



Big Data Lösungen mit Apache Hadoop

Gunnar Schröder, T-Systems Multimedia Solutions GmbH

•• **T** •• Systems ••

Was ist Big Data?



Charakteristiken von Big Data – Three Vs of Big Data



VOLUME

Terabytes
Petabytes
Exabytes
Zettabytes

VELOCITY

Batch
Near Time
Real Time
Streams

VARIETY

Strukturiert
Semistrukturiert
Unstrukturiert

ANALYTICS



Facebook in Zahlen (Stand August 2012)

- Insgesamt
 - 950 Millionen User
 - Im Durchschnitt 6,5 h pro Monat online
 - Cirka 180.000 Server
- Pro Tag:
 - 2,5 Milliarden Content Items (status updates, wall posts, photos, videos, comments)
 - 2,7 Milliarden Likes
 - 300 Millionen Photos uploaded
 - 500+ Terabytes neue Daten werden in Datenbanken geladen
 - März 2008: 1,2TB
 - März 2009: 12 - 18TB
 - September 2009: 30 - 36TB
 - Juli 2010: 80 - 90 TB



Big Data z.B. in der Logistik



- Datenerfassung in Verteilzentren
- Mobile Endgeräte für Angestellte
 - Bar Code Readers
 - Honeywell Dolphin 99EX
- GPS Tracking
- Sensordata
 - Vehicle Monitoring (Fuel)
 - DHL Smart Sensor
 - RFID Chips
- B2B
 - Shipment Tracking
 - Supply Chain Management
- Mobile Dienste für Kunden
 - Smartphones
 - Tablets
- Electronic Mail
- Web Usage Data

.. T .. Systems ..

Herausforderungen für Unternehmen

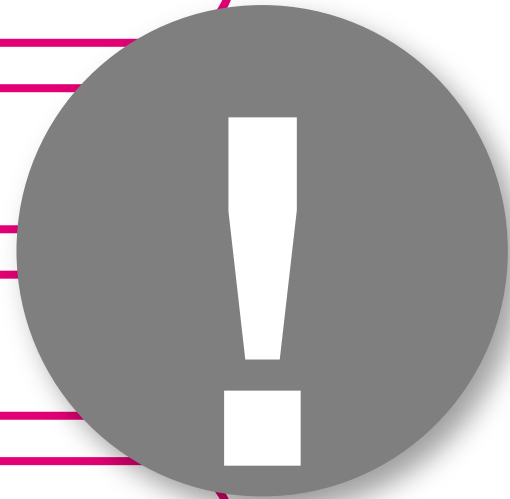


Datenintegration und -konsolidierung

Skalierbarkeit

Fehlertoleranz und Verfügbarkeit von Systemen

Analytik, Monitoring und Reporting



Technologien für Big Data Probleme

- Klassische relationale Datenbanksysteme:



- Verteilte, spaltenorientierte und In-Memory Datenbanken:



- Open Source basierte Big Data Lösungen:



.. T .. Systems ..

Google – Erfolgreich durch intelligente Big Data Lösungen

- Vielfältige Applikationen auf riesigen Datenmengen:
 - Google Web Suche
 - Google Maps / Google Earth
 - Gmail
 - YouTube
 - Picasa
 - Google+

- Wissenschaftliche Veröffentlichungen von Google:
 - The Google File System (2003)
 - MapReduce: Simplified Data Processing on Large Clusters (2004)
 - Big Table: A Distributed Storage System for Structured Data (2006)



Was ist Apache



- Entwicklung durch Doug Cutting bei Yahoo!
 - Nachimplementierung der Google Ideen
 - Veröffentlichung als Open Source Software
 - Top-Level Projekt der Apache Software Foundation



- Hauptentwickler und -nutzer unter anderem:
 - Yahoo!
 - Facebook



- Kostengünstige verteilte Big Data Lösung:
 - Open Source Software
 - 64 Bit Linux mit Java Virtual Machine
 - Commodity Hardware (Einfache Standardserver ohne RAID)



Professionelle Lösungen basierend auf Apache Hadoop

- Kommerzieller Distributionen von Apache Hadoop mit Service und Support:

cloudera


Hortonworks

MAPR™
TECHNOLOGIES

- Integration in führende DWH und BI Produkte:

ORACLE®

MicroStrategy
Best In Business Intelligence

Microsoft®
SQL Server 2012


SAP
Business Objects

TERADATA

IBM

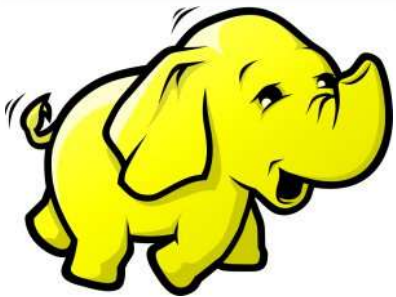
talend*
open integration solutions


pentaho™
open source business intelligence™


NETEZZA


+ a b l e a u
SOFTWARE

.. T .. Systems ..



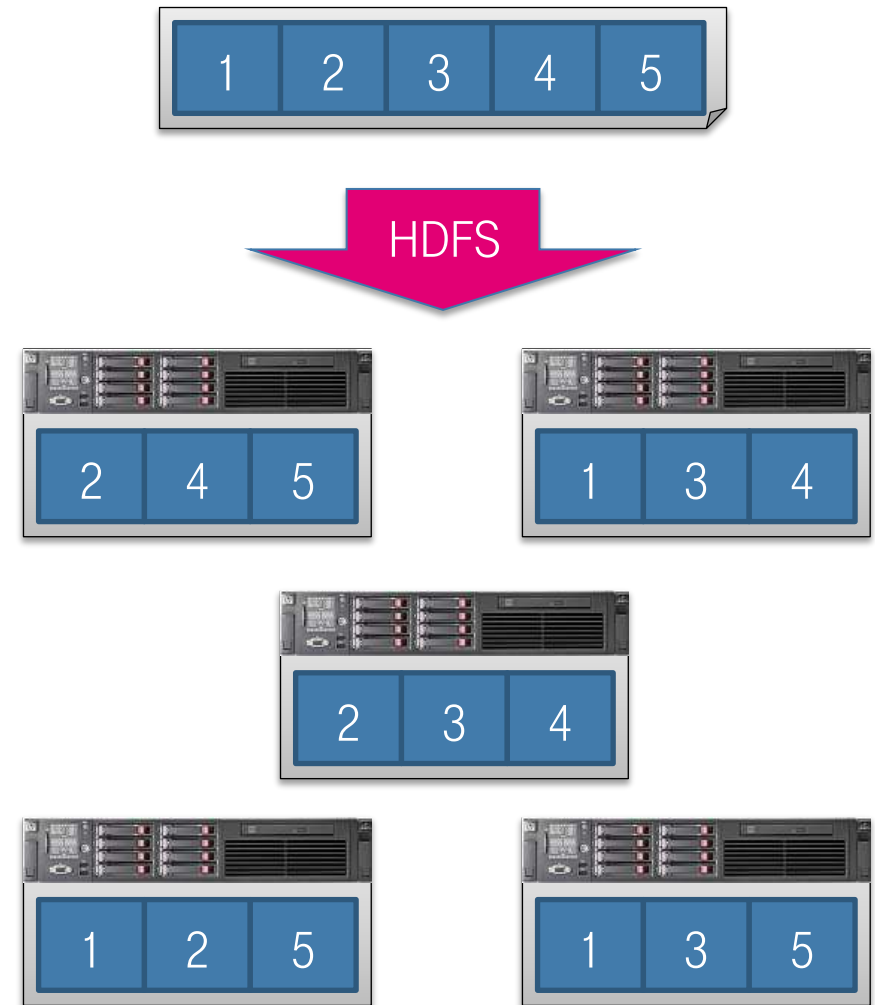
APACHE
HBASE



.. **T** .. **Systems** ..

HDFS: Hadoop Distributed File System

- Eine gegebene Datei wird in Blöcke (default=64/128 MB) aufgeteilt und diese über den Cluster repliziert (default=3).
- Optimiert für:
 - Durchsatz
 - Schreiben/Lesen/Löschen
 - Anfügen
- Block Replikation für:
 - Dauerhaftigkeit
 - Verfügbarkeit
 - Durchsatz
- Block Replikate werden über Server und Racks verteilt.



HDFS Kommandozeile

- Angelehnt an Linux Dateisystembefehle
 - ls, cp, mkdir, rm, rmdir, df, du
 - cat, tail
 - copyFromLocal, copyToLocal, moveFromLocal, moveToLocal
- Rechtesystem ähnlich zu Linux
 - chown, chgrp, chmod
- Integrationsmöglichkeiten für HDFS
 - Mount HDFS: fuse-dfs
 - HTTP/REST: WebHDFS
 - Java, C/C++ ...

```
[gunnar@W4DEUMSY9000918 ~]$ hadoop fs
Usage: hadoop fs [generic options]
[-cat [-ignoreCrc] <src> ...]
[-chgrp [-R] GROUP PATH...]
[-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
[-chown [-R] [OWNER][:[GROUP]] PATH...]
[-copyFromLocal <localsrc> ... <dst>]
[-copyToLocal [-ignoreCrc] [-crc] <src> ... <localdst>]
[-count [-q] <path> ...]
[-cp <src> ... <dst>]
[-df [-h] [<path> ...]]
[-du [-s] [-h] <path> ...]
[-expunge]
[-get [-ignoreCrc] [-crc] <src> ... <localdst>]
[-getmerge [-nl] <src> <localdst>]
[-help [cmd ...]]
[-ls [-d] [-h] [-R] [<path> ...]]
[-mkdir [-p] <path> ...]
[-moveFromLocal <localsrc> ... <dst>]
[-moveToLocal <src> <localdst>]
[-mv <src> ... <dst>]
[-put <localsrc> ... <dst>]
[-rm [-f] [-r|-R] [-skipTrash] <src> ...]
[-rmdir [--ignore-fail-on-non-empty] <dir> ...]
[-setrep [-R] [-w] <rep> <path/file> ...]
[-stat [format] <path> ...]
[-tail [-f] <file>]
[-test [-ezd] <path>]
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-usage [cmd ...]]
```

HDFS Beispiele

- Big Data Wikipediaseite lokal runterladen:
 - `curl -o bigdata.html http://en.wikipedia.org/wiki/Big_data`
- Kopieren ins HDFS:
 - `hadoop fs -copyFromLocal bigdata.html /user/gunnar/`
- Verzeichnis anzeigen:
 - `hadoop fs -ls /user/gunnar`

```
[gunnar@W4DEUMSY9000918 ~]$ hadoop fs -ls /user/gunnar
Found 5 items
drwx----- - gunnar supergroup          0 2013-02-14 01:00 /user/gunnar/.Trash
drwx----- - gunnar supergroup          0 2013-02-14 17:29 /user/gunnar/.staging
-rw-r--r--  3 gunnar supergroup    141266 2013-02-27 13:36 /user/gunnar/bigdata.html
drwxr-xr-x  - gunnar supergroup          0 2013-02-12 10:28 /user/gunnar/datasets
drwxr-xr-x  - gunnar supergroup          0 2013-01-10 16:33 /user/gunnar/test
```



Namenode Webinterface

NameNode 'WUM60070:8020'

Started: Sun Oct 28 15:42:30 CET 2012
Version: 1.0.1, r
Compiled: Tue Mar 20 12:08:58 EDT 2012 by jenkins
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[NameNode Logs](#)

Cluster Summary

215 files and directories, 135 blocks = 350 total. Heap Size is 28.53 MB / 966.69 MB (2%)

Configured Capacity	:	140.07 GB
DFS Used	:	4 GB
Non DFS Used	:	20.81 GB
DFS Remaining	:	116.07 GB
DFS Used%	:	2.85 %
DFS Remaining%	:	82.89 %
Live Nodes	:	0
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	2

NameNode Storage:

Storage Directory	Type	State
hdfs/name	IMAGE_AND_EDITS	Active

This is Apache Hadoop release 1.0.1

Contents of directory /user/gunnar

Goto:

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
.Trash	dir				2013-02-14 01:00	rwX-----	gunnar	supergroup
.staging	dir				2013-02-14 17:29	rwX-----	gunnar	supergroup
bigdata.html	file	137.96 KB	3	128 MB	2013-02-27 13:36	rw-r--r--	gunnar	supergroup
datasets	dir				2013-02-12 10:28	rwXr-xr-x	gunnar	supergroup
test	dir				2013-01-10 16:33	rwXr-xr-x	gunnar	supergroup

[Go back to DFS home](#)

NameNode 'WUM60070:8020'

Started: Sun Oct 28 15:42:30 CET 2012
Version: 1.0.1, r
Compiled: Tue Mar 20 12:08:58 EDT 2012 by jenkins
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[NameNode Logs](#)
[Go back to DFS home](#)

Live Datanodes : 3

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
WUM60071	0	In Service	46.79	1.33	6.75	38.71	2.85	<div style="width: 2.85%;"></div>	82.73	132
WUM60072	2	In Service	46.79	1.33	6.73	38.73	2.85	<div style="width: 2.85%;"></div>	82.76	132
WUM60073	0	In Service	46.79	1.33	6.83	38.62	2.85	<div style="width: 2.85%;"></div>	82.55	132

This is Apache Hadoop release 1.0.1



MapReduce – Ausnutzen von Datenlokalität

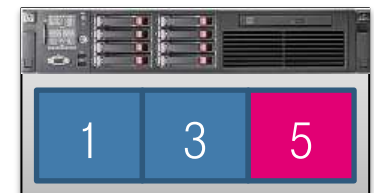
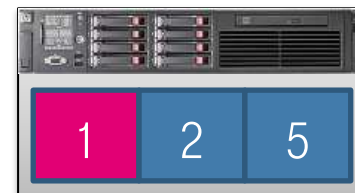
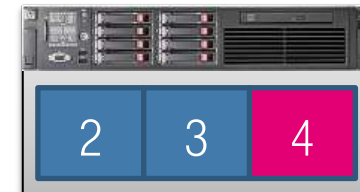
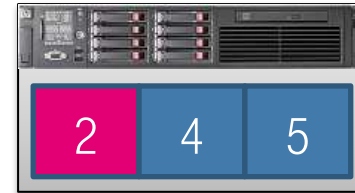
Ein gegebener Job wird in Map und Reduce Tasks aufgeteilt und diese jeweils möglichst nah zu ihren Daten eingeplant.

Programmieren von Map und Reduce Schritten in Java oder analytische Anfragesprachen (Pig, Hive)

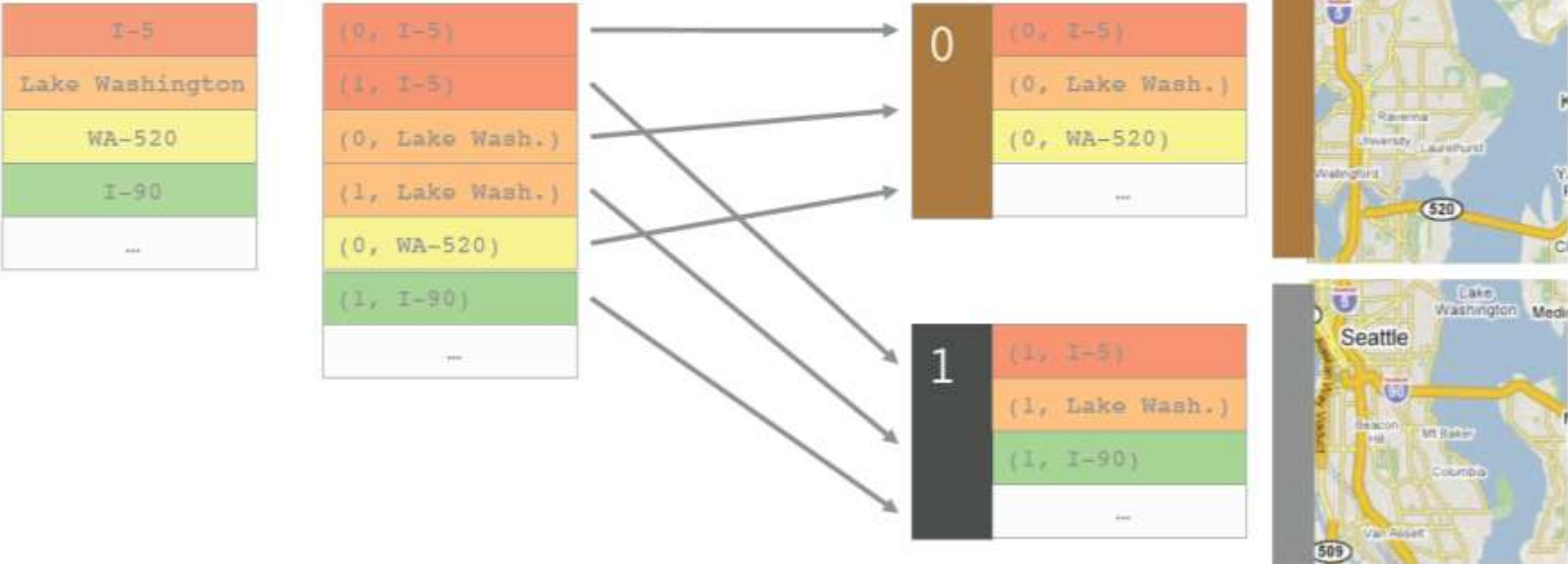
Optimiert für:

- Ausnutzen der Datenlokalität
- Batchverarbeitung
- Recovery von Fehlern

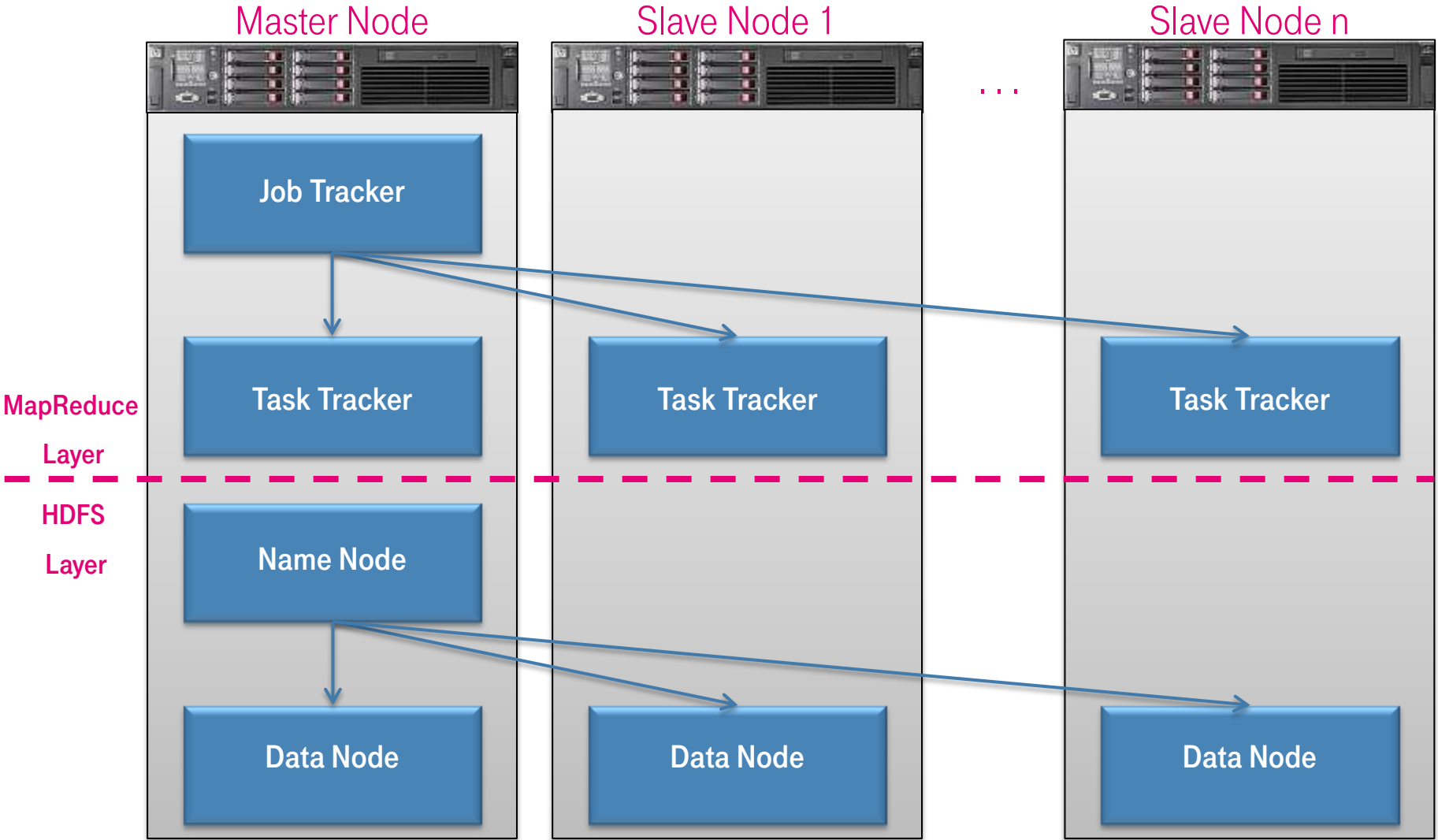
Das System erkennt hängende Tasks und führt spekulativ parallele Tasks auf den gleichen Datenteilen aus.



MapTile Rendering bei Google mit MapReduce



Die Hadoop Architektur



Anzahl Links pro URL in HTML-Seiten zählen

- Mapper
 - Bekommt HTML-Seite als Eingabe
 - Findet alle Links e.g. `Target`
 - Extrahiert die URL
 - Bereinigt die URL (z.B. `http://` entfernen)
 - Gibt pro Link eine 1 aus e.g. `www.target.com 1`
- Shuffle Phase
 - Sortiert die Key-Value Paare nach URL e.g. `www.target.com`
- Reducer
 - Bekommt einen Key und eine Liste von zugehörigen Values: `www.target.com [1,1,1,1]`
 - Summiert die Values auf und gibt ein Key-Value Paar aus: `www.target.com 4`



Links in HTML-Seiten zählen – Mapper

```
package com.tsystems.mms;
```

```
import java.io.IOException;
```

```
public class LinkCounter {
```

```
    public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {  
        private final static IntWritable one = new IntWritable(1);  
        private Text linkTarget = new Text();
```

```
        @Override
```

```
        protected void map(LongWritable key, Text text, Context context)
```

```
            throws IOException, InterruptedException {
```

```
                // Regular Expression Pattern for HTML-Links
```

```
                // e.g. <a href="www.target.com/index.html">The Target</a>
```

```
                Pattern linkPattern = Pattern.compile("<a\\b[^>]*href=\"[>]*>(.*?)</a>");
```

```
                // Regular Expression Pattern for href part e.g. href="www.target.com/index.html
```

```
                Pattern hrefPattern = Pattern.compile("href=\"[^\"]*");
```

```
                Matcher tagmatch = linkPattern.matcher(text.toString());
```

```
                while (tagmatch.find()) {
```

```
                    Matcher matcher = hrefPattern.matcher(tagmatch.group());
```

```
                    matcher.find();
```

```
                    // Remove href="http: to get www.target.com/index.html
```

```
                    linkTarget.set(matcher.group().replaceFirst("href=\"(http:)?(//)?", ""));
```

```
                    context.write(linkTarget, one);
```

```
                }
```

```
            }
```

```
    }
```



Links in HTML-Seiten zählen – Reducer

```
⊖ public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {  
⊖   @Override  
   protected void reduce(Text key, Iterable<IntWritable> values,  
       Context context) throws IOException, InterruptedException {  
       int sum = 0;  
       for (IntWritable val : values) {  
           sum += val.get();  
       }  
       context.write(key, new IntWritable(sum));  
   }  
}  
  
⊕ public static void main(String[] args) throws Exception {}  
}
```

hadoop jar ./LinkCounter.jar /user/gunnar/bigdata.html
/user/gunnar/results

```
[root@WUM60070 jar]# hadoop jar ./LinkCounter.jar /user/gunnar/bigdata.html /user/gunnar/results
13/02/27 23:04:24 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should
***hdfs://wum60070:8020/user/gunnar/bigdata.html
13/02/27 23:04:25 INFO input.FileInputFormat: Total input paths to process : 1
13/02/27 23:04:25 INFO mapred.JobClient: Running job: job_201210281542_0005
13/02/27 23:04:26 INFO mapred.JobClient: map 0% reduce 0%
13/02/27 23:04:42 INFO mapred.JobClient: map 100% reduce 0%
13/02/27 23:04:58 INFO mapred.JobClient: map 100% reduce 100%
13/02/27 23:05:03 INFO mapred.JobClient: Job complete: job_201210281542_0005
13/02/27 23:05:03 INFO mapred.JobClient: Counters: 29
13/02/27 23:05:03 INFO mapred.JobClient: Job Counters
13/02/27 23:05:03 INFO mapred.JobClient: Launched reduce tasks=1
13/02/27 23:05:03 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=16728
13/02/27 23:05:03 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
13/02/27 23:05:03 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
13/02/27 23:05:03 INFO mapred.JobClient: Launched map tasks=1
13/02/27 23:05:03 INFO mapred.JobClient: Data-local map tasks=1
13/02/27 23:05:03 INFO mapred.JobClient: SLOTS_MILLIS_REDUCE=14577
13/02/27 23:05:03 INFO mapred.JobClient: File Output Format Counters
13/02/27 23:05:03 INFO mapred.JobClient: Bytes Written=18500
13/02/27 23:05:03 INFO mapred.JobClient: FileSystemCounters
13/02/27 23:05:03 INFO mapred.JobClient: FILE_BYTES_READ=20819
13/02/27 23:05:03 INFO mapred.JobClient: HDFS_BYTES_READ=141376
13/02/27 23:05:03 INFO mapred.JobClient: FILE_BYTES_WRITTEN=85167
13/02/27 23:05:03 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=18500
13/02/27 23:05:03 INFO mapred.JobClient: File Input Format Counters
13/02/27 23:05:03 INFO mapred.JobClient: Bytes Read=141266
13/02/27 23:05:03 INFO mapred.JobClient: Map-Reduce Framework
13/02/27 23:05:03 INFO mapred.JobClient: Map output materialized bytes=20819
13/02/27 23:05:03 INFO mapred.JobClient: Map input records=999
13/02/27 23:05:03 INFO mapred.JobClient: Reduce shuffle bytes=20819
```



Jobtracker Webinterface

WUM60070 Hadoop Map/Reduce Administration

State: RUNNING
Started: Sun Oct 28 15:42:27 CET 2012
Version: 1.0.1, r
Compiled: Tue Mar 20 12:08:58 EDT 2012 by jenkins
Identifier: 201210281542

Cluster Summary (Heap Size is 28.5 MB/966.69 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node
0	1	5	3	0	1	0	0	6	6	4.00

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagnostic
job_201210281542_0005	NORMAL	root	linkcount	<input type="text" value="100.00%"/>	1	1	<input type="text" value="0.00%"/>	1	0	NA	NA

Completed Jobs

Jobid	Priority	User	Name	Map % Complete	Map Total	Maps Completed	Reduce % Complete	Reduce Total	Reduces Completed	Job Scheduling Information	Diagnostic



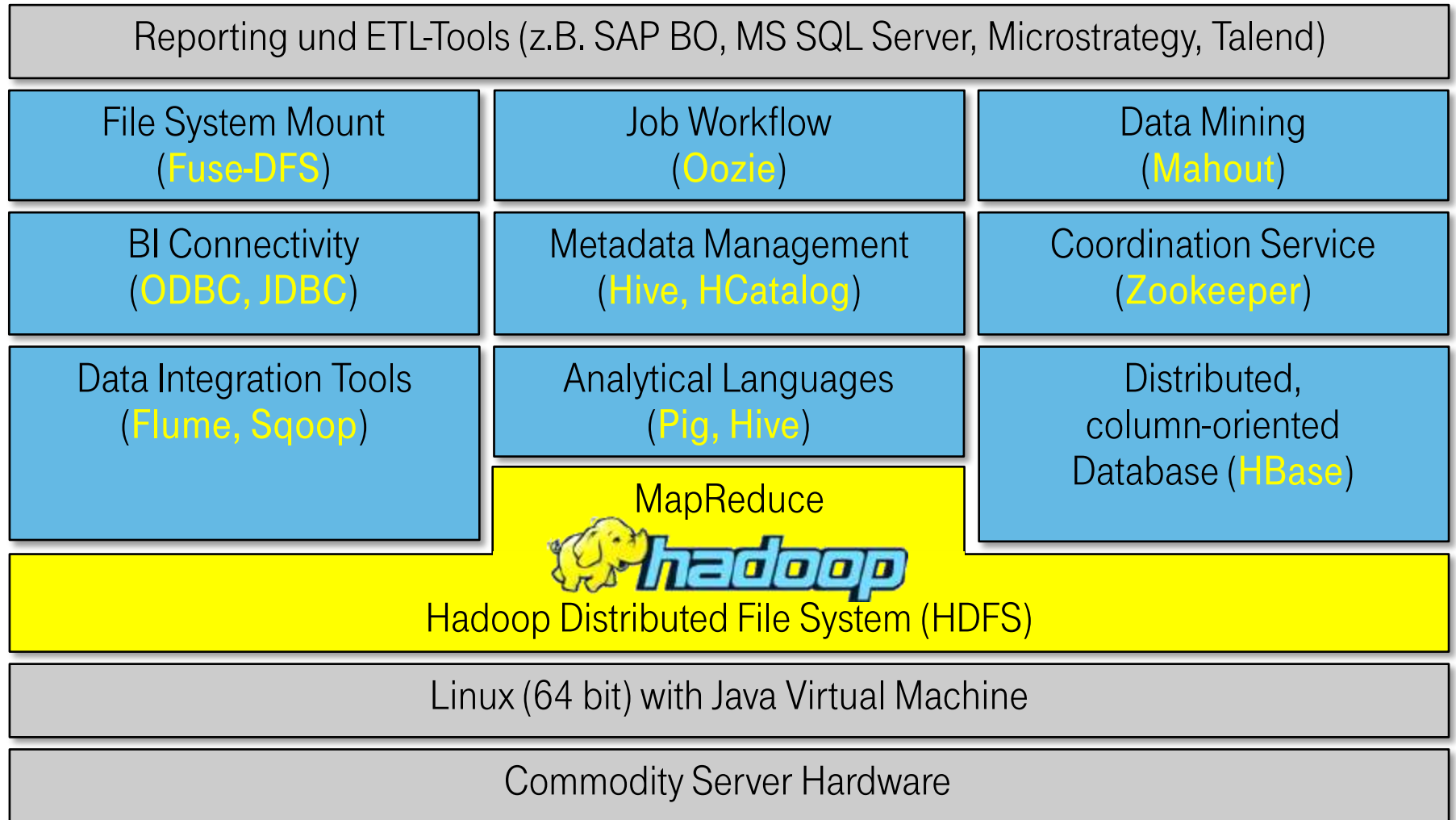
Ergebnisdatei

```
#cite_ref-ReadWriteWeb_9-0 1
#cite_ref-WH_Big_Data_30-0 1
#cite_ref-danah_58-0 1
#cite_ref-fico.com_35-0 1
#cite_ref-nature_25-0 1
#mw-head 1
#p-search 1
/w/index.php 27
/wiki/A/B_testing 1
/wiki/ACID 1
/wiki/Agent-based_model 1
/wiki/Agile_software_develop 1
/wiki/Amazon.com 1
/wiki/Anomaly_detection 1
/wiki/Apache_Hadoop 1
/wiki/Amazon.com_27-0 1
www.torbes.com/sites/benkerschberg/2012/01/09/what-technology-assi 1
www.fsn.co.uk/channel_bi_bpm_cpm/mastering_big_data_cfo_strategies 1
www.gartner.com/it/page.jsp 2
www.gartner.com/resId=2057415 2
www.guardian.co.uk/news/datablog/2012/mar/09/big-data-theory 2
www.hpcprojects.com/news/news_story.php 2
www.informationweek.com/government/enterprise-applications/image-g 1
www.mass.gov/governor/pressoffice/pressreleases/2012/2012530-gover 1
www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/B 1
www.mediawiki.org/ 1
www.nasa.gov/centers/goddard/news/releases/2010/10-051.html 1
www.nature.com/nature/journal/v455/n7209/full/455001a.html 2
www.nature.com/news/2011/110119/full/469282a.html 2
www.ncbi.nlm.nih.gov/pmc/articles/PMC3320057 1
www.ncbi.nlm.nih.gov/pmc/articles/PMC3320057/ 1
www.ncbi.nlm.nih.gov/pubmed/21310967 1
www.ncbi.nlm.nih.gov/pubmed/22482034 1
```


Klingt toll, aber mühsam?

- Apache Pig
 - Skriptsprache (Pig Latin) zum Schreiben von MapReduce Programmen
 - Entwicklung von Yahoo!
- Apache Hive
 - SQL ähnliche Anfragen die als MapReduce Programme ausgeführt werden
 - Entwicklung von Facebook
- Ich mag aber kein Java...
 - Hadoop Streaming API oder Hadoop Pipes
 - Kommunikation über StdIn StdOut oder Sockets
 - Andere Programmiersprachen (z.B. Ruby, Python, C/C++) können so Mapper und Reducer implementieren

Der Hadoop Software Stack



Hive: Datawarehousing auf Hadoop



- ein Datawarehouse-System, das auf Hadoop aufsetzt
- ursprünglich von Facebook entwickelt, seit Sommer 2008 Open-Source
- ermöglicht Datenabfragen mit einer SQL-ähnlichen Sprache (HiveQL)
- Anfragen werden in MapReduce Jobs übersetzt

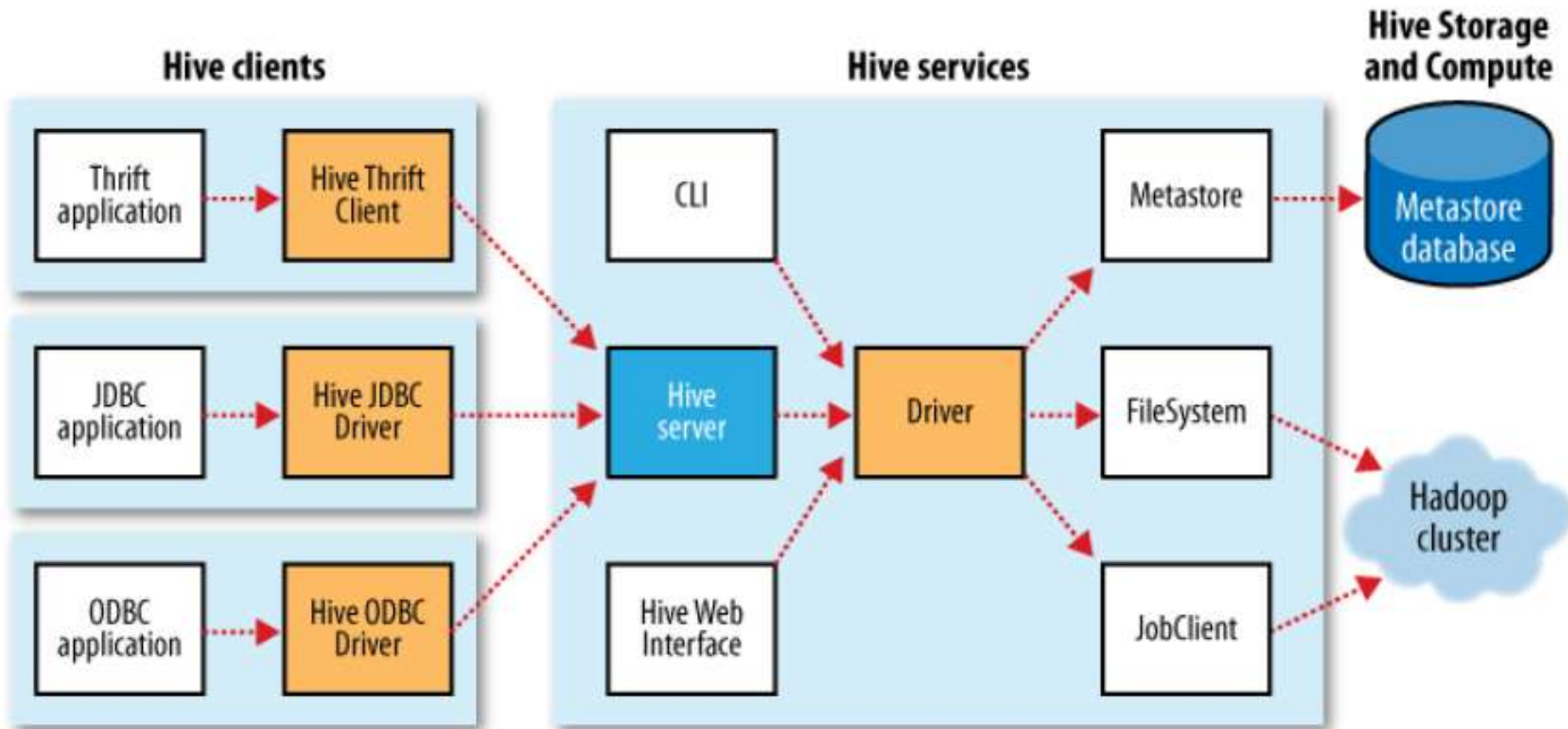
Motivation für die Entwicklung von Hive



- Probleme mit Hadoop:
 - Schwierigkeiten mit der Programmierung von Map-Reduce-Skripten
 - Viele Analysten mit schwachen Java-Kenntnissen aber SQL Erfahrung
 - Fehlende Schnittstelle an Reporting-Systeme
- Hive bietet vereinfachte Analysemöglichkeiten:
 - Hive interpretiert Logfiles, CSV-Files o.ä. als Tabellen und definiert ein Datenbankschema auf Dateien im HDFS
 - Hive übersetzt SQL-Skripte in MapReduce-Jobs
 - Auch Analysten ohne Java-Kenntnisse können Abfragen in Hadoop machen
 - Hive ermöglicht die Anbindung an Reporting-Tools (JDBC, ODBC)
 - 95% der Hadoop-Jobs bei Facebook werden über Hive durchgeführt



Hive-Architektur



Anlegen einer Hive-Tabelle

```
hive> CREATE TABLE outgoinglinks (  
  > url STRING,  
  > count INT)  
  > ROW FORMAT DELIMITED  
  > FIELDS TERMINATED BY '\t';  
OK  
Time taken: 0.146 seconds  
hive> show tables;  
OK  
outgoinglinks  
Time taken: 0.096 seconds  
hive> describe outgoinglinks;  
OK  
url      string  
count    int  
Time taken: 0.135 seconds  
hive> select * from outgoinglinks;  
OK  
Time taken: 0.388 seconds  
hive> exit;  
[root@WUM60070 jar]# hadoop fs -ls /user/hive/warehouse/  
Found 1 items  
drwxr-xr-x  - root supergroup          0 2013-02-28 15:58 /user/hive/warehouse/outgoinglinks  
[root@WUM60070 jar]#
```

Befüllen eine Hive Tabelle

- `hadoop fs -mv /user/gunnar/results/part-r-0000 /user/hive/warehouse/outgoinglinks`
- Beladen von Hive Tabellen = Kopieren von Dateien ins HDFS
- Hive arbeitet mit Rohdaten
- Sehr schnelle Beladung des Warehouses möglich
- Hinzufügen von Files (z.B. Logfiles) fügt Daten zur Tabelle hinzu
- Aber:
 - Keine einzelnen Inserts oder Updates
 - Keine Transaktionen
 - Kein ACID



Top 5 ausgehenden Links der Seite

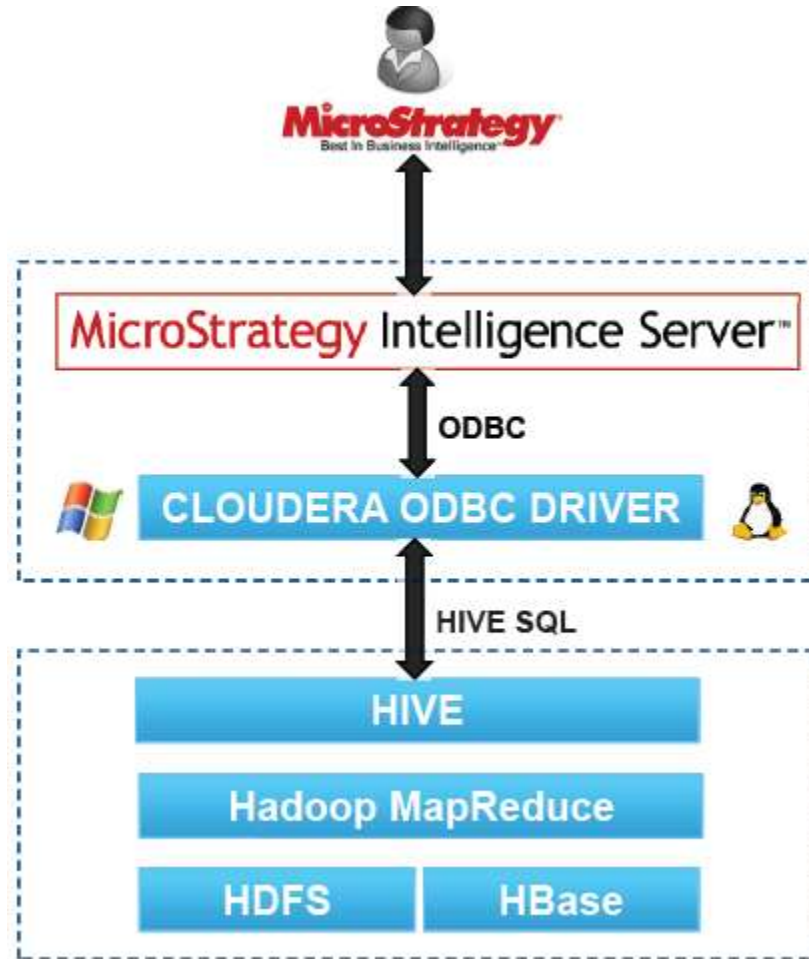
- Query: SELECT * FROM outgoinglinks ORDER BY count DESC, url LIMIT 5;

```
2013-02-28 16:19:53,852 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.56 sec
2013-02-28 16:19:54,858 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.56 sec
2013-02-28 16:19:55,863 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.56 sec
2013-02-28 16:19:56,870 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.56 sec
MapReduce Total cumulative CPU time: 3 seconds 560 msec
Ended Job = job_201210281542_0008
MapReduce Jobs Launched:
Job 0: Map: 1 Reduce: 1 Accumulative CPU: 3.56 sec HDFS Read: 18721 HDFS Write: 141 SUCESS
Total MapReduce CPU Time Spent: 3 seconds 560 msec
OK
/w/index.php 27
en.wikipedia.org/w/index.php 8
#cite_note-Economist-5 7
/wiki/Digital_object_identifier 7
#cite_note-HilbertBigData2013-39 4
Time taken: 32.118 seconds
hive>
```

- Fazit:
 - Viel langsamer auf kleinen Datenmengen (> 20 sec)
 - Sehr gute Skalierung auf großen Datenmengen (Tablescans auf vielen Giga/Terabytes)



Integration in Reporting System wie z.B. Microstrategy, SAP BO



•• T •• Systems ••

- Nachimplementierung von Googles Big Table Konzept (wie Cassandra)
 - Big Table: A Distributed Storage System for Structured Data (2006)



- Googles Big Table und deren Nachfolger sind wichtiger Bestandteil für Google Suche
- HBase ist Basis für Facebooks Messaging System



- Eigenschaften:
 - Columnstore Architektur (Columnfamilies und Kompression)
 - Automatische horizontale Partitionierung von Tabellen (Autosharding) basierend auf deren Zeilenschlüssel
 - Ausgelegt auf hohe Last von Random Reads und Random Writes
 - Nahezu lineare Skalierung durch Hinzufügen von Knoten
 - Flexible Anzahl von Spalten und dünnbesetzte Tabellen



Beispiel: Nutzerverwaltung in einem Forum

- Unterschiedliche Arten von Daten:
 - Schlüssel: Nutzername
 - Name, Vorname, Alter, Status
 - About me Seite (längere HTML Seite)
 - Nutzerbild (JPEG)
- Nutzername ist Schlüssel und wird für die Verteilung genutzt (Nicht gleichverteilt!)
- Drei Columnfamilies
 - userdata
 - aboutme
 - pic
- Unterschiedliche Kompression möglich für Columnfamilies
- Erweiterbarkeit: zusätzliche Nutzerdaten, Nutzerbilder in unterschiedlichen Größen



Anlegen und Befüllen einer Tabelle auf dem CLI

```
hbase(main):020:0> create 'usertable', 'userdata', 'aboutme', 'pic'
0 row(s) in 1.0320 seconds

hbase(main):021:0> put 'usertable', 'gunnar', 'userdata:fname', 'Gunnar'
0 row(s) in 0.0440 seconds

hbase(main):022:0> put 'usertable', 'gunnar', 'userdata:lname', 'Schröder'
0 row(s) in 0.0080 seconds

hbase(main):023:0> put 'usertable', 'gunnar', 'userdata:lname', 'Fabritius'
0 row(s) in 0.0250 seconds

hbase(main):024:0> scan 'usertable'
ROW          COLUMN+CELL
 gunnar      column=userdata:fname, timestamp=1362068810961, value=Gunnar
 gunnar      column=userdata:lname, timestamp=1362068861322, value=Fabritius
1 row(s) in 0.0160 seconds

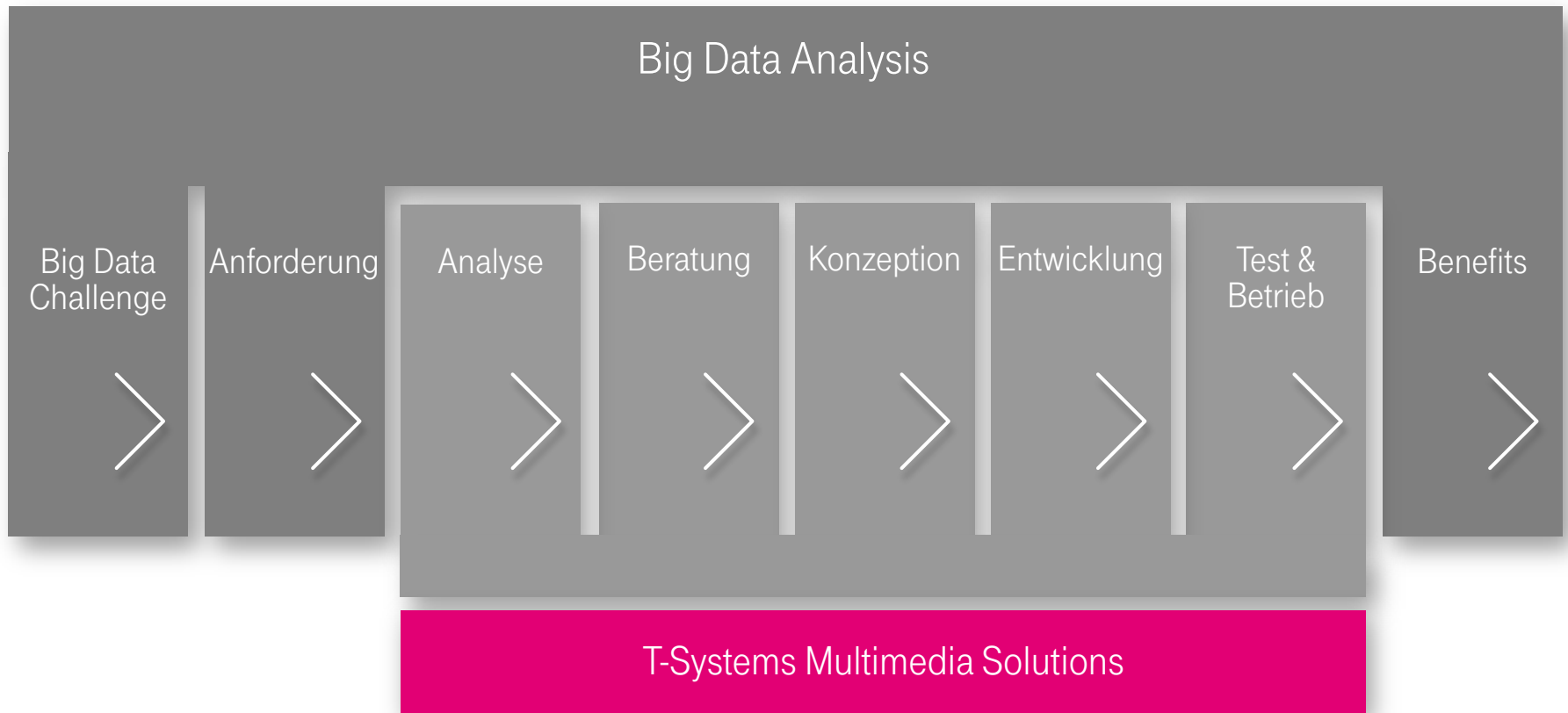
hbase(main):025:0> scan 'usertable', {VERSIONS => 2}
ROW          COLUMN+CELL
 gunnar      column=userdata:fname, timestamp=1362068810961, value=Gunnar
 gunnar      column=userdata:lname, timestamp=1362068861322, value=Fabritius
 gunnar      column=userdata:lname, timestamp=1362068855497, value=Schr\xC3\xB6der
1 row(s) in 0.0720 seconds
```

Key Business Values von neuartigen Big Data Technologien

- Flexibilität
 - Speichere jegliche Art von Daten
 - Analysiere jegliche Art von Daten
- Skalierbarkeit von Systemen
 - Von einem Terabyte auf drei Knoten bis zu Petabytes auf tausenden Knoten
 - On-Demand Speicher- und Rechenkapazität in der Cloud (Elastic Computing)
- Langlebigkeit von Daten
 - Wesentlich geringere Kosten pro Terabyte Speicherplatz



Wir unterstützen Sie bei der Lösung Ihrer Big Data Herausforderungen!



T-Systems – Ihr Partner für Big Data Lösungen

- Interesse an Big Data, Hadoop und T-Systems MMS?
- Wir suchen:
 - Motivierte und engagierte Mitarbeiter
 - Werksstudenten
 - Absolventen für Bachelor und Masterarbeiten



Gunnar Schröder
Technical Consultant

T-Systems Multimedia Solutions GmbH
Hausanschrift Rieser Straße 5, 01129 Dresden
Telekontakte Telefon +49 351 2820-2282
Telefax +49 351 2820-4282
E-Mail: gunnar.schroeder@t-systems.com

