ERMIA: Fast Memory-Optimized Database System for Heterogeneous Workloads

Kangnyeon Kim  Tianzheng Wang
Ryan Johnson
Ippokratis Pandis
Heterogeneous is the new OLTP

Convenient/traditional

- Short write-intensive
- Short read-only

Emerging workloads

- Short write-intensive
- Longer read-mostly

Worst case for current systems, esp. CC
Modern OCC 101

• Decentralized optimistic concurrency control (OCC)
  • Silo [Tu et al. ’13]
  • FOEDUS [Kimura ’15]

• No locking for reads

• Locally-cached writes

• Commit phase
  1. Lock writes
  2. Verify reads
  3. Apply writes

Database

Transaction 1

Read set
A
B
Write set
D

Transaction 2

Read set
A
Write set
B'

Clobbered ➔ abort
OCC: Not a panacea

Aggregated commit ratio

- **ERMIA**
- **Silo-OCC**
- **Ideal**
- **ERMIA**
- **Silo-OCC**

**TPC-C**

**TPC-C + long read-mostly tx**

Legend:
- Blue: NewOrder
- Orange: Payment
- Gray: OrderStatus
- Yellow: Delivery
- Blue: StockLevel
- Green: Read-mostly

**Fair**

**Starved**
OCC + Read-mostly = Wasted cycles

TPC-C

TPC-C with long read-mostly

- Commit NewOrder
- Commit Payment
- Commit Delivery
- Commit OrderStatus
- Commit StockLevel
- Commit Read-mostly
- Abort NewOrder
- Abort Payment
- Abort Delivery
- Abort OrderStatus
- Abort StockLevel
- Abort Read-mostly

Read-mostly tx: fair and robust CC needed
ERMIA comes to the rescue

Logical layer: robust && fair CC

Serial Safety Net (SSN)
Serializable

Read-friendly, fair, robust, timely abort

Snapshot Isolation

Serial Safety Net (SSN)

In-memory data

Database == Scalable centralized log

Epoch-based Resource Management

Scalable physical layer
Scalable **Centralized** Redo Logging

- **Private log buffer**
- **The only global communication**
- **Async. filling**

Upon commit,

\[ LSN = \text{XADD}(\text{current LSN}, \text{log size}) \]

**XADD → Scalability + Global Ordering**
Latch-free Indirection Array

- Object IDs rather than pointers in leaf nodes
- No update propagation to secondary indexes

**Lock-free Index**

**OIDs in leaf nodes**

<table>
<thead>
<tr>
<th>OID</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Versions in durable log**

Fast recovery: fetch header only

CAS to install new version
The Serial Safety Net

The Serial Safety Net: Efficient Concurrency Control on Modern Hardware
T. Wang, R. Johnson, A. Fekete, I. Pandis, DaMoN ‘15
Visualizing SSN

Exclusion window satisfied

Exclusion window violation
Evaluation

• 24-core, quad-socket Xeon E7-4807
• Everything in main memory
• OCC vs. ERMIA-SI vs. ERMIA-SSN
• “Convenient”/traditional OLTP
  • Original TPC-C
  • Original TPC-E
• Heterogeneous OLTP
  • TPC-C with read-mostly transaction (TPC-C-hybrid)
  • TPC-E with read-mostly transaction (TPC-E-hybrid)
“Convenient”

Higher is better

Comparative performance to OCC
Robust heterogeneous performance

*Higher is better*

- **Silo-OCC**
- **ERMIAS-SI**
- **ERMIAS-SSN**

(Much) better overall throughput
Robust heterogeneous performance

**Higher is better**

Norm. throughput of TPC-CH-Q2*

Size of TPC-CH-Q2* transaction

**Lower is better**

Abort ratio of TPC-CH-Q2* (%)

Size of TPC-CH-Q2* transaction

(Much) lower abort rate for read-mostly tx
Conclusions

• **Heterogeneous OLTP**: fair && robust CC needed

- Read-friendly

• ERMIA = Snapshot Isolation + SSN + Lock/latch-free physical layer

- Fair, robust, serializable

- Scalable

• Also performs well for traditional OLTP

Find out more in our paper and code repo!

https://github.com/ermia-db

Thank you!