Verteilte Systeme

Vorlesung 10 vom 24.06.2004

Dr. Sebastian Iwanowski
FH Wedel

Inhaltsverzeichnis für die Vorlesung

Technische Details zur Datensynchronisation

Wer initiiert das Anlegen von Datenkopien ?

- **automatisch: Permanente Kopien**

- **Server ergreift Initiative: push-Caches**

  aus Kapazitätsgründen

  Heranschieben der Daten an häufig fragende Clients

  Lösung: Regelungstechnischer Ansatz

  gibt Schwellen vor für:
  - ***Löschen einer Datenkopie***
  - ***Erstellen einer Datenkopie***

- **Client ergreift Initiative: Caches**

  zum Verbessern der Antwortzeiten

Was wird bei Aktualisierungen weitergegeben ?

1) Ein aktualisierter Datensatz

2) Benachrichtigung über einer Aktualisierung

3) Eine Aktualisierungsoperation, mit der man einen alten Datensatz auf den neuesten Stand bringen kann

Wie wird eine Aktualisierungen weitergegeben ?

- **push**

  für permanente Kopien

  für Server-initiierte Kopien

  für Client-initiierte Kopien bei Verwendung großer Caches durch mehrere Clients

- **pull**

  für Client-initiierte Kopien bei kleinen individuellen Caches

- **Mischform *lease*:**

  Client bitten Server um Versorgung mit Aktualisierungen (pull)

  Server verspricht und übernimmt Aktualisierungen für bestimmte Zeit (push)

Epidemische Aktualisierungsprotokolle

Wie verteilt man Aktualisierungen schnell übers Netz ?

- **Infektiöse Server**

  push: Server infiziert beliebigen anderen

- **Empfängliche Server**

  pull: Server fragt beliebigen anderen

Epidemische Aktualisierungsprotokolle

- **Variante: Geschwätzige Server**

  Server infiziert mit vorgegebener Wahrscheinlichkeit anderen Server

  Wenn der andere Server schon infiziert war, nimmt die Wahrscheinlichkeit ab

- **Zum Löschen von Daten: *Sterbeurkunden***

Konzepte zur Erzielung von Fehlertoleranz

Fehlertypen

Server-Fehler:

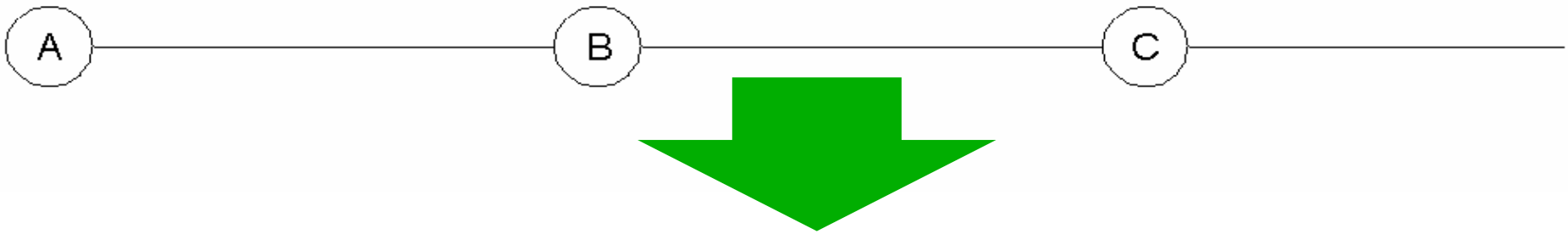
1) Totalausfall
2) Auslassungsfehler
3) Verzögerungsfehler
4) Verfälschungsfehler
5) Zufällige Fehler

Leitungsfehler:

1) Totalausfall
2) Sporadischer Ausfall

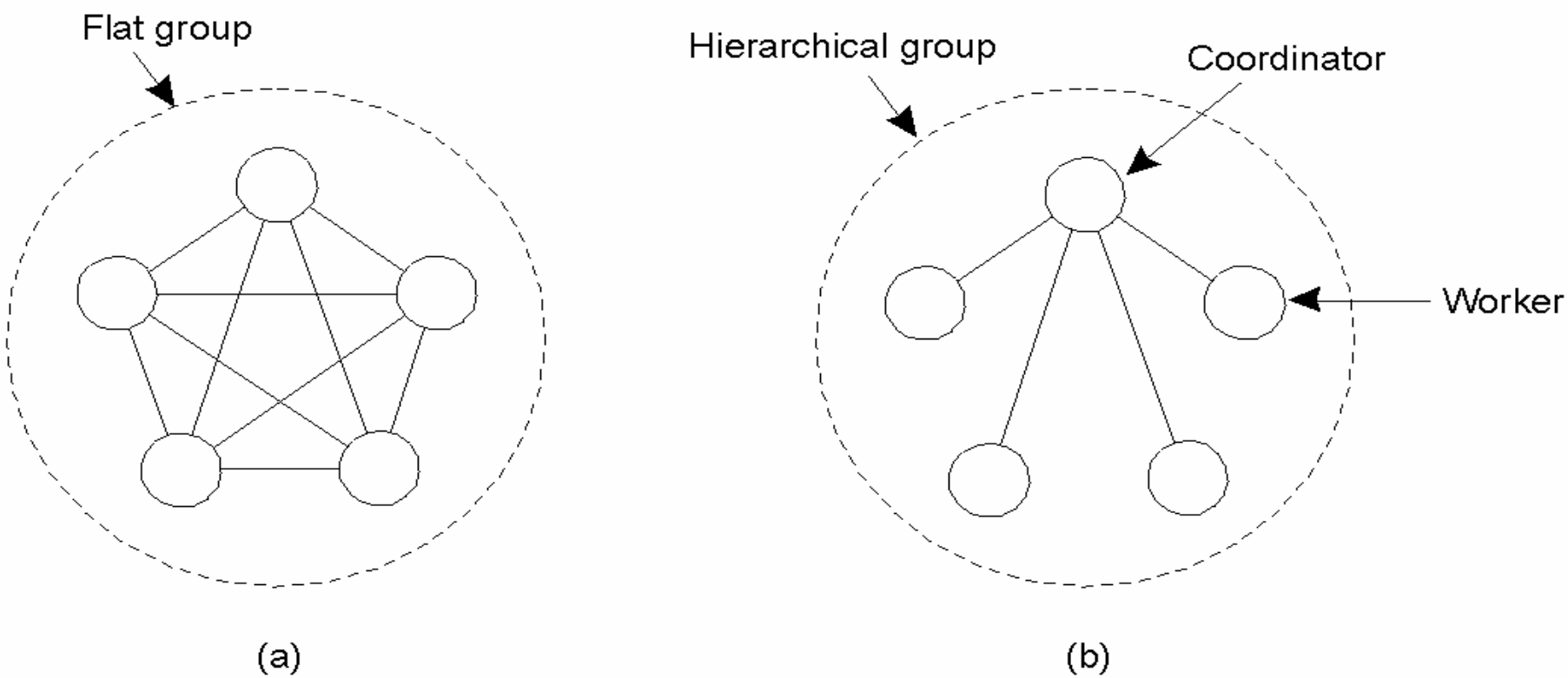
Fehlertoleranz durch Redundanz

Prinzip:



Koordination der redundanten Server

Flache vs. hierarchische Koordination:



Fehlermaskierungsverfahren (Server)

Gegeben:

- **n redundante Server**
- **Jeder Server sammelt Informationen**
- **Die Server tauschen periodisch die Informationen aus mit dem Ziel, dass alle den gleichen Informationsstand haben**
- **Einige Server können fehlerhaft arbeiten !**

Gesucht:

- **Verfahren, das erreicht, dass die korrekten Server am Ende die richtigen Informationen haben**

Fehlermaskierungsverfahren (Server)

Flache Koordination:

- **Die Informationen aller Server werden als gleichwertig angesehen und verarbeitet**

Hierarchische Koordination:

- **Alle Informationen gehen beim Koordinator ein und werden von dort weiter verteilt.**

Gesucht:

- **Verfahren, das erreicht, dass die korrekten Server am Ende die richtigen Informationen haben**

Fehlermaskierungsverfahren (Server)

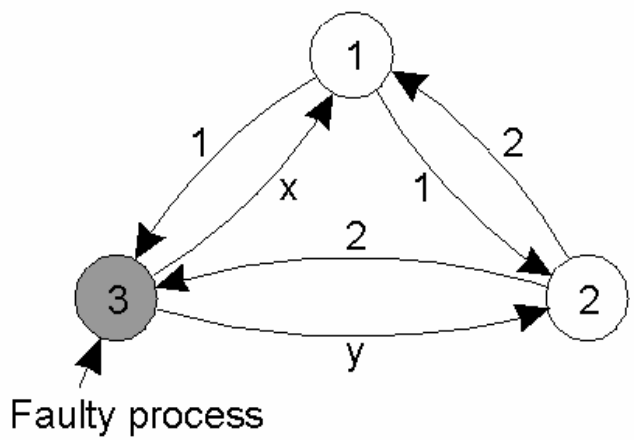
Lösungsverfahren für die flache Koordination: Mehrheitsentscheid

a) **Die Server tauschen sich paarweise aus.**

b) **Jeder Server bildet Vektoren aus den Werten, die er von den anderen Servern erhalten hat.**

c) **Die Server tauschen paarweise diese Vektoren aus.**

d) **Als gültige Werte werden dann die Mehrheiten an den jeweiligen Positionen angenommen**



(a)

Beispiel mit 3 Servern:

1 Got(1, 2, x)
2 Got(1, 2, y)
3 Got(1, 2, 3)

(b)

1 Got	2 Got
(1, 2, y)	(1, 2, x)
(a, b, c)	(d, e, f)

(c)

Fehlermaskierungsverfahren (Server)

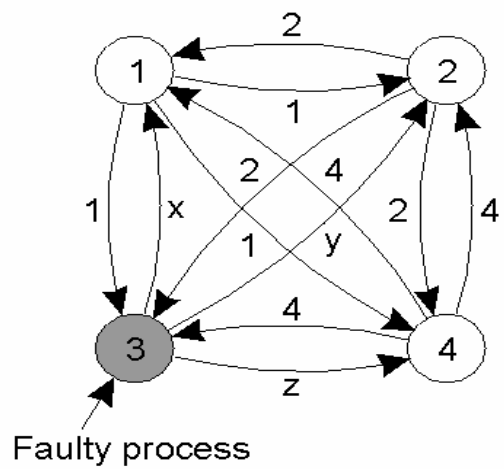
Lösungsverfahren für die flache Koordination: Mehrheitsentscheid

a) **Die Server tauschen sich paarweise aus.**

b) **Jeder Server bildet Vektoren aus den Werten, die er von den anderen Servern erhalten hat.**

c) **Die Server tauschen paarweise diese Vektoren aus.**

d) **Als gültige Werte werden dann die Mehrheiten an den jeweiligen Positionen angenommen**



(a)

Beispiel mit 4 Servern:

1	Got(1, 2, x, 4)
2	Got(1, 2, y, 4)
3	Got(1, 2, 3, 4)
4	Got(1, 2, z, 4)

(b)

1 Got	2 Got	4 Got
(1, 2, y, 4)	(1, 2, x, 4)	(1, 2, x, 4)
(a, b, c, d)	(e, f, g, h)	(1, 2, y, 4)
(1, 2, z, 4)	(1, 2, z, 4)	(i, j, k, l)

(c)

Fehlermaskierungsverfahren (Server)

3 anscheinend verschiedene Problemstellungen:

1) **Übereinstimmung (Consensus):**
   Vor.: Alle Server erhalten gleiche Werte.
   Ziel: Die korrekten Server haben den richtigen Wert.

2) **Byzantinische Generäle:**
   Vor.: Alle Server erhalten gleichen Wert von Koordinator.
   Ziel: Die korrekten Server haben den richtigen Wert.

3) **Interaktive Übereinstimmung:**
   Vor.: Alle Server erhalten einen beliebigen Wert.
   Ziel: Die korrekten Server haben die Werte aller anderen korrekten Server.

Satz: Die Probleme sind äquivalent

Beweis: siehe Coulouris, S. 454

Beim nächsten Mal:

Theoretische Betrachtungen zur Fehlertoleranz
Auswahl eines Koordinators
Fehlertoleranz bei Leitungsfehlern
Sicherheit