Research, Innovation and Teaching in Big Data Analytics

Challenges and Chances

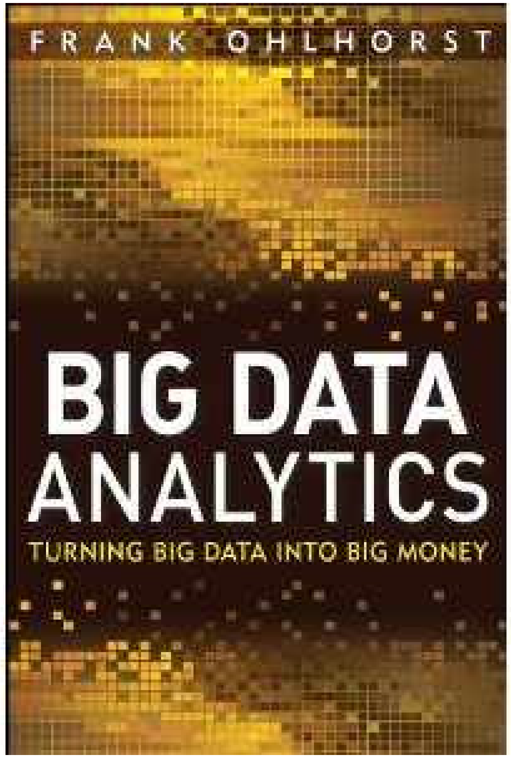
Prof. Dr. Volker Markl

Chair of the Database Systems and Information Management Lab
Technische Universität Berlin

www.dima.tu-berlin.de

Source: McKinsey

Source: Information Week



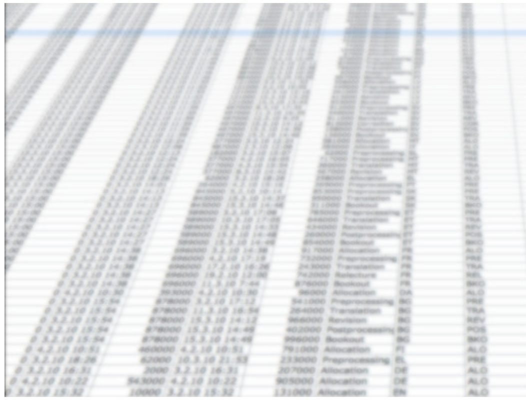
Source: Amazon

Source: NCSU

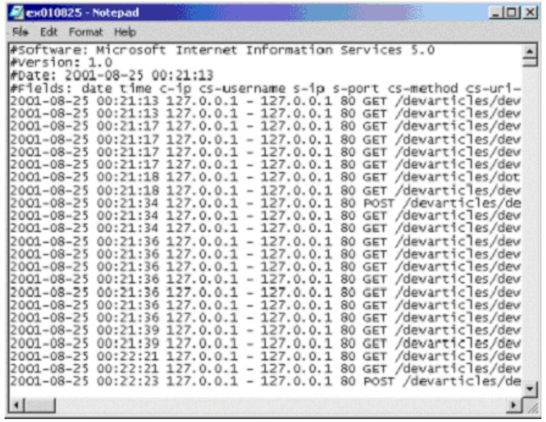
Agenda

- What is Big Data?
- Examples
- Parallel Data Processing and Other Enabling Technologies
- Applications
- Challenges
- Call to Action

„Medium Data Analytics“

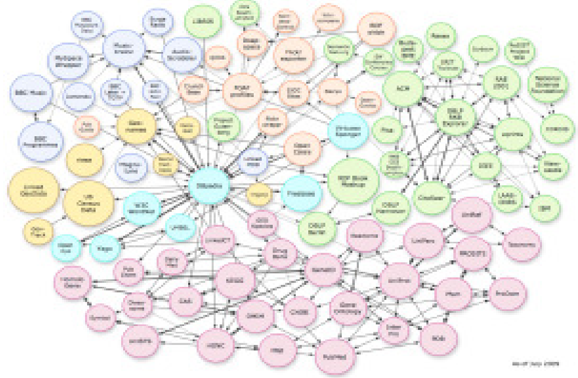


Transactional Data at „Webscale“



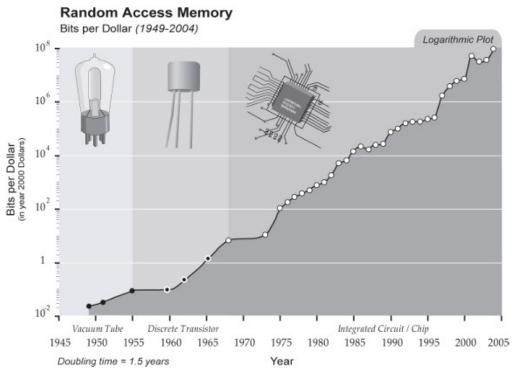
```
ex010825 - Notepad
File  Edit  Format  Help
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2001-08-25 00:21:13
#Fields: date time c-ip cs-username s-ip s-port cs-method cs-uri-
2001-08-25 00:21:13 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:13 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:17 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:17 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:17 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:17 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:18 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dot
2001-08-25 00:21:18 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:34 127.0.0.1 - 127.0.0.1 80 POST /devarticles/de
2001-08-25 00:21:34 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:34 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:36 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:39 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:21:39 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:22:21 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:22:21 127.0.0.1 - 127.0.0.1 80 GET /devarticles/dev
2001-08-25 00:22:23 127.0.0.1 - 127.0.0.1 80 POST /devarticles/de
```

Web Logfiles

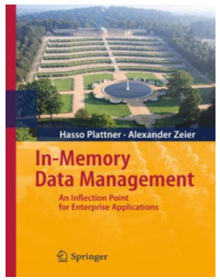


Linked Open Data
and many other
„human generated data sets“

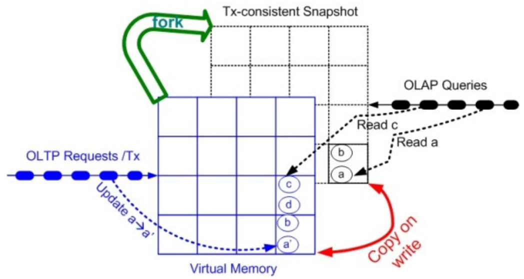
These data sets will fit into main memory soon!



IBM ISAO



SAP HANA



TUM Hyper

Many standard data management solutions exist!

Big Data for Real

Will Computers Crash Genomics?

New technologies are making sequencing DNA easier and cheaper than ever, but the ability to analyze and store all that data is lagging

Lincoln Stein is worried. For decades, computers have improved at rates that have boggled the mind. But Stein, a bioinformaticist at the Ontario Institute for Cancer Research (OICR) in Toronto, Canada, works in a field that is moving even faster: genomics.

The cost of sequencing DNA has taken a nosedive in the decade since the human genome was published—and it is now dropping by 50% every 5 months. The amount of sequence available to researchers has consequently skyrocketed, setting off warnings about a "data tsunami." A single DNA sequencer can now generate in a day what it took 10 years to collect for the Human Genome Project. Computers are central to archiving and analyzing this information, notes Stein, but their processing power isn't increasing fast enough, and their costs are decreasing too slowly, to keep up with the deluge. The torrent of DNA data and the need to analyze it "will swamp our storage systems and crush our computer clusters," Stein predicted last year in the journal *Genome Biology*.

Funding agencies have neglected bioinformatics needs, Stein and others argue. "Traditionally, the U.K. and the U.S. have not invested in analysis; instead, the focus has been investing in data generation," says computational biologist Chris Ponting of the University of Oxford in the United Kingdom. "That's got to change."

Within a few years, Ponting predicts, analysis, not sequencing, will be the main expense hurdle to many genome projects. And that's assuming there's someone who can do it; bioinformaticists are in short supply everywhere. "I worry there won't be enough people around to do the analysis," says Ponting.

Recent reviews, editorials, and scientists' blogs have echoed these concerns (see Perspective on p. 728). They stress the need for new software and infrastructures to deal with computational and storage issues.

In the meantime, bioinformaticists are trying new approaches to handle the data onslaught. Some are heading for the clouds—cloud computing, that is, a pay-as-you-go service, accessible from one's own desktop, that provides rented time on a large cluster of machines that work together in parallel as fast as, or faster than, a single powerful computer. "Surviving the data deluge means computing in parallel," says Michael Schatz, a bioinformaticist at Cold Spring Harbor Laboratory (CSHL) in New York.

Dizzy with data

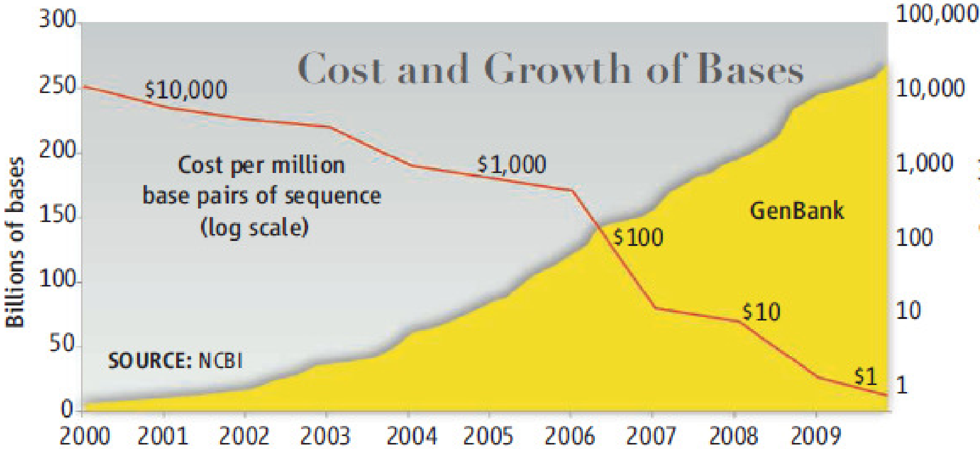
The balance between sequence generation and the ability to handle the data began to shift after 2005. Until then, and even today, most DNA sequencing occurred in large centers, well equipped with the computer personnel and infrastructure to support the analysis of a genome's data. DNA sequences churned out by these centers were deposited and stored in centralized public databases, such as those run by the European Bioinformatics Institute (EBI) in Hinxton, U.K., and the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Researchers elsewhere could then download the data for study. By 2007, NCBI had 150 billion bases of genetic information stored in its GenBank database.

Then several companies in quick succession introduced "next-generation" machines, faster sequencers that spit out data more cheaply. But the technologies behind these machines generate such short stretches of sequence—typically just 50

666 11 FEBRUARY 2011 VOL 331 SCIENCE www.sciencemag.org

Published by AAAS

It is cheaper to resequence than to store genome data!



SCIENCE 11 FEBRUARY 2011
VOL 331, ISSUE 6018, PAGES 639-806

Other Examples of Really Big Data Sets

Smart Grids

Video Streams

Web Archive

EXABYTE

(1,152,921,504,606,846,976 BYTES; 2^{60})

approx. 1,000,000,000,000,000,000 or 10^{18}

5 EXABYTES: ALL WORDS EVER SPOKEN BY HUMAN BEINGS

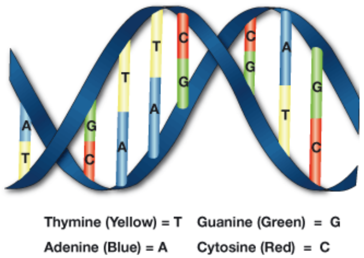
Sensor Data

ZETTABYTE

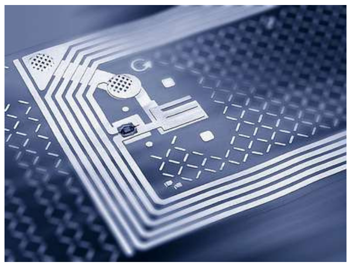
(1,180,591,620,717,411,303,424 BYTES; 2^{70})

approx. 1,000,000,000,000,000,000,000 or 10^{21}

Audio Streams



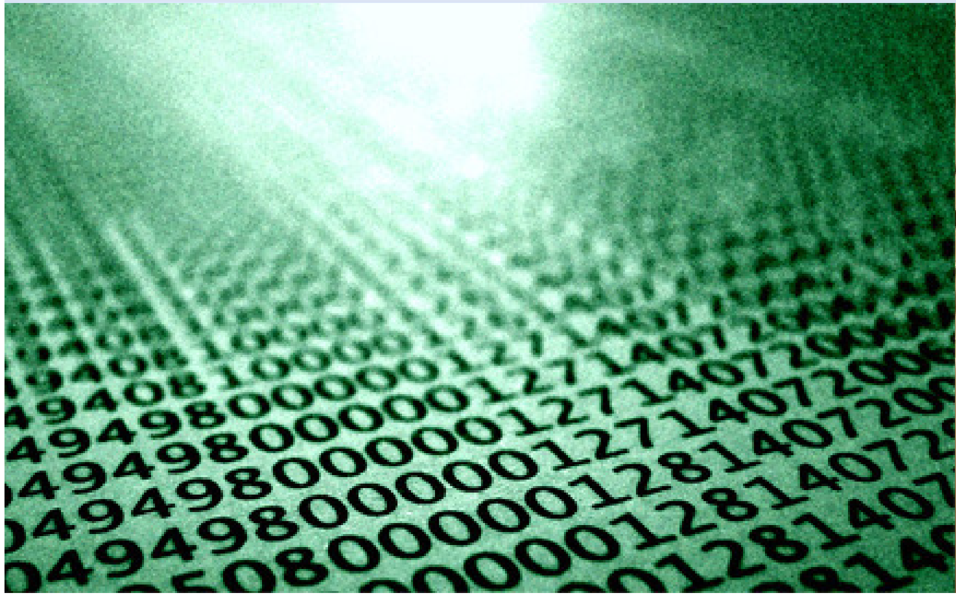
Genome Data



RFID Data

mostly machine generated or collaborative on web-scale

Defining Big Data Analytics by Complexity



Size
Format/Media Type
Uncertainty/Quality
Freshness
etc.

Data

Selection/Grouping
Relational Operators (Join)
Information Extraction & Integration
Data Mining
Predictive Models
etc.

Query

A Popular Definition: The „V"s

- Volume
  - Data size
- Velocity
  - Data ingestion speed
  - Analysis time window
- Variablity
  - Data formats
  - Media types
- Veracity
  - Uncertainty
  - Inconsistency

challenging for complex analysis questions

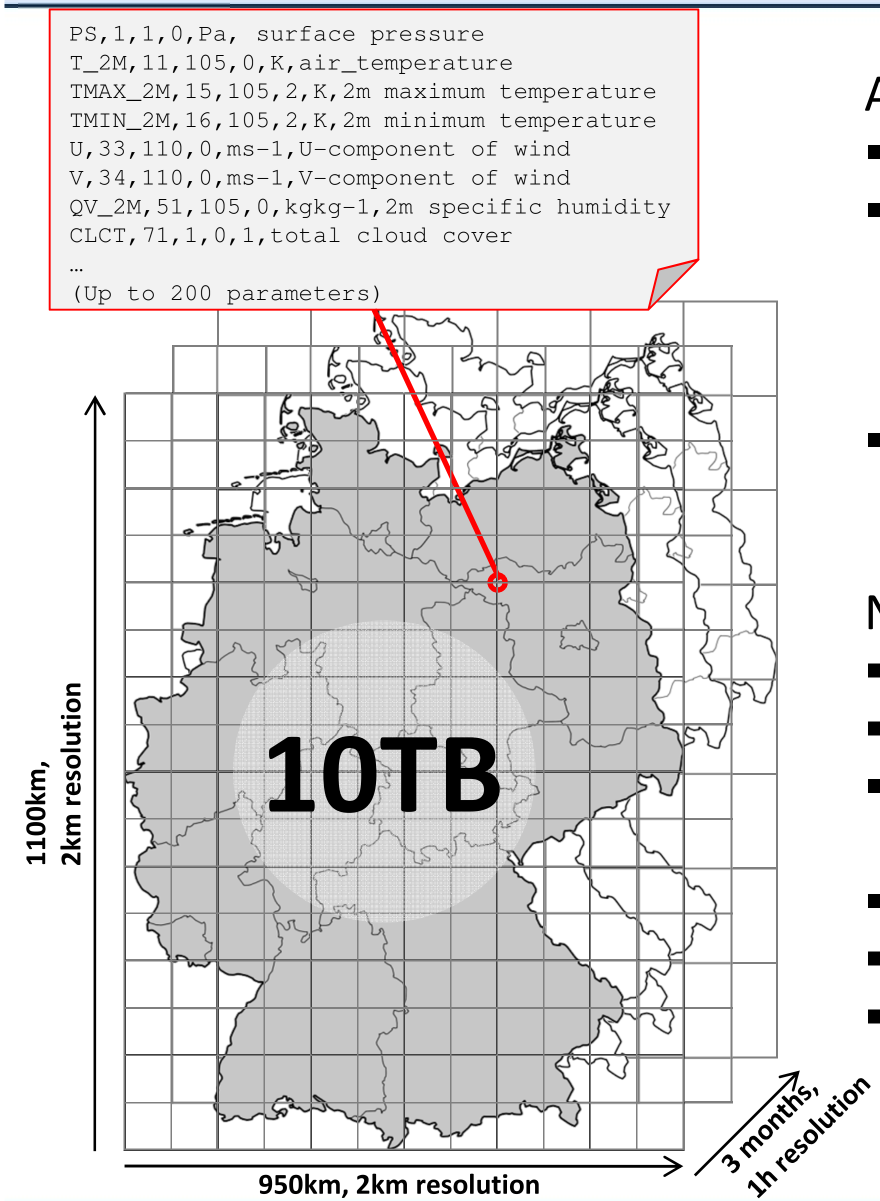
solutions exist for structured data and text

challenging for graph or audio/video data

challenging for automatic reasoning

Popular „definition", but misses some other aspects of complexity

Big Data Analytics Example: Climate Data Analysis



Analysis Tasks on Climate Data Sets

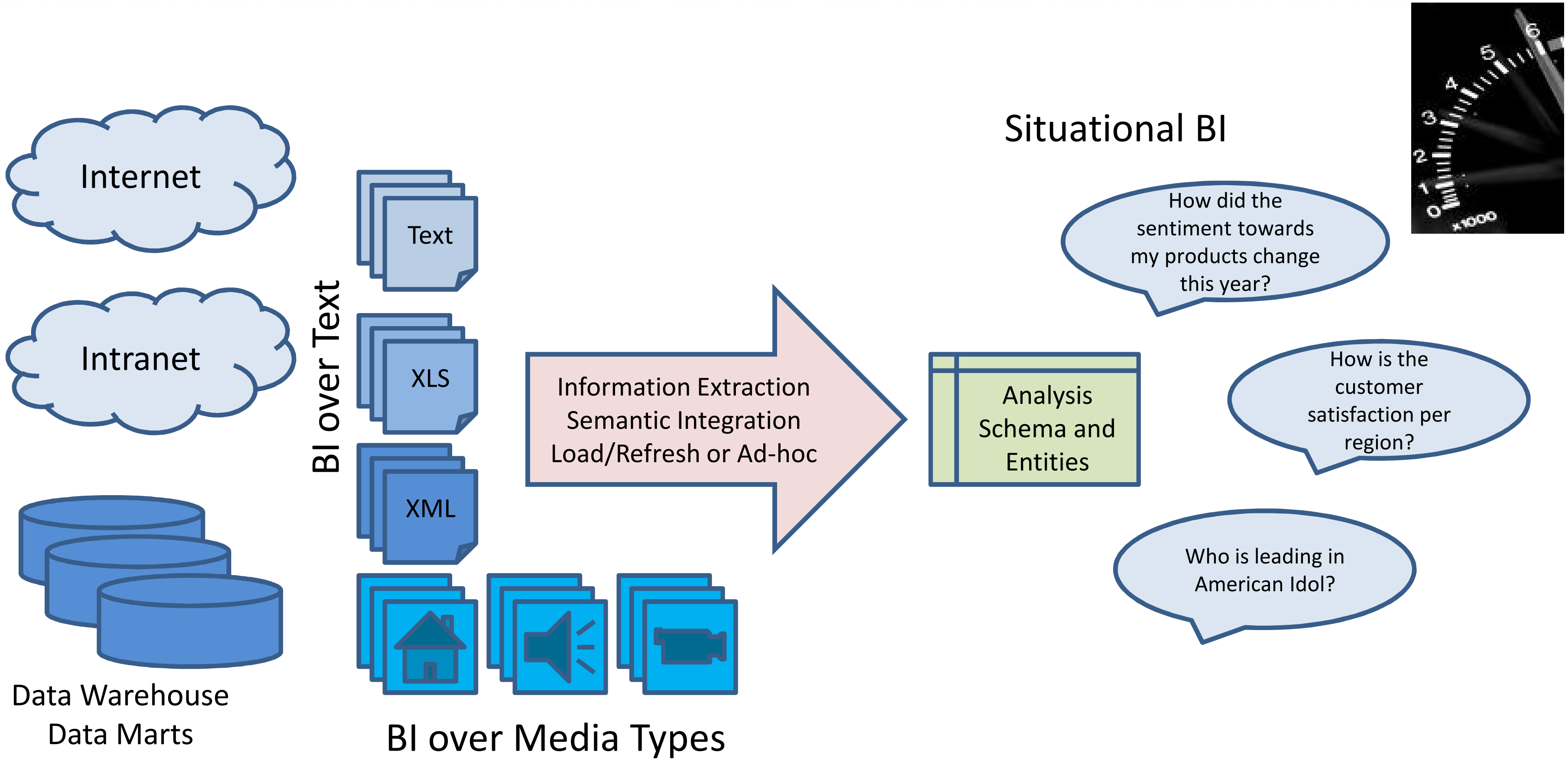
- Validate climate models
- Locate „hot-spots“ in climate models
  - Monsoon
  - Drought
  - Flooding
- Compare climate models
  - Based on different parameter settings

Necessary Data Processing Operations

- Filter
- Aggregation (sliding window)
- Join

- Multi-dimensional sliding-window operations
- Geospatial/Temporal joins
- Uncertainty

Another Example:
The Next Generation of Business Intelligence



The *next generation of Business Intelligence (NGBI)* will correlate data warehouses with text and other modalities from web services of information providers, corporate Intranets and the Internet

Further Applications

Healthcare

Home Automation

Water Management

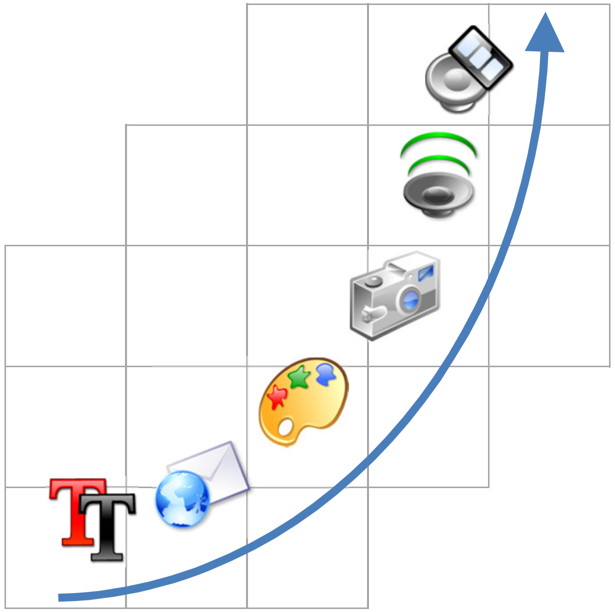
Lifecycle Management

Energy Management

Traffic Management

Sales/Marketing

- Need for data parallel processing
  - Increase of data complexity
  - Increase of query complexity
  - Moore’s Law: ManyCore and Cluster Computing
  - Scale-up no longer possible
  - ➔ **Scale-out is the name of the game**



- Parallel programming is not easy
  - (Network) Communication
  - Concurrent programming: Divide & Conquer
  - Synchronization as bottleneck (Amdahl’s Law)
  - Fault tolerance

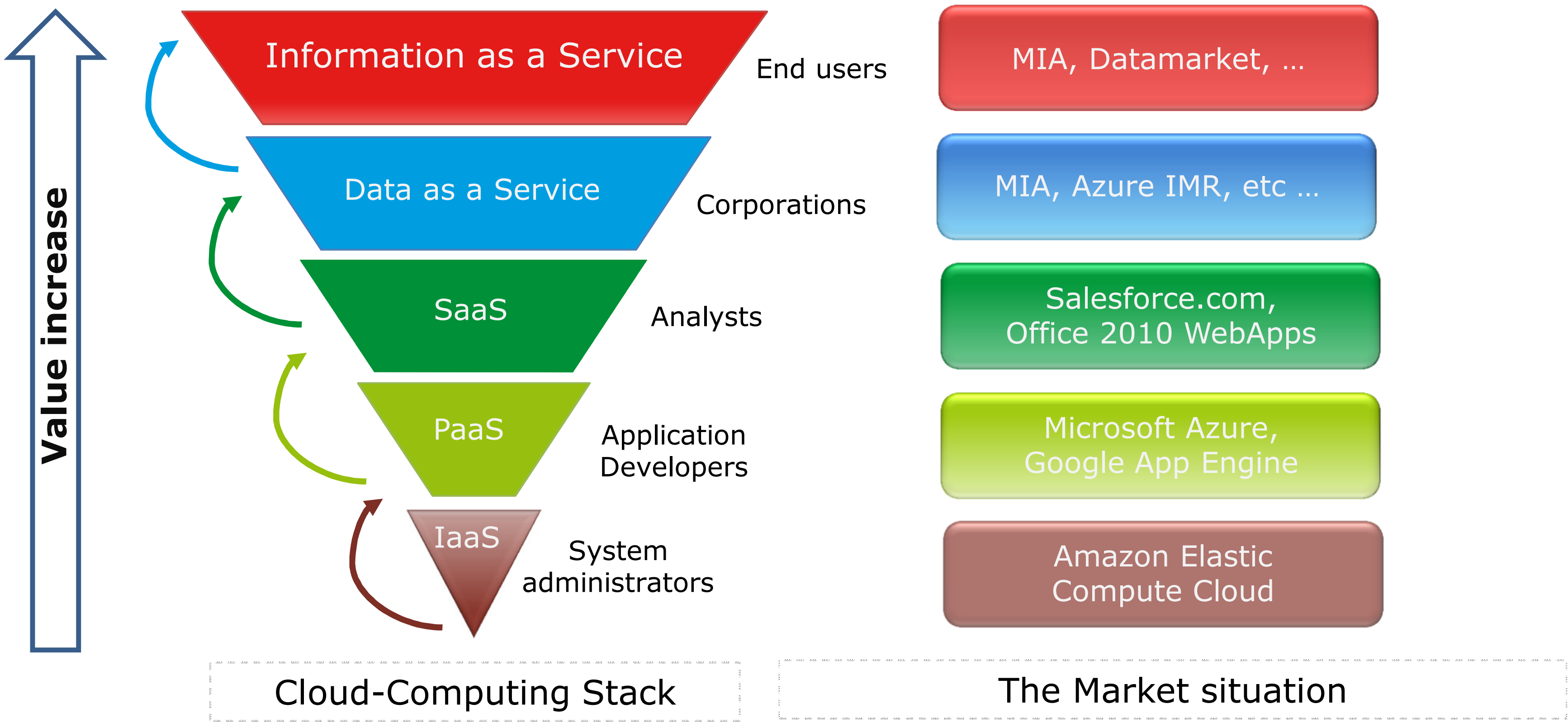
- Data Programming models must ease parallel data processing
  - Abstractions hide the gory details
  - Automatic adaption to hardware
    - Parallelization and Optimization
  - Beware: Data flow and control flow dependencies!
  - Popular simple model: map/reduce (e.g., Hadoop)

The Information Economy

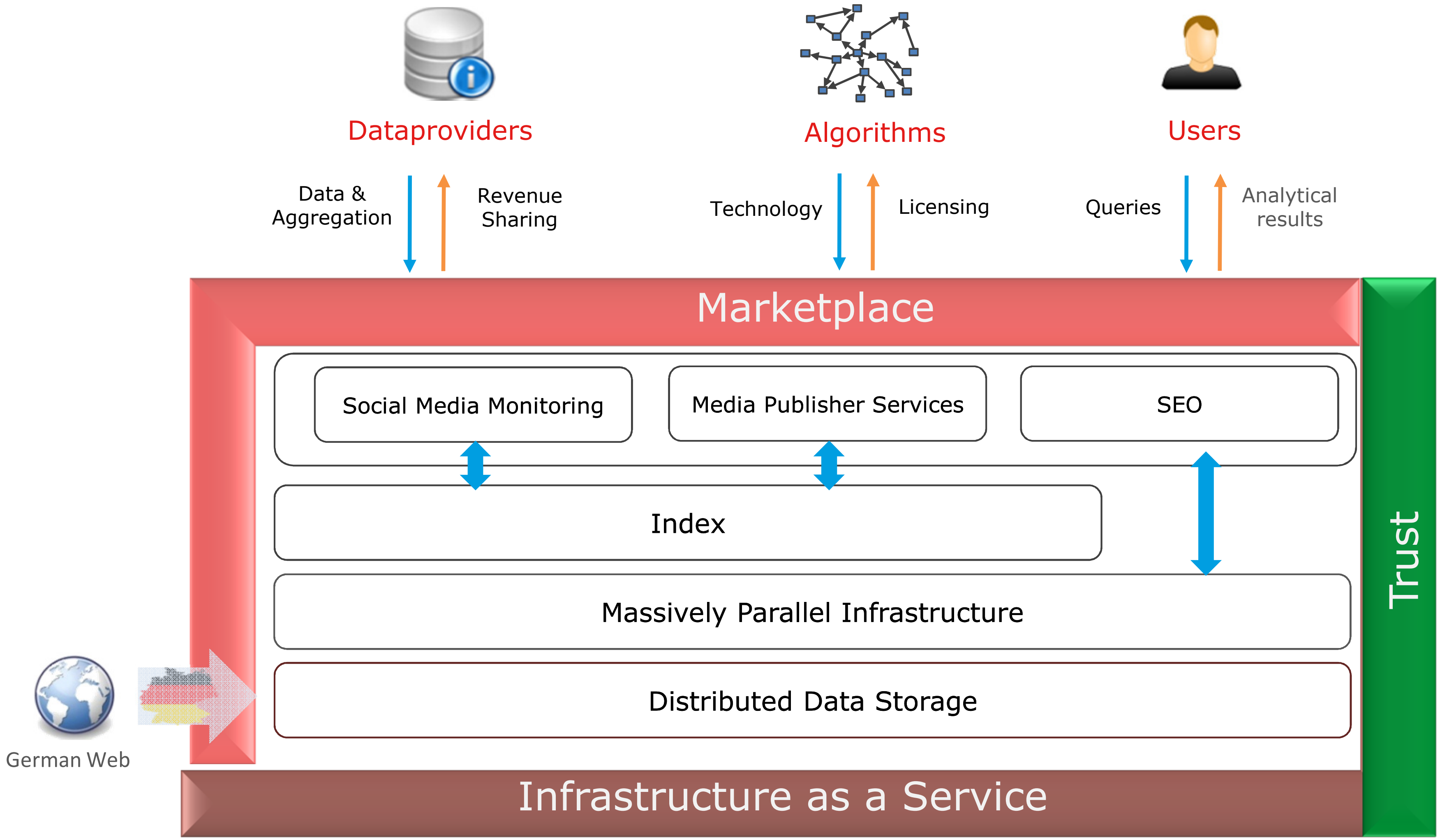
A major new trend in information processing will be the trading of original and enriched data, effectively creating an information economy.

„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)



MIA:
A Marketplace Infrastructure for Analytics



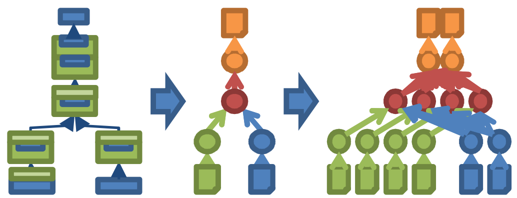
http://www.trusted-cloud.de/de/778.php

Stratosphere: Big Data Analytics Infrastructure for the Cloud

Use Cases

Scientific Data Life Sciences Linked Data

Parallel Analytics Query Processor

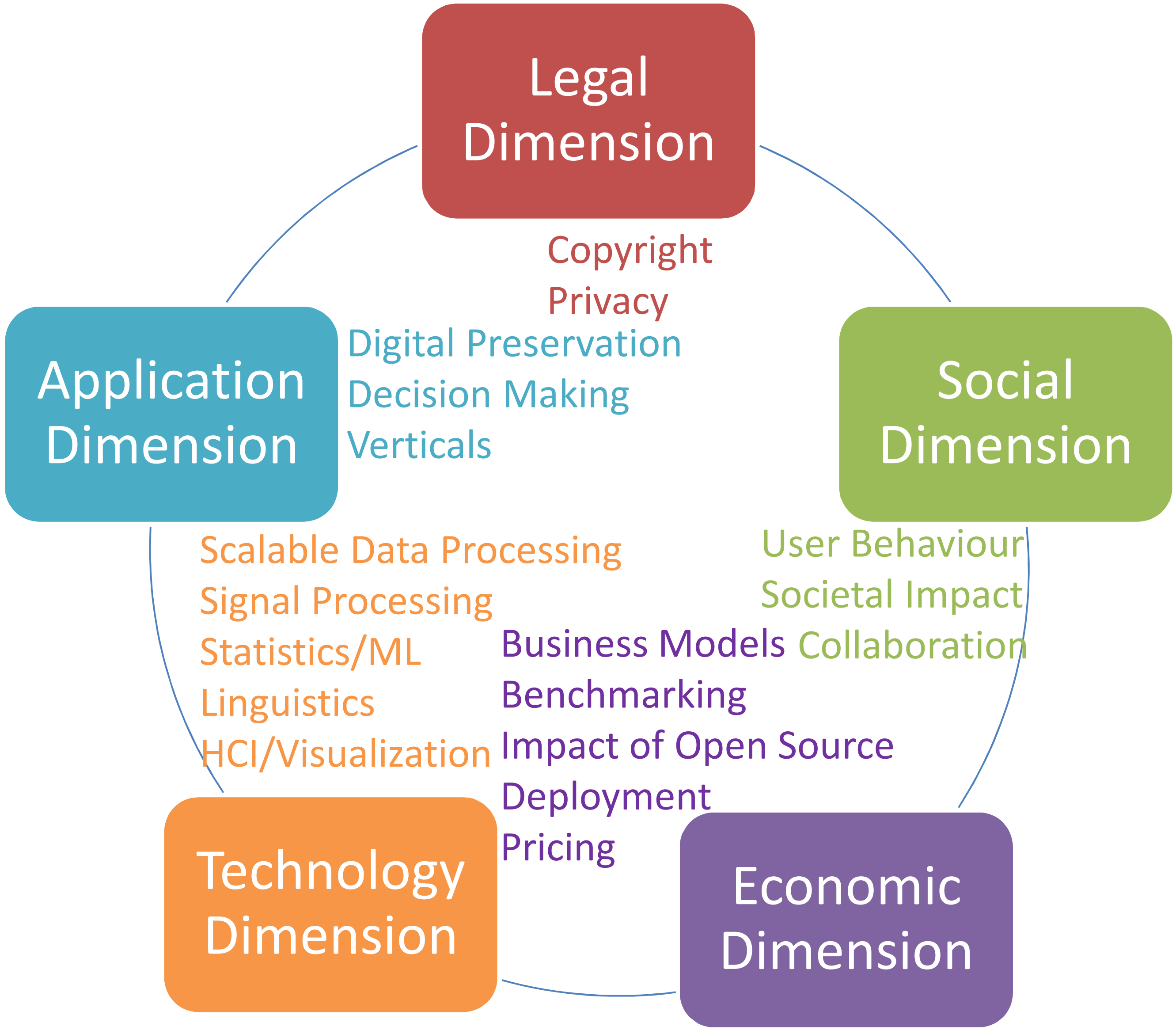


Infrastructure as a Service

- Explore the power of Cloud computing for complex analytical tasks
- Database-inspired approach
- Analyze, aggregate, and query
- Textual and (semi-) structured data
- Research and prototype a web-scale data analytics infrastructure

Stratosphere is publically funded by DFG and EIT. Open Source at **www.stratosphere.eu**

The „Big Data“ Space



Call to Action

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

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

We need to ensure a German technological leadership role in „Big Data"

Some Solutions

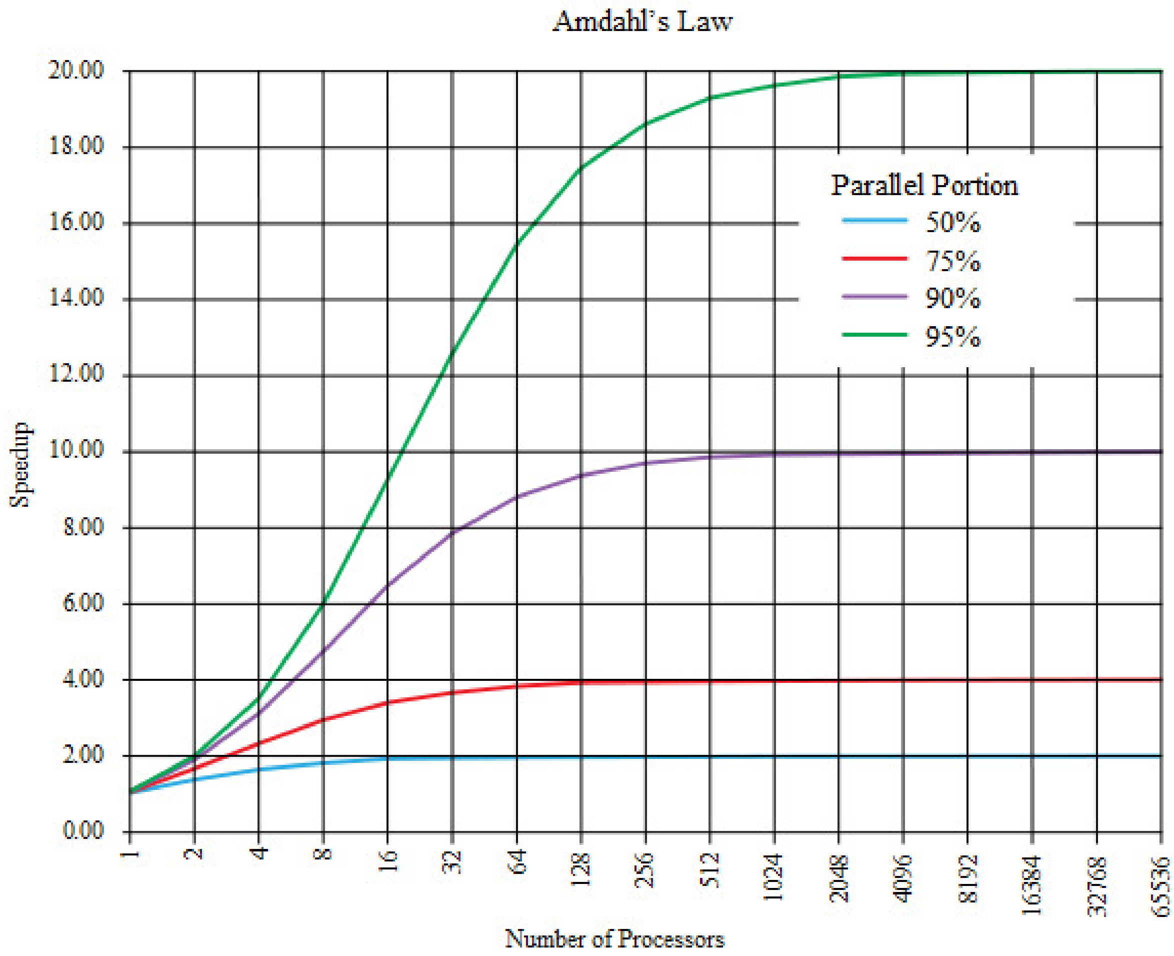
- Parallel Data Processing
- Parallel Data Managementb: ParStream et al.
- Map/Reduce: Hadoop, Stratosphere et al.

Parallel Speedup

- The Speedup is defined as: $S_p = {}^{T_1}/_{T_p}$
  - T_1: runtime of the sequential program
  - T_p: runtime of the parallel program on p processors

- Ahmdal's Law: „The maximal speedup is determined by the non-parallelizable part of a program" :
  - $S_{max} = \frac{1}{(1-f)+{}^{f}/_{p}}$ f: fraction of the program that can be parallelized
  - → Ideal speedup: S=p for f=1.0 (linear speedup)
  - However - since usually f<1.0 -, S is bound by a constant ! (e.g. ~10 for f=0.9)
  - → Fixed problems can only be parallelized to a certain degree!

Parallel Speedup



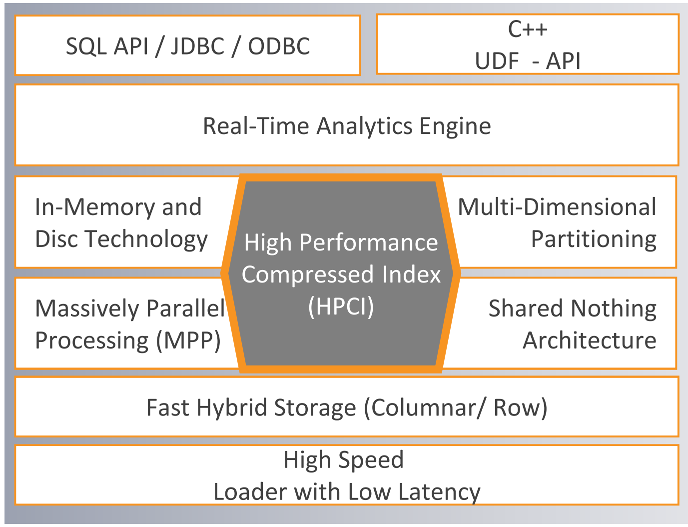
ParStream Big Data Analytics Platform

ParStream provides the unique combination of REAL-TIME – LOW-LATENCY – HIGH THROUGHPUT

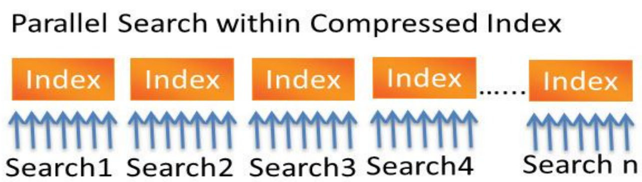
ParStream's key advantages:

- Unique Highly compressed Bitmap Index which can be analyzed in compressed form (patent application filed)
- Real-time analytics through Massively Parallel Processing (MPP)
- Very high throughput due to massively reduced workload (no decompression, small index, efficient algorithms)
- Low-Latency through continuous import of new data without slowing down analytics
- Columnar data / index allows very flexible analytics (multi-column, multi-value)
- Specialized data / index types and algorithms
- Shared Nothing Architecture

PARSTREAM PARALLEL ARCHITECTURE



PARSTREAM INDEX ARCHITECTURE



HUGE MARKET OPPORTUNITY

Big Data Analytics is a game changer in every industry and is a huge market opportunity

All Industries

Many Applications

eCommerce Services
- Facetted Search
- Web analytics
- SEO-analytics
- Online-Advertising

Social Networks
- Ad serving
- Profiling
- Targeting

Telco
- Customer attrition prevention
- Network monitoring
- Targeting
- Prepaid account mgmt

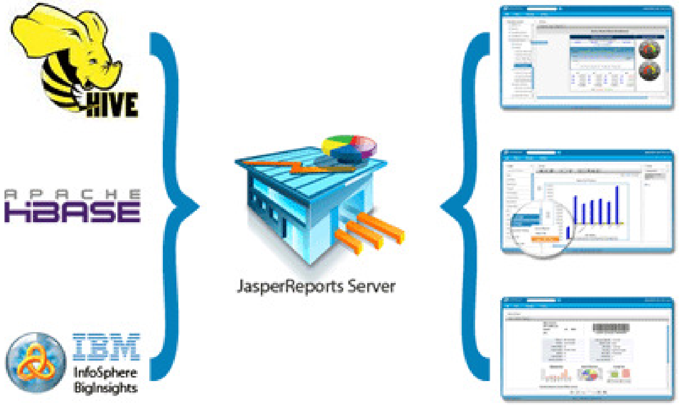
Finance
- Trend analysis
- Fraud detection
- Automatic trading
- Risk analysis

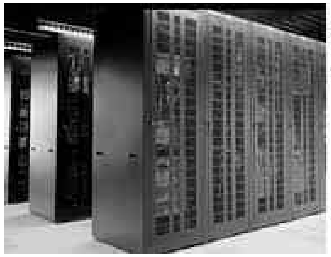
Energy Oil and Gas
- Smart metering
- Smart grids
- Wind parks
- Mining
- Solar Panels

Many More
- Production
- Mining
- M2M
- Sensors
- Genetics
- Intelligence
- Weather

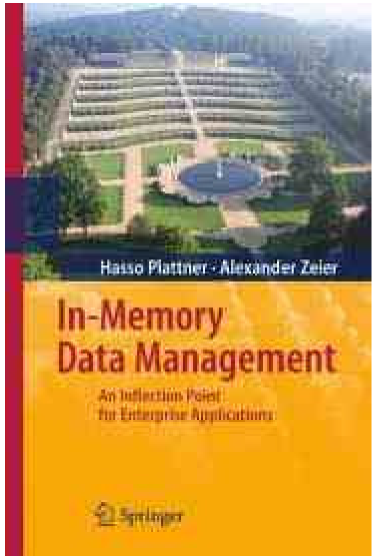
22

Further Players





Sybase IQ™



sas ANALYTICS

SPSS
STATISTICAL DATA ANALYSIS

aster data
big data. fast insights.

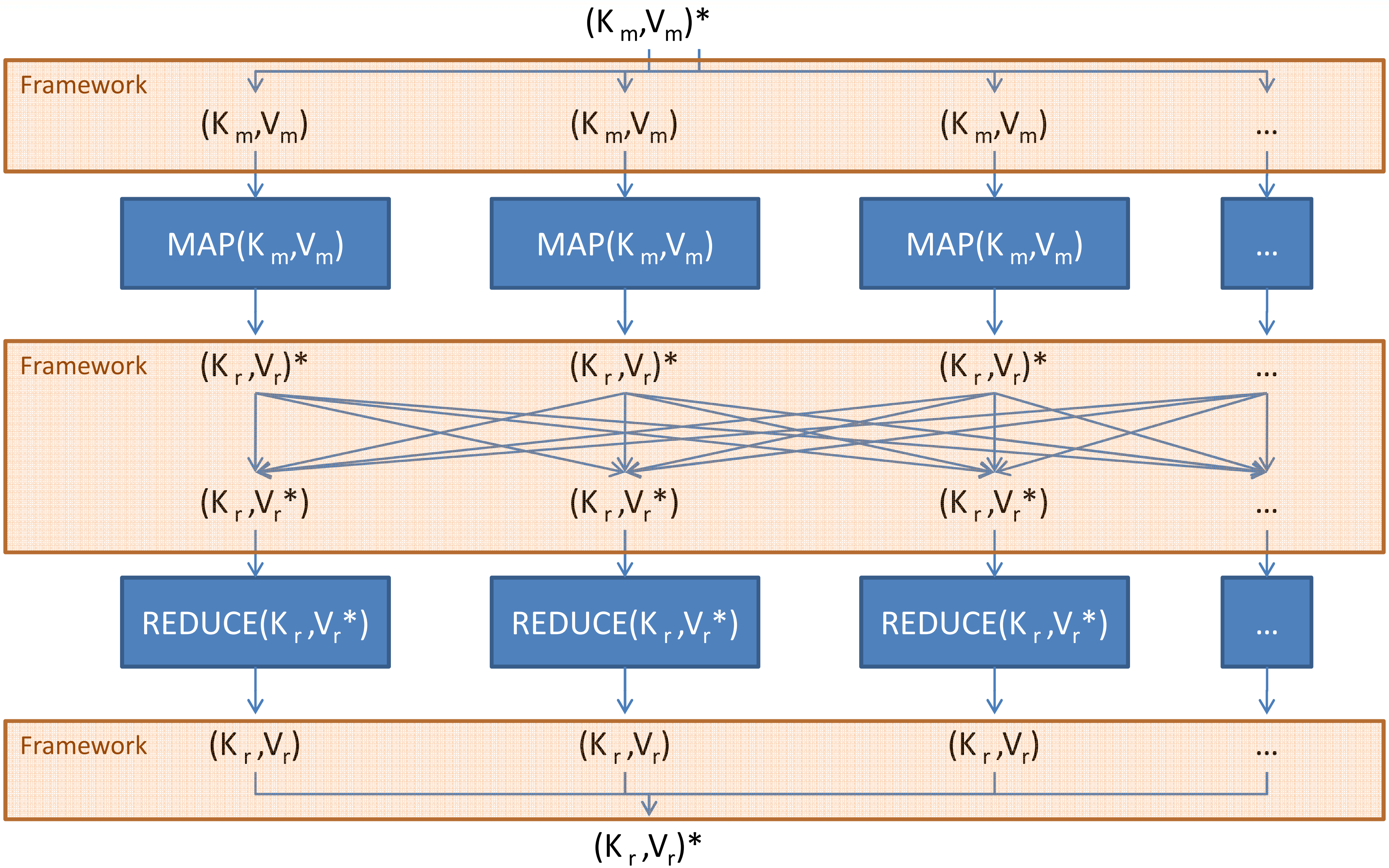
What is Map/Reduce?

- Programming model for data-intensive programming
  - well-suited for large scale parallel execution
  - automatic parallelization & distribution of data and computational logic
  - easy to extend with fault tolerance schemes
  - clean abstraction for programmers

- Based on functional programming
  - treats computation as the evaluation of mathematical functions and avoids state and mutable data
  - no changes of states (no side effects)
  - output value of a function depends only on its arguments

- `Map` and `Reduce` are higher-order functions
  - take user-defined functions as argument
  - return a function as result
  - to define a map/reduce job, the user implements the two functions m and r

Data Flow in Map/Reduce



Map Reduce Illustrated (1)

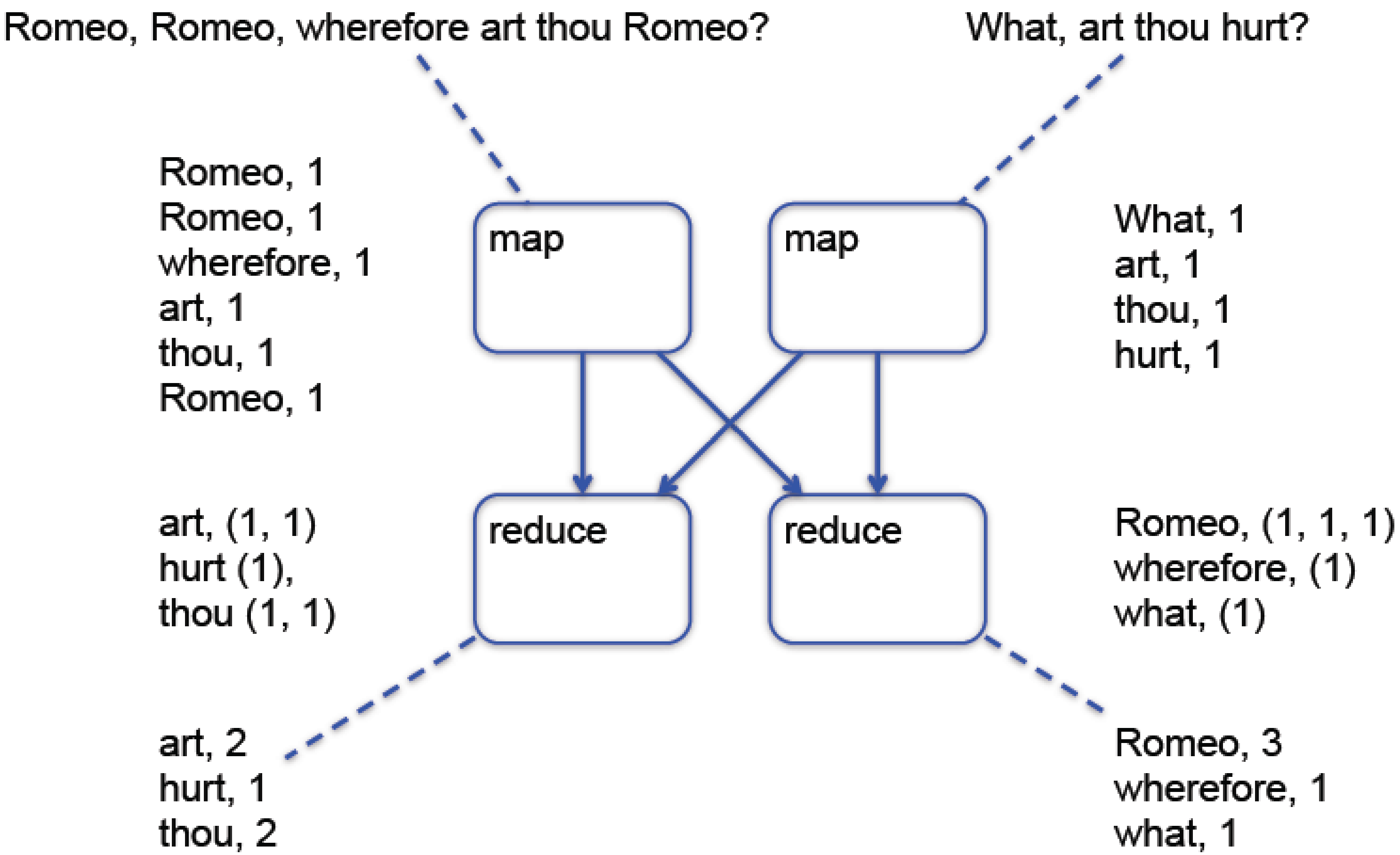
- *Problem*: Counting words in a parallel fashion
  - How many times different words appear in a set of files
  - **juliet.txt**: Romeo, Romeo, wherefore art thou Romeo?
  - **benvolio.txt:** What, art thou hurt?
  - Expected output: Romeo (3), art (2), thou (2), art (2), hurt (1), wherefore (1), what (1)

- *Solution*: Map-Reduce Job

```
m (filename, line) {
  foreach (word in line)
     emit(word, 1);
 }

r (word, numbers) {
  int sum = 0;
  foreach (value in numbers) {
    sum += value;
  }
  emit(word, sum);
}
```

Map Reduce Illustrated (2)



Relational Operators as Map/Reduce jobs

- Selection / projection / aggregation
  - SQL Query:

```
SELECT year, SUM(price)
FROM sales
WHERE area_code = “US”
GROUP BY year
```

  - Map/Reduce job:

```
map(key, tuple) {
  int year = YEAR(tuple.date);
  if (tuple.area_code = “US”)
    emit(year, {‘year’ => year, ‘price’ => tuple.price });
}

reduce(key, tuples) {
  double sum_price = 0;
  foreach (tuple in tuples) {
    sum_price += tuple.price;
  }
  emit(key, sum_price);
}
```

- Sorting
  - SQL Query:

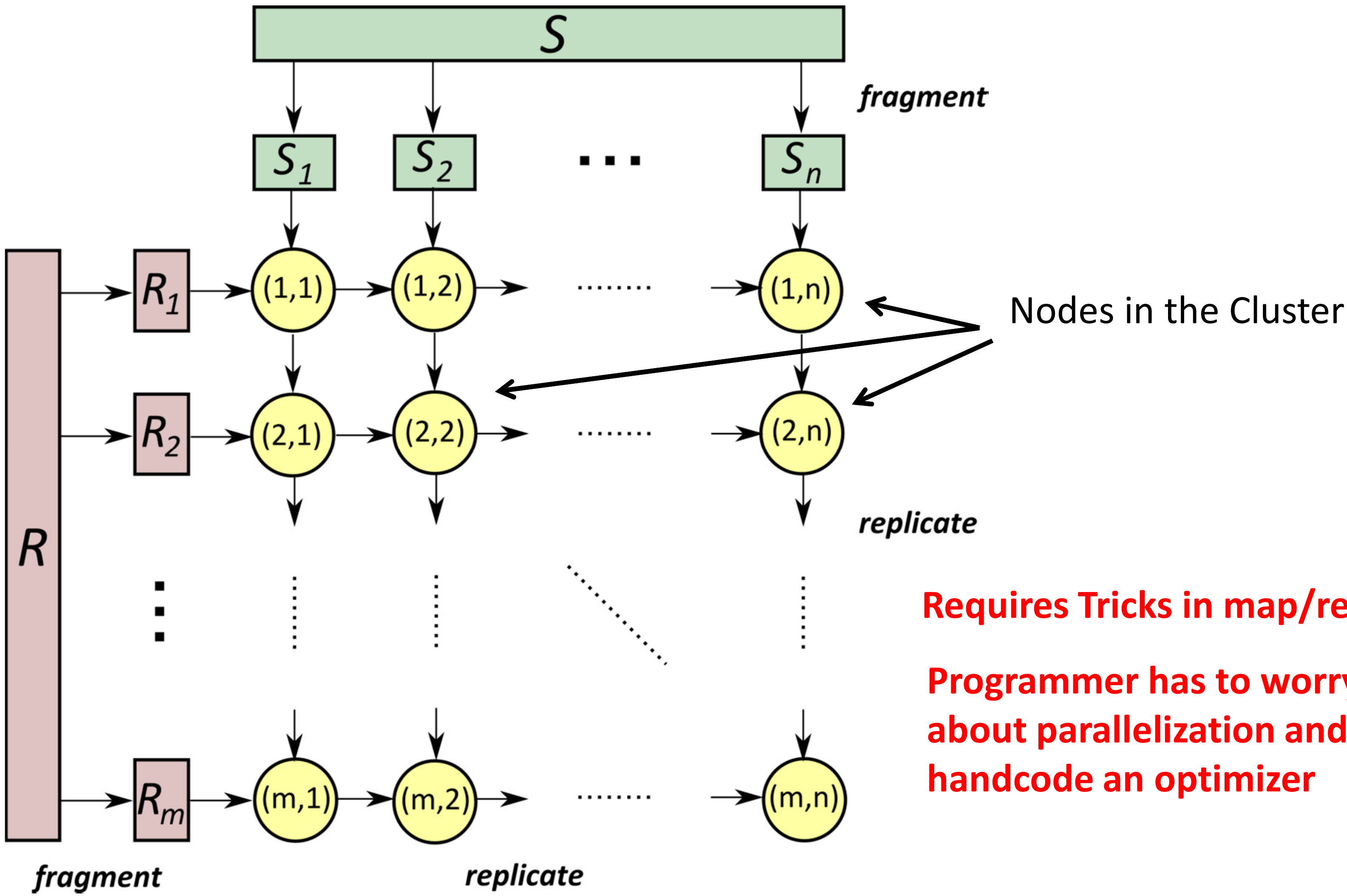
```
SELECT *
FROM sales
ORDER BY year
```

  - Map/Reduce job:

```
map(key, tuple) {
  emit(YEAR(tuple.date) DIV 10, tuple);
}

reduce(key, tuples) {
  emit(key, sort(tuples));
}
```

Symmetric Fragment-and-Replicate Join



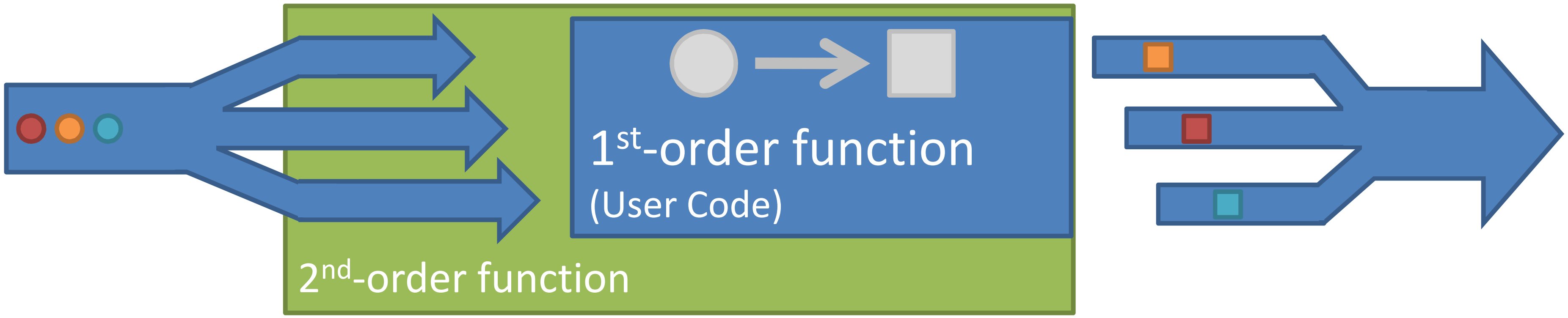
Requires Tricks in map/reduce

Programmer has to worry about parallelization and/or handcode an optimizer

Important generic example. Specialized parallel joins may exploit data locality

- PACT is a generalization and extension of MapReduce
  - PACT inherits many concepts from MapReduce

- Both are inspired by functional programming
  - Fundamental concept of programming model are 2nd-order functions
  - User writes 1st-order functions (user functions)
  - User code can be arbitrarily complex (albeit functional!)
  - 2nd-order function calls 1st-order function with independent data subsets
  - No common state should be held between calls of user function

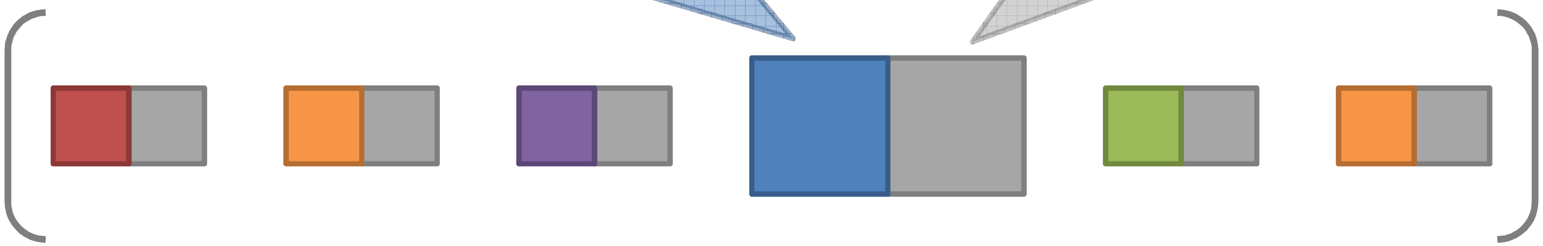


Common Concepts of MapReduce and PACT

- Both use a common data format
  - Data is processed as pairs of keys and values
  - Keys and values can be arbitrary data structures

<u>Key:</u>
- Used to build independent subsets
- Must be comparable and hashable
- Does not need to be unique
  - no Primary Key semantic!
- Interpreted only by user code

<u>Value:</u>
- Holds application data
- Interpreted only by user code

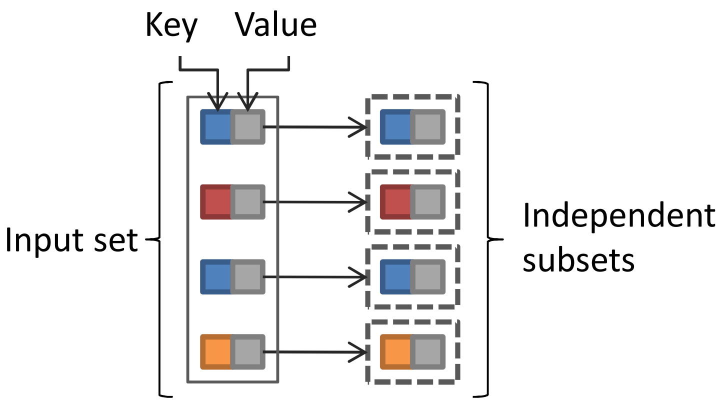


MapReduce Programming Model Revisited

- MapReduce provides two 2nd-order functions

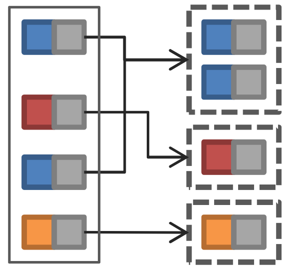
Map:

- All pairs are independently processed

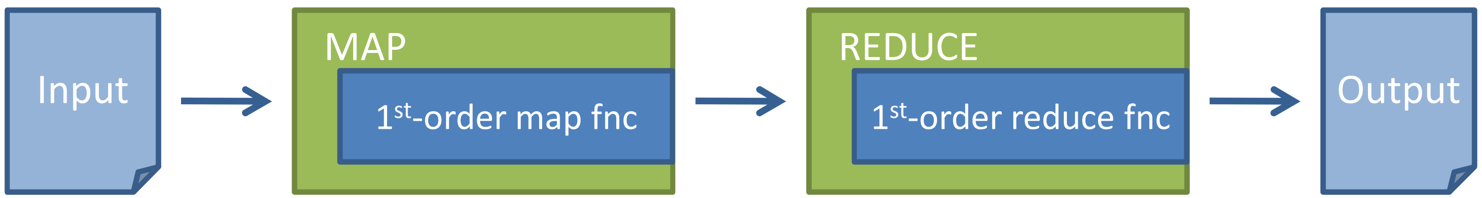


Reduce:

- Pairs with identical key are grouped
- Groups are independently processed

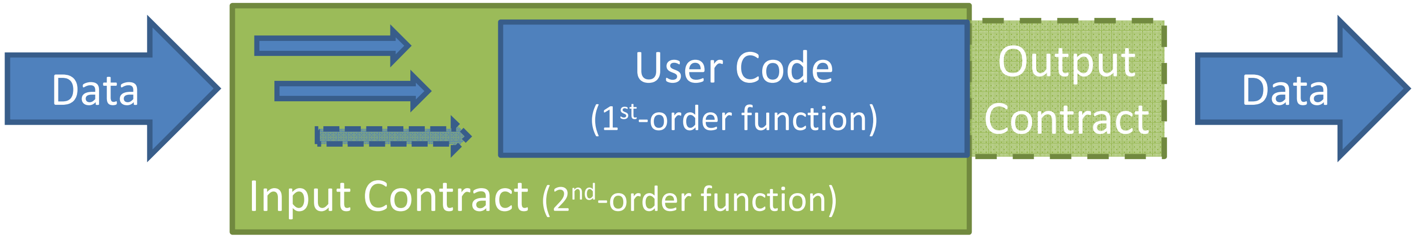


- MapReduce programs has fixed structure:



PACT Programming Model

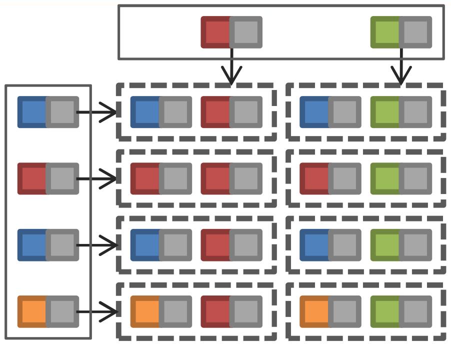
- Generalization and Extension of MapReduce Programming Model
- Based on Parallelization Contracts (PACTs)



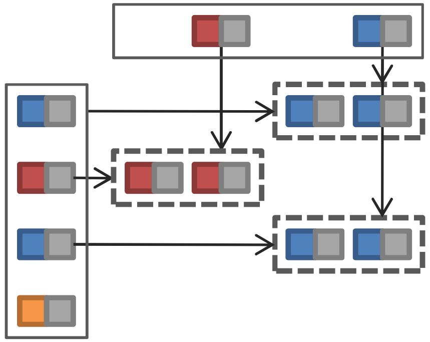
- Input Contract
  - 2nd-order function; generalization of Map and Reduce
  - Generates independently processable subsets of data
- User Code
  - 1st-order function
  - For each subset independently called
- Output Contract
  - Describes properties of the output of the 1st-order function
  - Optional but enables certain optimizations
  - Think: "interesting properties"

Input Contracts beyond Map and Reduce

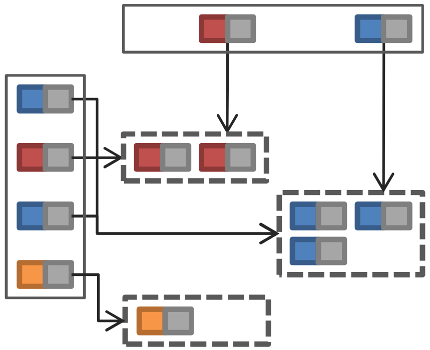
- Cross
  - Builds a Cartesian Product
  - Elements of CP are independently processed



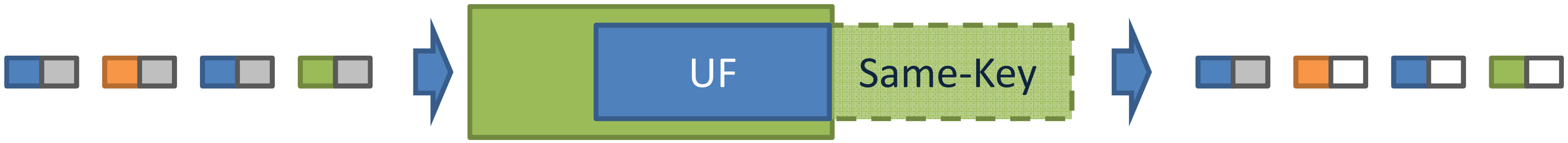
- Match
  - Performs an equi-join on the key
  - Join candidates are independently processed



- CoGroup
  - Groups each input on key
  - Groups with identical keys are processed together



- Same-Key
  - User Function does not alter the key



- Super-Key
  - Key generated by UF is a super-key of the input key

- Unique-Key
  - Data source or UF produces unique keys

PACT Programming Model

- PACT Programs effectively are data flow graphs
  - Data comes from sources and flows to sinks
  - PACTs process data in-between sources and sinks
  - Multiple sources and sinks allowed
  - Arbitrary complex directed acyclic data flows can be composed