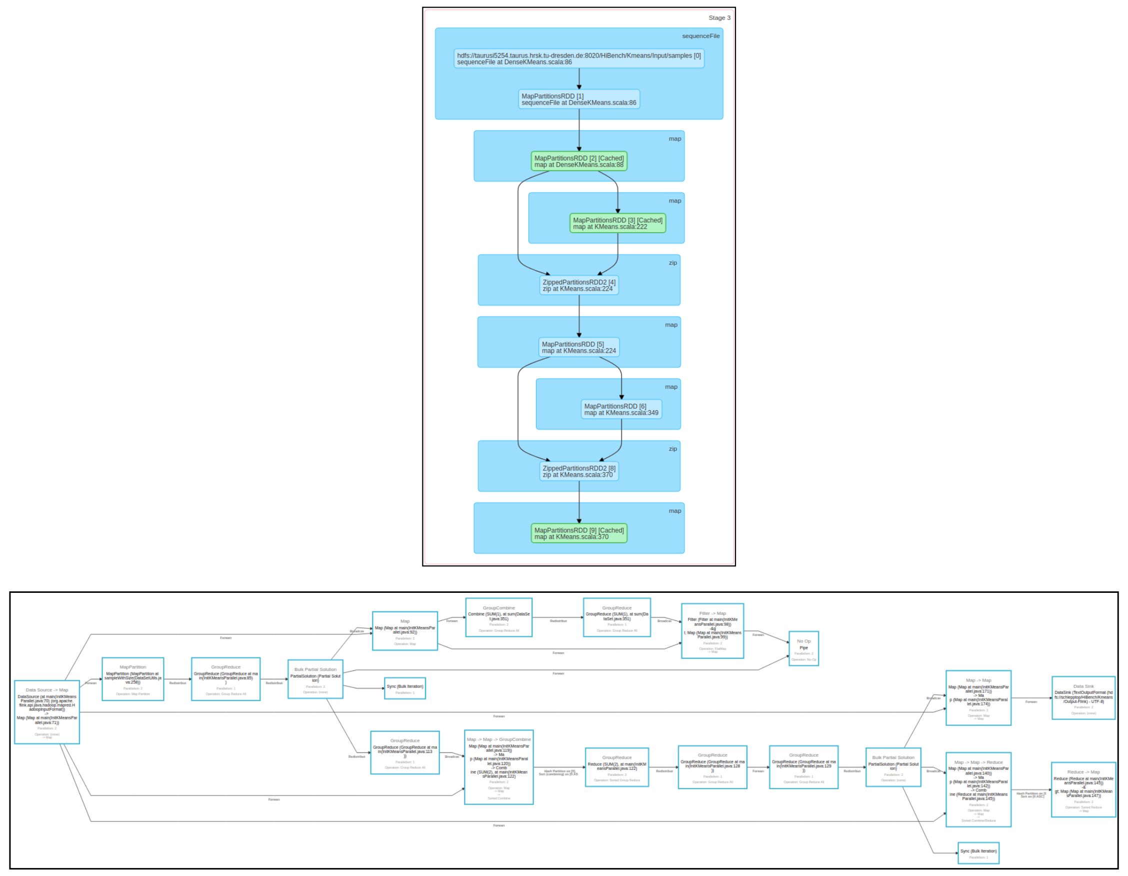
ScaDS DRESDEN LEIPZIG
www.scads.de

# Job Performance Overview of Apache Flink and Apache Spark Applications

## 1) Introduction

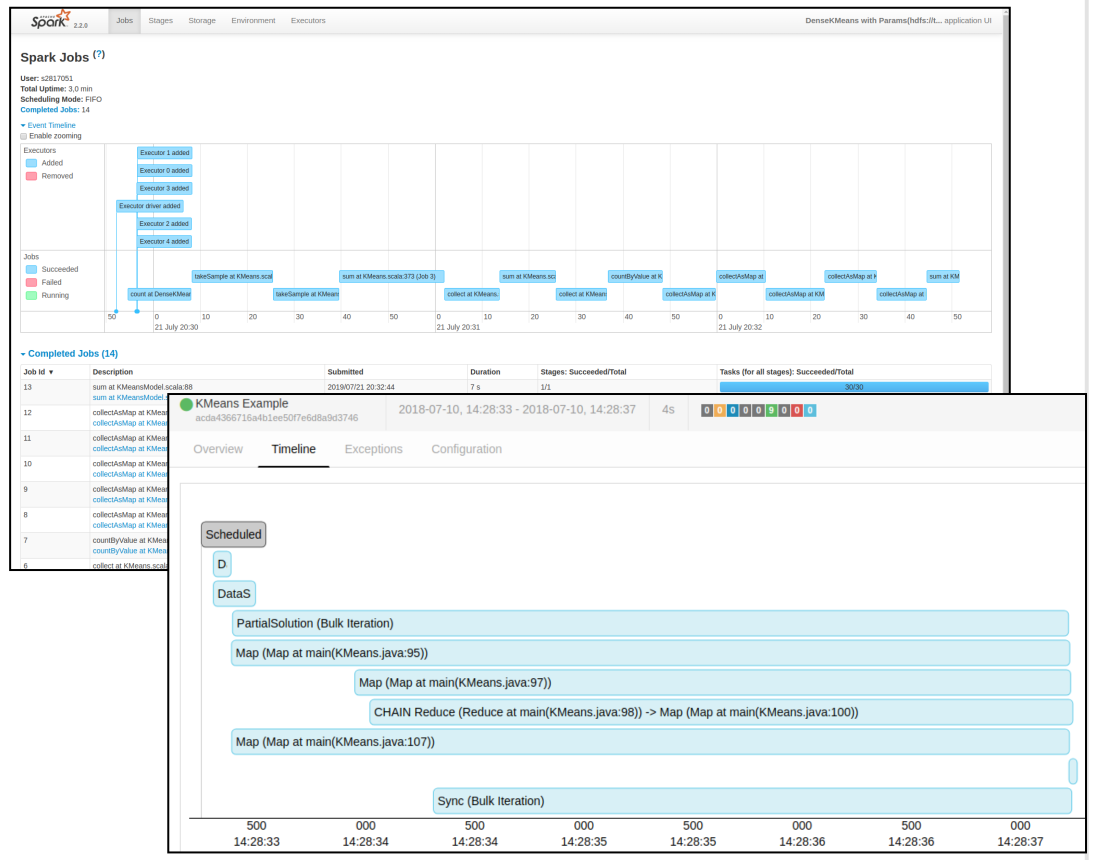
- Apache Spark and Apache Flink are two Big Data frameworks used for fast data exploration and analysis.
- Applications are implemented by chaining API functions which are customized by user-defined functions.
- Performance investigation and optimization is limited by the integrated dashboards and monitoring systems.
- The interaction of the user code and the framework is encapsulated in the programming model, i. e. the abstraction of second-order API functions.
- Due to this hidden information, performance investigation and optimization of workflows is hard.



The programming model of Spark and Flink consists of chaining API functions which apply user-defined functions to data partitions. Created graphs of functions can be viewed in the dashboard. The pictures show a graph of one Spark stage (top) and the full application graph for Flink (bottom) for a KMeans clustering application.

## 2) Challenges

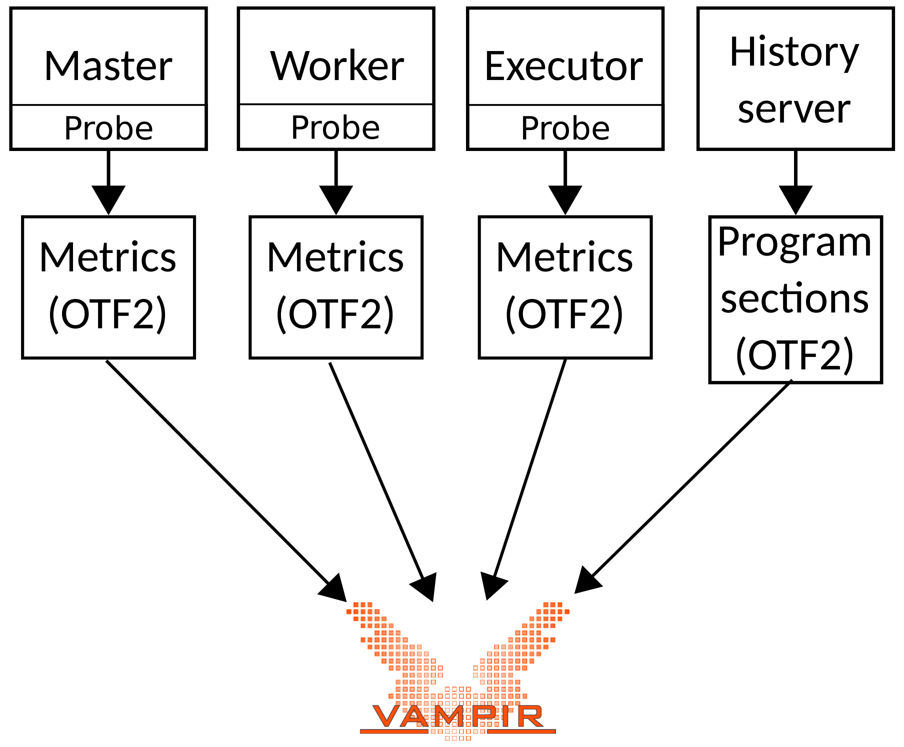
- Monitoring the execution with the integrated dashboards gives only aggregated data indicating performance of the workflow steps, e. g. numbers of records.
- The separation of timelines (e. g. jobs, stages and tasks in Apache Spark) and metrics (aggregated for e. g. jobs or tasks) in the dashboard limits insights into the workflow characteristics over time.
- A performance comparison of Spark and Flink requires to compare multiple views visually.
- Established performance investigation tools from the HPC area, such as Vampir, are not supported and cannot be directly applied, due to missing programming language support, e. g. Java or Scala.



Job timelines in Apache Spark (top/back) and Apache Flink (bottom/front)

## 3) Proposed approach

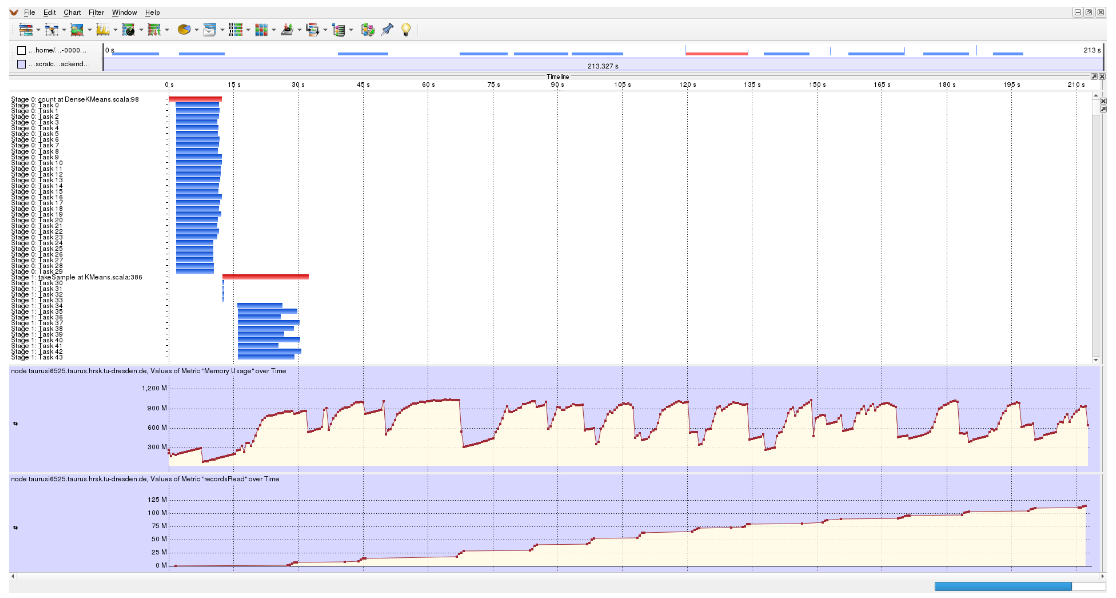
- Idea: Performance data sampled at runtime enhances data collected by in-built framework monitoring (history server).
- The framework-provided monitoring interfaces for the Java management extensions (JMX) system are enabled in order to be able to collect framework and JavaVM-related performance metrics.
- Probes are inserted in the processes to sample the JMX metrics in regular time intervals.
- The probes store data in Open Trace Format 2 (OTF2).
- After the execution, a separate process converts start and end times of program sections, such as stages and tasks, from the history server to OTF2.
- Trace files with exact time information and values of metrics are combined with timely information of program sections and displayed by Vampir.



Each process stores metrics in a trace. After execution, the program regions are obtained from the history server. The region and metrics timelines are shown together in Vampir.

## 4) Results and future work

- Our proposed work integrates the overview of program sections and related performance metrics.
- The view contains data of all levels of the hierarchical execution management model, e. g. stages and tasks.
- The overhead of the metrics sampling depends on the time interval between samples (<10% for 20 metrics @100Hz).
- Data for performance investigation is stored in an established format independent from Spark or Flink versions, so that state-of-the-art performance tools, i. e. Vampir, can be used for analysis.
- The system allows to enrich the traces with further data, such as call stack data gathered via Java bytecode instrumentation and to compare different implementations for a problem.



Spark KMeans example in Vampir: The display shows the overview of all stages and tasks obtained from the history server (top), in combination with per-process metrics from the measurement (bottom). Multiple user-selectable metric views are possible, such as the two views in the picture.

**Technische Universität Dresden/Center for information services and high-performance computing (ZIH)**

M.Sc. Jan Frenzel
E-Mail: jan.frenzel@tu-dresden.de

Dr. René Jäkel
E-Mail: rene.jaekel@tu-dresden.de

BMBF grant no. 01IS18026

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

TECHNISCHE UNIVERSITÄT DRESDEN
UNIVERSITÄT LEIPZIG
CBG Max Planck Institute of Molecular Cell Biology and Genetics
Leibniz-Institut für ökologische Raumentwicklung
HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
HELMHOLTZ ZENTRUM FÜR UMWELTFORSCHUNG UFZ