BIG DATA

An Introduction To The Fields Of Data Engineering, Development And Architecture Of Data-Intensive Applications.

—Winter Semester 2018 —

Munich, July 21, 2018

Marcel Mittelstädt
Mail: mittelstaedtmarcel@googlemail.com
Web: www.marcel-mittelstaedt.com

Cooperative State University Baden-Wuerttemberg

Preface

This lecture will give you a brief introduction to so what is called 'Big Data'. We will quickly refresh the basics about databases, data models and data processing you have learned so far and compare those to the distributed world of Big Data.
After that we will take a deep dive into the foundations of distributed data storages and data processing as well as the belonging concepts of reliability, scalability, replication, partitioning, batch and stream processing.
Later on we will take a look at the most common used software and frameworks (mostly the hadoop ecosystem).
At the end, as you know the basic concepts and you are able to setup and work with distributed environments and huge data sets, there will be a short introduction to data science.

At the end of each lesson, there will be some hands-on exercises, which we will start together and which have to be finished till the next week. This lecture will only be about 36 hours in 12 weeks (1 slot each week), which is very little time to cover such an extensive topic. So pay close attention and if you can't keep up, feel free to ask questions at the end of each lesson.

You can find:

- this **script** (and LaTeX-sources),
- **slides** presented within the lecture,
- **excercises** and **solutions**,
- **docker images**, **scripts** as well as **sample data sets**

here:

```
https://github.com/marcelmittelstaedt/BigData
```

You can just download everything directly or install `git` and get everything by using:

```
git clone https://github.com/marcelmittelstaedt/BigData.git
```

and

```
git pull
```

to get the most recent version.

If you find any mistakes or misspellings feel free to send me a mail (**mittelstaedtmarcel@googlemail.com**) or if you are able to, commit a push request.

One last point, as you may have noticed, Microsoft as well as other commercial vendors successfully fail at developing and providing adequate solutions for highly data-driven applications, so almost any software or framework you will encouter during this lecture or later, will be open-source and only be runnable on a UNIX-based operating system. Either you are already familiar with UNIX (lucky you), otherwise you will learn something valuable that will improve your life.

> *"Microsoft is not the answer. Microsoft is the question. NO is the answer."*
>
> — Erik Naggum ✝ , *Philantropist and Developer (Emacs, Lisp and SGML)*

Contents

1 Introduction To Big Data

> *"Big Data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it."*
>
> — Dan Ariely, *Professor of Psychology and Behavioral Economics, Duke University*

1.1 Definition of Big Data

Lorem Ipsum

1.2 Challenges

Lorem Ipsum

1.3 Use Cases

Lorem Ipsum

1.4 Dissociation Datawarehousing

Lorem Ipsum

2 Foundation Of Distributed Data Systems

"I'm not telling you it's going to be easy - I'm telling you it's going to be worth it."

— Arthur L. Williams Jr., *Founder of Primerica Financial Services*

In this chapter we will go trough the foundation of data systems, requirements and concepts which apply to any (data-driven) system. This covers in particular following topics:

- Section2.1 Requirements of **Data-Driven Systems**, e.g.
  - 2.1.1 Scalability,
  - 2.1.2 Availablity/Reliabaility,
  - 2.1.3 Maintainability,
  - 2.1.4 Extensability amongst others.
- Section 2.2 **Storage Concepts** for databases
- Section 2.3 **Data Models And Access** concepts
- Section 2.4 **Challenges Of Distributed Data Systems**, e.g.
  - 2.4.1 Partitioning,
  - 2.4.2 Replication,
  - 2.4.3 Transactions,
  - 2.4.4 Consistency amongst others.

At the end you will have a basic understanding about the difference between common and distributed systems and databases, the basic concepts of each of them and which one theoretically fits best to solve a certain problem. A more hands-on deep-dive

into related software, frameworks as well as specific problems and use cases will be demonstrated later in chapter XXX.

2.1 Data-Driven Systems

When we think about data-driven systems, we mostly think about the same requirements we expect of any other data system we already know:

- **Data Storage**: We need to store data and also need to be able to find it again later (*database*).
- **Data Querying**: We need to be able to query and filter data efficiently in certain kinds of ways (*transaction and indices*).
- **Retention and Performance**: We want results fast, especially of expensive read operations (*caching*).
- **Data Processsing**: We want to be able to process a huge amount of data (*batch processing*) as well as process data asynchronously (*stream processing*).

This sounds quite obvious, but remember those requirements are still the same as for the first database CODASYL[1] back in the 1960's. Even though there are and have been a lot of databases back in time, each of them with a diverse purpose and different approaches to solve e.g. indexing or caching - all of them still match those same requirements. Certainly those data systems evolved much further, especially within the last years, you may noticed:

- Relational Databases being able to handle NoSQL data (e.g. even "retirees" like IBM DB2[2] or Oracle[3]) as well as NoSQL databases being able to handle traditional SQL (e.g. ToroDB[4]) or
- databases becoming message queues (e.g. RethinkDB[5] or Redis[6][7]) and the

[1] https://en.wikipedia.org/wiki/CODASYL
[2] (IBM18), https://www.ibm.com/support/knowledgecenter/en/SSEPEK_11.0.0/json/src/tpc/db2z_jsonfunctions.html
[3] (ORC18), https://docs.oracle.com/database/121/ADXDB/json.htm
[4] (TOR18), https://www.torodb.com
[5] (RDB18), https://rethinkdb.com/docs/changefeeds/
[6] (RUD18), https://redis.io/commands/rpoplpush
[7] (Joh14), see 5. Adopting Redis for Application Data

other way around message queueing systems become databases (e.g. Apache Kafka[8]).

As you can see, boundaries between traditional databases and data-driven applications get blurred and in the same way more diversified. There is no one-size-fits-all solution, e.g. like you can find back in the past in the 1990's or early 2000s. At that time monolithic single-, 2 and 3-tier, architectures were state-of-the-art (see Figure 2.1 left-hand side).
Usually the **database layer** was represented by a data store like MySQL, Oracle, DB2 or even just files containing data stored on the local disk.
The **application layer** was usually a monolithic application developed in languages like PHP, Perl, C++ or Java and running on a web- or application server (e.g. Apache HTTP Server or IBM WebSphere).
And last but not least the **client layer**: a web browser like nowadays.
If you take a look at Figure 2.2 on page 6 you can see an example of this time

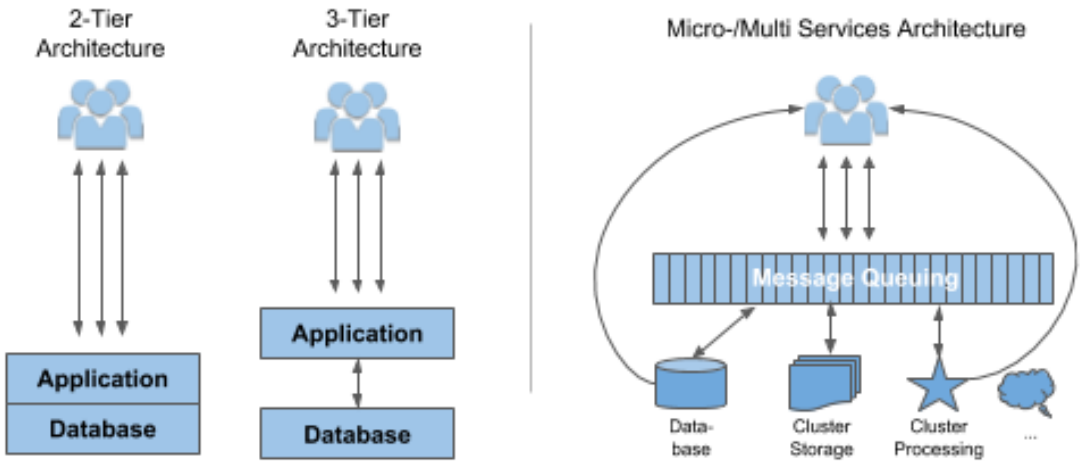

Figure 2.1: Schema - Application Architectures

you may know: ebay.com. They have used the classical 3-tier architecture as well: Oracle as the database running on Solaris as OS[9], C++ as application code running

[8](KFK18), https://kafka.apache.org/10/documentation/streams/developer-guide/interactive-queries.html
[9]OS, Operating System

on the Microsoft IIS[10] web server. As you can already guess, this architecture won't

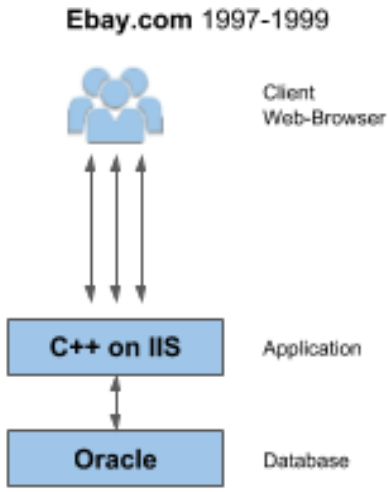

Figure 2.2: Schema - Architecture Ebay.com 1997-1999

scale very well today, in fact the only way to scale this application was to upgrade the single server (*scale up vertically*), in case of ebay.com they once switched from commodity hardware to a very pricey mainframe server (Sun Enterprise E10000[11]) to buy some time. But as you may have noticed there are much more ovious issues, e.g. if you think about:

- **Redundancy** does not exist at all (if the database itself or it's server suffers an outage the whole system will be unavailable.
- **Extensability** is not existent, the system is only able to scale up vertically and if this is needed, a downtime is inevitable, as no part of the data system is neither replicated nor virtualized.
- **Maintainablity** is also very limited as any maintenance of the database will require a certain amount of time in which the application will be unavailable.

But we will dicuss this later in the following chapters.

The previously mentioned issues are already sufficient reason but also the increasing

[10](IIS18), *Microsoft Internet Information Services, an extensible web server created by Microsoft for use with the Windows NT family.*

[11](SP06), slide 11

amount of data as well as required features of data systems these days (becoming more diversified in the same way) make it unfeasible to rely on a single tool. Instead each functionality is usually broken down into parts which can be done efficiently by suitable tools which are sticked together within the applications itself. This could probably look like as you can see in Figure 2.1 (right-hand side) on page 5, but that's just one plain example of many other.

Instead of having one single-purpose data store, there are several tied together, each one of them to fulfill it's specific part within the whole data system but all of them tied together as one application.
As you can see one part of the data system could be: a **database** (like you saw in Figure 2.2 on page 6), e.g. to store and serve:

- user data (e.g. in case of an application with login)
- product data (e.g. in case the applications is a web shop)
- user generated content (e.g. in case the application is a newspage, blog or forum)

Another part could be **cluster storage** like Hadoop, which could:

- keep a complete history of all raw data (e.g. page requests of a website or measured values of a sensor)
- serve for batch processing (e.g. crunching the whole history of data, which is impossible for a single database, as it couldn't even save the whole data and certainly wouldn't be able to process it later on)
- serve for analytic and reporting purposes (e.g. reports of how many people have visited the website within the last year based on the raw data)

Also frequently seen, an analytical **cluster processing engine**, e.g. Spark or Flink to:

- process data gathered in real-time (e.g. every page request of a website) for

analytical purposes

- use processed data, to run data science models on it (e.g. to serve targeted advertisements or customized content to a user on a website, based on his last page requests, browser user-agent or device)
- ...

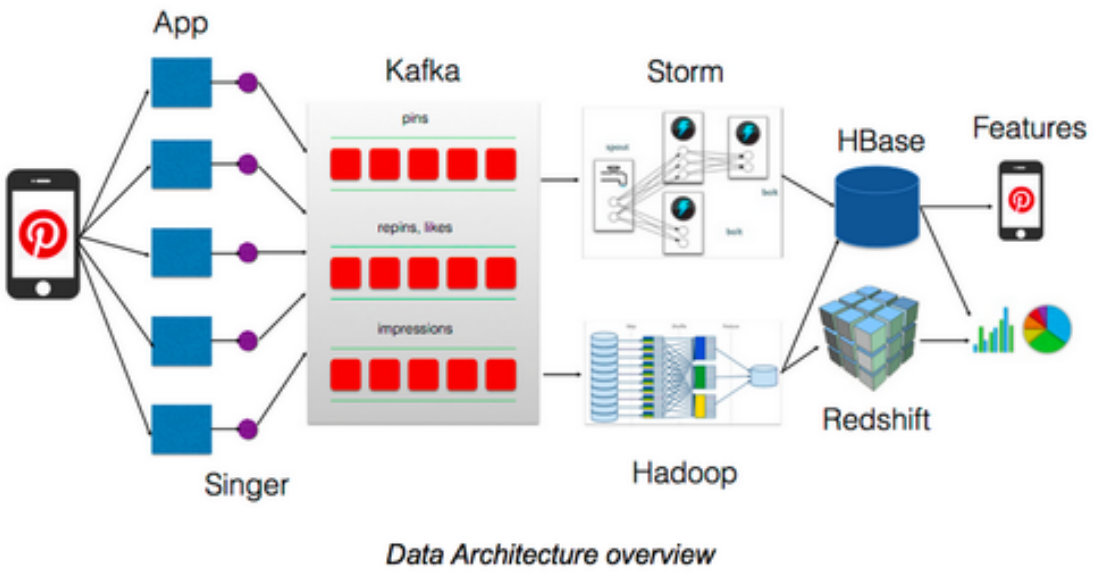

Figure 2.3: Schema - Architecture pinterest.com

If you take a look at Figure 2.3, you can see a comparable data system architecture, implemented by pinterest.com[12]. Redis as a database on top of the hadoop cluster storage to serve for analytical purposes (e.g. ad serving of pinterest's adbuyers) or HBase on top of Hadoop cluster storage and Storm to serve features for the actual end-user of pinterest.com.

But we need to take care here: by creating new and more complex data systems from special purpose data systems, complexity is growing with it. How to ensure the system is avalaible with a reliable performance if something crashes? How to make sure data remains consistent and complete if things go wrong? How to scale

[12](PIN18), Architecture of Giants: Data Stacks at Facebook, Netflix, Airbnb, and Pinterest

the data system to be able to handle increased load?...
There are many apects which are crucial and influence the architecture of a data system like regulary constraints like data security, location of servers, SLA's[13] or existing devlopment and operation skills - which very much depend on the specific situation.
Within the next chapters we will focus on the aspects which must be taken into account by any data system:

- Scalability (Chapter 2.1.1),
- Availablity/Reliabaility (Chapter 2.1.2),
- Maintainability (Chapter 2.1.3) and
- Extensability (Chapter 2.1.4).

As many people and companies usually mess around with those terms, firstly we will develop a clear understanding on what they mean and later on take a closer look on how to apply algorithms, development and architectures to fulfill them appropriately.

2.1.1 Scalability

As is evident from the introduction of this chapter: the fact that a system is working reliable today doesn't mean it will necessarily work reliable in the future. The data system of ebay.com in 1999 was maxed out at handling **50.000** active listings[14], imagine how the system would behave today at handling **1 billion** active listings[15]. Obviously handling increased load (e.g. a larger amount of data needed to store or request to handle) is a major factor of a scalable data system.

[13](WKS18), *Service Level Agreement, a commitment between a service provider and a client. Particular aspects of the service – quality, availability, responsibilities – are agreed between the service provider and the service user.*
[14](SP06), slide 9
[15](EBA18)

Scalability is the capability of data system to handle a growing amount of load (e.g. a larger amount of data needed to store or requests to handle). A data system is considered scalable if its capable of increasing it's total through-/output under an increased load when resources (typically hardware) are added.

Note that, "*scalability*" isn't a binary tag that could be attached to a data system. It's pointless to say "*a data system is scalable*" as well as "*a data system is not scalable*", in either way you must think about "*If the load of a data system grows in a certain way, what are the options on the table for coping with the growth?*" and "*How can we add ressources (hardware) to be able to handle the additional load?*". Therefore we will discuss the parameters and definition of **Load** (Section 2.1.1.1) and **Performance** (Section 2.1.1.2) within the next section as well as **Approaches** for coping with load to achieve a certain performance (Section 2.1.1.3).

2.1.1.1 Load

To get an idea what *load* of a data system actually means, how it could be described an measured, we will take a nother look at the example of ebay.com discussed so far. If we take a look back at the architecure of ebay at 1997-1999 (Figure 2.2 on page 6) we can already guess with increasing load (page requests to certain items listed on ebay.com and in this way calls to the database through the application) will reach its limit at the maximum amount of read requests the oracle database can serve. As the application tier (web server) has already been scaled horizontally to multiple nodes, the oracle database server reached its limit of physical growth in November 1999. So ebay added an additional server to not just elmininate the SPOF[16] to be able to failover but also they have splitted the database to be able to logically partition it into separate instances and in this way be able to scale horizontally. This was achieved in 2001 by splitting items by categories, as you can see in Figure XXX. In this simple way it was possible to distribute the load (mostly page requests for items) in an "equal" way to different physical nodes. Later on they segmented whole databases into functional areas like hosts for item, user, account or transaction data as well as partition the data by typical usage characteristics to

[16](WPS18), Single Point Of Failure

scale horizontally.

They obviously did furher optimizations at the whole data system to be able to cope with the increasing load, like disabling transactions, moving CPU-intensive[17] work to the application tier (e.g. joins, referential integrity or sorting), extensive use of prepared statements...but as this techniques are not mainly specific for distributed systems and some of them not even recommened nowadays, we won't focuse on them within this lecture.

In the example of ebay.com, requests per item and category could be a valuable *load parameter* for discussing scalability, since it determines the database requests per data record and partition - as proven by the the data system structure of ebay at 2002 as we see. Your or other data systems you have seen so far most likely have very different characteristics, but you can apply similar principles to reason about their load.

The **load** of a data system is a measurement of the amount of computational work it performs (depending on the architecture in-place), e.g. the number of (concurrent) reads from a data storage, writes to a data storage or the ratio between reads and writes. The maximum load is defined by the weakest part of the architecture (=*bottleneck*).

2.1.1.2 Performance

Now that we have described what *load* of a data system means as well as what *load parameters* could be, we will examine more closely what happens when the load increases. Usually there are two important cases you need to think about while developing data systems:

- The load parameter increases, but all ressources (e.g. number of server, CPU

[17] CPU, In computing, a processor or processing unit is an electronic circuit which performs operations on some external data source, usually memory or some other data stream is called central processing unit

or memory) stay the same - how is the performance of the data system affected?

- The load parameter increases - how much do you need to increase the ressources (e.g. number of server, CPU or memory) to keep the performance stay the same?

But how to answer them? Therefore we need performance numbers. In case of data systems measurement, methods usually are **throughput** (number of records that can be handled), e.g.:

- read/writes per second (in case of MongoDB up to 100.000 read/writes per second)
- messages processed (in case of Apache Kafka and Linkedin more than 2 million records per second on just 3 nodes)
- data processed (in case of Apache Hadoop and MapReduce terabytes of data within several seconds)

or if your buidling a data-system which works as the backend of a end-user facing application like a website, it's more about the **response time**, which means the time between sending a request and receiving the response. For instance if we think about the example of ebay.com within the previous chapters, as of 2016 they had 1 billion items accessible at any time, needed to serve 2 billion page requests each day and had to fulfill each of them within fractions of a second.

Regardless of throughput or response time - if we think about performance we don't think about a single number, but a distribution of values that we can measure. If you will repeat the same page request, read or writes on the same data system: response time and throughput will inevitable vary somehow. There are simply too many factors you usually cannnot contain:

- network issues (e.g. latency or TCP packet loss and retransmission)
- os issues (e.g. a page faults, context switches or running background processes)
- physical issues (e.g. a damaged disk/ssd or overheating of a CPU and, associated therwith a decreased processing power)

Text

Performance of a data system is defined by system throughput and response time, e.g. number of transactions (like read/write operations), processed records (like aggregations for analytical purposes) or even system commands (like an *update statistics* or rebalancing of several data nodes) under a given workload and for a specific timeframe. It usually depends on a variety of influencable as well as uninfluencable factors of the system itself, like network latency, a page fault or damaged disk.

2.1.1.3 Approaches

2.1.2 Availablity/Reliabaility

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.1.3 Maintainability

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.1.4 Extensability

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.2 Storage Concepts

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.3 Data Models And Access

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.4 Challenges Of Distributed Data Systems

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.4.1 Partitioning

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.4.2 Replication

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.4.3 Transactions

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

2.4.4 Consistency

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3 Data Processing On Distributed Systems

Begin at the beginning, the King said gravely, "and go on till you come to the end: then stop."

—Lewis Carroll, *Alice in Wonderland*

3.1 Batch Processing

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.2 Micro-Batch Processing

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo

duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.3 Stream Processing

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.4 Message Queuing

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

3.5 ETL and Workflow Automation

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et

dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

4 Software and Frameworks

> *Begin at the beginning, the King said gravely, "and go on till you come to the end: then stop."*
>
> —Lewis Carroll, *Alice in Wonderland*

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5 Data Science

"Big data is not bout the data."

— Gary King, *Harvard University*

5.1 Data Cleaning, Integration and Preparation

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.2 Data Visualization

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.3 Regression

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.4 Classification

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.5 Clustering

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.6 Association

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.7 Neural Networks

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

5.8 DataScience on Distributed Systems

Spark ML PySpark

6 Outlook

"Big data is not bout the data."

— Gary King, *Harvard University*

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

7 Appendix

Abkürzungsverzeichnis

CPU	Central Processing Unit
DHBW	Duale Hochschule Baden-WŸrttemberg
OS	Operating System
SLA	Service Level Agreement
SPOF	Single Point Of Failure

List of Figures

Auszugverzeichnis

Bibliography

[EBA18] *What we do - eBay is where the world goes to shop, sell, and give.).* https://www.ebayinc.com/our-company/who-we-are/. Version: July 2018, Abruf: 15. 07. 2018 15

[IBM18] *IBM DB2 Knowledgebase.* https://www.ibm.com/support/knowledgecenter/en/SSEPEK_11.0.0/json/src/tpc/db2z_jsonfunctions.html. Version: July 2018, Abruf: 13. 07. 2018 2

[IIS18] *Microsoft IIS Homepage.* https://www.iis.net/. Version: July 2018, Abruf: 14. 07. 2018 10

[Joh14] Johanan, Joshua: *Buidling Scalable Apps With Redis And NodeJS.* Packt Publishing, 2014 7

[KFK18] *Apache Kafka User Documentaion.* https://kafka.apache.org/10/documentation/streams/developer-guide/interactive-queries.html. Version: July 2018, Abruf: 13. 07. 2018 8

[ORC18] *IBM DB2 Knowledgebase.* https://docs.oracle.com/database/121/ADXDB/json.htm. Version: July 2018, Abruf: 13. 07. 2018 3

[PIN18] *The eBay Architecture (Randy Shoup and Dan Pritchett).* https://blog.keen.io/architecture-of-giants-data-stacks-at-facebook-netflix-airbnb-and-pinte
Version: April 2018, Abruf: 14. 07. 2018 12

[RDB18] *RethinkDB User Documentaion.* https://rethinkdb.com/docs/changefeeds/. Version: July 2018, Abruf: 13. 07. 2018 5

[RUD18] *Redis User Documentaion.* https://redis.io/commands/rpoplpush. Version: July 2018, Abruf: 13. 07. 2018 6

[SP06] Shoup, Randy ; Pritchett, Dan: *The eBay Architecture (Randy Shoup and Dan Pritchett)*. http://www.addsimplicity.com/downloads/eBaySDForum2006-11-29.pdf. Version: November 2006, Abruf: 14. 07. 2018 11, 14

[TOR18] *ToroDB Website*. https://www.torodb.com/. Version: July 2018, Abruf: 13. 07. 2018 4

[WKS18] *Wikipedia - SLA*. https://en.wikipedia.org/wiki/Service-level_agreement. Version: July 2018, Abruf: 13. 07. 2018 13

[WPS18] *Single Point of Failure*. https://de.wikipedia.org/wiki/Single_Point_of_Failure. Version: July 2018, Abruf: 20. 07. 2018 16