Distributed Computing

- Setting: synchronous, asynchronous, partially sync.
- Communication: message passing, shared memory
- Issues:
  - Atomic Commit
  - Leader election
  - Mutual Exclusion
  - TRB (Validity, Agreement, Integrity)
  - Consensus (V, A, I, Termination)
- Faults: process
  - connect
  - faulty
    - benign byzantine
    - crash
      - synch. a.m. arb. arb. w/ signature
    - gen. crash
      - receive
      - gen. omission
      - arb. w/ signature
      - arb. arb. arb.

- Parallel Comp = Classical par + Comp U Dist Comp
  - concurrency
  - multithreaded
  - communics
  - failures, asynch
Crashes

Arbitrary w/ message falsification
Sometimes arbitrariness
Arbitrary = Byzantine

e.g. Byzantine crash
4. Byzantine crash
Self-misled
Efficient card

Receive-omission
e.g. input buffer
beings

mip
if P is not
then iloves
signed by P

correct guy cannot sign for bad guy
by def
Relaxing FCP

PS
CS
MO
MBS - multiple broadcast / step
AS+R

(0, x, 1, 0, 0, x, 0, 0)
(0, 0, 1, 0, 1, 0, 0, 0)
Global Coin Proof

D e.g.

Round 1: $P_1 \ P_2 \ P_3$

if $C_1 = 0$, (coin toss was unlucky for agreement)

then strong adversary can prevent $P_2$ or $P_3$ to move to round 2 (say $P_2$)

Round 2: $P_1 \ P_3$

$C_2 = X$; learned when $P_1$ tosses

Round 1: $P_2$

$1 \ X$

strong adv lets $P_2$ move to round 2

wlog $X = 1$ then

$P_1 \ P_2 \ P_3$

$1 \ 0 \ 0$

and $C_2 = 1$ ⇒ repeat forever

$\Rightarrow \ \varepsilon = 0 \Rightarrow \Omega(\frac{1}{\varepsilon}) = \Omega(\infty)$
Unreliable Failure Detectors

\[ \text{FLP} \rightarrow \text{transition in revocable change bivalence} \rightarrow \text{univalence depends on weight in processes.} \]

\[ \text{perfect failure detector: visible} \]

\[ \text{a perfect oracle: complete, accuracy: } \forall x \in \text{List}, x \text{ is head} \]

Weak FD: OW

\[ \exists ! 3 \text{ connects, from } 1 \text{ onwards, oracle (connect-p) will return } \]

\[ \exists ! 3 \rightarrow \forall p, \text{ no } p \text{ suspects connect-p} \]

\[ \rightarrow \text{can solve FLP!} \]

Leader Election (Weak FD)

\[ \exists ! p, \text{ cor-p, Vp, p bears cor-p} \]
\[ \Diamond w \leftrightarrow \Omega \]

- FLP \rightarrow \text{partial synchronous}
- Ben Or (doesn't use 89\% of fine things so well)
- \[ q(t) \rightarrow \text{mutual excl. w message passing} \]
- Homework: show \( \Diamond w \leq \Omega \)
- scheduler = partially synch \( \leftrightarrow \) WeakFO
- EW/relative speed
- perfect failure detect \( \leftrightarrow \) synch sys
- research quest: \( \Sigma \)-class of FO beyond \( \Omega \)-complete
- k-set agreement
- skip impl. 3rd alg for Chap II
Failure Detections

- Distributed oracle
  - weakest
  - weak FD → Ø & W

Chanden Toug classes

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Strong: Vp ∈ connect, Vp ∈ E; p ≠ oracle; dead</th>
<th>Weak: Jp ∈ E, Vp ∈ E; p ≠ oracle; dead</th>
<th>ev. strong: Jp ∈ oracle is strong</th>
<th>ev. weak: Jp ∈ oracle is weak</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>Q</td>
<td>S</td>
<td>W</td>
</tr>
</tbody>
</table>

Distributed Reduction

Let D and D' be FDs. D ≤ D' (⇒ ∃ T₀ → D')

"D is red to D" (⇒ "D' is weaker than D"

Then P ≤ Q, S ≤ W, ∆P ≤ ∆Q

Cor Q ≤ P, etc. ∴ P ≤ Q

⇒ Consensus solvable in AS with f < ⌈n/2⌉, ∀D, D ≤ W

⇒ Consensus not solvable in AS with f ≥ ⌈n/2⌉, ∀D, ∆P ≤ O
QoS $p$ monitors $q$

Way $\rightarrow$ proof of loss dist $\rightarrow$

QoS $k$

requirements: $D = 10\text{ms}$

$p_d = .999$

ExFederal

$k$-set agreement: generalized consensus $k = 1$

partial sync: GST