Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS

Author: Lizhe Wang, Gregor von Laszewski, Jai dayal
Speaker: Jong Youl Cho
Community Grids Lab, Indian University
Outlook

• Background
• Problem definition
• Proposed algorithm
• Evaluation
• Conclusion
Background

• Parallel task scheduling
  – Static scheduling
  – Dynamic scheduling
• Dynamic voltage and frequency scaling (DVFS)
• Power aware task scheduling with DVFS
DVFS model

\[ V = \bigcup_{1 \leq m \leq M} \{v_m\} \]  \hspace{1cm} (1)

\[ F = \bigcup_{1 \leq m \leq M} \{f_m\} \]  \hspace{1cm} (2)

where,

- $v_m$ is the $m$-th processor operating voltage;
- $f_m$ is the $m$-th processor operating frequency;
- $v_{\text{min}} = v_1 \leq v_2 \leq \ldots \leq v_M = v_{\text{max}}$;
- $f_{\text{min}} = f_1 \leq f_2 \leq \ldots \leq f_M = f_{\text{max}}$;
- $1 \leq m \leq M$, $M$ is the total number of processor operating points.
Energy model

The energy consumption

\[ \xi = \sum_{\Delta t} (\delta \cdot v^2 \cdot f \cdot \Delta t) \]  

(8)

Where,
\( \delta \) is a constant determined by the PE.
\( v \) is the processor operating voltage during \( \Delta t \);
\( f \) is the processor operating frequency during \( \Delta t \);
\( \Delta t \) is a time period.
Cluster model

- \( p_{e_k} \cdot v^{op} \in V \) is the processor operating voltage
- \( p_{e_k} \cdot f^{op} \in F \) is the processor operating frequency

1 \( \leq k \leq K \), \( K \) is the total number of PEs.

A cluster \( C \) is defined by its set of processing elements

\[
C = \bigcup_{1 \leq k \leq K} \{p_{e_k}\} \tag{9}
\]
Job model

DAG model: $T = (J, E)$

$$J = \bigcup_{1 \leq n \leq N} \{job_n\}$$  \hspace{1cm} (10)

A job, $job_n$, has 3 properties:
- $weight$ is the instruction number of $job_n$.
- $t^{st}$ is the starting time of $job_n$.
- $t$ is the execution time of $job_n$. If $job_n$ is executed on $pe_k$, the job execution time is calculated as follows:

$$job_n.t = \frac{job_n.weight \cdot CPI}{pe_k.f^{op}}$$  \hspace{1cm} (11)
Job model

- \( E \): a set of precedence constraints (edges in a DAG). 
  \( E \) defines partial orders (operational precedence constraints) on \( J \). \( e_{ij} \) is an edge between \( job_i \) and \( job_j \), it means that \( job_i \) must be completed before \( job_j \) can begin, 
  \( 1 \leq i, j \leq N, job_i, job_j \in J \). \( e_{ij} \) sometime can also be represented \( job_i < job_j \).
  
  \( e \) has one property:
  \( e_{ij}.cost \geq 0 \), is the amount of data required to be transferred from \( job_i \) to \( job_j \), \( 1 \leq i, j \leq N, job_i, job_j \in J \). Data are transferred from the PE where \( job_i \) is executed to the PE where \( job_j \) is executed.
Problem definition (1)

• Problem 1: Best-effort scheduling
  – Schedule parallel tasks to a cluster
  – Minimize the makespan
  – Reduce energy consumption without increasing the makespan
Problem definition (2)

• energy-performance tradeoff scheduling
  – Users can adopt some performance loss, for example, increase the makespan
  – Schedule tasks to a cluster, minimize the energy consumption
Best Effort Scheduling Algorithm (1)

• schedule tasks via the ETF scheduling algorithm
• scale down PE’s voltages for all non-critical jobs
ETF scheduling algorithm

• ETF: Early task first algorithm
• Compute priorities for all tasks
  – Currently we use b_level, which is the long length from a task to the exist node
• Sort all tasks
• Put tasks that ready to execute in the ready queue with task priority
• Select the first task from ready queue
• Select a resource for this task, so as to give the earliest task finish time
• Loop this scheduling till all tasks are scheduled
Scale down non-critical tasks

- for all PEs
  - for all time slots in this PE
    - If this time slot executes a communication or this time slot is idle
    - Then scale down the voltage of this PE in this time slot
    - If this time slot execute a non-critical task
    - Then scale down the voltage of this PE in this time slot
Example DAG

- A
- B
- C
- D
- E
- F
Energy-performance tradeoff scheduling algorithm

• Execute Early task first algorithm (ETF)
• Scale down PE’s voltages for critical tasks with the predefined acceptable performance loss rate.
• Scale down PE’s voltages for non-critical jobs
Evaluation (1)

• Simulation study:
  – MT43 processor
  – Use synthetic DAG generation tool

• Results

<table>
<thead>
<tr>
<th>Energy aware DAG scheduling algorithm</th>
<th>Maximum energy saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>EADUS &amp; TEBUS [28]</td>
<td>16.8%</td>
</tr>
<tr>
<td>Energy Reduction Algorithm [31]</td>
<td>25%</td>
</tr>
<tr>
<td>LEneS [22]</td>
<td>28%</td>
</tr>
<tr>
<td>ECS [30]</td>
<td>38%</td>
</tr>
<tr>
<td>Our algorithm</td>
<td>44.3%</td>
</tr>
</tbody>
</table>
Result(2)
Conclusion and future work

• We study energy aware cluster scheduling algorithms
• Two research issues are studied
  – Best-effort scheduling issue
  – Energy-performance tradeoff issue
• We proposed two algorithms
• Future work
  • Workload characterization
  • Runtime support and implementation