Conflict Avoidance Scheduling using Grouping List for Transactional Memory

*Dongmin Choi*, **Seung Hun Kim**, and **Won W. Ro**

*Mobile Communication Division
Samsung Electronics Co., Ltd.
Gumi, Republic of Korea

**School of Electrical and Electronic Engineering
Yonsei University
Seoul, Republic of Korea

The 17th International Workshop in High-Level Parallel Programming Models and Supportive Environments
Overview

- **Transactional Memory systems**
  - A promising synchronization method for a parallel processing
  - Overcoming the *limitations of lock-based synchronization*
    - dead-lock, live-lock, priority inversion, convoying, less parallelism ...
  - Lock based synchronization

![Diagram of thread synchronization using lock-based and transactional memory systems showing a critical section and time critical section.](attachment:thread_synchronization_diagram.png)

---

Dongmin Choi, Seung Hun Kim, and Won W. Ro

HIPS 2012
Overview

- **Transactional Memory systems**
  - A promising synchronization method for a parallel processing
  - Overcoming the *limitations of lock-based* synchronization
    - dead-lock, live-lock, priority inversion, convoying, less parallelism ...
  - Flow of transactions

  ![Diagram of transactional memory system]

  - eliminating mutual exclusion
  - a hazard for a correct synchronization = conflict
  - rollback to initial state
Overview

- **Transactional Memory systems**
  
  - A promising synchronization method for a parallel processing
  
  - Overcoming the *limitations of lock-based synchronization*
    
    → dead-lock, live-lock, priority inversion, convoying, less parallelism ...
  
  - Flow of transactions

- **Performance degradation factor**
  
  - Frequent rollback operation!
    
    → especially, in the case of high contention with large-sized transactions
  
  - Proposed solution: *Conflict Avoidance Scheduling* (CAS) for transactions
    
    → 23% of performance improvement (avg.) over the original system
Outline

- Related Work
  - Contention management
  - Prior work

- Conflict Avoidance Scheduling Mechanism
  - Scheduling operation
  - Implementation

- Results and Analysis
  - Execution time analysis
  - Contention analysis

- Conclusion
Contention Management

- **Reducing abort penalty**
  - Conflicted transactions control

  (a) Conventional Back-off based Contention Management

  (b) Concept of Conflict Avoidance Scheduling

- **Scheduling using grouping list**
  - Grouping list: history of the conflicted threads
  - Conflict avoidance using the relationship among the threads
Contention Management

- **Reducing abort penalty**
  - Conflicted transactions control

- **Scheduling using grouping list**
  - Grouping list: history of the conflicted threads
  - Conflict avoidance using the relationship among the threads

(a) Conventional Back-off based Contention Management

(b) Concept of Conflict Avoidance Scheduling
Prior Work

➢ Design dimensions of TM systems
  • Conflict detection, version management, and contention management
    → diverse approaches: hardware, software, or their hybrid
    → various designs for the performance improvement: FlexTM, EazyHTM, FASTM ...
  • Transaction scheduling
    → not oriented to reactive contention management

➢ Adaptive Transaction Scheduling (ATS), Yoo et al., 2008
  • Measuring “contention intensity”
  • Serialized execution of all transactions in a single queue
    → over serialization

➢ Proactive Transaction Scheduling (PTS), Blake et al., 2009
  • Threads swapping based on the prediction of conflicts
  • Software and hardware overhead
Outline

- Related Work
  - Contention management
  - Prior work

- Conflict Avoidance Scheduling Mechanism
  - Scheduling operation
  - Implementation

- Results and Analysis
  - Execution time analysis
  - Contention analysis

- Conclusion
Scheduling Approach

- **Approach**
  - Concurrent execution control
  - Reducing the amount of contention

- **Grouping list**
  - Same group: threads that have high possibility of the conflict
  - Dedicated grouping list for each threads
  - List information
    - the ID of the threads in the same group and the running status of the threads
  - List makeup
    - measuring the number of repeated conflicts
    - compared to the predefined threshold
  - Repeatedly conflicts
    - additional conflicts in the future
Conflict Avoidance Scheduling

- **Scheduling operation**
  - Access on the grouping list for every transactional begin, restart, abort, and commit

**Initial status**

```
Th1  Th2  Th3 ...
```

```
Running Status
```

```
<table>
<thead>
<tr>
<th>Grouping List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th1</td>
</tr>
<tr>
<td>Th2</td>
</tr>
<tr>
<td>Th3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
```
Conflict Avoidance Scheduling

- **Scheduling operation**
  - Access on the grouping list for every transactional begin, restart, abort, and commit

Conflicts

```
| Th1  | Th2  | Th3 ... |
```

**Grouping List**

```
Th1  Th4  Th5  Th6
```
```
Th2  Th4  Th5
```
```
Th3  Th7  Th8  Th9
```
```
Conflict Avoidance Scheduling

- **Scheduling operation**
  - Access on the grouping list for every transactional begin, restart, abort, and commit

![Diagram showing scheduling operations and grouping lists for Th1, Th2, and Th3]
Conflict Avoidance Scheduling

- **Scheduling operation**
  - Access on the grouping list for every transactional begin, restart, abort, and commit

**Completion of thread 1 to 3**

![Diagram showing scheduling operation]

- Th7 Th8 Th9 ...
Conflict Avoidance Scheduling

Scheduling operation

• Access on the grouping list for every transactional begin, restart, abort, and commit

Status change

Running Status

Th7 Th8 Th9 ...

Grouping List

Th1

Th2

Th3

Th4 Th5 Th6

Th4 Th5 Th7 Th8 Th9

Th7 Th8 Th9 Th10 Th11

...
Scheduling operation

- Access on the grouping list for every transactional begin, restart, abort, and commit

Prohibition of start execution

Running Status

Reservation Station

Grouping List

<table>
<thead>
<tr>
<th>Th1</th>
<th>Th4</th>
<th>Th5</th>
<th>Th6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th2</td>
<td>Th7</td>
<td>Th8</td>
<td>Th9</td>
</tr>
<tr>
<td>Th3</td>
<td>Th7</td>
<td>Th8</td>
<td>Th9</td>
</tr>
<tr>
<td>Th</td>
<td>Th8</td>
<td>Th9</td>
<td>Th10</td>
</tr>
</tbody>
</table>

Waiting

Dongmin Choi, Seung Hun Kim, and Won W. Ro

HIPS 2012
Conflict Avoidance Scheduling

- Scheduling operation
  - Access on the grouping list for every transactional begin, restart, abort, and commit

Increasing in running status

Running Status

Waiting

Reservation Station

Grouping List

Th6  Th7  Th8  Th9  Th10  Th11

Th6  Th7  Th8  Th9  Th10  Th11

Dongmin Choi, Seung Hun Kim, and Won W. Ro

HIPS 2012
Conflict Avoidance Scheduling

- **Scheduling operation**
  - Access on the grouping list for every transactional begin, restart, abort, and commit

**Waiting on reservation station**

- Running Status

- Reservation Station

- Grouping List

Dongmin Choi, Seung Hun Kim, and Won W. Ro
Conflict Avoidance Scheduling

- Scheduling operation
  - Access on the grouping list for every transactional begin, restart, abort, and commit

Commit of thread 6

- Reservation Station
- Grouping List
  - Th4
  - Th5
  - Th6
  - Th7
  - Th8
  - Th9
  - Th10
  - Th11

Running Status
- Commit
- Waiting
Conflict Avoidance Scheduling

- **Scheduling operation**

  - Access on the grouping list for every transactional begin, restart, abort, and commit

**Status changing**

- Running Status
- Reservation Station
- Grouping List

Dongmin Choi, Seung Hun Kim, and Won W. Ro
HIPS 2012
Conflict Avoidance Scheduling

- **Scheduling operation**
  - Access on the grouping list for every transactional begin, restart, abort, and commit
Implementation

- **CAS hardware architecture**
  - Centralized CAS manager and distributed CAS registers
  - CAS register → grouping list
  - Management of the list → CAS manager

- Set 1 *enable to run* signal → 0 for all threads in the same group
Outline

- Related Work
  - Contention management
  - Prior work

- Conflict Avoidance Scheduling Mechanism
  - Scheduling operation
  - Implementation

- Results and Analysis
  - Performance improvement
  - Execution time and contention analysis

- Conclusion
Simulation Environment

- Simulator and TM system
  - **Simics**, a full system simulation platform (set as 16-core CMP)
  - **GEMS** (General Execution-driven Multiprocessor Simulator) from Wisconsin’s Multifacet Project
    - The baseline TM system: LogTM-SE of GEMS

- Benchmark suite
  - **STAMP** (Stanford Transactional Applications for Multi-Processing)
  - Large transactions with high contention than traditional programs

<table>
<thead>
<tr>
<th>Application</th>
<th>Input Parameters</th>
<th>Contention</th>
</tr>
</thead>
<tbody>
<tr>
<td>genome</td>
<td>-t15 -g256 -s16 -n16384</td>
<td>Low</td>
</tr>
<tr>
<td>kmeans</td>
<td>-p15 -m15 -n15 -t0.05 -i random -n2048 -d16 -c16</td>
<td>Low</td>
</tr>
<tr>
<td>ssca2</td>
<td>-t15 -s13 -i1.0 -u1.0 -l3 -p3</td>
<td>Low</td>
</tr>
<tr>
<td>vacation</td>
<td>-c15 -n8 -q10 -u80 -r65536 -t4096</td>
<td>Medium</td>
</tr>
<tr>
<td>bayes</td>
<td>-t15 -v32 -r1024 -n2 -p20 -i2 -e2</td>
<td>High</td>
</tr>
<tr>
<td>intruder</td>
<td>-t15 -a10 -l16 -n4096 -s1</td>
<td>High</td>
</tr>
<tr>
<td>labyrinth</td>
<td>-t15 -i random -x16 -y16 -z3 -n16</td>
<td>High</td>
</tr>
<tr>
<td>yada</td>
<td>-t15 -a20 -i 633.2</td>
<td>Medium</td>
</tr>
</tbody>
</table>
Performance Improvement

- Reduced number of aborts and execution time
  - Compared to the baseline system
  - Effectiveness of the proposed method
    → in the contention medium/high applications
  - No harm in contention-low applications
  - 23% of average improvement for eight applications
Execution Cycles Analysis

- Cycle analysis during the parallel phase of each program
  - Metric: seven different cycles
    - Parallel processing using TM systems
Cycle analysis during the parallel phase of each program

- Necessary Wasted Cycles (NWCs)
  - sum of the cycles used for stall, back-off, flush, and abort
- Factors of improving the performance
  - the amount of NWCs, contention
Execution Cycles Analysis

- Cycle analysis during the parallel phase of each program
  - Necessary Wasted Cycles (NWCs)
    - sum of the cycles used for stall, back-off, flush, and abort
  - Factors of improving the performance
    - the amount of NWCs, contention

Extremely small amount of NWC $\rightarrow$ low expected gain

- Graph showing normalized amount of cycles for various programs:
  - Genome
  - Kmeans
  - Ssca2
  - Vacation
  - LogTM CAS
  - Bayes
  - Intruder
  - Labyrinth
  - Yada

- NWCs:
  - STALL
  - BACKOFF
  - FLUSH
  - ABORT
  - GOOD_TX
  - BARRIER
  - NON_TX
Execution Cycles Analysis

- Cycle analysis during the parallel phase of each program
  - Necessary Wasted Cycles (NWCs)
    → sum of the cycles used for stall, back-off, flush, and abort
  - Factors of improving the performance
    → the amount of NWCs, contention

Normalized amount of cycles

<table>
<thead>
<tr>
<th>Normalized Execution Time</th>
<th>STALL</th>
<th>BACKOFF</th>
<th>FLUSH</th>
<th>ABORT</th>
<th>GOOD_TX</th>
<th>BARRIER</th>
<th>NON_TX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genome</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
<td>1</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>Kmeans</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSCA2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vacation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intruder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labyrinth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yada</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Large portion of NWC

0.13/0.17
Execution Cycles Analysis

- Cycle analysis during the parallel phase of each program
  - Necessary Wasted Cycles (NWCs)
    - sum of the cycles used for stall, back-off, flush, and abort
  - Factors of improving the performance
    - the amount of NWCs, contention
Contention Analysis

- Prohibiting of concurrent threads execution of the same group

<table>
<thead>
<tr>
<th></th>
<th>bayes rate</th>
<th>results</th>
<th>intruder rate</th>
<th>results</th>
<th>labyrinth rate</th>
<th>results</th>
<th>yada rate</th>
<th>results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th1</td>
<td>Th2 20.73%</td>
<td>0.647</td>
<td>Th15 9.00%</td>
<td>0.767</td>
<td>Th15 51.61%</td>
<td>0.344</td>
<td>Th7 15.34%</td>
<td>1</td>
</tr>
<tr>
<td>Th2</td>
<td>Th6 24.28%</td>
<td>0.235</td>
<td>Th15 8.70%</td>
<td>0.84</td>
<td>Th15 49.18%</td>
<td>0.367</td>
<td>Th12 14.59%</td>
<td>0.439</td>
</tr>
<tr>
<td>Th3</td>
<td>Th10 41.67%</td>
<td>0.55</td>
<td>Th15 8.28%</td>
<td>0.709</td>
<td>Th15 45.20%</td>
<td>0.333</td>
<td>Th12 14.33%</td>
<td>0.425</td>
</tr>
<tr>
<td>Th4</td>
<td>Th3 33.33%</td>
<td>0.5</td>
<td>Th15 9.61%</td>
<td>0.682</td>
<td>Th15 56.41%</td>
<td>0.5</td>
<td>Th2 16.30%</td>
<td>0.307</td>
</tr>
<tr>
<td>Th5</td>
<td>Th2 33.33%</td>
<td>0.5</td>
<td>Th15 8.89%</td>
<td>0.766</td>
<td>Th15 50.00%</td>
<td>0.379</td>
<td>Th11 13.88%</td>
<td>0.667</td>
</tr>
<tr>
<td>Th6</td>
<td>Th10 27.16%</td>
<td>0.545</td>
<td>Th15 8.89%</td>
<td>0.766</td>
<td>Th15 50.00%</td>
<td>0.379</td>
<td>Th13 13.88%</td>
<td>0.667</td>
</tr>
<tr>
<td>Th7</td>
<td>Th9 21.86%</td>
<td>0.285</td>
<td>Th15 8.65%</td>
<td>0.764</td>
<td>Th15 48.78%</td>
<td>0.55</td>
<td>Th14 12.13%</td>
<td>0.482</td>
</tr>
<tr>
<td>Th8</td>
<td>Th2 25.30%</td>
<td>1.142</td>
<td>Th15 8.73%</td>
<td>0.769</td>
<td>Th15 43.63%</td>
<td>0.458</td>
<td>Th14 13.78%</td>
<td>0.436</td>
</tr>
<tr>
<td>Th9</td>
<td>Th2 25.84%</td>
<td>0.521</td>
<td>Th15 8.62%</td>
<td>0.765</td>
<td>Th15 45.83%</td>
<td>0.5</td>
<td>Th15 66.66%</td>
<td>2</td>
</tr>
<tr>
<td>Th10</td>
<td>Th9 46.37%</td>
<td>0.375</td>
<td>Th15 9.03%</td>
<td>0.777</td>
<td>Th15 76.92%</td>
<td>0.55</td>
<td>Th2 13.97%</td>
<td>0.579</td>
</tr>
<tr>
<td>Th11</td>
<td>Th2 25.97%</td>
<td>0.65</td>
<td>Th15 9.55%</td>
<td>0.745</td>
<td>Th15 50.74%</td>
<td>0.324</td>
<td>Th2 12.33%</td>
<td>0.324</td>
</tr>
<tr>
<td>Th12</td>
<td>Th10 27.96%</td>
<td>0.091</td>
<td>Th15 9.03%</td>
<td>0.723</td>
<td>Th15 100.00%</td>
<td>0.526</td>
<td>Th2 14.47%</td>
<td>0.295</td>
</tr>
<tr>
<td>Th13</td>
<td>Th2 38.18%</td>
<td>0.619</td>
<td>Th15 9.54%</td>
<td>0.686</td>
<td>Th15 52.27%</td>
<td>0.478</td>
<td>Th4 14.78%</td>
<td>0.262</td>
</tr>
<tr>
<td>Th14</td>
<td>Th2 52.23%</td>
<td>0.257</td>
<td>Th8 7.86%</td>
<td>0.922</td>
<td>Th15 62.50%</td>
<td>0.55</td>
<td>Th2 18.45%</td>
<td>0.186</td>
</tr>
<tr>
<td>Th15</td>
<td>Th10 39.76%</td>
<td>0.307</td>
<td>Th3 8.66%</td>
<td>0.705</td>
<td>-</td>
<td>-</td>
<td>Th12 19.71%</td>
<td>0.214</td>
</tr>
</tbody>
</table>
## Contention Analysis

> Prohibiting of concurrent threads execution of the same group

<table>
<thead>
<tr>
<th></th>
<th>bayes rate</th>
<th>intruder results</th>
<th>labyrinth rate</th>
<th>yada results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th1</td>
<td>20.73%</td>
<td>0.647</td>
<td>51.61%</td>
<td>0.344</td>
</tr>
<tr>
<td>Th2</td>
<td>24.28%</td>
<td>0.235</td>
<td>49.18%</td>
<td>0.367</td>
</tr>
<tr>
<td>Th3</td>
<td>41.67%</td>
<td>0.5</td>
<td>45.20%</td>
<td>0.333</td>
</tr>
<tr>
<td>Th4</td>
<td>33.33%</td>
<td>0.5</td>
<td>56.41%</td>
<td>0.5</td>
</tr>
<tr>
<td>Th5</td>
<td>27.16%</td>
<td>0.545</td>
<td>50.00%</td>
<td>0.379</td>
</tr>
<tr>
<td>Th6</td>
<td>21.86%</td>
<td>0.0</td>
<td>12.13%</td>
<td>0.425</td>
</tr>
<tr>
<td>Th7</td>
<td>25.30%</td>
<td>0.1</td>
<td>14.59%</td>
<td>0.307</td>
</tr>
<tr>
<td>Th8</td>
<td>25.84%</td>
<td>0.2</td>
<td>66.66%</td>
<td>2</td>
</tr>
<tr>
<td>Th9</td>
<td>21.26%</td>
<td>0.481</td>
<td>47.05%</td>
<td>0.458</td>
</tr>
<tr>
<td>Th10</td>
<td>46.37%</td>
<td>0.375</td>
<td>76.92%</td>
<td>0.55</td>
</tr>
<tr>
<td>Th11</td>
<td>25.97%</td>
<td>0.65</td>
<td>50.74%</td>
<td>0.324</td>
</tr>
<tr>
<td>Th12</td>
<td>27.96%</td>
<td>0.091</td>
<td>100.00%</td>
<td>0.526</td>
</tr>
<tr>
<td>Th13</td>
<td>38.18%</td>
<td>0.627</td>
<td>52.27%</td>
<td>0.478</td>
</tr>
<tr>
<td>Th14</td>
<td>52.23%</td>
<td>0.257</td>
<td>62.50%</td>
<td>0.55</td>
</tr>
<tr>
<td>Th15</td>
<td>39.76%</td>
<td>0.307</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- **largest amount of conflict**
- **0.647 times decrease**

<table>
<thead>
<tr>
<th></th>
<th>Th1</th>
<th>Th2</th>
</tr>
</thead>
<tbody>
<tr>
<td>rate</td>
<td>20.73%</td>
<td>0.647</td>
</tr>
<tr>
<td>0.647 times decrease</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Contention Analysis

- Prohibiting of concurrent threads execution of the same group

<table>
<thead>
<tr>
<th></th>
<th>bayes rate</th>
<th>bayes results</th>
<th>intruder rate</th>
<th>intruder results</th>
<th>labyrinth rate</th>
<th>labyrinth results</th>
<th>yada rate</th>
<th>yada results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th1</td>
<td>20.73%</td>
<td>0.647</td>
<td>9.00%</td>
<td>0.767</td>
<td>51.61%</td>
<td>0.344</td>
<td>15.44%</td>
<td>1</td>
</tr>
<tr>
<td>Th2</td>
<td>24.28%</td>
<td>0.235</td>
<td>8.70%</td>
<td>0.84</td>
<td>49.18%</td>
<td>0.367</td>
<td>14.59%</td>
<td>0.439</td>
</tr>
<tr>
<td>Th3</td>
<td>41.67%</td>
<td>0.55</td>
<td>8.28%</td>
<td>0.709</td>
<td>45.20%</td>
<td>0.333</td>
<td>14.33%</td>
<td>0.425</td>
</tr>
<tr>
<td>Th4</td>
<td>33.33%</td>
<td>0.5</td>
<td>9.61%</td>
<td>0.682</td>
<td>56.41%</td>
<td>0.5</td>
<td>16.30%</td>
<td>0.307</td>
</tr>
<tr>
<td>Th5</td>
<td>27.16%</td>
<td>0.545</td>
<td>8.89%</td>
<td>0.766</td>
<td>50.00%</td>
<td>0.379</td>
<td>13.88%</td>
<td>0.667</td>
</tr>
<tr>
<td>Th6</td>
<td>21.86%</td>
<td>0.285</td>
<td>8.65%</td>
<td>0.764</td>
<td>48.78%</td>
<td>0.55</td>
<td>12.13%</td>
<td>0.482</td>
</tr>
<tr>
<td>Th7</td>
<td>25.30%</td>
<td>1.142</td>
<td>8.73%</td>
<td>0.769</td>
<td>43.63%</td>
<td>0.458</td>
<td>13.78%</td>
<td>0.436</td>
</tr>
<tr>
<td>Th8</td>
<td>25.84%</td>
<td>0.521</td>
<td>8.62%</td>
<td>0.765</td>
<td>45.93%</td>
<td>0.5</td>
<td>66.66%</td>
<td>0.2</td>
</tr>
<tr>
<td>Th9</td>
<td>21.26%</td>
<td>0.71</td>
<td>8.63%</td>
<td>0.76</td>
<td>45.93%</td>
<td>0.5</td>
<td>66.66%</td>
<td>0.2</td>
</tr>
<tr>
<td>Th10</td>
<td>46.37%</td>
<td>0.375</td>
<td>9.95%</td>
<td>0.743</td>
<td>50.74%</td>
<td>0.324</td>
<td>12.33%</td>
<td>0.324</td>
</tr>
<tr>
<td>Th11</td>
<td>25.97%</td>
<td>0.65</td>
<td>9.53%</td>
<td>0.743</td>
<td>30.74%</td>
<td>0.324</td>
<td>12.33%</td>
<td>0.324</td>
</tr>
<tr>
<td>Th12</td>
<td>27.96%</td>
<td>0.091</td>
<td>9.03%</td>
<td>0.723</td>
<td>100.00%</td>
<td>0.526</td>
<td>14.47%</td>
<td>0.295</td>
</tr>
<tr>
<td>Th13</td>
<td>38.18%</td>
<td>0.619</td>
<td>9.54%</td>
<td>0.686</td>
<td>52.27%</td>
<td>0.478</td>
<td>14.78%</td>
<td>0.262</td>
</tr>
<tr>
<td>Th14</td>
<td>52.23%</td>
<td>0.257</td>
<td>7.86%</td>
<td>0.922</td>
<td>62.50%</td>
<td>0.55</td>
<td>18.45%</td>
<td>0.186</td>
</tr>
<tr>
<td>Th15</td>
<td>39.76%</td>
<td>0.307</td>
<td>8.66%</td>
<td>0.705</td>
<td>-</td>
<td>-</td>
<td>19.71%</td>
<td>0.214</td>
</tr>
</tbody>
</table>

- Th7
  - 25.30%
  - 1142

---

Dongmin Choi, Seung Hun Kim, and Won W. Ro

HIPS 2012
## Contention Analysis

- **Prohibiting of concurrent threads execution of the same group**

<table>
<thead>
<tr>
<th></th>
<th>bayes rate</th>
<th>bayes results</th>
<th>intruder rate</th>
<th>intruder results</th>
<th>labyrinth rate</th>
<th>labyrinth results</th>
<th>yada rate</th>
<th>yada results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th1</td>
<td>20.73%</td>
<td>0.647</td>
<td>9.00%</td>
<td>0.767</td>
<td>51.61%</td>
<td>0.344</td>
<td>15.44%</td>
<td>1</td>
</tr>
<tr>
<td>Th2</td>
<td>24.28%</td>
<td>0.235</td>
<td>8.70%</td>
<td>0.84</td>
<td>49.18%</td>
<td>0.367</td>
<td>14.59%</td>
<td>0.439</td>
</tr>
<tr>
<td>Th3</td>
<td>41.67%</td>
<td>0.55</td>
<td>8.28%</td>
<td>0.709</td>
<td>45.20%</td>
<td>0.333</td>
<td>14.33%</td>
<td>0.425</td>
</tr>
<tr>
<td>Th4</td>
<td>33.33%</td>
<td>0.5</td>
<td>9.61%</td>
<td>0.682</td>
<td>56.41%</td>
<td>0.5</td>
<td>16.30%</td>
<td>0.307</td>
</tr>
<tr>
<td>Th5</td>
<td>27.16%</td>
<td>0.545</td>
<td>8.89%</td>
<td>0.766</td>
<td>50.00%</td>
<td>0.379</td>
<td>13.88%</td>
<td>0.667</td>
</tr>
<tr>
<td>Th6</td>
<td>21.86%</td>
<td>0.285</td>
<td>8.65%</td>
<td>0.764</td>
<td>48.78%</td>
<td>0.55</td>
<td>12.13%</td>
<td>0.482</td>
</tr>
<tr>
<td>Th7</td>
<td>25.30%</td>
<td>1.142</td>
<td>8.73%</td>
<td>0.769</td>
<td>43.63%</td>
<td>0.458</td>
<td>13.78%</td>
<td>0.436</td>
</tr>
<tr>
<td>Th8</td>
<td>25.84%</td>
<td>0.521</td>
<td>8.62%</td>
<td>0.765</td>
<td>45.83%</td>
<td>0.5</td>
<td>66.66%</td>
<td>2</td>
</tr>
<tr>
<td>Th9</td>
<td>21.26%</td>
<td>0.481</td>
<td>8.82%</td>
<td>0.674</td>
<td>47.05%</td>
<td>0.458</td>
<td>15.59%</td>
<td>0.382</td>
</tr>
<tr>
<td>Th10</td>
<td>46.37%</td>
<td>0.375</td>
<td>9.03%</td>
<td>0.777</td>
<td>76.92%</td>
<td>0.55</td>
<td>13.97%</td>
<td>0.579</td>
</tr>
<tr>
<td>Th11</td>
<td>25.97%</td>
<td>0.65</td>
<td>9.55%</td>
<td>0.745</td>
<td>50.74%</td>
<td>0.324</td>
<td>12.33%</td>
<td>0.324</td>
</tr>
<tr>
<td>Th12</td>
<td>27.96%</td>
<td>0.091</td>
<td>9.03%</td>
<td>0.723</td>
<td>100.00%</td>
<td>0.526</td>
<td>14.47%</td>
<td>0.295</td>
</tr>
<tr>
<td>Th13</td>
<td>38.18%</td>
<td>0.619</td>
<td>9.54%</td>
<td>0.686</td>
<td>52.27%</td>
<td>0.478</td>
<td>14.78%</td>
<td>0.262</td>
</tr>
<tr>
<td>Th14</td>
<td>52.23%</td>
<td>0.257</td>
<td>7.86%</td>
<td>0.922</td>
<td>62.50%</td>
<td>0.55</td>
<td>18.45%</td>
<td>0.186</td>
</tr>
<tr>
<td>Th15</td>
<td>39.76%</td>
<td>0.307</td>
<td>8.66%</td>
<td>0.705</td>
<td>-</td>
<td>-</td>
<td>19.71%</td>
<td>0.214</td>
</tr>
</tbody>
</table>
Outline

➢ Related Work
  • Contention management
  • Prior work

➢ Conflict Avoidance Scheduling Mechanism
  • Scheduling operation
  • Implementation

➢ Results and Analysis
  • Execution time analysis
  • Contention analysis

➢ Conclusion
Conclusion

- A new contention management policy
  - Providing fast program execution by reducing the abort penalty
    - 23% of average performance improvement
  - Proposal of the grouping list for the contention management
    - analysis based on the thread relationship

- Future work
  - Measuring overhead of the scheduling
Thank you!

- Questions, comments:
  - kseunghun@gmail.com
  - http://escal.yonsei.ac.kr
Supplement-1

bayes

intruder
Supplement-2

labyrinth

yada