Towards Automatic Grid Application Performance Modeling

Bo Liu, Fang Liu, Albert M.K. Cheng
Dept of Computer Science, University of Houston, Houston TX 77204
Contact Author: (boliu@cs.uh.edu)

Objectives
An online scheduling algorithm for Multiple Feasible Intervals on low-power Multiple Processor System-on-Chip (MPSoC) platforms aims to:
- reduce Power consumption subject to the operation temperature threshold
- reduce deadline miss ratios and improve throughput

Thermal Energy Issues
- Leakage power consumption: the result of leakage currents in CMOS circuits
- Leakage power consumption increases exponentially with the temperature increase
- High rising operation temperature: raises power consumption and reduces the system's reliability and lifetime
- Clock timing is sensitive to temperature variation
- and usually decreases the performance of circuits

Thermal Model
- Temperature function
  \[ \Theta(T) = (\Theta(T_0) - a \cdot f_c / b) e^{\Delta T} + a \cdot f_c / b \]
  where a, b, and f_c are curve fitting constants, and \( \Theta(T_0) \) is the operation temperature at any time \( T_0 \), and \( f_c \) is a constant operation frequency.
- Operation frequency function
  \[ f(x) = b \cdot (\theta - \Delta \theta) \]
  where a, b, and \( \theta \) are the same as in the above, and \( \Delta \theta \) is a constant temperature.
- Temperature change function
  \[ \Delta \Theta = b \cdot (\Theta(T_0) - a \cdot f_c / b) e^{\Delta T} \]
  where a, b, and \( \Theta(T_0) \) and \( f_c \) are the same as in the above.

MFI Examples
- Multiple Feasible Intervals
  - The tasks are not required to be periodic
  - Tasks are supposed to be independent and have arbitrary release time and deadline
  - A task instance in a periodic task is thus considered as an independent task in our system
  - The tasks' intervals are known after they are released

Integrate Thermal Model
- The system is subject to a temperature restriction set by the temperature threshold \( \Theta^* \) besides timing constraints.
- Each on-chip-processor is characterized by a tuple \((\theta_{current}, \Delta \theta)\).
- Temperature changes \( \Delta \theta \) are calculated based on task execution time and predicted task execution time, and then \( \theta_{current} \) is updated with \( \Delta \theta \).
- \( \theta_{current} \) is compared with \( \Theta^* \) to decide if a task can be assigned to a processor.
- \( \theta_{current} \) is compared with \( \Theta^* \) to decide if a processor should be halted or not.
- Re-assign the un-executed tasks to other processors.

Online Dynamic MFI Algorithm
1) Put \( T_i \) into \( Queue_{-}T() \) when there is less than \( N \) tasks in the queue and within time period \( j \rightarrow c_i \).
2) Merge \( L_j \) into \( Queue_{-}IE() \) and \( Queue_{-}IS() \).
3) While \( Queue_{-}IE() \) is not empty:
   - Check the head node in the queue.
   - If the interval has empty task link, move head pointer to next interval until found an interval \( I \), \( J \) with non-empty task link;
     - Choose scale factor and return true if there is un-passable slack
     - If boolean
       - If fetch ahead a feasible task
       - If perfect fetch ahead
     - Release memory of chosen interval in \( Queue_{-}IE() \), \( Queue_{-}IS() \) and corresponding task node in \( Queue_{-}T() \), and move head pointer to the next; run task \( T_i \).

Offline Performance Prediction
- Architecture specific performance model based on statistical fitting of a limited number of parameters to execution times from a limited number of cases

Experiment Design
- Future work
  - Energy efficiency
  - Deadline miss ratio
  - Algorithm overhead

Acknowledgements
- We acknowledge and sincerely appreciate
  - NSF financial support
  - University of Houston partners
  - Lamar University partners

Project3d Cross Platform Prediction
Make3diter Cross Platform Prediction
Hardware Counters Prediction
Accurate Models!!!
Predictable Behavior!!!

Problem Size
CPU_CYCLES
Hardware Counters Prediction
Accurate Models!!!
Predictable Behavior!!!

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>CPU_CYCLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1E+11</td>
<td>2E+11</td>
</tr>
<tr>
<td>2E+11</td>
<td>3E+11</td>
</tr>
<tr>
<td>3E+11</td>
<td>4E+11</td>
</tr>
<tr>
<td>4E+11</td>
<td>5E+11</td>
</tr>
<tr>
<td>6E+11</td>
<td>7E+11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>CPU_CYCLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1E+11</td>
<td>2E+11</td>
</tr>
<tr>
<td>3E+11</td>
<td>2E+11</td>
</tr>
<tr>
<td>4E+11</td>
<td>2E+11</td>
</tr>
<tr>
<td>5E+11</td>
<td>2E+11</td>
</tr>
<tr>
<td>6E+11</td>
<td>2E+11</td>
</tr>
<tr>
<td>7E+11</td>
<td>2E+11</td>
</tr>
</tbody>
</table>

Acknowledgements
- We acknowledge and sincerely appreciate
  - NSF financial support
  - University of Houston partners
  - Lamar University partners

Experiment Design
- Future work
  - Energy efficiency
  - Deadline miss ratio
  - Algorithm overhead