How Fast can a Distributed Transaction Commit?

Atomic Commit
- At the heart of distributed transaction commit
- Ensure consistency despite concurrency and failures
- Each node votes "yes" (commit) or "no" (abort)
- Every node eventually decides despite failures (Termination; T)
- Agreement (A)
- Validity (V)

A Classical and Relevant Problem
Atomic commit in transactional systems:
- Date back to 70's [1]
- Of today: HP's Sinfonia, Yahoo's PNUTS, Google's Percolator and Spanner, etc.

Popular protocol:
- Two-phase commit (2PC)
- Efficient despite no guarantee on termination

Complexity of atomic commit:
- Open for decades
- Except for some results on synchronous commit

New Complexity Results
Tradeoff between robustness and best-case complexity (complexity of nice/failure-free (FF) executions)

<table>
<thead>
<tr>
<th></th>
<th>Less robust</th>
<th>More robust</th>
</tr>
</thead>
<tbody>
<tr>
<td>All properties in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>every execution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Less robust</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All properties only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>in every NICE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>execution</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Impossibility [2] = infinite complexity
- Robustness (X, Y): all properties in FF; X in CF (where nodes may crash); Y in NF (CF where messages may arrive late)
- Fraction d/m: d message delays and m messages are tight lower bounds
- #msgs: a₁ = n-1+f, a₂ = 2n-2+f, a₃ = 2n-2
- NB: No tradeoff between time and messages for 9 cases (1/0); tradeoffs for the rest

New Protocol
INBAC:
- Despite failures, satisfies AVT in NF (Indulgent atomic commit; I)
- Tolerates violations of timeouts
- Optimal in best-case complexity
- ≃ 2PC in efficiency (when f = 1)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#delays</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>#msgs</td>
<td>2fn</td>
<td>2n-2</td>
<td>fn+2n-2</td>
<td>2fn+2n-2f-2</td>
</tr>
<tr>
<td>Atomic commit</td>
<td>I</td>
<td>Blocking</td>
<td>I</td>
<td>I</td>
</tr>
</tbody>
</table>

- f = #crashes; n = #nodes

Main idea of INBAC:
- Best effort in nice executions
- Consensus to the rescue

References