

1. Aufgabenblatt

Skalierbare Schnittmengenberechnung mit Constraints

Abgabe: 08.11.17, 23:59 Uhr

Patrick Schäfer

Berlin, 23. Oktober 2017

patrick.schaefer@hu-berlin.de

Vorlesung:

https://hu.berlin/vl_dwhdm17

Übung:

https://hu.berlin/ue_dwhdm17

Aufgabenstellung

- Thema: Umgang mit großen Datenmengen und begrenzten Ressourcen
- Aufgabe: **Berechne Schnittmenge zweier Dateien**
 - Beide Dateien sind **1,2 GB groß** und bestehen aus jeweils 100.000.000 Einträgen
 - Das Ergebnis (**Schnittmenge**) soll in eine dritte Datei geschrieben werden
- Anforderung: Euer Programm **muss** mit **100 MB RAM** auskommen
- Wettbewerbsaufgabe: es gibt Punkte für die schnellste, korrekte Implementierung

Eingabeformat

- Zeilenweise (ASCII-Format) Einträge der Form:
A2938230859
E3404634063
C3069349063
F3094364306
- Format:
 - 1. Element ein Buchstabe (A-J)
 - 2.-11. Element jeweils Ziffern
 - (newline)

Gleichheit der Einträge

- Kriterium:
Zwei Einträge sind gleich, falls die Zahlen gleich sind
- Beispiele:
$$C3209820397 = C3209820397$$
$$B3209820397 = H3209820397$$
$$C9912842891 \neq C9912842892$$
$$C9912842891 \neq H3209820397$$

(Vereinfachtes) Beispiel

- Datei 1: A5, D6, A3, F8, J3
- Datei 2: B9, F3, D6, F1
- Ausgabe (Schnittmenge) in beliebiger Reihenfolge:
D6
A3
F3
J3
- Ein Element pro Zeile

Hinweise

- Starte die JVM mit einer maximalen Heapsize von 100 MB:
java -Xmx 100m
- D.h., eure Lösung muss mit 100 MB RAM auskommen!
- Temporäre Dateien in Java anlegen:

```
File tmpFile = File.createTempFile(prefix, suffix);  
tmpFile.deleteOnExit();
```
- Die Schnittmenge hat 2.097.858 Element
- Testen mit kleineren Dateien:

```
head -1000000 file1.txt > file1_small.txt  
head -1000000 file2.txt > file2_small.txt
```

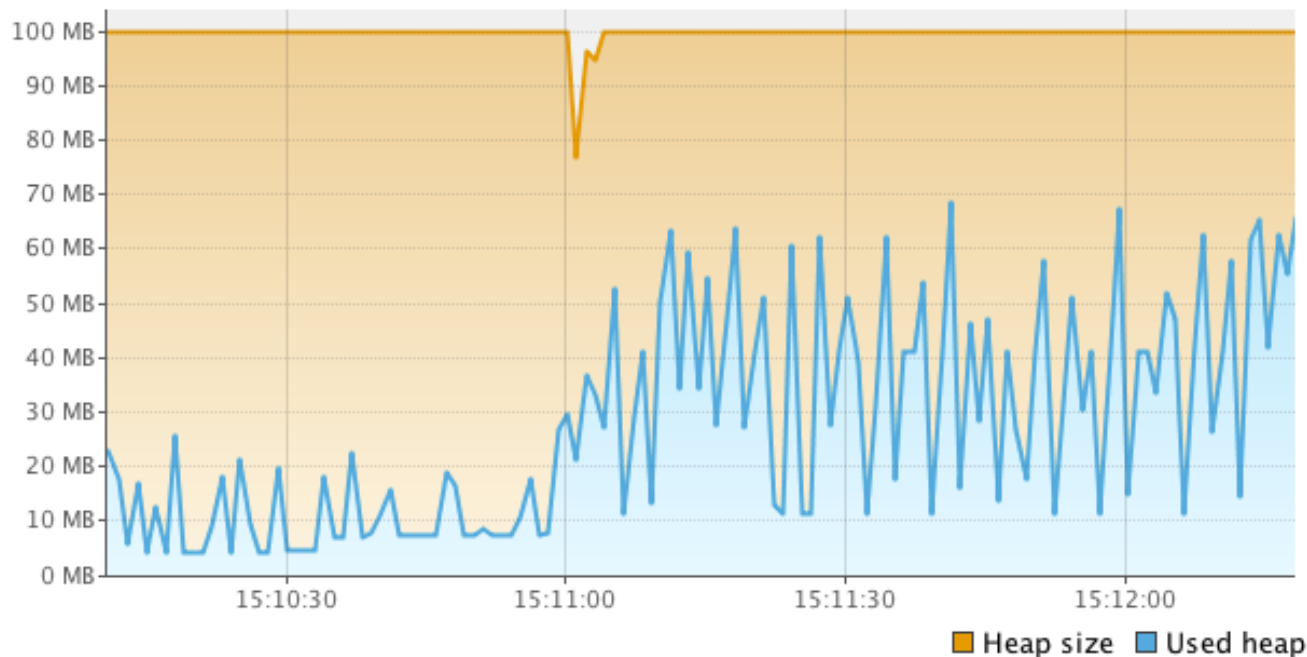
 - Schnittmenge hat hier 190 Element

Hinweis

- Mit **VisualVM** könnt ihr den Speicherverbrauch und euer Programm Profilen (Sampeln)

Size: 104.857.600 B
Max: 104.857.600 B

Used: 69.273.080 B



Wettbewerb

- Testet euer Programm auf gruenau2.informatik.hu-berlin.de
- Ziel: Korrekte Schnittmenge so schnell wie möglich bilden
- Wettbewerbspunkte:
 - 1. Gruppe: 5 Punkte
 - 2. Gruppe: 3 Punkte
 - 3. Gruppe: 2 Punkte

Abgabe

- Deadline ist **Mittwoch, 08.11.17, 23:59 Uhr**
- Upload eines Archivs (zip, tar.gz) als ‚**GruppenName.zip**‘ per HUBox:
https://hu.berlin/dwhdm_ue1
- Die Abgabe im Einzelnen:
 - **Ausführbares JAR** (!) und **Quellcode** der Lösung
 - Eine **Anleitung** für euer Programm:
 - Wie wird es aufgerufen?
 - Anzahl der Ergebnisse der Schnittmenge
 - Eure selbstgemessenen Laufzeiten (time java ...)
 - Es zählt wall-clock-time!
- Ihr dürft kein DBMS (Oracle) benutzen
- Ihr dürft keine Libraries verwenden, die Joins implementieren.

Präsentation

- Ihr dürft euch aussuchen, was und wann ihr präsentieren wollt (first-come-first-served).
- Montags (Gruppe 1)
[https://dudle.inf.tu-dresden.de/dwhdm mo ue1/](https://dudle.inf.tu-dresden.de/dwhdm_mo_ue1/)
- Mittwochs (Gruppe 2)
[https://dudle.inf.tu-dresden.de/dwhdm mi ue1/](https://dudle.inf.tu-dresden.de/dwhdm_mi_ue1/)
- Kurzpräsentation der Lösung am 13.11. (Gruppe 1) bzw. 15.11. (Gruppe 2).

FAQ

- Gibt es Duplikate in den Dateien?
 - **Ja**, z.B. B4057272120 kommt in Datei 1 zweimal vor. Insgesamt hat Datei 1: 55315 und Datei 2: 55610 Duplikate.
 - **Aber**: An der Schnittmenge ist keines der Duplikate beteiligt.

Fragen?