Massiv parallele kNN-Suche auf der GPU im Performancevergleich zur kNN-Suche über Indexstrukturen

Exposé zur Studienarbeit

eingereicht von: David Salomon
geboren am: 07.08.1987
geboren in: Cottbus

Gutachter/innen: Prof. Dr. Ulf Leser

eingereicht am: 03.03.2015

1 Einleitung

Recurrence Quantification Analysis (RQA) ist eine Methode zum Auffinden von Wiederholungen (*Reccurences*) in einer Zeitreihe. Hierfür wird die Verteilung der vertikalen und diagonalen Linien in einer Ähnlichkeitsmatrix, der sogenannten *Reccurence Matrix*, bestimmt. Zur Konstruktion der Ähnlichkeitsmatrix wird die Ähnlichkeit jedes Vektors zu allen anderen berechnet. Als Ähnlichkeitsmaß dient z.B. die euklidische Distanz. Unterschreitet die Distanz zwischen zwei Vektoren eine definierte Schranke wird eine 1 in die Matrix eingetragen. Vertikale und diagonale Linien ergeben sich durch aufeinanderfolgende Einsen [1].

Die RQA wird in vielen wissenschaftlichen Bereichen eingesetzt. Zum Beispiel wird das Herz-Lungen-System mittels der RQA untersucht um Einblicke in dessen Funktionsweise zu erhalten. Zusätzlich können Fehlfunktionen gemessen werden, um lebensbedrohliche Zustände zu diagnostizieren [2].

Bisherige Berechnungsansätze erlaubten die Analyse von wenigen Datenpunkten (5000 - 20000). Der Ansatz des deutschen GeoForschungsZentrums Potsdam (GFZ) erlaubt es sehr lange Zeitreihen (> 1 Mio. Datenpunkte) in akzeptabler Zeit zu analysieren. Erreicht wird dies durch einen stark parallelisierten Berechnungsprozess [2].

Zur parallelen Verarbeitung wird die Matrix aufgeteilt und die Berechnung der einzelnen Teilmatrizen auf mehrere *Computing Devices* aufgeteilt. Die Verteilung der vertikalen und diagonalen Linien in der *Recurrence* Matrix wird nun durch Zusammenführen der Ergebnisse jeder einzelnen Teilmatrix erzeugt [2].

2 k-Nächste Nachbar-Suche

In einem Datenset D sucht die k-Nächste Nachbar-Suche alle $k \in D$ Elemente, welche einem gegebenen Datenobjekt $q \in D$ am ähnlichsten sind. Hierzu muss ein angemessenes Ähnlichkeitsmaß ausgewählt werden, bspw. die euklidische Distanz.

Ein aktuelles Einsatzgebiet der k-Nächsten Nachbar-Suche (kNN-Suche), ist beispielsweise der Aktienmarkt. Hier wird die kNN-Suche als Teil des Data-Mining eingesetzt, um Marktvorhersagen zu treffen. Diese Vorhersagen beinhalten Markt Trends, Investment Strategien und Kaufempfehlungen. Ein weiteres Einsatzgebiet ist die Medizin, in der unter anderem mit ihrer Hilfe ein Risikofaktor für Prostata Krebs auf Basis klinischer und demografischer Variablen ermittelt wird [12].

3 Ziel der Studienarbeit

Das Problem der RQA ähnelt der kNN-Suche, da jeweils eine große Anzahl an Ähnlichkeitsvergleichen durchgeführt werden muss. Dies motiviert zu der Zielstellung dieser Arbeit: Kann eine kNN-Suche mit dem massiv parallelen Berechnungsansatz des

GFZ durchgeführt werden, und ist diese Suche schneller als bestehende Ansätze zur Durchführung der kNN-Suche?

Hierfür muss der bestehende parallele Ansatz an die Eigenschaften der kNN-Suche angepasst werden. Es entsteht eine Implementierung, welche die kNN-Suche parallelisiert durchführt. Im Ergebnis muss festgestellt werden ob der angepasste parallele Ansatz im Kontext der kNN-Suche Performance Gewinne erzielt. Zusätzlich muss analysiert werden, welche Eigenschaften die Datensets benötigen um diesen Performance Gewinn zu erzielen.

4 Aktueller Stand der Technik

Für die kNN-Suche gibt es in der Informatik mehrere Ansätze, um die Performance der Berechnung zu beschleunigen. Zwei Kategorien werden im folgenden vorgestellt:

4.1 Parallelisierung

Das Ziel ist es, die hohe Anzahl an Ähnlichkeitsvergleichen parallel durchzuführen. Dies ist möglich, da alle Ähnlichkeitsvergleiche unabhängig voneinander durchgeführt werden können und in ihrer Komplexität gering sind.

In [4] wurden die parallelen Verarbeitungsmöglichkeiten von GPUs im Kontext der kNN-Suche eingesetzt. Hierzu wurden eine hohe Anzahl von Ähnlichkeitsvergleichen nebenläufig durchgeführt. Es wurde gezeigt, dass ebenfalls evaluierte Algorithmen (eine Indexstruktur, 3 nicht parallelisierte Brute-Force Algorithmen) deutlich langsamer waren, als ihre GPU Variante. Die Testreihe umfasste maximal 38400 Datenpunkte bei 96 Dimensionen.

Die Autoren von [4] haben aufgezeigt, dass im niedrigen Dimensionsbereich Indexstrukturen schneller als eine parallele Verarbeitung sein können. Dies konnte für höhere Dimensionen nicht bestätigt werden, was wahrscheinlich auf die geringe Datenmenge zurückzuführen ist.

Die Parallisierungsmethode des GFZ basiert auf dem Konzept „*Divide and Combine*“. „*Divide and Combine*“ sorgt dafür, dass die parallele Verarbeitung immer nur auf so großen Datenteilmengen arbeitet wie in den Speicher des *Computing Device* passt. Somit wird erreicht, dass beliebig große Datenmengen benutzt werden können.

4.2 Indexstrukturen

Der zweite Ansatz zum Beschleunigen der kNN-Suche sind Indexstrukturen. Durch Indexstrukturen wird versucht, die Anzahl der tatsächlich durchgeführten Ähnlichkeitsvergleiche bei der kNN-Suche stark zu reduzieren. Hierfür werden die Daten in eine Datenstruktur überführt, welche es erlaubt, die k nächsten Nachbarn zu einem Anfragepunkt p zu finden, ohne dass p mit allen Datenobjekten verglichen werden muss. Die kNN-Suche wird dadurch drastisch beschleunigt, allerdings benötigen diese Ansätze viel Zeit zum Erstellen einer solchen Struktur.

Neben der zuvor beschriebenen paralellen kNN-Variante wurde in [4] ebenfalls eine Implementierung auf Basis eines k-d-Trees evaluiert. Hierbei wurde gezeigt, dass diese Implementierung bei kleineren Dimensionen eine bessere Performance bietet als der parallele Ansatz auf der GPU. Zu evaluieren ist, ob der k-d-Tree bei höherer Dimensionalität und einer größeren Datenmenge ebenfalls so performant ist.

Eine weitere Indexstruktur ist der M-Tree, welche unter anderem zum Beschleunigen der kNN-Suche geeignet ist. Die Autoren von [5] beschreiben mehrere Algorithmen wodurch der Aufbau eines solchen Baumes beschleunigt werden kann.

Sowohl für den k-d-Tree[7], als auch für den M-Tree[8] existieren frei verfügbare Implementierungen, welche für die Studienarbeit benutzt werden können. Die Implementierung des M-Trees basiert auf [9].

5 Datensets

5.1 Datenmenge

Zur Evaluierung werden sowohl reale als auch synthetisch erzeugte Datensets verwendet. Alle Datensets bestehen aus Vektoren realer Zahlen. Daraus folgt, dass für die Ähnlichkeitsvergleiche die Euklidische Norm verwendet wird.

Die verwendeten Datensets sollen sich außerdem in der Anzahl der Dimensionen ihrer Datenpunkte unterscheiden. Es soll jeweils ein Datenset mit ~10, ~40 und ~60 Dimensionen vertreten sein.

Zusätzlich zu der Anzahl der Elemente und Dimensionen müssen alle benutzen Datensets in ihren Eigenschaften beschrieben werden: Verteilung der Datenpunkte, Duplikate (identische Vektoren) enthalten, etc. Die Eigenschaften eines Datensets dienen der Bewertung der Performance der kNN-Suche.

5.2 KDD-Cup Netzwerkdaten

Ein Datenset sind die Daten des KDD Cup von 1999 [3]. Für diesen Cup wurde ein Datenset bestehend aus ca. 5 Millionen Datenpunkten bereitgestellt. Dieses Datenset beinhaltet simulierten Netzwerkverkehr bestehend aus normalen und schlechten Verbindungen. Angriffe auf das Netzwerk stellen schlechte Verbindungen dar.

Für die Evaluation müssen alle Variablen mit nicht numerischen Werten entfernt werden. Es entsteht ein Datenset der Dimension 38 mit ca. 75% Duplikaten.

5.3 MovieLens

Ein weiteres Datenset sind die Daten einer Platform zur Bewertung von Filmen. Bereitgestellt wird dieser von GroupLens Research [10]. Dieses Datenset enthält 1 Millionen Film-Bewertungen. Zusätzlich werden demographische Daten der Bewerter und Daten der bewerteten Filme zur Verfügung gestellt.

Für diese Arbeit werden die Filmbewertungen und Bewerterdaten zusammengefügt. Nach dem Entfernen aller nicht euklidischen Werte entsteht ein Datensatz mit 1 Millionen Datenpunkten der Dimension 7. Der entstandene Datensatz enthält keine Duplikate.

5.4 Metric Spaces Library

Die Metric Spaces Library ist eine kleine Sammlung von Datensets des *International Workshop on Similarity Search and Applications (SISAP)* [11]. Diese Sammlung enthält unter anderem ein Datenset bestehend aus 100.000 Datenpunkten der Dimension 105. Jede Variable enthält einen euklidischen Wert zwischen 0 und 10.

Durch die hohe Dimensionalität können aus diesem Datenset weitere Datensets geringer Dimensionalität gewonnen werden.

6 Testdurchführung

Als Testrechner wird eine Maschine des GFZ benutzt. Diese enthält 2 GPU`s mit jeweils 2 Recheneinheiten.

Für den Test selbst wird zuerst ein Warm-Up Run durchgeführt und anschließend 3 Evaluierungsläufe. Der Mittelwert aus diesen 3 Läufen bildet das Testergebnis. Die Zeitmessung beginnt, nachdem die Daten in den Speicher des Programms geladen wurden.

Das Durchführen der kNN-Suche wird in 2 Teilschritten durchgeführt:

1. Die Berechnung aller Distanzen / Die Erstellung der Indexstrukturen
2. Die Extraction der k-Nächsten Nachbarn.

Neben der Gesamtzeit wird auch die Zeit jeden Teilschrittes gemessen.

7 Literaturverzeichnis

[1] N. Marwan, M.C. Romano, M. Thiel, J. Kurths: Recurrence Plots for the Analysis of Complex Systems, Physics Reports 438(2007) 237-329

[2] T. Rawald, M. Sips, N. Marwan, D. Dransch, Fast Computation of Recurrences in long Time Series, Springer 2014

[3] KDD Cup - http://www.sigkdd.org/kdd-cup-1999-computer-network-intrusion-detection

[4] V. Garcia, E. Debreuve, M. Barland: Fast k Nearest Neighbor Search using GPU, 2008

[5] P. Zezula, G. Amato, V. Dohnal, M. Batko: Similarity Search – The Metric Space Approach, Springer, 2006

[6] M. Slaney, M. Casey: Locality-Sensitive Hashing for Nearest Neighbors, IEEE Signal Processing Magazine, 2008

[7] http://libkdtree.alioth.debian.org/

[8] https://github.com/erdavila/M-Tree

[9] P. Ciaccia, M. Patella, P. Zezula: M-Tree: An Efficient Access Method for Similarity Search in Metric Space, 23rd VLDB Conference, Greece 1997

[10] grouplens.org/datasets/movielens/

[11] K. Figueroa, G. Navarro, E. Chávez: Metric Spaces Library, 2007, Available at http://www.sisap.org/Metric_Space_Library.html

[12] S. B. Imandoust, M. Bolandraftar: Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background, Int. Journal of Engineering Research and Application, Sep-Oct 2013