# Bestimmen der optimalen Joinreihenfolge mit Dynamischem Programmieren

10. Dezember 2019

Wir betrachten die Relationen $R$,$S$ und $T$ sowie den Join $R \bowtie S \bowtie T$; dabei haben $R$ und $T$ keine gemeinsamen Attribute. Weiterhin sind folgende Parameter bekannt:

- $|R| = 10$
- $|S| = 20$
- $|T| = 10$
- $|R \bowtie S| = 4$ (also $SF_J(R, S) = 0.02$)
- $|S \bowtie T| = 100$ (also $SF_J(S, T) = 0.5$)
- $|R \bowtie T| = 100$ (also $SF_J(R, T) = 1$)
- $|R \bowtie S \bowtie T| = 20$ ($= |R| \cdot |S| \cdot |T| \cdot SF_J(R, S) \cdot SF_J(S, T)$)

Die Kosten für den Join der Relationen $P$ und $Q$ seien $|P| \cdot |Q|$.

## 1 Bestimmen der optimalen Joinreihenfolge in zentralen Systemen

In diesem Fall müssen wir nur die beste Joinreihenfolge finden; der Ort der Ausführung ist klar, da es nur einen Rechner gibt, also keine Kommunikationskosten berücksichtigt werden müssen. In der DP-Tabelle merken wir uns also für jeden Teiljoin seine optimalen Kosten, für deren Berechnung wir alle Möglichkeiten betrachten, den Teiljoin aus bereits berechneten kleineren Joins zusammenzustellen. Zusätzlich berechnen wir für alle Joins, die nicht bereits alle Relationen beinhalten, die Größe des Zwischenergebnisses.

| Join-Baum | Größe d. Zw.-erg. | minimale Kosten |
|---|---|---|
| $R \bowtie S$ | 4 | $10 \cdot 20 = 200$ |
| $S \bowtie T$ | 100 | $10 \cdot 20 = 200$ |
| $R \bowtie T$ | 100 | $10 \cdot 10 = 100$ |
| $(R \bowtie S) \bowtie T$ | - | $200 + 4 \cdot 10 = 240$ |
| $(S \bowtie T) \bowtie R$ | - | $200 + 100 \cdot 10 = 1200$ |
| $(R \bowtie T) \bowtie S$ | - | $100 + 100 \cdot 20 = 2100$ |

Die beste Joinreihenfolge ist also $(R \bowtie S) \bowtie T$.

# 2 Bestimmen der optimalen Joinreihenfolge in verteilten Systemen

Im verteilten Fall müssen wir die möglicherweise notwendige Kommunikation der Daten bei der Kostenberechnung berücksichtigen. Wir nehmen dazu an, dass es drei Rechner $N_R$, $N_S$ und $N_T$ gibt. Weiterhin sei $R$ auf $N_R$ alloziert, $S$ auf $N_S$, und $T$ auf $N_T$. Alle Rechner können untereinander kommunizieren, und die Kosten, um ein Tupel von Rechner $A$ nach Rechner $B$ zu übertragen, seien 1. Wir nehmen an, dass die Kommunikationskosten die Kosten für die Joinberechnung dominieren (WAN-Szenario aus der Vorlesung). Die Kosten für einen Join bestehen daher in unserem Kostenmodell ausschließlich aus ggf. erforderlichen Kommunikationskosten. In der DP-Tabelle müssen wir nun für jeden Teiljoin alle Alternativen betrachten, wo dieser Join ausgeführt werden kann. Dies führt zu neuen Möglichkeiten, einen Join aus bereits berechneten Teiljoins zusammenzusetzen.

In der DP-Tabelle bezeichnet $\rightarrow$ das Versenden von Daten, z.B. bedeutet $B \rightarrow N$, dass Tabelle $B$ an Rechner $N$ geschickt wird.

Beachten Sie: Für eine konkrete Anfrage ist bekannt, auf welchem Rechner am Ende das Ergebnis verfügbar sein soll. Daher wird üblicherweise für Joins aller Relationen nur die Spalte des Zielrechners ausgefüllt.

| Join-Baum | Größe d. Zw.-erg. | Minimale Kommunikationskosten mit Ergebnis in | | |
|---|---|---|---|---|
| | | $N_R$ | $N_S$ | $N_T$ |
| $R \bowtie S$ | 4 | 14 ($R \rightarrow N_S$, $R \bowtie S \rightarrow N_R$) | 10 ($R \rightarrow N_S$) | 14 ($R \rightarrow N_S$, $R \bowtie S \rightarrow N_T$) |
| $S \bowtie T$ | 100 | 30 ($S, T \rightarrow N_R$) | 10 ($T \rightarrow N_S$) | 20 ($S \rightarrow N_T$) |
| $R \bowtie T$ | 100 | 10 ($T \rightarrow N_R$) | 20 ($R, T \rightarrow N_S$) | 10 ($R \rightarrow N_T$) |

| Join-Baum | Größe d. Zw.-erg. | Minimale Kommunikationskosten mit Ergebnis in | | |
|---|---|---|---|---|
| | | $N_R$ | $N_S$ | $N_T$ |
| $(R \bowtie S) \bowtie T$ | - | **24** ($R \rightarrow N_S$, $R \bowtie S \rightarrow N_R$, $T \rightarrow N_R$) | **20** ($R, T \rightarrow N_S$) | **14** ($R \rightarrow N_S$, $R \bowtie S \rightarrow N_T$) |
| $(S \bowtie T) \bowtie R$ | - | 30 ($S, T \rightarrow N_R$) | **20** ($R, T \rightarrow N_S$) | 30 ($R, S \rightarrow N_T$) |
| $(R \bowtie T) \bowtie S$ | - | 30 ($S, T \rightarrow N_R$) | **20** ($R, T \rightarrow N_S$) | 30 ($R, S \rightarrow N_T$) |



Wenn wir beispielsweise den Teiljoin $R \bowtie S$ so berechnen wollen, dass das Ergebnis auf $N_R$ verfügbar ist, können wir entweder $S$ mit Kosten 20 auf Rechner $N_R$ übertragen und dort den Join ausführen, oder $R$ mit Kosten 10 auf Rechner $N_S$ übertragen, dort den Join ausrechnen und das Ergebnis mit Kosten 4 auf Rechner $N_R$ übertragen. Letztere Methode ist billiger und wird daher an der entsprechenden Stelle in der DP-Tabelle eingetragen.

Für die Berechnungsreihenfolge $(R \bowtie S) \bowtie T$ mit Ergebnis auf $N_R$ ergeben sich folgende Möglichkeiten:

- Berechnen von $R \bowtie S$ mit Ergebnis auf $N_R$ (optimale Kosten 14), Transport von $T$ nach $N_R$ (Kosten 10), dort Join; Gesamtkosten 24.
- Berechnen von $R \bowtie S$ mit Ergebnis auf $N_S$ (optimale Kosten 10), Transport von $T$ nach $N_S$ (Kosten 10), dort Join, Transport des Ergebnisses nach $N_R$ (Kosten 20); Gesamtkosten 40.
- Berechnen von $R \bowtie S$ mit Ergebnis auf $N_T$ (optimale Kosten 14), lokaler Join mit $T$, Transport des Ergebnisses nach $N_R$ (Kosten 20); Gesamtkosten 34.

Die erste Joinreihenfolge ist also in diesem Fall die beste.