

<Witziger Titel> (work in progress)

Frederik Wille <3wille@informatik.uni-hamburg.de>

Alexander Timmermann <3timmerm@informatik.uni-hamburg.de>

Synopsis

- **Effiziente Verteilung von Passagieren auf Flüge und Plätze**
 - Fluggäste geben kein genaues Abflugdatum an, sondern Präferenzen (morgens/mittags/abends)
- **Clusterung von zusammengehörenden Gruppen**
 - möchten gerne zusammengehörige Plätze haben
- **Einteilung in verschiedene Klassen**
 - 1st, Business, Economy...
- **Präferenzen**
 - Gang, Fenster, Mitte

Sequentieller Algorithmus

INPUT:

- Buchungsdaten der Reisenden für einen Zeitraum
- Informationen der Flüge in diesem Zeitraum
 - Anzahl der Plätze pro Klasse
 - Abflugdaten

LOOP:

- Gruppen nach Größe absteigend auf Flüge verteilen
- Gruppen in verschiedene Klassen einteilen

Sequentieller Algorithmus

LOOP (cont.):

- Einzelpersonen auf die Flüge verteilen
- Einzelpersonen auf die verbleibenden Plätze verteilen

IF verbleibende freie Plätze > 10:

- versuchen, Flugzeug auf ein kleineres downzugraden

Lösungsansatz

- **Clusterung der zusammengehörenden Gruppen beachten!**
- **parallelen Algorithmus für Verteilung auf die verschiedenen Flüge finden**
 - naiv: einfach rein bis alle Flugzeuge voll ist
 - intelligenter(?): verschiedene Kombinationen ausprobieren und die “fitness” der jeweiligen Modelle vergleichen
- **Parallelisierung der Verteilung auf die Klassen und einzelnen Plätze**
 - Herausforderung: Gemeinsamer Zugriff auf eine Datenbasis, deren Änderungen ständig beachtet werden müssen

Zeitplan

- **Zwei Prototypen/Beta-Versionen zu den jeweiligen Statustreffen**
- **Zum 1. Treffen**
 - grundlegender Algorithmus mit Basis-Implementation
- **Zum 2. Treffen**
 - Effizientere Version, grundlegend parallelisiert
- **Zum Praktikumsabschluss**
 - Fertiggestellte Implementation, final optimiert

Parallelisierungsschema

- **Daten-Aufteilung:**
 - nach Passagierklasse
 - nach zusammengehörenden Gruppen
 - nach Flugziel
 - nach Abflugpräferenzen