>0.17</td>
<td></td>
<td>0.17</td>
<td>0.55</td>
<td>1.00</td>
<td><strong>0.55</strong></td>
<td><strong>1.53</strong></td>
</tr>
</tbody>
</table>

- Task i-th relative speed = \( S_{v_{k,i}} \times share_{k,i} \)
- \( T = e_k / \text{relative speed} \)
Simulation Results (1/6)

• Simulation Environment
  – GridSim Toolkit
  – 32 DVS-enabled processors
  – 1000 bag-of-tasks jobs
  – The number of tasks in a job: randomly from 2 and 32
  – Jobs deadline: 20% to 100% of average execution time at 1.4GHz
  – Assume all tasks have the same $\beta$ (Same relative speed)
  – Inter-arrival time follows a Poisson distribution

Table 2  Operating points of simulated processor.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Voltage</th>
<th>Relative Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8 GHz</td>
<td>0.9 V</td>
<td>0.4</td>
</tr>
<tr>
<td>1.0 GHz</td>
<td>1.0 V</td>
<td>0.5</td>
</tr>
<tr>
<td>1.2 GHz</td>
<td>1.1 V</td>
<td>0.6</td>
</tr>
<tr>
<td>1.4 GHz</td>
<td>1.2 V</td>
<td>0.7</td>
</tr>
<tr>
<td>1.6 GHz</td>
<td>1.3 V</td>
<td>0.8</td>
</tr>
<tr>
<td>1.8 GHz</td>
<td>1.4 V</td>
<td>0.9</td>
</tr>
<tr>
<td>2.0 GHz</td>
<td>1.5 V</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Fig. 7  Job acceptance ratio.
Simulation Results (3/6)

![Graph showing average number of jobs vs. inter-arrival time for different systems.](image)

**Fig. 9** Jobs per accepted task.
Fig. 8  Energy consumption of job normalized to EDF-1.5V at inter-arrival time of 2 mins.
Simulation Results (5/6)

**Fig. 10**  Normalized energy consumption per job.

**Table 3**  Normalized performance of DVS.

<table>
<thead>
<tr>
<th>Inter-arrival time (min)</th>
<th>EDF-DVS vs. EDF-1.5V</th>
<th>PShare-DVS vs PShare-1.5V</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Energy Reduction (%)</td>
<td>Acceptance Degradation (%)</td>
</tr>
<tr>
<td>2</td>
<td>13.6</td>
<td>13.3</td>
</tr>
<tr>
<td>3</td>
<td>21.3</td>
<td>13.0</td>
</tr>
<tr>
<td>4</td>
<td>31.2</td>
<td>11.2</td>
</tr>
<tr>
<td>5</td>
<td>34.4</td>
<td>7.9</td>
</tr>
<tr>
<td>6</td>
<td>38.6</td>
<td>6.3</td>
</tr>
<tr>
<td>7</td>
<td>41.5</td>
<td>4.0</td>
</tr>
<tr>
<td>8</td>
<td>44.3</td>
<td>2.7</td>
</tr>
</tbody>
</table>
Simulation Results (6/6)

Fig. 11  EDF-DVS vs. PShare-DVS.
Conclusion

• The proposed SLA-based scheduling algorithms select appropriate supply voltages of processing elements to minimize energy consumption, which decreases the operational cost and increases the system reliability.

• Simulation results show that both DVS schemes reduce much energy consumption with little degradation of deadline missing.

• We plan to conduct further research on SLA-based scheduling on multicore-based cluster systems.