BFT with Additional Components

Administrivia
- Logistics
  - Poster session next week
  - Report due May 7
  - Exam May 8

See posting on Campuswire

Where we are
Impossibility results

Async 3-FLP

But solvable in some practical scenarios

All

3 Unauth BFT $n \leq 3t$ nodes

New model assumptions

- Partial synchrony
- Authentication
- Failure Detectors

Reality

- Fail-stop/fail-recover protocols are widely used

$\Rightarrow$ Likely interacted with one in the last few hours

- Modern networks engineered & provisioned so that partial synchrony holds

$\Rightarrow$ Warning: Not a formal statement

$\Rightarrow$ But things “mostly just work”

- BFT deployments are rare

$\Rightarrow$ Concern: Cost ($$$)/Performance vs Utility
Where deployed-trade-offs make these concerns moot

- Specialized areas
  - Space
  - Critical Infra
- Cryptocurrency, etc.

Common (practical) question

Can we add something to our networks or machines to change tradeoffs?

Ideally

\[
\text{Cost of something (think like FD)} + \text{Cost of BFT today} < \text{Cost of BFT w/ something}
\]

Today's papers

- Proposal for something
- Analysis of whether, why & how

Notes

1. The precise elements are somewhat influenced by history.
TPMs (~2009): Platform integrity checks
   \[ \rightarrow \text{H/W RNG} \]
   \[ \rightarrow \text{Attest to hardware \\& software} \]
   \[ \rightarrow \text{Binding} \]
   \[ \rightarrow \ldots \]

Enclaves (~2013/14): Protect state from adv. or physical access
   \[ \rightarrow \text{Encrypt data on the memory \\
   \hspace{1cm} \text{bus}} \]

Both of these are reasonably deployed
   \[ \rightarrow \text{Nearly everyone has a TPM module} \]
   \[ \rightarrow \text{Requirement to boot some} \]
   \[ \hspace{1cm} \text{versions of Windows \\& \text{OSX}} \]
   \[ \rightarrow \text{Most reasonably new phones have} \]
   \[ \hspace{1cm} \text{enclaves} \]

Deployment concerns are not the main impediment
to adoption.

\(\text{2) The analysis (both in Power of Non-Equiv \& Dissecting BFT)\)}
\(\text{while a little less formal than previous}\)
\(\text{papers take a similar form to before, so} \)
\(\text{should be something you can follow}\)
From two classes ago:

How can Byzantine failures manifest:

- Crashes
- Sending malformed messages
- Internal state corruption
- Equivocation
- Omission
- ...

Avoiding Equivocation

- One idea: binding (InInc, ...)

What does this mean:
Core concern or question: Power of Non-Equivocation is trying to raise

When possible?

Non-equivocation ≠ Authentication

Result: Even with non-equivocation, BFT requires ≥ 3f+1 nodes for agreement.

Termination

Validity → If proposer is honest, deliver proposal.
Q1. What is the prog assuming
   synchrony?
   partial synchrony?
   asynchrony?

Q2. How to fix?
Non-equivocation + Authentication is better?

Basic idea: - No sender can equivocate

  - Safe to use relayed messages

  [If process A relays a message for process B by forwarding it to C, C can check relay correctness]

  \[\Rightarrow\] Broadcast received messages to avoid omission

PBFT-EA [A2M]

One idea: \(2f + 1\) is enough

Why? Limited how faulty nodes
What doesn’t change

- Clients still wait for f+1 responses.
  \( \rightarrow \) Why?

- Clients still broadcast command after delay
  \( \rightarrow \) Why?

What does change

- View change
  \( \rightarrow \) Can only require f+1 requests to initiate

- Must ensure that a node can participate in view \( \nu \) after suggesting moving to \( \nu+1 \).
  Why?

- Details: see A2M paper (if interested, not required)
Can set up scenarios where client gets stuck

- A faulty — Runs protocol with only B

  - Client gets response from B

    ▶ Not sufficient to make progress

- Client cannot trigger view change

  ▶ B won't trigger V.C., saw client message

  ▶ A in faulty

  ▶ C alone insufficient

Also, mismatch b/w what endlaves provide & \( T_{INC/A2I} \)

Not a sign that any paper got it wrong, just a reflection on what got deployed
So big claim

- T5iNC /A2M + PBFT-EA w/2f+1 nodes
  \[ \rightarrow \text{Not live} \]

- Are they still useful in some way?
  \[ \rightarrow \text{Reduce number of rounds} \]

Aside: Non-Equivalence translation from CF7 → BPF

\[
\text{Client} \quad A \\
\text{B} \\
\text{C} \\
\text{D}
\]

\[
\begin{align*}
\text{Prepare}\, & \quad \text{prepare} \, y, \langle x, \ldots, z \rangle \quad \text{Note, binding to avoid equivocation from primary} \\
\text{Reply} \quad & \quad \text{NOT binding}
\end{align*}
\]

\[ \text{Relays command, preventing omission!} \]
Basic idea—① Nodes send history w/each message
  ② On receiving history, simulate & check that message would be sent, etc.
  ③ Deliver.

Trickier to do than it seems.

Observations at the End.