Seminararbeit im Studiengang Angewandte Informatik (BSc)

Self-Adaptive Architecture in Stream Processing

Version 1.0 vom 21. Dezember 2019
(Vor Abgabe entfernen)

Leon Meister

285631

meister@uni-hildesheim.de

Betreuer:
MSc Cui Qin, SSE

Arbeitsgruppe Software Systems Engineering • Institut fi¿½r Informatik
Universität Hildesheim • Universitätsplatz 1 • D-31134 Hildesheim

Eigenständigkeitserklärung

Erklärung über das selbstständige Verfassen von "Self-Adaptive Architecture in Stream Processing"

Ich versichere hiermit, dass ich die vorstehende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der obigen Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, habe ich in jedem Fall durch die Angabe der Quelle bzw. der Herkunft, auch der benutzten Sekundärliteratur, als Entlehnung kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen sowie für Quellen aus dem Internet und anderen elektronischen Text- und Datensammlungen und dergleichen. Die eingereichte Arbeit ist nicht anderweitig als Prüfungsleistung verwendet worden oder in deutscher oder einer anderen Sprache als Veröffentlichung erschienen. Mir ist bewusst, dass wahrheitswidrige Angaben als Täuschung behandelt werden.

Hildesheim, den 21. Dezember 2019

Leon Meister

Kurzfassung

Eine kurze Zusammenfassung der Arbeit, die Interesse beim Leser wecken soll.

Abstract

Gerne zusätzlich oder alternativ in Englisch.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcode-Verzeichnis

Abkürzungsverzeichnis

DPS	Data Stream Processing
EDF	Elastic and Distributed DSP Framework
IDC	International Data Corporation
SPS	Stream Processing System

1 Introduction

Situation nowadays -> Lots of data (industry 4.0, other use cases, etc.)

Motivation Goal -> Research question definieren und spezifizieren, discuss the different architectures Struktur erlÃ¤utern

The advancements in technology of the past decades has lead to enormous data creation. Technology has become ubiquitous, with the evolution of cell phones to smartphones and the digitilization of industrial processes, Industry 4.0, causing creation of information to grow exponentially. It is estimated that the global datasphere (**Todo: Explain in glossary**) will reach the size of 175 zettabytes by 2025, as shown in figure 1.1.

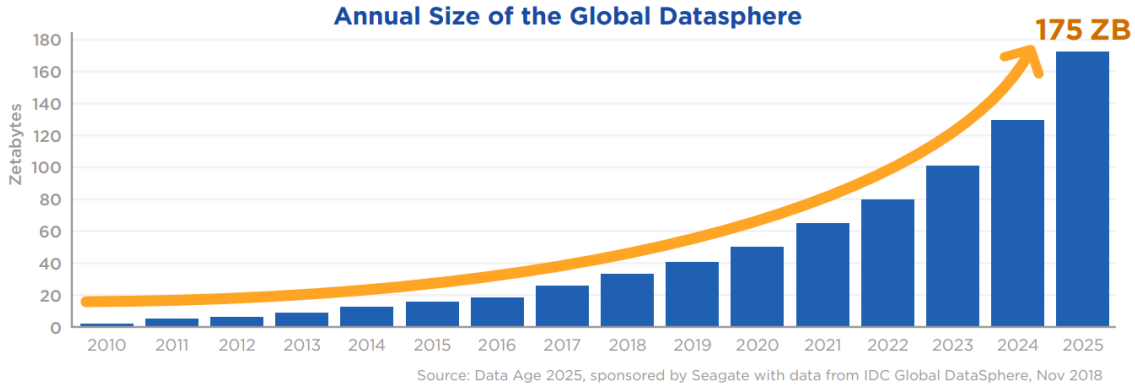

Abbildung 1.1: The Growth of the Global Datasphere [SOURCE]

Data has become an important factor in decision making and optimization in virtually every industry, especially in finances. **TODO: Wieso?** The financial market is dominated by data driven decisions, with emphasis on data processing in a (near) real-time fashion. However, real-time data is becoming of importance in multiple sectors; the International Data Corporation estimates that real-time data will be responsible for a share of 30 percent of the total global datasphere by 2025, as shown in figure 1.2.

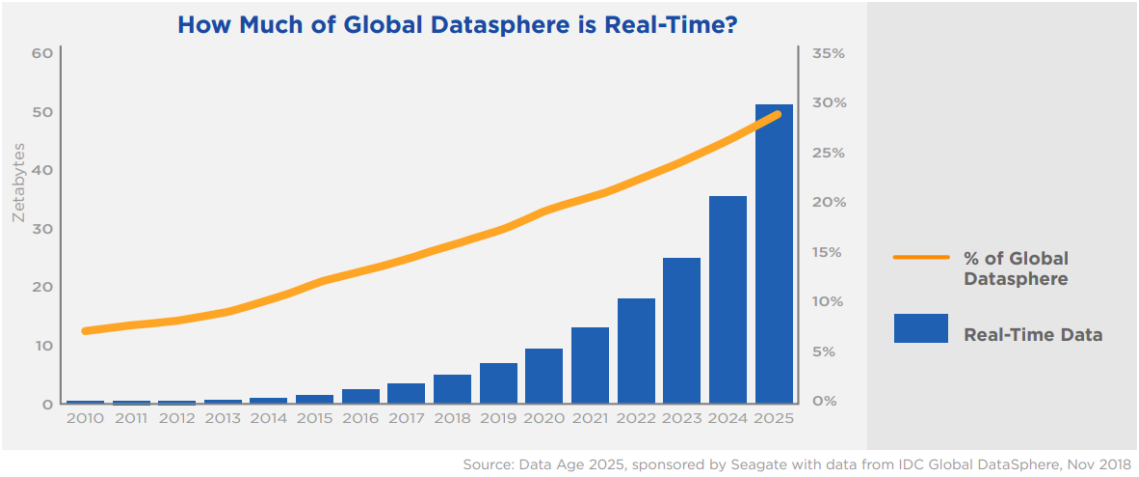

Abbildung 1.2: The growth of real-time data as part of the Global Datasphere [SOURCE]

A global study led by IBM in 2012 has shown that 71 percent of the firms in the financial market use information (including big data) in order to achieve an advantage over their competitors, compared to 36 percent, which IBM has found in an earlier study conducted in 2010. [Todo: SOURCE]

As it is no longer feasible to save all the data before then analyzing it (in batches), due to computational cost and lack of storage capacity, a new approach was designed in order to handle data in a (near) real-time fashion, Stream Processing Systems (SPS).

2 Fundamental Concepts

TODO: In this chapter we... This chapter lays out the fundamental concepts that are needed in order to discuss the topic. Subchapter 2.1 presents and elucidates the basic concept of stream processing, afterwards subchapter 2.2 defined self-adaptive systems and explains their core features.

2.1 Stream Processing

Note (Will be removed): This subchapter should explain why stream processing is needed, when it is applied, how it works

2.1.1 Stream Processing Systems

TODO: Wie sieht Data in SPS aus? discuss with DBMS approach, too

2.1.2 Data Stream Management Systems

vergleichen mit dbms blabla

2.1.3 Requirements for Stream Processing Systems

HIER NOCH DIE EINZELNEN PUNKTE ELABORATEN WIESO DIESE WICHTIG FÃR UNS SIND Due to the nature of the fields in which SPS are used, there are important requirements that SPS should meet in order to be viable, which Stonebraker et al. point out in [SOURCE], which can be summarized as the following:

1. **Keep the Data Moving:** In order to minimize latency, data must not be stored, as these are costly operations.
2. **Handle Stream Imperfections:** Expecting only perfect data is utopian, so one must prepare the system with built-in mechanisms for data that might be missing or out-of-order.
3. **Integrate Stored and Streaming Data:** For an SPS to be able to perform comparisons between "predecessor"data and current data, operators must keep an efficiently manageable state.
4. **Guarantee Data Safety and Availability:** Recovering from a failure is detrimental for real-time data processing, so a system must be in place to guarantee the highest availability possible.
5. **Process and Respond Instantaneously:** Systems must be highly optimized in order to provide (near) real-time responses.

An SPS takes in one or multiple continuous streams of data, each element of the stream then gets processed by a number of operators and eventually, the SPS puts out a stream of processed data.

In order to increase efficiency, an SPS can, if (computational) resources are available, create replicas of operators to introduce parallelity. Conversely, if there is little input, it may also reduce the amount of replicas in order to save or free up additional resources. *Figure 2.3 should be here*

2.2 Self-Adaptive Systems

This subchapter should explain what self-adaptive systems are and how they function. What kind of self-adaptive systems are around?

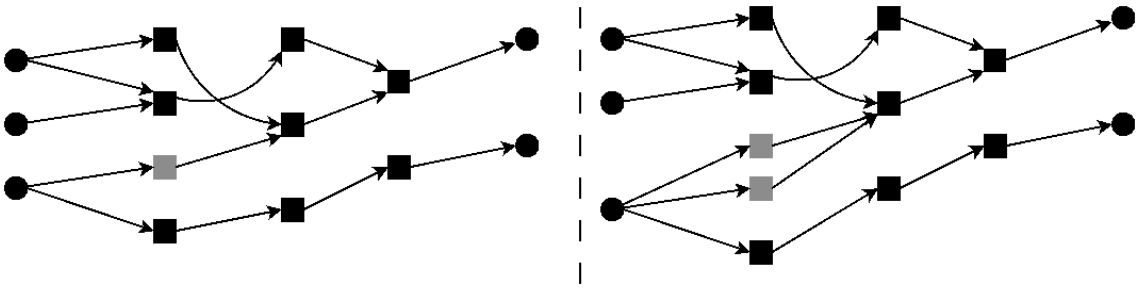

Abbildung 2.1: Left: An example for an SPS displayed as a directed acyclic graph. Right: Same SPS with introduced parallelity in one operator, marked gray for visibility. Circles are input/outputs, squares are operators, arrows are streams.

2.3 MAPE-K Loop

Explain the MAPE-K Loop as it is a valuable basis for many different approaches in adaptive systems

3 Approaches for Self-Adaptive Architectures in Stream Processing

Explain that this chapter showcases a few select strategies, which are then elaborated on further in the subchapters Question: Even more approaches? e.g. Master-Slave attern or Coordinated Control pattern (Both MAPE based)?

3.1 Dhalion

Quick Introduction to Dhalion, this chapter will deal with the Dhalion paper.

3.1.1 An Outline of Heron

Small outline of Heron, as Dhalion is built on top of Twitter's Heron.

3.1.2 Dhalion's Architecture

Explanation of Dhalion's Architecture

3.1.3 Evaluation of Dhalion

Present the paper's findings and evaluations of the Dhalion System.

3.2 Hierarchical Control Architectures

Quick Introduction to hierarchical control architectures, this chapter deals with the Cardellini paper (An example for such an architecture)

3.2.1 Elastic and Distributed DSP Framework

Explanation of the EDF Architecture, their approaches

3.2.2 Possible Solutions for Controlling the Adaptation of Data Stream Processing Operators

3.2.3 Evaluation of EDF

4 Summary And Conclusion

4.1 Summary

Summarize the paper

4.2 Conclusion

Conclude the paper

A Anhang

A.1 Beispiele

Zitat ohne Seitenangabe: [BKPS04]

Zitat mit Seitenangabe: [BKPS04, S.1]

Referenz eines Glossareintrags: Computer

Eine normale Liste:

- Ein Punkt
- Ein anderer Punkt

Eine nummerierte Liste:

1. Erstens
2. Zweitens
3. ...

Eine Tabelle:

Tabelle A.1: Eine Tabelle

Spalte 1	Spalte 2	Spalte 3
1	2	3

Fetter Text

Kursiver Text

Eine Referenz: Siehe Tabelle A.1: Eine Tabelle auf Seite 7.

Ein Bild:

Abbildung A.1: Das Logo der SUH

Eine abgesetzte Formel:

$$\sum_{n=0}^{3} n = 6 \tag{A.1}$$

Eine Formel im Fließtext: $2^2 = 4$

Ein Listing aus einer Datei:

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         System.out.println("Hello World!");
4     }
5 }
```

Listing A.1: HelloWorld

Ein 'on-the-fly' erstelltes Listing:

```
1 (Sun Sep 13 23:02:20 2009): ODBC Driver <system32>\wbemdr32.dll not present
2 (Sun Sep 13 23:02:20 2009): Successfully verified WBEM OBDC adapter (incompatible version
      removed if it was detected).
3 (Sun Sep 13 23:02:20 2009): Wbemupgd.dll Registration completed.
```

Listing A.2: Beispiel eines Log-Eintrags

Glossar

Computer ist ein elektronisches Gerï¿½t, das Daten verarbeitet.

Literaturverzeichnis

[BKPS04] Böckle, Günther ; Knauber, Peter ; Pohl, Klaus ; Schmid, Klaus: *Software-Produktlinien*. Dpunkt-Verlag, 2004