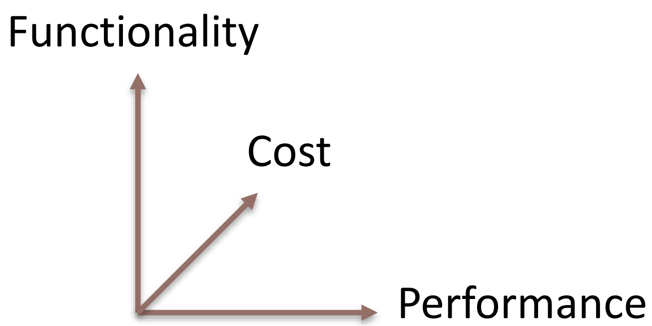
Big Data Management and Apache Flink

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

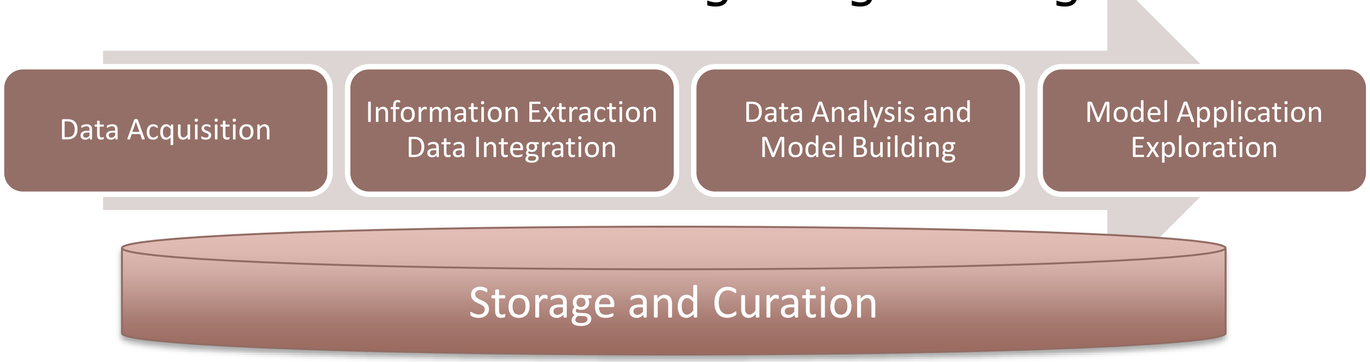
volker.markl@tu-berlin.de

Data Management

- Managing the Complexity of Data Processing



- Industrialization of the Knowledge Engineering Process



- Study Interactions and Trade-Offs & Build Generic Systems
  - 3P (**p**rocessing environment, **p**rograms, **p**eople)
  - Performance metrics
    - Throughput vs. quality measures
    - Human latency vs. system latency

A (rather incomplete) Data Management timeline

70s	80s	90s	00s	2004 - 2009	2009 - now
Appearance of Relational Databases	*First parallel shared-nothing Architectures*	*Open Source Projects and mainstream Databases*	*First columnar storage Databases*	*NoSQL and UDF-based commodity analytics*	*Alternative MapReduce implementations go mainstream*
SQL/OLTP	***OLAP/Warehouse***	***OODBMS***	***XML-DBMS***	***„map/reduce“***	***„in-memory“***
Ingres Oracle System-R	Gamma TRACE Teradata	DB2 Oracle BerkeleyDB MS Access PostgreSQL MySQL	MonetDB C-Store Vertica Aster Data GreenPlum ParAccel	MapReduce Hadoop	Spark Flink Impala Giraph Hive HBase SAP Hana Blu

1974: SQL

Open Source rises

2003: Internet Explodes

Criticism of MapReduce

Google publishes MapReduce

Some Aspects of the Data Processing Pipeline

Data Generation Acquisition	Data Analysis Processing	Data Storage Curating	Data Visualisation Usage & Services
• Structured & Unstructured Data • Event processing • Streams • Sensor Networks • Multimodality	• Data pre-processing • In Memory processing • Semantic & Sentiment analysis • Data Correlation • Pattern Recognition • Real time analysis • Machine learning	• In Memory storage • Data augmentation & annotation • Data validation & redundancy elimination • Cloud • No/NewSQL • Consistency • Revision and updating	• Decision support • Modelling • Simulation • Prediction • Exploration • Domain specific usage • Control

Security, Data protection , Privacy, Trust,

Source: bigdatavalue.eu

Data & Analysis: Increasingly Complex!

scalability DM

data volume too large — ***V**olume*
data rate too fast — ***V**elocity*
data too heterogeneous — ***V**ariability*

data too uncertain — ***V**eracity*

algorithms ML

Data

Reporting — aggregation, selection
Ad-Hoc Queries — SQL, XQuery
ETL/ELT — MapReduce

Data Mining — MATLAB, R, Python
Predictive/Prescriptive — MATLAB, R, Python

DM scalability

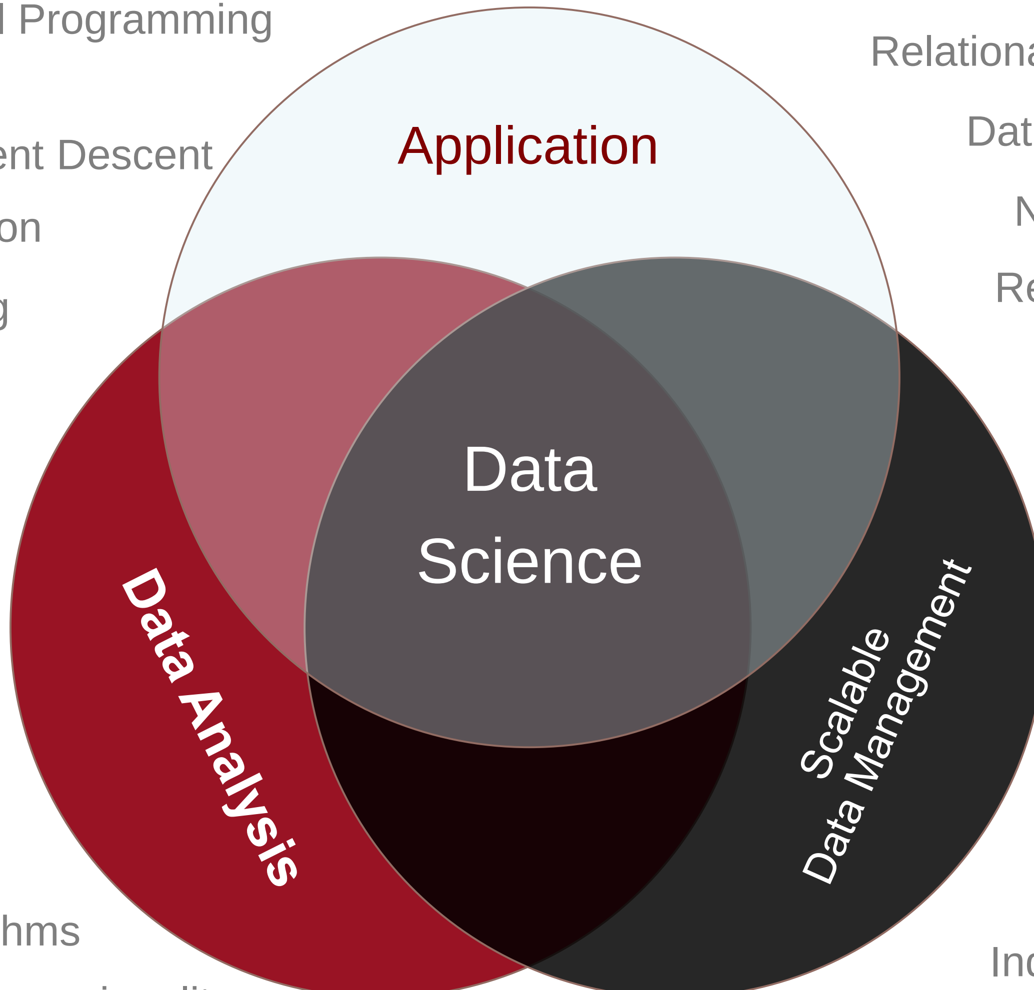
ML algorithms

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
Accuracy



Relational Algebra / SQL
Data Warehouse/OLAP
NF^2 /XQuery
Resource Management
Hardware Adaptation
Fault Tolerance
Memory Management
Parallelization
Scalability
Memory Hierarchy
Compression
Compiler
Query Optimization
Indexing
Data Flow
Control Flow
Real-Time
Performance

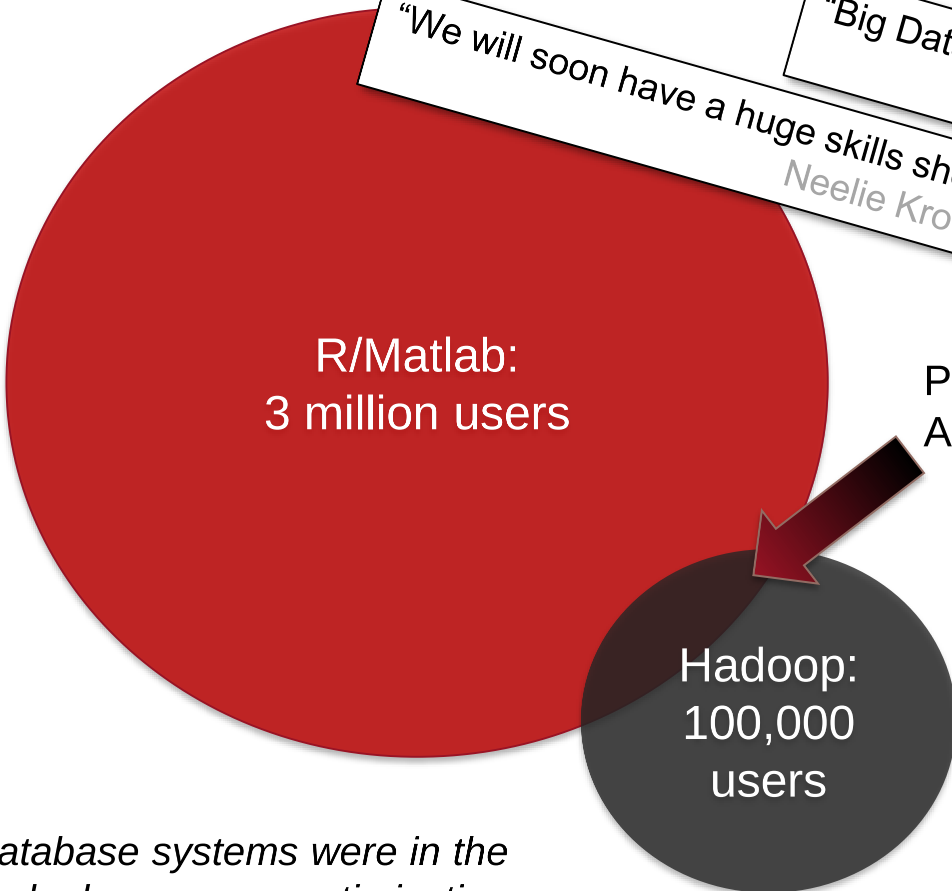
New Technology to the Rescue!

Big Data Analytics Requires Systems Programming

"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

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



People with Big Data Analytics Skills

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

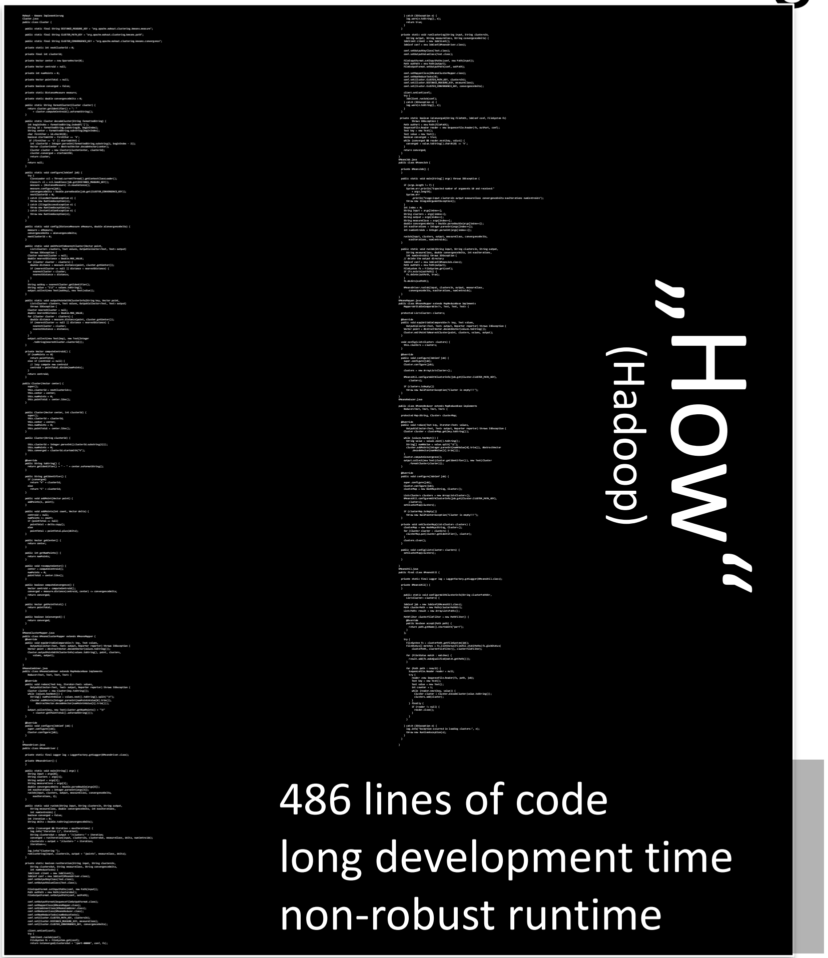
Big Data is now where database systems were in the 70s (prior to relational algebra, query optimization and a SQL-standard)!

Declarative languages to the rescue!

Declarative Languages to the Rescue!

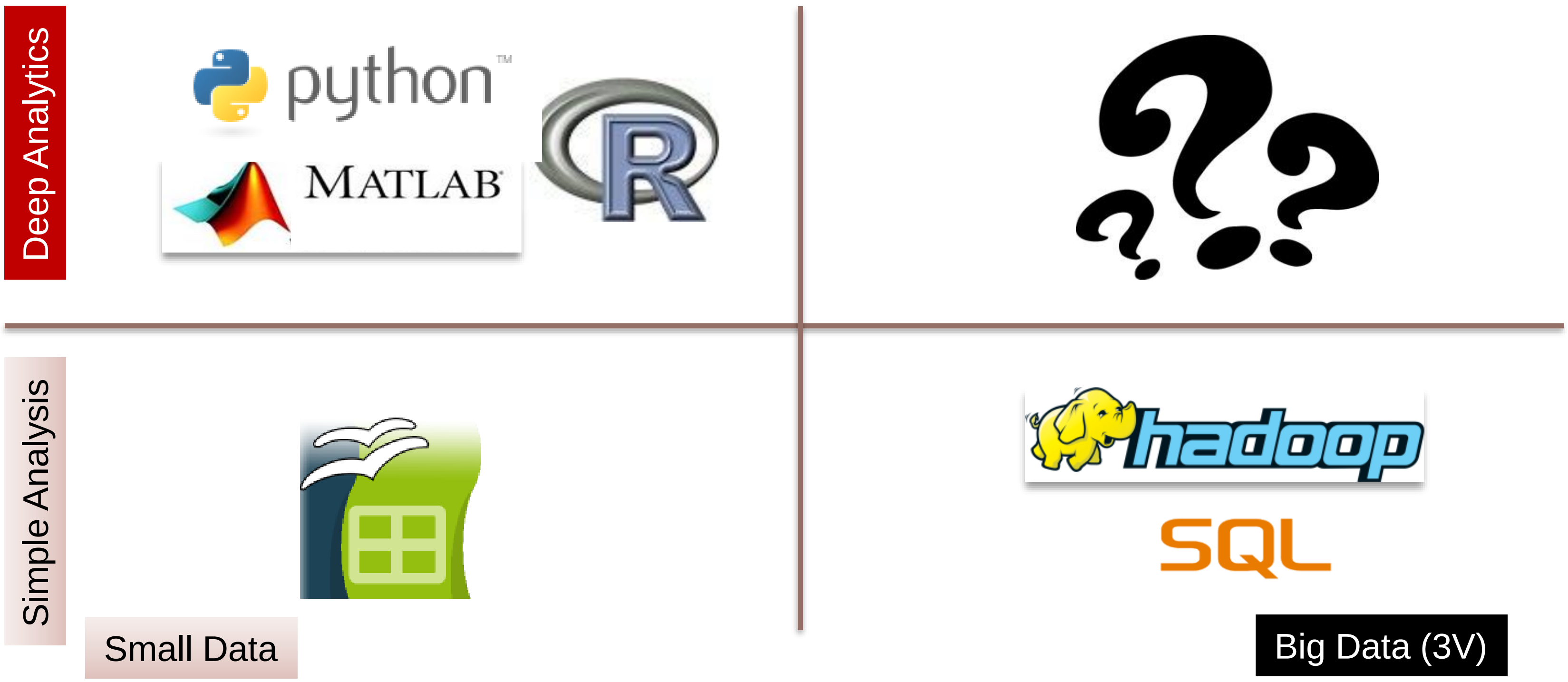
„What“, not „how“ Example: k-Means Clustering

Declarative data analysis program with automatic optimization, parallelization and hardware adaption

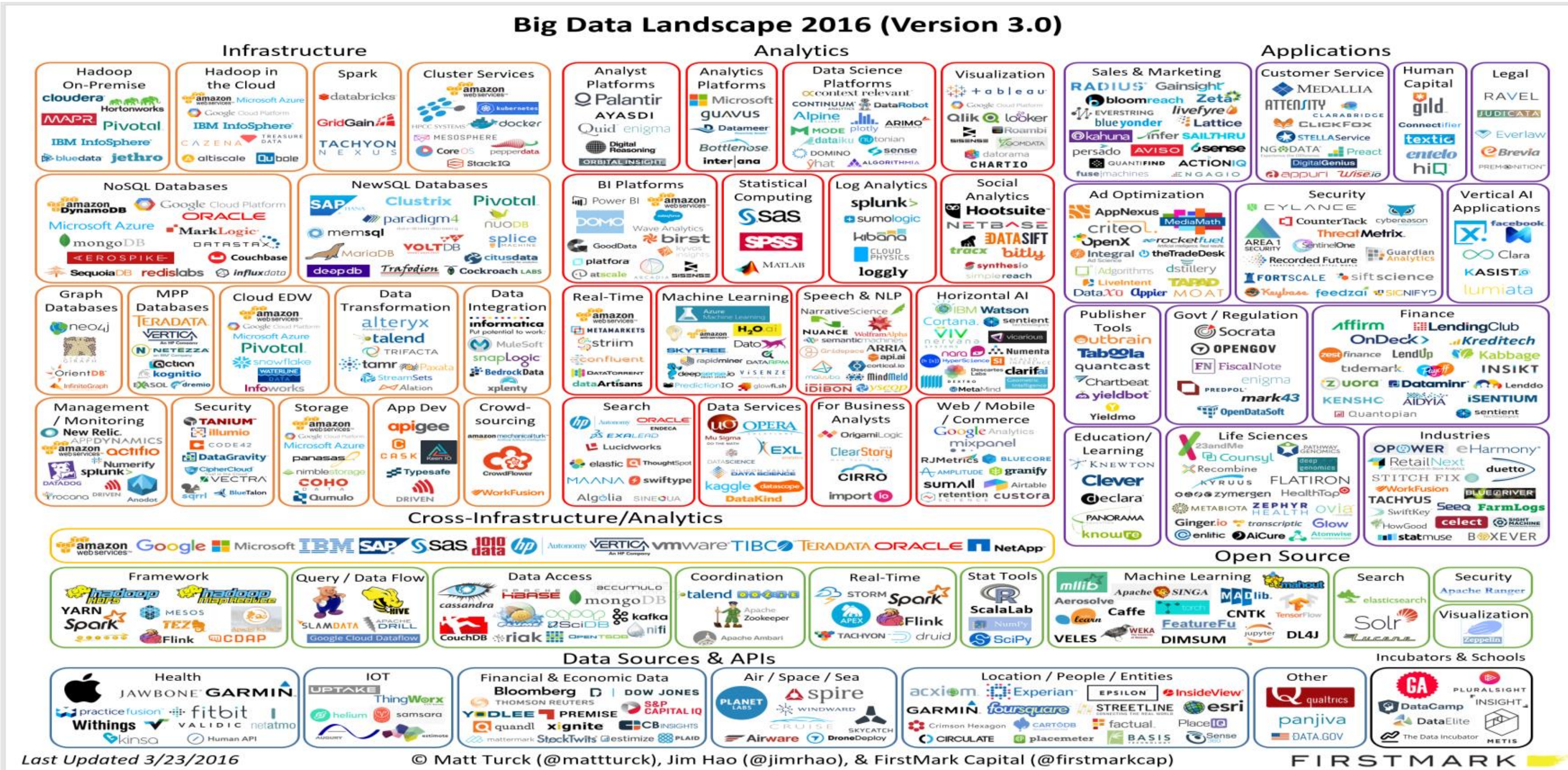


Hand-optimized code (data-, load- and system dependent)

Deep Analysis of „Big Data“ is Key!



A Zoo of Technologies!



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

Challenge: Technologies for Data Science at the Intersection of Data Management and Machine Learning

Declarative Languages
Automatic Adaption
Scalable processing

DM

Think DA-algorithms in a scalable way

declarative

Process (iterative) algorithms in a scalable way

DA

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

aggregation → relational algebra → UDF → iteration/recursion → linear algebra
graph algebrae
etc.

Agenda

- Big Data Management
- Apache Flink
  - Data Stream Analysis
  - Iterations
  - The Flink Community
- Further Aspects
  - Fault Tolerance
  - Declarative Languages

StratoSphere
Above the Clouds

Apache Flink – Big Data Batch and Stream Processing

Alexandrov, R. Bergmann, S. Ewen, J. C. Freytag, F. Hueske, A. Heise, O. Kao,
M. Leich, U. Leser, V. Markl, et al, "The Stratosphere platform for big data analytics,"
VLDB J. , vol. 23, no. 6, 2014
http://flink.apache.org
http://www.stratosphere.eu

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

14

Timeline

2008
Initial vision for a big data analytics platform

2009
DFG Proposal for Stratosphere I

2010
Grant Award
Start of Stratosphere I

2010
Nephele / PACTs paper published

2011
DFG Proposal for Stratosphere II

2012
Spinning Fast Iterative Dataflows paper published
VLDB

2012
Grant Award
Start of Stratosphere II

2014
Apache Flink Incubator Project

The VLDB Journal
Stratosphere System paper published

2015
Apache Flink Top Level Project

2015
dataArtisans Founded

2015
1st Flink Forward Conference
Flink Forward BERLIN 12/13 OCT 2015

2016
Flink Community Groups Across Europe

Berlin	758
Paris	500
Madrid	384
Stockholm	313
Brussels	279
London	190

2016
21 Meetups Worldwide

186 Contributors
6326 Members
20 Cities
12 Countries

2016
FlinkForward
2nd Flink Forward Conference September 2016
flink-forward.org

What is Apache Flink?

Apache Flink is an open source platform for scalable batch and stream data processing.

A distributed system that you can use to process data

Like a DBMS but not exactly a DBMS

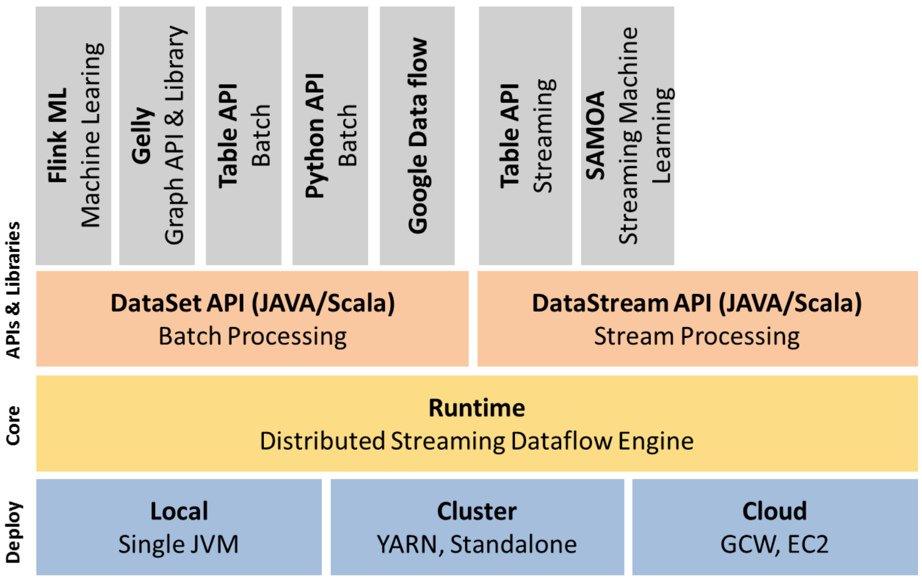
What kind of data?

Data that comes in the form of streams

What kind of processing

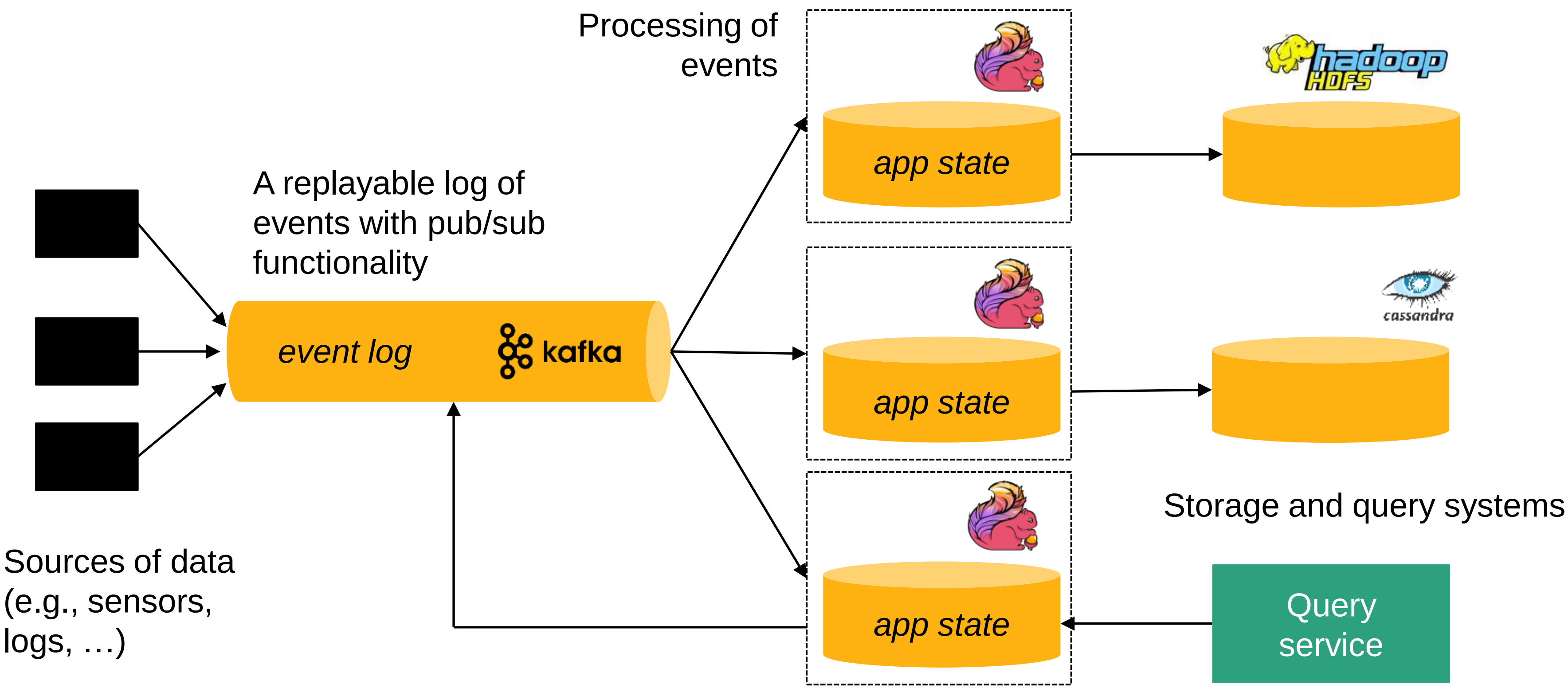
Quite flexible. You can use Java/Scala APIs similar to programming with Java collections, the new SQL API, etc

Distributed: runs on many (1000s) of machines and hides this complexity from the user



http://flink.apache.org

Basic application architecture

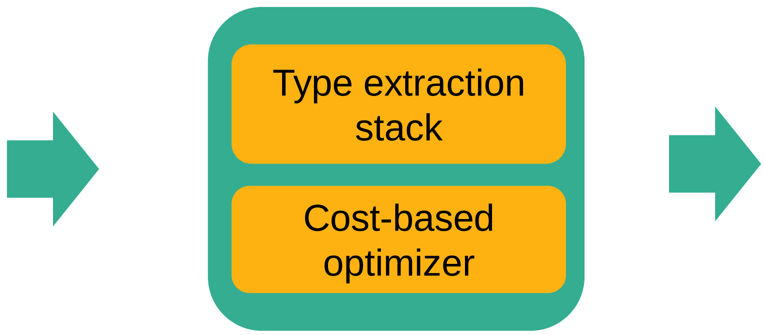


By courtesy of Kostas Tzoumas

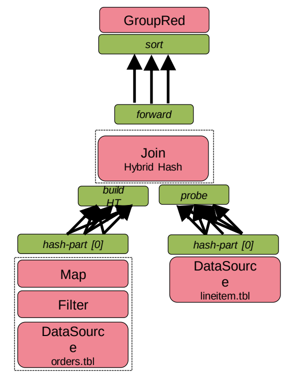
Technology inside Flink

```
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
  }
```

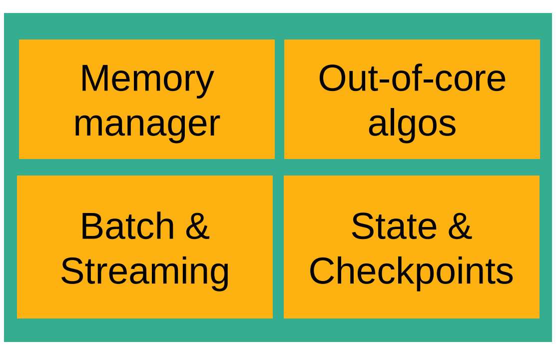
Program

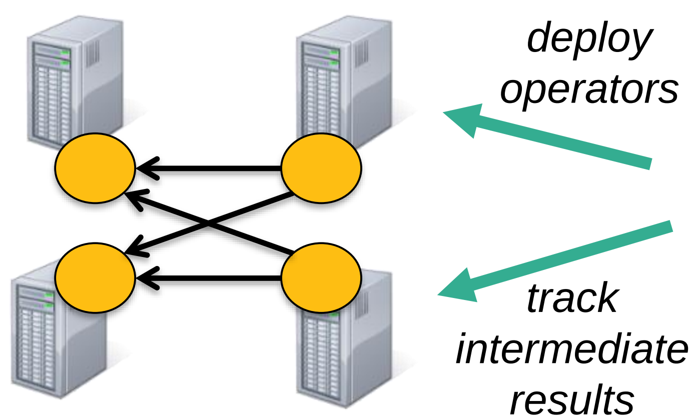


Pre-flight (Client)

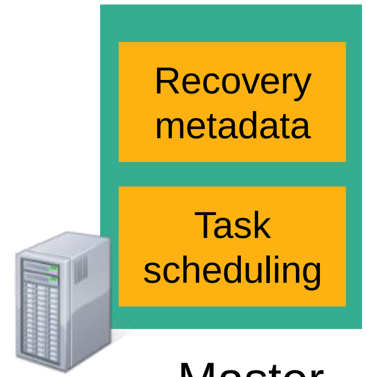


Dataflow Graph



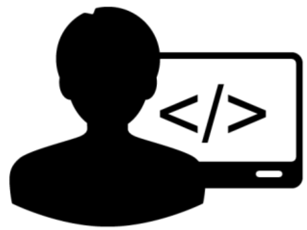


Workers



Master

Effect of optimization



Hash vs. Sort
Partition vs. Broadcast
Caching
Reusing partition/sort

Execution
Plan A

Execution
Plan B

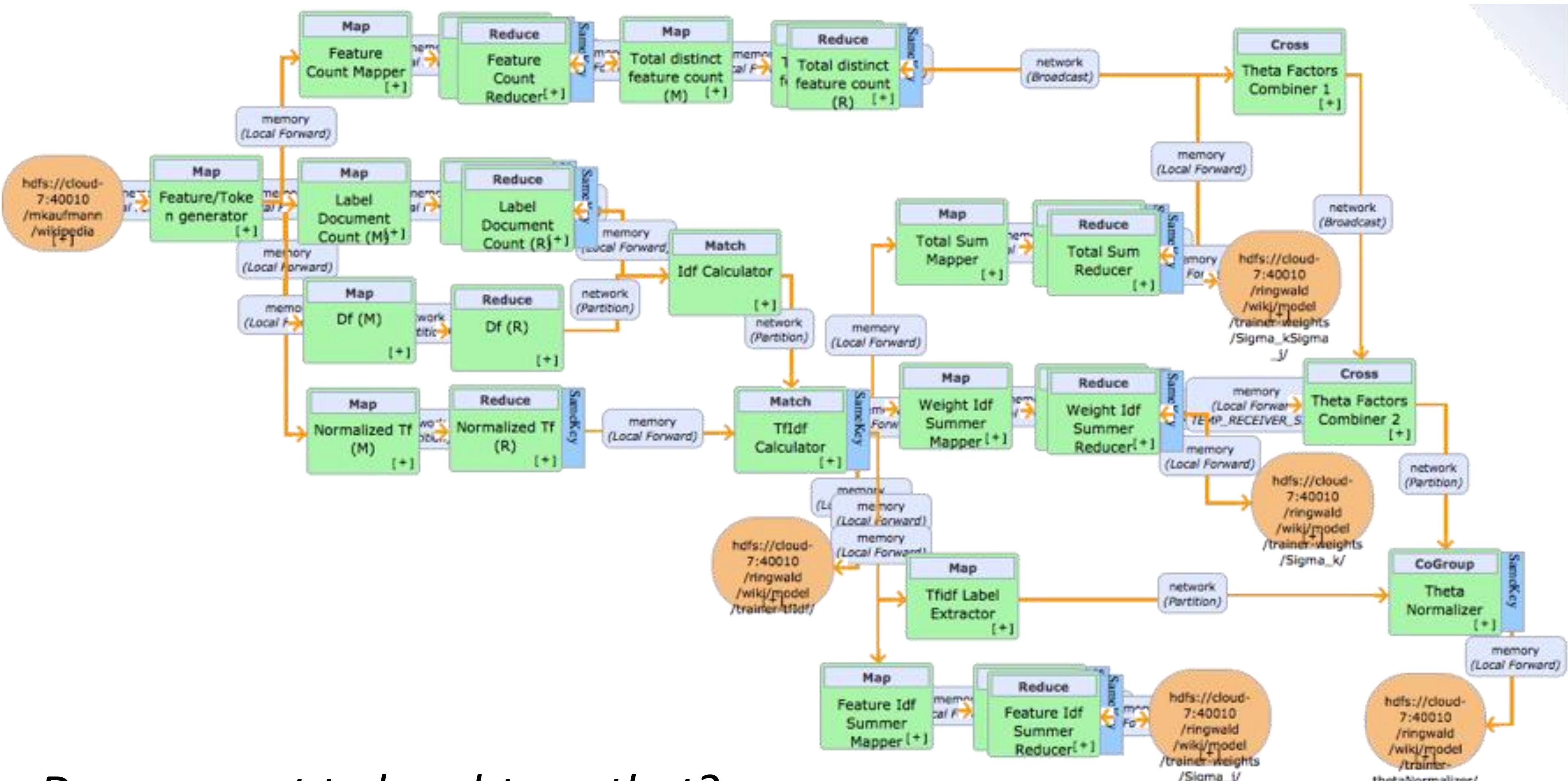
Execution
Plan C

Run on a sample
on the laptop

Run on large files
on the cluster

Run a month later
after the data evolved

Why optimization ?

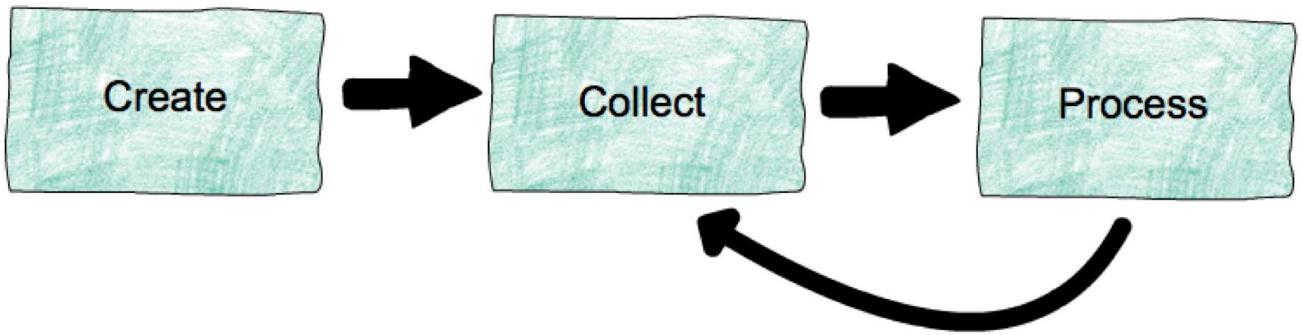


Do you want to hand-tune that?

P. Carbone, A. Katsifodimos, A. Ewen, V. Markl and e. al:, "Apache Flink™: Stream and Batch Processing in a Single Engine.," IEEE Data Eng. Bull., vol. 38, no. 4, pp. 28-38, 2015

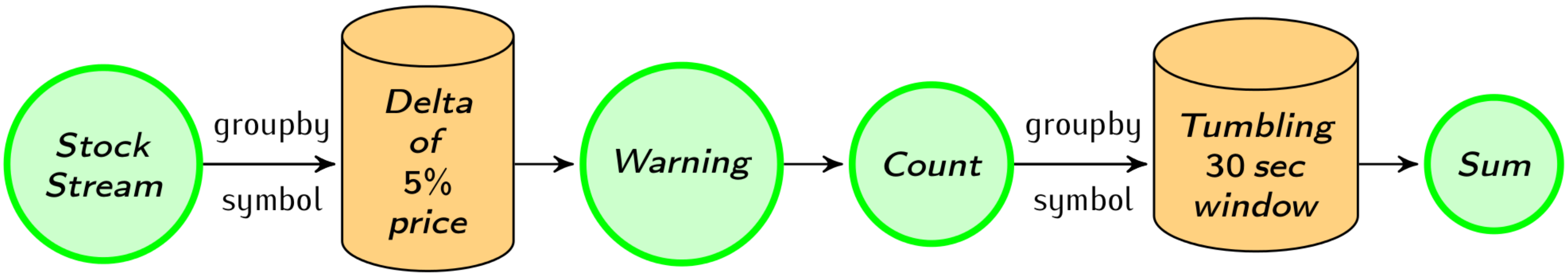
DATA STREAM ANALYSIS

Life of data streams



- **Create:** create streams from event sources (machines, databases, logs, sensors, …)
- **Collect:** collect and make streams available for consumption (e.g., Apache Kafka)
- **Process:** process streams, possibly generating derived streams (e.g., Apache Flink)

Stream Analysis in Flink




```scala
case class Count(symbol: String, count: Int)
val defaultPrice = StockPrice("", 1000)

//Use delta policy to create price change warnings
val priceWarnings = stockStream.groupBy("symbol")
  .window(Delta.of(0.05, priceChange, defaultPrice))
  .mapWindow(sendWarning _)

//Count the number of warnings every half a minute
val warningsPerStock = priceWarnings.map(Count(_, 1))
  .groupBy("symbol")
  .window(Time.of(30, SECONDS))
  .sum("count")
```

More at: http://flink.apache.org/news/2015/02/09/streaming-example.html

Defining windows in Flink

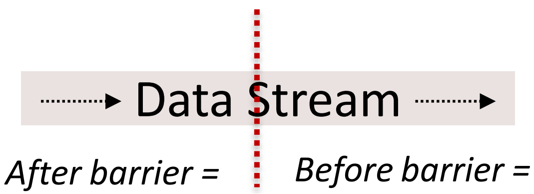
- Trigger policy
  - When to trigger the computation on current window
- Eviction policy
  - When data points should leave the window
  - Defines window width/size
- E.g., count-based policy
  - evict when #elements > n
  - start a new window every n-th element
- Built-in: Count, Time, Delta policies

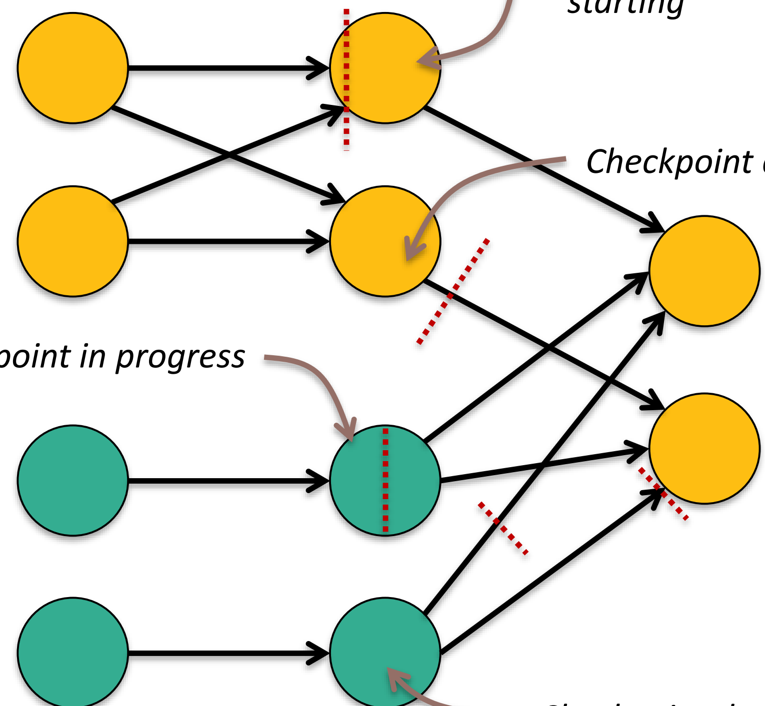
Checkpointing / Recovery

- 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





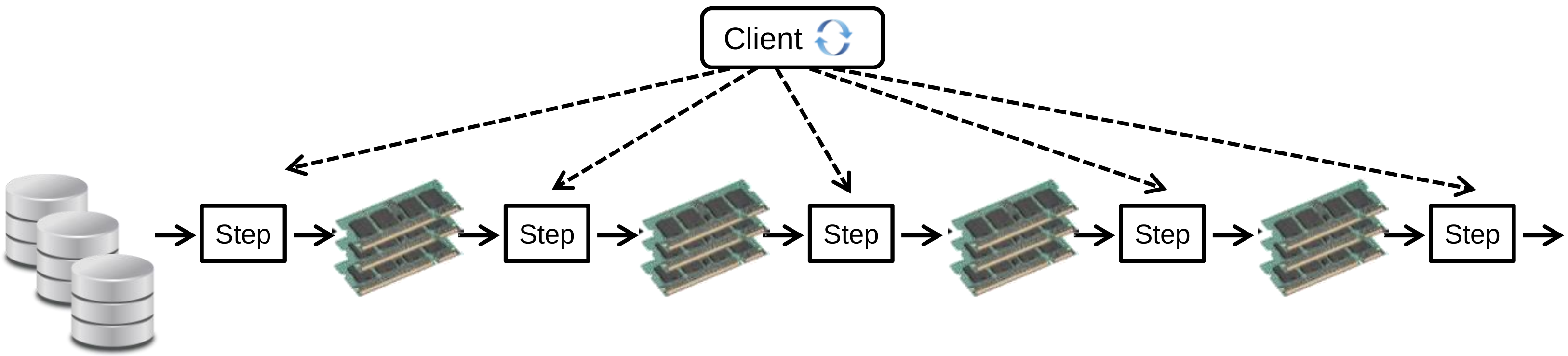
Chandy-Lamport Algorithm for consistent asynchronous distributed snapshots

26

S. Ewen, K. Tzoumas, M. Kaufmann and V. Markl, "Spinning Fast Iterative Data Flows.," PVLDB, vol. 5, no. 11, pp. 1268-1279, 2012.

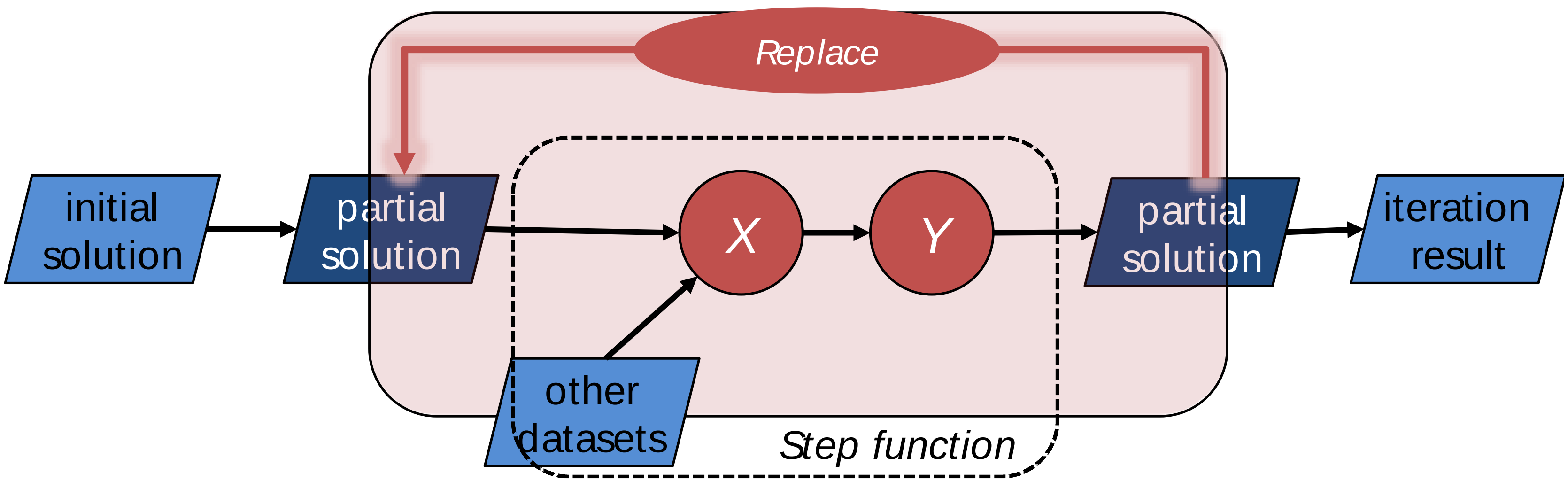
ITERATIONS IN DATA FLOWS → MACHINE LEARNING ALGORITHMS

Iterate by looping



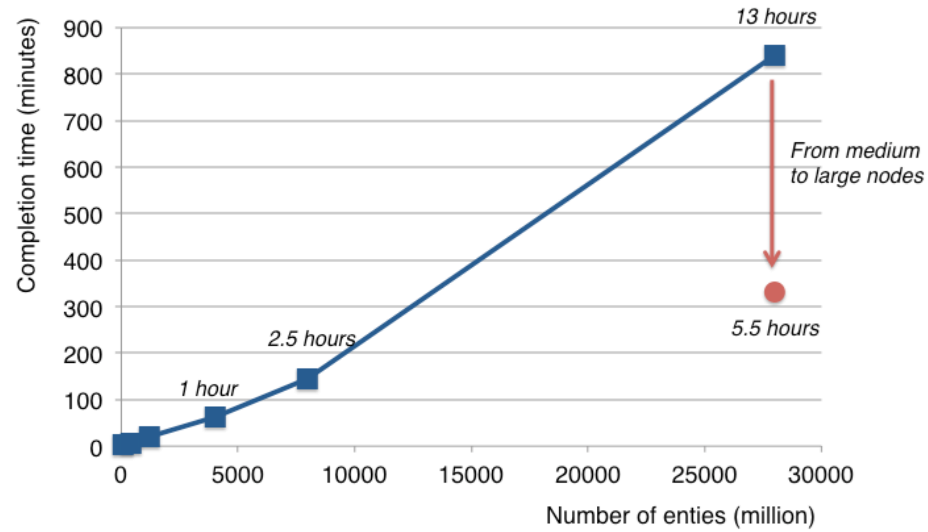
- for/while loop in client submits one job per iteration step
- Data reuse by caching in memory and/or disk

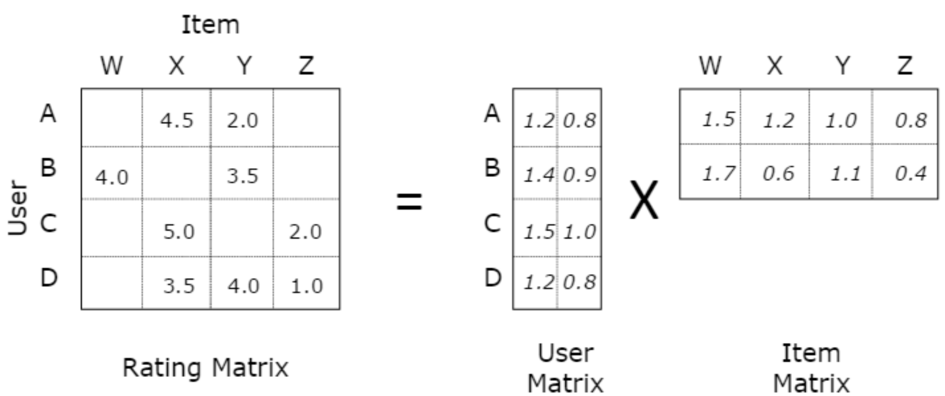
Iterate in the Dataflow



Large-Scale Machine Learning

Factorizing a matrix with 28 billion ratings for recommendations





(Scale of Netflix or Spotify)

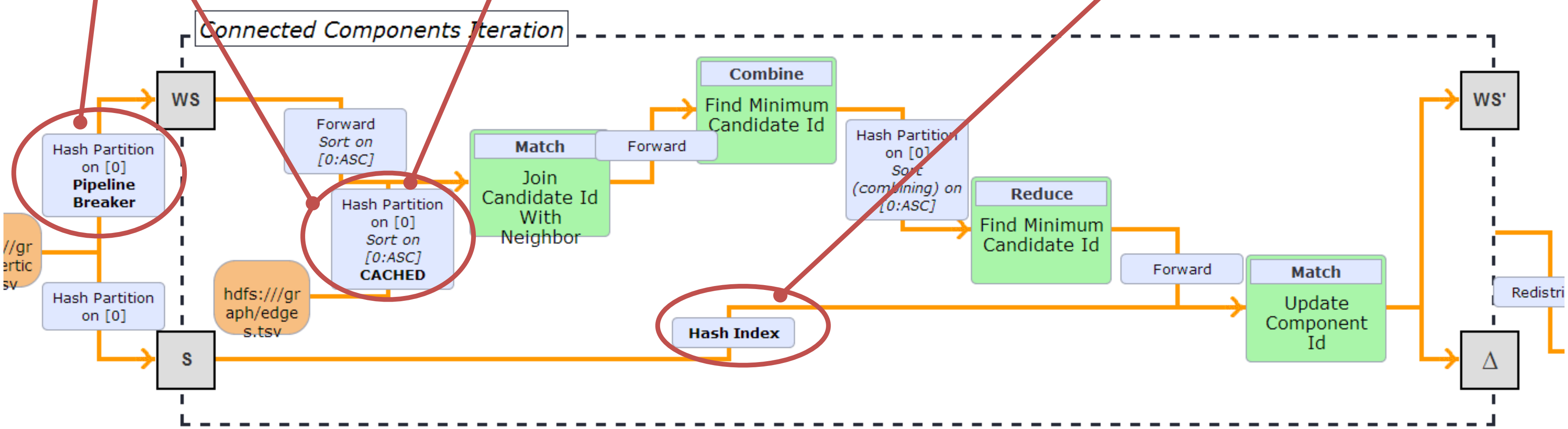
More at: http://data-artisans.com/computing-recommendations-with-flink.html

Optimizing iterative programs

Pushing work „out of the loop“

Caching Loop-invariant Data

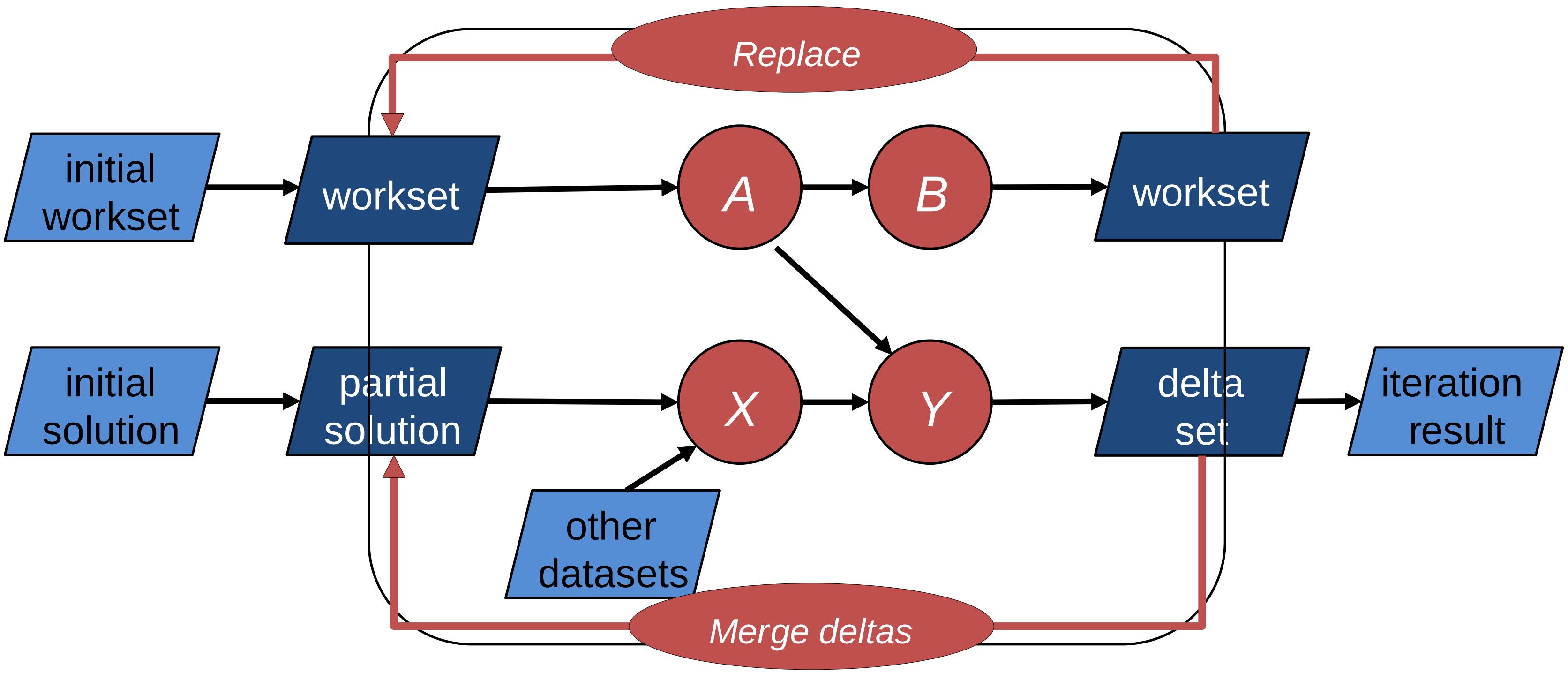
Maintain state as index



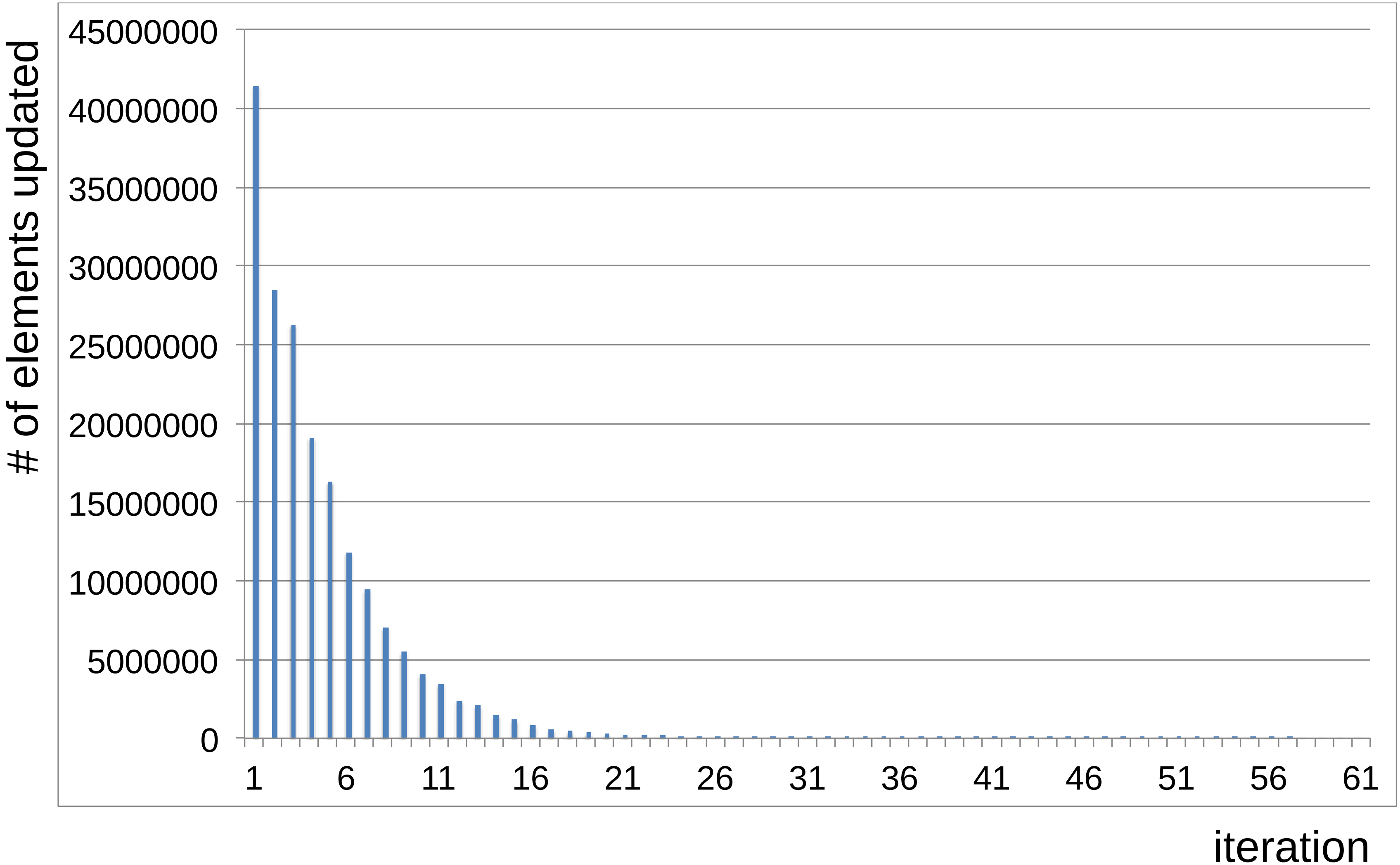
S. Ewen, K. Tzoumas, M. Kaufmann and V. Markl, "Spinning Fast Iterative Data Flows.," PVLDB, vol. 5, no. 11, pp. 1268-1279, 2012.

STATE IN ITERATIONS → GRAPHS AND MACHINE LEARNING

Iterate natively with deltas

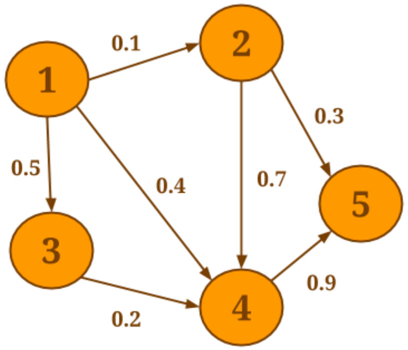


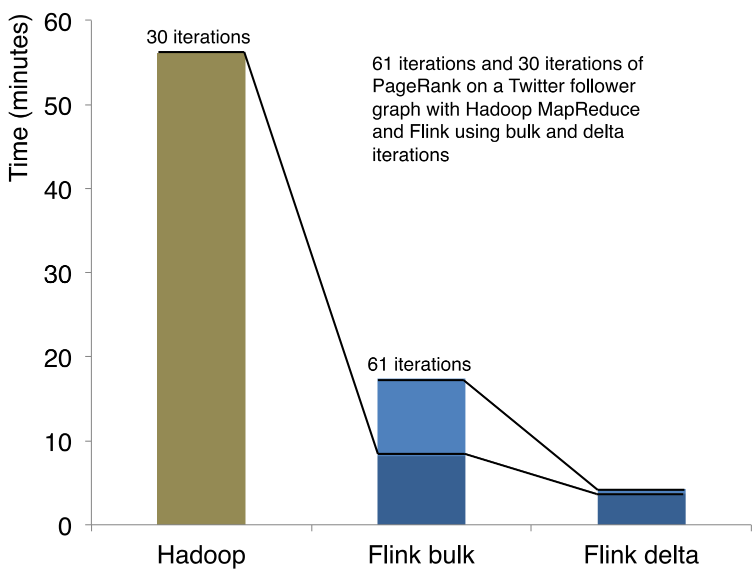
Effect of delta iterations…



… very fast graph analysis

Performance competitive with dedicated graph analysis systems

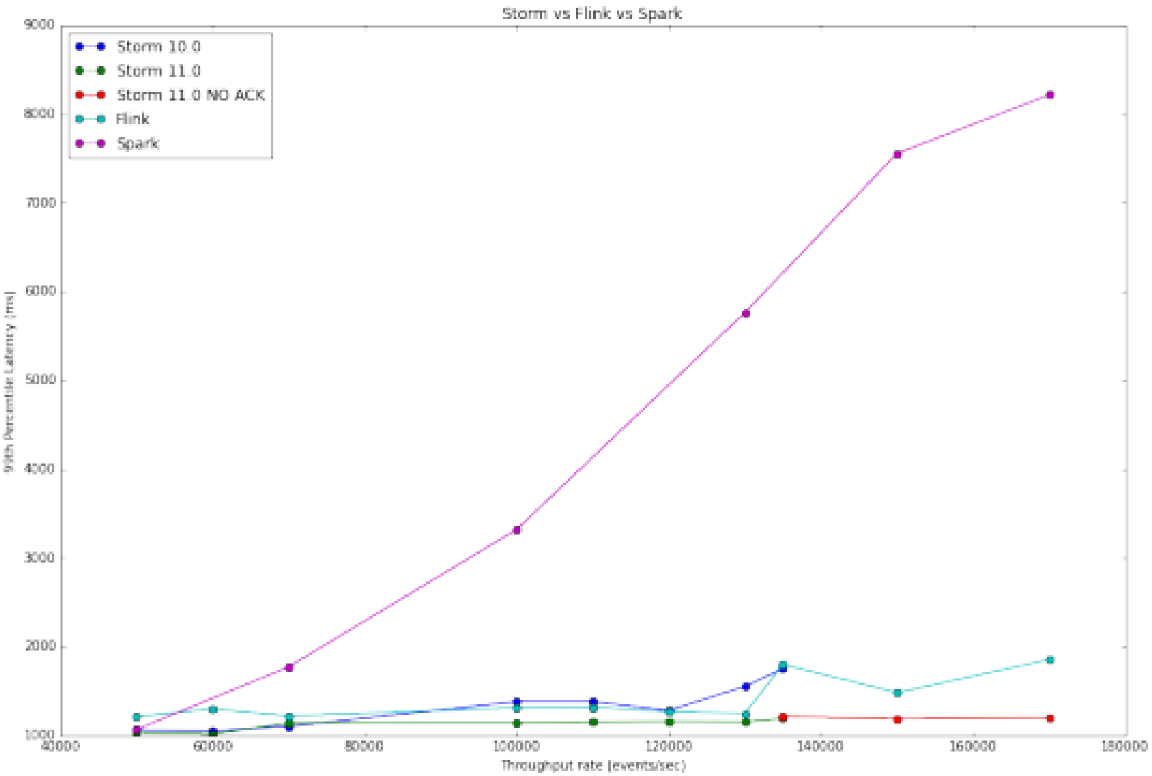




… and mix and match ETL-style and graph analysis in one program

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

A Benchmark Result

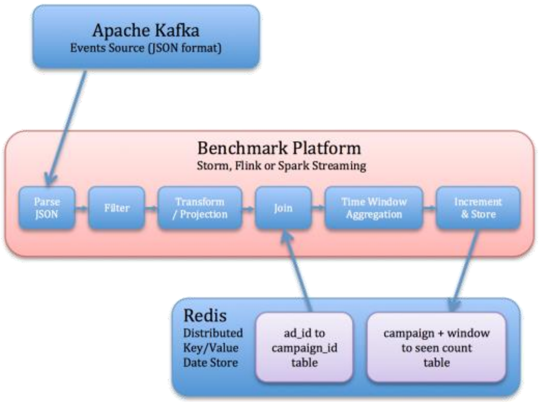


Source: http://yahooeng.tumblr.com/post/135321837876/benchmarking-streaming-computation-engines-at

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.

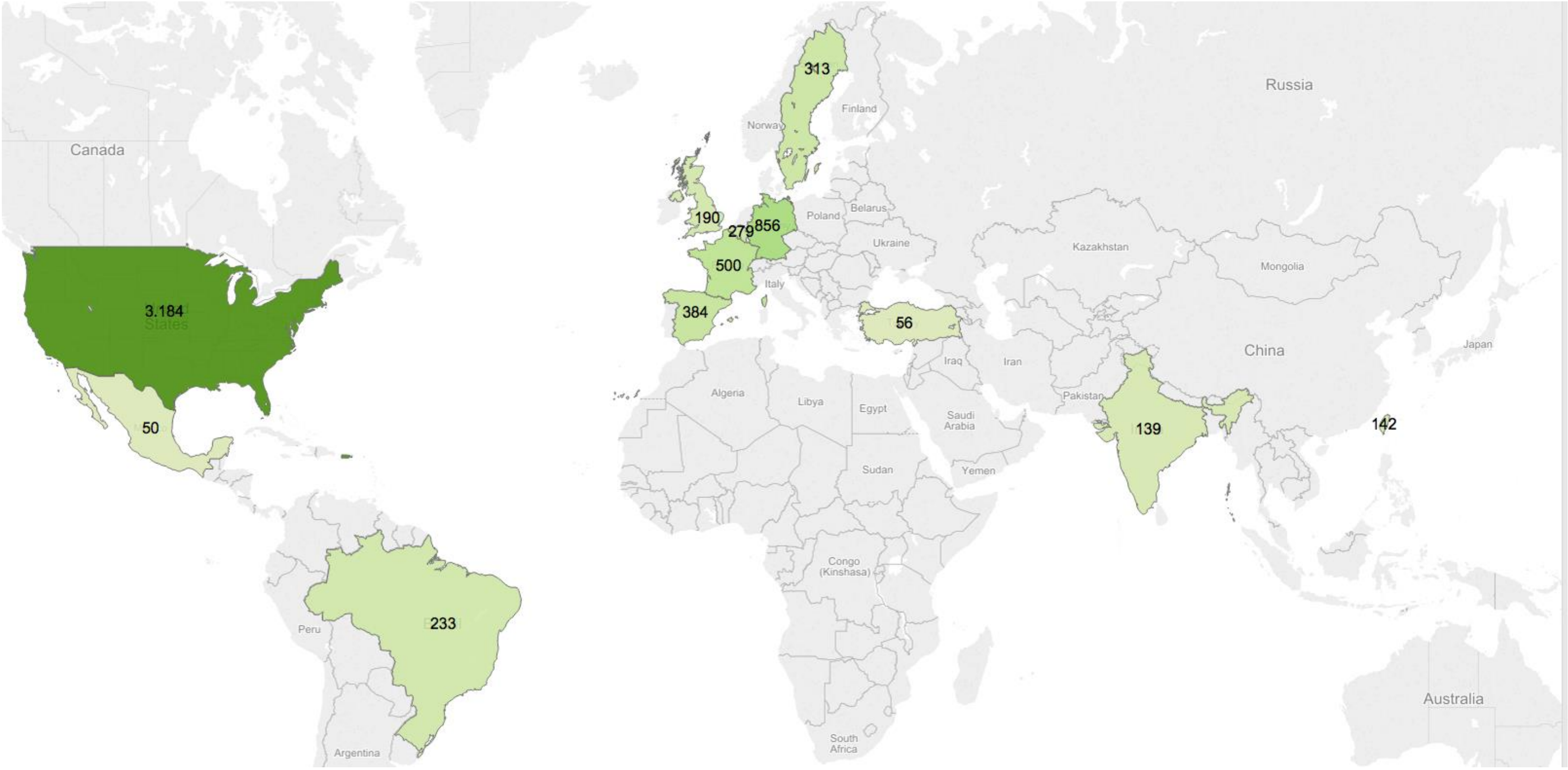
Flink achieves highest throughput with competitive low latency!



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

THE FLINK COMMUNITY

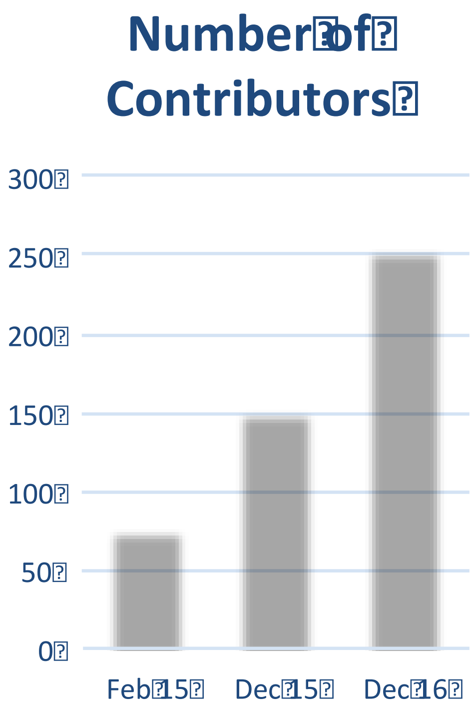
The Flink Community:
Meetups By Country Concerning Flink

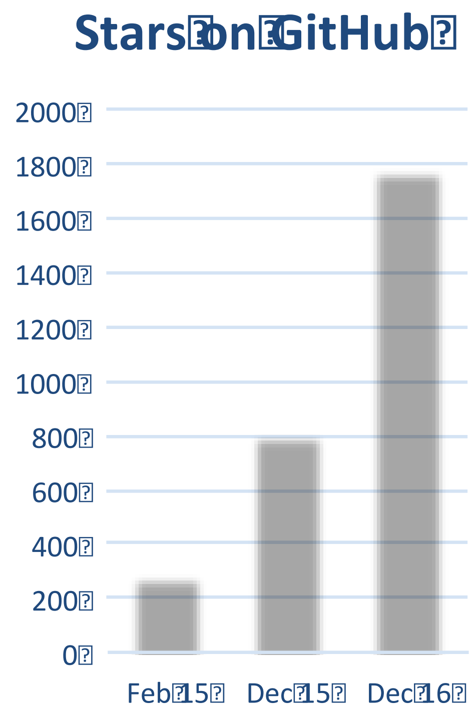


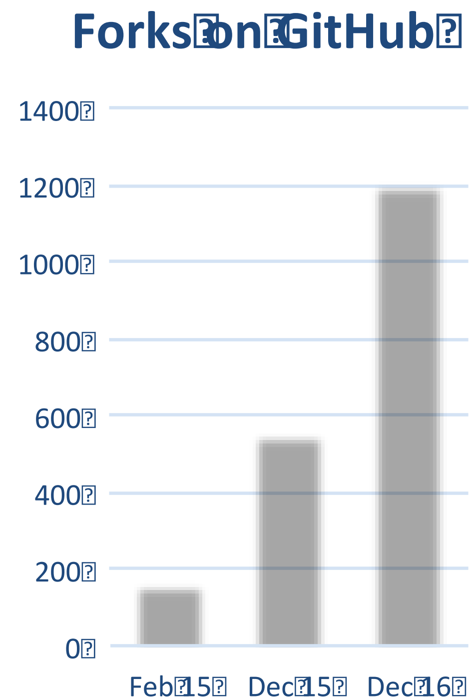
Apache Flink Meetups Worldwide (Data accurate as of **30.5.16**)
6326 members *strictly focused on* Apache Flink (comprising 57%)
4771 members *broader in scope*, including Flink (comprising 43%)

Flink community (2)

- More than 250 people have contributed code to Flink

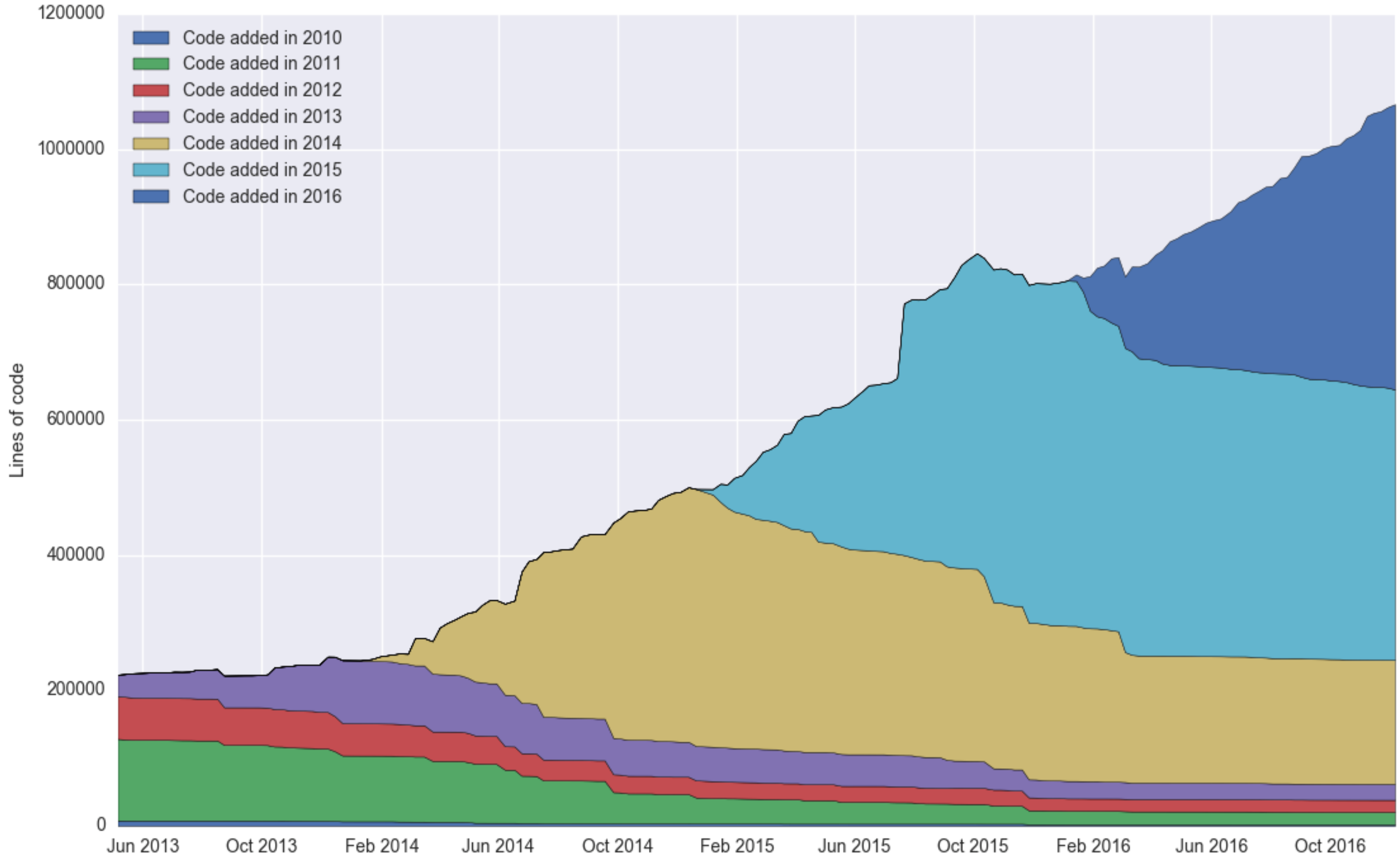






By courtesy of Kostas Tzoumas

Code survival in Flink



By courtesy of Kostas Tzoumas

Powered by Flink

Zalando, one of the largest ecommerce companies in Europe, uses Flink for real-time business process monitoring.

King, the creators of Candy Crush Saga, uses Flink to provide data science teams with real-time analytics.

Alibaba, the world's largest retailer, built a Flink-based system (Blink) to optimize search rankings in real time.

Bouygues Telecom uses Flink for real-time event processing over billions of Kafka messages per day.

By courtesy of Kostas Tzoumas

> 20 Companies Using Flink

Alibaba.com
bouygues TELECOM
OKKAM
ERICSSON
Pragsis Bidoop Big Data Analytics
treelogic
CapitalOne
ResearchGate
King
mbrtargeting
zalando
otto group
EURA NOVA
RADICALBIT Deep, Different, Disruptive.
FIRELAYERS
FIRELAYERS
Atos
LINE
dataArtisans
PT
amadeus

> 8 Software Projects Using Flink

Google Cloud Platform

CLOUD DATAFLOW
A fully-managed cloud service and programming model for batch and streaming big data processing.

CASCADING

Apache Flink is a replacement for MapReduce to support large-scale batch workloads and streaming data flows. It eliminates the concept of mapping and reducers and leverages in-memory storage, resulting in significant performance gains over MapReduce.

Apache SAMOA
Scalable Advanced Massive Online Analysis

Apache incubator

Apache SAMOA is a distributed streaming machine learning (ML) framework that contains a programing abstraction for distributed streaming ML algorithms.

Apache Streams™

mahout

The Apache Mahout™ project's goal is to build an environment for quickly creating scalable performant machine learning applications.

DMLC

Apache MRQL

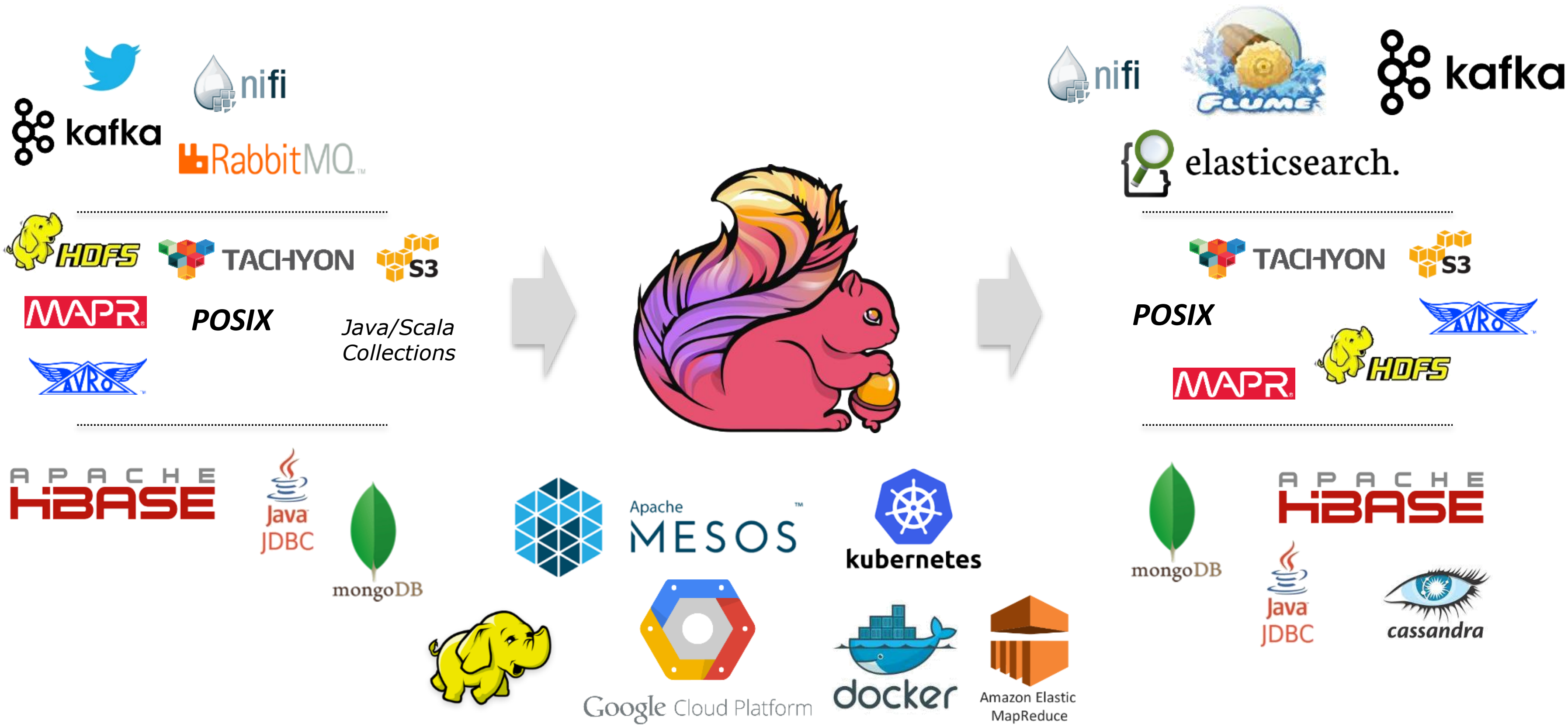
MRQL is a query processing and optimization system for large-scale, distributed data analysis, built on top of Apache Hadoop, Hama, Spark, and Flink.

beam

Apache Beam is an open source, unified programming model that you can use to create a data processing **pipeline**.

> 10 Research Institutions Using Flink

Flink in the ecosystem

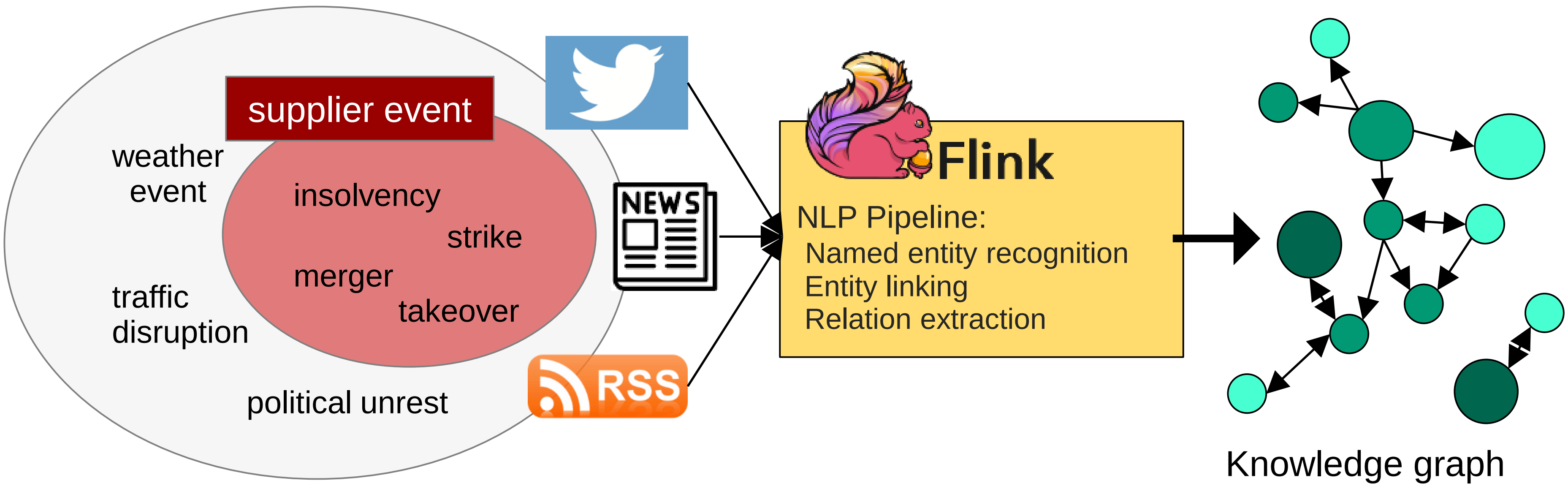


45

By courtesy of Kostas Tzoumas

Smart Data Web

Scenario
How can we detect problems, risks and potential bottlenecks in the supply chain based on the knowledge derived from natural language data?

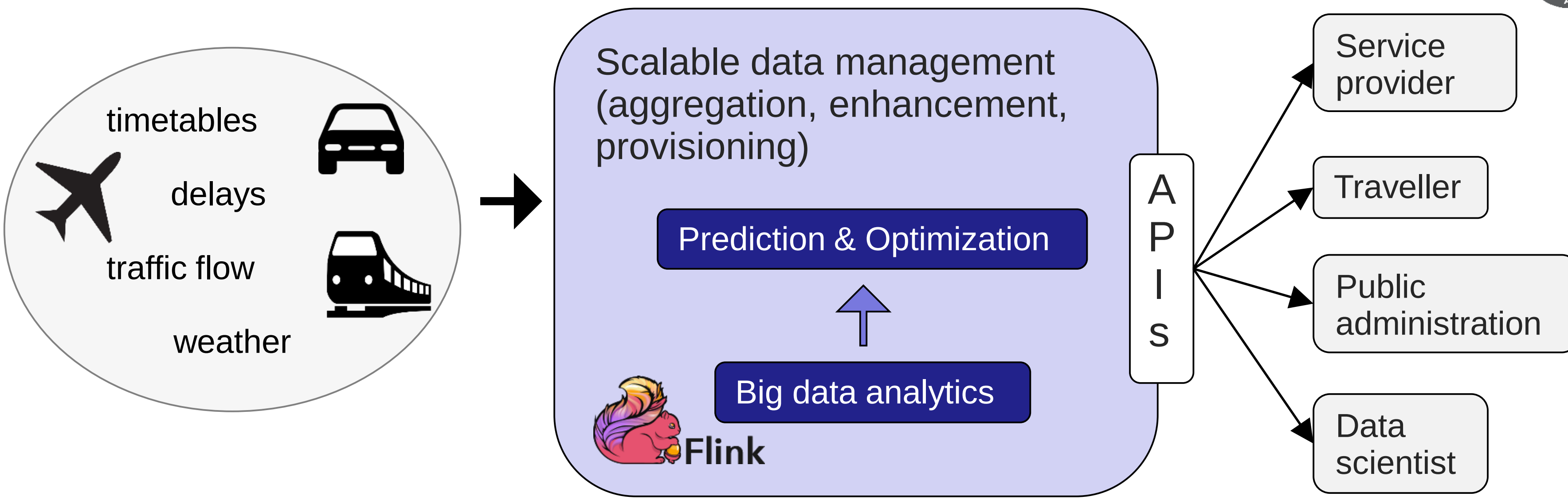


Results
Knowledge graph for supply chain management
- continously updated with relevant facts and events using parallel analysis of textual data streams
- linking company-internal with open knowledge

BMWi Project “Smart Data Web”: http://www.smartdataweb.de/

Smart Data for Mobility

Scenario
How do we aggregate and make use of mobility data from multiple modalities (air, road, railway)?



Results
- Big data analytics platform providing data and smart mobility services
- Prediction based on historical and real-time data considering a wide range of mobility-related aspects

BMWi Project “Smart Data For Mobility”: http://www.sd4m.net

SePiA.Pro

DAIMLER

Foto: dpa

<u>Scenario</u>

- Service platform for industrial data analytics services for **production chain optimization**.
- Improvement of production planning and execution
- Cross customer analysis of machines from the perspective of a single machine manufacturer.

POWERED BY APACHE FLINK

<u>Data Sources</u>

Machinery park

Actuators

Sensors

Streaming

<u>Flink Solution</u>

Data Science Component

Industrial Analytics Platform

<u>Output</u>

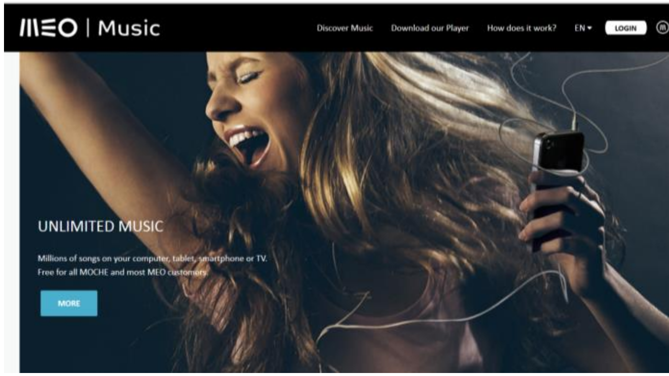
- Parameters for production chain optimization
- Smart Prediction models
- Cross customer analysis results

<u>Results</u>

- Scalable data analytics platform optimized for industrial settings
- Smart services (data, algorithms, structures, policies) for automatic provisioning

BMWi Project SePiA.Pro

STREAMLINE



Scenario

- Transactions related with user actions on content are considered on the next day for new content recommendations.
- The goal is to do context aware content recommendation in real time.

Data Sources

User actions:
- download
- like
- skip

Flink Solution

Pre-processing data

MUSIC RECOMMENDATION ENGINE

Output

Real time music recommendation

Results

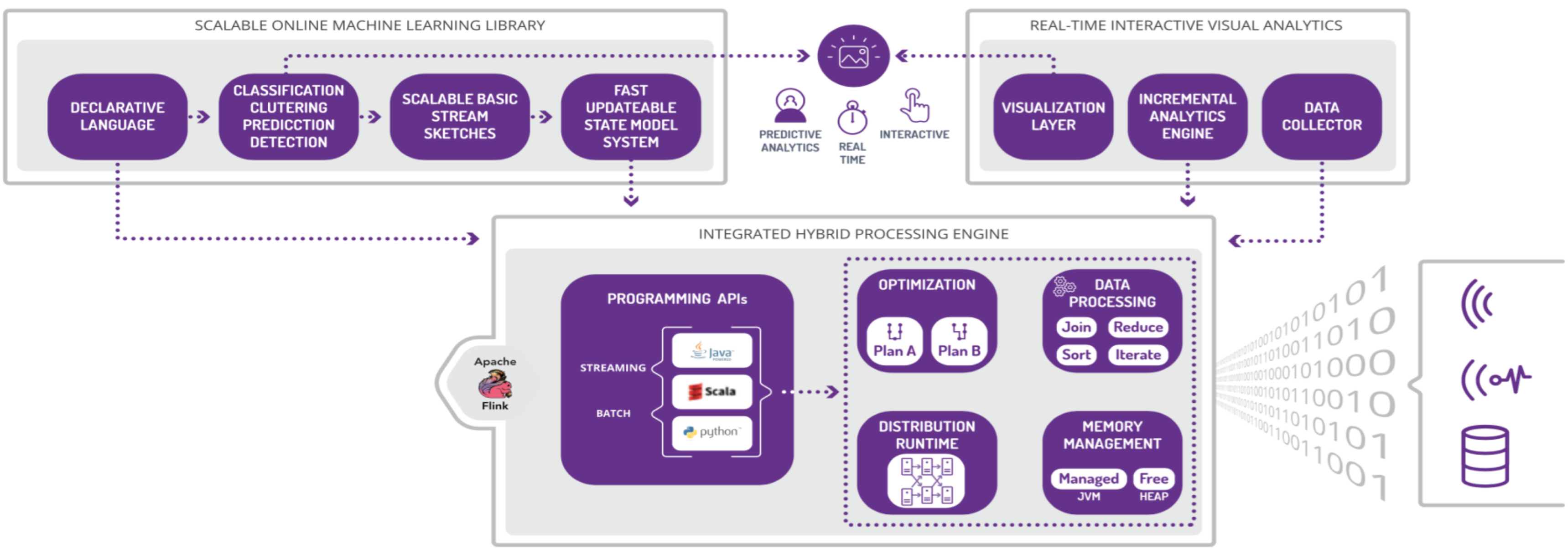
- Expected increase revenue by 25%
- Reduce time to recommend new content, move from daily batch to real time
- Reduce costs with manual back-office processes such as manually curating content; cost reduction by 60%
- Increase the total number of users consuming (clicking on) recommended content per day; increase by 50%

STREAMLINE, Horizon *2020, Ref:* 688191

PROTEUS

ArcelorMittal

Scenario
- Defects introduced in early processes of steel production have a great economic impact due to the costs
- The sooner defects are detected, the sooner the process can be modified in order to stop producing defective subsequent coils



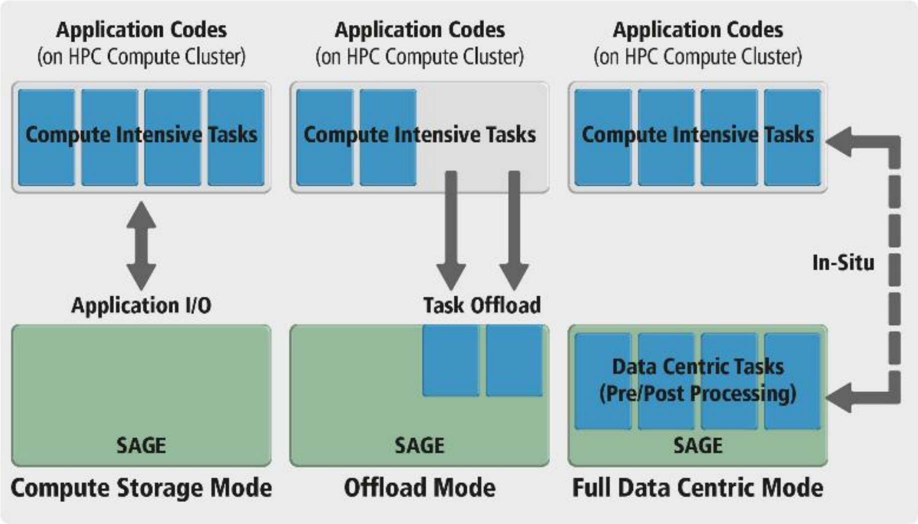
Results
- Expected to achieve a reduction of 20% of defections coils and reducing rejected material by 15%.

PROTEUS, Horizon 2020, Ref: 687691

SAGE

Scenario

In scientific research areas as vast as brain simulation, gene sequencing, and space weather forecasting, experiments and simulations generate increasingly large data sets. As these scale into the range of exabytes (billions of gigabytes), novel storage, processing, and analytics solutions must be devised to continue deriving insights and innovation in research.



Results

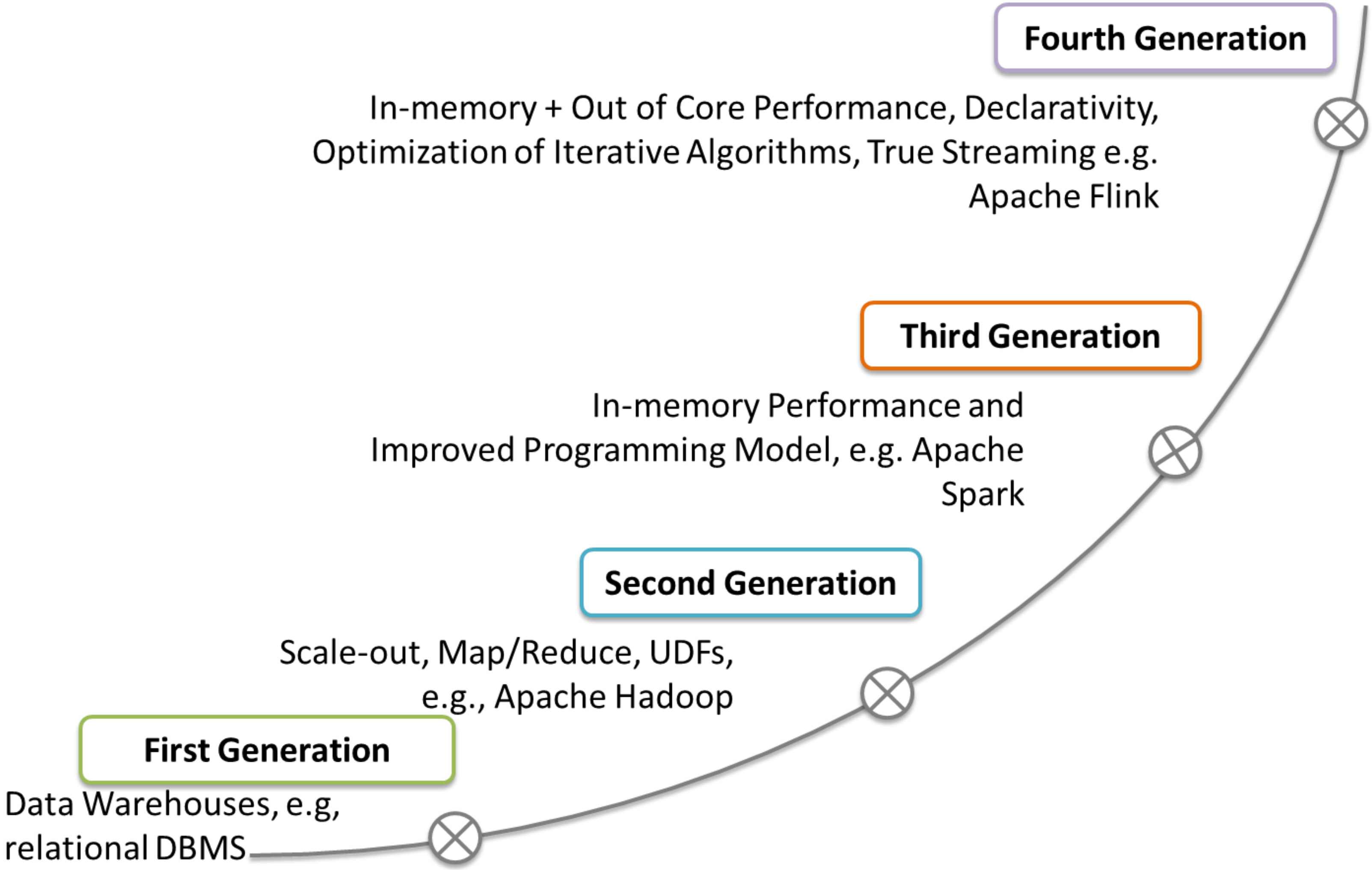
Development of a next-generation data storage system supporting current- and future-generation persistent storage media for exascale data processing. Project SAGE will integrate both current- and future-generation storage media within a multi-tiered hierarchy and introduce computational capabilities to the storage layer.

Thanks to my team members and students

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

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

Evolution of Big Data Platforms



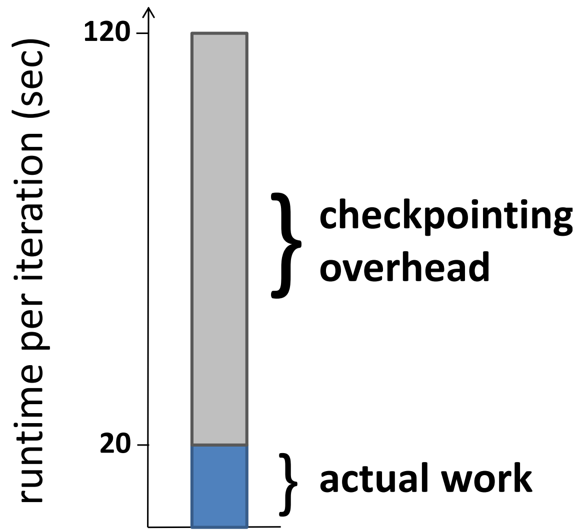
Fault tolerance

Pessimistic Recovery:

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

Problematic:

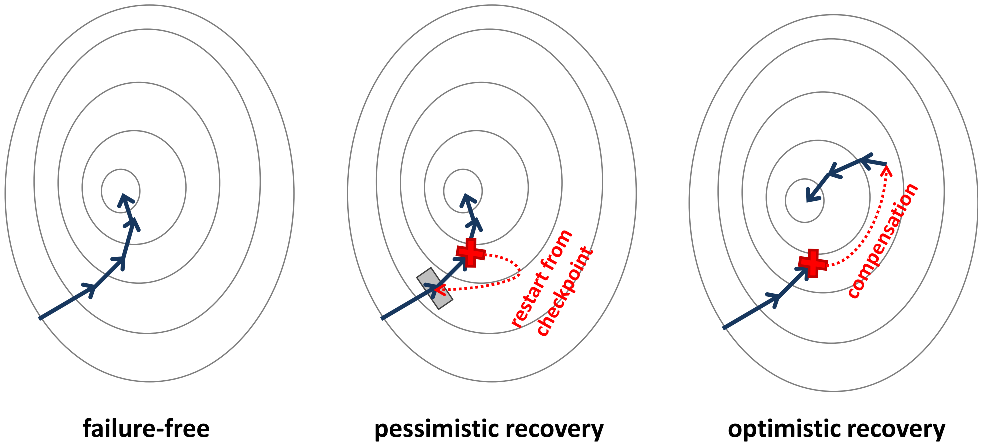
- High overhead, checkpoint must be replicated to other machines
- Overhead always incurred, even if no failures happen!



➢ **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 absense of failures**!
- algorithm-specific **compensation function** must restore state

All Roads lead to Rome

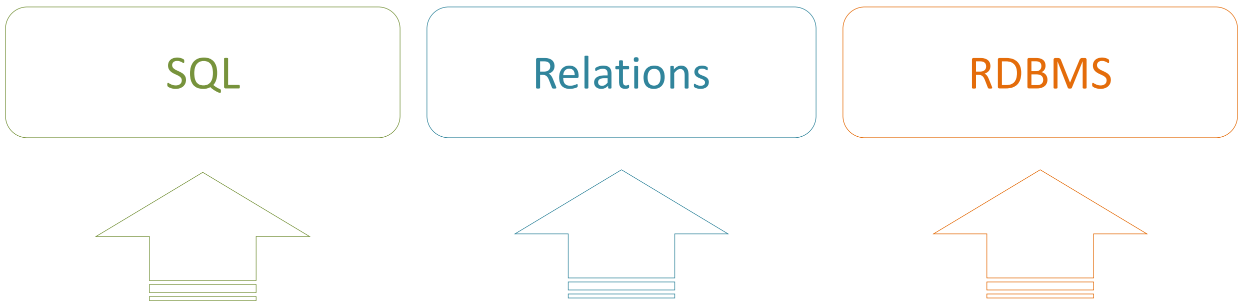
If you are interested more, read our CIKM 2013 paper:

Sebastian Schelter, Stephan Ewen, Kostas Tzoumas, Volker Markl: **"All roads lead to Rome": optimistic recovery for distributed iterative data processing**. *CIKM 2013*: 1919-1928

Sergey Dudoladov, Chen Xu, Sebastian Schelter, Asterios Katsifodimos, Stephan Ewen, Kostas Tzoumas, Volker Markl: **Optimistic Recovery for Iterative Dataflows in Action**.
To appear in *SIGMOD 2015*

Declarative Data Processing and Big Data

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...

VERTICA | IBM DB2 | SAP HANA | ORACLE DATABASE | Microsoft SQL Server

SQL | Relations | RDBMS

Declarative Data Processing

Is Being Revised

SQL | Relations | RDBMS

Declarative Data Processing

Second-Order Functions | Distributed Collections | Parallel Dataflow Engines

Mosaics of Theories and Systems

- First results
  - Alexander Alexandrov, Andreas Salzmann, Georgi Krastev, Asterios Katsifodimos, Volker Markl: Emma in Action: Declarative Dataflows for Scalable Data Analysis. SIGMOD 2016
  - Alexander Alexandrov, Asterios Katsifodimos, Georgi Krastev, Volker Markl: Implicit Parallelism through Deep Language Embedding. SIGMOD Record 45(1): 51-58 (2016)
- Next Steps
  - Open-Source Release
- Vision (Frontend): Multi-model DSL based on type contracts
  - Collection Processing *DataBag[A]*
  - Linear Algebra *Matrix[A], Vector[A]*
  - Stream Processing *Stream[A]*
- Vision (Backend): Target more execution engines
  - Column Stores
  - GPUs