Schnelle Anfrageverarbeitung im Big Data Umfeld

Alexander Ponomarenko

HAW Hamburg, Master Informatik

Hauptseminar Wintersemester 2016

20.12.2016

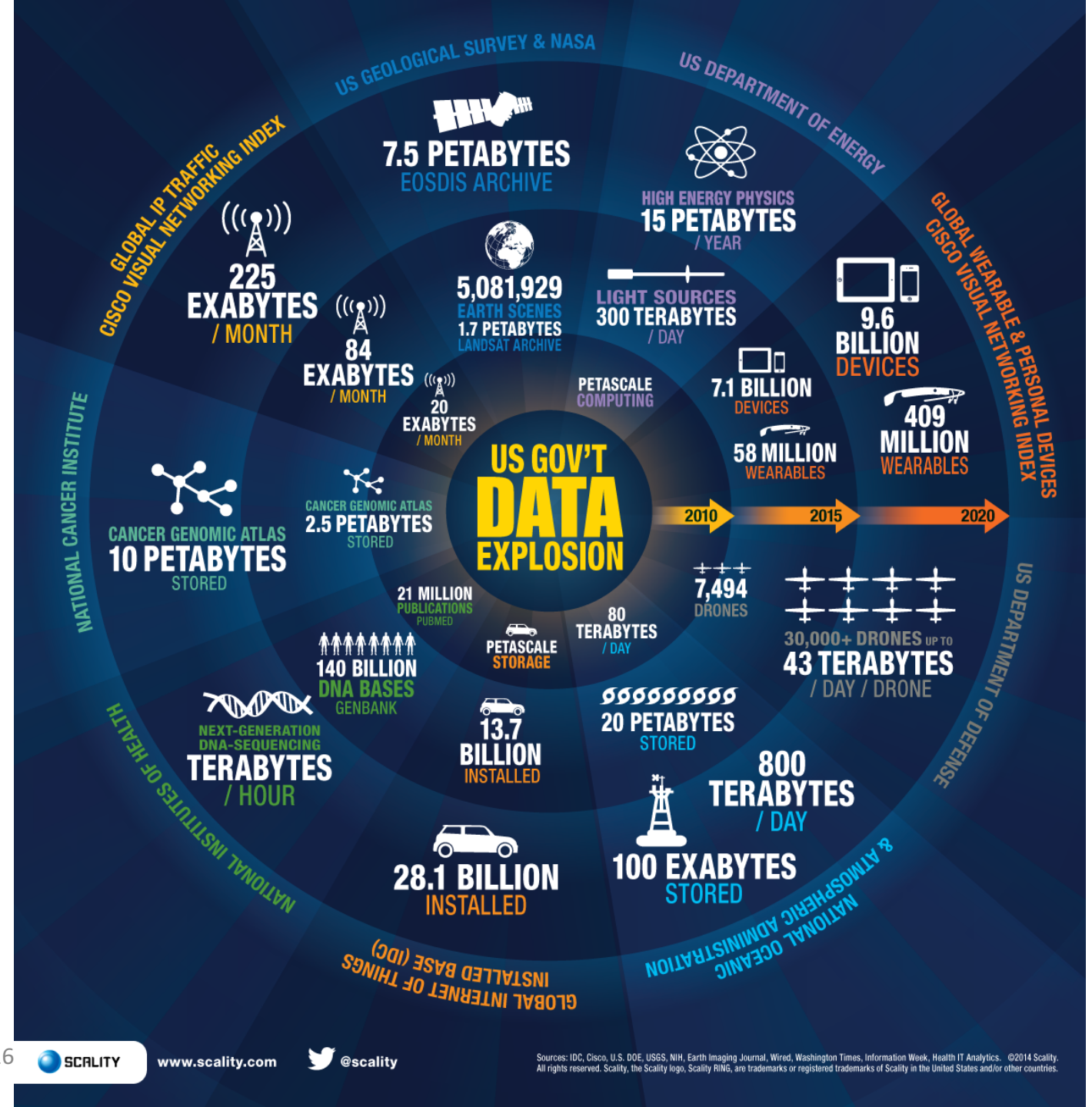
Agenda

- Motivation
- Big Data und NoSQL
- Approximative Anfrageverarbeitung
- BlinkDB
- SnappyData
- Ziele
- Risiken

Motivation [2]

- Datenmengen nehmen zu
  - 2012: 2,8 Zettabyte
  - 2020: 40 Zettabyte
- Große Datenmengen: Auswertung dauert lange

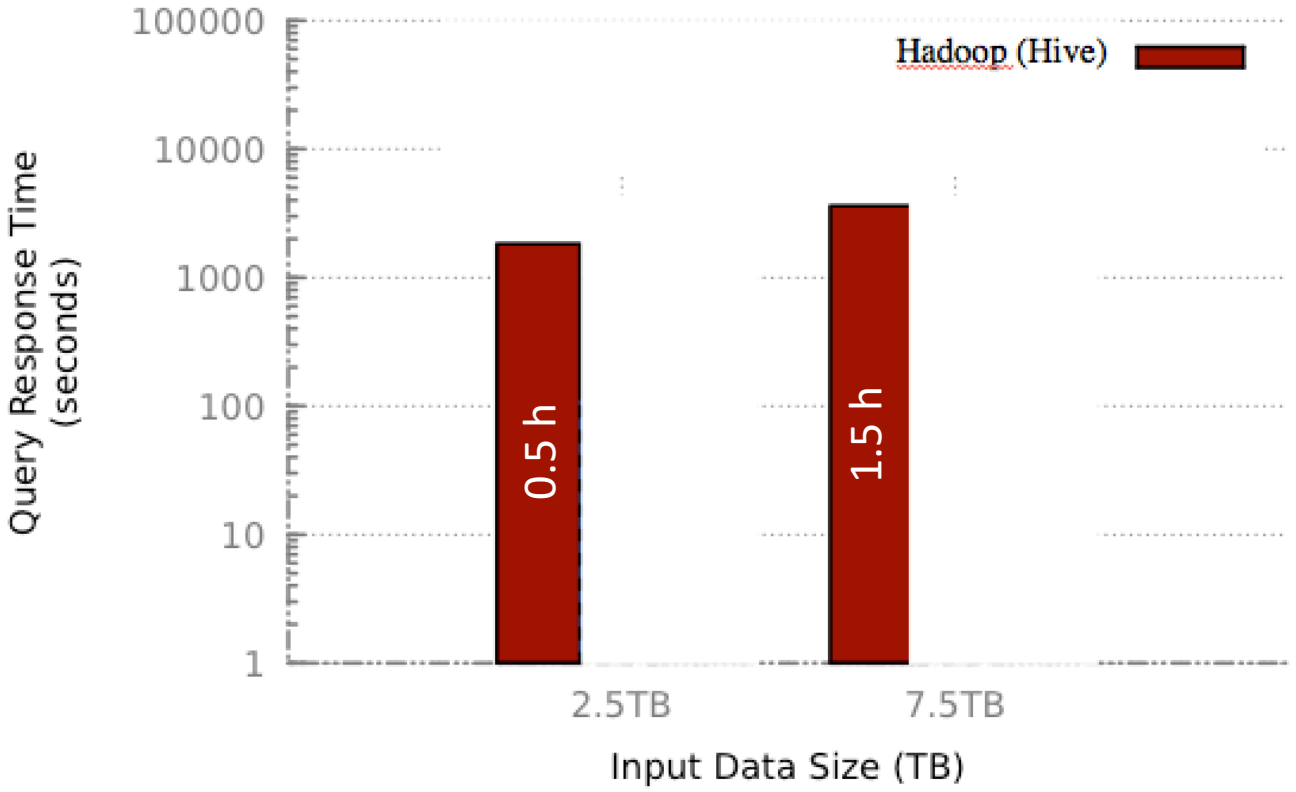
1000	Kilobyte
1000^2	Megabyte
1000^3	Gigabyte
1000^4	Terabyte
1000^5	Petabyte
1000^6	Exabyte
1000^7	Zettabyte


US GEOLOGICAL SURVEY & NASA
7.5 PETABYTES
EOSDIS ARCHIVE
US DEPARTMENT OF ENERGY
HIGH ENERGY PHYSICS
15 PETABYTES
/ YEAR
GLOBAL IP TRAFFIC
CISCO VISUAL NETWORKING INDEX
225
EXABYTES
/ MONTH
84
EXABYTES
/ MONTH
5,081,929
EARTH SCENES
1.7 PETABYTES
LANDSAT ARCHIVE
LIGHT SOURCES
300 TERABYTES
/ DAY
9.6
BILLION
DEVICES
GLOBAL WEARABLE & PERSONAL DEVICES
CISCO VISUAL NETWORKING INDEX
20
EXABYTES
/ MONTH
PETASCALE
COMPUTING
7.1 BILLION
DEVICES
409
MILLION
WEARABLES
58 MILLION
WEARABLES
NATIONAL CANCER INSTITUTE
CANCER GENOMIC ATLAS
10 PETABYTES
STORED
CANCER GENOMIC ATLAS
2.5 PETABYTES
STORED
US GOV'T
DATA
EXPLOSION
2010
2015
2020
21 MILLION
PUBLICATIONS
PUBMED
80
TERABYTES
/ DAY
7,494
DRONES
30,000+ DRONES UP TO
43 TERABYTES
/ DAY / DRONE
US DEPARTMENT OF DEFENSE
PETASCALE
STORAGE
140 BILLION
DNA BASES
GENBANK
NEXT-GENERATION
DNA-SEQUENCING
TERABYTES
/ HOUR
13.7
BILLION
INSTALLED
20 PETABYTES
STORED
800
TERABYTES
/ DAY
NATIONAL INSTITUTES OF HEALTH
28.1 BILLION
INSTALLED
100 EXABYTES
STORED
NATIONAL OCEANIC
& ATMOSPHERIC ADMINISTRATION
GLOBAL INTERNET OF THINGS
INSTALLED BASE (IDC)
SCALITY
www.scality.com
@scality
Sources: IDC, Cisco, U.S. DOE, USGS, NIH, Earth Imaging Journal, Wired, Washington Times, Information Week, Health IT Analytics. ©2014 Scality.
All rights reserved. Scality, the Scality logo, Scality RING, are trademarks or registered trademarks of Scality in the United States and/or other countries.

[1]

Motivation [3]

- 7,5 Terabyte Daten
- Durchschnittswert berechnen
- Verteilt auf 100 Amazon EC2
- Hive/Hadoop



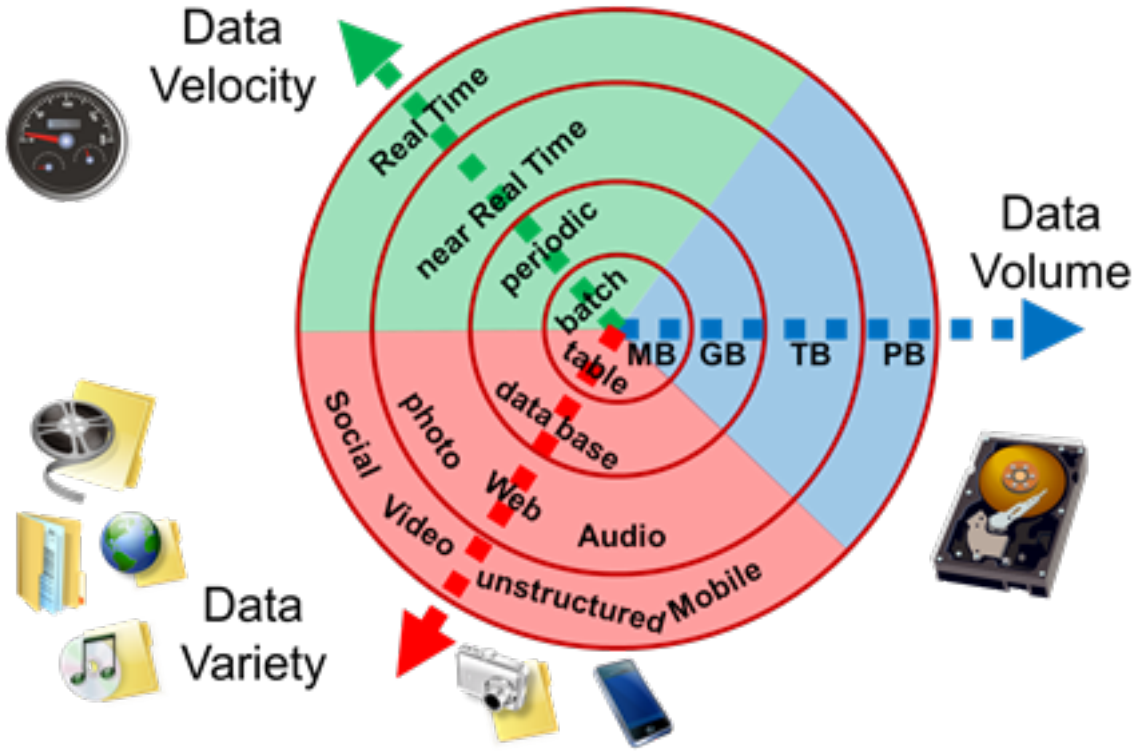
Motivation

- Auswertung dauert zu lange
  → Ergebnis möglicherweise nutzlos
- Wo werden schnelle Antworten benötigt?
  - Preisvergleiche
  - Kurzfristiger Handel mit Wertpapieren
  - Soziale Netzwerke
  - …

Big Data [4, 5]

Pro Minute:
Google – über 2 Millionen Suchabfragen
Amazon – 80.000$ Umsatz
YouTube – 72 Stunden Videomaterial

Pro Minute:
Facebook – 2,5 Millionen Inhalte
Twitter – 600.000 Tweets
Instagram – 220.000 Fotos
Email – über 200 Millionen



Daten sind unterschiedlich und nicht strukturiert

Relationale DB dafür nicht ausgelegt

NoSQL[6, 7]

- Nichtrelationale Daten-Modelle
- Unnötige SQL-Features
- Sehr große Datenmengen
- Verteilte DB-Systeme: Schnelles Lesen und Schreiben
- Hochverfügbarkeit wichtiger als Konsistenz

Noch schneller??

[B1]

100 TB auf 1000 Knoten [9]

½ - 1 Stunde

Hard Disks

1 - 5 Minuten

Memory

1 Sekunde

?

Anfrageverarbeitung auf Samples

Anfrageverarbeitung auf Samples [10, 11]

- Sample: „Stichprobe“ / „Beispiel“ / „Muster“
  → Also ein Teil der originalen Daten
- Nur ein Teil der Daten wird ausgewertet
- Dadurch schneller
- Samples werden vorberechnet
- Antwort ist ungenau
- Aber der Fehler der Antwort ist bekannt

Samples [11]

ID	Stadt	Gehalt
1	Hamburg	50.000 €
2	Berlin	62.492 €
3	Hamburg	78.212 €
4	Hamburg	120.242 €
5	Berlin	98.341 €
6	Hamburg	75.453 €
7	Hamburg	60.000 €
8	Berlin	72.492 €
9	Berlin	88.212 €
10	Hamburg	92.242 €
11	Berlin	70.000 €
12	Hamburg	102.492 €

Was ist das Durchschnittsgehalt aller Einträge in der Tabelle?

80.848,17 €

Samples [11]

ID	Stadt	Gehalt
1	Hamburg	50.000 €
2	Berlin	62.492 €
3	Hamburg	78.212 €
4	Hamburg	120.242 €
5	Berlin	98.341 €
6	Hamburg	75.453 €
7	Hamburg	60.000 €
8	Berlin	72.492 €
9	Berlin	88.212 €
10	Hamburg	92.242 €
11	Berlin	70.000 €
12	Hamburg	102.492 €

Sample

Was ist das Durchschnittsgehalt aller Einträge in der Tabelle?

ID	Stadt	Gehalt	Rate
1	Hamburg	50.000 €	1/4
5	Berlin	98.341 €	1/4
8	Berlin	72.492 €	1/4

~~80.848,17 €~~

73.611,00 € +/- 7.237,17 €

Samples [11]

ID	Stadt	Gehalt
1	Hamburg	50.000 €
2	Berlin	62.492 €
3	Hamburg	78.212 €
4	Hamburg	120.242 €
5	Berlin	98.341 €
6	Hamburg	75.453 €
7	Hamburg	60.000 €
8	Berlin	72.492 €
9	Berlin	88.212 €
10	Hamburg	92.242 €
11	Berlin	70.000 €
12	Hamburg	102.492 €

Sample

Was ist das Durchschnittsgehalt aller Einträge in der Tabelle?

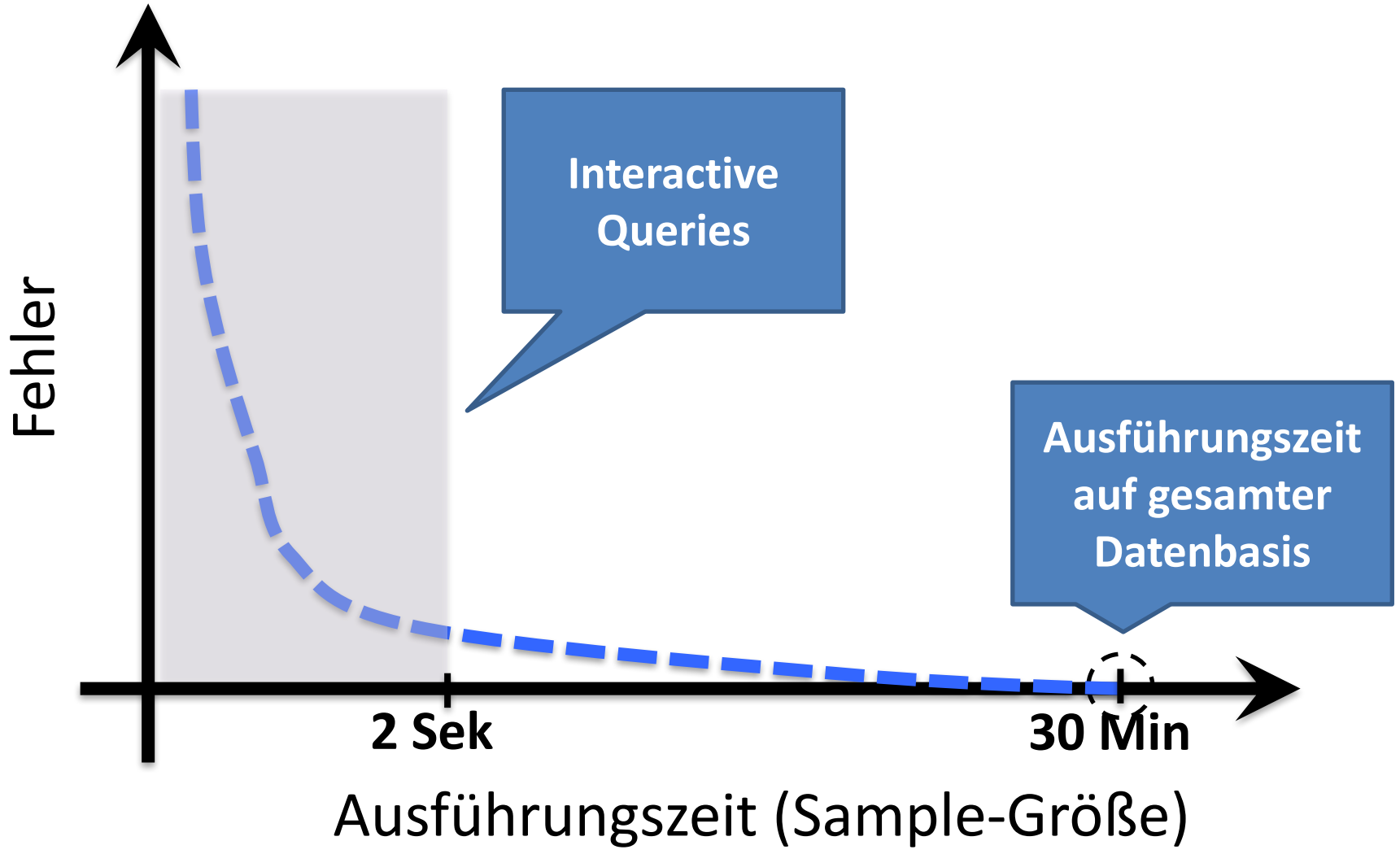
ID	Stadt	Gehalt	Rate
1	Hamburg	50.000 €	1/2
4	Hamburg	120.242 €	1/2
5	Berlin	98.341 €	1/2
7	Hamburg	60.000 €	1/2
10	Hamburg	92.242 €	1/2
11	Berlin	70.000 €	1/2

~~80.848,17 €~~

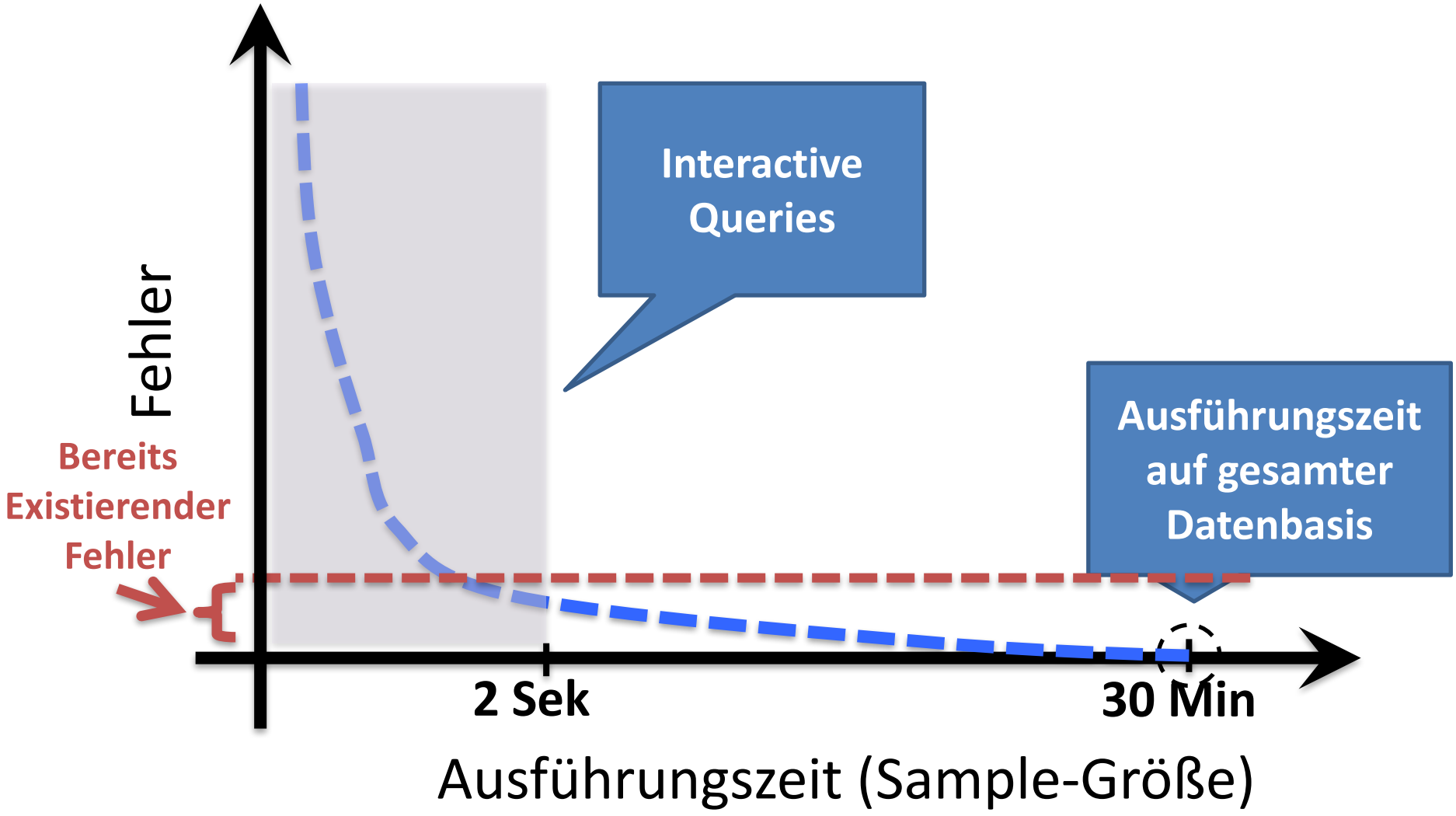
~~73.611,00 € +/- 7.237,17 €~~

81.804,17 € +/- 956,00 €

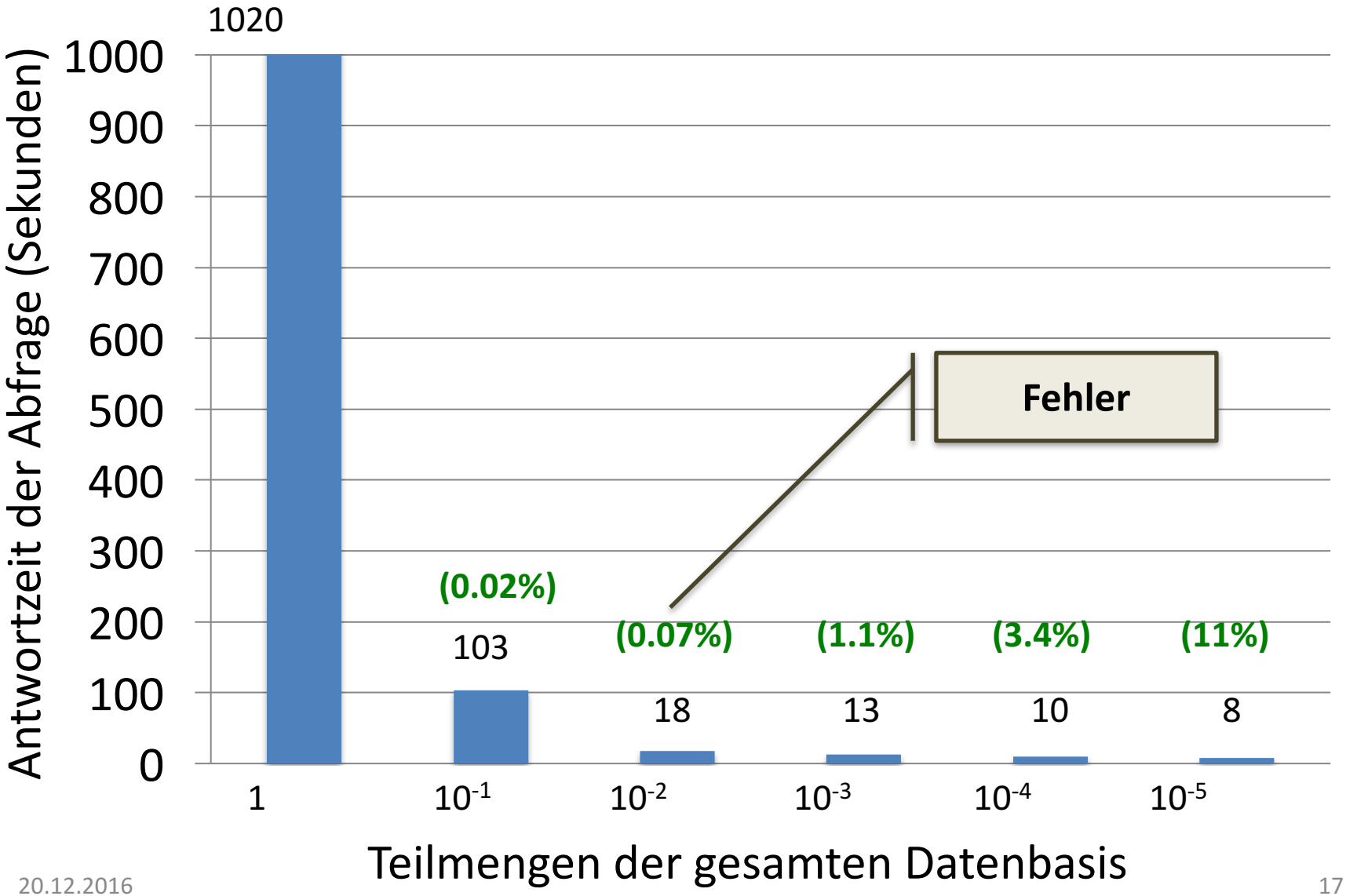
Geschwindigkeit / Genauigkeit [9]



Geschwindigkeit / Genauigkeit [9]



Sampling / kein Sampling [9]



Uniform / Stratified Samples [10, 11]

ID	Stadt	Gehalt
1	Hamburg	50.000 €
2	Berlin	62.492 €
3	Hamburg	78.212 €
4	Hamburg	120.242 €
5	Berlin	98.341 €
6	Hamburg	75.453 €
7	Hamburg	60.000 €
8	Berlin	72.492 €
9	Berlin	88.212 €
10	Hamburg	92.242 €
11	Berlin	70.000 €
12	Hamburg	102.492 €

Was ist das Durchschnittsgehalt aller Einträge in der Tabelle?

Uniform Sample

ID	Stadt	Gehalt	Gewicht
1	Hamburg	50.000 €	1/6
5	Berlin	98.341 €	1/6

Stratified Sample

ID	Stadt	Gehalt	Gewicht
1	Hamburg	50.000 €	1/7
5	Berlin	98.341 €	1/5

BlinkDB [12]

- Massiv paralleles Framework für approximative Anfrageverarbeitung auf großen Datenmengen
- Antworten in einer sehr kurzen Zeit
- Antworten mit einer garantierten Fehlerquote
- Skalierbares System, das für Petabyte von Daten ausgelegt ist

[B2]

BlinkDB: Anfragen [10]

```
SELECT COUNT(*)
FROM Sessions
WHERE Genre = ’western’
GROUP BY OS
ERROR 0.1 CONFIDENCE 95%
```

```
SELECT COUNT(*), ERROR AT 95% CONFIDENCE
FROM Sessions
WHERE Genre = ’western’
GROUP BY OS
WITHIN 5 SECONDS
```

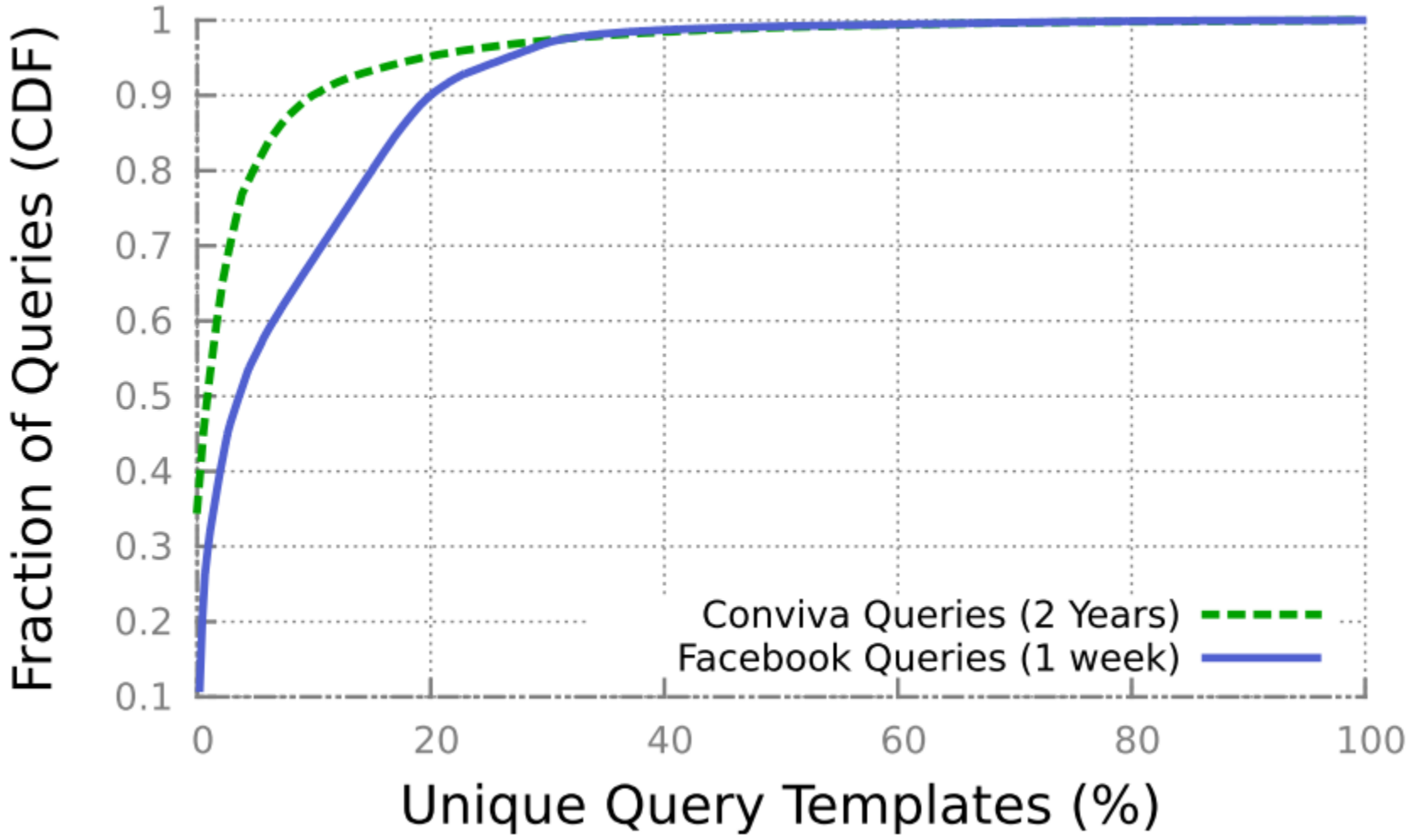
Sample Management [13]

- Zukünftige Anfragen sind „ähnlich“
- Aber was ist „ähnlich“?
- Verschiedene Modelle:
  - Vorhersagbare Anfragen
    - Werte in WHERE, GROUP BY, HAVING identisch
  - Vorhersagbare Anfrage-Prädikate
    - Häufigkeit von WHERE, GROUP BY, HAVING bleibt gleich
  - Mengen der Spalten ändern sich nicht
    - Keine Annahme über WHERE, GROUP BY, HAVING

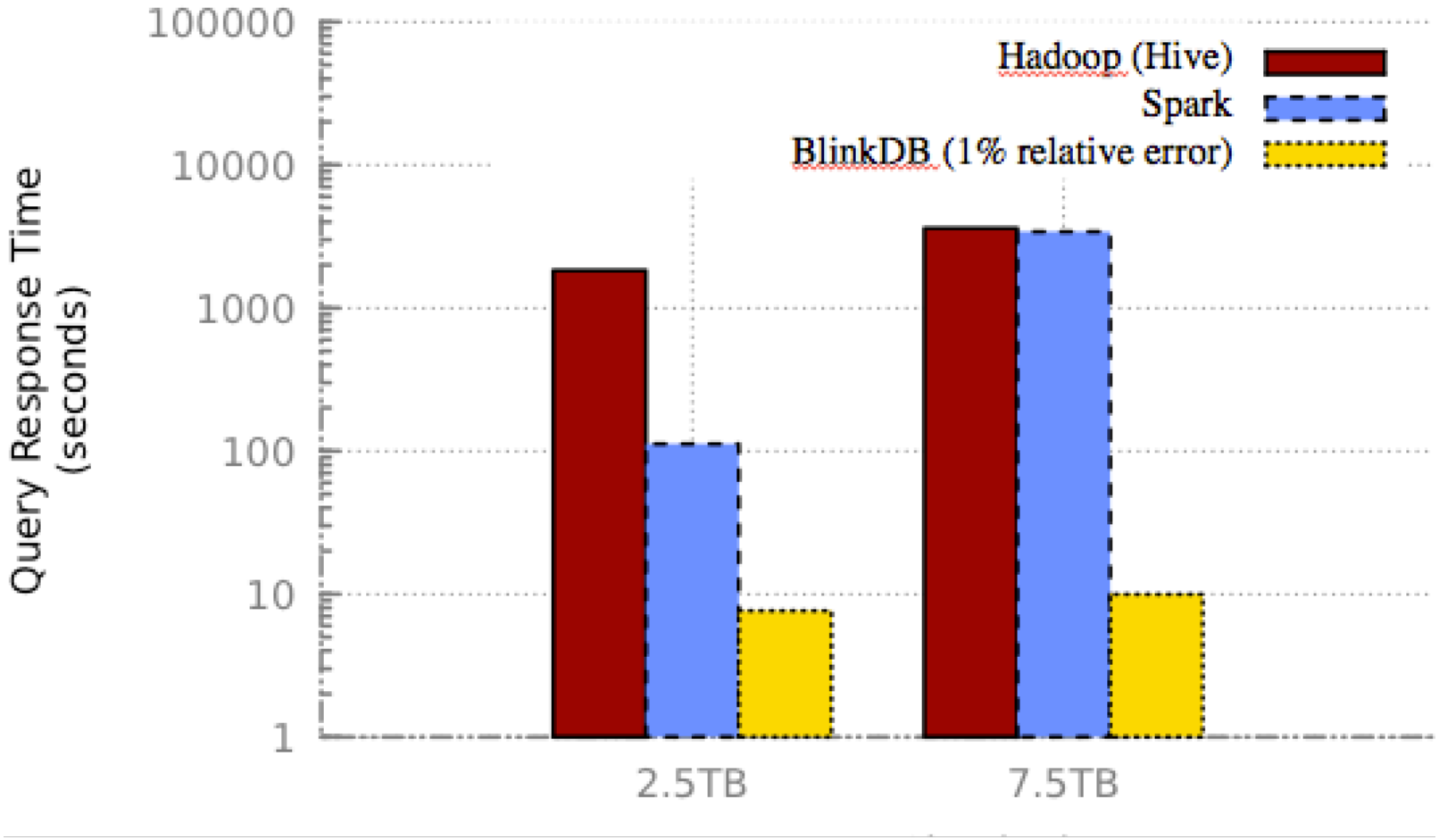
BlinkDB: Sample Management [13]

- Mengen der Spalten ändern sich nicht
  → „Query Column Set“ (QCS)
- In realen Umgebungen kommen dieselben QCSs sehr häufig vor
  - Beispieldaten: Facebook + Conviva

BlinkDB: Beispiel für QCSs[13]



BlinkDB: Ausführungsgeschwindigkeit [3]



BlinkDB: Aktueller Stand[19]

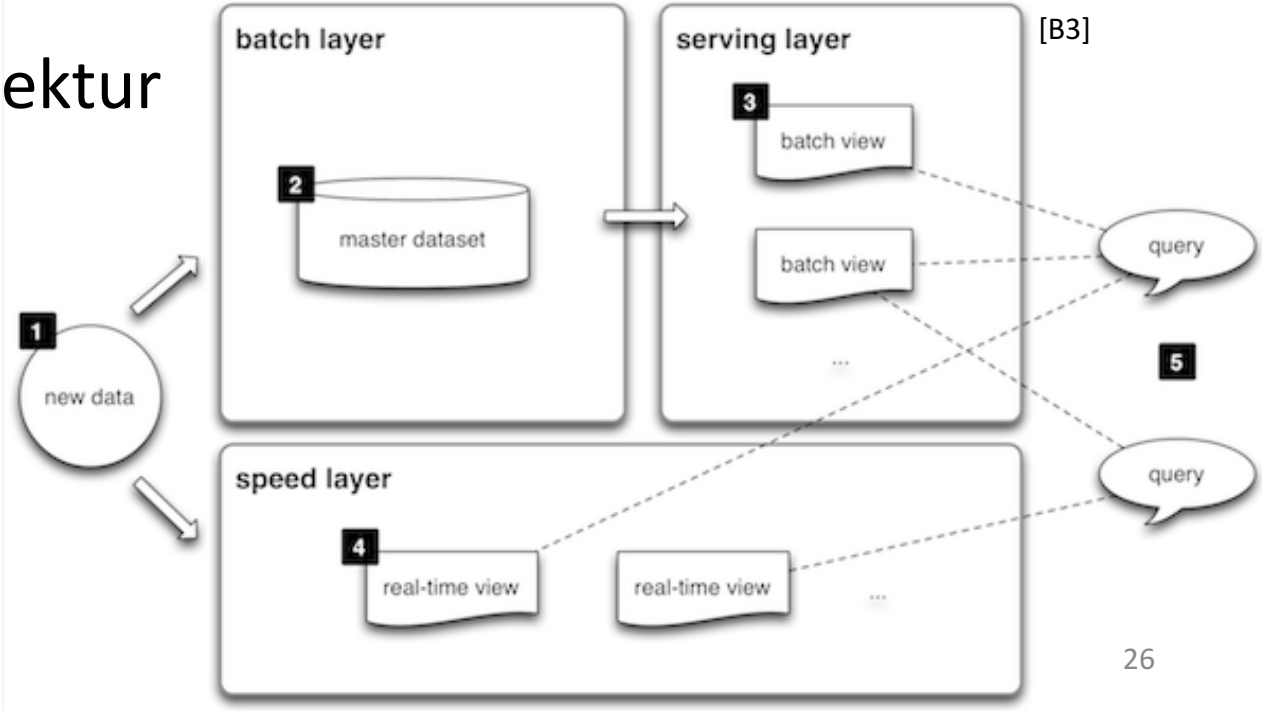
- Letzter Commit 2014
- Projekt steht still

→ Gibt es etwas Aktuelleres?

Anforderungen an Big-Data-Umgebungen [18]

- Kontinuierliches Stream-Processing
- Sehr schreibintensive Workloads (OLTP)
- Interaktive SQL-Analysen (OLAP)

→ Lambda-Architektur



Vorhandene Lösungen [18]

- Hohe Komplexität
- Niedrige Performance
- Verschwendete Ressourcen
- Interaktive Analysen ungenügend

→ Verbesserungen möglich?

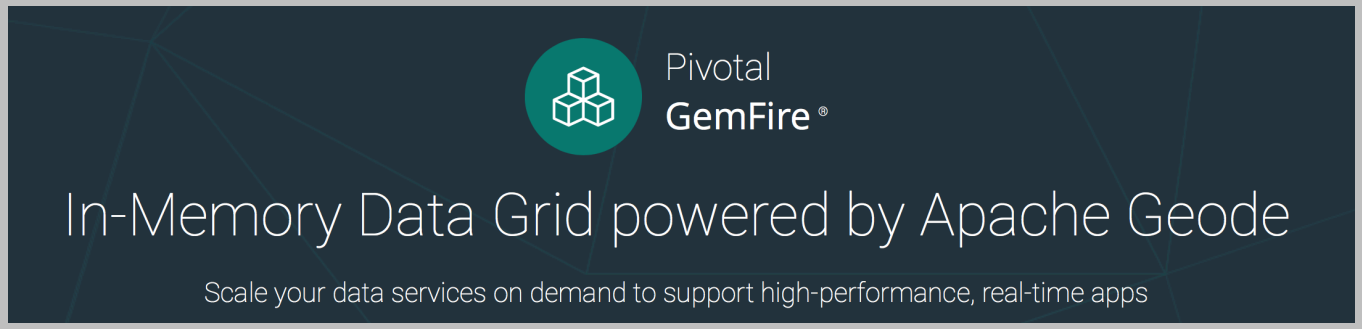
[B4]

SnappyData [14, 15, 16]

Batch Design
Hoher Durchsatz

Single Unified Cluster
OLTP + OLAP + Streaming

Approximate Query Processing



Real Time Design
Niedrige Latenz
Hochverfügbarkeit
Concurrency

SnappyData: Ziele [18]

- Schnelle interaktive Analysen
- Geringe Investitionen in Cluster-Infrastruktur
- Geringere Komplexität
  (im Vergleich zur Lambda-Architektur)
- Einschränkungen:
  - Terabytes, nicht Petabytes
  - Keine Use Cases, die eine sehr geringe Latenz benötigen (Bsp: Wertpapierhandel)

Meine Ziele (1)

- Experimentierplattform mit SnappyData auf HAW-Cluster realisieren (→ gerade dabei)
- Experimentierplattform mit SnappyData bei einem Cloud-Anbieter (AWS?) realisieren
- Performance-Analysen durchführen (TPC Benchmarks)

Meine Ziele (2)

- Gibt es eine Grenze, ab der sich die approximative Anfrageverarbeitung lohnt?
  - Ab welcher Datenmenge?
  - Bei welchen Anwendungsfällen?
  - Bei welchen Abfragearten?
- Lässt sich die Qualität der Genauigkeit verbessern?

Risiken

- Probleme bei der Installation (HAW-Cluster + Cloud-Anbieter)
- Kosten der Cloud-Anbieter
- Keien Grenze definierbar, ab der sich AQP lohnt
- Hürde: Statistische Modelle

Fragen?

Literatur

1. U.S. Government's Data Explosion (Infographic), https://whatsthebigdata.com/2014/09/15/u-s-governments-data-explosion-infographic/, letzter Zugriff: 12.12.2016
2. 2.800.000.000.000.000.000.000 Byte: Das digitale Universum schwillt an. http://www.spiegel.de/netzwelt/web/das-internet-der-dinge-erzeugt-2-8-zettabyte-daten-a-872280.html, letzter Zugriff: 18.12.2016
3. Mozafari, B. BlinkDB: A Massively Parallel Query Engine for Big Data, 2013. http://istc-bigdata.org/index.php/blinkdb-a-massively-parallel-query-engine-for-big-data/, letzter Zugriff: 18.12.2016
4. Dominik Klein, Phuoc Tran-Gia, M. H. Big data, 2013. http://www.gi.de/service/informatiklexikon/detailansicht/article/big-data.html, letzter Zugriff: 18.12.2016.
5. The Data Explosion in 2014 Minute by Minute – Infographic, 2014. http://aci.info/2014/07/12/the-data-explosion-in-2014-minute-by-minute-infographic/, letzter Zugriff: 18.12.2016
6. Hecht, R., and Jablonski, S. NoSQL evaluation: A use case oriented survey. Proceedings - 2011 International Conference on Cloud and Service Computing, CSC 2011 (2011), 336–341.
7. Pritchett, D. Base: an Acid Alternative. Queue 6, 3 (2008), 48–55.
8. Dean, J., and Ghemawat, S. Mapreduce: simplified data processing on large clusters. Commun. ACM 51, 1 (2008), 107–113.
9. Agarwal, S. Blinkdb: Approximate Queries on Very Large Data, 2013. https://spark-summit.org/wp-content/uploads/2013/10/BlinkDB-SparkSummit-v3.pptx, letzter Zugriff: 18.12.2016.
10. Agarwal, S., Mozafari, B., Panda, A., Milner, H., Madden, S., and Stoica, I. BlinkDB: queries with bounded errors and bounded response times on very large data. Proceedings of the 8th ACM European Conference on Computer Systems - EuroSys '13 (2013), 29.
11. Agarwal, S. Blinkdb: Qureying petabytes of data in seconds using sampling, 2014. http://de.slideshare.net/Hadoop_Summit/t-1205p212agarwalv2, letzter Zugriff: 18.12.2016.
12. Agarwal, S., Iyer, A. P., Panda, A., Madden, S., Mozafari, B., and Stoica, I. Blink and it's done: interactive queries on very large data. Proceedings of the VLDB Endowment 5, 12 (2012), 1902–1905.
13. Upreti, N. Introduction to BlinkDB : Queries with Bounded Errors and Bounded Response Times on Very Large Data, 2014. http://de.slideshare.net/nitishupreti/blinkdb, letzter Zugriff: 18.12.2016
14. http://www.snappydata.io/product, letzter Zugriff: 12.12.2016
15. https://pivotal.io/big-data/pivotal-gemfire, letzter Zugriff: 12.12.2016
16. https://spark.apache.org, letzter Zugriff: 12.12.2016
17. http://www.slideshare.net/sawjd/explore-big-data-at-speed-of-thought-with-spark-20-and-snappydata, letzter Zugriff: 12.12.2016
18. Mozafari, Jags Ramnarayan1 Barzan and Menon, Sumedh Wale1 Sudhir and Chakraborty, Neeraj Kumar1 Hemant Bhanawat1 Soubhik and Bachhav, Yogesh Mahajan1 Rishitesh Mishra1 Kishor, SnappyData: Streaming, Transactions, and Interactive Analytics in a Unified Engine, http://www.snappydata.io/snappy-industrial, letzter Zugriff: 18.12.2016
19. https://github.com/sameeragarwal/blinkdb, letzter Zugriff: 17.12.2016

Bilder

B1: http://www.amusingtime.com/images/017/funny-running-dog-picture.jpg, letzter Zugriff: 12.12.2016

B2: http://blinkdb.org/figures/blinkdb-logo-withaffiliations.png, letzter Zugriff: 12.12.2016

B3: http://lambda-architecture.net/img/la-overview_small.png, letzter Zugriff: 19.12.2016

B4: http://nonbinaryreview.com/wp-content/uploads/doc_0.jpg, letzter Zugriff: 19.12.2016