LIRA: Adaptive Contention-Aware Thread Placement for Parallel Runtime Systems

Alexander Collins*, Tim Harris†, Murray Cole*, Christian Fensch‡

* University of Edinburgh
† Oracle Labs, UK
‡ Heriot Watt University
The Problem

- Multi-socket machines common-place
- Run multiple parallel programs
- Co-location affects performance
- *Which programs should we co-locate?*
The Problem

- System workload is constantly changing
- Best co-location changes
- *Need an online adaptive solution*
Our Insight

- Balance load instruction rate across sockets
Our Solution

• Schedule programs to sockets
• Maximise difference in load instruction rate (LIRA heuristic)
• Built on top of Callisto\textsuperscript{[1]}
• Each program pins one thread to each core
• One thread on each core is high priority
• High priority thread runs unless it stalls

\textsuperscript{[1]} Callisto: Co-scheduling Parallel Runtime Systems, Harris et al. EuroSys ‘14
Our Solution

--- Main thread
Our Solution

- Main thread
- Worker thread
Our Solution

Load Instruction Rates

- Main thread
- Worker thread

Shared memory
Our Solution
Our Solution

![Diagram of load instruction rates with different thread types: Main thread, Worker thread, Scheduler thread, and shared memory.](image-url)
Our Solution

Load Instruction Rates

Current Schedule

- Main thread
- Worker thread
- Scheduler thread
- Shared memory
Our Solution
Our Solution

Load Instruction Rates

Current Schedule

- Main thread
- Worker thread
- Scheduler thread
- Shared memory
Socket 1

Socket 2

Heuristic 13

Time
Evaluation

- 11 benchmarks from SPEC OpenMP 2001
- 4 from GreenMarl project
- 1 using CDDP (betweenness-centrality)
- Dual-socket Xeon E5-2660
- 8 cores each (hyperthreading disabled)
Evaluation

• Measure 32 combinations of four programs
• ANTT and STP system performance metrics
• Comparing:
  • Socket unaware Callisto
  • LIRA static tuning
  • LIRA adaptive tuning
Evaluation

% improvement in STP

-2% 0% +2% +4%
Callisto worst-static LIRA-static best-static LIRA-adaptive

% improvement in ANTT

-6% -2% 0% +2% +6%
Callisto worst-static LIRA-static best-static LIRA-adaptive
Conclusions

• Co-location affects performance
• Adaptive online tuning is required
• LIRA heuristic improves performance
• More details in the paper
LIRA: Adaptive Contention-Aware Thread Placement for Parallel Runtime Systems

Alexander Collins*, Tim Harris†, Murray Cole*, Christian Fensch‡

* University of Edinburgh
† Oracle Labs, UK
‡ Heriot Watt University
<table>
<thead>
<tr>
<th>Program measured</th>
<th>ammp</th>
<th>b331</th>
<th>bc</th>
<th>art</th>
<th>apsi</th>
<th>ammp</th>
</tr>
</thead>
<tbody>
<tr>
<td>wupwise</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>tricount</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>swim</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>spin</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>pagerank</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>md</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ilbdc</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>hop_dist</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>fma3d</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>equake</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dom_bc</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>cachehog</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bwaves</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bt331</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bc</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>art</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>apsi</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>ammp</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
The diagram shows box plots for different sampling delays. The x-axis represents the sampling delay in cycles, with categories 0.5e9 cycles, 1e9 cycles, 2e9 cycles, and LIRA-static. The y-axis represents the STP values. Each box plot indicates the distribution of STP values for each delay category.