

0.1 Überblick

- **probabilistischer Informationsaustausch**
keine Garantien
- **Wiederholung der einzelnen Arbeitsschritte (endlos)**
- **analog zur Gerüchteverbreitung oder zu Krankheitsepidemien**
disease - contaminated - infect - epidemics
- **historisch zur Sicherung der Konsistenz verteilter Datenbanken**
inzwischen viele Anwendungsgebiete (später mehr)
über 20 Jahre alt, aber erst in den letzten Jahren beliebter geworden.
robust, indifferent gegenüber veränderungen in der gruppensam-
mensetzung in verteilten systemen benötigt wg. continuous change
→ convergent behaviour

0.2 Struktur

Nach “Gossiping in Distributed Systems”, Anne-Marie Kermarrec, Maarten van Steen

Einschub:

Begriffserklärung Peers:

- **Prozesse**
possibly dyn changing
- **haben Cache mit Referenzen zu anderen Peers**
- **ggf. auch peer-spezifische Informationen im cache**
z.B. gewichtung

Peer 1 wählt zufälligen Peer 2 für Kommunikation → wählt Information und sendet an 2 → 2 wählt Information und sendet an 1 → evtl. müssen Daten aus dem Cache ersetzt werden, wenn dieser voll ist.

0.2.1 Peerauswahl

- **verschiedene Auswahlkriterien je nach Anwendung**
z.B. - Zufall, Auswahl nach bestimmten Kriterien
- **Unterschiede bei Auswahl über kabellose oder kabelgebundene Verbindungen**
kabellos kann nur peers in reichweite wählen oder verliert Datendurchsatz falls peers als "Proxy" arbeiten müssen
- **Simulation eines anderen Verbindungstyps möglich, häufig teurer und unnötig**
- **kaum Unterschiede auf Applikationsschicht zwischen synchron und asynchron**
asynchron heißt, dass peer 2 keine infos an peer 1 zurückgibt
- **asynchron ist kein "richtiges" Gossiping**

0.2.2 Datenaustausch

- **peers entscheiden, welche Daten sie austauschen**
- **entweder Applikationsdaten oder Referenzen zu anderen peers werden ausgetauscht**

0.3 Datenverarbeitung

- **stark anwendungsabhängig**