Practical Byzantine Fault Tolerance

Miguel Castro & Barbara Liskov, MIT, 1999

Presented by Yixi Zhang
Related Work

- Byzantine Agreement Problem
- Earlier solutions
Contribution

• Algorithm
• Optimizations
• Implementations
• Evaluations
Model
Model

- Asynchronous
Model

- Asynchronous
- Deterministic state machines
Model

- Asynchronous
- Deterministic state machines
- $3k+1$ for at most $k$ faulty replicas
Concepts
Concepts

• View
Concepts

- View
- Primary
Concepts

- View
- Primary
- Backups
Algorithm

• client sends request to primary
• primary multicasts it to backups
• replicas exec the request and send replies to client
• client accepts result from at least f+1 same replies
Figure 1: Normal Case Operation
Detail

- Messages logged
Detail

- Messages logged
- Execution order
• Messages logged
• Execution order
• Log garbage collection
Detail

- Messages logged
- Execution order
- Log garbage collection
- Checkpoints
Optimization
Optimization

- only one replica sends result, others send digest
Optimization

- only one replica sends result, others send digest
- tentative execution and reply
Optimization

• only one replica sends result, others send digest
• tentative execution and reply
• read-only operations
Optimization

• only one replica sends result, others send digest
• tentative execution and reply
• read-only operations
• MACs replace digital signatures
Replication Lib
Replication Lib

- Client
Replication Lib

- Client
- Server
Replication Lib

- Client
- Server
- Communication
Figure 2: Replicated File System Architecture.
Checkpoints
Checkpoints

- copy-on-write
Checkpoints

- copy-on-write
- incremental state digest
Table 1: Micro-benchmark results (in milliseconds); the percentage overhead is relative to the unreplicated case.
## Evaluation

<table>
<thead>
<tr>
<th>phase</th>
<th>BFS</th>
<th>r/o lookup</th>
<th>BFS-nr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>strict</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.55 (57%)</td>
<td>0.47 (34%)</td>
<td>0.35</td>
</tr>
<tr>
<td>2</td>
<td>9.24 (82%)</td>
<td>7.91 (56%)</td>
<td>5.08</td>
</tr>
<tr>
<td>3</td>
<td>7.24 (18%)</td>
<td>6.45 (6%)</td>
<td>6.11</td>
</tr>
<tr>
<td>4</td>
<td>8.77 (18%)</td>
<td>7.87 (6%)</td>
<td>7.41</td>
</tr>
<tr>
<td>5</td>
<td>38.68 (20%)</td>
<td>38.38 (19%)</td>
<td>32.12</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>64.48 (26%)</strong></td>
<td><strong>61.07 (20%)</strong></td>
<td><strong>51.07</strong></td>
</tr>
</tbody>
</table>

Table 2: Andrew benchmark: BFS vs BFS-nr. The times are in seconds.
## Evaluation

<table>
<thead>
<tr>
<th>phase</th>
<th>BFS</th>
<th></th>
<th>NFS-std</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>strict</td>
<td>r/o lookup</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.55 (-69%)</td>
<td>0.47 (-73%)</td>
<td>1.75</td>
</tr>
<tr>
<td>2</td>
<td>9.24 (-2%)</td>
<td>7.91 (-16%)</td>
<td>9.46</td>
</tr>
<tr>
<td>3</td>
<td>7.24 (35%)</td>
<td>6.45 (20%)</td>
<td>5.36</td>
</tr>
<tr>
<td>4</td>
<td>8.77 (32%)</td>
<td>7.87 (19%)</td>
<td>6.60</td>
</tr>
<tr>
<td>5</td>
<td>38.68 (-2%)</td>
<td>38.38 (-2%)</td>
<td>39.35</td>
</tr>
<tr>
<td>total</td>
<td>64.48 (3%)</td>
<td>61.07 (-2%)</td>
<td>62.52</td>
</tr>
</tbody>
</table>

Table 3: Andrew benchmark: BFS vs NFS-std. The times are in seconds.
Future

- Reducing number of replicas
- Reducing the number of copies of state to $f+1$
Questions?