

Peer-to-Peer Systeme

-

Content-Addressable Networks (CAN)

Björn Schieble

`<schiesbn@studi.informatik.uni-stuttgart.de>`

Universität Stuttgart

13. Juli 2007

Inhalt

- 1 Einführung
- 2 Ein Content-Addressable Network
 - Aufbau
 - Routing
 - Knoten einfügen
 - Knoten verlässt das Netz
 - Fehlerbehandlung
- 3 Optimierungsmöglichkeiten
 - Mehrdimensionaler Koordinatenraum
 - Mehrere Koordinatenräume
 - Expressways
 - Idee
 - Aufbau der Routingtabelle
 - Aufbau des Expressways
 - Routing
- 4 Zusammenfassung

Einführung

Bekannte Peer-to-Peer Systeme:

- **Napster:** Datenaustausch dezentral zwischen den Peers. Der Suchindex wurde auf einem zentralen Server verwaltet.

Nachteile: "Single point of failure" und "Bottleneck" durch zentralen Suchindex

Einführung

Bekannte Peer-to-Peer Systeme:

- **Napster:** Datenaustausch dezentral zwischen den Peers. Der Suchindex wurde auf einem zentralen Server verwaltet.

Nachteile: "Single point of failure" und "Bottleneck" durch zentralen Suchindex

- **Gnutella:** Dezentralisierung des Suchindex.

Nachteile: Suche geschieht mittels Flooding, was dazu führt, dass unter Umständen Daten nicht gefunden werden obwohl sie sich im Netz befinden.

Grundidee: Sich die bereits bekannte und effektive Datenstruktur der Hashtabellen in verteilten Systemen zu Nutze zu machen.

Grundidee: Sich die bereits bekannte und effektive Datenstruktur der Hashtabellen in verteilten Systemen zu Nutze zu machen.

Distributed Hash Tables (DHT)

Ein Content-Addressable Network

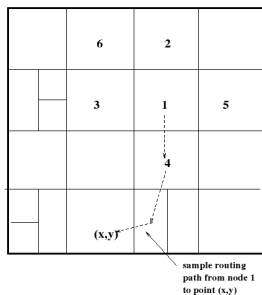
Aufbau und Funktionsweise eines CAN nach "A Scalable Content-Addressable Network" nach Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. (2001)

Ein Content-Addressable Network

Idee:

- Aufteilung in ein logischen Koordinatenraum
- Daten werden mit einem Hash-Wert in den Koordinatenraum abgebildet
- Knoten teilen sich den Koordinatenraum und speichern jeweils den Teil des Suchindex, der in ihren Bereich abgebildet wird.

Routing



- Eine Nachricht beinhaltet immer die Koordinaten des Zielpunktes
- Die Nachricht wird immer an den Nachbar weitergeleitet, der der Zielkoordinate am nächsten liegt.
- Es gibt immer mehrere mögliche Pfade, was die Ausfallsicherheit erhöht.

Knoten einfügen

- 1 Neuer Knoten muss einen Knoten finden der bereits ins CAN eingebunden ist (Einstiegspunkt).
- 2 Es muss ein Knoten gefunden werden der seinen Bereich mit dem neuen Knoten teilt
- 3 Die Nachbarn des geteilten Bereichs müssen ihre Routingtabelle anpassen.

Einstiegspunkt finden

- Mehrere Verfahren denkbar
- Eine Möglichkeit bieten sogenannte "Rendezvous Hosts"
- Diese Hosts verwalten eine Liste von Knoten die bereits im CAN eingebunden sind.
- Der neue Knoten verbindet sich über den Domainname mit dem "Rendezvous Host" und bekommt von ihm einen Knoten, der sich im CAN befindet.

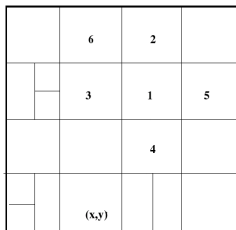
Knoten im CAN finden

- Der neue Knoten wählt einen beliebigen Punkt im Netz aus und sendet an diesen eine JOIN-Anfrage.
- Anfrage wird vom Eintrittsknoten zum Zielknoten geroutet.
- Der Zielknoten halbiert seine Zone.
- Dem neuen Knoten wird eine Hälfte der Zone und die zugehörigen Einträge des Suchindex übergeben.

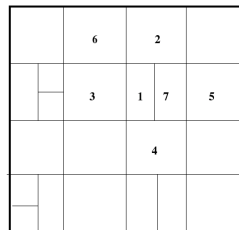
Routingtabelle anpassen

- Zuerst aktualisiere die Knoten, die sich einen Bereich geteilt haben, ihre Routingtabellen:
 - Der alte Knoten gibt die Nachbarn, die jetzt an den neuen Knoten grenzen an diesen ab.
 - Beide Knoten tragen ihr Nachbarschaftsverhältnis in ihre Routingtabellen ein.
- Als nächstes schicken die beiden Knoten eine Update-Nachricht an ihre Nachbarn, damit diese die neuen Nachbarschaftsverhältnisse lernen.

Beispiel

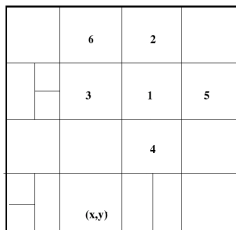


1's coordinate neighbor set = {2,3,4,5}
7's coordinate neighbor set = {}

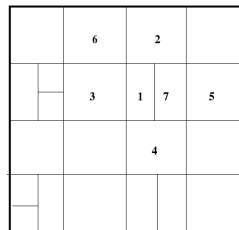


1's coordinate neighbor set = {2,3,4,7}
7's coordinate neighbor set = {1,2,4,5}

Beispiel



1's coordinate neighbor set = {2,3,4,5}
7's coordinate neighbor set = {}



1's coordinate neighbor set = {2,3,4,7}
7's coordinate neighbor set = {1,2,4,5}

Das Einfügen neuer Knoten beeinflusst nur einen kleinen Bereich des Koordinatenraums. Es müssen nur die Routingtabellen der direkten Nachbarn angepasst werden. Die Anzahl der Nachbarn ist nur von der Dimension ($2*d$) abhängig. Das Einfügen neuer Knoten beeinflusst also nur $O(d)$ Knoten.

Knoten verlässt das Netz

- Knoten übergibt seine Zone an einen seiner Nachbarknoten
- Dieser schickt eine Update Nachricht an seine Nachbarn, damit diese ihre Routingtabelle anpassen können.

Fehlerbehandlung

Knoten verlässt auf Grund eines Fehlers das CAN.

Fehlerbehandlung

Knoten verlässt auf Grund eines Fehlers das CAN.

- Jeder Knoten sendet in regelmäßigen Abständen Update-Nachrichten an seine Nachbarn mit seinen Koordinaten und einer Liste seiner Nachbarknoten.
- Bekommen die Nachbarn keine Update-Nachrichten von einem Knoten, dann gehen sie von einem Fehlerfall aus.

Fehlerbehandlung

Wiederherstellung eines lückenlosen CANs:

- Entscheidet ein Knoten, dass einer seiner Nachbarn nicht mehr aktiv ist, startet dieser einen Übernahme-Timer. Der Timer wird in Abhängigkeit der Größe der eigenen Zone initialisiert.
- Die anderen Nachbarn werden unabhängig voneinander ebenfalls einen Übernahme-Timer starten.
- Wenn der Timer abgelaufen ist schickt der Knoten eine Übernahme-Nachricht mit der Größe seiner Zone an alle Nachbarknoten
 - Empfängt ein Knoten eine Übernahme-Nachricht von einem Knoten der kleiner ist als er, dann beendet er seinen Timer
 - Ansonsten schickt er eine Übernahme-Nachricht.

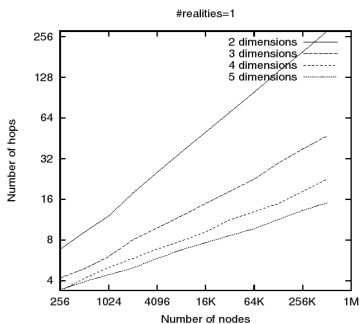
Optimierungsmöglichkeiten

Im vorgestellten CAN beträgt die durchschnittliche Pfadlänge beim Routing $O(n^{1/d})$ (für d Dimensionen und n Knoten). Dies lässt

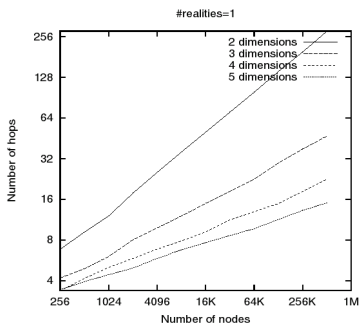
sich noch verbessern, durch

- mehrdimensionaler Koordinatenraum
- mehrere Koordinatenräume
- Expressways

Mehrdimensionaler Koordinatenraum

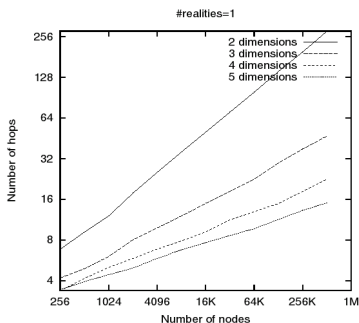


Mehrdimensionaler Koordinatenraum



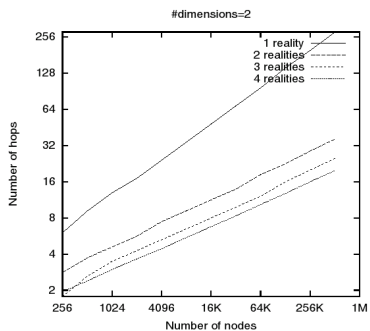
- Vorteile:
 - Reduzierte Pfadlänge beim Routing
 - Höhere Fehlertoleranz, da die Knoten mehr Nachbarn haben

Mehrdimensionaler Koordinatenraum

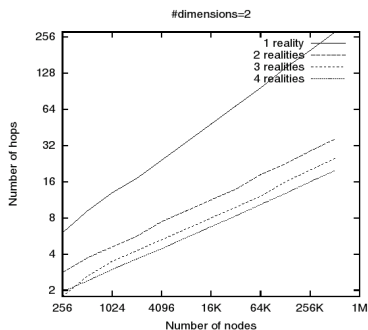


- Vorteile:
 - Reduzierte Pfadlänge beim Routing
 - Höhere Fehlertoleranz, da die Knoten mehr Nachbarn haben
- Nachteile:
 - größere Routingtabellen

Mehrere Koordinatenräume

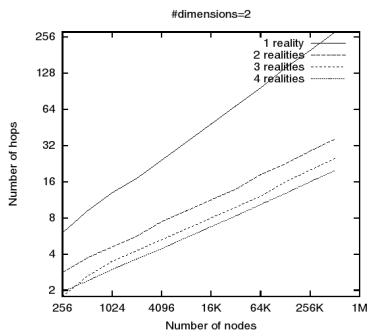


Mehrere Koordinatenräume



- Vorteile:
 - Reduzierte Pfadlänge beim Routing
 - Höhere Fehlertoleranz, da die Knoten in mehreren Ebenen routen können.

Mehrere Koordinatenräume



- Vorteile:
 - Reduzierte Pfadlänge beim Routing
 - Höhere Fehlertoleranz, da die Knoten in mehreren Ebenen routen können.
- Nachteile:
 - größere Routingtabellen

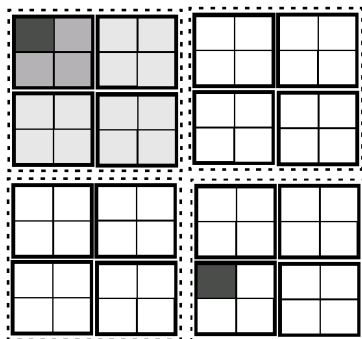
Expressways

Das bisher vorgestellte CAN hat beim Routing eine durchschnittliche Pfadlänge von $O(n^{1/d})$. Expressways bieten eine Möglichkeit dies bis auf $O(\log_K(n))$ zu verbessern.

Idee

- Optimierung des Routings durch Routingtabellen mit einer größeren Ausdehnung
- Der Koordinatenraum wird in Zonen mit unterschiedlichen Größen aufgeteilt
- Die kleinsten Zonen sind die Bereiche die wir aus dem CAN kennen
- Die größeren Zonen bilden die Expressway-Zonen.
- Jeder Knoten gehört als zu einer CAN-Zone und liegt gleichzeitig auch in Expressway-Zonen.

Idee



- CAN Zonen liegen auf Ebene 3 und machen je $1/64$ des gesamten Koordinatenraums aus
- 4 CAN Zonen ergeben dann eine Ebene 2 Expressway-Zone
- 4 Ebene 2 Expressway-Zonen bilden eine Ebene 1 Expressway-Zone

Aufbau der Routingtabelle

Wir unterscheiden zwischen der "Standard-Routingtabelle" und der "vollständigen Routingtabelle".

Die "Standard-Routingtabelle" entspricht der Routingtabelle, die das CAN aufbaut.

Die "vollständige Routingtabelle" vereint die "Standard-Routingtabelle" und die Routingtabellen der Expressways.

Aufbau der Routingtabelle

$$R_T = \langle R_0, R_1, \dots, R_L \rangle$$

- R_i ($i = 0$ bis $i = L - 1$): Expressway-Routingtabellen.
 - Je kleiner das i desto größer die Expressway-Zone
 - R_i beinhaltet die i . größte Zone, die den Knoten einschließt und die Nachbarzonen der gleichen Größe.
- R_L : Standard-Routingtabelle des entsprechenden Knoten.

Aufbau des Expressways

Die Expressways müssen dynamische aufgebaut werden, wenn das Netz wächst. Hierzu wird der "envolving snapshot"-Algorithmus verwendet.

"envolving snapshot"-Algorithmus

Unter $x.R_i.Z$ verstehen wir die Zone Z der Routingtabelle R auf Ebene i des Knoten x .

- Wenn ein neuer Knoten das Netz betritt, übernimmt er die Routingtabelle des Knoten, von dem er ein Teil des Koordinatenraums zugeteilt bekommt, bis auf $x.R_L$, $x.R_L$ wird nach den normalen Regeln des CAN erzeugt.
- Beide Knoten überprüfen dann, ob $x.R_L \leq x.R_{L-1}.Z/K$
 - Ist das der Fall, wird ein "Snapshot" erzeugt (es entsteht eine neue Expressway Zone): L wird um eins erhöht, $x.R_{L-1}$ wird auf $x.R_L$ kopiert.
 - Ist dieser Fall noch nicht erreicht, werden lediglich die CAN-Routingtabellen $x.R_L$ angepasst.

"involving snapshot"-Algorithmus

Procedure for a node y joins node x $y.R_T = \langle x.R_0, \dots, x.R_{L-1}, y.R_L \rangle$

Repeat procedure for testing for new snapshot

Procedure for testing for new snapshot// executed by both x and y If $(R_L.Z \leq R_{L-1}.Z/K)$ { $R_{L+1} = R_L$ $R_T = \langle R_0, R_1, \dots, R_L, R_{L+1} \rangle$ $L = L+1$

}

Abbildung: Algorithmus, wenn ein neuer Knoten das System betritt

Routing

Route with Expressway

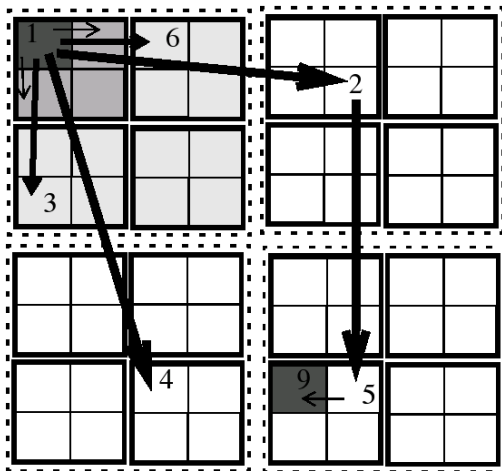
```
If (pt ∈ RL.Z) Return;  
For (i=0; i ≤ L; i++)  
  If (pt ∉ (Ri.Z) )  
    Route using Ri;
```

Route with R_i

```
For (j=0; j < d; j++)  
  If (pt < Ri.Z.Lj || pt > Ri.Z.Uj) {  
    Route to x ∈ Ri.Nj that is closest to pt  
    Break;  
  }
```

Abbildung: Routing Algorithmus

Routing



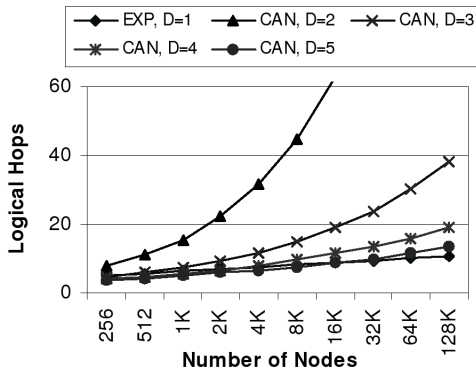
Zusammenfassung

Content-Addressable Networks sind:

- Vollständig verteilt: Es benötigt keinerlei zentrale Steuerung oder Konfiguration
- Skalierbar: Knoten verwalten nur eine kleine Menge an Information und das CAN kann jederzeit erweitert oder verkleinert werden.
- Fehlertolerant: Es gibt immer mehrere Wege zum Ziel, so dass Fehler umgangen werden können.

Zusammenfassung

Abschließender Vergleich der vorgestellten Optimierungen:



Fragen?

This work is licensed under the
Creative Commons Attribution-Share Alike 3.0 License.

To view a copy of this license, visit

<http://creativecommons.org/licenses/by-sa/3.0/deed>

or send a letter to

Creative Commons, 559 Nathan Abbott Way, Stanford, California
94305, USA.