

Thema

**Studienarbeit**

des Studienganges Angewandte Informatik

an der Dualen Hochschule Baden-Württemberg Mannheim

von

Dominik Heßmer

Abgabedatum

Bearbeitungszeitraum 12 Wochen

Matrikelnummer, Kurs 1690631, TINF15 AI-BI

Ausbildungsfirma Atos Information Technology GmbH

Betreuer der Dualen Hochschule Prof. Dr. Rainer Colgen

Erklärung zur Eigenleistung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: *THEMA* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ort, Datum Unterschrift

Abstract

Inhaltsverzeichnis

[Kapitelüberblick 7](#_Toc511046313)

[State of the Art 8](#_Toc511046314)

[Aufgabenstellung 9](#_Toc511046315)

[1 Theoretische Grundlagen 10](#_Toc511046316)

[2 Praktische Umsetzung 11](#_Toc511046317)

[3 Fazit 12](#_Toc511046318)

[4 Literaturverzeichnis 13](#_Toc511046319)

Tabellen- und Abbildungsverzeichnis

Abkürzungsverzeichnis

Kapitelüberblick

State of the Art

Aufgabenstellung

# Theoretische Grundlagen

## Data Streams

Ein Data Stream S ist eine geordnete Sammlung von Data Items s1,s2,… mit den folgenden zwei Eigenschaften.

1. Data Items werden kontinuierlich von einer oder mehreren Quellen generiert und an eine oder mehrere Verarbeitungsobjekte.
2. Die Anordnung, in welcher die Data Items gesendet wurden, kann nicht durch die Verarbeitungsobjekte bestimmt werden.

Data Streams können in zwei Kategorien eingeteilt werden, Basis Streams und abgeleitete Streams. Als Basis Stream werden Streams bezeichnet, welche ihren „Ursprung“ direkt aus der Datenquelle haben. Abgeleitete Streams sind der Wortbedeutung nach Basis Streams, welche schon zuvor verarbeitet wurden (1 Liu und Özsu 2009, S. 638).

Die Data Items können je nach Anwendungsgebiet aus Datenpaketen, Rohtext, Events, Tupel oder komplexere Datenstrukturen, wie XML oder JSON bestehen. Des Weiteren besitzen die Data Items in der Regel zwei Timestamps. Ein Timestamp der Erzeugung des Objektes und ein Timestamp bei der Ankunft am Verarbeitungsobjekt (1 Liu und Özsu 2009, S. 638).

Die Schwierigkeiten beim Verarbeiten von Data Streams liegen in weiteren Eigenschaften. Data Streams können eine sehr hohe Frequenz haben und ihre Frequenz kann mit der Zeit variieren. Zudem kann die Verarbeitungslogik nicht vorhersehen, wann der Stream aufhört, wobei dieser theoretisch unendlich bestehen kann (1 Liu und Özsu 2009, S. 638).

Die hohe Frequenz der Data Streams wirkt sich vor allem auf die Datenübertragung, die Verarbeitung und die Speicherung der Daten aus. Auf English *transmit* (T), *compute* (C) und *store* (S), kurz TCS. Bei der Datenübertragung könnte die Verbindung langsam oder instabil werden. Dabei kann es zwar zu Verzögerungen der Daten kommen, jedoch werden die Daten früher oder später an die Verarbeitungslogik übertragen. Bei der Verarbeitung der Daten kann es je nach Komplexität und Rechenleistung zu langen Abfragen kommen, jedoch sollte die Verarbeitung im Prinzip funktionieren (2 Muthukrishnan 2005, S. 6).

Die Datenspeicherung macht bei der Verarbeitung von Streams die wenigsten Probleme, da die Skalierung von Speicherplatz über dafür spezielle Datenstrukturen schon erforscht sind (3 Vitter 2001, S. 209–271).

Die Herausforderungen an die TCS Architektur besteht durch automatisch generierten und hochdetaillierten Datenströme. Im Verbund mit der Analyse dieser Datenströme entstehen besonders Anforderungen an die Datenübertragung (T) und Verarbeitung (C). Um diese Herausforderungen zu bewältigen gibt es einige Ansätze.

Ein weit verbreiteter und bekannter Ansatz ist die Parallelisierung. Die Verarbeitung und die Speicherung kann sehr gut parallelisiert werden. Beispielsweise kann durch das Clustern der Verarbeitungslogik und der Festplatten die Kapazität erhöht werden (2 Muthukrishnan 2005, S. 7). Die Datenübertragung zur Verarbeitungslogik kann nicht so gut über Parallelisierung verbessert werden. Da eine aktive Verbesserung der Datenübertragungsrate schwer zu erreichen ist, muss anders vorgegangen werden. Die Datenrate kann erhöht werden, indem der Datenstrom entweder durch kontrolliertes Sampling oder unkontrolliertes Shredding verkleinert wird. Ein weiterer Ansatz ist die hierarchisch gegliederte Analyse. Dabei werden zunächst schnelle, dafür aber simple Analysen wie Filterung oder Aggregation durchgeführt. Auf den höheren Leveln werden anschließend komplexere Analysen auf kleineren Datensätzen gefahren (2 Muthukrishnan 2005, S. 7).

## Stream Processing Engine

Die erste generelle, relationale Stream Processing Architektur wurde 2001-2002 entwickelt (1 Liu und Özsu 2009, S. 639). Eine Stream Processing Engine (SPE) unterscheidet sich von einer traditionellen Datenbank Engine in 3 Charakteristika.

1. Continuous Query Modell
2. Inbound Processing Modell
3. Single Process Modell

Beim continuous Query Modell, werden die Abfragen ausgeführt sobald neue Daten zur Verfügung stehen. Dies steht im Gegensatz zum one-time Query Modell, bei dem die Abfragen von den Clients abgesetzt werden und nur ein Ergebnis produzieren.

Beim continuous Query Modell sind die Abfragen persistent und die Daten sind vergänglich. Beim traditionellen Modell sind die Daten persistent und die Abfrage sind vergänglich, (1 Liu und Özsu 2009, S. 639). Ein Vergleich zwischen traditionellem und Stream Processing ist in Abbildung 1 gegeben.



Abb.: 1 Vergleich traditionellem und Stream Processing Modell

Inbound Processing bedeutet, dass eingehende Streams sofort von der Verarbeiteten Logik bearbeitet werden, wenn sie im System ankommen. Dabei werden die Nachrichten ständig verarbeitet und produzieren ständig Ergebnisse. Dies geschieht soweit es geht im Hauptspeicher, wobei Schreiben und Lesen in den Speicher optional und meistens asynchron geschehen. Dadurch das für die Verarbeitung die Daten nicht in den Storage geladen werden müssen, sondern im Hauptspeicher verarbeitet werden, kann ein erheblicher Performancevorteil gegenüber konventionellen Datenbanken erzielt werden (1 Liu und Özsu 2009, S. 639).

Beim Single Process Modell hingegen wird ein spezieller Verarbeitungsbereich reserviert. In diesem Bereich werden alle zeitkritischen Operationen durchgeführt. Dadurch werden Kontextwechsel, welche mit viel Overhead verbunden werden vermieden (1 Liu und Özsu 2009, S. 639).



Abb.: 2 Stream Processing Engine

Abbildung 2 stellt die Kernkomponenten eines SPE dar. Der Queue Manager stellt die Speicherressourcen für Input und Output Buffering zu Verfügung. Des Weiteren kümmert er sich um alternative Verarbeitungstechniken, falls benötigte Speicherressourcen nicht garantiert werden können.

Der Storage Manager pflegt Zeiger auf externen Tabellen und führt asynchrone Lese- und Schreibezugriffe aus. Erweiterte Funktionen sind das approximieren der Streams und das Verlagern von Streams auf die Festplatte bei einer schwindenden Speicherverfügbarkeit. Ein Stream Manager interagiert mit der Netzwerkschicht, da die Daten oft über TCP/UDP Sockets gesendet werden. Er ist auch dafür Zuständig, neue Inputs an den Scheduler zu melden und die Umwandlung von Datenformaten durch integrierte Adapter. Schließlich beinhalten SPE noch Query Optimierer, wie zum Beispiel adaptive Plan-Optimierer (4 Valduriez 2004, S. 407–418). Er überwacht den Status der ausführenden Abfragen in Abhängigkeit von Statistiken um dynamisch vorteilhafte Modifikationen an Ausführungsplänen zu machen.(1 Liu und Özsu 2009, S. 641).

# Praktische Umsetzung

# Fazit

Literaturverzeichnis

1 Liu, Ling; Özsu, M. Tamer (Hg.) (2009): Encyclopedia of database systems. New York, NY: Springer (Springer reference). Online verfügbar unter http://dx.doi.org/10.1007/978-0-387-39940-9.

2 Muthukrishnan, S. (2005): Data streams: Now Publ (Foundations and trends in theoretical computer science, 1,2).

3 Vitter, Jeffrey Scott (2001): External memory algorithms and data structures. Dealing with massive data. In: *ACM Comput. Surv.* 33 (2), S. 209–271. DOI: 10.1145/384192.384193.

4 Valduriez, Patrick (2004): Proceedings of the 2004 ACM SIGMOD international conference on Management of data: ACM. Online verfügbar unter http://dl.acm.org/citation.cfm?id=1007568.