Multi-dimensional Evaluation of Haswell’s Transactional Memory Performance

Marcio M. Pereira, Matthew Gaudet, J. Nelson Amaral, Guido Araújo

UNICAMP & University of Alberta

26th International Symposium on Computer Architecture and High Performance Computing
SBAC-PAD 2014
Transactional Memory

- an approach in concurrency control with reduced programming burden

- promises the performance of finer grain locks combined with lower programming complexity
Intel Transactional Synchronization Extensions (TSX)

- Provides two software interfaces:
  - Hardware Lock Elision (HLE)
  - Restricted Transactional Memory (RTM)
Intel Transactional Synchronization Extensions (TSX)

- Provides two software interfaces:
  - Hardware Lock Elision (HLE)
  - Restricted Transactional Memory (RTM)
Intel Transactional Synchronization Extensions (TSX)

- Provides two software interfaces:
  - Hardware Lock Elision (HLE)
  - Restricted Transactional Memory (RTM)
Goals

- do a performance study from the application perspective

- provide a precise evaluation of the strengths and weaknesses of this architectural feature
Main Findings

- TSX Performance is most sensitive to
  - transaction footprint
  - working-set size
  - transactional write ratio

- TSX Performance is also affected by the contention level
Main Findings

- TSX Performance is most sensitive to
  - transaction footprint
  - working-set size
  - transactional write ratio

- TSX Performance is also affected by the contention level
How to yield performance gains?

- **Low Contention Scenario**
  - Convert multiple locks (or critical sections) \(\Rightarrow\) single transactional region

- **High Contention Scenario**
  - Maintain transactions with smaller footprint
How to yield performance gains?

- **Low Contention Scenario**
  - Convert multiple locks (or critical sections) $\Rightarrow$ single transactional region

- **High Contention Scenario**
  - Maintain transactions with smaller footprint
More Findings

- **Increase Temporal locality**

- **Reduce Pollution (write ratio)**
  - e.g., data structures based on Hash

- **Right choice of fallback policy**
More Findings

- Increase Temporal locality
- Reduce Pollution (write ratio)
  - e.g., data structures based on Hash
- Right choice of fallback policy
Forward-Progress Policies

- MaxRetry Policy

- Backoff Policy

- SerControl Policy (← novelty)
Forward-Progress Policies

- MaxRetry Policy
- Backoff Policy
- SerControl Policy (← novelty)
Forward-Progress Policies

- MaxRetry Policy
- Backoff Policy
- SerControl Policy (← novelty)
Findings about the policies

- MaxRetry and Backoff
  - low overhead
  - sensitive to the tuning for serialization
  - can be detrimental to some applications

- SerControl
  - reduce potential conflicts
  - most successful in delivering performance
Findings about the policies

- **MaxRetry and Backoff**
  - low overhead
  - sensitive to the tuning for serialization
  - can be detrimental to some applications

- **SerControl**
  - reduce potential conflicts
  - most successful in delivering performance
Multi-dimensional Evaluation of Haswell’s Transactional Memory Performance

Marcio M. Pereira

Introduction

Contribution

Experimental Evaluation

Conclusion

Analysis of TSX using Eigenbench

- designed to enable independent exploration of the properties of a TM

- Methodology
  - initial empirical exploration of the space of these properties
  - systematic exploration of these space varying one value at a time
Analysis of TSX using Eigenbench

- designed to enable independent exploration of the properties of a TM

- Methodology
  - initial empirical exploration of the space of these properties
  - systematic exploration of these space varying one value at a time
Transaction Length - Performance & outcomes

---

**Introduction**

**Contribution**

**Experimental Evaluation**

**Conclusion**

---

Multi-dimensional Evaluation of Haswell’s Transactional Memory Performance

Marcio M. Pereira

---

Transaction Length - Performance & outcomes

---

![Graphs showing Speedup, Abort Fraction, and Serialized-Capacity fraction over transaction length for different write percentages.](image)
Multi-dimensional Evaluation of Haswell’s Transactional Memory Performance

Marcio M. Pereira

Introduction
Contribution
Experimental Evaluation
Conclusion

Working-set Size - Performance & outcomes

![Graphs showing speedup and abort fractions vs. working-set size]

<table>
<thead>
<tr>
<th>Zone size = 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper-bound</td>
</tr>
<tr>
<td>MaxRetry</td>
</tr>
<tr>
<td>Backoff</td>
</tr>
<tr>
<td>SerControl</td>
</tr>
</tbody>
</table>
Temporal Locality – Performance

Pollution – Performance
How does Contention affect performance in TSX?

![Graphs showing speedup over serial (8 threads) for different TX lengths and expected contention levels. The graphs compare different upper-bound, MaxRetry, Backoff, and SerControl scenarios.](image-url)
How does TSX perform with a more realistic benchmark?

Experiments with STAMP

- evaluating the behaviour of real applications across the input-size dimension
- understanding if real applications see the same effects seen in synthetic benchmarks
- unconventional use but very appropriate for this evaluation
Kmeans Low & Kmeans High – Analysis on Speedup

![Graph showing speedup over sequential for Kmeans Low & Kmeans High.](image)

Vacation & Yada – Analysis on Speedup

![Graph showing speedup over sequential for Vacation & Yada.](image)
Intruder – Analysis on Speedup & Outcomes

Ssca2 – Analysis on Speedup & Outcomes
Genome – Analysis on Speedup & Outcomes

Genome – Tuning on SerControl policy
Experiments with Haswell shows that

- TSX is simple and capable over a variety of workloads
- Performance can depend strongly on the software support systems
- Transaction footprint and working-set size constraints dictate the range of effective transactions
- Fallback policy can affect performance, especially when capacity-limited transactions are introduced into the system.