ResilientDB: Global Scale Resilient Blockchain Fabric

Suyash Gupta  
Sajjad Rahnama  
Jelle Hellings  
Mohammad Sadoghi

Exploratory Systems Lab  
University of California Davis

*Proceedings of the VLDB Endowment, Vol. 13, No. 6.*
Types of Blockchain Systems

- **Permissionless** → **Open Access**
  - Anyone can participate.
  - Identities of the replicas unknown.
  - Face blockchain *forks*.

- **Permissioned (Our focus)** → **Restricted Access**
  - Only a selected group of replicas, although untrusted can participate.
  - Identities of the replica known a priori.
At the core of any Blockchain application is a Byzantine Fault-Tolerant (BFT) consensus protocol.
## Challenges For Geo-Scale Blockchains

<table>
<thead>
<tr>
<th></th>
<th>Ping round-trip times (ms)</th>
<th>Bandwidth (Mbit/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>O</td>
<td>I</td>
</tr>
<tr>
<td>Oregon (O)</td>
<td>≤ 1</td>
<td>38</td>
</tr>
<tr>
<td>Iowa (I)</td>
<td>≤ 1</td>
<td>33</td>
</tr>
<tr>
<td>Montreal (M)</td>
<td>≤ 1</td>
<td>82</td>
</tr>
<tr>
<td>Belgium (B)</td>
<td>≤ 1</td>
<td>252</td>
</tr>
<tr>
<td>Taiwan (T)</td>
<td>≤ 1</td>
<td>137</td>
</tr>
<tr>
<td>Sydney (S)</td>
<td>≤ 1</td>
<td></td>
</tr>
</tbody>
</table>

Real-world inter- and intra-cluster communication costs in terms of the ping round-trip times (which determines latency) and bandwidth (which determines throughput). Measurements taken on Google Cloud using clusters of n1 machines (replicas) that are deployed in six different regions.
Limitations of Existing Consensus Protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Decisions</th>
<th>Communication (Local)</th>
<th>Communication (Global)</th>
<th>Centralized</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoBFT (our paper)</td>
<td>z</td>
<td>$O(2zn^2)$</td>
<td>$O(fz^2)$</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>↓ single decision</td>
<td>$O(4n^2)$</td>
<td>$O(fz)$</td>
<td>No</td>
</tr>
<tr>
<td>Steward</td>
<td>1</td>
<td>$O(2zn^2)$</td>
<td>$O(z^2)$</td>
<td>Yes</td>
</tr>
<tr>
<td>Zyzzyva</td>
<td>1</td>
<td>$O(zn)$</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>PBFT</td>
<td>1</td>
<td>$O(2(zn)^2)$</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>PoE</td>
<td>1</td>
<td>$O((zn)^2)$</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>HotStuff</td>
<td>1</td>
<td>$O(8(zn))$</td>
<td></td>
<td>Partly</td>
</tr>
</tbody>
</table>

The normal-case metrics of BFT consensus protocols in a system with z clusters, each with n replicas of which at most f, n > 3f, are Byzantine. GeoBFT provides the lowest global communication cost per consensus decision and operates decentralized.
Vision Geo-Scale Byzantine Fault-Tolerance
GeoBFT Protocol

GeoBFT is a topology-aware protocol, which groups replicas into clusters. Each cluster runs the PBFT consensus protocol, in parallel and independently.

**Local Replication**
Each cluster runs PBFT to select, locally replicate, and certify a client request.

**Inter-cluster Sharing**
Primary at each cluster shares the certified client request with other clusters.

**Ordering and Execution**
Order the certified requests, execute them, and inform local clients.
Local Replication (PBFT)

• First practical Byzantine Fault Tolerant Protocol.

• Tolerates up to $f$ failure out of $3f+1$ replicas

• Three phases of which two require quadratic communication complexity.

• Safety is always guaranteed and Liveness is guaranteed in periods of partial synchrony.

• View-Change protocol for replacing malicious primary
PBFT Civil Execution

Construct certificates that include T and n-f Commit messages.
The Primary $P_{C1}$ sends a certificate that includes the client request and commit messages from $n-f$ replicas of Cluster $C_1$. 
Ordering and Execution

- GeoBFT orders requests deterministically.
- For $i < j$, requests of Cluster $C_i$ are executed before requests of cluster $C_j$.
- For example: requests of $C_1$ are executed before $C_2$. 
ResilientDB associates a multi-threaded deep-pipelined architecture with each replica.

ResilientDB is open-sourced at https://resilientdb.com/
• In ResilientDB, $i^{th}$ block in the ledger contains the $i^{th}$ executed request.

• In each round of GeoBFT, each replica executes $z$ requests, each belonging to a different cluster $C_i$, $1 \leq i \leq z$.

• Hence, in each round, each replica creates $z$ blocks.

• To ensure immutability, each block includes both client requests and exchanged certificates.
Evaluation on ResilientDB

- Google cloud used for deploying replicas and clients.
- Each replica used 8-core Intel Skylake CPUs and had access to 16 GB memory.
- Total 160K clients deployed on eight 4-core machines.
- Workload provided by Yahoo Cloud Serving Benchmark (YCSB).
- Replicas deployed across six different regions: Oregon, Iowa, Montreal, Belgium, Taiwan and Sydney.
- Primaries for centralized protocol placed at Oregon (highest bandwidth).
Impact of Geo-Scale Deployments

Throughput as a function of the number of clusters; zn = 60 replicas.
Throughput as a function of the batch size; $z = 4$ and $n = 7$. 
Conclusions and Final Remarks

• For achieving faster local replication, other efficient BFT protocols, such as PoE, can be employed.

• Modern cryptographic techniques such as Threshold signatures can be used in place of sending n-f Commit messages.

• If a cluster does not have a request, it can send “no-op” messages.

• GeoBFT optimizes consensus by reducing global communication costs.

• Parallel local replication helps to increase system throughput.

• GeoBFT is a topology-aware protocol.


