Success Story Berlin Big Data Center

Prof. Dr. 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 & Analysis: Increasingly Complex!

**Data**
- data volume too large
- data rate too fast
- data too uncertain

**Analysis**
- Volume
- Velocity
- Variability
- Veracity

Reporting
- Ad-Hoc Queries
- ETL/ELT

aggregation, selection
- SQL, XQuery
- MapReduce

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

scalability
- algorithms

scalability
“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

New Technology to the Rescue!
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

R/Matlab: 5 million users

Hadoop: 200,000 users

“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

Declarative languages to the rescue!

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

Many new companies and products are emerging to enable the deep analysis of big data. **Strong European contenders**, include Apache Flink, Parstream, and Exasol. **“New companies“ are the (b)leading users of these technologies.** For example, in the information economy (e.g., Zalando, Amazon, ResearchGate, SoundCloud, & Spotify). **“Traditional large companies“ are following** and still developing strategies (e.g., Industrie 4.0, Logistics, Telco). Most **SMEs are not yet ready to capitalize on Big Data.**
Apache Flink—A Success Story Originating in Berlin
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
What is Apache Flink?

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

- The core of Flink is a distributed streaming dataflow engine.
  - Executing dataflows in parallel on clusters
  - Providing a reliable foundation for various workloads
- **DataSet** and **DataStream** programming abstractions are the foundation for user programs and higher layers
Technology Embedded in Flink

```scala
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

Dataflow Graph

Memory Manager | Out-of-core Algorithms
Batch & Streaming | State & Checkpoints

Workers

Deploy operators

track intermediate results

Recovery Metadata

Task Scheduling

Master
Current Benchmark Results

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!

Flink Community

#unique contributor ids by Git commits
(Strictly) Flink European Meetups with Member Totals (as of 30.5.16)

<table>
<thead>
<tr>
<th>Country</th>
<th>Total Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berlin</td>
<td>758</td>
</tr>
<tr>
<td>Paris</td>
<td>500</td>
</tr>
<tr>
<td>Madrid</td>
<td>384</td>
</tr>
<tr>
<td>Stockholm</td>
<td>313</td>
</tr>
<tr>
<td>Brussels</td>
<td>279</td>
</tr>
<tr>
<td>London</td>
<td>190</td>
</tr>
<tr>
<td>Munich</td>
<td>98</td>
</tr>
<tr>
<td>Istanbul</td>
<td>56</td>
</tr>
</tbody>
</table>
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%)
Distribution of (Strictly) Flink Meetup Group Members by Country (as of 30.5.16)

<table>
<thead>
<tr>
<th>Country</th>
<th>Total Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>3184</td>
</tr>
<tr>
<td>Germany</td>
<td>856</td>
</tr>
<tr>
<td>France</td>
<td>500</td>
</tr>
<tr>
<td>Spain</td>
<td>384</td>
</tr>
<tr>
<td>Sweden</td>
<td>313</td>
</tr>
<tr>
<td>Belgium</td>
<td>279</td>
</tr>
<tr>
<td>Brazil</td>
<td>233</td>
</tr>
<tr>
<td>UK</td>
<td>190</td>
</tr>
<tr>
<td>Taiwan</td>
<td>142</td>
</tr>
<tr>
<td>India</td>
<td>139</td>
</tr>
<tr>
<td>Turkey</td>
<td>56</td>
</tr>
<tr>
<td>Mexico</td>
<td>50</td>
</tr>
</tbody>
</table>
> 13 Companies Using Flink

- ResearchGate
- bouygues
- KKAM
- CapitalOne
- amadeus
- king
- treelogic
- otto group
- RadicalBit
- EuryNovo
- FireLayers
- dataArtisans
> 6 Software Projects Using Flink

Google Cloud Platform

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

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.

CASCADING

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

Apache SAMOA
Scalable Advanced Massive Online Analysis

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

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.

Apache 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
Integration (A Representative List)
Is Apache Flink Europe’s Wild Card into the Big Data Race?

How an ultra-fast data engine for Hadoop could secure Europe’s place in the future of open-source

**In Yahoo!’s tests, Spark’s results were considerably worse than Flink and Storm’s, going up to 70 sec without back-pressure and 120 sec with back-pressure, compared with less than 1 sec for Flink and Storm**

The cards are dealt anew!

https://medium.com/chasing-buzzwords/is-apache-flink-europes-wild-card-into-the-big-data-race-a189fcf27c4c
BBDC - Climbing to the Next Level
Machine Learning + Data Management = X

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

Think ML-algorithms in a scalable way
declarative

Process iterative algorithms in a scalable way

declarative

Goal: Data Analysis without System Programming!

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

Declarative Languages
Automatic Adaption
Scalable processing

Parallelization
Compiler
Memory Management
Memory Hierarchy
Data Analysis Language
Query Optimization
Dataflow
Indexing
What, Not How! Consider K-means Clustering.

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

65 lines of code
short development time
robust runtime

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

486 lines of code
long development time
non-robust runtime

"What"

(Apache Flink)
(Scala frontend)

"How"

(Hadoop)
Big Data Analytics Without Systems Programming! (What, Not How!)

Description of "What?"
(declarative specification)

Data Analyst

Larger human base of "data scientists"
Reduction of "human" latencies
Cost reduction

Description of "How?"
(state of the art in scalable data analysis)
Map/Reduce, MPI

Machine
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

Declarative Data Processing

SQL  Relations  RDBMS

Second-Order Functions  Distributed Collections  Parallel Dataflow Engines
Linguistic Reuse

• Reuse host language constructs in the DSL
  – simplified learning curve, better adoption

• Examples
  – variables, assignments
  – primitive types, tuple types
  – lambda terms, second-order functions

```scala
val q = "What is the ultimate answer?" // of type 'String'
val a = 42.0 // of type 'Int'
val t = (q, a)                         // of type '(String, Int)'
val X = env parallelize Seq(t)        // of type 'RDD[(String, Int)]'
val Y = X map { case(q, a) => a }      // of type 'RDD[Int]'
```
Problem

- Linguistic reuse is currently exploited by embedded DSLs to a limited extent.
- The reason for this is the shallow (i.e., pure library-based) embedding approach.

This is what the DSL compiler sees from the code snippet!
Solution

- A deep embedding approach provides more
  1. program context,
  2. opportunities for linguistic reuse and optimization.

- Feasible for host-languages with rich meta-programming facilities, such as Scala.

```scala
val q = "What is the ultimate answer?" // of type 'String'
val a = 42.0 // of type 'Int'
val t = (q, a) // of type '(String, Int)'
val X = env parallelize Seq(t) // of type 'RDD[(String, Int)]'
val Y = X map { case(q, a) => a } // of type 'RDD[Int]'
```
Comprehension Syntax

- Can be defined over **every** generic data type $T[A]$ that provides the following methods
  - $map(f: A \Rightarrow B): T[B]$
  - $flatMap(f: A \Rightarrow T[B]): T[B]$
  - $filter(p: A \Rightarrow Boolean): T[A]$

- Distributed collections, such as
  - Spark’s $RDD[A]$ and
  - Flink’s $DataSet[A]$ do!

- So we can target them from a deeply embedded Scala DSL (Emma)!
Emma: A Deeply Embedded Scala DSL for Declarative Data Analysis

• Linguistic reuse enabled by deep embedding
  – comprehensions as first-class citizens (SQL-like dataflows)
  – control flow as first-class citizen

• Coarse-grained parallelism contracts
  – based on types (Bag[A]) instead of functions (map)

• Algebraic foundations to structure, analyze, and reason about
  – program semantics
  – optimizing program transformations
Overall Vision and Next Steps

• **Vision (Frontend):** Multi-model DSL based on type contracts
  – Collection Processing  \(\text{DataBag}[A]\)
  – Linear Algebra  \(\text{Matrix}[A], \text{Vector}[A]\)
  – Stream Processing  \(\text{Stream}[A]\)

• **Vision (Backend):** Target more execution engines
  – Column Stores
  – GPUs

• **Next Steps**
  – Open-Source Release
Achievements
BBDC Research Key Achievement
ACM SIGMOD 2015 Research Highlight Award

Implicit Parallelism through Deep Language Embedding

Alexander Alexandrov
Laurel Thomsen
Andreas Kunkt
Odej Kao
 Apostolos Katsifodimos
Tobias Herb
Felix Schüler
Volker Markl
TU Berlin
firstname.lastname@tu-berlin.de

ABSTRACT
The appeal of MapReduce has spawned a family of systems
that implement or extend it. In order to enable parallel
communication processing with User-Defined Functions (UDFs),
these systems expose extensions of the MapReduce program-
ing model as language-based database APIs that are
tightly coupled to their underlying runtime engine. Explo-
ings data analysis algorithms with complex data and control
flow structures using such APIs reveals a number of limi-
tations that impede programmers’ productivity.
This paper proposes a novel data analysis language, called
DNES, that builds on existing data analysis languages and
language-based database APIs to enable scalability.

“Congratulations on the selection of your paper, Implicit Parallelism through Deep Language Embedding, for the inaugural 2015 ACM SIGMOD Research Highlight Award!”

I am writing as the Chair of the Selection Committee appointed by the SIGMOD Executive Committee. The SIGMOD Research Highlight Award is [...] to showcase a set of research projects that exemplify core database research. In particular, these projects address an important problem, represent a definitive milestone in solving the problem, and have the potential of significant impact.

[...] The selection process for the award is carefully organized and highly selective. We asked the PC chair(s) of a number of premier database conferences to nominate papers, and then the award selection committee discussed all nominated papers in great detail. [...] So congratulations on the selection of your paper!”

- Yanlei Diao, on behalf of the Selection Committee of the ACM SIGMOD Research Highlight Award
Other Research Achievements


• **Conferences**: 21 (ICDE, SIGMOD, ACL, ICDCS, ICAART, LREC, INFOCOM, PLOS ONE, OHBM)

• **Workshops**: 10 (ADMS, NLP-TEA, Parallel Tools, Workshop on Pattern Recognition and Neuroscience)

• **Demos**: 4 (SIGMOD, ACL)

• **Talks**: 40

• **Keynote Presentations**: Over 50
Apache Flink Big Data Platform and Its Community

- Project initiated at TU Berlin under the name “Stratosphere” in late 2008. In 2009, it became a DFG funded research unit, including TU Berlin, HU Berlin, and Hasso Plattner Institute researchers. Further supported by FP7, H2020, BMBF, BMWi, EIT Digital, and industry.


- Fast growing community of open source users and hundreds of developers worldwide. Used in academia (e.g., SICS/KTH, INRIA, BBDC, Big Data Europe) and companies (e.g., Amadeus, ResearchGate, Spotify, Zalando, Otto, Google, Twitter, Bouygues Telecom, and Capital One). Meetup groups all over the world, with thousands of participants.

More information: http://flink.apache.org
Visitor Highlights

03.12.2015
Eric Schmidt, Executive Chairman
Alphabet

25.02.2016
Mark Zuckerberg, Chairman & CEO
Facebook
Additional Visitors

- IBM
- ORACLE
- SAP
- T·Systems
- Huawei
- Intel
- Teradata
- Microsoft
- ParStream
- Audi
- Amazon
- Zalando
- Rolls-Royce
- Siemens
Recruited Dr. Ziawasch Abedjan, newly appointed Junior Professor at TU Berlin for Data Integration in the BBDC. Formerly, a Postdoc at MIT & Collaborator of the 2015 Turing Award winner Prof. Mike Stonebraker.

Establishment of a European Program for Data Science Education at TU Berlin, under the auspices of EIT Digital
BBDC Lessons Learned & Challenges

• International competitors have far more funding and visibility
  – Berkeley AmpLab: $30 Mio for 6 years, recently grown into Berkeley Institute for Data Science with another $40 Mio funding
  – UK Turing Institute: GBP 67 Mio funding
  – And many others (e.g., across the US, China, Korea, and Japan)

• German companies are followers, not leaders in big data
  – Many of Germany‘s large companies have not yet developed a big data strategy and are risk-averse (i.e., slow moving), or focus too much on short-term and established solutions (i.e., they lack an innovative and entrepreneurial spirit).
  – It is far easier to work with „new“ companies to transfer novel technologies
    • ResearchGate, Zalando, King, IMR, and Spotify, among others
  – Open source solutions and/or establishing new companies are the best route to turn research into innovation
  – US and large international companies are easier to collaborate with, often times via their respective German subsidiary.
Many thanks!