

Big Data: Challenges and Some Solutions

Stratosphere, Apache Flink, and Beyond

Volker Markl

<http://www.user.tu-berlin.de/marklv/>

<http://www.dima.tu-berlin.de>

<http://www.dfki.de/web/forschung/iam>

<http://bbdc.berlin>

volker.markl@tu-berlin.de

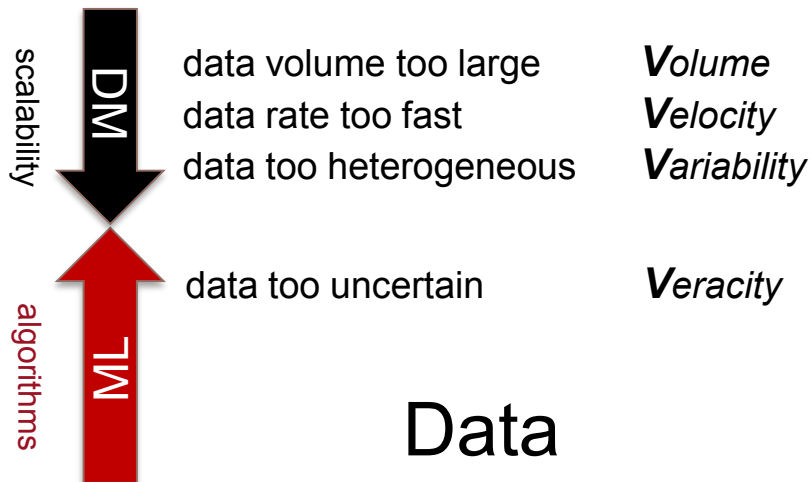


Thanks to my team members and students

- Dr. Stephan Ewen
- Sebastian Schelter
- Dr. Kostas Tzoumas
- Dr. Asterios Katsifodimos
- Fabian Hüske
- Alexander Alexandrov
- Max Heimel

and many more members of the Stratosphere Project, the Berlin Big Data Center, and the Apache Flink community

Data & Analysis: Increasingly Complex!



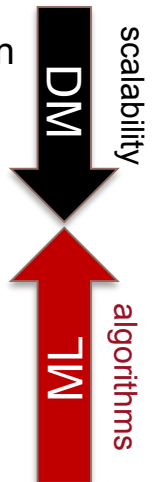
Reporting
Ad-Hoc Queries
ETL/ELT

aggregation, selection
SQL, XQuery
MapReduce

Data Mining
Predictive/Prescriptive

MATLAB, R, Python
MATLAB, R, Python

Analysis



“Data Scientist” – “Jack of All Trades!”

Domain Expertise (e.g., Industry 4.0, Medicine, Physics, Engineering, Energy, Logistics)

Mathematical Programming

Linear Algebra

Stochastic Gradient Descent

Error Estimation

Active Sampling

Regression

Monte Carlo

Statistics

Sketches

Hashing

Convergence

Decoupling

Iterative Algorithms

Curse of Dimensionality

Relational Algebra / SQL

Data Warehouse/OLAP

NF²/XQuery

Resource Management

Hardware Adaptation

Fault Tolerance

Memory Management

Parallelization

Scalability

Memory Hierarchy

Data Analysis Language

Compiler

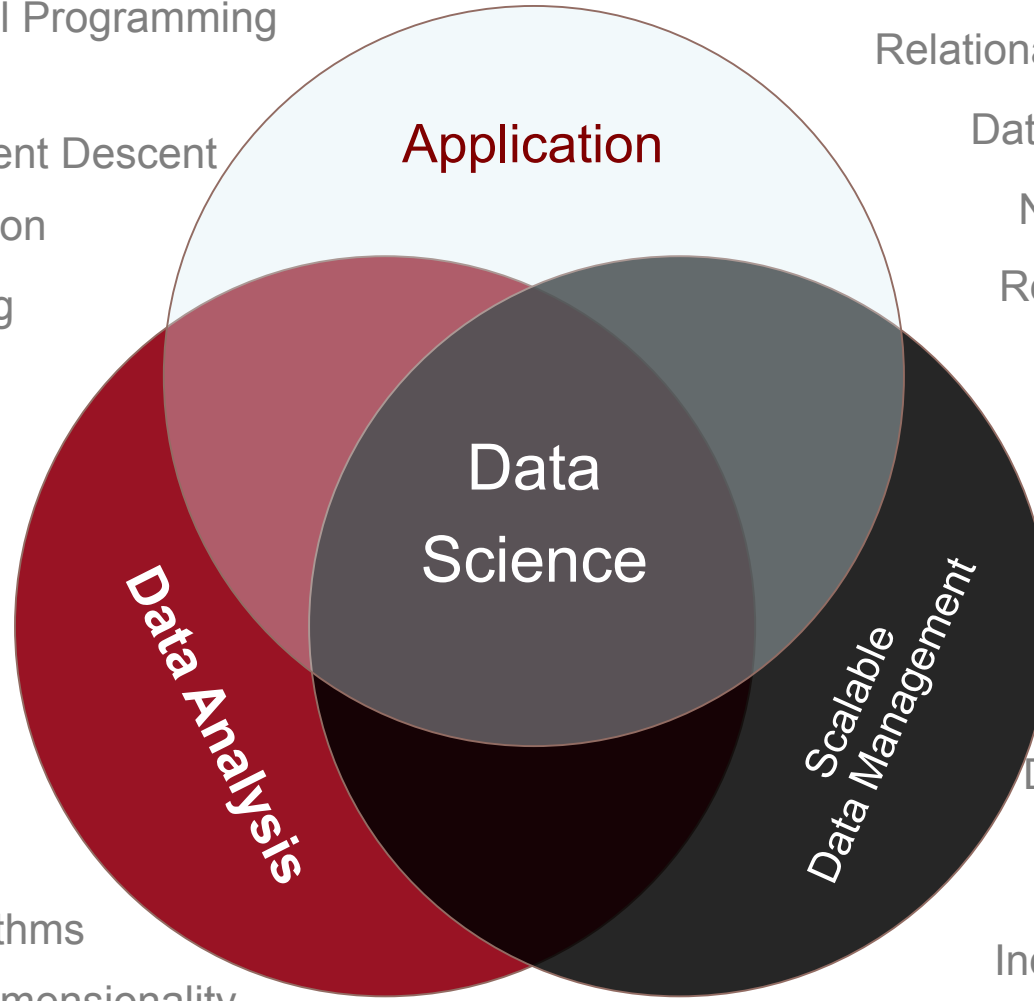
Query Optimization

Indexing

Data Flow

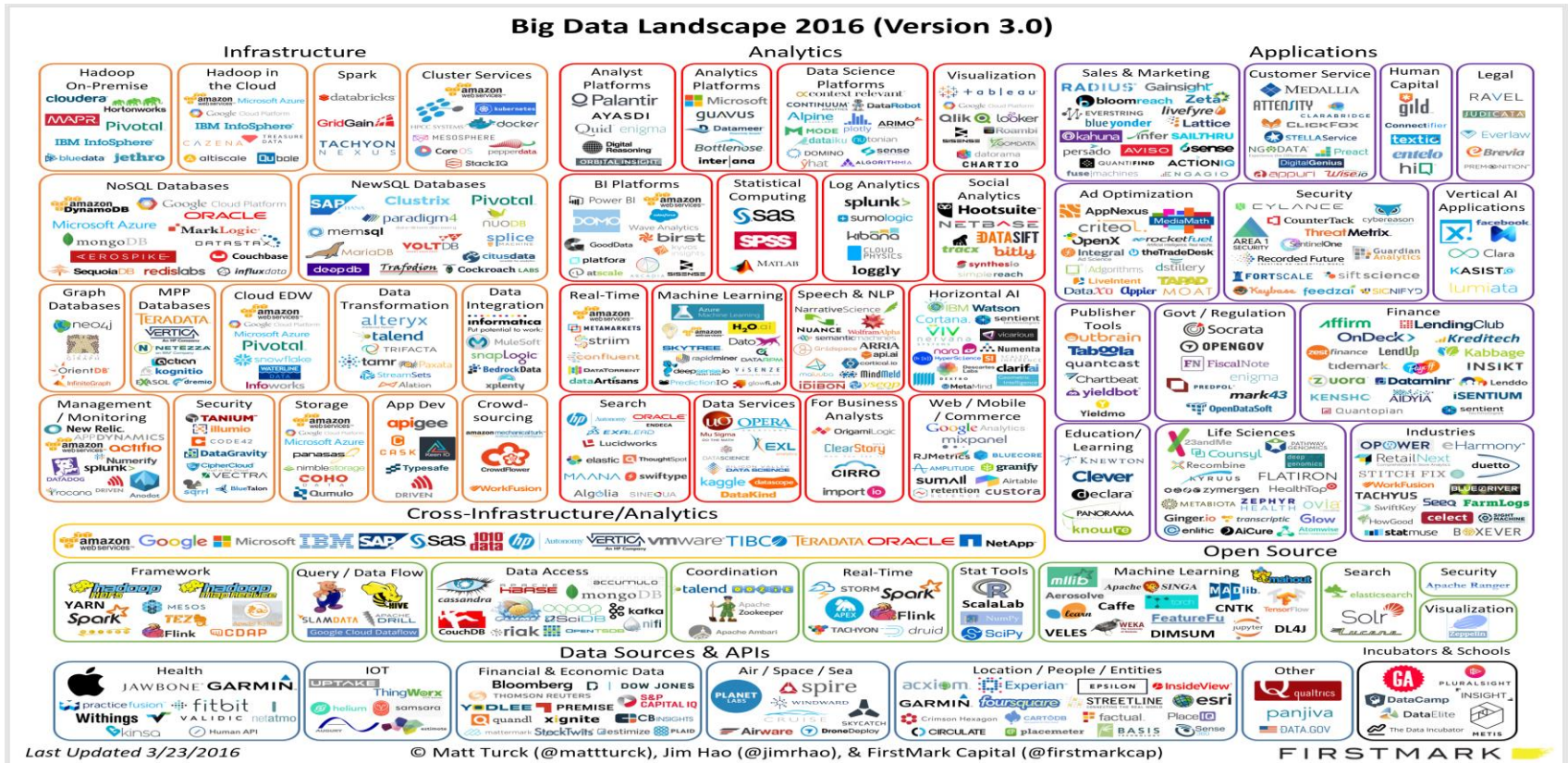
Control Flow

Real-Time



New Technology to the Rescue!

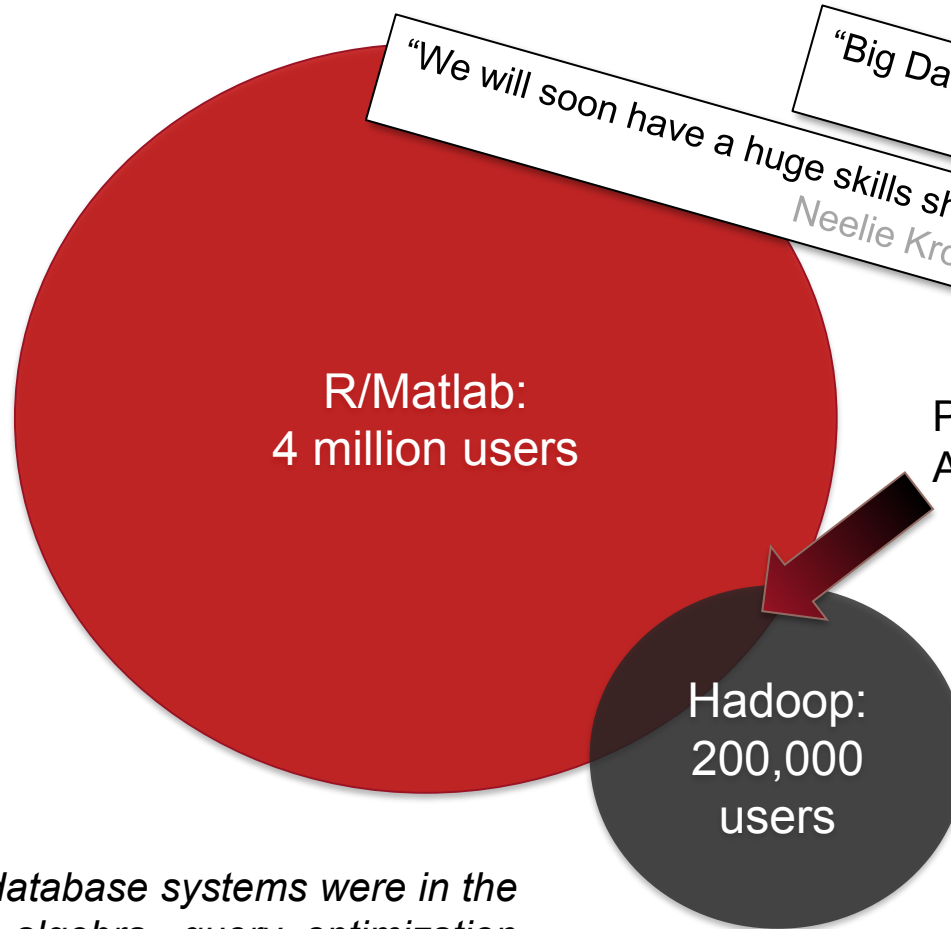
A Zoo of Technologies!



<http://mattturck.com/wp-content/uploads/2016/03/Big-Data-Landscape-2016-v18-FINAL.png>

Big Data Analytics Requires Systems Programming

Data Analysis
Statistics
Algebra
Optimization
Machine Learning
NLP
Signal Processing
Image Analysis
Audio-, Video Analysis
Information Integration
Information Extraction
Data Value Chain
Data Analysis Process
Predictive Analytics



“We will soon have a huge skills shortage for data-related jobs.”
Neelie Kroes (ICT 2013, Nov. 7, Vilnius)

“Big Data’s Big Problem: Little Talent”
Wall Street Journal

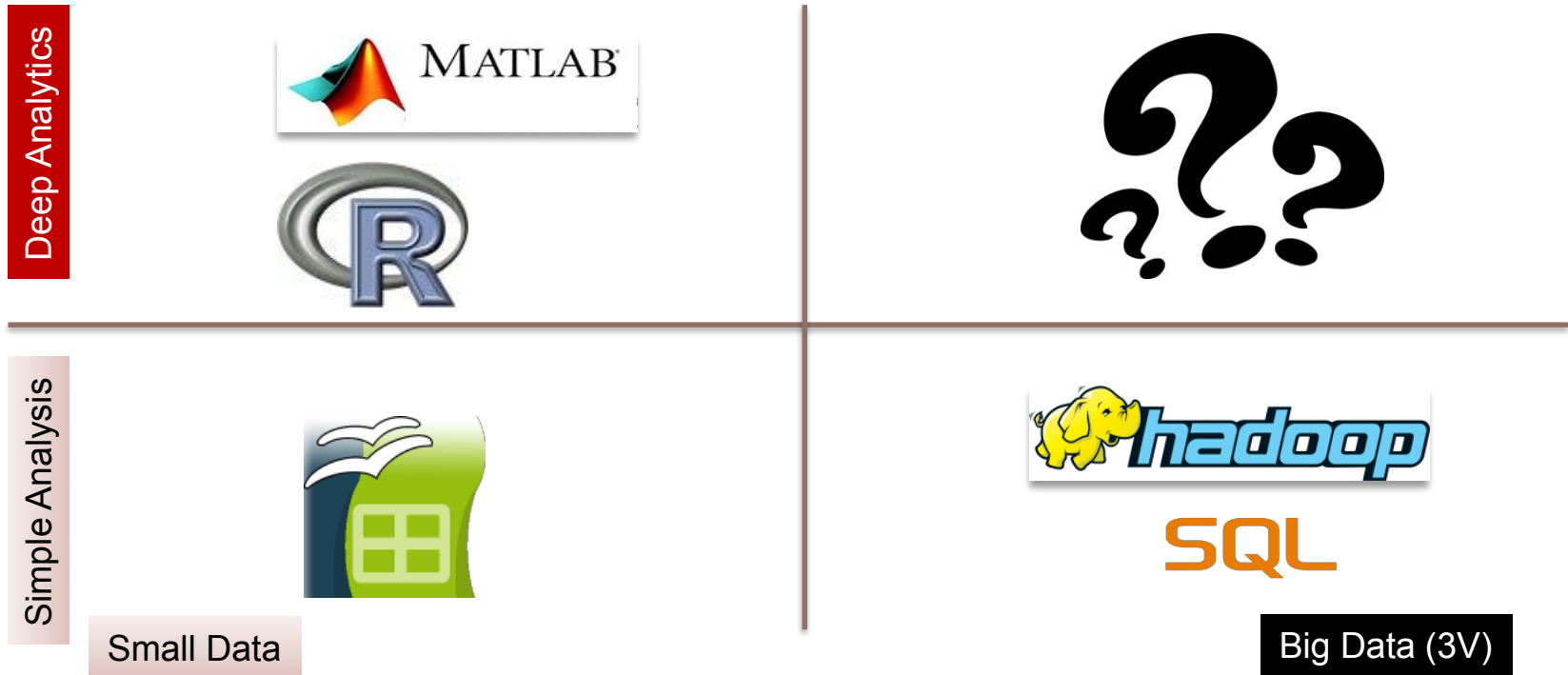
People with Big Data Analytics Skills

- Indexing
- Parallelization
- Communication
- Memory Management
- Query Optimization
- Efficient Algorithms
- Resource Management
- Fault Tolerance
- Numerical Stability

Big Data is now where database systems were in the 70s (prior to relational algebra, query optimization and a SQL-standard)!

Declarative languages to the rescue!

Deep Analysis of „Big Data“ is Key!



Many new companies and products are emerging to enable deep big data analysis; **strong European contenders** include Apache Flink, Parstream, and Exasol.

„New companies“ are the **(b)leading users of these technologies**, e.g., in the information economy (e.g., Zalando, Amazon, Researchgate, Soundcloud, Spotify).

„Traditional Big companies“ are **following** and still determining strategies (Industrie 4.0, Logistics, Telco, etc.). Most **SMEs are not ready yet to capitalize on Big Data**.

Machine Learning + Data Management = X



Technology X

Relational Algebra/SQL
Data Warehouse/OLAP
NF²/XQuery Scalability
Hardware adaption
Fault Tolerance
Resource Management

ML

DM

*Think ML-algorithms
in a scalable way*

declarative

*Process iterative
algorithms
in a scalable way*

Declarative Languages
Automatic Adaption
Scalable processing

**Goal: Data Analysis without
System Programming!**

Parallelization Compiler
Memory Management
Memory Hierarchy
Data Analysis Language
Query Optimization
Dataflow Indexing

Mathematical Programming
Linear Algebra
Error Estimation
Active Sampling
Regression Monte Carlo

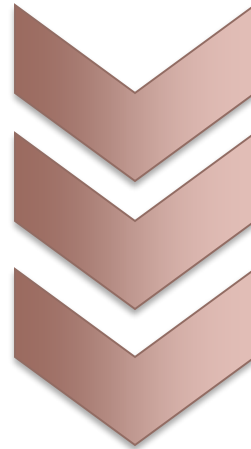
Feature Engineering
Representation
Algorithms (SVM, GPs, etc.)

Statistic
Sketches Hashing
Isolation Convergence
Curse of Dimensionality
Iterative Algorithms
Control flow

Agenda

- On Stratosphere and Flink
 - Conception, Initial Contributions
 - Streaming Data Analysis
 - The Flink Community
- On Roman Generals and Big Data Analytics
- On Emma and Mosaics

June 2, 2008



Aug 26, 2014



Apache Flink

Stratosphere: General Purpose Programming + Database Execution

Draws on
Database Technology

- Relational Algebra
- Declarativity
- Query Optimization
- Robust Out-of-core

Adds

- Iterations
- Advanced Dataflows
- General APIs
- Native Streaming

Draws on
MapReduce Technology

- Scalability
- User-defined Functions
- Complex Data Types
- Schema on Read

A. Alexandrov, D. Battré, S. Ewen, M. Heibel, F. Hueske, O. Kao, V. Markl, E. Nijkamp, D. Warneke: Massively Parallel Data Analysis with PACTs on Nephele. PVLDB 3(2): 1625-1628 (2010)

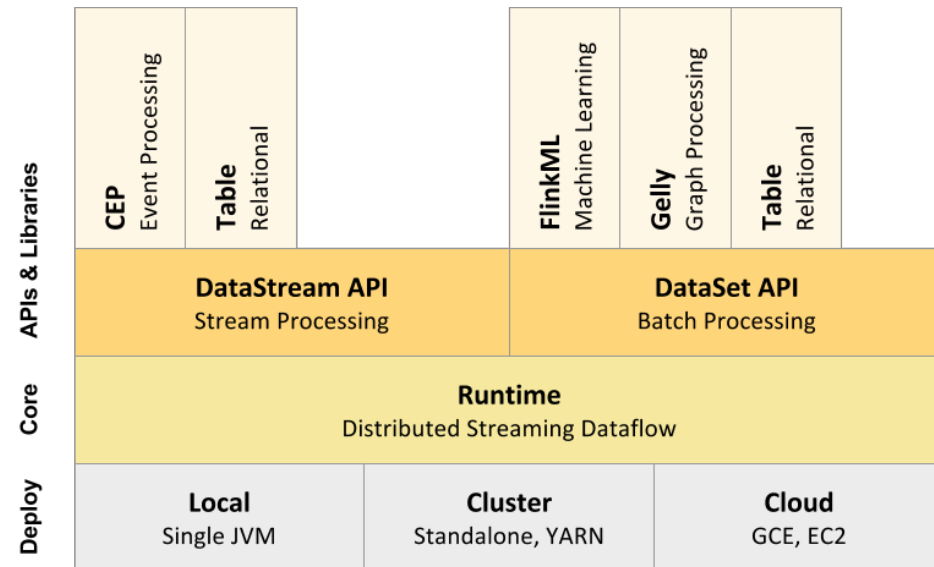
D. Battré, S. Ewen, F. Hueske, O. Kao, V. Markl, D. Warneke: Nephele/PACTs: a programming model and execution framework for web-scale analytical processing. SoCC 2010: 119-130

A. Alexandrov, R. Bergmann, S. Ewen, et al: The Stratosphere platform for big data analytics. VLDB J. 23(6): 939-964 (2014)

What is Apache Flink?

Apache Flink® is an open-source stream processing framework for distributed, high-performing, always-available, and accurate data streaming applications.

- Key Features:
 - Bounded and unbounded data
 - Event time semantics
 - Stateful and fault-tolerant
 - Running on thousands of nodes with very good throughput and latency
 - Exactly-once semantics for stateful computations.
 - Flexible windowing based on time, count, or sessions in addition to data-driven windows
- **DataSet** and **DataStream** programming abstractions are the foundation for user programs and higher layers



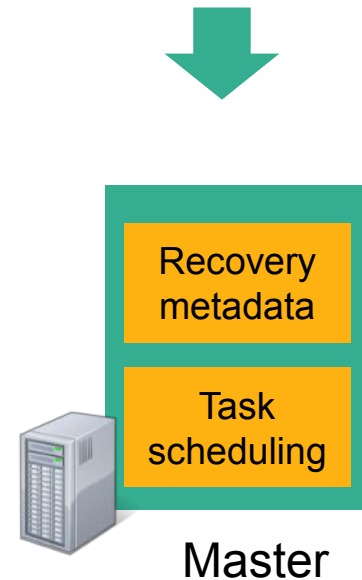
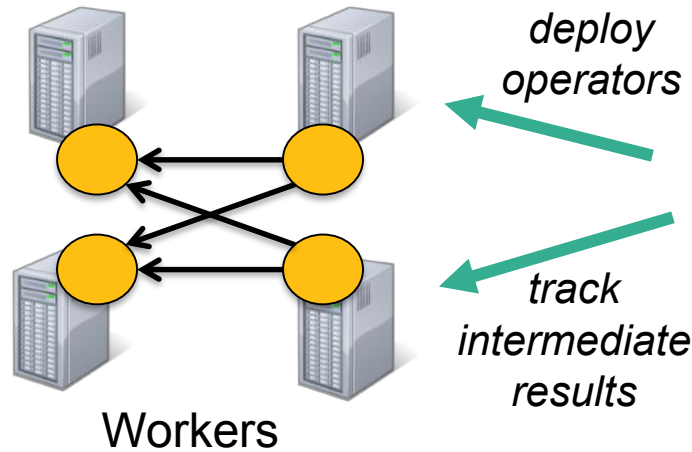
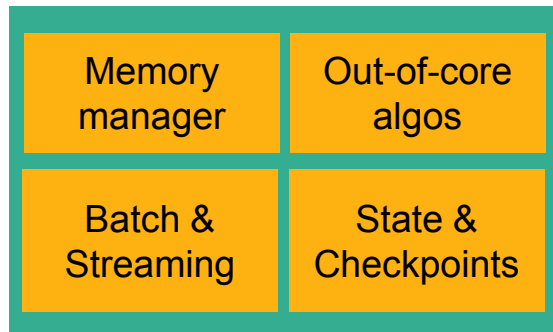
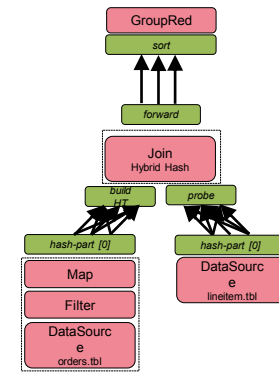
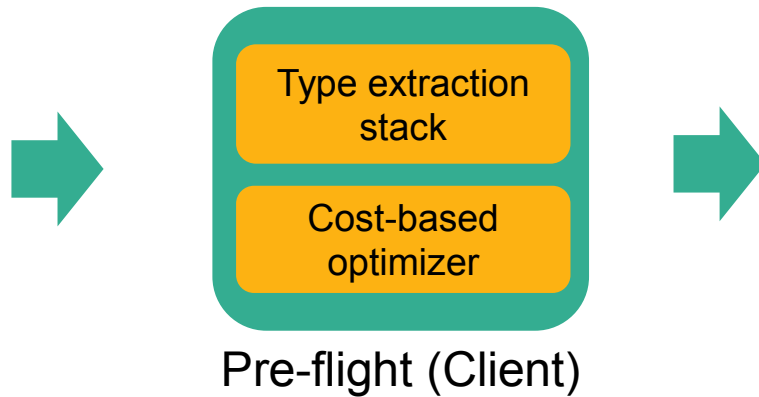
<http://flink.apache.org>

Technology inside Flink

```

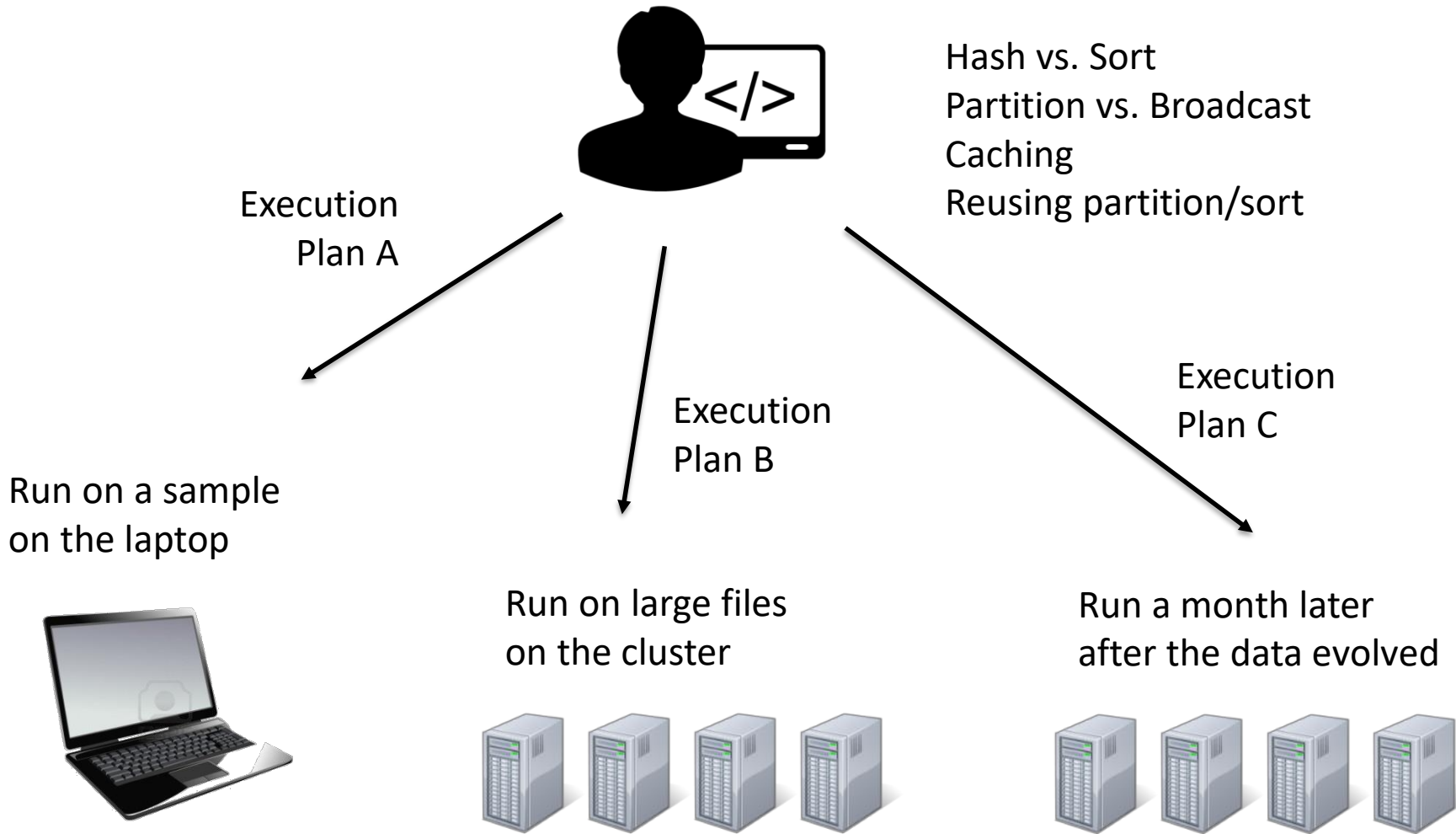
case class Path (from: Long, to:
Long)
val tc = edges.iterate(10) {
paths: DataSet[Path] =>
val next = paths
.join(edges)
.where("to")
.equalTo("from") {
(path, edge) =>
Path(path.from, edge.to)
}
.union(paths)
.distinct()
next
}
    
```

Program

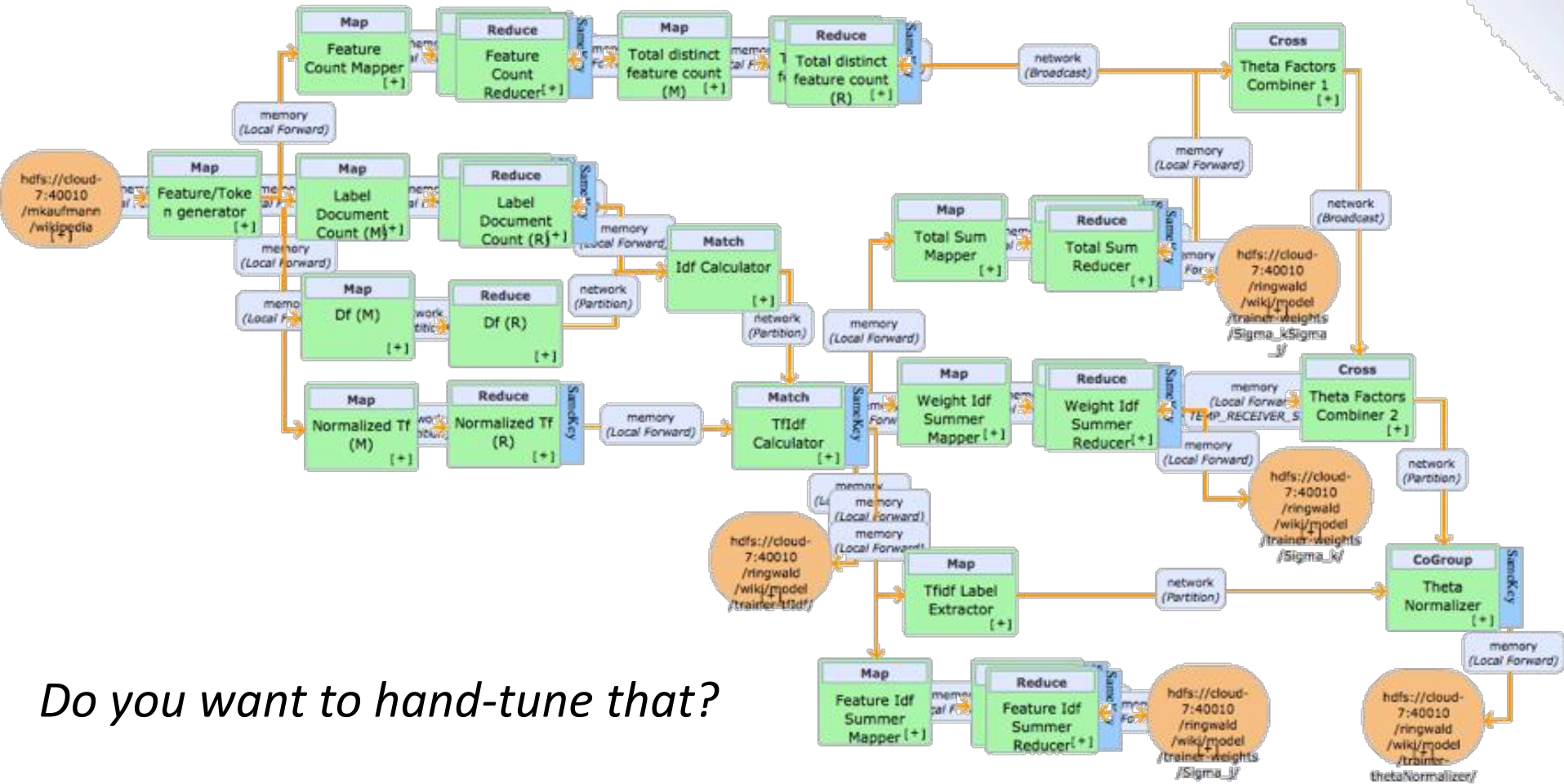


P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, K. Tzoumas:
 Apache Flink™: Stream and Batch Processing in a Single Engine. IEEE Data Eng. Bull. 38(4): 28-38 (2015)

Effect of optimization



Why optimization ?



Do you want to hand-tune that?

F. Hueske, M. Peters, A. Krettek, M. Ringwald, K. Tzoumas, V. Markl, J.C. Freytag:
Peeking into the optimization of data flow programs with MapReduce-style UDFs. ICDE 2013: 1292-1295

S. Ewen, S. Schelter, K. Tzoumas, D. Warneke, V. Markl: Iterative Parallel Data Processing with Stratosphere: an Inside Look. SIGMOD 2013

S. Ewen, K. Tzoumas, M. Kaufmann, V. Markl:

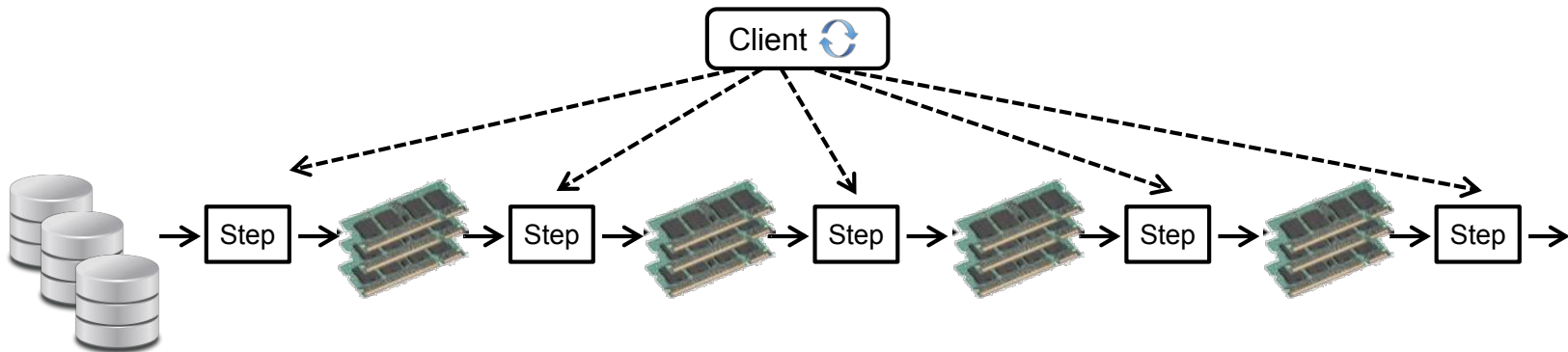
Spinning Fast Iterative Data Flows. PVLDB 5(11): 1268-1279 (2012)

ITERATIONS IN DATA FLOWS

→ MACHINE LEARNING

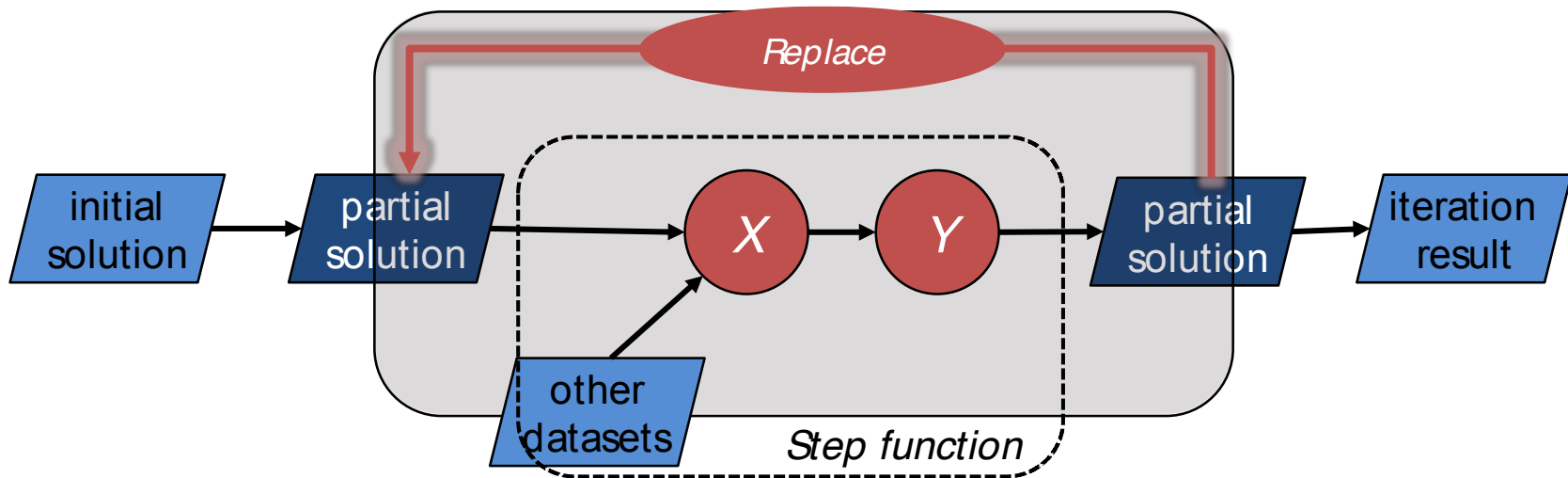
ALGORITHMS

Iterate by looping



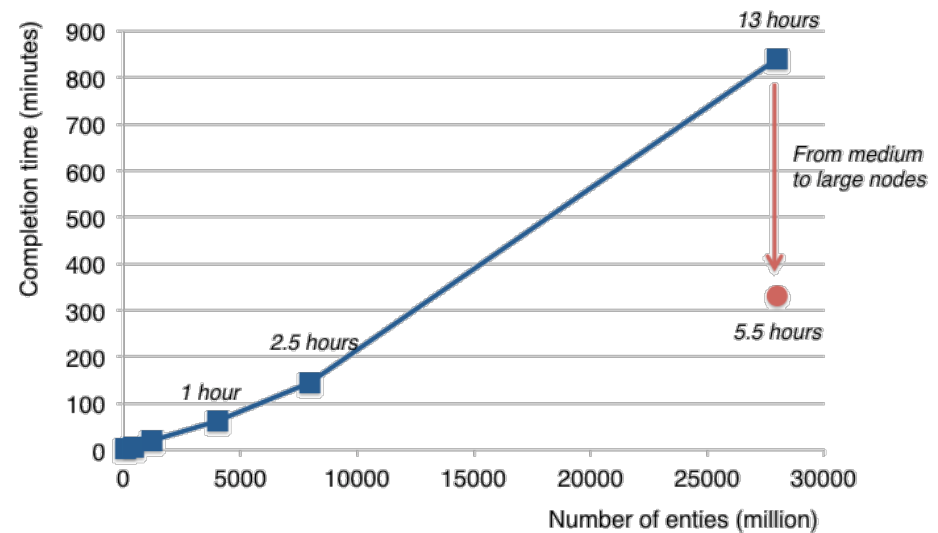
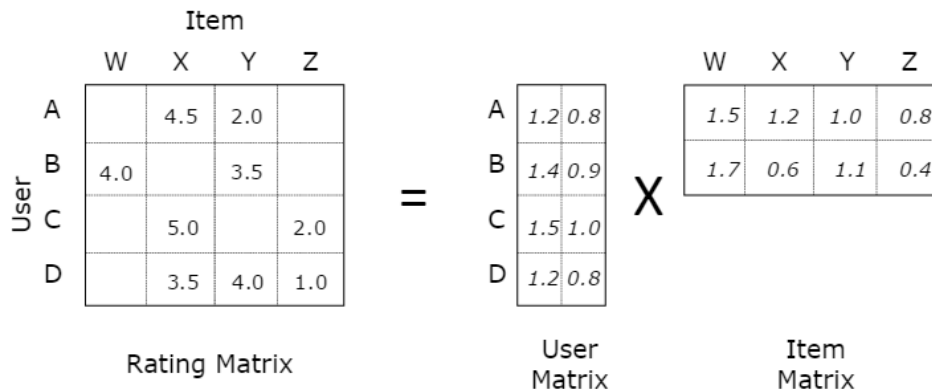
- for/while loop in client submits one job per iteration step
- Data reuse by caching in memory and/or disk

Iterate in the Dataflow



Large-Scale Machine Learning

Factorizing a matrix with
28 billion ratings for
recommendations



*(Scale of Netflix
or Spotify)*

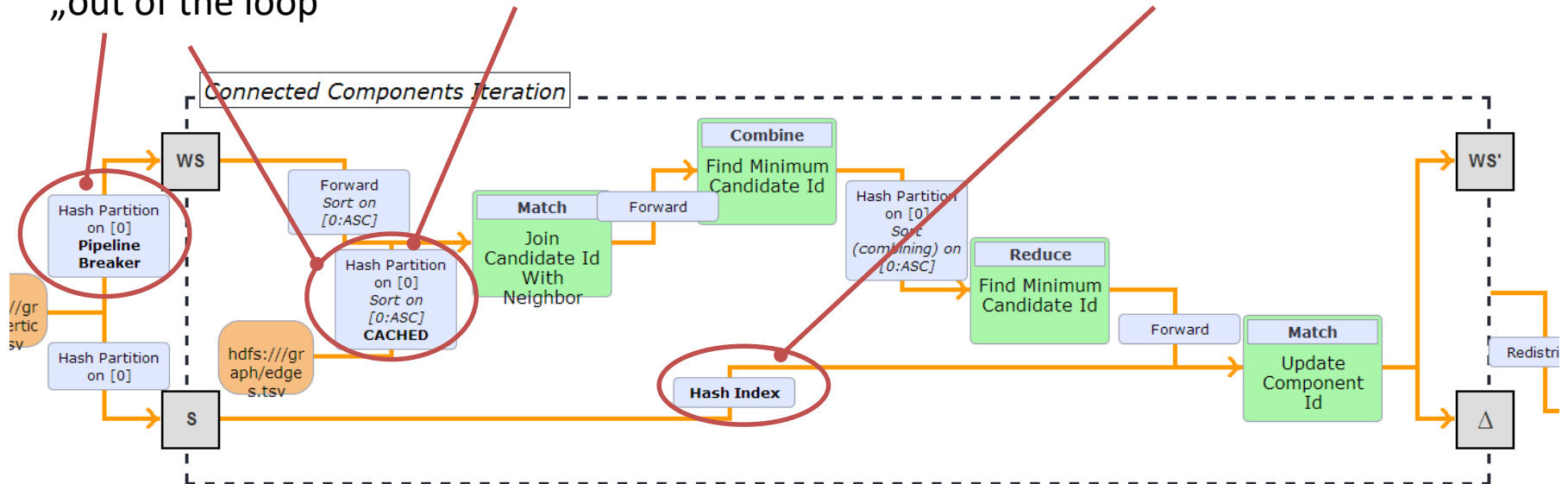
More at: <http://data-artisans.com/computing-recommendations-with-flink.html>

Optimizing iterative programs

Pushing work
„out of the loop“

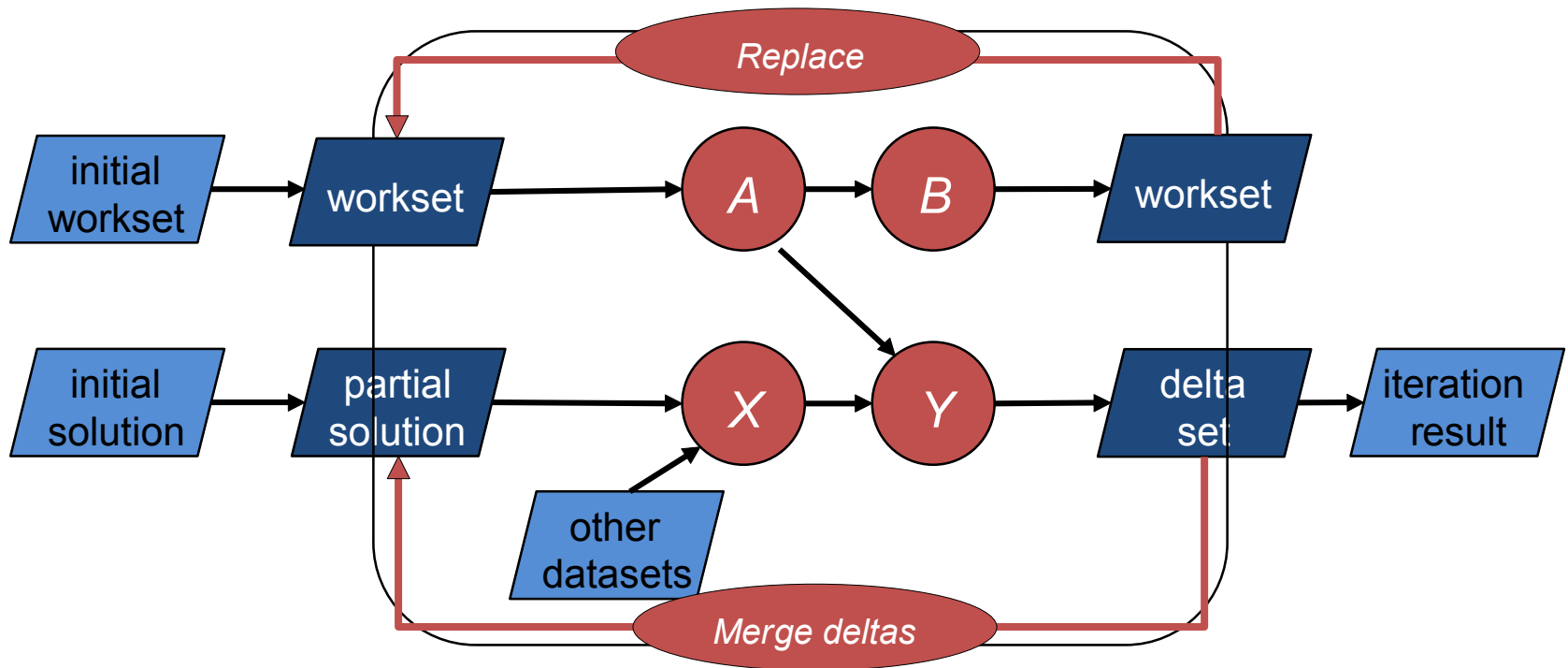
Caching Loop-invariant Data

Maintain state as index

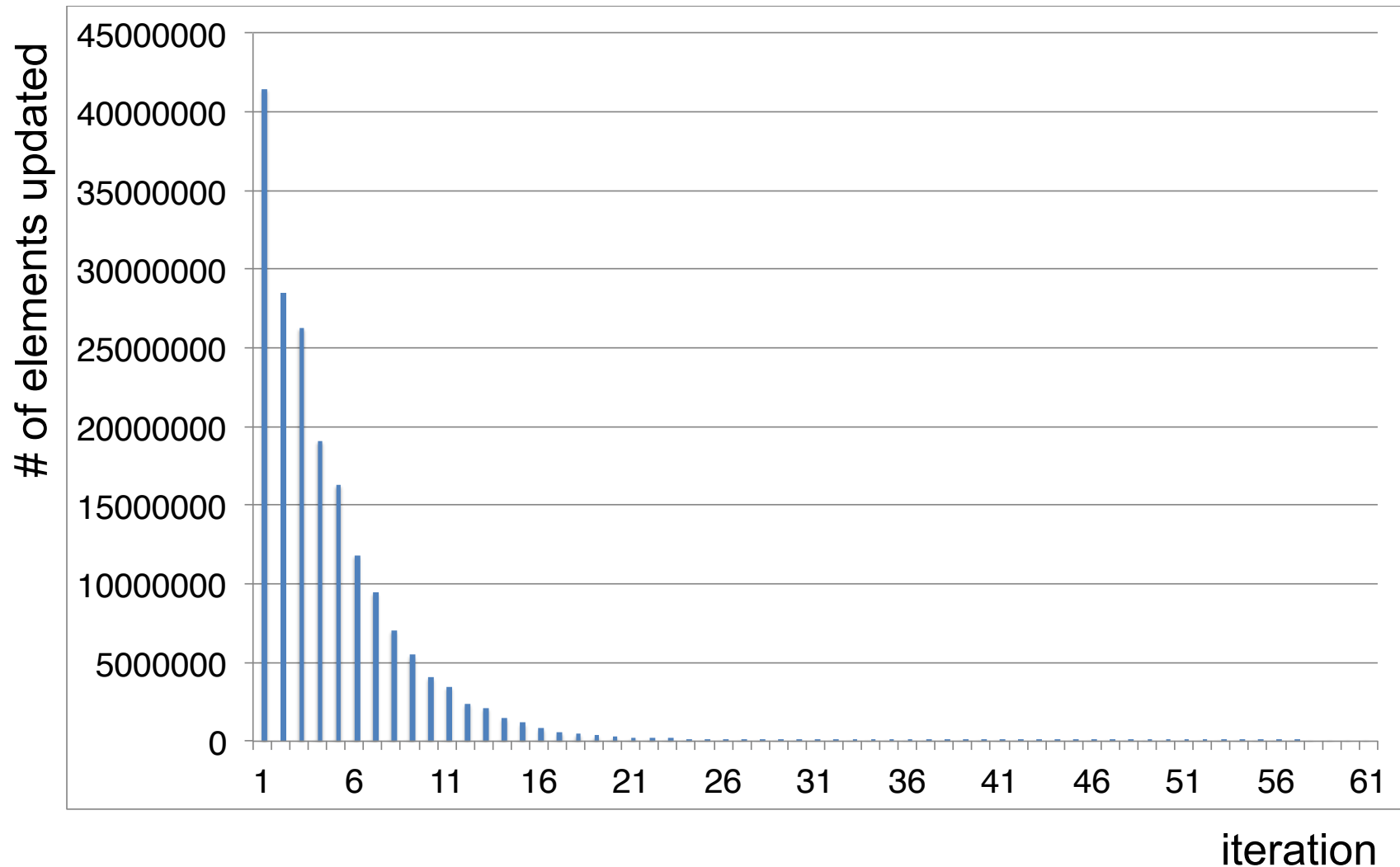


STATE IN ITERATIONS → GRAPHS AND MACHINE LEARNING

Iterate natively with deltas

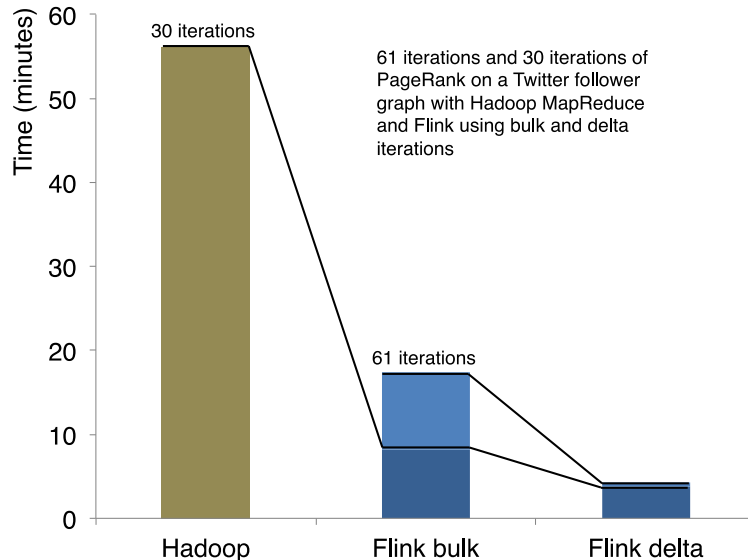
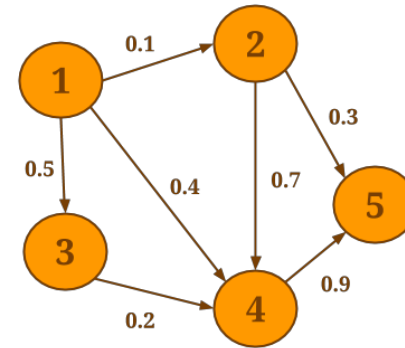


Effect of delta iterations...



... very fast graph analysis

Performance competitive
with dedicated graph
analysis systems



61 iterations and 30 iterations of PageRank on a Twitter follower graph with Hadoop MapReduce and Flink using bulk and delta iterations

... and mix and match
ETL-style and graph analysis
in one program

More at: <http://data-artisans.com/data-analysis-with-flink.html>

“STREAMING DATA” ANALYSIS

Bounded and unbounded data

- Bounded data: a dataset with a natural beginning and end
 - E.g., current position of all trucks in a fleet, content of a data warehouse
- Unbounded data: a dataset without a natural beginning and end
 - E.g., customers of our products, tweets about our product
- Note: few data is bounded by nature; most bounded data is a view over unbounded data

By courtesy of Kostas Tzoumas

Stream and batch processing

- Stream processing: continuous processing that continuously produces results
 - E.g., a Java program that connects to a socket and parses the socket contents
 - Apache Flink, Apache Storm
- Batch processing: processing that takes finite time to complete and produces results only in the end
 - E.g., sorting a file
 - Apache Hadoop MapReduce, Apache Spark

By courtesy of Kostas Tzoumas

How do they fit together

- Batch processing over bounded data
 - Natural
- Stream processing over bounded data
 - Treats bounded data as subset of stream
- Batch processing over unbounded data
 - Needs a pre-processing stream processing phase that splits stream into chunks
- Stream processing over unbounded data
 - Natural

By courtesy of Kostas Tzoumas

A different view

- What changes faster? Your code or your data?
- $d\text{data}/dt \gg d\text{code}/dt$ is a data streaming problem
- $d\text{code}/dt \gg d\text{data}/dt$ is a data exploration problem (and likely to become a data streaming problem later)

Credits Joe Hellerstein

Defining windows in Flink



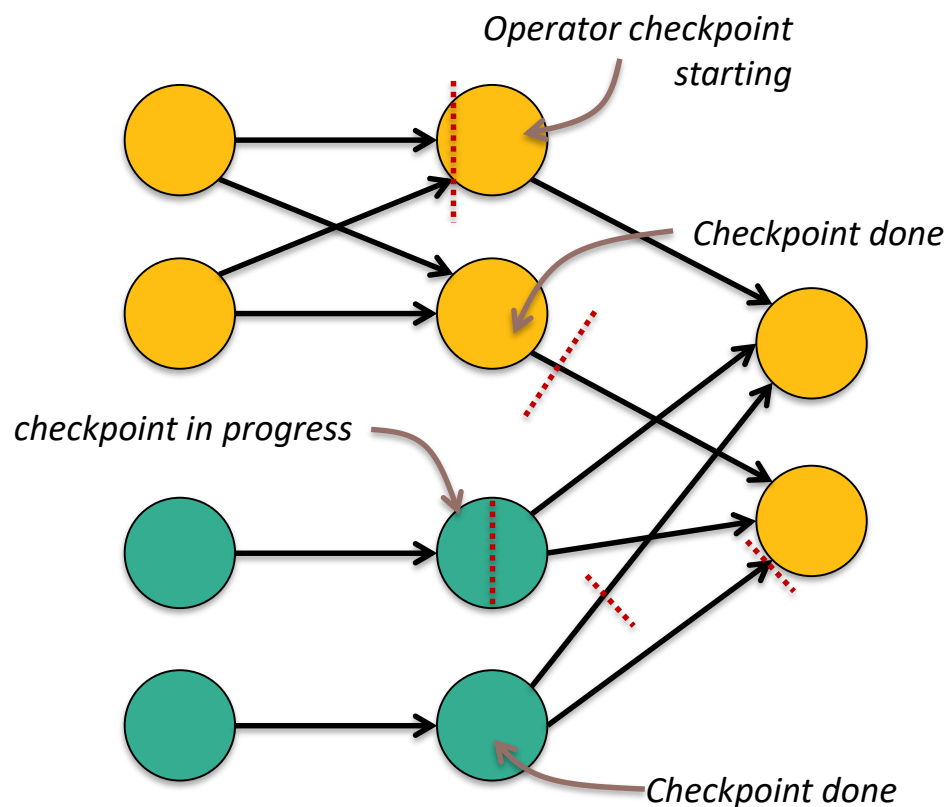
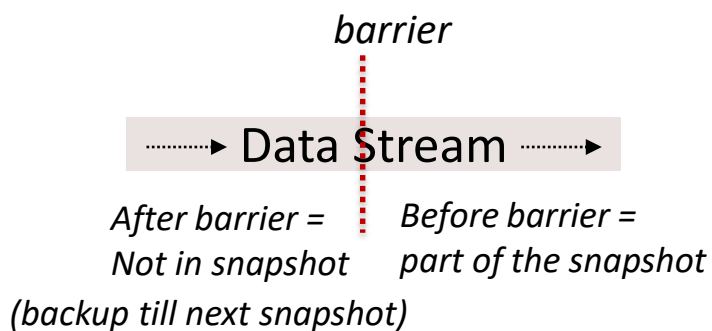
- Trigger policy
 - When to trigger the computation on current window
- Eviction policy
 - When data points should leave the window
 - Defines window width/size
- E.g., count-based policy
 - evict when $\#elements > n$
 - start a new window every n -th element
- Built-in: Count, Time, Delta policies

Checkpointing / Recovery

- Flink acknowledges batches of records
 - Less overhead in failure-free case
 - Currently tied to fault tolerant data sources (e.g., Kafka)
- Flink operators can keep state
 - State is checkpointed
 - Checkpointing and record acks go together
- Exactly one semantics for state

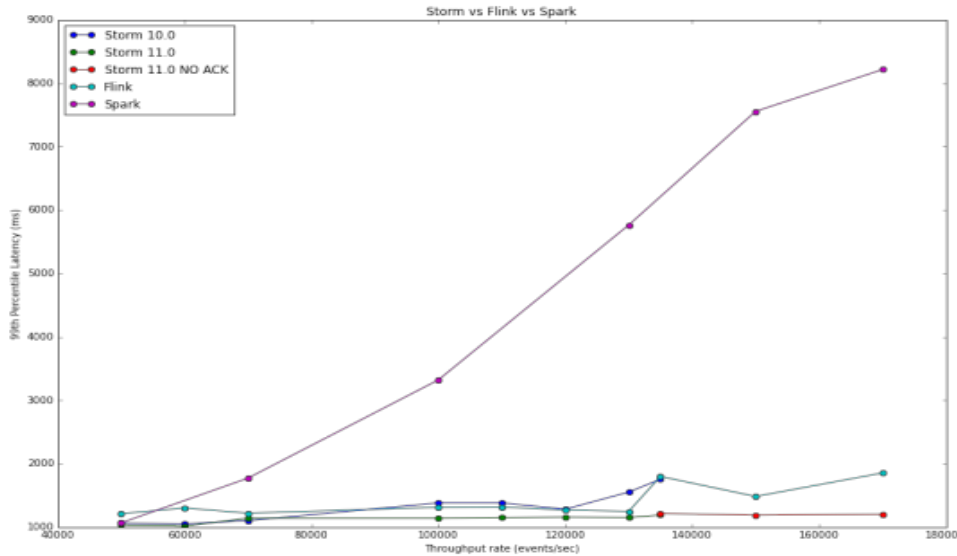
Checkpointing / Recovery

Pushes checkpoint barriers through the data flow



Chandy-Lamport Algorithm for consistent asynchronous distributed snapshots

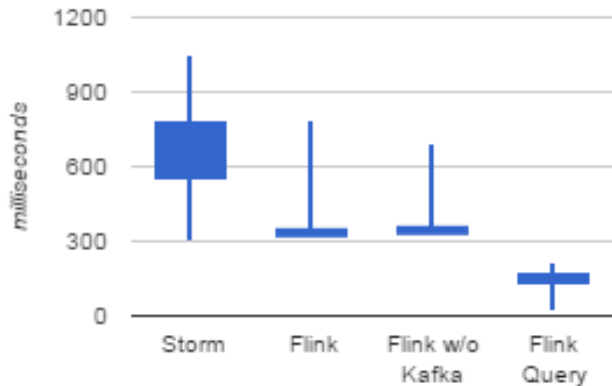
Some Benchmark Results



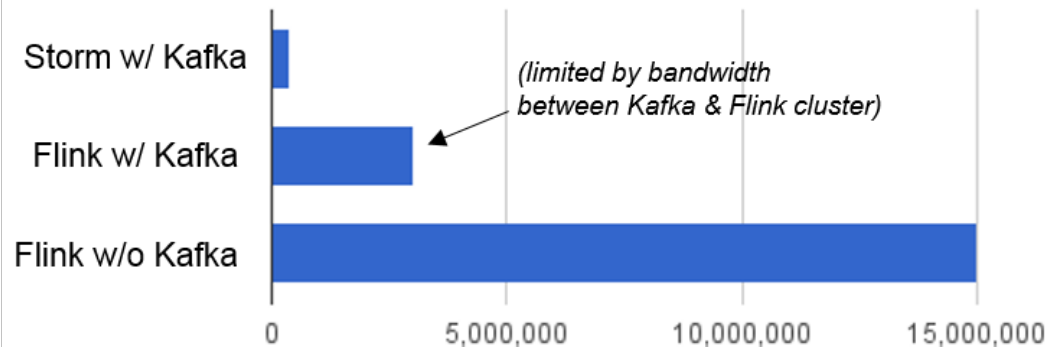
Initially performed by Yahoo! Engineering, Dec 16, 2015,

[..]Storm 0.10.0, 0.11.0-SNAPSHOT and Flink 0.10.1 show sub-second latencies at relatively high throughputs[..]. Spark streaming 1.5.1 supports high throughputs, but at a relatively higher latency.

Latency Distribution



Maximum Throughput (events/sec)

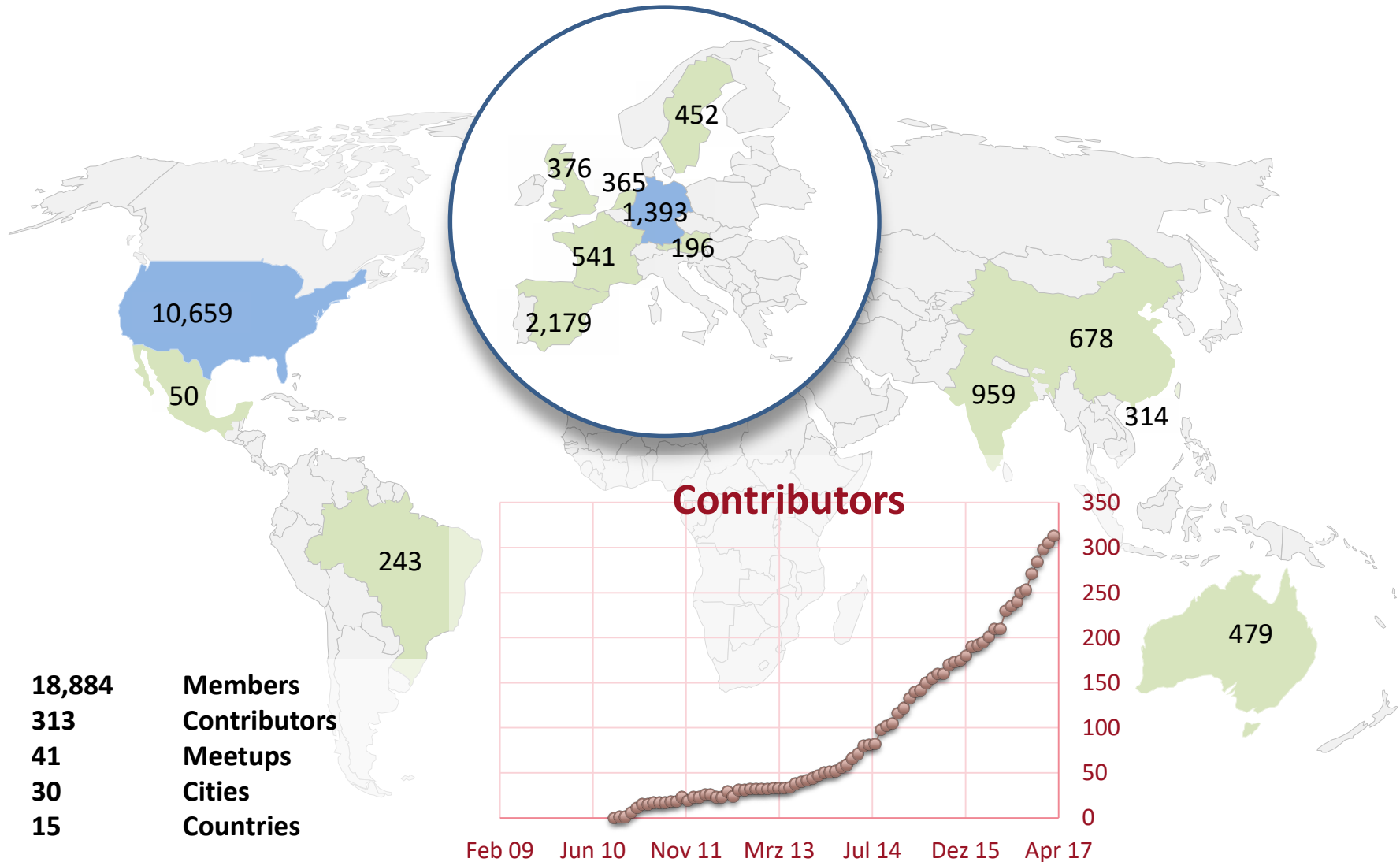


<http://yahooeng.tumblr.com/post/135321837876/benchmarking-streaming-computation-engines-at>
<https://data-artisans.com/extending-the-yahoo-streaming-benchmark/>

J. Soto and V. Markl, "A Historical Account of Apache Flink," [Online].
Available: [http://www.dima.tu-berlin.de/fileadmin/fg131/
Informationsmaterial/Apache_Flink_Origins_for_Public_Release.pdf](http://www.dima.tu-berlin.de/fileadmin/fg131/Informationsmaterial/Apache_Flink_Origins_for_Public_Release.pdf)

THE FLINK COMMUNITY

Flink Community as of Today



Some Highly Engaged Users



Largest job has > 20 operators, runs on > 5000 vCores in 1000-node cluster, processes millions of events per second



Complex jobs of > 30 operators running 24/7, processing 30 billion events daily, maintaining state of 100s of GB with exactly-once guarantees



30 Flink applications in production for more than one year. 10 billion events (2TB) processed daily

By courtesy of Kostas Tzoumas

Other Companies in the Flink Community

otto group

amadeus

UBER

 **zalando**

Pragsis  **Bidoop**
Big Data Analytics



NETFLIX



PARALLEL
MACHINES



ING  DiBa



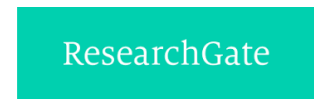
 Lightbend

EMC²



 **tree logic**

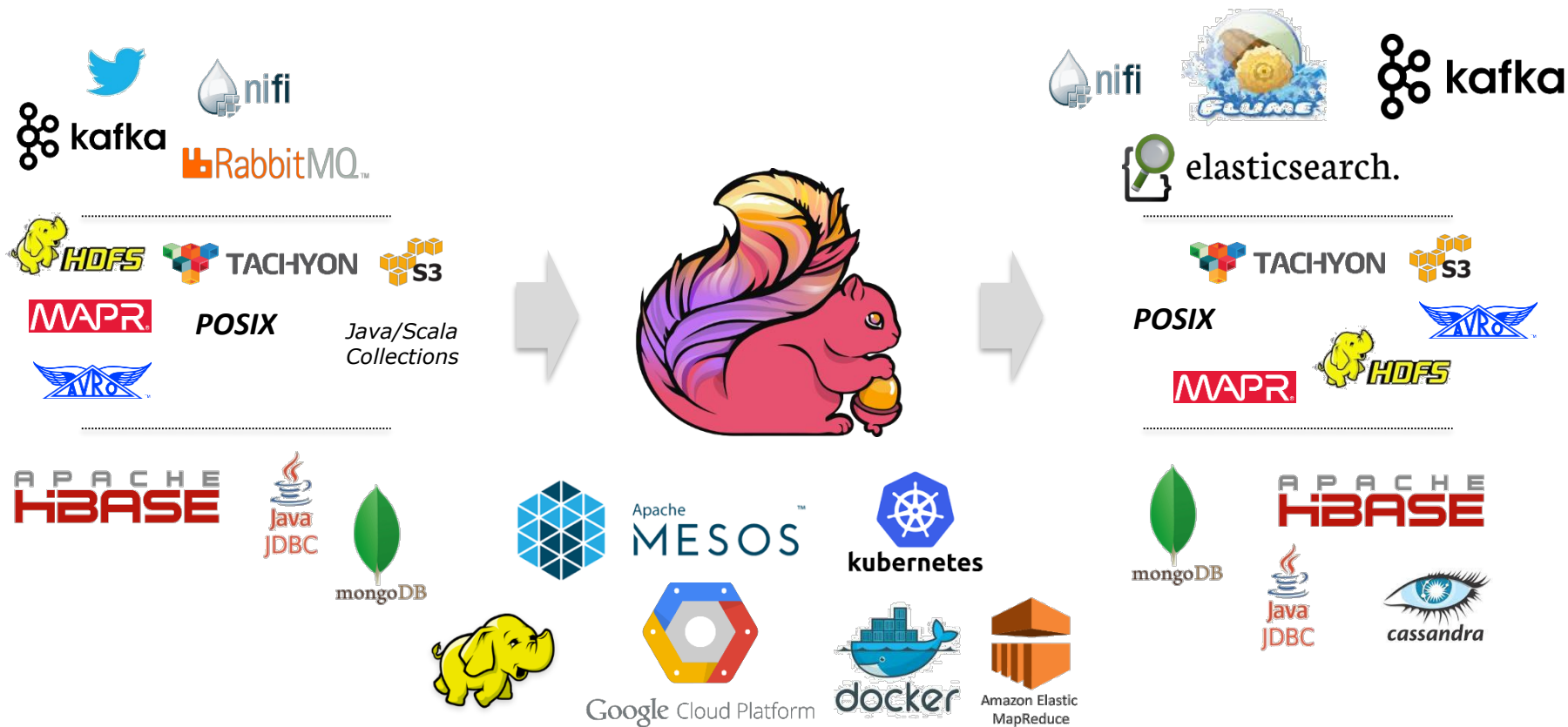
EURV
NOVA



MUX

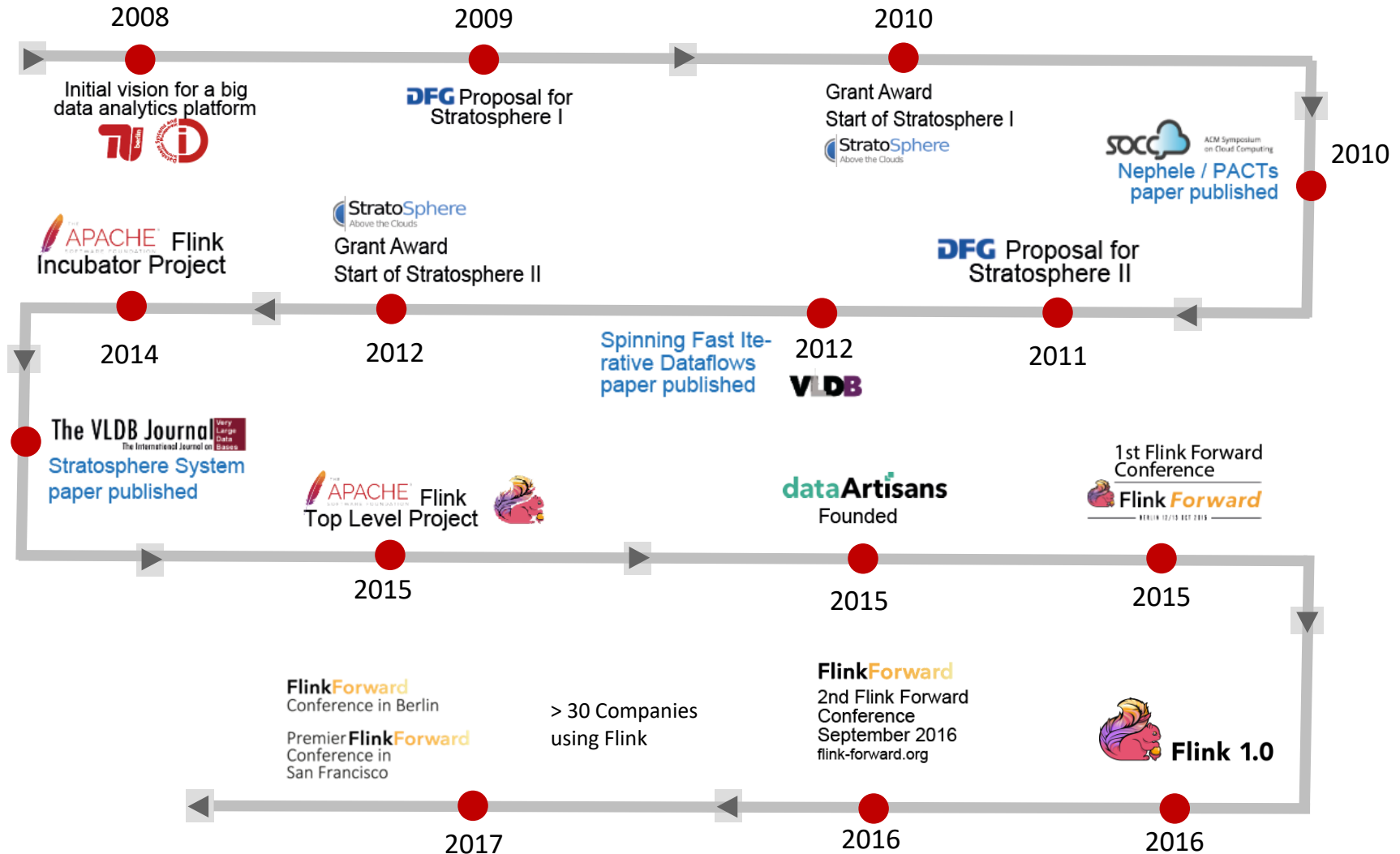
<https://flink.apache.org/poweredby.html>

Flink in the ecosystem

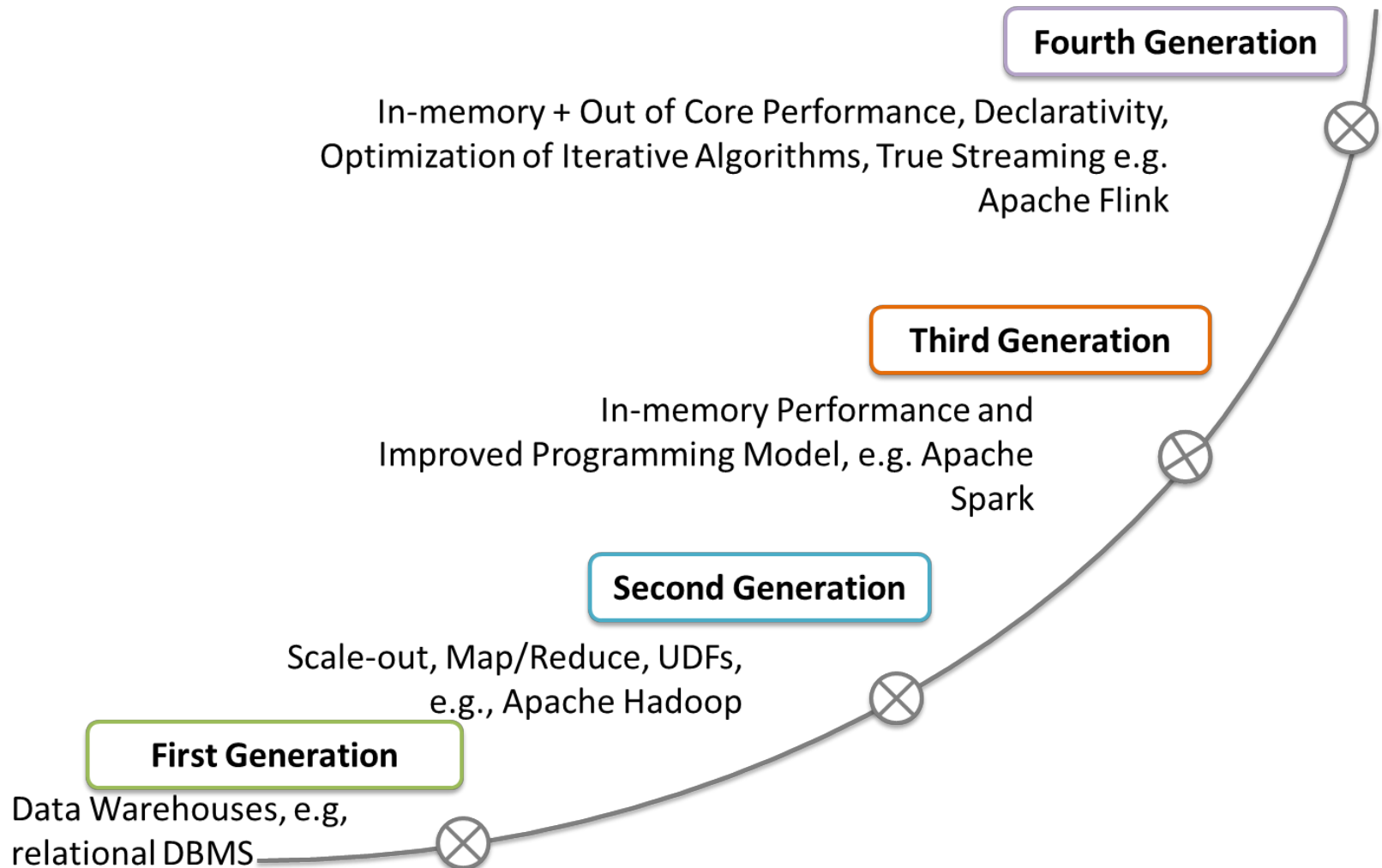


By courtesy of Kostas Tzoumas

Timeline of Flink



Evolution of Big Data Platforms



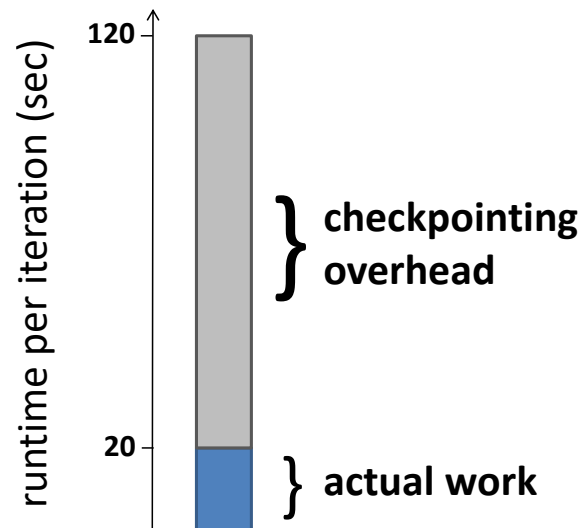
Fault tolerance

Pessimistic Recovery:

- Write intermediate state to stable storage
- Restart from such a checkpoint in case of a failure

Problematic:

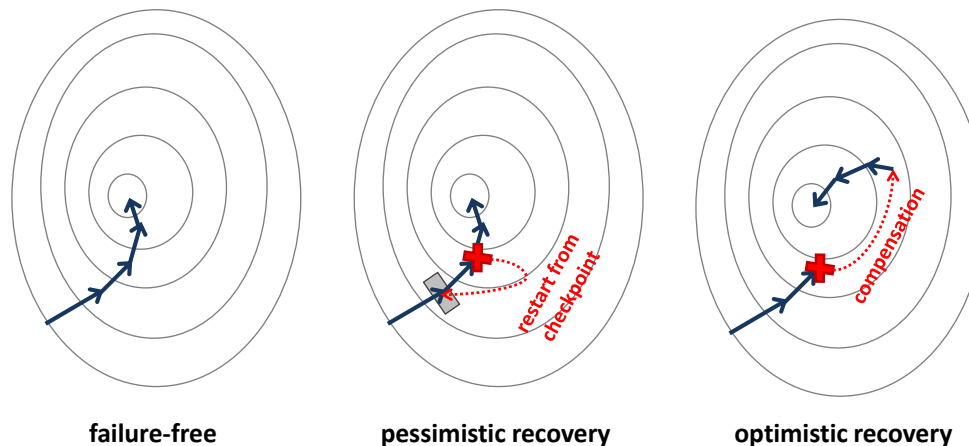
- High overhead, checkpoint must be replicated to other machines
- Overhead always incurred, even if no failures happen!



- **How can we avoid this overhead in failure-free cases**

Optimistic recovery

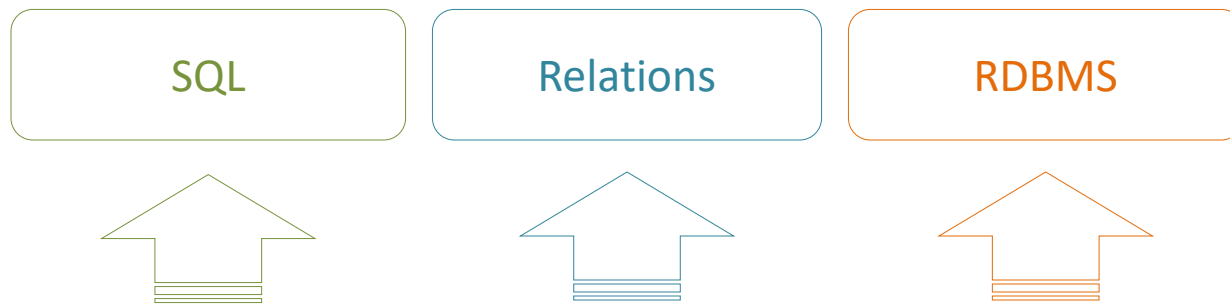
- Many data mining algorithms are **fixpoint algorithms**
- **Optimistic Recovery**: jump to a different state in case of a failure, still converge to solution



- No checkpoints → **No overhead in absence of failures!**
- algorithm-specific **compensation function** must restore state

Declarative Data Processing and Mosaics

A Billion \$\$\$ Mantra...



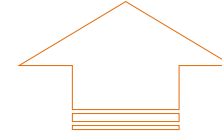
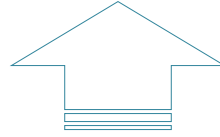
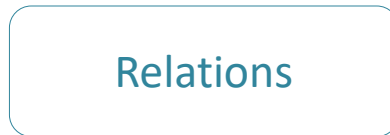
Declarative Data Processing

An effective, formal foundation based on relational algebra and calculus (Codd '71).

A simple, high-level language for querying data (Chamberlin '74).

An efficient, low-level execution environment tailored towards the data (Selinger '79).

With 40+ years of success...



Declarative Data Processing

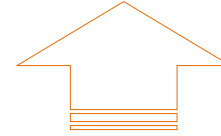
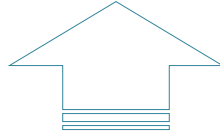
Is Being Revised



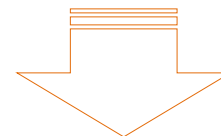
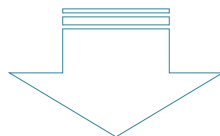
SQL

Relations

RDBMS



Declarative Data Processing



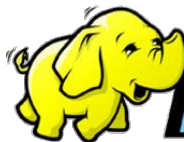
Second-Order Functions

Distributed Collections

Parallel Dataflow Engines



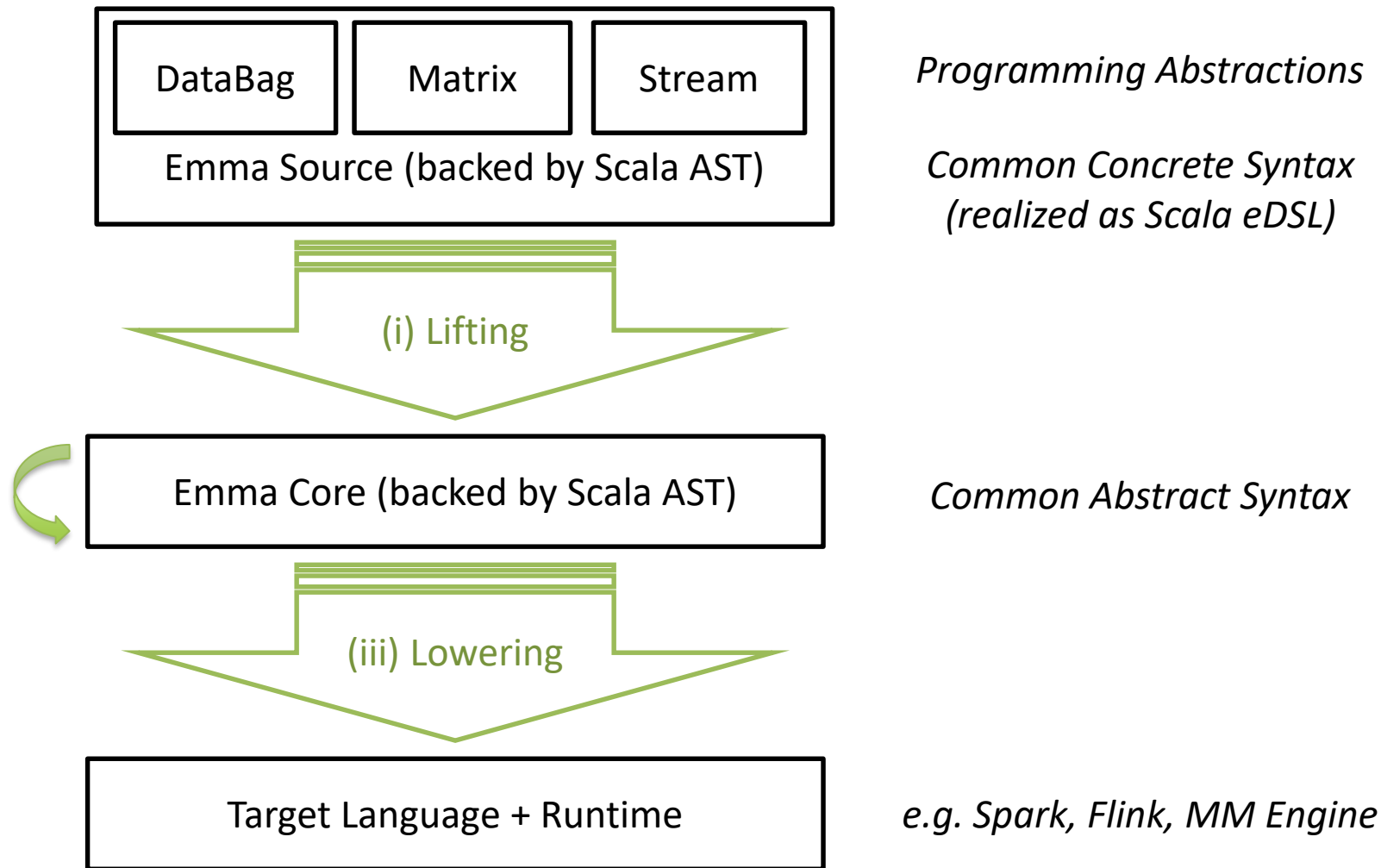
Flink



hadoop

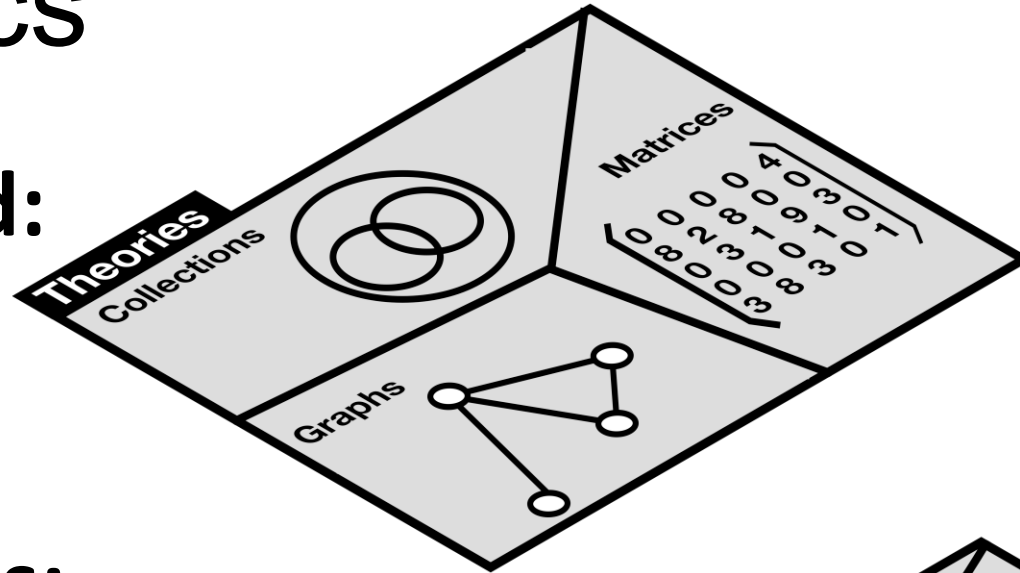
Spark

Fine grained parallelism through deep language Embedding (EMMA)

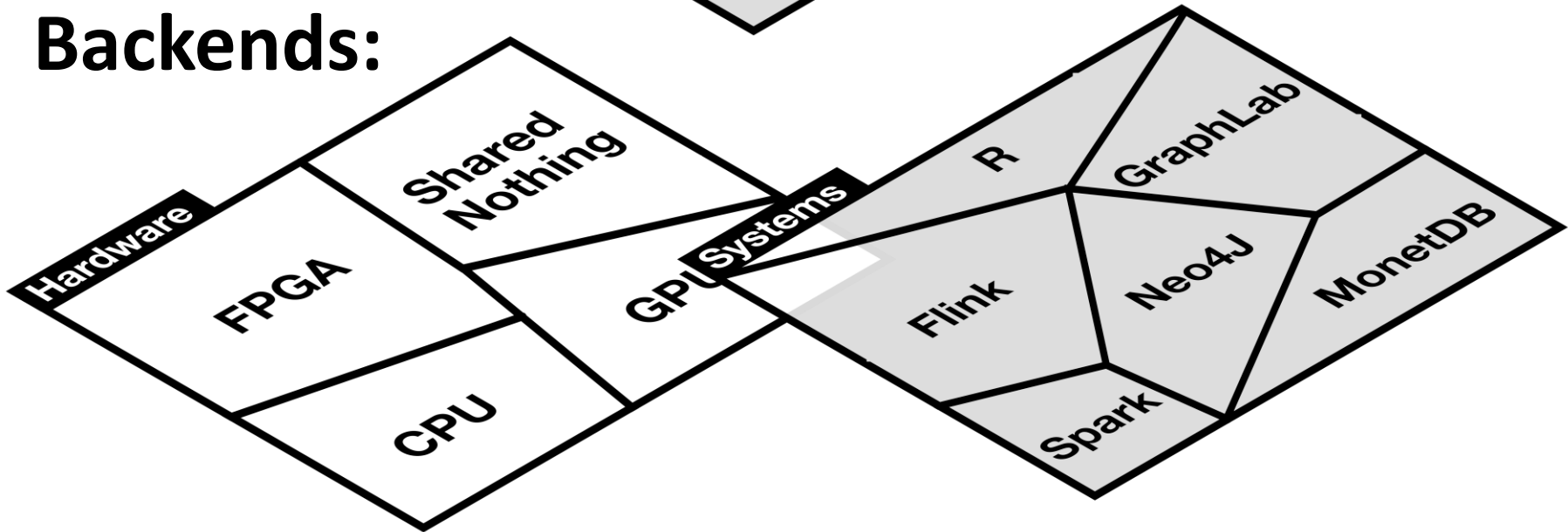


Mosaics

Frontend:

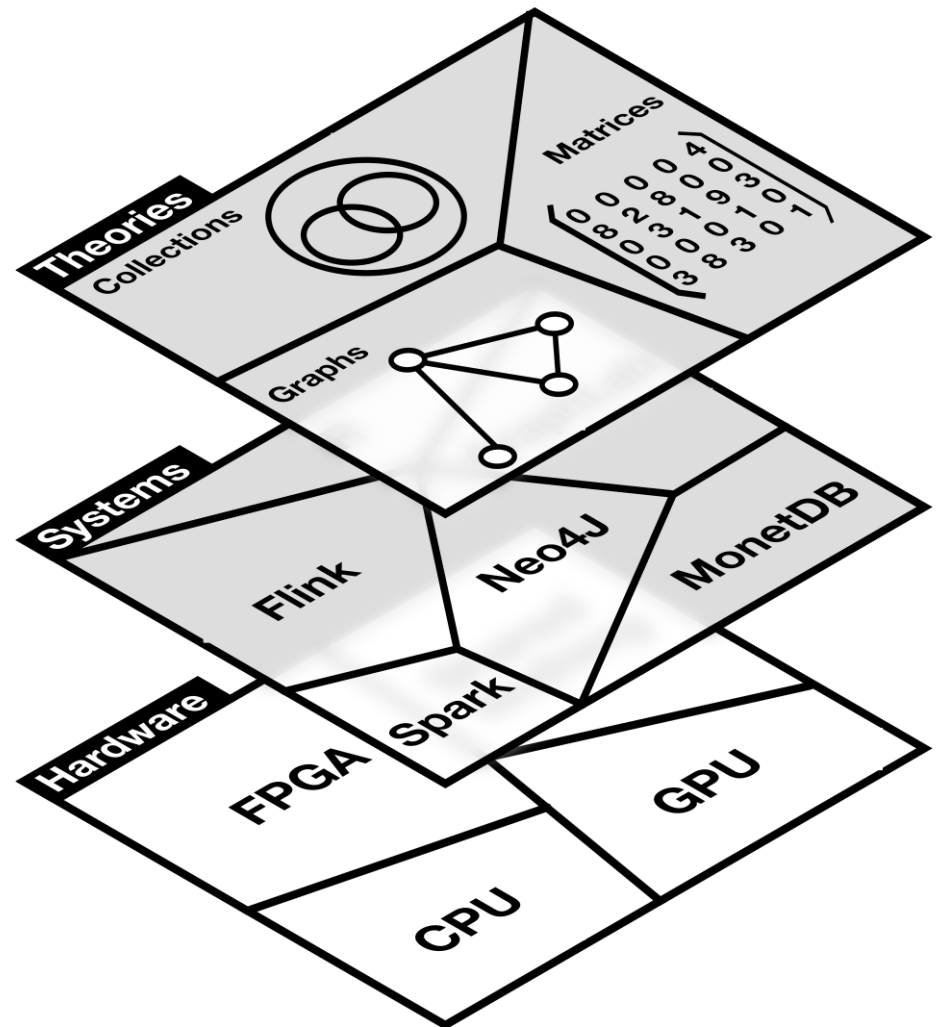


Backends:

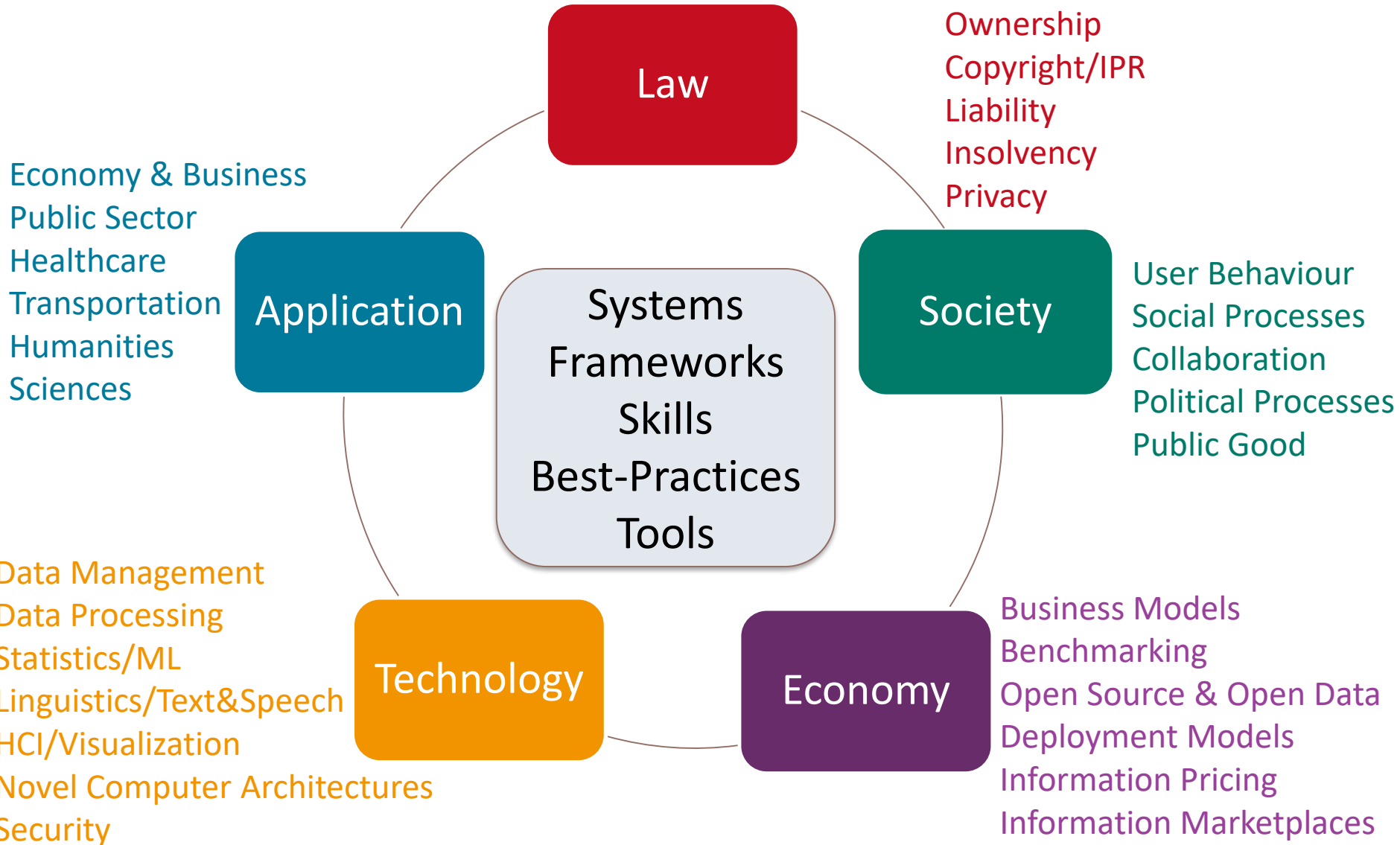


Research

1. Unifying Modelling Across Theories
2. Cross Theory Optimization
3. Optimizing Across Engines
4. Predicting and Learning Program Runtimes
5. Optimizing Across Hardware
6. Generating Hardware-Targeted Code



The Five Dimensions of Big Data



Acknowledgements

I would like to thank the **members of the Stratosphere** project, the **Berlin Big Data Center**, and my **national and international collaborators**. In particular, I would like to thank my **former and current students and postdocs at DIMA/TU Berlin and DFKI**. Without them it would not have been possible to realize the visions into concrete software system artifacts. Last, but not least, I would like to thank **all of the contributors and users in the Apache Flink community**, without them the Stratosphere project and the correlated research of the Berlin Big Data Center would not have achieved the worldwide impact that we have experienced over the last few years