

Organic Computing Peer-to-Peer-Netzwerke

Rolf Wanka

Sommersemester 2008

rwanka@cs.fau.de



Inhalte

- Kurze Geschichte der Peer-to-Peer-Netzwerke
- Das Internet: Unter dem Overlay
- Die ersten Peer-to-Peer-Netzwerke
 - Napster
 - Gnutella
- CAN
- Chord
- Pastry
- **Gradoptimierte Netzwerke**
 - Viceroy
- Netzwerke mit Suchbäumen
 - Skipnet und Skip-Graphs
 - P-Grid
- Selbstorganisation
 - Pareto-Netzwerke
 - Zufallsnetzwerke
- Sicherheit in Peer-to-Peer-Netzwerken
- Anonymität
- Datenzugriff: Der schnellere Download
- Peer-to-Peer-Netzwerke in der Praxis
 - eDonkey
 - FastTrack
 - Bittorrent
- Peer-to-Peer-Verkehr
- Juristische Situation



Viceroy

A Scalable and Dynamic Emulation of the Butterfly

Dahlia Malkhi, Moni Naor, David Ratajczak

Tagungsband

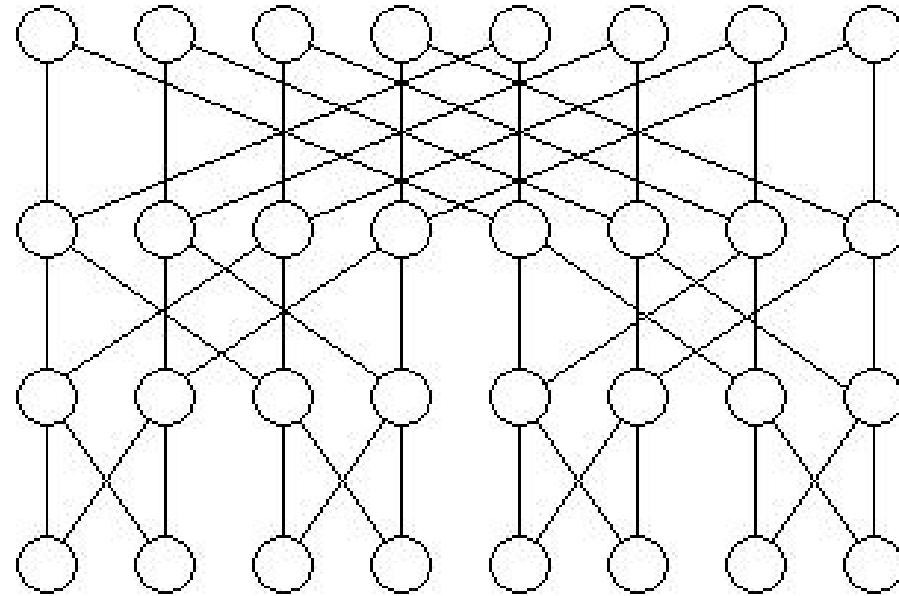
21st Symposium on **P**inciples of **D**istributed **C**omputing,
PoDC 2001, S. 183 - 192.

Erreichbarer Durchmesser

- CAN:
 - Durchmesser $O(n^{1/2})$
- Gesucht:
 - Netzwerk mit kleinem Ausgangsgrad
 - D.h. Eingangsgrad, Ausgangsgrad konstant, $O(1)$
 - Durchmesser $O(\log n)$
- Eine mögliche Lösung:
 - Viceroy

Definition Butterfly-Graph (mit wrap-around-Kanten)

- Knoten: (i, S)
 - i ist aus $\{0, \dots, k-1\}$
 - S ist k -stelliger Binärstring
- Interpretation
 - $m = 2^k$ Knoten pro Ebene
 - k Ebenen
 - In der Regel werden die Knoten der k -ten Ebene zweimal gezeichnet
- Kanten: Von $(i, (b_0, \dots, b_i, \dots, b_{k-1}))$ nach
 - $((i+1) \bmod k, (b_0, \dots, b_i, \dots, b_{k-1}))$ und
 - $((i+1) \bmod k, (b_0, \dots, 1-b_i, \dots, b_{k-1}))$
- $n = k 2^k$ Knoten



Eigenschaften Butterfly-Graph

- Kleiner Grad
 - Eingangsgrad + Ausgangsgrad = 4
- Kleiner Durchmesser
 - mit $\log m = \log n + \log \log n$ optimal
- Gute Simulationseigenschaften
 - Andere Netzwerke können sehr effizient in einen Butterfly-Graphen eingebettet werden
 - D.h. Eine Kante eines anderen Netzwerks wird durch sehr kurze Pfade im Butterfly-Graph ersetzt
- Einfache Routing-Algorithmen
 - Routing-Entscheidung in konstanter Zeit möglich
- Kaum Flaschhälse
 - Mit hoher W'keit: Kein Knoten wird beim Routing mit Nachrichten überlastet
- Hohe Fehlertoleranz
 - Ausfall von Knoten kann gut toleriert werden

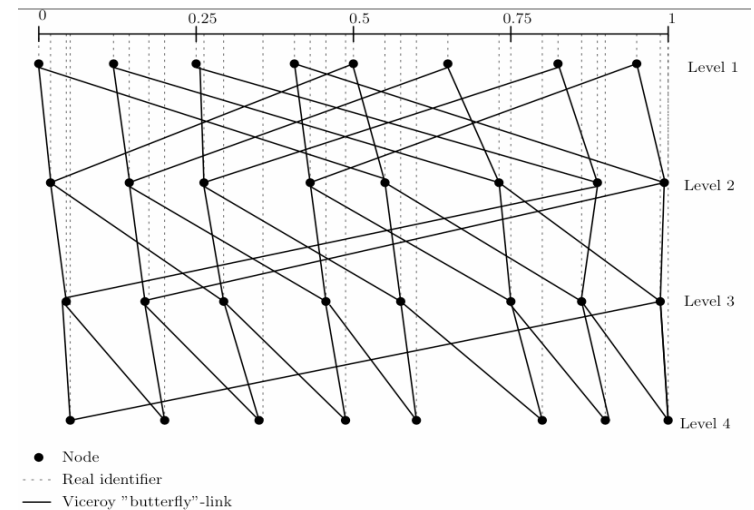
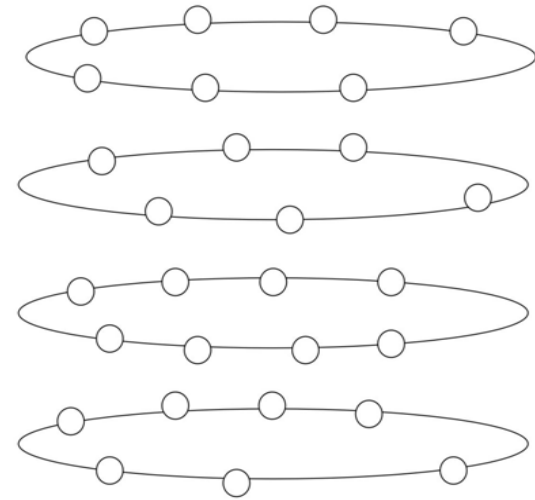


Übersicht Viceroy

- Zielsetzung
 - Skalierbarkeit
 - Bewältigung von Dynamik
 - Verteilung des *Traffic*
- *Congestion*
 - Maximale Anzahl Nachrichten, die ein Peer zu bewältigen hat
- Kosten für Peer-Aufnahme/Entfernen
 - Minimierung Anzahl Nachrichten/Zeit
- *Dilation*, Länge des Suchpfads
- Viceroy war das erste P2P-Netzwerk mit optimaler Grad/Durchmesser-Beziehung
 - Techniken lassen sich auf andere Netzwerke übertragen

Aufbau Viceroy

- Knoten in Viceroy
 - Wählen anfangs zufällig eine Ebene des Butterfly-Netzwerks
 - (Hierzu ist es notwendig, $\log n$ zu kennen)
- Kombination aus drei Netzwerk-Strukturen
 1. Ein Ring für alle Knoten
 - Verkettet alle Knoten
 2. Für jede Ebene ein Ring
 - mit dem Intervall $[0,1)$
 3. Das Butterfly-Netzwerk zwischen den Ebenen,
 - d.h. in Ebene i gibt es Links von (i,x) zu
 1. Nachfolger von $(i+1,x)$
 2. Nachfolger von $(i+1,x+2^i)$
 3. Vorgänger von $(i-1,x)$
 4. Vorgänger von $(i-1,x-2^i)$



Schätzung von $\log n$

- Betrachte Ringnachbar auf dem „Ring für alle“
- Sei d der Abstand auf dem Ring $[0,1)$
- Dann gilt
 - $E[d] = 1/n$
 - $\Pr[d > c (\log n)/n] < n^{-O(1)}$
 - $\Pr[d < 1/n^{1+c}] < n^{-c}$
- Damit ist $\log(1/d)$ mit hoher Wahrscheinlichkeit eine $(1+\varepsilon)$ -Approximation von $\log n$

Einfügen eines Peers

1. Füge Knoten an zufälliger Stelle des „Rings für alle“ ein
 2. Schätze $\log n$ ab durch Messen des Abstands zweier Peers
 3. Wähle zufällige Ebene i uniform aus $[1, \dots, \log n]$ und $x \in [0, 1)$
 4. Suche Stelle im ViceRoy-Netzwerk durch LookUp ausgehend vom Ringnachbarn
 5. Füge Peer in die Ebene i des ViceRoy-Netzwerks, indem
 - der Peer in den Ring i des Netzwerks eingefügt wird
 - die Ausgangszeiger von (i, x)
 - Nachfolge-Peer der Position $(i+1, x)$
 - Nachfolge-Peer der Position $(i+1, x+2^i)$
 - Vorgänger-Peer der Position $(i-1, x)$
 - Vorgänger-Peer der Position $(i-1, x-2^i)$ausgehend von den Zeigern des Nachbarn im Ring i gesucht werden
- Laufzeit/Anzahl Nachrichten
 - Zeit von Lookup ($O(\log n)$) +
 - Finden der Nachfolger/Vorgänger ($O(\log n)$)

Suche

- Peer (i,x) erhält Suchanfrage nach (j,y)
 - IF $i=j$ und $|x-y| \leq (\log n)^2/n$ THEN
 - Leite Suchanfrage auf Ringnachbar der Ebene i weiter
 - ELSE
 - IF y weiter rechts als $x+2^i$ THEN
 - Leite Suchanfrage an Nachfolger von $(i+1,x+2^i)$ weiter
 - ELSE
 - Leite Suchanfrage an $Z =$ Nachfolger von $(i+1,x)$ weiter
 - IF Nachfolger Z weiter rechts als x THEN
 - Suche auf dem Ring $(i+1)$ von Z aus einen Knoten $(i+1,p)$ mit $p < x$

Lemma

Mit hoher Wahrscheinlichkeit benötigt Suche $O(\log n)$ Zeit und Nachrichten



Eigenschaften von ViceRoy

- Ausgangsgrad konstant
- Erwarteter Eingangsgrad konstant
- Durchmesser $O(\log n)$
- Durch den Butterfly-Graphen kann die Kommunikationslast balanciert werden
 - In den Butterfly-Graphen können beliebige andere Graphen effizient eingebettet werden

Die Sache mit dem Grad

- Ausgangsgrad: $2+2+2+2 = 8$
- Bei perfekter Verteilung, d.h. Abstand zum Nachbarn $\Theta(1/n)$
 - Eingangsgrad konstant
- Tatsächlich
 - Eingangsgrad: erwartet konstant
 - aber auch $\Omega(\log n / \log \log n)$ kommt mit Wahrscheinlichkeit $1-n^{-1}$ vor
- Problem:
 - Große Abstände auf dem Viceroy-Ring ziehen viele Links an
 - Kleine Abstände bombardieren die Ringnachbarn
- Lösung:
 - Ordne die Peers **nicht** durch Hashing zu
 - Verwende das Prinzip der mehrfachen Auswahl (multiple choice).

Das Prinzip der mehrfachen Auswahl

- Beim Einfügen würfelt jeder Peer $c \log n$ Positionen
- Für jede Position $p(j)$ wird die Abstand $a(j)$ zwischen potentiellen linken und rechten Nachbar gemessen
- Peer wird an einer Position $p(j)$ in der Mitte zwischen linken und rechten Nachbarn eingefügt, wo $a(j)$ maximal in der Auswahl ist

Lemma:

Mit hoher Wahrscheinlichkeit ist der Abstand zweier Peers in den $\log n$ Viceroy-Ringen nur einen konstanten Faktor größer oder kleiner als der durchschnittliche Abstand $(\log n)/n$.



Beweis der Korrektheit des Prinzips der mehrfachen Auswahl

Lemma

Nach dem Einfügen von n Peers auf einem Ring $[0,1)$, wobei das Prinzip der mehrfachen Auswahl angewendet wird, bleiben nur Intervalle der Größe $1/(2n)$, $1/n$ und $2/n$ übrig.

1. Teil: *Mit hoher W'keit gibt es kein Intervall länger als $2/n$:*

Beweis folgt aus folgenden Lemma:

Lemma **

Sei das längste Intervall c/n (c kann von n abhängen).
Dann sind nach Einfügen von $2n/c$ Peers alle Intervalle kürzer als $c/(2n)$ mit hoher W'keit.

Wendet man dieses Lemma für $c=n/2, n/4, \dots, 4$ an, folgt der erste Teil des Lemmas

Beweis der Korrektheit des Prinzips der mehrfachen Auswahl

Lemma

Nach Einfügen von n Peers auf einem Ring $[0,1)$, wobei das Prinzip der mehrfachen Auswahl angewendet wird, bleiben nur Intervalle der Größe $1/(2n)$, $1/n$ und $2/n$ übrig.

Beweis

2. Teil: Keine Intervalle kleiner als $1/(2n)$ entstehen

Die Gesamtlänge der Intervalle der Größe $1/(2n)$ ist vor jedem Einfügen höchstens $n/2$

Ein solches Intervall wird mit Wahrscheinlichkeit höchstens $1/2$ gewählt

Die Wahrscheinlichkeit, dass unter $c \log n$ Intervallen nur solche gewählt werden, ist

$$2^{-c \log n} = n^{-c}$$

Damit wird für $c > 1$ mit polynomiell kleiner Wahrscheinlichkeit ein Intervall der Länge $1/(4m)$ weiter unterteilt.

Chernov-Schranke

- Bernoulli-Experiment
 - Entweder 0 mit Wahrscheinlichkeit $1-p$
 - Oder 1 mit Wahrscheinlichkeit p

Theorem 3 Chernoff-Schranke

Seien x_1, \dots, x_n unabhängige Bernoulli-Experimente mit $\mathbf{P}[x_i = 1] = p$ und $\mathbf{P}[x_i = 0] = 1 - p$. Sei $S_n = \sum_{i=1}^n x_i$. $\mathbf{E}[S_n] = pn$

1. Dann gilt für jedes $c \in [0, 1]$:

$$\mathbf{P}[S_n \geq (1 + c)\mathbf{E}[S_n]] \leq e^{-\frac{c^2}{3}pn} .$$

2. Für $c \in [0, 1]$:

$$\mathbf{P}[S_n \leq (1 - c)\mathbf{E}[S_n]] \leq e^{-\frac{c^2}{2}pn} .$$

Beweis von Lemma (**)

Sei das längste Intervall c/n (c kann von n abhängen). Dann sind nach Einfügen von $2n/c$ Peers alle Intervalle kürzer als $c/(2n)$ mit hoher W'keit.

- Betrachte ein Intervall der Länge c/n
- Mit W'keit c/n wird ein solches Intervall von einem Peer getroffen
- Angenommen es werden für jeden Peer $t \log n$ Intervalle untersucht
- Die erwartete Anzahl von Treffern in diesem Intervall ist also

$$E[X] = \frac{c}{n} \cdot \frac{2n}{c} \cdot t \log n = 2t \log n$$

- Die Chernov-Schranke liefert dann:

$$P[X \leq (1 - \delta)E[X]] \leq n^{-\delta^2 t}$$

- Ist $\delta^2 t \geq 2$, werden alle diese Intervalle mindestens $2(1 - \delta)t \log n$ -mal getroffen
- Jedesmal, wenn ein Intervall geteilt wird, gab es $t \log n$ (zusätzliche) Treffer
- Wir wählen $2(1 - \delta) \geq 1$
- Dann wird jedes Intervall (mit hoher W'keit) der Länge c/n geteilt.

Resumee ViceRoy

- Butterfly-Graph
 - für Routing sehr gut geeignet
 - viele erprobte, gute Algorithmen bekannt
- Erstes Peer-to-Peer-Netzwerk mit konstantem Ein- und Ausgangsgrad
- Aber:
 - Mehrfache Ringstruktur relativ kompliziert
 - Durch das Prinzip der mehrfachen Auswahl ist der Aufwand zum Einfügen $O((\log n)^2)$
 - Das Prinzip der mehrfachen Auswahl kann auch bei DeBruijn-Netzwerken angewendet werden
 - Diese sind wesentlich einfacher
 - Haben die gleichen Netzwerkeigenschaften

Ende

