TDDD36

Secure Mobile Systems

Systems Software

Lecture 5: Dependability & Replication

Simin Nadjm-Tehrani

Real-time Systems Laboratory

Department of Computer and Information Science
Linköping University

Overview

Last lecture:

- distributed systems are hard to design with guaranteed services when failures are considered

This lecture:

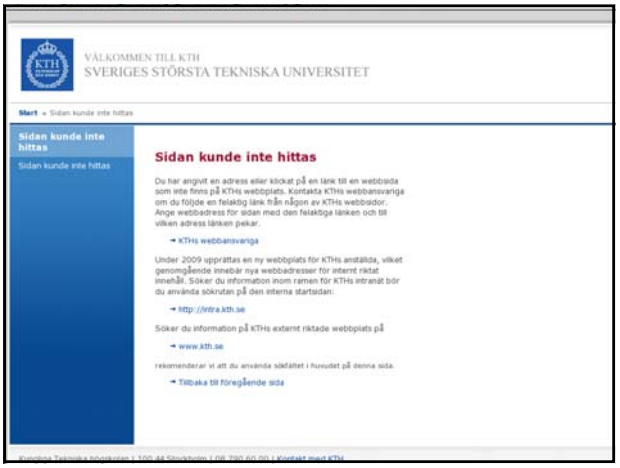
- we will have a closer look at failures and related concepts
- We will look at replication as a common means of achieving fault tolerance

Booking an aerobics pass

```
Microsoft OLE DB Provider for ODBC
  Drivers error '80040e31'

[MySQL][ODBC 3.51 Driver][mysqld-
  5.0.27-community-nt]Table
  '.\kdk_se\foreningsinfo' is marked
  as crashed and should be repaired

/eKop\funktioner.asp, line 114
```

Känner ni igen detta?

Google's 100 min outage

September 2, 2009:

A small fraction of Gmail's servers were taken offline to perform routine upgrades.

"We had slightly underestimated the load which some recent changes (ironically, some designed to improve service availability) placed on the request routers ."

Dependability

- How do things go wrong?
- Why?
- What can we do about it?

- Basic notions in dependable systems and replication for fault tolerance

(sv. **Pålitliga datorsystem**)

What is dependability?

*Property of a **computing system** which allows reliance to be justifiably placed on the service it delivers.*

[Avizienis et al.]

The ability to avoid service failures that are more frequent or more severe than is acceptable.

Attributes of dependability

IFIP WG 10.4 definitions:

- **Safety**: absence of harm to people and environment
- **Availability**: the readiness for correct service
- **Integrity**: absence of improper system alterations
- **Reliability:** continuity of correct service
- **Maintainability**: ability to undergo modifications and repairs

Reliability

[Sv. Tillförlitlighet]

Means that the system (functionally) behaves as specified, and does it *continually* over measured intervals of time.

Typical measure in aerospace: 10^{-9}

Another way of putting it: MTTF - One failure in 10^{9} flight hours.

Availability

- Not to be confused with reliability…

- How to measure availability?

Once upon a time…

- We could trust that some services were available and reliable
  - Police and Rescue services
  - Banking sector
  - Telecommunications
  - …

26 January 2007

- Stockholm: The police telephone system stopped working at 2 a.m. and was not back to working state until 7 a.m.
- Both the emergency number and several local exchanges were out of order
- Happened when the telecom operator was installing an updated answering machine

2 June 2007

- The Royal Bank of Scotland, which also owns NatWest, has apologised after its cashpoint, online, and telephone banking systems all crashed.
- A spokeswoman said: "We are very sorry, and we're working to sort it out."

20 August 2007

- The network outage was caused by "massive restart of [its] user's computers across the globe within a very short timeframe after a routine software update."
- Resulted in a high number of log-in requests, leading to what Skype calls a "chain reaction."

Faults, Errors & Failures

- **Fault**: a defect within the system or a situation that *can* lead to failure
- **Error**: manifestation (symptom) of the fault - an unexpected behaviour
- **Failure**: system not performing its intended function

Examples

- Year 2000 bug
- Bit flips in hardware due to cosmic radiation in space
- Loose wire
- Air craft retracting its landing gear while on ground

Effects in time:

Permanent/ transient/ intermittent

Means to achieve

... dependability according to [IFIP 10.4]:

1. Fault prevention
2. Fault removal
3. ⇨ Fault tolerance
4. Fault forecasting

Fault ⇒ Error ⇒ Failure

- Goal of system verification and validation is to eliminate faults

Some will remain...

- Goal of hazard/risk analysis is to focus on important faults
- Goal of fault tolerance is to reduce effects of errors if they appear - *eliminate or delay failures*

On-line fault management

- Fault detection
  - By program or its environment
- Fault tolerance using redundancy
  - software
  - hardware
  - Data
- Fault containment by architectural decisions

Redundancy

From D. Lardner: Edinburgh Review, year 1824:

"The most certain and effectual check upon errors which arise in the process of computation is to cause the same computations to be made by separate and independent computers; and this check is rendered still more decisive if their computations are carried out by different methods."*

** people who compute*

Static or Dynamic

Static redundancy:

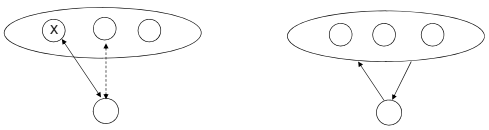
- Used all the time (whether an error has appeared or not), just in case...

Dynamic redundancy:

- Used only when the error appears and specifically aids the treatment

Server replication models

- Primary backup
- Active replication



: Denotes a replica group

State consistency

- Primary backup:
  - Every response to clients is preceded by an update to all servers and their acknowledgement
- Active replication:
  - Typically via a reliable multicast
  - Some ordering requirement (FIFO, ...)
- States of backups (or active replicas) should be the same when a replica failure takes place

Group view

- All servers are notified about a crashing server at the same “logical” time defined by the order of received messages

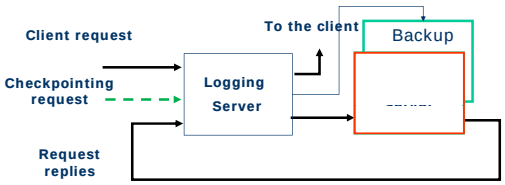
Supporting fault tolerance

- Server replication in larger groups requires complex lower layer services (like group membership, multicast, order processing)

Middleware!

- Guaranteeing state consistency in presence of failures is difficult (needs proofs about those underlying protocols) and requires synchrony assumptions

Checkpointing



Challenge: to checkpoint often enough but not too often

Questions?