



Web Science – Investigating the Future of Information and Communication

Big Data

Prof. Robert Jäschke

Forschungszentrum L3S

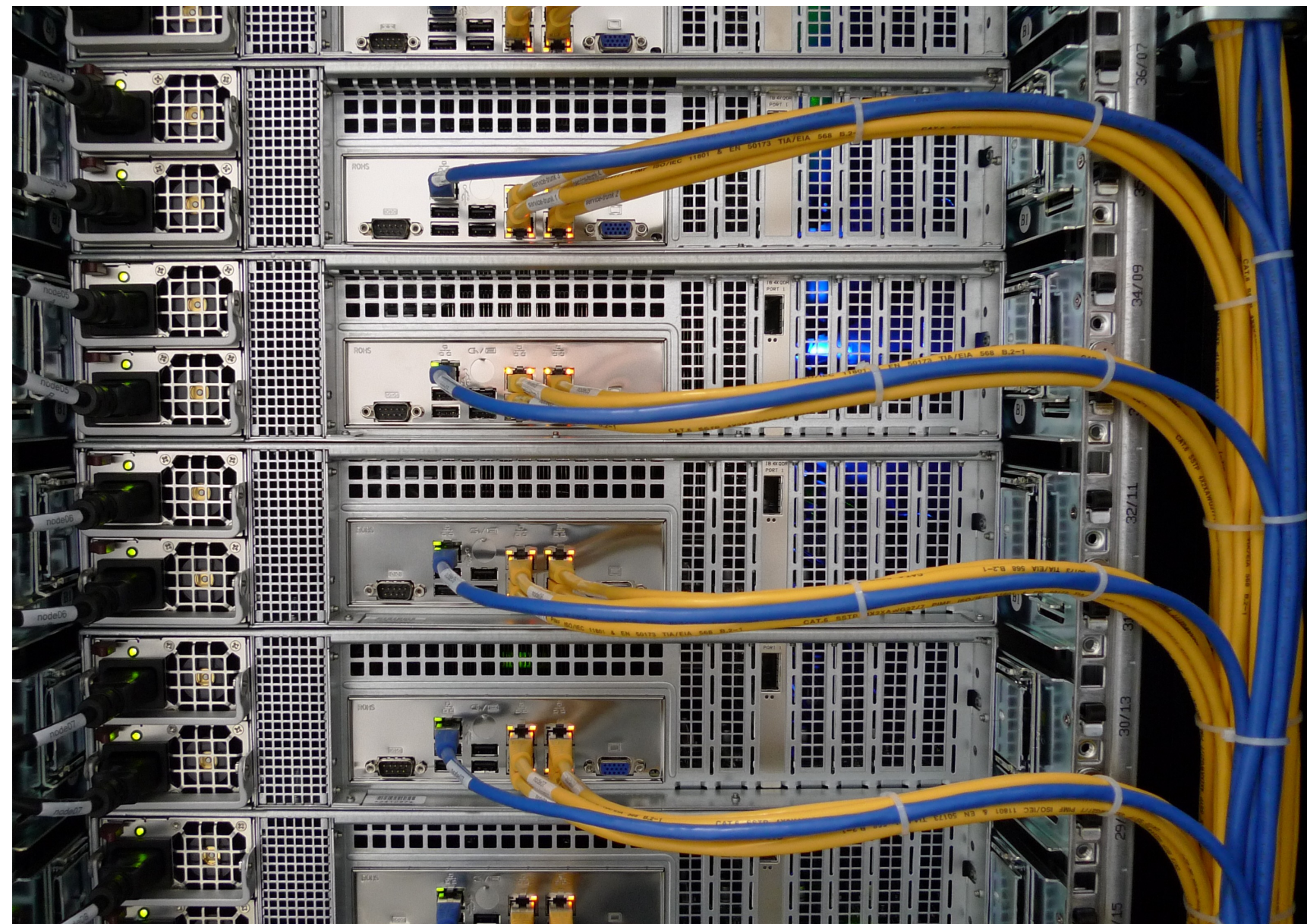
Leibniz Universität Hannover



Leibniz
Universität
Hannover

Agenda

- Was ist Big Data?
- Parallele Programmierung
- Map/Reduce
- Der “Big Data Zoo”



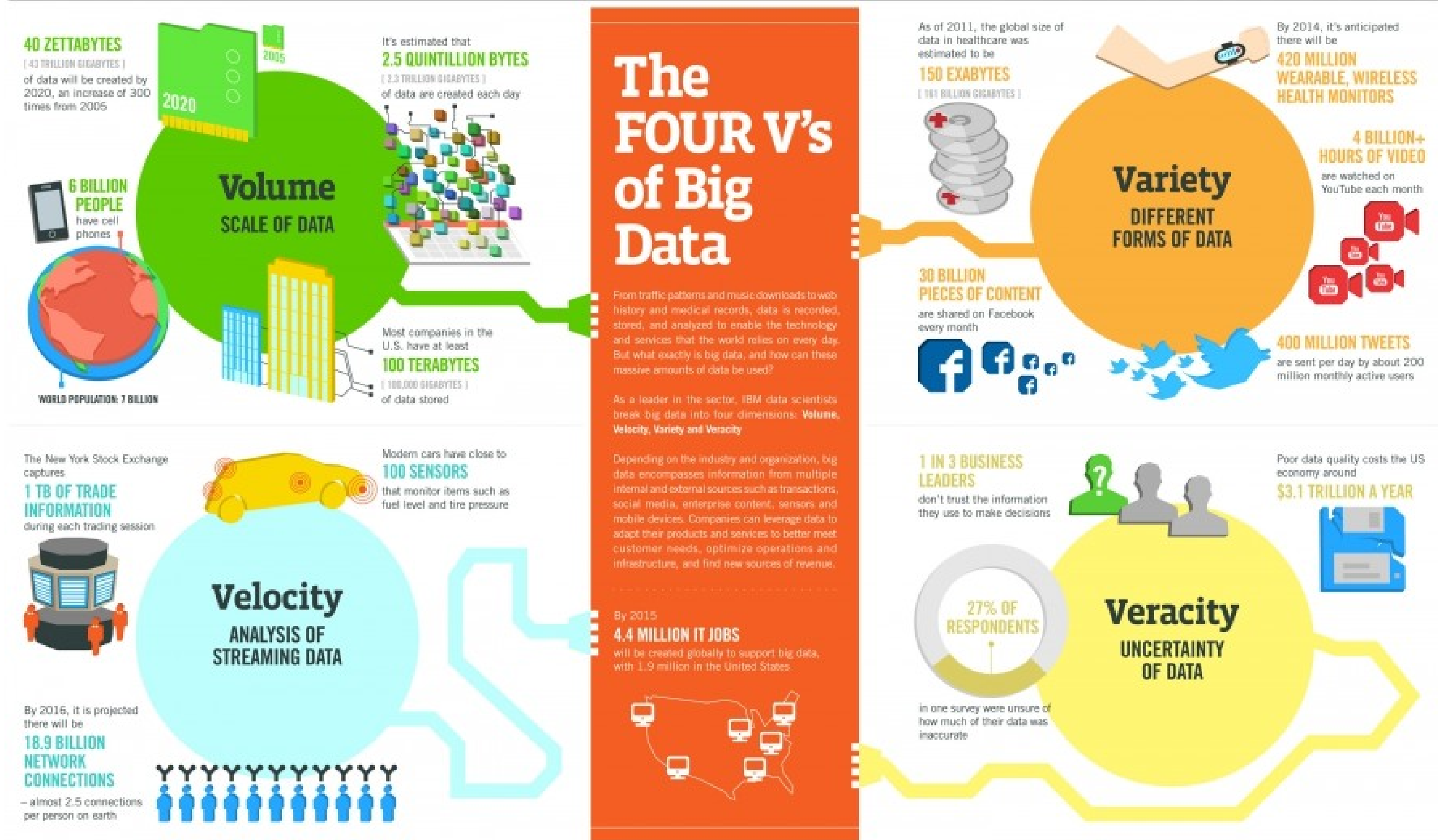
“3Vs” oder: Was ist Big Data?

- Deutsche Telekom: <http://youtu.be/Z0PWQd3CvGE>
- Landesanstalt für Medien NRW: <http://youtu.be/otWN5o1C2Bc>

“3Vs” oder: Was ist Big Data?

- Daten zu groß, für Verarbeitung mit traditionellen Anwendungen
- **Aufgaben:** Sammeln, Speichern, Teilen, Transferieren, Visualisieren, Durchsuchen, Indizieren, Analysieren, etc.
- **Beispiele:** Meteorologie, Physikalische Simulationen (LHC), Finanzwirtschaft, Internetsuche, Wirtschaftsinformatik, ...
- **Ursachen:** viele neue Sensoren, Logdateien, Kameras, etc.
- **Problem:** schwierig mit traditionellen RDBMs/Software zu verarbeiten
- **Stattdessen:** massiv parallele Software, hunderte/tausende Server
- **“groß”:** hängt von Anwendung/Organisation ab, ändert sich

“3Vs” oder: Was ist Big Data?



Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, OAS

How BIG is Big Data?



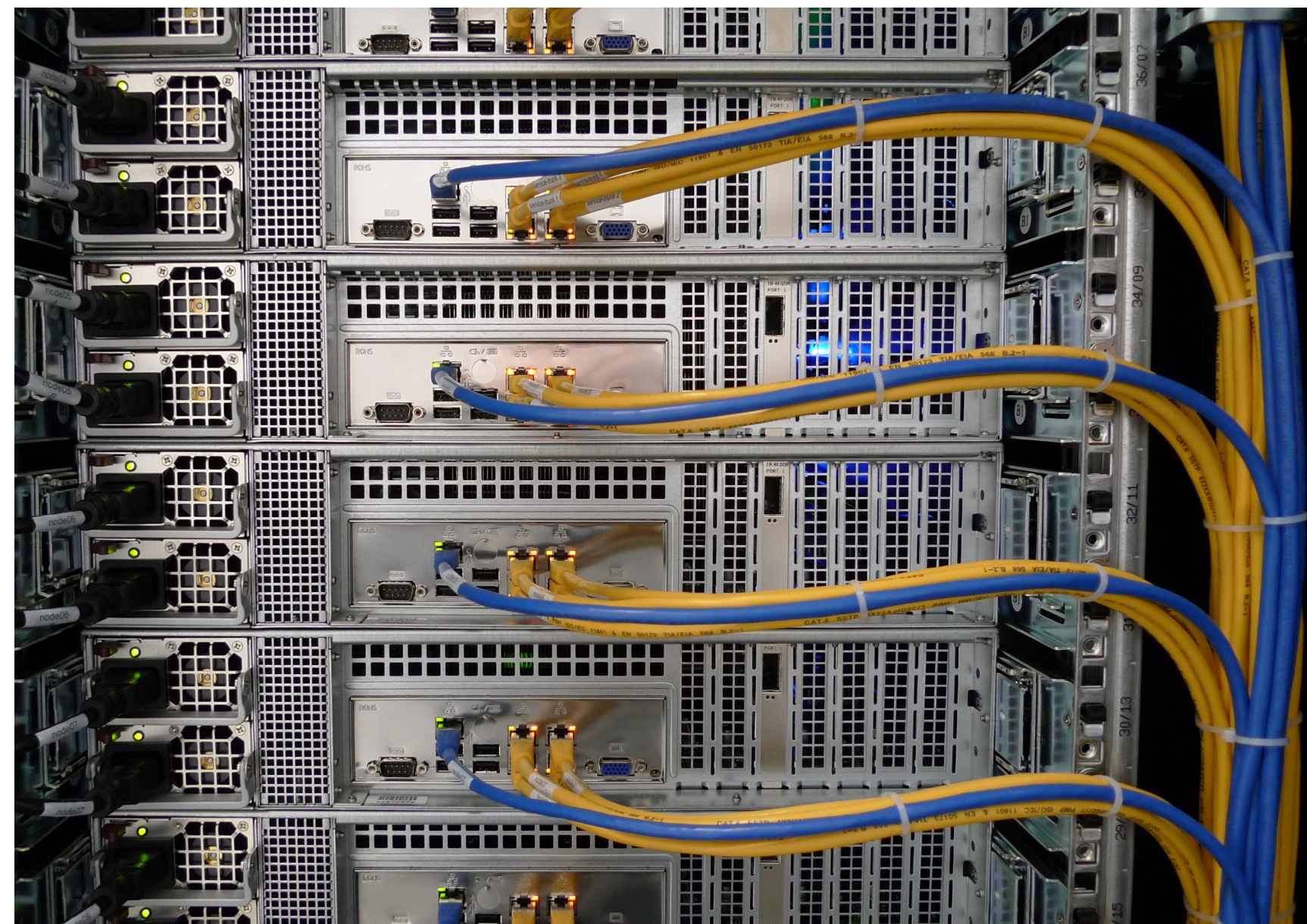
Sources:
 Roe, Charles. "The Growth of Unstructured Data: What To Do with All Those Zettabytes?" Dataversity. 15 March 2012.
 Tam, Donna. "Facebook processes more than 500 TB of data daily." CNet. 22 August 2012.
 Huggins, James. "How Much Data is That?" James S. Huggins' Refrigerator Door.
 Abagond, Julian. "Library of Alexandria." Abagond Blog. 15 September 2006.

Fallstudie

- Erarbeiten Sie ein Szenario aus Ihrem Fach-/Interessensgebiet, in dem Big Data eine wesentliche Rolle spielt (z.B. Energienetze, ...)
 - Welche Daten fallen an?
 - Welche Möglichkeiten, Herausforderungen und Probleme ergeben sich durch die Nutzung dieser Daten?
 - Suchen Sie nach Lösungsansätzen für Ihr Szenario.
- Arbeit in Kleingruppen (3-4 Personen), 10 Minuten Zeit
- Ergebnissicherung: Präsentation/Ausfüllen der Tabelle an der Tafel
 - (Spalten: Szenario, Daten, Möglichkeiten, Herausforderungen, Probleme, Lösungsansätze)

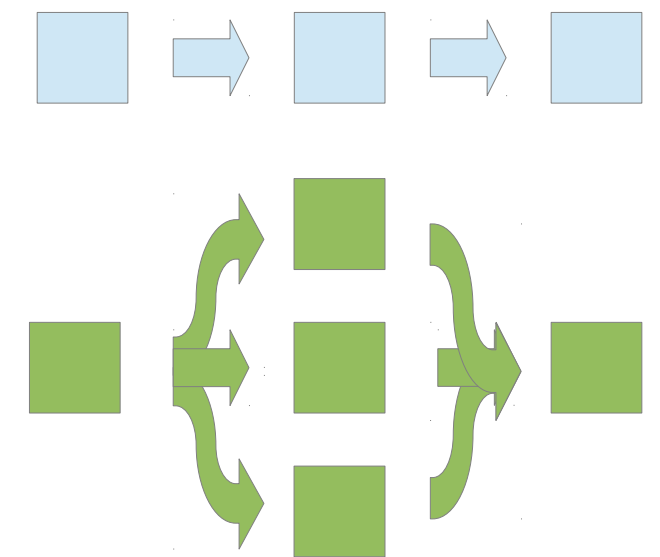
Agenda

- Was ist Big Data?
- **Parallele Programmierung**
- Map/Reduce
- Der “Big Data Zoo”



Hintergrund

- traditionelle Programme laufen seriell
- paralleles Programmieren:
 - Aufgaben in Teile zerlegen, die parallel auf mehreren Rechnern/Prozessoren laufen können
- Herausforderung:
 - Aufgaben identifizieren, die parallel laufen können
 - ... oder Datensätze, die parallel verarbeitet werden können
 - nicht alle Probleme können parallelisiert werden

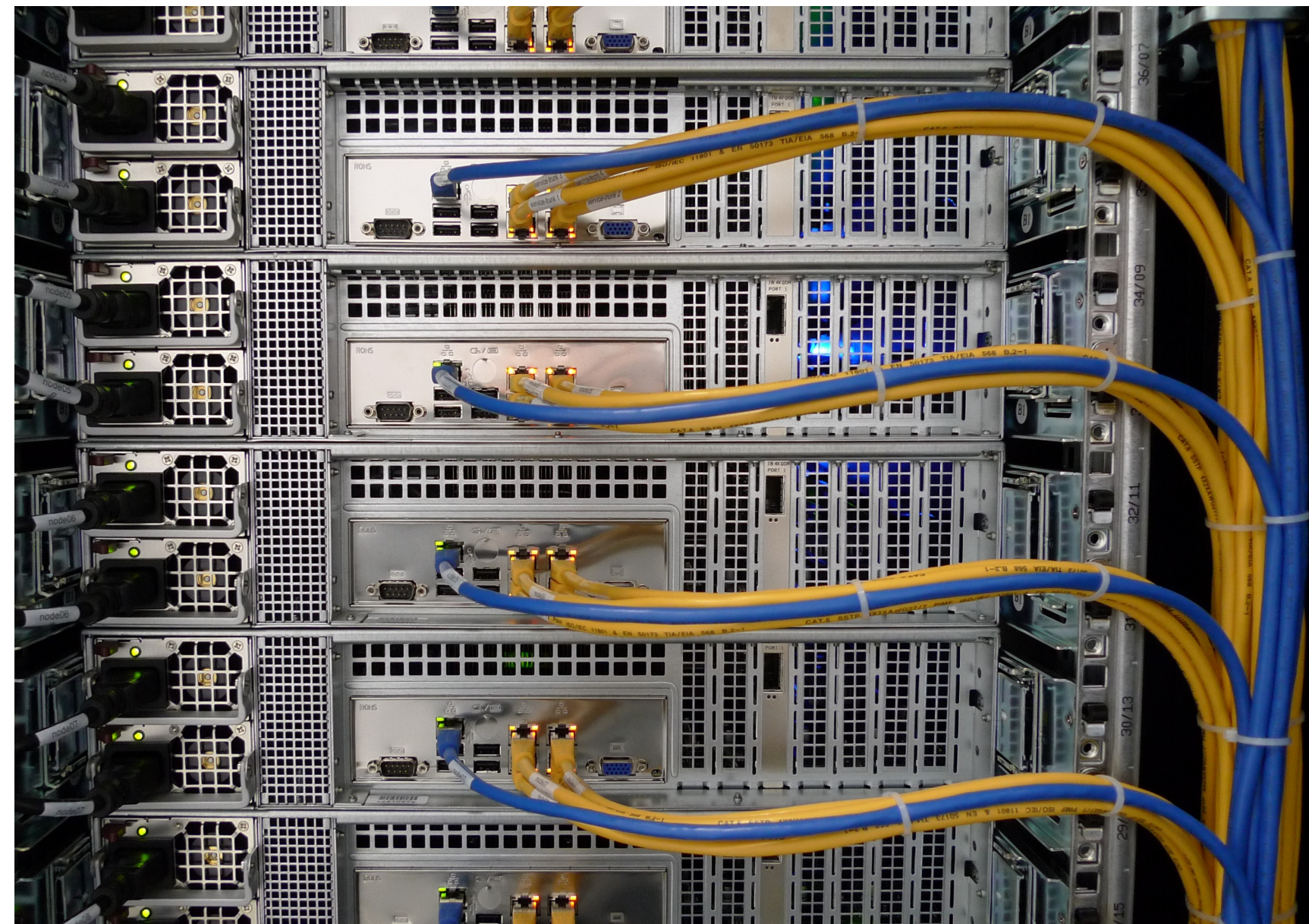


Parallele Programmierparadigmen

- Shared Memory Systems
 - Multi-Threading
- GPU / CUDA
- Distributed Memory Systems / Message Passing
 - MPI
 - **Map/Reduce**
 - Dremel (Apache Drill)

Agenda

- Was ist Big Data?
- Parallele Programmierung
- **Map/Reduce**
- Der “Big Data Zoo”



Überblick

- Map/Reduce (Jeffrey Dean und Sanjay Ghemawat, Google, 2004)
 - Programmiermodell + Ausführungsumgebung für Datenparallelismus
 - Entwickelt für Daten-intensive Anwendungen
 - einfache Schnittstelle, separiert *Was* vom *Wie*
 - Verarbeitung sehr großer Datenmengen (TB-PB) auf tausenden von Prozessoren
- Google File System
 - Verteiltes Dateisystem
 - Optimiert für günstige Hardware
- Open Source Implementierungen
 - Hadoop: Map/Reduce
 - HDFS: Google File System



Ideen hinter Map/Reduce

- viele günstige Standard-Server statt weniger High-End-Geräte
- Fehler als Normalfall
- Programm zu den Daten verschieben statt umgekehrt
- Daten sequentiell lesen
- Details vom Programmierer fernhalten
- Skalierbarkeit

Ideen hinter Map/Reduce

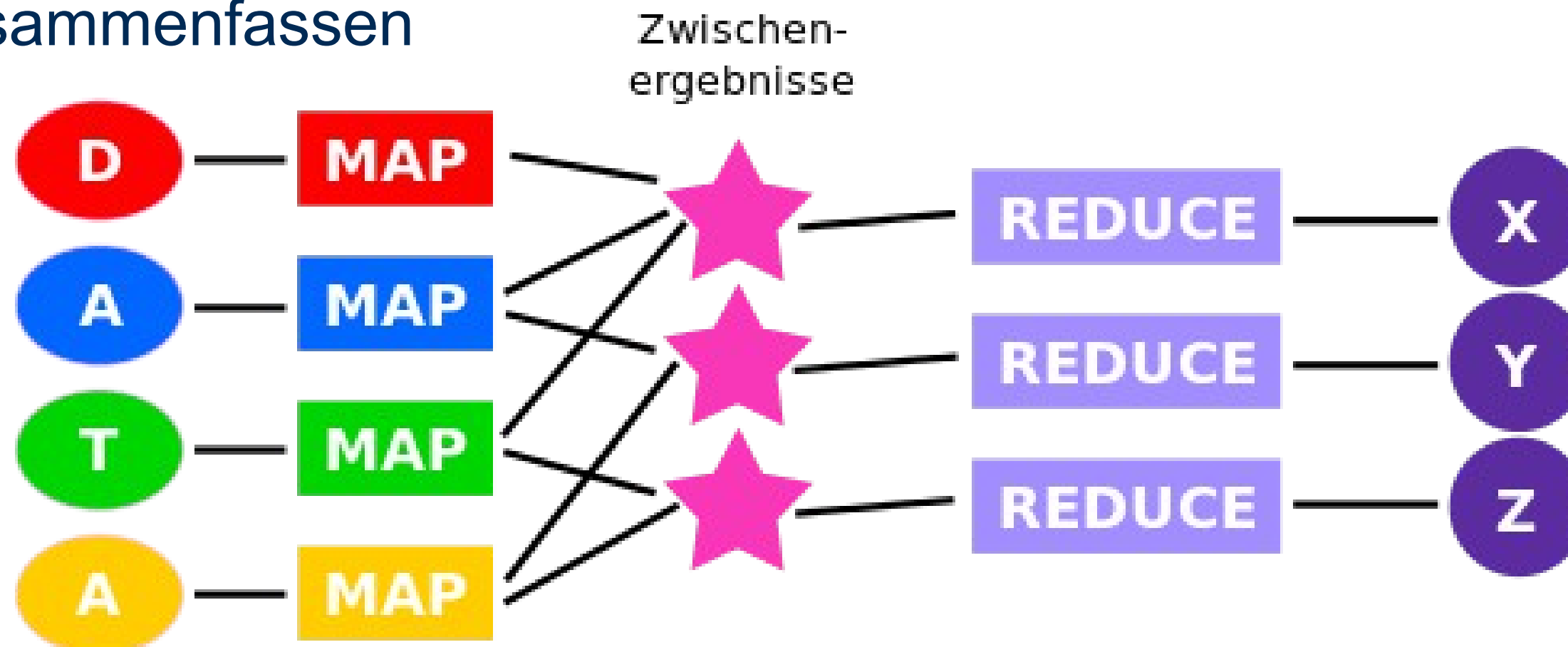
- keine Abhängigkeiten zwischen Daten
- Daten können in gleichgroße Datensätze geteilt werden
- Jeder Prozess verarbeitet einen Datensatz
- Master/Worker-Ansatz:
 - Master
 - teilt Daten in Datensätze
 - verteilt diese an Worker
 - erhält Ergebnisse von Workern
 - Worker
 - erhalten Datensätze von Master
 - verarbeiten diese
 - senden Ergebnisse an Master

Ideen hinter Map/Reduce

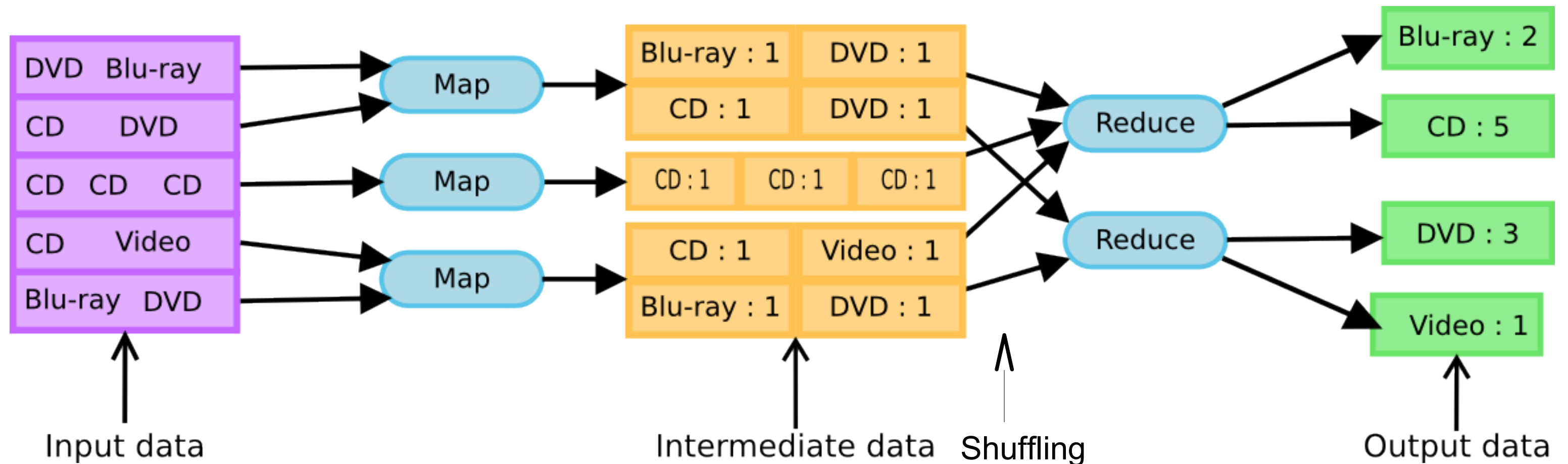
- inspiriert durch LISP
 - **Map (function, set of values)**
 - wendet Funktion auf jeden Wert in der Menge an
 - `(map 'length ' (()) (a) (a b) (a b c))) → (0 1 2 3)`
 - **Reduce (function, set of values)**
 - kombiniert Werte mit Hilfe einer binären Funktion (z.B. +)
 - `(reduce #' + ' (0 1 2 3)) → 6`

Das Map/Reduce Modell

- Map-Schritt
 - Schlüssel/Wert-Paare als Eingabe mit geg. Funktion verarbeiten ...
 - Schlüssel/Wert-Paare als Zwischenergebnis speichern
- Reduce-Schritt
 - Zwischenergebnisse mit dem gleichen Schlüssel mit geg. Funktion zusammenfassen



Beispiel: Wörter zählen



```
map(key, value) :
```

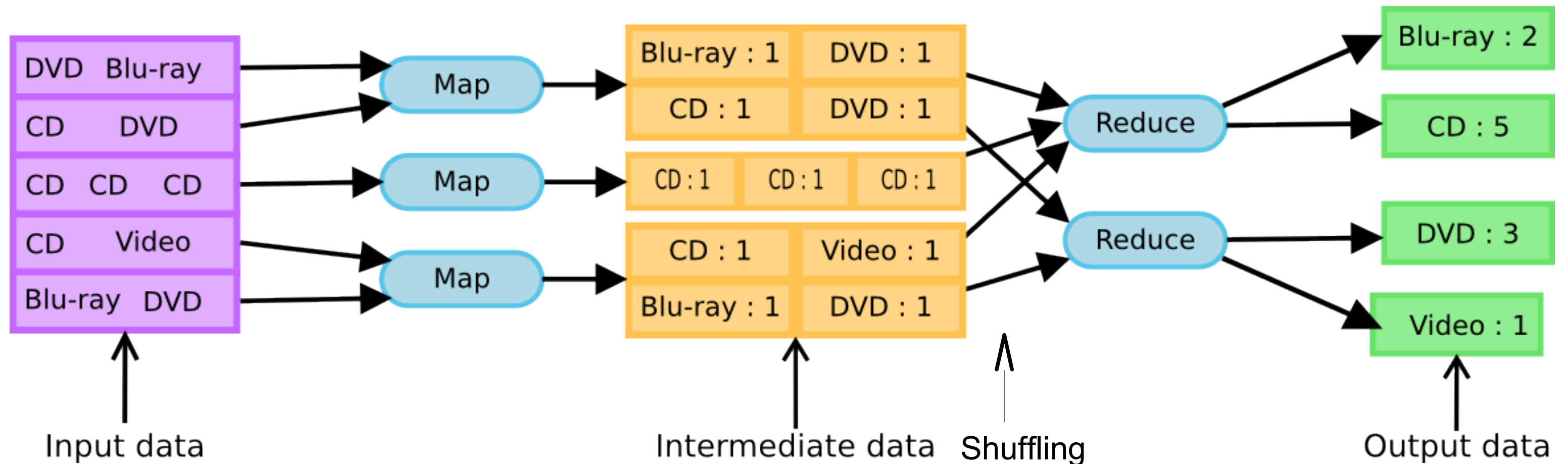
```
// key: Dokumentname
```

```
// value: Dokumentinhalt
```

```
for word in value:
```

```
    EmitIntermediate(word, 1) ;
```

Beispiel: Wörter zählen



reduce(key, values) :

// key: ein Wort

// values: eine Liste von Werten

result = 0;

for value in values:

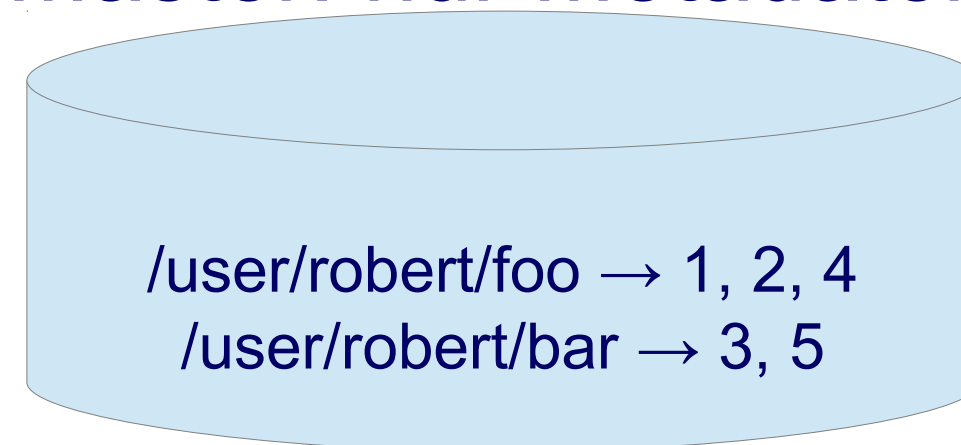
 result += v;

Emit(result);

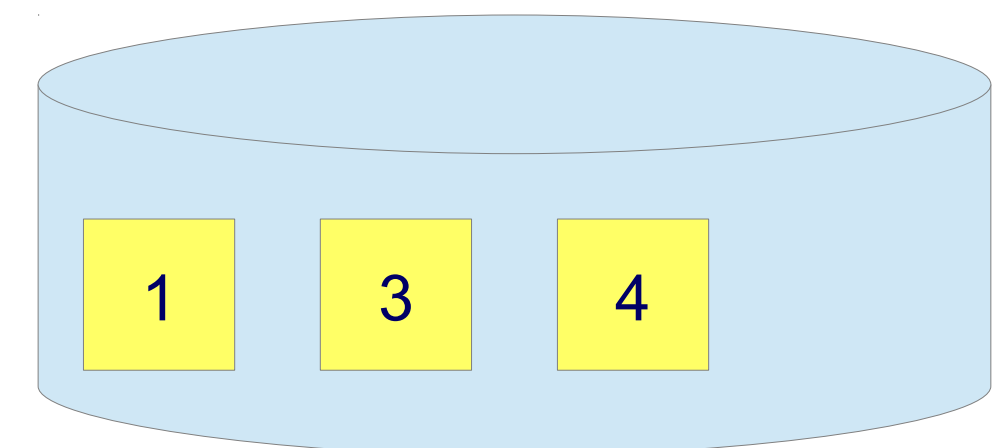
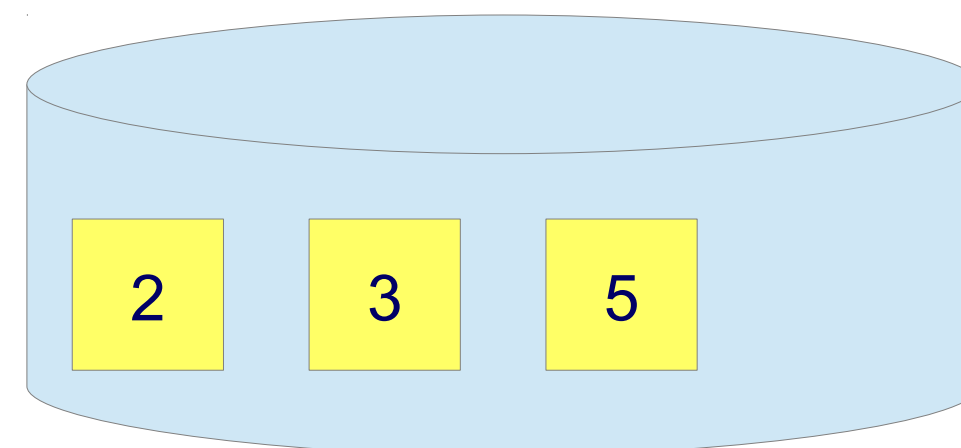
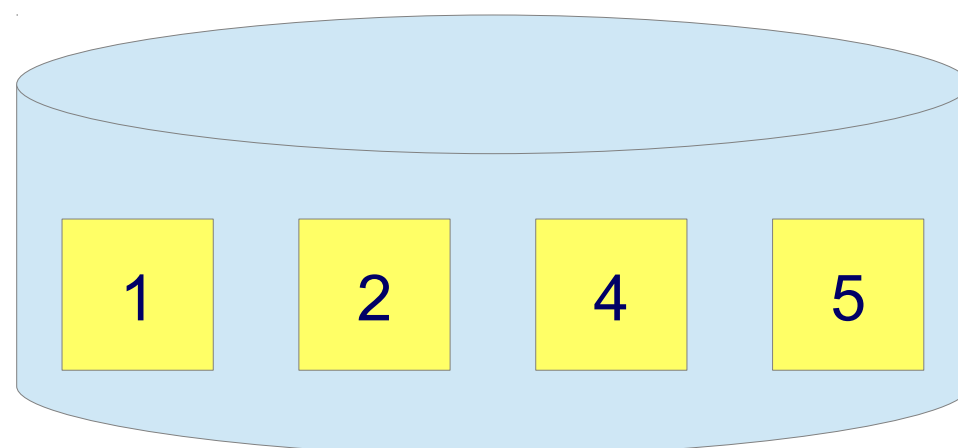
Wie kommen die Daten zu den Workern?

- mittels HDFS!
- Daten verteilt auf den lokalen Festplatten der Rechner gespeichert
- Worker führen Map-Schritt auf ihren lokalen Daten durch

Master: nur Metadaten



Worker: Dateiblöcke



Map/Reduce anwenden

- Geben Sie für folgende Beispiele die Map/Reduce-Schritte an:
 - eine Zeichenkette in Dokumenten suchen
 - Zugriffe in Webserver-Logdateien zählen
 - eingehende Links zu einer Webseite im Linkgraphen finden
 - gemeinsames Auftreten von Wörtern in Dokumenten zählen
 - einen invertierten Index bauen (welches Dokument enthält welches Wort)
 - Datensätze sortieren
 - Kacheln von (geographischen) Karten rendern
- Ein Thema pro Gruppe, 5 Minuten Zeit, danach Präsentation

Map/Reduce anwenden

- Betrachten Sie Ihre Fallstudie vom Anfang – ist Map/Reduce ein geeignetes Modell zur Lösung Ihres Problems bzw. von Teilproblemen?
- Welche Schwierigkeiten können Sie aufzeigen?
- Arbeit in Kleingruppen (3-4 Personen), 5 Minuten Zeit
- Ergebnissicherung: kurzes Statement pro Gruppe

Beispiele

- Yahoo: Hadoop **sortiert ein Petabyte in 16,25 Stunden** und ein Terabyte in 62 Sekunden (3800 Knoten, Mai 2009)
- Jimmy Lin: Hadoop zählt das gemeinsame Auftreten von Wörtern in einem Korpus aus 2,27 Millionen Dokumenten (1,8GB komprimierte Daten) in 700 Sekunden (38 Knoten, 2010)
- Eric Baldeschwieler (Hortonworks): **Yahoo hat 42000 Hadoop-Rechner** mit 180-200 Petabyte an Daten (Juli 2011)
- Jack Norris (MapR): „Hadoop costs on the order of **a few hundred dollars per terabyte** to store raw data compared to something like **\$16,000 per terabyte** for a data warehouse based on relational databases.“ (2013)
- L3S: Cluster mit 19 Knoten (196 CPU Cores, 2 TB RAM, 720 TB HDD) zur Analyse des deutschen Web-Archivs (>80TB)

Nicht alles ist perfekt ...

- Modell mit begrenzten Möglichkeiten
- Aber: viele Zeichenketten-Verarbeitungsprobleme lassen sich ganz natürlich darauf abbilden
- Kann iterativ genutzt werden (ggf. auch mehrere Map-Schritte hintereinander)
- **Map/Reduce wurde bei Google genutzt um die Daten von Webseiten während des Crawlens zu verarbeiten**
 - Extraktion von Links und Metadaten für die Suche
 - Berechnung des PageRanks
 - das ganze dauerte acht Stunden
 - → zu lang für die derzeitige Dynamik des Webs
 - Ziel: Aktualisierung des Suchindex im Sekundenbereich

Nicht alles ist perfekt ...

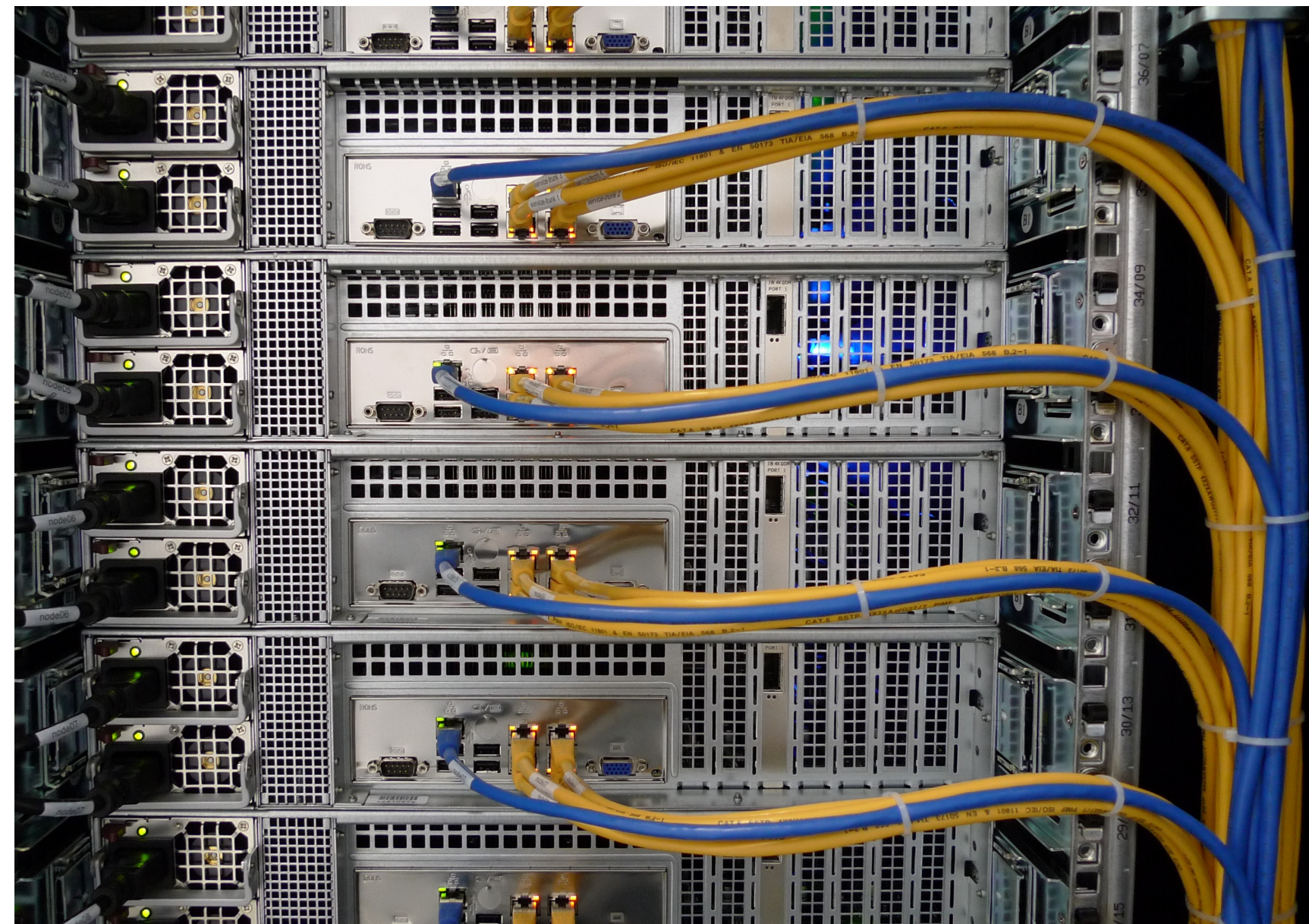
- Daher (25. Juni 2014):

*„The Elephant was a Trojan Horse:
On the Death of Map-Reduce at
Google“*

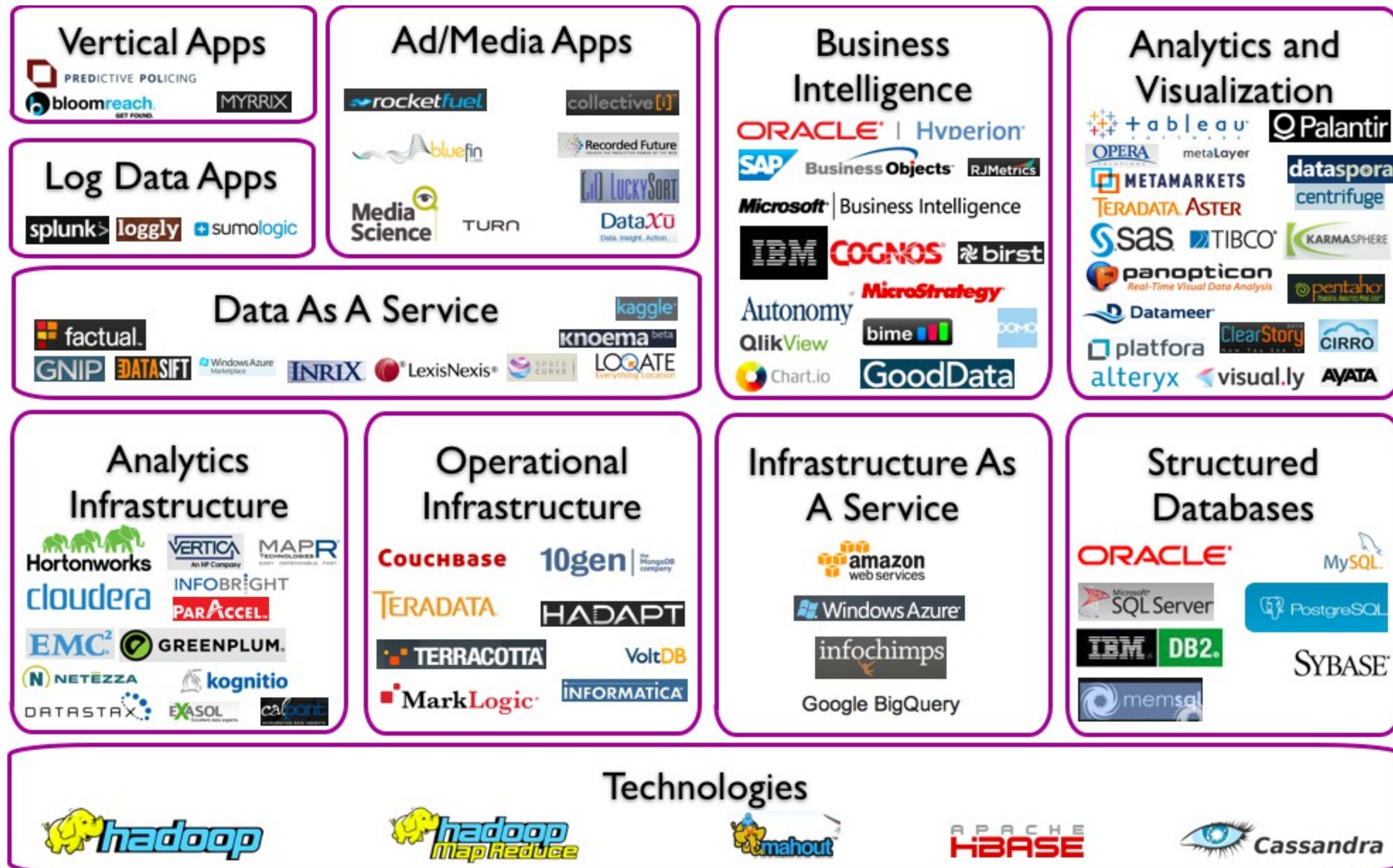
<http://the-paper-trail.org/blog/the-elephant-was-a-trojan-horse-on-the-death-of-map-reduce-at-google/>

Agenda

- Was ist Big Data?
- Parallele Programmierung
- Map/Reduce
- Der “Big Data Zoo”



Der “Big Data Zoo”



Der “Big Data Zoo”

- Suchen Sie sich aus der folgenden Liste an Tools eines aus:
 - Hadoop, Cloudera, HDFS, MapReduce, Zookeeper, HBase, Hive, Spark, Storm, Pig, Drill, Mahout, Kafka, YARN, Oozie, Solr, Flume, Impala, Parquet, Cassandra, MongoDB, ElasticSearch, Kibana, Logstash, ...
- Finden Sie heraus, welche Aufgabe es hat, und welche Beziehungen es zu anderen Tools gibt (5 Minuten)
- Ergebnissicherung:
 - Zettel mit Tool an Tafel heften
 - Beziehungen zu anderen Tools einzeichnen

Zusammenfassung

Was sollten Sie nach von dieser Vorlesung mitnehmen?

- Was ist Big Data? Volume, Velocity, Variety
- Ideen hinter Map/Reduce
- Map/Reduce Programmiermodell
- Grenzen des Modells
- existierende Anwendungen

Quellen und Literatur

- Folien basieren teilweise auf Material folgender Vorlesungen:
 - EN 600.320/420 (Randal Burns, John Hopkins University)
 - Distributed Systems (Paul Krzyzanowski, Rutgers University)
 - Cloud Course (Jimmy Lin, University of Maryland)
- J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters, OSDI, 2004.
- Tutorials:
 - Apache Hadoop 2.5.1: <http://bit.ly/1rG2nXB>
 - +Beispiele: <http://research.google.com/pubs/pub36249.html>
 - <http://developer.yahoo.com/hadoop/tutorial/module3.html>
- Weitergehendes Material:
 - <http://www.ft.com/cms/s/2/21a6e7d8-b479-11e3-a09a-00144feabdc0.html>
 - http://archive.wired.com/science/discoveries/magazine/16-07/pb_theory
 - <http://the-paper-trail.org/blog/the-elephant-was-a-trojan-horse-on-the-death-of-map-reduce-at-google/>