VerteilteWebInf Hausaufgabe 7

Gruppe 6

November 26, 2014

Aufgabe 1

a) Annahme: Es wären zwei Joinpartner $r \in R$ und $s \in S$ in unterschiedlichen Partitionen $\mathcal{R}_i$ und $\mathcal{R}_j$.
Dann gilt $h(r.A) mod\ p \neq h(s.B) mod\ p$ bei gleicher Hashfunktion h, also gilt $r.A \neq s.B$, was ein Widerspruch dazu ist, dass r und s Joinpartner sind.

b) unter der Annahme, dass jedem Knoten genau eine Partition zugeteilt wird muss Knoten i
$\frac{p-1}{p} \mid \mathcal{R}_i \mid + \frac{p-1}{p} \mid \mathcal{S}_i \mid$ Tupel verschicken.
Zu addieren sind noch die Ergebnistupel aus dem Join von $\mathcal{R}_{j,i}$ und $\mathcal{R}_{j,i} \forall j$, die gegebenenfalls an eine andere Station gesendet werden müssen.

c) Bei einem Join mit Broadcast verschickt jeder Knoten $\mid \mathcal{R} \mid$ Tupel.
Zu addieren sind noch die Ergebnistupel aus dem Join von $\mathcal{R}$ und $\mathcal{S}_i$, die gegebenenfalls an eine andere Station gesendet werden müssen.

d) $\mathcal{R} < \frac{p-1}{p} \mid \mathcal{R}_i \mid + \frac{p-1}{p} \mid \mathcal{S}_i \mid$
Annahme: R und S sind gleichmäßig auf die p Knoten verteilt (d.h. die Partitionen sind alle gleich groß)
$\mathcal{R} < \frac{p-1}{p} \cdot \frac{1}{p} \cdot \mid \mathcal{R} \mid + \frac{p-1}{p} \cdot \frac{1}{p} \cdot \mid \mathcal{S} \mid$
$\iff \mathcal{R} < \frac{p-1}{p^2-p-1} \mid \mathcal{S} \mid$

e) Alle Knoten $i \neq k$ schicken gleichzeitig D Daten an Knoten k. Dieser empfängt mit Bandbreite b.
Die gesamte Zeit, bis Knoten k alle Daten vollständig erhalten hat, ist somit $t_{vj} = \frac{(n-1)D}{b}$.

f) Die Knoten müssen jetzt insgesamt $p(n-1) = n(n-1)$ Partitionen an Knoten k verschickt werden, also: $t = n\frac{(n-1)D}{b}$. Es dauert also n mal länger als beim verteilten Join.

g) Knoten 1 muss jetzt doppelt so viele Daten senden, die Zeit ingesamt ergibt sich dann zu $t = (n-2)\frac{D}{b} + \frac{2D}{b}$

h) Wenn $p > n$ und keine redundanten Partitionen vorhanden sind, muss jeder Knoten u.U. nicht alle seine Tupel senden, sondern es reicht, eine seiner Partitionen zu senden.
Wenn $p > n$ und es vorkommt, dass Partitionen redundant auf verschiedenen Knoten vorhanden sind, dann kann bei einem Join ein Knoten, der seine Partition an Knoten k senden soll, aus allen Knoten mit der benötigten Partition ausgewählt werden. Z.B. kann der Knoten, der am nächsten zum Knoten k liegt, ausgewählt werden, oder der Knoten, der am wenigsten ausgelastet ist.

Aufgabe 2

a) *min_data.lp*

- Partition 0: Knoten 0
- Partition 1: Knoten 0
- Partition 2: Knoten 3
- Partition 3: Knoten 0
- Partition 4: Knoten 1
- Partition 5: Knoten 0
- Partition 6: Knoten 0
- Partition 7: Knoten 0
- Partition 8: Knoten 0
- Partition 9: Knoten 0
- Partition 10: Knoten 0
- Partition 11: Knoten 0
- Partition 12: Knoten 0
- Partition 13: Knoten 0
- Partition 14: Knoten 0
- Partition 15: Knoten 0

b) *min_antwortzeit.lp*

- Partition 0: Knoten 3
- Partition 1: Knoten 0
- Partition 2: Knoten 3
- Partition 3: Knoten 2
- Partition 4: Knoten 1
- Partition 5: Knoten 2
- Partition 6: Knoten 2
- Partition 7: Knoten 3
- Partition 8: Knoten 0
- Partition 9: Knoten 0
- Partition 10: Knoten 0
- Partition 11: Knoten 1
- Partition 12: Knoten 1
- Partition 13: Knoten 0
- Partition 14: Knoten 0
- Partition 15: Knoten 3

c) *Aufgabe a*:
gesamte Übertragungskosten: 313
Übertragungszeit: 264
Aufgabe b:
gesamte Übertragungskosten: 328

Übertragungszeit: 95
insgesamt sieht man also, dass bei Aufgabe a) die gesamten Übertragungskosten minimiert sind, während bei b) die Übertragungszeit minimiert wurde.
Besonders auffällig ist hierbei, dass bei nur geringer höheren Gesamtübertragungskosten die Übertragungszeit dramatisch sinkt, weshalb in diesem Fall vermutlich b) zu bevorzugen wäre.