More and more data is available to science and business!

Drivers:
- Cloud Computing
- Internet of Services
- Internet of Things
- Cyberphysical Systems

Underlying Trends:
- Connectivity
- Collaboration
- Computer generated data

- video streams
- sensor data
- web archives
- audio streams
- simulation data
- RFID data
Data are becoming increasingly complex!

- Size (volume)
- Freshness (velocity)
- Format/Media Type (variability)
- Uncertainty/Quality (veracity)
- etc.

Data
Data and analyses are becoming increasingly complex!

Data

- Size
- Freshness
- Format/Media Type
- Uncertainty/Quality
- etc.

(\textit{volume})
(\textit{velocity})
(\textit{variability})
(\textit{veracity})

Analysis

- Selection/Grouping
- Relational Operators
- Extraction & Integration,
- Data Mining
- Predictive Models
- etc.

(\textit{map}/\textit{reduce})
(Join/\textit{Correlation})
(\textit{map}/\textit{reduce} or dataflow systems)
(R, S\textsuperscript{+}, Matlab)
(R, S\textsuperscript{+}, Matlab)
Data-driven applications …

… will revolutionize decision making in business and the sciences!

… have great economic potential!

Big Data Analytics | Volker Markl | BDOD Big Data – Chances and Challenges
Slide 5
General consensus seems to be that people (would like to) have lots of data, which they would like to analyze. How do we go about that?
Running in Circles?

---

SQL -- NoSQL

---

SQL

---

Running in circles
Re-implement some code paths of parallel databases or run restricted non-compatible SQL on Hadoop?

---

NoMapReduce

---

SQL
What is Wrong with this Picture?

- scripting
- SQL--
- XQuery?
- wrong platform?

- column store++
- a query plan
- scalable parallel sort
Big Data Technologies Worldwide

- **MapReduce und Tools**: Hadoop, Cloudera Impala (USA), InfoSphere BigInsights (IBM)
- **Data Mining und Informationsextaktion**: SystemT (IBM),
- **Statistical Packages**: R (Universität Auckland), Matlab (Mathworks;USA), SAS, SPSS
- **DBMS**: Aster (Teradata), Greenplum (EMC), DB2 (IBM), SQLServer (Microsoft), Oracle Exadata (Oracle)
- **Business Analytics/Intelligence**: SAS Analytics (SAS), Vertica (HP), SPSS IBM
- **Script Sprachen**: Pig, Hive, JAQL
- **Graph Databases**: Neo4j (USA), AllegroGraph (Franz Inc.; Giraph (Apache)
- **Visualization**: Tableau (Tableau Software; USA),
- **Suche / Indexing**: Lucene (Apache), Solr (Apache)
- **Other**: UIMA (IBM)
<table>
<thead>
<tr>
<th>System</th>
<th>Type</th>
<th>Stakeholder</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParStream</td>
<td>DBMS (in-memory)</td>
<td>ParStream</td>
</tr>
<tr>
<td>SAP Hana</td>
<td>DBMS (in-memory)</td>
<td>SAP</td>
</tr>
<tr>
<td>Hyper</td>
<td>DBMS (in-memory)</td>
<td>TU München</td>
</tr>
<tr>
<td>BigMemory</td>
<td>DBMS (in-memory)</td>
<td>Terracotta Inc./Software AG</td>
</tr>
<tr>
<td>Scalaris</td>
<td>Storage (parallel)</td>
<td>Konrad-Zuse-Zentrum für Informationstechnik</td>
</tr>
<tr>
<td>RapidMiner</td>
<td>Data mining toolkit</td>
<td>Rapid-I</td>
</tr>
<tr>
<td>ImageMaster</td>
<td>Enterprise Content Management</td>
<td>T-Systems</td>
</tr>
<tr>
<td>SalesIntelligence</td>
<td>Market research</td>
<td>Implisense</td>
</tr>
<tr>
<td>Predictive-Analytics-Suite</td>
<td>Business Analytics</td>
<td>Blue Yonder</td>
</tr>
<tr>
<td>Stratosphere</td>
<td>Scalable Data Analytics System</td>
<td>TU Berlin</td>
</tr>
</tbody>
</table>
3 Big Trends in Big Data

- Solving complex data analysis problems
  - Predictive analytics
  - Infinite data streams
  - Distributed data
  - Dealing with uncertainty
  - Bringing the human in the loop: Visual analytics

- Producing results in near-real time
  - First results fast
  - Low latency
  - Exploiting new Hardware
    (manycore, OpenGL/CUDA, FPGA, heterogeneous processors, NUMA, remote memory)

- Hiding complexity
  - Declarative data analysis programs
    - aggregation
    - relational algebra
    - iterative algorithms
    - distributed state
  - Fault tolerance (pessimistic/optimistic)
Another “V“ - Value:
Big Data in the Cloud - the Information Economy

A major new trend in information processing will be the trading of original and enriched data, effectively creating an information economy.

„When hardware became commoditized, software was valuable. Now that software is being commoditized, data is valuable.“ (Tim O’Reilly)

„The important question isn’t who owns the data. Ultimately, we all do. A better question is, who owns the means of analysis?“ (A. Croll, Mashable, 2011)
Information Marketplaces: Enabling SMEs to capitalize on Big Data

Dataplayers

Data & Aggregation

Revenue Sharing

Technology

Licensing

Queries

Analytical results

Marketplace

e.g., Social Media Monitoring

e.g., Media Publisher Services

e.g., SEO

Index

Massively Parallel Infrastructure

Distributed Data Storage

Trust

German Web

http://www.mia-marktplatz.de

http://www.dopa-project.eu

Big Data Analytics | Volker Markl | BDOD Big Data – Chances and Challenges
Slide 13
Big Data looks Tiny from the Stratosphere

Stratosphere is a European declarative, massively-parallel Big Data Analytics System, funded by DFG and EIT, available open-source with Apache License.

Data analysis program

```plaintext
employees = read 'employees.json';
$result = transform $emp in $employees into {
  taxes: $emp.brutto - $emp.netto
  address: {
    $emp.address.*,
    country: 'Germany'
  }
};
write $result to 'output.json';
```

Execution plan

Stratosphere optimizer
Automatic selection of parallelization, shipping and local strategies, operator order and placement

Runtime operators
Hash- and sort-based out-of-core operator implementations, memory management

Parallel execution engine
Task scheduling, network data transfers, resource allocation, checkpointing

http://www.stratosphere.eu
### Comparison

<table>
<thead>
<tr>
<th></th>
<th>MapReduce</th>
<th>Pregel</th>
<th>Stratosphere/Naiad</th>
<th>GraphLab</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Programming Model</strong></td>
<td>Fixed Functions – Map and Reduce</td>
<td>Supersteps over a data graph with messages passed</td>
<td>Iterative dataflow with operators and UDFs</td>
<td>Data graph with shared data table and update functions</td>
</tr>
<tr>
<td><strong>Parallelism</strong></td>
<td>Concurrent execution of tasks within map and reduce phases</td>
<td>Concurrent execution of user functions over vertices within a superstep</td>
<td>Concurrent execution of operators during a stage</td>
<td>Concurrent execution of non-overlapping scopes, defined by consistency model</td>
</tr>
<tr>
<td><strong>Data Handling</strong></td>
<td>Distributed file system</td>
<td>Distributed file system</td>
<td>Flexible data channels: Memory, Files, DFS etc.</td>
<td>Undefined – Graphs can be in memory or on disk</td>
</tr>
<tr>
<td><strong>Task Scheduling</strong></td>
<td>Fixed Phases – HDFS Locality based map task assignment</td>
<td>Partitioned Graph and Inputs assigned by assignment functions</td>
<td>Job and Stage Managers assign operators to available daemons/tasks</td>
<td>Pluggable schedulers to schedule update functions</td>
</tr>
<tr>
<td><strong>Fault Tolerance</strong></td>
<td>DFS replication + Task reassignment / Speculative execution of Tasks</td>
<td>Checkpointing and superstep re-execution</td>
<td>Operators/Task failure recovery</td>
<td>Synchronous and asynchronous snapshots</td>
</tr>
<tr>
<td><strong>Developed by</strong></td>
<td>Google</td>
<td>Google</td>
<td>TU Berlin / Microsoft</td>
<td>Carnegie Mellon</td>
</tr>
</tbody>
</table>

“Scalable Machine Learning for Big Data” Tutorial by Microsoft at the IEEE International Conference on Data Engineering 2013
5 Mistakes Data Scientists Make When Putting Big Data To Work

- Selecting a platform that scales
  - Performance (runtime, first-results fast)
  - People

- Selecting the right data
  - Information Quality
  - Timeliness

- Choosing the right model
  - Assumptions about the data

- Reproducability
  - Numerical stability

- Trusting your results
  - Confidence and statistical significance
  - Lineage
“Big Data“ is Interdisciplinary

- **Legal Dimension**: Copyright, Privacy, Liability
  - Digital Preservation, Decision Making, Verticals

- **Application Dimension**: Scalable Data Processing, Signal Processing, Statistics/ML, Linguistics, HCI/Visualization

- **Social Dimension**: User Behaviour, Societal Impact, Collaboration

- **Technology Dimension**: Business Models, Benchmarking, Impact of Open Source, Deployment, Pricing

- **Economic Dimension**:
Call to Action

- **Educate Data Scientists to create the required talent**
  - T-shaped students
  - Information „literacy“
  - Data Analytics Curriculum

- **Research Big Data Analytics Technologies**
  - Data management (uncertainty, query processing under near real-time constraints, information extraction)
  - Programming models
  - Machine learning and statistical methods
  - Systems architectures
  - Information Visualization

- **Innovate to maintain competitiveness**
  - Demonstrate flagship use-cases to raise awareness
  - Promote startups in the area of data analytics
  - Transfer technologies to enterprises, in particular SMEs
  - Determine legal frameworks and business models

We need to ensure a European technological leadership role in „Big Data“
7 commandments for Big Data Analytics
1: Thou shall use declarative languages

All-pairs shortest paths using recursive doubling in Stratosphere’s Scala front-end

```scala
val vertices = ...; val edges = ...
val result = findSP iterate {
  s0 = vertices
  map { v -> Pathv, v, 0 } }
val distinctBy { p -> {p.from, p.to} },
ws0 = edges
}
def findSP = is: DataStream[Path], ws: DataStream[Path] => {
def wsHops(x) =
  ws join x
  on { p -> p.to } isEqualTo { p -> p.from }
  map { (p1, p2) -> Path(p1.from, p2.to, p1.dist + p2.dist) }
val sNewPaths = wsHops(s)
val wsNewPaths = wsHops(ws)
val s1 =
sNewPaths cogroup s
  on { p -> {p.from, p.to} } isEqualTo { p -> {p.from, p.to} }
  map { (p1, p2) -> (p1 ++ p2) minBy { _._dist } }
val ws1 =
s1 cogroup wsNewPaths
  on { p -> {p.from, p.to} } isEqualTo { p -> {p.from, p.to} }
  flatMap { (p5, pWS) -> if (p5.isEmpty) pWS else empty }
(s1, ws1)
```
2: Thou shall accept external (dynamic) sources

“In situ” data - no load

3: Thou shall use rich primitives

*Beyond MapReduce*

Map

Match

Reduce

Cross

CoGroup
4: Thou shall deeply embed UDFs

Concise and flexible

5: Thou shall optimize

Static code analysis of UDFs permits auto-parallelization and optimization à la relational databases
6: Thou shall iterate

Needed for most interesting analysis cases

Pregel as a Stratosphere query plan with comparable performance.
7: Thou shall use a scalable and efficient execution engine

Pipeline and data parallelism, flexible checkpointing, optimized network data transfers
An Overview of Stratosphere
Meteor program

```lua
$employees = read 'employees.json';
$result = transform $emp in $employees
  into {
    taxes: $emp.brutto - $emp.netto
    address: {
      $emp.address.*,
      country: 'Germany'
    }
  }
write $result to 'output.json';
```

Pact program

Meteor compiler

Stratosphere optimizer
- Picks data shipping and local strategies, operator order

Execution plan

Job graph

Execution graph

Runtime
- Hash- and sort-based out-of-core operator implementations, memory management

Nephele execution engine
- Task scheduling, network data transfers, resource allocation, checkpointing
Stratosphere bulk iterations

Currently making the optimizer iteration-aware and exposing PACT-level abstraction
PACT4S: Embedding of PACTs in Scala

Master thesis developed language and Scala compiler plug-in, ongoing
Master thesis will integrate in query optimizer, still in (very) prototypical mode
Ingredients of a Big Data Project

- Technology
  - (data preparation, scalable processing)
- Mathematical analysis methods
- Application
- Database systems, scalable platforms
- Machine learning/statistics/optimization
- Real-world analysis problem
- Visualization
- NLP
- Signal processing
- ...