

## Aufgabenblatt 3 Benchmarking

- Abgabetermin: **Dienstag, 08.12.2015 (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 Aufgabe soll in Zweiergruppen bearbeitet werden.
- Abgabesystem unter  
`http://www.dcl.hpi.uni-potsdam.de/submit`
  - eine Zip-Datei mit den erforderlichen Dateien
  - beschriftet mit Namen

### Aufgabe 1: Hashtable Index vs. Full Table Scan

Implementiere die folgenden beiden Datenstrukturen in Java. Die Daten sollen zur Vereinfachung ausschließlich im Hauptspeicher gehalten werden. Es sollen Datensätze mit zwei Attributen gespeichert werden – einem Schlüssel („key“, Integer) und einem zugehörigen Wert („value“, String). 32 Datensätze werden zu einem Block zusammengefasst.

- Variante 1: Organisiere die Daten mit Hilfe eines Hashtable Index mit 256 Buckets. Der vollständige Datensatz soll im entsprechenden Bucket (ein Block plus Overflow-Blöcke) gespeichert werden. Verwende als Hashfunktion  $h$  für den key  $k$  die Funktion  $h(k) = k \bmod |\text{Buckets}|$ .
- Variante 2: Die Datensätze werden in der Einfüge-Reihenfolge in eine Sequenz von Blöcken gespeichert.

Die zu speichernden Datensätze sind in der Datei DatenDump.txt auf

```
R:\lehrveranstaltungen\FG_Informationssysteme\VL DBS II\uebung benchmarking
```

gegeben. Jede Zeile entspricht einem Datensatz, die Attribute sind durch ein Leerzeichen getrennt.

- Lies die Datensätze ein und suche die Werte zu den folgenden Schlüsseln: 371018, 395689, 668010, 175587, 175585. Gib jeweils die Anzahl der gelesenen Blöcke und die Anzahl der verglichenen Schlüsselwerte an. **10 P**
- Wie sind die Daten auf die Buckets in Variante 1 verteilt, d.h. wieviele Blöcke sind (durchschnittlich, minimal, maximal) in einem Bucket. Entspricht das deiner Erwartung? Warum (nicht)? **5 P**
- Was ist die Gesamtanfragezeit für alle unter a) gesuchten Werte für beide Speichervarianten? Entspricht das deiner Erwartung? Warum (nicht)? Wie würde sich das Verhältnis zwischen beiden Laufzeiten verändern, wenn die Blöcke im Sekundärspeicher gehalten würden? **5 P**

Hinweise:

- Implementiere die beschriebenen Datenstrukturen und die Anfragen in Java 1.7 oder höher.
- Verwende für Blöcke das unter  
`R:\lehrveranstaltungen\FG_Informationssysteme\VL DBS II\uebung benchmarking\java`  
gegebene Interface `Block.java`. Überlege dir geeignete weitere Datenstrukturen, um einen Index auf die Blöcke aufzubauen und z.B. Hash Buckets zu realisieren.
- Der Java Heap space sollte ausreichen; bei Fehlern lässt er sich erhöhen durch `(java -Xmx<size> -Xms<size> ;`

- Die ausführbare Klasse ist: `de.unipotsdam.hpi.is.dbsii.indexes.DataStoreExec`. Sie erhält als erstes Kommandozeilenargument den Pfad zu den Daten.
- Ausgabeformat:  
    `nonIndexed:`  
    `<key>: <value>, #blocks read: <count>, #values compared: <count>`  
    `...`  
    `runtime: <runtime in ms>`  
  
    `indexed:`  
    `<key>: <value>, #blocks read: <count>, #values compared: <count>`  
    `...`  
    `runtime: <runtime in ms>`
- Zippe sämtliche Java-Dateien und gib sie im Submit-System ab.

## Aufgabe 2: Picasso Database Query Optimizer Visualizer

Gegeben ist eine virtuelle Maschine (VM), in der eine DB2-Express Datenbank läuft. In die Datenbank wurde bereits ein 1GB Datensatz des TPC-H eingelesen sowie das Picasso-Tool installiert<sup>1</sup>. Wir wollen diesen Datensatz nutzen, um eine Reihe von TPC-H Anfragen und die je nach Anfrage und erwarteten Kardinalitäten variierenden resultierenden Anfragepläne zu analysieren.

### Hintergrund zum TPC-H Benchmark

Im Rahmen des TPC-H Benchmarks werden zunächst neue Tupel in die Tabellen *lineitem* und *orders* eingefügt, anschließend 22 Anfragen abgesetzt und zuletzt Daten aus den genannten Tabellen gelöscht. **In dieser Übung verwenden wir lediglich einige Anfragen, die ohne Inserts und Deletes auskommen** (Wer will, kann diese zum Selbststudium aber natürlich auch noch betrachten).

Verwende das Tool Picasso, um die Diagramme zu Ausführungsplänen und -kosten für die folgenden Anfragen zu betrachten. Verwende Plot Resolution 10 und betrachte die Pläne auf Operator Level. Gebe für die Aufgaben Screenshots und kurze textuelle Beschreibungen bzw. Erklärungen ab.

- a) Betrachte folgende Query:

```
SELECT * FROM customer WHERE c_custkey :varies
```

Hinweise:

- Die Tabelle *customer* sollte nach *customer\_pk* organisiert sein. Statistiken nicht vergessen!
- 1) Wo waren also die Grenzen in Bezug auf *c\_custkey* für die unterschiedlichen Ausführpläne? **2 P**
  - 2) Vergleiche das abgeschätzte und das reale Kostendiagramm (Comp Cost Diag und Exec Cost Diag). Was ist der Unterschied? **2 P**
  - 3) Vergleiche die Diagramme der erwarteten und gemessenen Ausführungszeit bei einer Plot Resolution 10 und 30. Wie wirkt sich die Veränderung aus? Erläutere kurz die Unterschiede und deine Vermutungen über dessen Gründe **4 P**

---

<sup>1</sup><http://dsl.serc.iisc.ernet.in/projects/PICASSO/>

b) Betrachte TPC-H Query 5 (in Picasso: Load Query Template -> db2 -> 5.sql)

Hinweis:

- Die Tabelle customer sollte jetzt nach c\_acctbal organisiert sein. Statistiken nicht vergessen! Hinweis: REORG benötigt einen Index, nach dem organisiert wird.
- 1) Was sind die Unterschiede der Plan-Diagramme
    - an der Grenze 85 - 95 % Selektivität auf c\_acctbal? 2 P
    - zwischen den Plänen für 25 und 40 % Selektivität auf c\_acctbal? 2 P
  - 2) Vergleiche den Plan für 25 % Selektivität auf c\_acctbal (auf Optimizer-Level 5) mit dem Plan auf Optimizer-Level 9. Was ist der Unterschied? Hinweis: Im Picasso Jargon sucht ihr einen Foreign Plan Tree. 4 P
- c) Betrachte die Plan-Diagramme für TPC-H Query 2 unter jeweils verschiedenen, folgenden Voraussetzungen: Wo fallen dir Unterschiede auf, wie erklärst du dir diese? Gib für jeden Schritt einen Screenshot des Plan-Diagrams ab!
- 1) Analysiere Query 2, ohne Optimierungen vorzunehmen. 2 P
  - 2) Füge einen Index auf p\_retailprice ein, ohne jedoch Statistiken neu zu berechnen. 2 P
  - 3) Berechne nun die Statistiken neu. 2 P
  - 4) Reorganisiere die Tabelle Part nach dem neu erstellten Index, ohne weitere Optimierungen vorzunehmen. 2 P
  - 5) Berechne erneut die Statistiken. 2 P

#### Hinweise zu Picasso:

- Dokumentation zu Picasso unter `/picasso2.1/PicassoDoc/index.htm`
- Zum Aufruf des Servers: `cd /picasso2.1//PicassoRun/Unix; ./runServer.sh`
- Zum Aufruf des Clients: `cd /picasso2.1//PicassoRun/Unix; ./runClient.sh`
- Einige Interaktionen mit Picasso sind nur per Tasten- und Mauskombinationen verfügbar: Help -> Mouse-Key Mappings.
- Query Template Descriptor: Name kann man sich selbst ausdenken, sollte eindeutig sein, *muss* angegeben sein.
- Die DBConnection ist schon angelegt: `db2_connection`.
- **Statistiken** erstellt man mit dem Befehl `RUNSTATS TABLE`. Eine Beispieldatei findest du in `/home/db2inst1/tpch/runstats-table.sql`.
- Manchmal hängt Picasso, nach einem Neustart von Client/Server geht es wieder.
- Eine Reorganisation einer Tabelle kannst du mit dem `REORG TABLE` Befehl ausführen, der auch im IBM Data Studio ausgelöst werden kann (Suche im Administration Explorer nach den Tabellen, schau im Kontextmenü einer Tabelle.)

#### Allgemeine Hinweise zur VM

Für die praktischen Aufgaben stellen wir die virtuelle Maschine "DB2 Express-C 9.7 32-bit" zur Verfügung. Die virtuelle Maschine (VM) ist folgendermaßen eingerichtet:

- Betriebssystem: *SUSE Linux Enterprise Server*
- Datenbank: *DB2 Express-C 9.7*
- Datenbank-Instanz: *db2inst1*
- Datenbank-Name: *db2db1* (enthält bereits alle Tabellen und Daten des TPC-H mit scale factor 1, also 1GB)

In der virtuellen Maschine wurde ein Nutzer für das Betriebssystem und die Datenbank angelegt mit den folgenden Zugangsdaten:

- Nutzernamen: *db2inst1*
- Passwort: *ws2011*

### Hinweise zum Umgang mit der VM

- Arbeit an einem Poolrechner:
  - Die Arbeit an einem Pool-Rechner ist *NICHT* mehr möglich, da auf Poolrechnern HyperV installiert ist, welches inkompatibel mit VMware ist. Die Aufgabe muss daher am eigenen Rechner gelöst werden.
- Arbeit mit dem eigenen Rechner:
  - Voraussetzung: VMware Player (Freeware)
  - VM kopieren von Laufwerk  
R:\lehrveranstaltungen\FG\_Informationssysteme\VL DBS II\uebung benchmarking
    - \* Alle Dateien werden benötigt
    - \* Die ausführbare Datei ist *DB2 Express-C 9.7 32-bit.vmx*
  - Starten der VM über:
    - \* Starte VMware Player > Open a Virtual Machine > *DB2 Express-C 9.7 32-bit.vmx*
    - \* ODER: Doppelklick auf *DB2 Express-C 9.7 32-bit.vmx*
- Die virtuellen Festplatten der VM sind nicht schreibbar, d.h. alle Änderungen an einer laufenden Instanz gehen nach dem Neustart der VM verloren, so dass sich die VM wieder im „Auslieferungszustand“ befindet. Falls du die VM auf deinem eigenen Rechner verwendest, besteht jedoch die Möglichkeit sie im VMware Player zu „suspenden“ (Virtual Machine > Power > Suspend).
- Beim Start fragt die VM evtl. ob Software-Updates installiert werden sollen. Das ist nicht notwendig (Remind me later).
- Einstellung für deutsches Tastaturlayout:  
Computer > Control Center > Hardware > Keyboard > Layouts > Add > Germany > set as Default
- Falls du plötzlich eine arabische Schrift hast drücke STRG + Leertaste.
- Der Zugriff auf das Internet ist nicht freigegeben. Daten können aber per Drag-and-Drop in das VM-Fenster mit dem Host-Rechner ausgetauscht werden.

### Möglichkeiten zum Ausführen von Anfragen

- Kommandozeile
  - Shell öffnen:  
Computer > Gnome Terminal
  - Verbindung mit der Datenbank herstellen:  
db2 connect to DB2DB1
  - Queries ausführen (Beachte die Anführungsstriche!):  
db2 "<sqlquery>"
  - Ein (selbstgeschriebenes) SQL-Skript ausführen:  
db2 -tvf <skriptname>
  - Verbindung mit der Datenbank trennen:  
db2 connect reset
- IBM Data Studio
  - *Data Studio* starten:  
Desktop > DB2 Data Studio
  - Verbindung zur Datenbank herstellen:  
Administration Explorer > localhost/5001/DB2DB1 expandieren, rechtsklicken und "Connect" wählen > Nutzerdaten angeben > "Save Password" setzen > OK

- Projekt anlegen:  
File > New > Data Development Project > Next > Finish > No
- Script anlegen:  
Rechtsklick auf Projekt > New > SQL or SQuery Script > Finish
- SQL-Anfragen ausführen:
  - \* SQL in Script eintippen
  - \* SQL zum Ausführen markieren (falls nichts markiert ist, werden alle Anfragen ausgeführt)
  - \* Play-Button klicken (alternativ: Script > Run SQL)
- Ausführungsplan anzeigen mittels Visual Explain:  
SQL-Anfrage markieren > Rechtsklick > Open Visual Explain > Finish

Alle notwendigen Skripte, die VM und die TPC-H Spezifikation liegen auf dem HPI-Netzlaufwerk

R:\lehrveranstaltungen\FG\_Informationssysteme\VL DBS II\uebung benchmarking

### Hilfreiche Hinweise

Die TPC-H Daten sind bereits in die Datenbank db2db1 geladen und können daher sofort genutzt werden.

Im Home-Verzeichnis des Nutzers liegen außerdem

- optionale Updates und Skripte zum Zurücksetzen der Updates
- die Rohdaten und load logfile
- die TPC-H queries (Achtung: query22 liegt hier in einer veralteten Version vor!)

Folgende Tools können bei der Bearbeitung der Aufgabe helfen:

- *Visual Explain* im DB2 Studio
- *db2advsi* im Terminal
- **Die DB2-Dokumentation findest du unter:**  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>
- *Hilfe zu typischen Picasso-Fehlern gibt es unter* [http://dsl.serc.iisc.ernet.in/projects/PICASSO/picasso\\_download/doc/Usage/troubleshooting.htm](http://dsl.serc.iisc.ernet.in/projects/PICASSO/picasso_download/doc/Usage/troubleshooting.htm)

Falls db2advsi den Fehler "Explain tables not set up properly ..." ausgibt, dann können die Explain Tables auf der Konsole folgendermaßen neu erstellt werden:

```
db2 -tf ~/sqllib/misc/EXPLAIN.DDL
```