ORDIX® best practice

einfach. gut. beraten.

# Big Data und SQL - das passt!

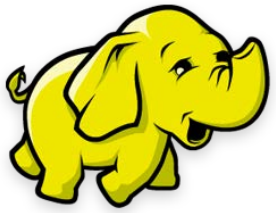
DOAG
Konferenz + Ausstellung 2015
19.11.2015, Nürnberg
Philipp Loer

info@ordix.de www.ordix.de

# Agenda

ORDIX AG

- Einführung in Hadoop
- Big Data und SQL – passt das wirklich?
- Apache Hive
- Apache Sqoop
- Oracle Big Data Appliance



# Agenda

ORDIX AG

- Einführung in Hadoop
- Big Data und SQL – passt das wirklich?
- Apache Hive
- Apache Sqoop
- Oracle Big Data Appliance
- Live Demo

# Relationale Datenbank vs. Big Data

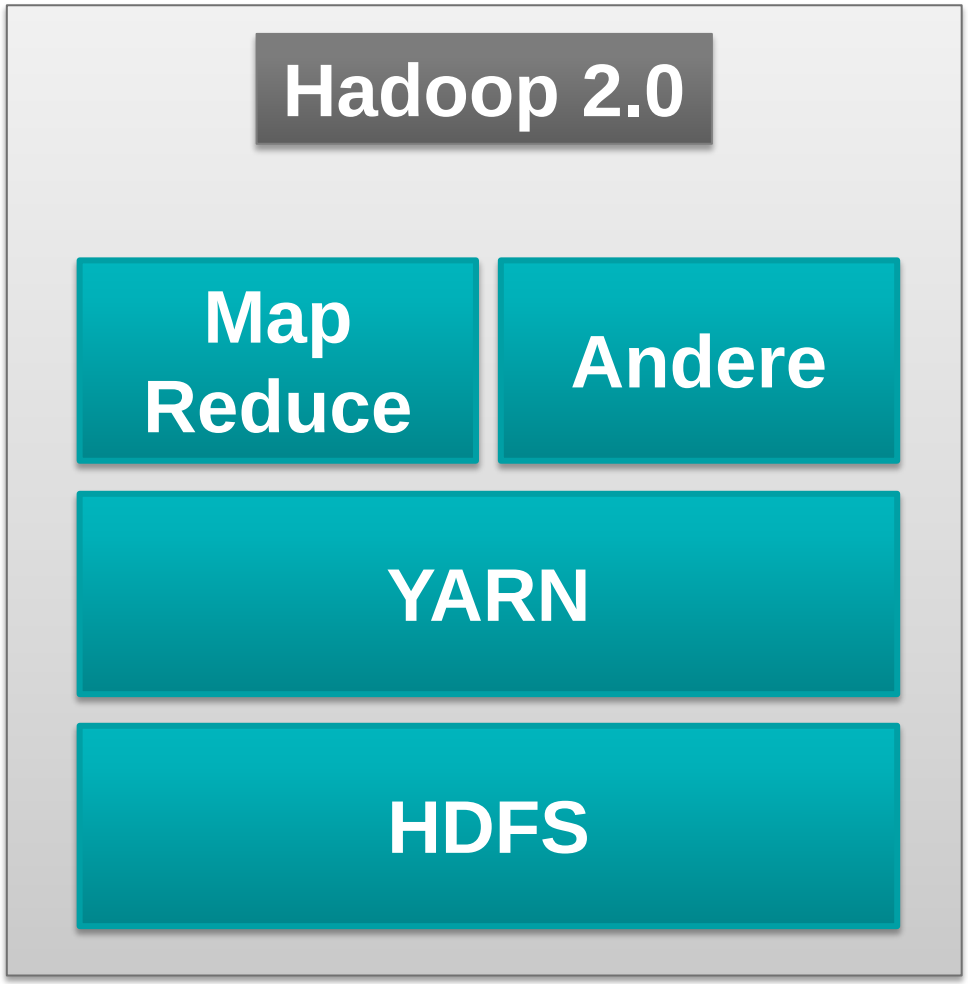
ORDIX AG

## Relationale Datenbank


## Big Data

# Hadoop-Komponenten

ORDIX AG



# Hadoop vs. RDBMS

ORDIX AG

| RDBMS | Hadoop |
|---|---|
| Nur strukturierte Daten | Un-, semi- oder strukturierte Daten |
| Schema on write | Schema on read |
| Hohe Datenintegrität | Geringe Datenintegrität |
| Nicht linear skalierbar | Nahezu linear skalierbar |
| Write / Read / Upadate many times | Write once / Read many |

# Agenda

ORDIX AG

- Einführung in Hadoop
- Big Data und SQL – passt das wirklich?
- Apache Hive
- Apache Sqoop
- Oracle Big Data Appliance
- Live Demo

# Warum kein SQL?

ORDIX AG

- Geringere Flexibilität
  - Big Data ist nicht relational: Informationen können und werden mehrfach vorgehalten
  - Informationen müssen nicht in Tabellen vorgehalten werden sondern können z. B. auch in JSON-Dokumenten abgelegt werden
- Datenwachstum
  - Klassische relationale Datenbanken sind nicht auf die Speicherung von „Big Data“ ausgelegt

Volume | Velocity | Variety | Value | Veracity

# SQL oder kein SQL? (I)

ORDIX AG

- SQL ist standardisiert
- SQL skaliert, ist vielseitig und hat sich bewährt
- SQL funktioniert als Abfragesprache auch auf nicht-relationalen Modellen
- SQL ist frei kombinierbar mit anderen Konzepten zur Datendarstellung und Datenlagerung (wie JSON, XML, etc.)
- SQL reduziert Aufwand und Komplexität in der Entwicklung
- SQL ermöglicht interaktive Abfragen
- SQL kann durch User Defined Functions (UDF) erweitert werden

# SQL oder kein SQL? (II)

ORDIX AG

- SQL beschreibt lediglich das gewünschte Ziel und macht keine Vorgaben bezüglich der Datenverarbeitung
- In Big-Data-Umgebungen ist SQL nur eine von vielen Möglichkeiten zur Datenabfrage und -manipulation
- Nicht alle Bestandteile von SQL eignen sich für einen Einsatz im Big-Data-Umfeld
- Overhead in verteilten Umgebungen deutlich größer als in klassischen Datenbanken
- Haupteinsatzgebiet: „Langläufer“

**Nicht jede Aufgabe kann mit einem SQL gelöst werden. Aber SQL kann einen sinnvollen Beitrag zur Lösung darstellen.**

# Agenda

ORDIX AG

- Einführung in Hadoop
- Big Data und SQL – passt das wirklich?
- Apache Hive
- Apache Sqoop
- Oracle Big Data Appliance
- Live Demo

# SQL und Big Data
Hive-Alternativen

ORDIX AG

Spark SQL


presto

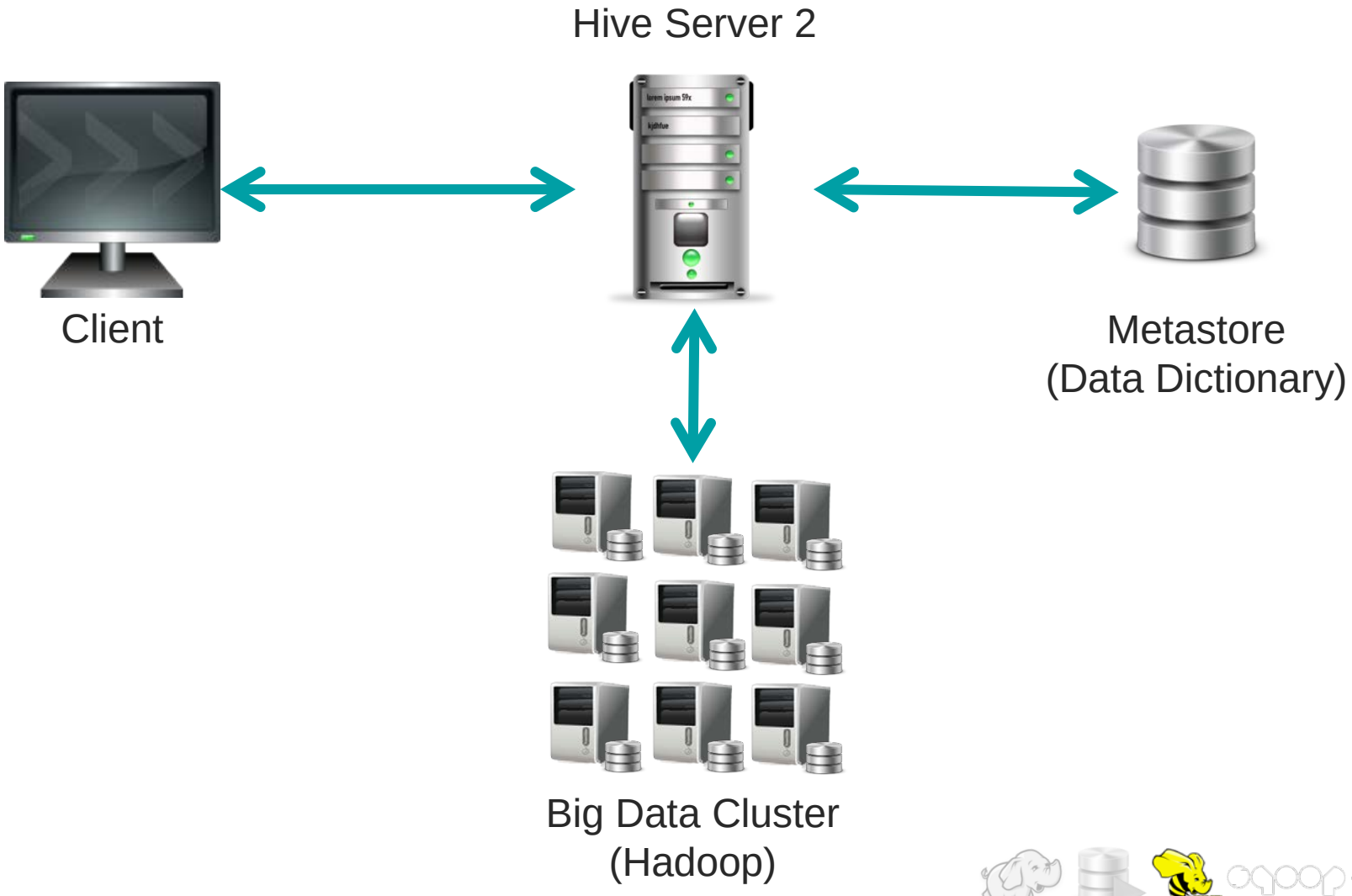
HAWQ

APACHE DRILL

hadapt

# Hive-Architektur

ORDIX AG



# Hive
## Physikalische Speicherung

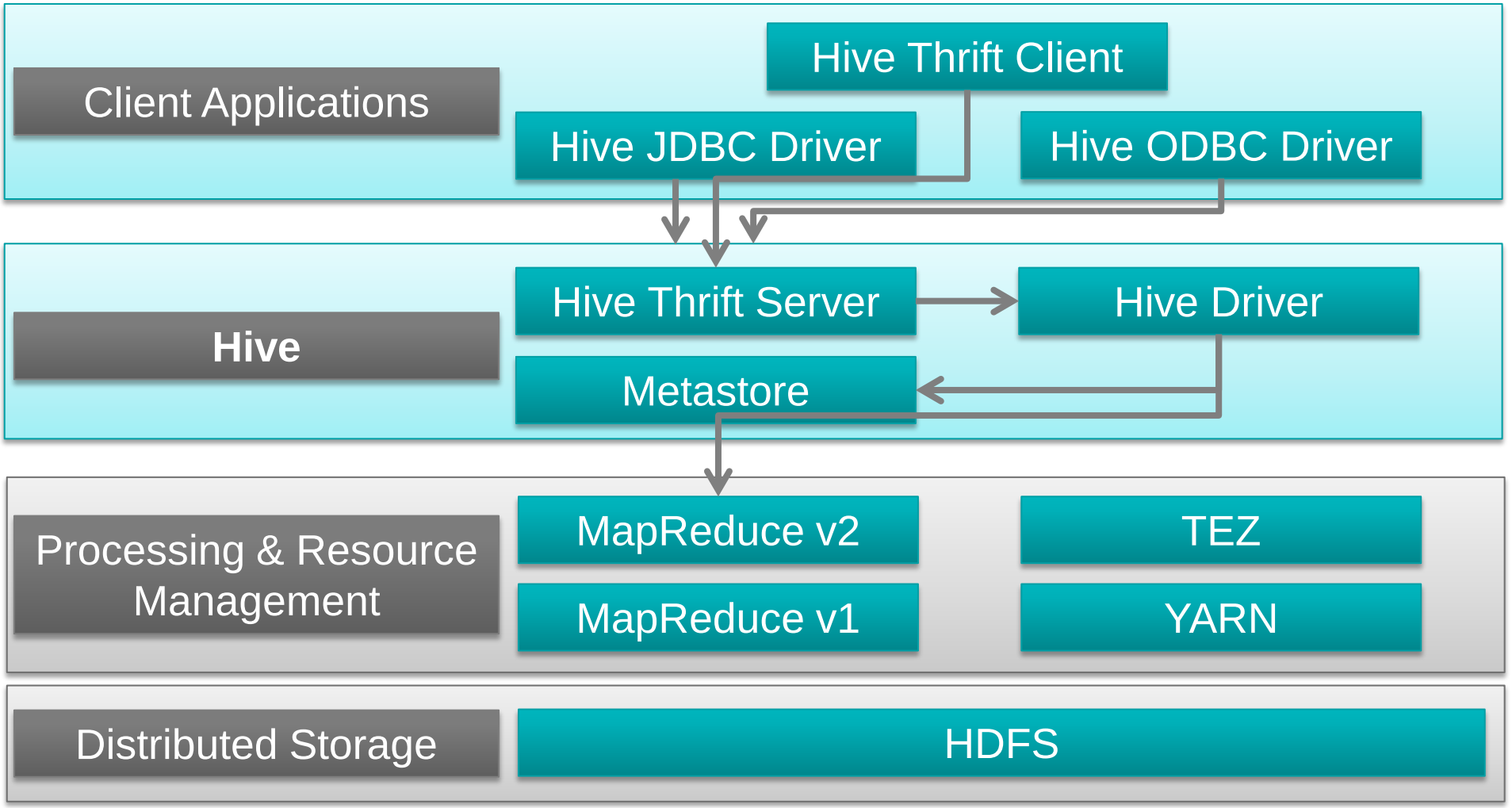
ORDIX AG

- Speicherung der Daten erfolgt weiterhin in HDFS.
- Jede Tabelle (bzw. Partition) wird in einem eigenen Verzeichnis im HDFS-Dateisystem gespeichert.
- Interne Tabellen der Default-Datenbank befinden sich in Unterverzeichnissen des HDFS-Verzeichnisses **`/user/hive/warehouse.`** Für weitere Datenbanken wird hier ein Unterverzeichnis **`<Datenbankname.db>`** erstellt.
- Partitionen liegen in Unterverzeichnissen des Tabellenverzeichnisses.
- Die Partition **`Entwickler`** der nach der Spalte **`Job`** partitionierten Tabelle **`emp`** der Hive-Datenbank **`Scott`** befände sich im Verzeichnis **`/user/hive/warehouse/scott.db/emp/job=entwickler/`**
- Die Daten externer Tabellen befinden sich ebenfalls in HDFS, jedoch außerhalb dieser Verzeichnisstruktur.

# Hive-Architektur

Hive Server 2

ORDIX AG



# Hive DML

ORDIX AG

| DML-Typ | Verfügbar seit |
|---|---|
| Insert (Batch) | 0.1.0 |
| Insert (Single Row) | 0.14.0 (November 2014) |
| Update | 0.14.0 (November 2014) |
| Delete | 0.14.0 (November 2014) |
| Merge | - |
| Query | 0.1.0 |

# Hive Create Table

Beispiel

ORDIX AG

```
CREATE TABLE IF NOT EXISTS page_view(
      viewTime INT,
      userid BIGINT,
      page_url STRING,
      ip VARCHAR(64) COMMENT 'IP Address of the User'
)
COMMENT 'This is the page view table‘
PARTITIONED BY(country STRING)
CLUSTERED BY(userid) SORTED BY(viewTime) INTO 256 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\;'
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

# Hive-Datentypen

ORDIX AG

- Primitiv
  - Int (Tinyint, Smalling, Bigint)
  - Float, Double, Decimal (ab 0.13.0 mit Precision und Scale)
  - Boolean, String, Varchar, Char, Binary, Timestamp, Date
- Komplex
  - Stuct
  - Map
  - Array
  - Uniontype

# Hive-Dateiformate und Kompressionsalgorithmen

ORDIX AG

- Dateiformate
  - Text File (e.g. CSV)
  - SequenceFile
  - RC File (Record Columnar File)
  - ORC (Optimized Row Columnar)
  - Avro
  - Parquet
- Kompressionsalgorithmen
  - GZIP
  - Snappy (Google)
  - Deflate
  - BZIP2
  - LZO

# Hive DML
Insert

ORDIX AG

- Insert AS SELECT

```
INSERT INTO TABLE tablename1
[PARTITION (partcol1=val1, partcol2=val2 ...)]
select_statement
FROM from_statement;
```

- Multiple Table Insert

```
FROM from_statement
INSERT OVERWRITE TABLE tablename1 [PARTITION
(partcol1=val1, partcol2=val2 ...)[IF NOT EXISTS]]
select_statement1
[INSERT OVERWRITE TABLE tablename2 [PARTITION ... ]
select_statement2
```

# Hive DML

Insert aus SQL

ORDIX AG

```
CREATE TABLE students (
       name VARCHAR(64),
       age INT,
       gpa DECIMAL(3,2)
)
CLUSTERED BY (age) INTO 2 BUCKETS STORED AS ORC;

INSERT INTO TABLE students VALUES (
       'fred flintstone',
       35,
       3.28),(
       'barney rubble',
       32,
       1.32
);
```

# Hive DML
## Insert mit LOAD

ORDIX AG

- Ziel: Laden der Daten aus HDFS nach Hive
- Syntax:

```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE] INTO
TABLE tablename [PARTITION (partcol1=val1, partcol2=val2
...)]
```

- Beispiel

```
LOAD DATA
INPATH '/user/ordix/data/weather/station-data'
INTO TABLE weather_station_data;
```

# Hive DML
Select (I)

ORDIX AG

- SELECT
- FROM
- WHERE
- GROUP BY / HAVING
- ORDER BY / SORT BY / DISTRIBUTE BY / CLUSTER BY
- LIMIT
- UNION

# Hive DML

## Select (II)

ORDIX AG

- Relationale Operatoren (<, >, =, LIKE, RLIKE)
- INNER JOIN, OUTER JOIN (LEFT, RIGHT, FULL) in der ANSI-Syntax
- Subqueries (Correlated ab Hive 0.13.0)
- Mathematische Operatoren (+, -, Modulo etc.)
- Konditionale Funktionen (IF, Coalesce, Case)
- Eingebaute Funktionen (round, floor, rand, lower, to_date)
- Aggregationen (min, max, avg, count, std, variance)
- User Defined Functions

# Hive Security
## Authentification

ORDIX AG

- None
- Kerberos
- LDAP
- PAM
- Custom

# Hive Security
## Authorization

ORDIX AG

- Storage Based Authorization
  - Hive User hat direkten Zugang zu den Daten in HDFS
  - Berechtigungen erfogen in HDFS nach dem POSIX-Standard bzw. über Accesss Control Lists (ACL)
  - Keine Beschränkung auf einzelne Spalten oder Zeilen möglich
  - Der Zugang zu den Metadaten kann von Hive äquivalent beschränkt werden
- SQL Standards Based Authorization in HiveServer2 (ab Hive 0.13.0)
  - Erlaubt Fine Grained Access Control
  - Zugangsbeschränkung per Grant- / Revoke Statement

# Agenda

ORDIX AG

- Einführung in Hadoop
- Big Data und SQL – passt das wirklich?
- Apache Hive
- Apache Sqoop
- Oracle Big Data Appliance
- Live Demo

# Was kann Sqoop?

ORDIX AG

- Import und Export von Daten aus strukturierten Datenspeichern (RDBMS, NoSQL-DB etc.)
- Zusammenarbeit mit Oozie (Big Data Workflow Management), um Import- und Export-Aufgaben zu ermöglichen
- Sqoop besitzt eine mehrschichtige Architektur, die es ermöglicht neue Datenspeicher durch die Erstellung spezifischer Konnektoren anzubinden

# Warum Sqoop?

ORDIX AG

- RDBMS → Hadoop
  - Einige Daten müssen konsistent vorgehalten werden
  - Durchführung der Berechnungen im RDBMS zu aufwändig/teuer
- Hadoop → RDBMS
  - Direkter Zugriff auf externe Daten aus Hadoop-Prozessen heraus erfordert komplexere Anwendungen
  - Gefahr der Überlastung des RDBMS durch multiple, parallele Zugriffe der Knoten des Hadoop Cluster

# Verarbeitung in Sqoop

ORDIX AG

- Sqoop läuft im Hadoop Cluster
- Sqoop hat Zugang zu Hadoop Core
  - Map-Tasks werden für parallelen Import genutzt
  - Daten werden in HDFS geschrieben
- Sqoop Import: RDBMS/NoSQL → Hadoop
- Sqoop Export: Hadoop → RDBMS/NoSQL
- Datenmenge wird partitioniert, jede Partition wird von einem Map-Task gelesen und geschrieben
- Typsichere Übertragung: Metadaten der Datenbank werden ausgewertet, um passende Java-Datentypen zu finden (z. B. VARCHAR2(30) zu String)

# Sqoop-Kommandozeile

ORDIX AG

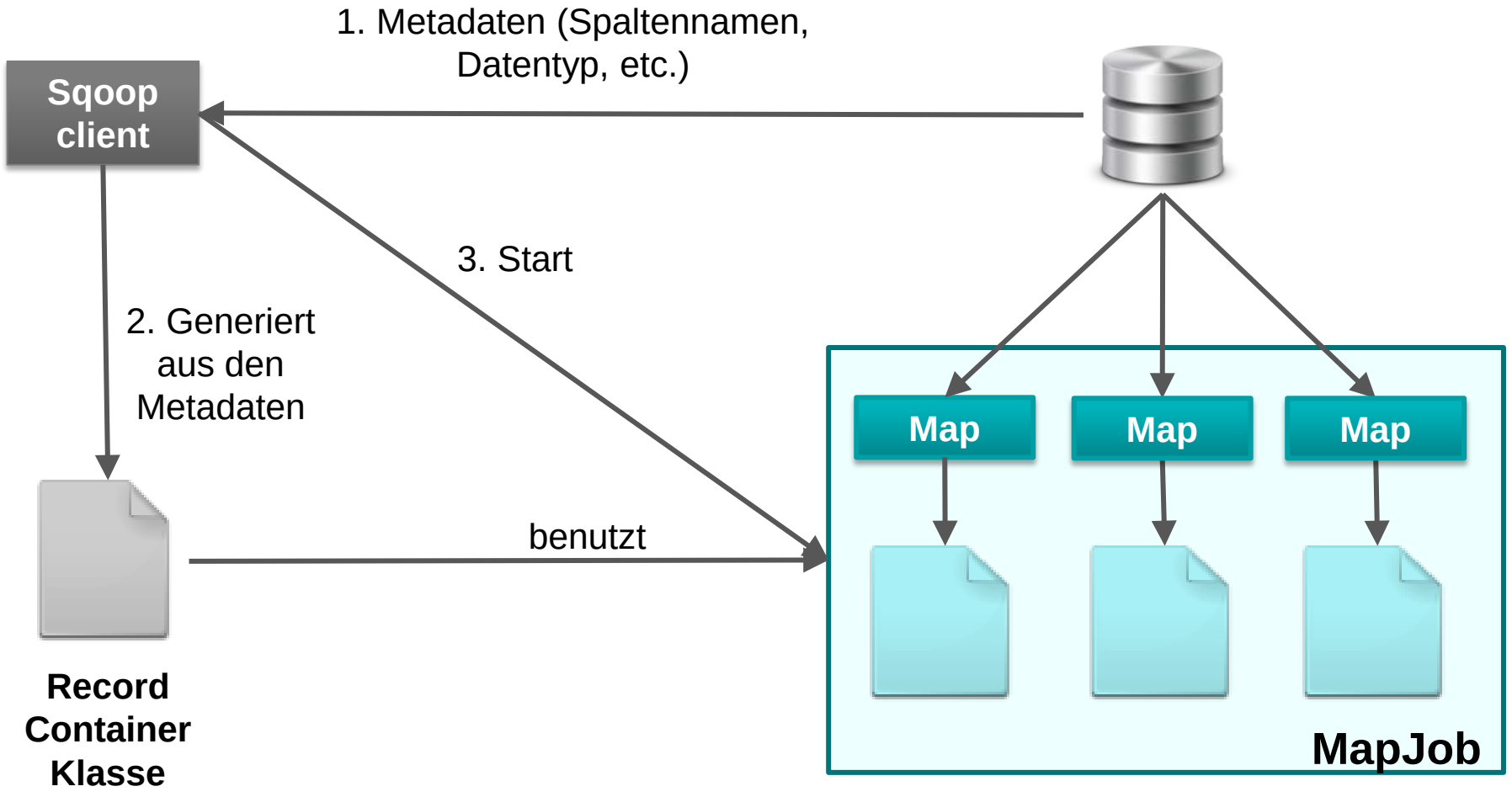
- Sqoop ist ein kommandozeilenbasiertes Werkzeug
- Wichtigste Befehle: **`sqoop import`** bzw. **`sqoop export`**
- Beispiel:

```
sqoop import \
--connect jdbc:oracle:thin:@//ppl_centos65:1521/ordix \
--table EMP \
--username scott \
--password tiger
```

- Speicherung der Daten in HDFS, per default als csv-Datei
  - Andere Dateiformate sind möglich
- Import- und Export-Befehle haben weitere Optionen

# Ablauf eines Imports mit Sqoop

ORDIX AG



# Sqoop Import
Hive

ORDIX AG

- Daten können durch Sqoop direkt in Hive-Tabellen importiert werden.
- Zu verwendende Option: **`--hive-import`**
- Während des Hive-Imports werden die Daten von den Datentypen des Quellsystems in Hive-Datentypen konvertiert (z. B. ab Hive 1.2: VARCHAR2 in VARCHAR).
- Dateiformat entspricht dem Dateiformat, mit dem die Tabelle erstellt wurde (z. B. Textfile mit Tabulator als Trennzeichen, SequnceFile etc.)
- Nach Abschluss des Imports kann die Tabelle wie eine „normale“ Hive-Tabelle verwendet werden.

# Sqoop-Konnektoren

ORDIX AG

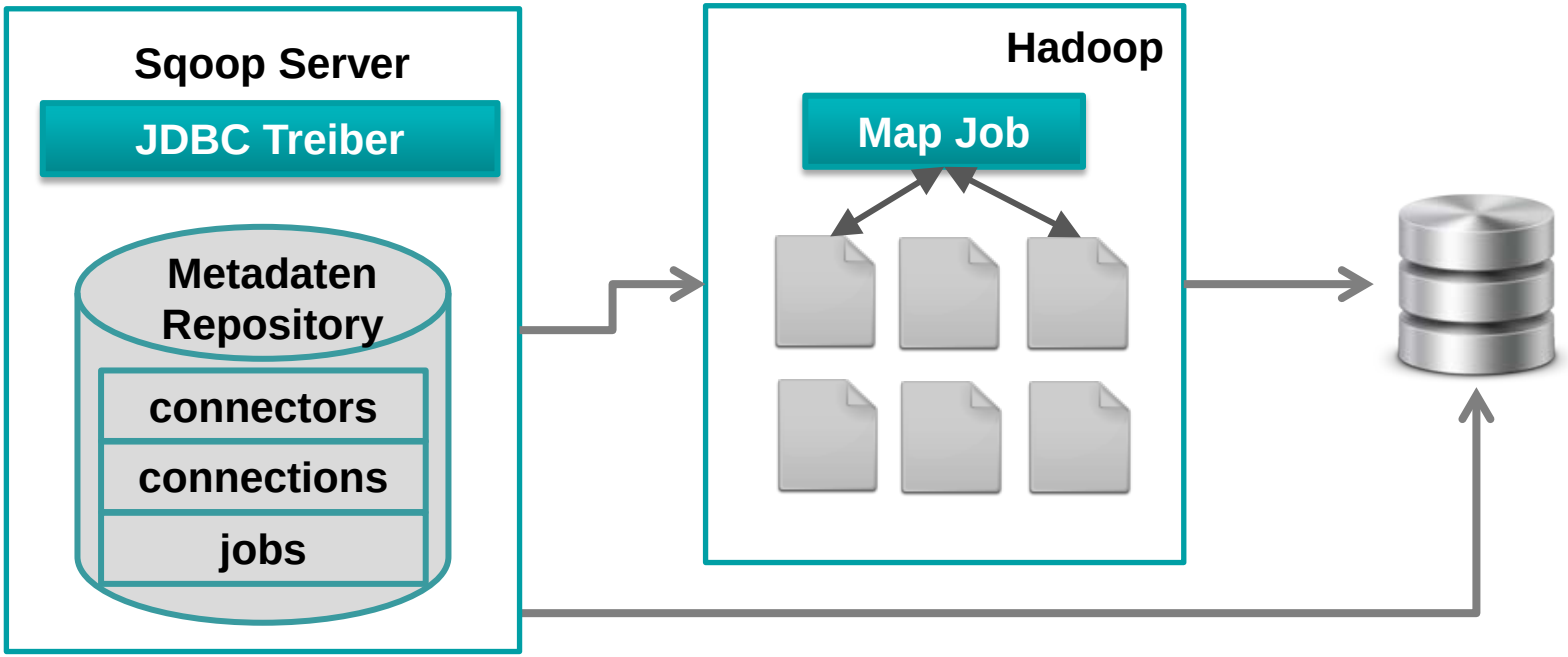
- Generic JDBC:
  Kann genutzt werden, um auf jede Datenbank zuzugreifen, die JDBC unterstützt.
- Default Connector:
  Unterstützt den Zugriff auf Oracle, MySQL, PostgreSQL, SQL Server und DB2.
- Fast Path Connector:
  MySQL und PostgreSQL
- OraOOP:
  Fast Path Connector für Oracle von Cloudera

# Sqoop Server 2

Coming soon…

ORDIX AG

- Neuimplemtierung von Sqoop mit Client-Server-Modell
- Derzeit (Stand September 2015) nicht für den produktiven Einsatz vorgesehen



# Agenda

ORDIX AG

- Einführung in Hadoop
- Big Data und SQL – passt das wirklich?
- Apache Hive
- Apache Sqoop
- Oracle Big Data Appliance
- Live Demo

# Gibt’s da nicht auch was von Oracle? (I)

ORDIX AG

- Oracle Big Data Connectors
  - Ermöglichen den schnellen Zugriff aus der Oracle-Datenbank auf Hadoop
  - Transferraten bis zu 5 TB/h
  - Transformation von Daten aus verschiedenen Formaten (JSON, XML, Avro u. a.)
- Oracle Big Data SQL
  - Erlaubt Abfragen aus Hadoop direkt aus der Oracle-Datenbank
  - Ermöglicht die Kombination der Daten aus der Oracle-Datenbank mit Daten aus NoSQL und Hadoop in einer Abfrage
  - Erweiterung der Oracle Zugriffs- und Authentifizierungsmechanismen auf die Hadoop-Welt
- Voraussetzung für beide:
  - Erfüllung der Oracle Big Data Appliance
  - Verwendung der Oracle Exadata

# Gibt’s da nicht auch was von Oracle? (II)

ORDIX AG

- Oracle Big Data Appliance
- Besteht aus drei Komponenten
  - 1 Rack aus 18 Sun x86 Servern (Full Variante, Starter: 6 Server)
  - Cloudera Distribution mit Apache Hadoop für die Datenerfassung und -organisation
  - Oracle NoSQL Database Community Edition zur Datenerfassung
- Kosten:

| Item | Cost | Notes |
|---|---|---|
| *Appliance (18 node)* | *$525,000* | Oracle’s list price |
| *Hardware and Software Support* | *$189,000* | $63k/year for 3 years of full support |
| *Installation* | *$14,150* | By Oracle professional services |
| ***Total*** | ***$728,150*** | |

*Source: Oracle, 2014.*

- zzgl. Kosten für Oracle Exadata (~300.000 - 2 Mio. USD inkl. drei Jahre Support)
- zzgl. Weitere Systeme für Entwicklung, Test, Backup etc.

# Vielen Dank für Ihre Aufmerksamkeit!

ORDIX AG

Zentrale Paderborn
Westernmauer 12 - 16
33098 Paderborn
Tel.: 05251 1063-0
Fax: 0180 1 67349 0

Seminarzentrum Wiesbaden
Kreuzberger Ring 13
65205 Wiesbaden
Tel.: 0611 77840-00

Weitere Geschäftsstellen
in Essen, Gersthofen,
Köln und Münster

info@ordix.de
www.ordix.de