Database Management Systems

**OLTP**
(Online Transaction Processing)

- ATM
- Online Shopping
- Retail Sales
- Financial Transaction
Why Logging is Important – Durability

Checking Account: $0

Transaction 1
Deposit $100 to checking

Checking: $100

Transaction 2
Withdraw $100 from checking

Checking: ?
Why Logging is Important – Durability

Checking Account: $0

Transaction 1
Deposit $100 to checking

SYSTEM CRASH!

Transaction 2
Withdraw $100 from checking

Checking: $100

Checking: ?
Why Logging is Important – Durability

Checking Account: $0

Transaction 1
Deposit $100 to checking

SYSTEM CRASH!

Transaction 2
Withdraw $100 from checking

Checking: $100

Logging solves this problem!!!

Checking: ?
Why Logging is Important – Durability

Checking Account: $0

Transaction 1
Deposit $100 to checking

SYSTEM CRASH!

Transaction 2
Withdraw $100 from checking

Checking: $100

Checking: ?

Write to persistent storage

Logging solves this problem!!!
How to log?

- The algorithm must be more scalable and efficient than current algorithms
- Serial logging
- Batch logging
- Parallel logging
Transaction Dependency

- If a transaction is dependent on another transaction, they must be logged in order.
- The logging algorithm must account for these dependencies!
Serial Logging

- The easiest solution to the dependency problem: log transactions in order
- Each transaction acquires a unique Log Sequence Number (LSN) at commit time.
Optimizing Serial Logging

Standard, without optimization

With Optimization

Time spent waiting
Batch Logging

- With serial logging, each transaction needs an LSN from the Global LSN
- Quickly becomes bottleneck with large number of transactions
Batch logging provides one solution to this problem by having multiple loggers with multiple local LSNs instead of one global LSN. This removes the bottleneck.

**Serial Logging**

- Txn
  - LSN
- Txn
  - LSN+1
- Txn
  - LSN+2
- Txn
  - LSN+3
- Txn
  - LSN+4

**Global LSN**

**Batch Logging**

- Txn
  - LSN
- Local LSN 1
- Txn
  - LSN
- Local LSN 2
- Txn
  - LSN
- Local LSN 3
- Txn
  - LSN
- Local LSN 4
- Txn
  - LSN
- Local LSN 5
Batch Logging

- Assume dependencies between loggers
- Sync before returning to user
- Our implementation: Flush all loggers when one becomes full
- Drawback: high latency
Read(A)
Read(B)
Write(C)

Read(A)
Write(B)

Read(A)
Write(A)
Write(B)
Read(C)
Serial Logging

Txn 1
Read(A)
Read(B)
Write(C)

Txn 2
Read(A)
Write(B)

Txn 3
Write(A)
Write(B)
Read(C)

Batch Logging

Txn 1
Read(A)
Read(B)
Write(C)

Txn 2
Read(A)
Write(B)
Write(A)

Txn 3
Write(B)
Write(C)
Read(C)
Transaction Dependency

- No dependency
- RAW (Read After Write)
- WAW (Write After Write)
- WAR (Write After Read)

Txn 1
Read(A)
Logging

Txn 2
Read(A)
Write(B)
Logging

Independent Logging?
Transaction Dependency

- No dependency
- RAW (Read After Write)
- WAW (Write After Write)
- WAR (Write After Read)

Txn 1
Read(A)
Logging

Txn 2
Read(A)
Write(B)
Logging

Independent Logging? YES
Transaction Dependency

- No dependency
- RAW (Read After Write)
- WAW (Write After Write)
- WAR (Write After Read)

Independent Logging? YES
Transaction Dependency

- No dependency
- RAW (Read After Write)
- WAW (Write After Write)
- WAR (Write After Read)
Transaction Dependency

- No dependency
- RAW (Read After Write)
- **WAW (Write After Write)**
- WAR (Write After Read)

**Independent Logging?**
- YES
- No
Transaction Dependency

- No dependency
- RAW (Read After Write)
- **WAW (Write After Write)**
- WAR (Write After Read)

Independent Logging?
- YES
- No

---

<table>
<thead>
<tr>
<th>Txn 1</th>
<th>Txn 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write(A)</td>
<td>Write(A)</td>
</tr>
<tr>
<td></td>
<td>Write(B)</td>
</tr>
<tr>
<td>Logging</td>
<td>Logging</td>
</tr>
</tbody>
</table>

---

![Image of transactions and logging](image-url)
Transaction Dependency

- No dependency
- RAW (Read After Write)
- WAW (Write After Write)
- WAR (Write After Read)

Independent Logging?

YES
No
No
Transaction Dependency

- No dependency
- RAW (Read After Write)
- WAW (Write After Write)
- WAR (Write After Read)

Independent Logging?

YES
No
YES
Transactions

Txn 1
- Read(A)
- Read(B)
- Write(C)

Txn 2
- Read(A)
- Write(B)

Txn 3
- Write(A)
- Write(B)
- Read(C)

Parallel Logger

<table>
<thead>
<tr>
<th>LSN</th>
<th>Txn ID</th>
<th>Data Tuple</th>
<th>Dependency info</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Logger 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wait Buffer 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Logger 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wait Buffer 2</td>
</tr>
</tbody>
</table>
Transactions

<table>
<thead>
<tr>
<th></th>
<th>Txn 1</th>
<th>Txn 2</th>
<th>Txn 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>A</td>
<td>Read(A)</td>
<td>Read(A)</td>
</tr>
<tr>
<td>Read</td>
<td>B</td>
<td>Write(A)</td>
<td>Write(B)</td>
</tr>
<tr>
<td>Write</td>
<td>C</td>
<td>Write(B)</td>
<td>Read(C)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parallel Logger

<table>
<thead>
<tr>
<th>LSN</th>
<th>Txn ID</th>
<th>Data Tuple</th>
<th>Dependency info</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Logger 1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>C</td>
<td>[0, 0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wait Buffer 1</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>A</td>
<td>[1,1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Logger 2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>B</td>
<td>[0,0]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Wait Buffer 2</td>
</tr>
</tbody>
</table>
Transactions

<table>
<thead>
<tr>
<th>LSN</th>
<th>Txn ID</th>
<th>Data Tuple</th>
<th>Dependency info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>C</td>
<td>[0, 0]</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>A</td>
<td>[1,1]</td>
</tr>
</tbody>
</table>

Parallel Logger

<table>
<thead>
<tr>
<th>LSN</th>
<th>Txn ID</th>
<th>Data Tuple</th>
<th>Dependency info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>B</td>
<td>[0, 0]</td>
</tr>
</tbody>
</table>

Logger 1

Wait Buffer 1

Logger 2

Wait Buffer 2
Scalability and Results

Log Times for Serial Logging

Average Log Time per Txn (s)

Buffer Size

4 threads, no optimization
4 threads, optimization
8 threads, no optimization
8 threads, optimization
16 threads, no optimization
16 threads, optimization
Batch Logging Results
Conclusion and Future Work

- **Accomplishments**
  - Implemented serial, batch, and parallel logging
  - Determined areas of improvement
  - Tested scalability and efficiency

- **Future Goals**
  - Gather results for parallel logging
  - Other optimizations for serial logging
  - Log recovery
  - Publish paper
Special thanks to...

- Our mentor, Xiangyao Yu
- Prof. Srini Devadas for his help and guidance
- The PRIMES program
Thank you!