Eiger: Stronger Semantics for Low Latency Geo-Replicated Storage

Wyatt Lloyd (Princeton → Facebook → USC)
Michael J. Freedman (Princeton)
Michael Kaminsky (Intel Labs)
David G. Andersen (CMU)

http://www.istc-cc.cmu.edu/
is the backend of massive websites

“Halting is Undecidable”
Storage Dimensions

Shard Data Across Many Nodes

Data Geo-Replicated In Multiple Datacenters
Sharded, Geo-Replicated Storage
Low Latency
- Improves user experience
- Correlates with revenue

Fundamentally in Conflict
[LiptonSandberg88, AttiyaWelch94]

Strong Consistency
- Obey user expectations
- Easier for programmers
Strong Consistency or Low Latency

Megastore [SIGMOD '08]
Spanner [OSDI '12] →
Gemini [OSDI '12] →
Walter [SOSP '11] →
Lynx [SOSP '13] →

Dynamo [SOSP '07]

← COPS [SOSP '11]

 ← Eiger [NSDI '13]
Obey user expectations
Easier for programmers

Causal+ Consistency
Rich Data Model
Read-only Txns
Write-only Txns
Eiger Ensures Low Latency

Keep All Ops Local
Causal+ Consistency Across DCs

• If $A$ happens before $B$
  ▫ Everyone sees $A$ before $B$

• Obeys user expectations

• Simplifies programming
Causal For Column Families

- Operations update/read many columns
- Range query columns concurrent w/ deletes
- Counter columns
- See paper for details
Viewing Data Consistently Is Hard

Asynchronous requests + distributed data = ??????
Read-Only Transactions

- Logical time gives a global view of data store
  - Clocks on all nodes, carried with all messages

- Insight: Store is consistent at all logical times
Read-Only Transactions

- Extract consistent up-to-date view of data
  - Across many servers

- Challenges
  - Scalability
    - Decentralized algorithm
  - Guaranteed low latency
    - At most 2 parallel rounds of local reads
    - No locks, no blocking
  - High performance
    - Normal case: 1 round of reads
Read-Only Transactions

- Round 1: Optimistic parallel reads
- Calculate *effective time*
- Round 2: Parallel read_at_times

**Client**

```
0  A_1
1  A_2
3  B_2
4  C_2
5  
6  
```

**Distributed Storage**

```
A
0  A_1
2  A_2
B
0  B_1
3  B_2
C
0  C_1
5  C_2
```

Logical Time
Transaction Intuition

- **Read-only transactions**
  - Read from a single logical time

- **Write-only transactions**
  - Appear at a single logical time

**Bonus:** Works for Linearizability
Eiger Provides

✓ Low latency
✓ Rich data model
✓ Causal+ consistency
✓ Read-only transactions
✓ Write-only transactions

But what does all this cost?
Does it scale?
• Fork of open-source Cassandra

• +5K lines of Java to Cassandra’s 75K

• Code Available:
  ▫ https://github.com/wlloyd/eiger
Evaluation

- **Cost of stronger consistency & semantics**
  - Vs. eventually-consistent Cassandra
  - Overhead for real (Facebook) workload
  - Overhead for state-space of workloads

- **Scalability**
Local Datacenter (Stanford)

Replication

Remote DC (UW)
Facebook Workload Results

6.6% Overhead
Facebook Workload

Normalized Throughput (log)

Servers/Cluster (log)

Scales out

NSF Probe Kodiak Testbed ➔ 384 Machines!
<table>
<thead>
<tr>
<th></th>
<th><strong>COPS</strong></th>
<th><strong>Eiger</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data model</td>
<td>Key-Value</td>
<td>Column-Family</td>
</tr>
<tr>
<td>Read-only Txns</td>
<td>Causal stores</td>
<td>All stores</td>
</tr>
<tr>
<td>Write-only Txns</td>
<td>None</td>
<td>Yes</td>
</tr>
<tr>
<td>Performance</td>
<td>Good</td>
<td>Great</td>
</tr>
<tr>
<td>DC Failure</td>
<td>Throughput degradation</td>
<td>Resilient</td>
</tr>
</tbody>
</table>
I’m headed to USC

• Please ask your great students to apply for PhD

• **Ramesh Govindan**
  ▫ Cloud-enabled mobile computing

• **Ethan Katz-Basset**
  ▫ Internet measurement, routing, perf, reliability

• **Minlan Yu**
  ▫ Datacenter networking, software defined networking

• **Wyatt Lloyd**
  ▫ Distributed systems
Eiger

- Low-latency geo-replicated storage
  - Causal+ for column families
  - Read-only transactions
  - Write-only transactions

- Demonstrated in working system
  - Competitive with eventual
  - Scales to large clusters
  - https://github.com/wlloyd/eiger