Randomisierte Algorithmen: Ben-Or

- Literatur
  - Michael Ben-Or: *Another Advantage of Free Choice: Completely Asynchronous Agreement Protocols*. Proc. 2nd ACM Symposium on Principles of Distributed Computing,1983
  - Nancy Lynch: *Distributed Algorithms*.

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

Randomisierte Algorithmen: Ben-Or

- Rahmenbedingungen
  - Asynchrones Systemmodell
  - Randomisierte Lösung des Einigungsproblem
    - Partielle Korrektheit: Wenn der Algorithmus terminiert, ist das Ergebnis korrekt (alle Knoten einigen sich auf den selben Wert)
    - Terminierung: *Keine deterministische Aussage möglich! Aber:* Die Wahrscheinlichkeit, dass der Algorithmus in endlicher Zeit terminiert, konvergiert gegen 1
  - BO-1: Gutmütige Ausfälle
  - BO-2: Byzantinische Fehler

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

Randomisierte Algorithmen: Ben-Or

- BO-1 (gutmütige Ausfälle)

  N Knoten; max. t Ausfälle, N > 2t; Rundenzähler r, Knoten P hat Wert x_P

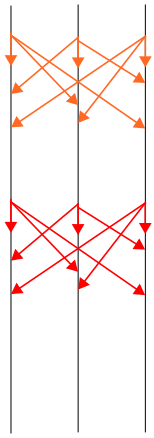
  Schritt 0: r := 1

  Schritt 1: Sende **(pre-vote, r, x_P)** an alle Knoten

  Schritt 2: Warte auf N - t Nachrichten **(pre-vote, r, x)**;
  Anzahl Nachrichten mit gleichem Wert x=v:
  Mehr als N/2: sende **(vote, r, v)** an alle
  ansonsten: sende **(vote, r, ?)** an alle

  Schritt 3: Warte auf N - t Nachrichten **(vote, r, v)**
  (a) Falls eine Nachricht mit v ≠ ?: x_P := v
  (b) Falls mehr als t solche Nachrichten: fertig
  (c) sonst: x_P := zufällig gewählt aus {0, 1}

  Schritt 4: r := r + 1; weiter mit Schritt 1



(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

Randomisierte Algorithmen: Ben-Or

- Überlegungen zur Korrektheit von BO-1
  - Alle Knoten senden in Schritt 2 **vote**-Nachrichten entweder nur für einen Wert aus {0,1}, oder für '?'
    (gleicher Initialwert von mehr als N/2 Knoten)
  - Alle Knoten, die in einer Runde in Schritt 3a einen Wert annehmen, nehmen daher den gleichen Wert an
  - Falls mehr als t Knoten eine vote-Nachricht für 0/1 versendet haben, ist in jeder Teilmenge aus N-t Knoten mindestens eine solche empfangen worden.
  - D.h. Falls sich ein Knoten in Schritt 3b endgültig entscheidet, nehmen alle anderen, die sich noch nicht entscheiden, in Schritt 3a den gleichen Wert an und entscheiden sich in der nächsten Runde

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

Randomisierte Algorithmen: Ben-Or

- Eigenschaften von BO-1
  - Beginnen alle Knoten mit dem selben Wert, terminiert der Algorithmus in nur einer Runde. Ansonsten gilt für die Dauer bis zur Einigung (ohne Beweis):
    - In bestimmten Fällen liegt eine exponentielle erwartete Rundenzahl vor (insbes. bei großem t)
    - für den Speziallfall $t = O(\sqrt{N})$ ist der Erwartungswert der Rundenzahl jedoch konstant

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

Randomisierte Algorithmen: Ben-Or

- BO-2 (byzantinische Fehler)

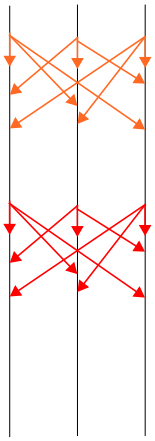
N Knoten; max. t fehlerhaft, N>5t; Rundenzähler r, Knoten P hat Wert x_P

Schritt 0: r := 1

Schritt 1: Sende **(pre-vote, r, x_P)** an alle Knoten

Schritt 2: Warte auf N - t Nachrichten **(pre-vote, r, x)**;
Anzahl Nachrichten mit gleichem Wert x=v:
Mehr als (N+t)/2: sende **(vote, r, v)** an alle
ansonsten: sende **(vote, r, ?)** an alle

Schritt 3: Warte auf N - t Nachrichten **(vote, r, v)**
(a) Falls > t Nachrichten mit $v \neq ?$: $x_P := v$
(b) Falls > (N+t)/2 solche Nachrichten: fertig
(c) sonst: x_P := zufällig gewählt aus {0, 1}



Schritt 4: r := r + 1; weiter mit Schritt 1

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

Randomisierte Algorithmen: Ben-Or

- Überlegungen zur Korrektheit BO-2
  - Ähnlich zu BO-1, nur
    - statt "mindestens 1 Knoten":
      "mindestens 1 korrekter Knoten", d.h. mindestens t+1 Knoten
    - statt "Mehrheit der Knoten":
      Mehrheit der korrekten Knoten, d.h. mehr als (N+t)/2 Knoten
    - Schritt 3 b:
      (N+t)/2 vote-Nachrichten für v $\Rightarrow$
      mindestens (N+t)/2 - t korrekte Knoten dabei $\Rightarrow$
      jeder Knoten erhält mindestens ((N+t)/2 - t) - t solche vote-Nachrichten

      $((N+t)/2 - t) - t > t \Leftrightarrow N > 5t$

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

Randomisierte Algorithmen: Ben-Or

- Überlegungen zur Korrektheit BO-2
  - D.h. falls N > 5t gilt, folgt folgendes:

    Sobald sich ein Knoten in Schritt 3b endgültig entscheidet, verwenden alle andere (noch nicht entschiedenen Knoten) in Schritt 3a den gleichen Wert und entscheiden sich endgültig in der nächsten Runde

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

Randomisierte Algorithmen: Ben-Or

- Folgerungen
  - Das Einigungsproblem ist in asynchronen Systemen durch randomisierte Algorithmen grundsätzlich lösbar
  - Bei byzantinischen Fehlern ermöglicht BO-2 eine Lösung bei N > 5t. Noch unklar bleibt, ob es randomisierte Lösungen mit 5t >= N > 3t gibt
  - Bei kleinem t ($t = O(\sqrt{N})$) können die Algorithmen von Ben-Or eine effiziente Lösung darstellen; für große Werte von t steigt die erwartete Rundenzahl aber schnell an, BO ist dann nicht mehr praktikabel

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Aktuelle Forschungsarbeit bei IBM Research in Zürich,
  - ABBA: Asynchronous Binary Byzantine Agreement
  - Sintra: Secure INtrusion Tolerant Replication Architecture
- Literatur
  - C. Cachin, K. Kursawe, V. Shoup: *Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement using Cryptography*. ACM Symposium on Principles of Distributed Computing, July 2000
  - C. Cachin, K. Kursawe, F. Petzold, V. Shoup: *Secure and Efficient Asynchronous Broadcast Protocols*. Proc. Crypto 2001
  - C. Cachin, J. Poritz: *Secure Intrusion-tolerant Replication on the Internet*. Proc. Intl. Conference on Dependable Systems and networks, 2002

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Überblick
  - Grundlagen
    - Schwellwert-Signaturen und verteilt erzeugte Zufallszahlen
  - ABBA: Asynchronous Byzantine Binary Agreement
    - Systemmodell
    - Algorithmus
    - Betrachtungen zur Korrektheit
  - Verwendung in Sintra

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- ABBA: Asynchronous Byzantine Binary Agreement
  - Asynchrones Systemmodell, byzantinische Fehler
  - Lösung des Einigungsproblem in einem randomisierten Systemmodell
  - Garantiert werden neben partieller Korrektheit auch Deadlock-Freiheit und schnelle Konvergenz
    - *Konsistente Einigung (partielle Korrektheit):* Sobald sich ein Knoten endgültig für einen Wert entschieden hat, verwendet jeder andere Knoten, der sich ebenfalls für einen Wert entscheidet, genau diesen Wert

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- *Deadlock-Freiheit:* Es kann kein Zustand erreicht werden, in dem:
  - der Algorithmus in allen korrekten Knoten aktiviert wurde
  - der Algorithmus noch nicht terminiert ist
  - alle Nachrichten, die von aktivierten korrekten Knoten ausgesendet wurden, zugestellt und verarbeitet worden sind (Sonst würde das System auf den Empfang von Nachrichten warten, aber kein korrekten Knoten würde Nachrichten aussenden)
- *Schnelle Konvergenz:* Die Anzahl x der Nachrichten, die bei einer Einigung von allen korrekten Knoten erzeugt werden, ist beschränkt durch die Warscheinlichkeitsaussage

  $P(x > 2 r + 1) < 2^{-r} + \varepsilon$ mit ε vernachlässigbar klein

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

ABBA

ABBA Systemmodell

- Alle Knoten verhalten sich *berechenbar:* Alle Schritte durch probabilistisch polynomiale Berechnungen beschreibbar
- Insgesamt n Knoten $P_1 ... P_n$, davon maximal t fehlerhaft; Initialisierung des Systems durch einen "Trusted Dealer"
- Verwendung von:
  - (n, t+1, t) Schwellwert-Signatur S_0
  - (n, n-t, t) Schwellwert-Signatur S
  - (n, n-t, t) Schema für verteilte Zufallszahlen (Coin Shares)
- Jede Entscheidung hat eindeutige Kennung (TID). Jeder Knoten i hat für eine TID einen Initialwert V_i aus {0,1}
- Alle Nachrichten werden authentifiziert

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

ABBA

ABBA: Überblick über den Algorithmus

(1) pre-vote mit aktuellem lokalem Wert

(2) n-t pre-vote empfangen:
n-t gleiche Werte: main-vote für diesen Wert
sonst: main-vote mit "abstain"

(3) n-t main-vote empfangen:
n-t gleiche Werte empfangen: endgültige Entscheidung
ansonsten: "coin share" wird veröffentlicht

(4) neuer lokaler Wert wird aus main-vote ermittelt:
nur "abstain": Zufallswert aus coin shares
sonst: Wert aus main-vote

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

ABBA: Der Algorithmus im Detail

- Nur erste Runde: Vorverarbeitungs-Schritt:
  - Jeder sendet seinen Initialwert b (0/1) durch (pre-process, b) mit S_0 signiert an alle anderen
  - Sobald 2 t + 1 Nachrichten eingetroffen sind, wird für den Mehrheitswert (t+1 gleiche Nachrichten) eine S_0-Signatur erzeugt. Dieser Wert wird als Initialwert b für die folgenden Schritte verwendet
  - Falls alle gutartigen Knoten den selben Initialwert besitzen, ist sichergestellt, dass es nur für diesen Wert eine Signatur geben kann

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

ABBA: Der Algorithmus im Detail (2)

- Alle Runden r
  - Jeder sendet (pre-vote, r, b, justification) mit gültiger S-Teilsignatur an alle;

    justification: Beweis für die Gültigkeit des Werts b:

    Runde r=1: (pre-process, b) mit S_0-Signatur
    Runde r>1: (pre-vote, r-1, b) mit S-Signatur ("hard pre-vote")
    oder (main-vote, r-1, abstain) mit S-Signatur, und Wert b ermittelt durch b := coin(TID, r-1) ("soft pre-vote für b")

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

ABBA

ABBA: Der Algorithmus im Detail (3)

- Nach Empfang von n-t gültigen pre-vote-Nachrichten:
  - jeder Knoten erzeugt (main-vote, r, b, justification) mit b aus {0, 1, abstain} und justification als Gültigkeitsbeweis dafür
    - pre-vote sowohl für b=0 als auch b=1: b := abstain;
      justification: je ein pre-vote mit b=0,1
    - Ansonsten: n-t pre-vote für einen Wert b;
      justification: daraus gewonnene gültige S-Signatur
- Nach Empfang von n-t gültigen main-vote-Nachrichten:
  - Falls alle Werte gleich: Entscheidung für diesen Wert
  - Ansonsten: weiter mit nächster Runde;
    - Aufdeckung von coin(TID,r)
    - Falls alle n-t “abstain”-Werte: gültige S-Signatur für abstain
    - Ansonsten: mind. ein Wert, als justification: pre-vote mit S-Sig.

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg
Einigung.fm: 10.12.03 13:20

ABBA - Korrektheitsüberlegungen

- Möglichkeiten des Angreifers
  - Nachweis der Korrektheit aller Schritte durch Signaturen
  - Dadurch Möglichkeiten eines Angreifers stark eingeschränkt
- Entscheidungsfindung
  - Nur für einen Wert b aus {0,1} kann es eine erfolgreich S-signierte pre-vote-Nachricht geben
  - Wenn sich der erste ehrliche Knoten in Runde r entscheidet, dann entscheiden sich alle ehrlichen Knoten in Runde r oder r+1

    n-t main-vote für einen Wert b in Runde r
    ⇒ mindestens n-2t main-vote für b von ehrlichen Knoten in Runde r
    ⇒ maximal n - (n-2t) = 2t main-vote für abstain, 2t < n-t
    ⇒ kein soft pre-vote in nächster Runde möglich
    ⇒ alle pre-vote in Runde r für den selben Wert b

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

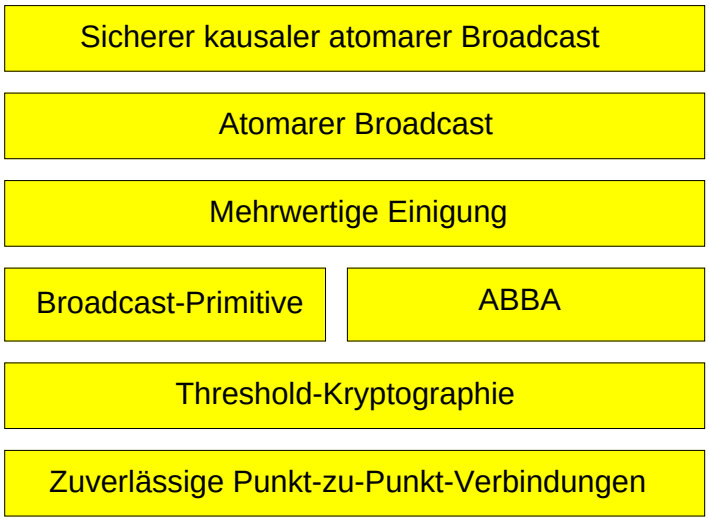
ABBA - Korrektheitsüberlegungen

- Deadlock-Freiheit und schnelle Konvergenz
  - Deadlock-Freiheit:
    - Es wird vorausgesetzt, dass maximal t Knoten sich fehlerhaft verhalten dürfen, und die Kommunikation zwischen fehlerfreien Knoten zuverlässig ist
    - Bei jeder Warteoperation wird auf n-t Nachrichten gewartet. Diese kommen sicher sicher immer an, weil es in jeden Fall so viele korrekte Knoten gibt, und jeder korrekte Knoten die erwartete Nachricht aussendet
  - Schnelle Konvergenz:
    - Es lässt sich zeigen, dass die Wahrscheinlichkeit, dass sich ein korrekter Knoten nicht innerhalb 2 r + 1 Runden entscheidet, nach oben durch $2^{-r} + \varepsilon$ beschränkt ist, mit ε vernachlässigbar

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- SINTRA: Secure Intrusion Tolerant Replication Architecture



(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Broadcast-Primitive
  - Zuverlässiger Broadcast: Alle ehrliche Knoten erhalten die selbe Nachricht, oder gar nichts
    - Der Sender sendet die Nachricht an alle
    - Jeder Knoten sendet ein "Echo" der Nachricht an alle
    - Sobald $\left\lceil \frac{n+t+1}{2} \right\rceil$ "Echo"-Nachrichten oder t + 1 "Fertig"-Nachrichten empfangen wurden, sendet ein Knoten "Fertig"-Nachrichten an alle
    - Sobald 2 t + 1 "Fertig"-Nachrichten empfangen wurden, akzeptiert ein Knoten die Nachricht und liefert diese an die nächste Schicht aus

      Falls der Sender nicht ehrlich ist, terminiert dieser Algorithmus u.U. nicht

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Mehrwertige Einigung

Einigung soll nur auf Werte erfolgen, die tatsächlich von mindestens einem Knoten vorgeschlagen wurden

- Jeder Knoten sendet seinen Initialwert als Vorschlag an alle über den beschriebenen zuverlässigen Broadcast
- Sobald ein Knoten von n - t Knoten Vorschläge für Initialwerte empfangen hat, führt er folgende Schritte wiederholt aus:
  - In jeder Wiederholung wird nach einem festgelegten Verfahren ein Kandidat P_a aus der Menge aller Knoten ausgewählt
  - Der Knoten startet ABBA schlägt den Wert 1 genau dann vor, wenn von P_a ein gültiger Vorschlagswert empfangen wurde
  - Falls die Einigungs auf 0 lautet, erfolgt der nächste Schleifendurchlauf, ansonsten ist P_a der Einigungswert

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Atomarer Broadcast
  - Garantie einer totalen Ordnung auf einer Folge von Nachrichten
  - Umsetzung mit Hilfe der mehrwertigen Einigung

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Sicherer kausaler atomarer Broadcast
  - Kausalität erfordert, dass ein Knoten den Inhalt einer Nachricht erst dann erfährt, wenn feststeht, wann diese Nachricht an die Empfänger ausgeliefert wird
    - Ein bösartiger Knoten soll also nicht in der Lage sein, abhängig von einer noch nicht ausgelieferten Nachricht eine andere Nachricht zu erzeugen, die vorher ausgeliefert wird
  - Umsetzung mit Hilfe von Schwellwert-Kryptographie
    - Sender verschlüsselt die Nachricht mit Hilfe eines globalen öffentlichen Schlüssels
    - Sobald die Nachricht ausgeliefert werden soll, kann sie von n - t Knoten gemeinsam entschlüsselt werden

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg

ABBA

- Zusammenfassung
  - Praktikable Lösung des Einigungsproblem in einem asynchronen, probabilistischen Modell
  - Tolerierung von bis zu t Fehlern bei insgesamt n > 3 t Knoten
  - Schnelle Konvergenz auf den Einigungswert
  - Hierarchische Struktur aus einzelnen modularen Blöcken
    - von einfacher, binärer Einigung
    - über atomaren Broadcast
    - bis hin zu sicheren kausalen atomaren Broadcast

(C) 2003 Hans P. Reiser, Rüdiger Kapitza, Lehrstuhl für Informatik 4, Universität Erlangen-Nürnberg