Failure Detectors and QoS of Failure Detectors

Narasimha Raghavan

INF5360 - Seminar on dependable and adaptive distributed systems
University of Oslo
February 22, 2012
What are Failure Detector (FD)?

- Responsible for discovering failures in distributed systems
- A fundamental block in building fault-tolerant systems
- Provides information on which processes are crashed

Question?

Why FD’s are not provided as services similar to DNS/NTP?
What is the QoS of FD?

A specification that quantifies:

1. **Speed**: how fast an FD detects a crash
2. **Accuracy**: how well an FD avoids the wrong suspicions
QoS metrics of FD

Overview of QoS metrics:

- Detection Time $T_D$
- Query Accuracy probability $P_A$ (similar to availability measure)
- Average Mistake Rate $\lambda_m$ (similar to failure rate)
- Mistake Duration $T_M$ (similar to time to recover)
- Good Period Duration $T_G$ (similar to high reliability measure)
- Mistake Recurrence Time $T_{MR}$ (similar to time between failures)
Detection Time $T_D$

Measures how fast a failure detector detects a process crash.

FD

trust

suspect

Process p

up

down

$T_D$
Query Accuracy Probability $P_A$

$P_A$ is the probability that FD provides correct output at random time.
Why $P_A$ is not sufficient to fully describe the accuracy of FD?

FD1 and FD2 have same QPA values, however they have different...
Average Mistake Rate $\lambda_M$

$\lambda_M$ measures the rate at which a failure detector makes mistakes.

- FD1
- FD2
- Process p

$p$ is not sufficient.
$P_A$ and $\lambda_M$ are not sufficient: why?

FD1 is better than FD2 in terms of $P_A$ and $\lambda_M$ but FD2 can quickly correct its mistakes.
More Accuracy Metrics

- Mistake Duration $T_M$
- Good Period Duration $T_G$
- Mistake Recurrence Time $T_{MR}$

FD

$T_M$ $T_G$ $T_{MR}$
A Simple FD Algorithm

- $q$ monitors $p$
- $p$ sends heart beat messages to $q$ at regular intervals $\eta$
- $q$ suspects $p$ if a heart beat message is not reached to $q$ within time out period ($TO$)
- A $TO$ is a constant added to the $\eta$

Problem

- Choosing a smaller value for $TO$ leads to high number of wrong suspicions
- Choosing a larger value for $TO$ leads to high detection time
A Better FD Algorithm

When clocks are synchronized,

- Set $TO$ based on the transmission time of the heart beat messages.
- Advantage: Maximal detection time is bounded

When clocks are not synchronized,

- Estimate the arrival time of the next heart beat using the past samples
- A constant is added as a safety margin to the estimation
Accrual Failure Detector (AFD)

Basic idea

- Traditional FD provides a boolean output regarding the status of the monitored processes
- AFD provides a continuous scale output regarding the status of the monitored processes

FDs are defined from architectural perspective: three basic parts

- Monitor: gathers information heart beat intervals from others processes/networks
- Interpretation: Monitoring information used decide that a process should be suspected
- Action: Actions are executed as a response to triggered suspicions.
Accrual Failure Detector (AFD) ctd.,

On the QoS Specifications of FD ctd.,

Fig. 2. Structure of traditional failure detectors. Monitoring and interpretation are combined. Interactions with applications and protocols is boolean.

Fig. 3. Structure of accrual failure detectors. Monitoring and interpretation are decoupled. Applications interpret a common value based on their own interpretation.

On the other hand, if $\Delta_i$ is a lot larger than $\Delta_{tr}$, then $\Delta_i$ will almost entirely determine the detection time. Increasing it further will increase the detection time accordingly, but will have a near-linear effect in reducing the already low load on the network. Hence, we can conclude that any reasonable value of $\Delta_i$ should be roughly equal to the average transmission time $\Delta_{tr}$. The only exception is when upper limits are set on the acceptable usage of network bandwidth for control messages.

Although the above argument is rather informal, it suggests that there exists, with every network, some nominal range for the parameter $\Delta_i$ with little or no impact on the accuracy of the failure detection. In other words, we can consider that the parameter $\Delta_i$ is given by the underlying system rather than computed from application requirements.

III. ACCRUAL FAILURE DETECTORS

The principle of accrual failure detectors is simple. Instead of outputting information of a boolean nature, accrual failure detectors output suspicion information on a continuous scale. Roughly speaking, the higher the value, the higher the chance that the monitored process has crashed.

In this section, we first describe the use of accrual failure detectors from an architectural perspective, and put this in contrast with conventional failure detectors. Then, we give a more precise definition of accrual failure detectors. Finally, we conclude the section by showing the relation between accrual failure detectors and conventional ones. In particular, we show how an accrual failure detector can be used to implement a failure detector of class $\square \mathcal{P}$.
The suspicion level about a process is expressed by the value of $\phi$.

If the process is crashed, the value of $\phi$ accrues over time and tend toward infinity.

Each application interprets $\phi$ based on their QoS requirements.

Application have a threshold value and it compares its threshold with $\phi$. Based on the QoS requirements and comparison results, it can take its actions.