HADES: Hardware-Assisted Distributed Transactions in the Age of Fast Networks and SmartNICs

Apostolos Kokolis, Antonis Psistakis, Benjamin Reidys, Jian Huang and Josep Torrellas

University of Illinois at Urbana-Champaign

International Symposium on Computer Architecture (ISCA) 2024
June 2024
Introduction

- There is an unprecedented increase of user data
- Distributed systems must split data across nodes
- Cannot replicate everywhere
  - Too costly and inefficient

- Systems leverage advanced hardware
  - RDMA & RPC
  - RDMA has great improvements
    - Sub-μs latency
    - More than 200 Gb/s BW
    - Some support for atomic ops
    - Programmable Smart-NICs
Distributed Transactions

- ACID semantics
- Provide the illusion of atomicity to the users
- Serializable transactions simplify distributed applications
- The users require low-latency

**Challenge:** Increasing inefficiency of SW in Distributed Transactions

**Proposal:** Hardware support for efficient Distributed Transactions
Contribution: HADES

- Analyze the main sources of overhead in start-of-the-art distributed software transactional system
- Propose new hardware for distributed transactions to eliminate high-overhead SW operations
- Devise and evaluate HADES protocol: both local and remote transactions in HW
- HADES-H protocol: only remote transactions in HW
Software Protocols (Microsoft FaRM\cite{Dragojevic15})

1. \texttt{txn\_start}
2. \texttt{read}(A)
3. \texttt{read}(C)
4. \texttt{write}(B)
5. \texttt{write}(D)
6. \texttt{txn\_end}

\begin{itemize}
  \item Coordinator Node
  \item Records: A, B
  \item Remote Node-1
  \item Record: C
  \item Remote Node-2
  \item Record: D
\end{itemize}

\textbf{Execution} \hspace{2cm} \textbf{Validation} \hspace{2cm} \textbf{Commit}

\[\text{[1] No compromises: distributed transactions with consistency, availability, and performance, A. Dragojevic et al. SOSP'15}\]
Software Protocols (Microsoft FaRM[1])

- Software Distributed Transactions are a multi-step process
- Records need to be augmented with extra fields

<table>
<thead>
<tr>
<th>Lock</th>
<th>Incarnation</th>
<th>Seq. Num</th>
<th>V Line</th>
<th>V Line</th>
<th>...</th>
</tr>
</thead>
</table>

1. txn_start
2. read(A)
3. read(C)
4. write(B)
5. write(D)
6. txn_end

[1] No compromises: distributed transactions with consistency, availability, and performance, A. Dragojevic et al. SOSP’15
Inefficiencies of SW protocols

<table>
<thead>
<tr>
<th>Software Implementation overheads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing Read and Write Sets</td>
</tr>
<tr>
<td>Update record version for writes</td>
</tr>
<tr>
<td>Check read-atomicity on reads. No zero-copy.</td>
</tr>
<tr>
<td>Operate at record granularity</td>
</tr>
<tr>
<td>Additional RDMA requests</td>
</tr>
<tr>
<td>• Lock/unlock records and poll their completion</td>
</tr>
<tr>
<td>• Re-read record versions for conflict detection</td>
</tr>
</tbody>
</table>

- We evaluate software overheads to account for 59%-71% for SW distributed transactions
HADES Hardware Overview

Software Overheads
- Manage RD/WR sets
- Update record version for WR
- Check for read atomicity
- Operate at record granularity
- Lock/Unlock, re-read record versions for validation

Hardware Mechanism
- Bloom filters (BFs) next to LLC for local and in the NIC for remote accesses
- No record versions
- Use BFs to ensure read atomicity
- Operate at cache line granularity
- Support new messages that offload operations to the NIC. Use BFs to partially lock the directory on commit
HADES Hardware Details

2. Dir + Shared LLC

<table>
<thead>
<tr>
<th>WrTX_ID</th>
<th>Dir Tags</th>
</tr>
</thead>
</table>

1. Tags in Private Caches

<table>
<thead>
<tr>
<th>Recorded RD bit</th>
<th>Recorded WR bit</th>
<th>Addr Tag</th>
</tr>
</thead>
</table>

Diagram:
- Processor Node
- C_0
- C_n
- Private Caches
- Shared LLC
- Tags
- TX HW
- Dir Tags
- WrTX_ID
- Dir
- NIC
- PCIe
- MC
- DRAM
**HADES Hardware Details**

**3** HW for local TX

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Local Read BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Local Write BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**4a** For remote TXs

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Remote Read BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Remote Write BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**3** TX HW

**1** Tags

**2** Dir

**4a** NIC 4a, 4b

**1** Tags

**2** Dir

**3** TX HW

**4a** NIC 4a, 4b

**4b** For local TXs

**1** Tags

**2** Dir

**3** TX HW

**1** Tags

**2** Dir

**3** TX HW

**4a** NIC 4a, 4b

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Node ID</th>
<th>Start Addr</th>
<th>End Addr</th>
<th>Data Location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>List of remote node IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**HW in the NIC**

**Processor Node**

- C₀
- Cₙ
- Private Caches
- Shared LLC
- PCIe
- MC
- DRAM

For remote TXs

- TX_ID
- Node ID
- Start Addr
- End Addr
- Data Location

For local TXs

- TX_ID
- List of remote node IDs
HADES Protocol Operations

Remote Read
- Send request to remote node
- Add addresses to Remote Read BF
- Fetch data to local node
HADES Protocol Operations

Remote Write
- Send request to remote node
- Add the addresses of partially written lines to remote Write BF
- Fetch the partially written lines to the local node
- Buffer the updates of all addresses to the local node

HW in the NIC for local TXs

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Node ID</th>
<th>Start Addr</th>
<th>End Addr</th>
<th>Data Location</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>List of remote node IDs</th>
</tr>
</thead>
</table>

4a  HW in the NIC
For remote TXs

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Remote Read BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX_ID</td>
<td>Remote Write BloomFilter</td>
</tr>
</tbody>
</table>

4b  HW in the NIC for local TXs

Remote Write(B)
**Local Read**

- Check directory tag to check if another transaction wrote this line
- Add address to local Read BF

<table>
<thead>
<tr>
<th>Step</th>
<th>HW for local TX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Dir + Shared LLC</td>
</tr>
<tr>
<td></td>
<td>WrTX_ID</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Local Read BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Local Write BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HADES Protocol Operations

Local Write
- Check directory tag to check if another transaction wrote this line
- Check local Read BFs if another transaction has read this line
- Add address to local Write BF

Dir + Shared LLC

<table>
<thead>
<tr>
<th>WrTX_ID</th>
<th>Dir Tags</th>
</tr>
</thead>
</table>

HW for local TX

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Local Read BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TX_ID</th>
<th>Local Write BloomFilter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HADES Protocol Operations

1. Lock Dir with Rd+Wr BFs
2. Detect conflicts between local Wr and NIC BFs
3. Intention to commit and updates of remote writes
4. Lock Dir with Rd+Wr NIC BFs
5. Detect conflicts between WR addresses and LLC+NIC BFs
6. ACK
7. Validation
8. Perform updates, clear BFs, unlock Dir
8. Perform updates, clear BFs and unlock Dir
Partially Locking the Directory

- Mechanism to allow serialization of commits

```
Address

Bloom filter check

LLC Access

WR BFs

RD BFs

WrTX_ID

Dir/LLC Tags

LLC Data Array

Present in BF?

Y  N

Retry

Hit in LLC

Miss. Send to main memory
```
HADES-H Protocol

**Goal**: Minimize processor changes

- Local operations $\rightarrow$ SW transactions
- Remote operations $\rightarrow$ HADES approach

- HADES-H utilizes the same NIC HW and partial Dir locking
Evaluation - Throughput
Great improve in throughput: HADES (2.7x), HADES-H (2.3x)
Applications with many small read-write requests see higher gains
More in the paper

- **Design**
  - Details on HW structures
  - Bloom filter operations

- **Evaluation**
  - Methodology
  - More results
  - Bloom filter characterization
  - Sensitivity analysis to scale and network latency
Conclusion

- **HADES** utilizes hardware to alleviate the software overheads of distributed transactions

- **HADES**: achieves 2.7x higher throughput than state-of-the-art software-based design

- **HADES-H**
  - Mostly keeps the processor unmodified
  - 2.3x higher throughput than state-of-the-art software-based design
HADES: Hardware-Assisted Distributed Transactions in the Age of Fast Networks and SmartNICs

Apostolos Kokolis, Antonis Psistakis, Benjamin Reidys, Jian Huang and Josep Torrellas

University of Illinois at Urbana-Champaign

International Symposium on Computer Architecture (ISCA) 2024
June 2024