

Aufgabenblatt 6: Webscale Data Management

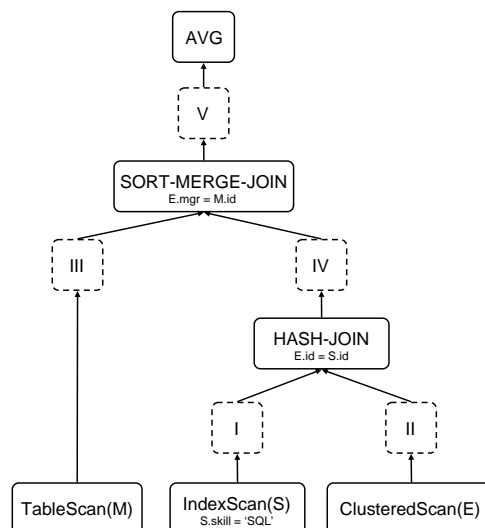
- Abgabetermin: **Dienstag, 09.02.10 (23:59 Uhr)**
- Zur Prüfungszulassung muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte.
- Die Aufgaben sollen in Zweiergruppen bearbeitet werden.
- Abgabe:
 - per E-Mail an dbs2-200910@hpi.uni-potsdam.de mit Subject
Abgabe DBS II: Aufgabenblatt <n> Namen
 - ausschließlich pdf-Dateien
 - eine Datei pro Aufgabe mit folgendem Dateinamen:
blatt<aufgabenblattNr>aufgabe<aufgabenNr><Nachnamen>.pdf
Bitte **keine Leerzeichen, Unterstriche, Umlaute, Sonderzeichen, ...** im Dateinamen!
 - **jedes Blatt beschriftet mit Namen**
 - Wir korrigieren die Abgaben aufgabenweise. Das beschriebene Verfahren vereinfacht uns die Arbeit erheblich!

Aufgabe 1: Parallel Query Processing

Die folgende Anfrage bestimmt das Durchschnittseinkommen aller SQL-Experten, die mehr als ihr Chef verdienen:

```
SELECT AVG(E.sal)
FROM Employee E, Employee M, Skills S
WHERE E.id = S.id AND E.mgr = M.id AND E.sal > M.sal AND S.skill = 'SQL'
```

Die Abbildung zeigt den Plan für diese Anfrage ohne die nötigen Sortieroperationen.



- Vervollständige den Ausführungsplan mit möglichst wenig Sortieroperationen. Füge dafür an den gekennzeichneten Stellen (I-V), wenn nötig, Sortieroperationen ein. Begründe deine Entscheidung für jede der fünf Positionen. Es ist bekannt, dass der TableScan keine sortierte Ausgabe liefert. Die Ausgabe des Clustered-Scan ist nach dem Schlüssel des Index, E.id, sortiert. **5 P**
- Markiere im vervollständigten Ausführungsplan alle Pipelines und die blockierenden Operatoren. Begründe, zwischen welchen der Operatoren Pipeline-Parallelismus möglich ist. **5 P**

Aufgabe 2: MapReduce

Ein optisches Meßverfahren erfasst die Oberfläche von Objekten als Menge von 3D-Ortsvektoren $(x, y, z)^T$ und speichert diese in der Datenbanktabelle $S(\underline{vectorId}, \underline{dimension}, \underline{value})$ ab. Durch Störeinflüsse werden beim Messen nicht alle Ortsvektoren vollständig erfasst.

Für eine grafische Simulation werden nun die Vektorlängen $length$ aller **vollständig erfasster** Ortsvektoren benötigt, $length = \sqrt{x^2 + y^2 + z^2}$. Aufgrund der großen Datenmenge soll die Berechnung auf einem MapReduce-Cluster erfolgen.

- Erzeuge als Ausgabe eine Tabelle $T(\underline{vectorId}, \underline{length})$, in der für jeden vollständig erfassten Ortsvektor die Vektorlänge steht. Die Ausgabe wird partitioniert auf den verschiedenen Nodes ins verteilten Dateisystem geschrieben. Löse die Aufgabe mit nur einem MapReduce-Job. Verwende auch die in der Vorlesung vorgestellte Funktion *combine*, um Netzwerklast zu reduzieren. Beschreibe dein Vorgehen kurz in wenigen Sätzen. Erstelle den Pseudocode für *map*, *combine* und *reduce*. **12 P**
- Die Abbildung zeigt beispielhaft eine Verteilung von S auf drei MapReduce-Nodes. Zeige für dieses Beispiel die Ausgabe der einzelnen Phasen deines MapReduce-Jobs auf jeder der drei Nodes. **5 P**

<i>id</i>	<i>dim</i>	<i>val</i>
1	x	2
2	z	4
1	y	3
3	y	2
4	y	4

<i>id</i>	<i>dim</i>	<i>val</i>
3	z	4
4	z	2
2	x	2
2	y	4

<i>id</i>	<i>dim</i>	<i>val</i>
5	y	3
3	x	4
4	x	4

Aufgabe 3: Pig Latin

Gegeben sei folgendes Pig Latin Skript:

```
S1 = LOAD 's.txt' USING PigStorage('\t') AS (A,B);
T1 = LOAD 't.txt' USING PigStorage('\t') AS (C,D);
S2 = FILTER S1 BY A > 20;
T2 = FILTER T1 BY D < 40;
U1 = COGROUP S2 BY B, T2 BY C;
U2 = FOREACH U1 GENERATE $0, SUM(S2.A) + SUM(T2.D);
```

- Erzeuge die Ausgabe für die folgenden Eingabetabellen $S1$ und $T1$. Bitte dokumentiere deine Zwischenschritte ($S2$, $T2$, $U1$, $U2$). **5 P**

<i>A</i>	<i>B</i>
30	1
50	3
15	4
40	3
60	1

<i>C</i>	<i>D</i>
3	0
1	20
2	100
3	10
3	30
1	45

- Ersetze das Pig Latin Skript mit genau einem MapReduce-Job. Beschreibe dein Vorgehen kurz in wenigen Sätzen und erstelle den Pseudocode. Die Funktion *map* erhält als Eingabe die Tupel beider Tabellen, wobei der übergebene Schlüssel die jeweilige Tabelle, $S1$ oder $T1$, identifiziert. **8 P**