

Verteilte Systeme

Prof. Dr. Martin Becke

CaDS - HAW Hamburg

Version 0.9

Algorithmen

Garbage

- ▶ Objekte über verschiedene Knoten hinweg verteilt
- ▶ Kommunikation zwischen den Knoten für gemeinsamen Status des Objektes
- ▶ Verschiedene Ansätze interessant.
Ansätze zeigen wie auf Anforderungen des Netzes oder die Applikation eingegangen werden kann.

Garbage

Ansätze

- ▶ Referenzzählung (Keine Zyklischen Referenzen)
- ▶ Verteilte Mark-and-Sweep (Kommunikation über eine Baumstruktur)
- ▶ Garbage Collection mit Leases (Gültigkeit über Zeit - bedingt zuverlässige Kommunikation oder Fehlertoleranz)
- ▶ Verteilte Garbage Collection durch Tracing (Mehr Kommunikation als Mark and Sweep)
- ▶ Generational Garbage Collection (Bedingt bekannte Code Base, sonst Speicherleaks)

Algorithmen

Election

- ▶ Election in Rings (Ring Election Algorithmus) (Stabil, bekannte Nodes)
- ▶ Bully Algorithmus (Weniger stabil, weniger bekannte Nodes)
- ▶ Echo-Algorithmus mit Ausnahmen (Unbekannte Nodes, recht Fehlertolerant)
- ▶ Tree Election Algorithmus (Unbekannte Nodes aber sehr stabil in ihrer Ordnung)
- ▶ Minimum Spanning Tree (Unbekannte Nodes, stabil aber mit weichen Kriterium)

Election

Anwendungsfall Bussystem

- ▶ IEEE 1394 ist ein serieller Busstandard
- ▶ Zur Datenübertragung und Kommunikation eingesetzt (auch als FireWire bekannt)
- ▶ Zentrale Komponente ist der Bus Manager
- ▶ Wahl des Bus Manager nahe am Bully

IEEE 1394

Wahl des Bus Managers

- ▶ Selbstnominierung : Jedes Gerät auf dem Bus, das die Rolle des Bus Managers übernehmen kann und will, sendet eine Selbstevaluierungsnachricht. Diese Nachricht enthält die Fähigkeiten des Geräts.
- ▶ Vergleich der Fähigkeiten : Jedes Gerät auf dem Bus vergleicht seine eigenen Fähigkeiten mit den in den Selbstevaluierungsnachrichten der anderen Geräte angegebenen Fähigkeiten.
- ▶ Wahl : Das Gerät mit den höchsten Fähigkeiten wird zum Bus Manager gewählt. (Gleichstandregel)
- ▶ Kommunikation der Wahl : Das gewählte Gerät sendet eine Nachricht an alle anderen Geräte auf dem Bus.

IEEE 1394

IEEE 1394 und Bully

- ▶ Beide Algorithmen versuchen, einen Leader in einem verteilten System zu wählen
- ▶ Bully-Algorithmus setzt eher auf die Identität der Knoten
- ▶ Bei IEEE 1394 Algorithmus Fokus auf Ressourcenvergleich

Election

Wahl in anonymen Ring

- ▶ Alle Knoten in einem Ring identisch und ununterscheidbar (anonym)
- ▶ Auch keine eindeutige ID zuweisbar
- ▶ In solchen Fällen ist es unmöglich, einen eindeutigen Leader oder Koordinator zu wählen
- ▶ In synchronen Netzwerken mit begrenzten erwarteten Verzögerungen kann das Problem jedoch gelöst werden
- ▶ Dieser Algorithmus funktioniert, weil die begrenzten erwarteten Verzögerungen in einem synchronen Netzwerk garantieren, dass die Nachrichten in einer vorhersehbaren Reihenfolge ankommen

Election

Wahl in anonymen Ring

- ▶ Voraussetzung : alle Operationen finden in vorhersehbaren und begrenzten Zeitschritten statt
- ▶ Die Zeit ist Unterscheidungsmerkmal
 - ▶ Jeder Knoten sendet zu Beginn einer Zeiteinheit eine Nachricht
 - ▶ Wenn ein Knoten eine Nachricht erhält, bevor er seine eigene Nachricht sendet, gibt er die Wahl auf und wird zum Follower
 - ▶ Dieser Prozess wird so lange fortgesetzt, bis nur noch ein Kandidat übrig ist, der zum Leader wird

Algorithmen

Synchronizer

- ▶ Ziel : asynchrone verteilte Systeme zu synchronisieren
- ▶ Es gibt keine globale Uhr oder garantierte Nachrichtenlieferzeiten
- ▶ Bekannte Synchronizer sind Alpha, Beta und Gamma

Synchronizer

Funktionsweise

- ▶ Nachrichtenaustausch : Jeder Knoten in einem asynchronen Netzwerk sendet Nachrichten an seine Nachbarn und empfängt Nachrichten von ihnen.
- ▶ Synchronisierung : Der Synchronizer-Algorithmus stellt sicher, dass alle Knoten in regelmäßigen Abständen, die als Runden bezeichnet werden, synchronisiert werden. Am Ende jeder Runde wartet jeder Knoten, bis er Nachrichten von allen seinen Nachbarn erhalten hat, bevor er zur nächsten Runde übergeht.

Synchronizer

Alpha, Beta und Gamma

- ▶ Unterscheidung der Art und Weise wie Synchronisierung erreicht wird, sowie in ihrer Effizienz und Komplexität
- ▶ Im Allgemeinen sind sie unterscheidbar in Anzahl der benötigten Nachrichten und der Anzahl der Runden

Algorithmen

Mutual Exclusion

- ▶ Mutual Exclusion, oder wechselseitiger Ausschluss zum Schutz von kritischen Bereichen
- ▶ Notwendig in VS zur Sicherung der Datenkonsistenz und -integrität in Mehrprozess- oder Mehrfadenanwendungen

Mutual Exclusion

Ansätze

- ▶ Ricart-Agrawala-Algorithmus (Hohe Kommunikation)
- ▶ Raymond's Tree-Based Algorithmus (Hohe Netzstabilität)
- ▶ Agrawal-El Abbadi-Algorithmus (Schnelle und stabile Netze)
- ▶ Peterson's Algorithmus (Akademischer Ansatz zur Lehre)
- ▶ Bakery Algorithmus (Problem bei vielen Teilnehmern)
- ▶ Fischer's Test-Set-and-Set-Lock Algorithmus (Bei geringen Einsätzen)
- ▶ Queue-Locks (Lamport Clocks, Multicast mit Acks)

Algorithmen

Mutual Exclusion

Klassisches Warten auf andere Ablauf­fäden

- ▶ Sense-Reversing Barrier (Einfacher Counter)
- ▶ Combining Tree Barrier (Baumstruktur)
- ▶ Tournament Barrier (Unstrukturiert mit Paaren)
- ▶ Dissemination Barriers (Gossip-ähnlich, Runden-basiert)

Algorithmen

Self-Stabilization

- ▶ Ansatz der Fehlertoleranz für Algorithmen
- ▶ Aus beliebigen initialen Zuständen in einen korrekten Zustand zu wechseln
- ▶ Wichtig es garantiert nach einer endlichen Anzahl von Schritten einen korrekten Zustand

Self-Stabilization

Ansätze

- ▶ Dijkstra's Token Ring for Mutual Exclusion (Garantiert Existenz von einem Token)
- ▶ Arora-Gouda Spanning Tree (Kann beliebig neue Bäume erstellen)
- ▶ Afek-Kutten-Yung Spanning Tree (Verbesserung von Arora-Gouda Spanning Tree - Gruppen von Knoten)

Algorithmen

Smart Contracts

- ▶ « Distributed Ledger Technology » (DLT) \Rightarrow Zentrale Datenbasis
- ▶ « Ledger » ist im Grunde eine Liste von (digitalen) Transaktionen
- ▶ Bekanntes Beispiel Blockchain

Smart Contracts

Blockchain

- ▶ Blockchains sind eine Form der verteilten Ledger-Technologie
- ▶ Protokollierungen von Transaktionen in einem Peer-to-Peer-Netzwerk
- ▶ Jede Transaktion wird in einem Block aufgezeichnet
- ▶ Jeder Block ist mit dem vorhergehenden Block verbunden
- ▶ Ergibt eine Kette von Blöcken oder eine « Blockchain »
- ▶ Jede Transaktion kann von jedem Knoten im Netzwerk gesehen und verifiziert werden
- ▶ Das Verbinden wird mit kryptographischen Verfahren abgesichert

Smart Contracts

Idee

- ▶ Smart Contracts sind Programme, die auf einer Blockchain ausgeführt werden
- ▶ Sie sind « smart », weil sie automatisch ausgeführt werden
- ▶ Sobald ein Smart Contract auf die Blockchain hochgeladen wurde, kann er nicht mehr geändert oder gestoppt werden (Dezentrale Börse)
- ▶ Ethereum ist eine andere Open-Source-Blockchain-Plattform
- ▶ Aktuell viele dezentrale Anwendungen (dApps), die auf Smart Contracts basieren (Beispiel : Wettsysteme, Logistik, Spiele, etc)
- ▶ Jede Transaktion kann von jedem Knoten im Netzwerk gesehen und verifiziert werden