Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Datenbanksysteme · Prof. Dr. Grust

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Datenbanksysteme II

SS 2016

Benjamin Dietrich, Dennis Butterstein

1. Übungsblatt

Ausgabe: 18. April 2016 · Besprechung: 25. April 2016

Aufgabe 1: Git (4 Punkte)

Da ihr nun im Besitz dieses Übungsblattes seid, ist davon auszugehen, dass es euch gelungen ist, ein Repository für euer Team bei *GitHub* anzulegen und auszuchecken.

Die erste Aufgabe wird sein, darin ein Text-File `team.txt` anzulegen, in dem jeder Teampartner gesondert *Name* und *Matrikelnummer* hinterlässt. Hierzu checkt ein Teampartner ein solches File ein. Der zweite Teampartner kann dieses sodann auschecken, ergänzen und wieder einchecken.

Auf diesem Wege werdet ihr künftig nicht nur eure Abgaben einreichen und ein Korrekturfeedback erhalten, ihr könnt und sollt so auch die Zusammenarbeit im Zweierteam organisieren.

Hinweis: Falls ihr eure Abgaben als PDF oder in Form eines anderen Binary-Files einreicht, lohnt es sich entsprechend die (LaTeX-)Sourcen einzuchecken. Nur so kann der Teampartner (oder auch der Korrektor) die Abgabe bearbeiten und ergänzen. Generell gilt beim Arbeiten mit Versionskontrolle der Grundsatz: Dateien, die aus den Quellen in einem Repo generiert werden können, werden nicht eingecheckt.

Aufgabe 2: PostgreSQL (6 Punkte)

Für weitere Übungen und zum Nachvollziehen der Vorlesungsinhalte ist eine funktionierende PostgreSQL-Instanz nötig.

Ausführliche Dokumentation zu PostgreSQL findet ihr unter http://www.postgresql.org/docs/current/static/index.html.

1. Installiert die aktuelle Version (9.5.2) von PostgreSQL. Dazu könnt ihr beispielsweise verfügbare Pakete für eure Linux-Distribution, die für Mac OS X bereitgestellte Postgres.app oder den Installer für Windows benutzen (http://www.postgresql.org/download/).
2. Legt eine Datenbank und eine Tabelle mit einem selbst gewählten Schema an (`createdb`, `CREATE TABLE`).
   (Hinweise: http://www.postgresql.org/docs/current/static/app-createdb.html
   und http://www.postgresql.org/docs/current/static/sql-createtable.html)
3. Gebt den *filenode* eurer Relation an. Gebt zudem den Pfad der Datei an, in dem eure Tabelle abgelegt ist. Wie seid ihr an diese Information gelangt?
   (Hinweise: http://www.postgresql.org/docs/current/static/catalog-pg-class.html
   und http://www.postgresql.org/docs/current/static/storage-file-layout.html)

Seagate Enterprise Performance 15K.5[a]

Average Seek Time = 3.4 ms
Rotational Speed = 15,000 RPM
Average Transferrate = 196 MB/s

[a] http://www.seagate.com/internal-hard-drives/enterprise-hard-drives/hdd/enterprise-performance-15k-hdd/#specs

Abbildung 1: Daten einer Serverplatte

Aufgabe 3: Review Questions (4 Punkte)

Erklärt **kurz** die Begriffe *Speicherhierarchie*, *seek time*, *rotational delay* sowie *transfer time*.

Aufgabe 4: Disk Space Manager (2 Punkte)

In der Vorlesung wurde der *Disk Space Manager* als Komponente eines Datenbanksystems angesprochen. Auch das Betriebssystem selbst verwaltet den verfügbaren Sekundärspeicher und stellt die Abstraktion von byte-addressierbaren *Dateien* zur Verfügung. Welche Gründe könnte es geben, diese Funktionalität in einem Datenbanksystem noch einmal zu implementieren?

Aufgabe 5: Access Time (4 Punkte)

Bereits in der Vorlesung haben wir gesehen, dass selbst moderne Festplatten einen "Flaschenhals" in der Architektur eines Datenbanksystems darstellen.
In der folgenden Aufgabe soll eine aktuelle *server-class* Festplatte (siehe Abb. 1) und der Sekundärspeicher in eurem Computer (Festplatte oder SSD) verglichen werden. Recherchiert bzw. messt die benötigten Merkmale für euer Gerät und ergänzt sie gegebenenfalls durch sinnvolle Annahmen.

1. Wir nehmen an, dass wir jeweils 8 KB Blöcke lesen. Berechnet unter dieser Annahme für beide Geräte, wie lange es im Mittel dauert
   - sequentiell[1] 10,000 Blöcke zu lesen?
   - wahlfrei 10,000 Blöcke zu lesen?
2. Vergleicht jeweils die Zeiten für ein sequentielles und wahlfreies Auslesen der Speichermedien. Wie kommt es zustande, dass sequentielles Lesen (im Falle einer Festplatte) so viel schneller als wahlfreies ist?

[1] Vervollständigt hierbei die angegebenen Spezifikationen durch sinnvolle Annahmen über *Track to Track Seek Time* und die durchschnittliche Menge an Daten pro Track (z.B. aus dem Vorlesungsskript Storage-7).