How to build a Failure Detectors

Etienne Renault

2 octobre 2020

https://www.lrde.epita.fr/~renault/teaching/algorep/

How to build a failure detector ?

Hire 1000 people, each to monitor one machine in the datacenter and report to you when it fails

Write a failure detector program (distributed) that automatically detects failures and reports to your workstation.

Desirable Properties

- **Completeness** : each failure is detected
- **Accuracy** : there is no mistaken detection
- **Speed** : Time to first detection of a failure
- **Scale** :
  - Equal Load on each member
  - Network Message Load

Desirable Properties

- **Completeness** : each failure is detected
- **Accuracy** : there is no mistaken detection
- **Speed** : Time to first detection of a failure
- **Scale** :
  - Equal Load on each member
  - Network Message Load

Impossibility

Completeness and Accuracy are impossible together in lossy networks (see Chandra and Toueg)

What Real Failure Detectors Prefer ?

Guarantee & Accuracy

- Guarantee about completeness
- Partial guaranty about accuracy

Scale & Speed

- Scale : no bottleneck
- Speed : time until **some** process detects the failure

Centralized Heartbeating

- A processus heartbeats periodically each other processus
- If heartbeat not received from a process within timeout, mark this process as failed

Hotspot

The "hearbeating" process is an hotspot !

Ring Heartbeating

- A processus heartbeats periodically its neighbours on a bidirectionnal ring
- If heartbeat not received from a process within timeout, mark this process as failed

Problem

Unpredictable on simultaneous multiple failures

Equal load per member

All-to-all Heartbeating

- A processus heartbeats periodically all its neighbours
- If heartbeat not received from a process within timeout, mark this process as failed

Equal load per member

Loss of an heatbeat $\Rightarrow$ False Detection

Next

How do we increase the robustness of all-to-all heartbeating ?

Gossiping Heartbeating

- A processus heartbeats periodically subset of its neighbours
- **When heartbeating send an array of heartbeats (subset)**
- If heartbeat not received from a process within timeout, mark this process as failed

Good accuracy properties

Gossiping Heartbeating : details

- Nodes periodically gossip their membership list : pick random nodes, send it list
- On receipt, it is merged with local membership list
- When an entry times out, member is marked as failed

Processes maintain :

- **Address** : the destination node
- **Heartbeat Counter** : number of received answers
- **time** : local

Gossip-Style Failure Detection

If the heartbeat has not increased for more than Tfail seconds, the member is considered failed

And after a further Tcleanup seconds, it will delete the member from the list

Additionnal Timeout

Why an additional timeout ? Why not delete right away ?

Example

Conclusion

Heartbeating is a fundamental of Failure Detection

Other approaches

SWIM Failure Detector