SUBSCALE Outline

William Mendat[1] Max Ernst[2] Steven Schall[3] Matthias Reichenbach[4]

1 Verteilung

1.1 CoreSets

Der Algorithmus beginnt damit die CoreSets jeder Dimension zu bilden. Die Bildung der CoreSets geschieht innerhalb einer Dimension und ist zudem auch komplett unabhängig von den anderen Dimensionen. Dabei wird sich innerhalb einer Dimension jeder punkt angeguckt und die entfernungen dieses Punktes betrachtet. Sollte die Entfernung kleiner als der angegebene Epsilon wert sein, so wird der Punkt zu dem CoreSet hinzugefügt.

Da in diesem Prozess überhaupt keine abhängigkeit von anderen Dimension besteht und einzig und alleine die Daten der jeweiligen Dimension benötigt wird, kann die Bildung der CoreSets für jede Dimension parallel abgearbeitet werden.

Somit kann erreicht werden, dass jeder PC in einem Cluster nur ein Subset von Dimensionen abarbeitet, um die Corsets zu bilden.

1.2 DenseUnits

1.3 System vergleicht Dense Units von Zwei Dimensionen

Nachdem die Dense Units für eine Dimension und deren CoreSets erzeugt wurden, müssen für diese Dense Units Signaturen gebildet werden und die Punkte mit deren Dimension in eine Tabelle eingetragen werden. Die Berechnung der Signatur ist unabhängig von anderen Dimensionen und Dense Units, somit kann dies auf verschiedene Rechnern Verteilt werden. Das eintragen der Daten muss jedoch über Rechnergrenzen hinweg synchronisiert werden. Wenn ein Rechner für die übermittelte Dense Unit die Signatur erzeugt, muss dieser vorerst prüfen ob bereits ein Eintrag mit dieser Signatur existiert und ergänzt die Dimension

[1] Hochschule Offenburg, Offenburg, Deutschland wmendat@stud-hs.offenburg.de
[2] Hochschule Offenburg, Offenburg, Deutschland mernst@stud-hs.offenburg.de
[3] Hochschule Offenburg, Offenburg, Deutschland sschall@stud-hs.offenburg.de
[4] Hochschule Offenburg, Offenburg, Deutschland mreichen@stud-hs.offenburg.de

an diesem Eintrag. Wenn beispielsweise für die Datenhaltung eine SQL oder NoSQL Datenbank verwendet wird, muss bevor ein Eintrag gespeichert wird, zuerst abgefragt werden ob die Signatur bereits hinterlegt ist. Bei großen Datenmengen erzeugt das ständige abfragen ob die Signatur bereits existiert einen großen overhead beim speichern. Eine mögliche Alternative um diese Abfrage zu umgehen, kann eine eigene Implementierung einer Distributed Hash Table verwendet werden. Dabei werden die Dense Units sowie die Dimension dem Peer-to-Peer Netzwerk übergeben, dieses berechnet aus den Punkten die Summe (Hashwert) der Dense Unit und beauftragt den zuständigen Knoten die Daten zu speichern. Der Zuständige Knoten müsste bei einer Kollision der Signatur die Dimension am Eintrag ergänzen. Dadurch würde das berechnen der Signatur auf dem Peer-to-Peer Netzwerk verteilt werden und für Kollisionen ist nur der Zuständige Knoten verantwortlich.

1.4 Gruppierung nach Dimensionen

2 Framework und Architektur

2.1 Master-Worker-Architektur

2.2 Pipeline

Um die im Vorherigen beschriebenen Kapitel möglichen Verteilungen zu realisieren, könnte eine Pipeline aufgebaut werden um diese schritte abzuarbeiten. Dabei würde jeder Rechner Aufgaben aus einer Queue bekommen und diese an die Nächste Queue weiterschicken. Somit wäre eine gleichmäßige Verteilung der Aufgaben möglich und die Rechner sind voneinander entkoppelt. Die Entkopplung der Rechner ermöglicht außerdem die Verwendung von verschiedenen Programmiersprachen bei den unterschiedlichen Aufgaben. Um dies zu Realisieren benötigt es jedoch einen Rechner der als Initiator des Algorithmus wirkt und einen Rechner zum einsammeln der Ergebnisse, der am Ende der Pipeline auf die Berechnung wartet.

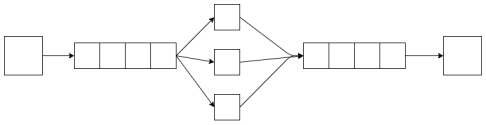

Abb. 1: Beispiel der Pipeline

2.3 Peer-to-Peer

2.4 TCP Posix

Mithilfe der Posix Bibliothek könnte eine Client-Server Architektur aufgebaut werden, wo sich die einzelnen Clients mit dem Server verbinden und der Server, die Aufgaben verteilt und die Ergebnisse einsammelt.

Vorteile:

- Leichte Übertragbarkeit
- Es ist ein weitverbreiteter Standard, also existieren viele Dokumentationen und Beispiele im Internet

Nachteile:

- Es ist eine alte Technologie und auch komplexer als neuere Technologien