The Apache Flink Platform

“Die Apache Flink Plattform zur parallelen Analyse von Datenströmen und Stapeldaten”

J. Traub | T. Rabl | F. Hueske | T. Rohrmann | V. Markl

Platform Overview and Related Systems

Core Features of Flink
- General-purpose data processing for clusters
- Compatible: Kafka, Hadoop YARN, HDFS …
- Fully pipelined native streaming runtime
- Built-in program optimizations
- Flexible stream window discretization
- Streaming with exactly once guarantee
- State full operators and UDF support

One system for a variety of workloads
- Batch Processing
- Stream Processing
- Graph Processing
- Machine Learning

Flinks Modular System Stack

Flink in the Analytics Eco System

Applications
- Hive
- Graph
- Mahout
- Pig
- Crunch
- Spark
- Storm
- Flink
- Tez

Data processing engines
- MapReduce
- Spark
- Storm
- Tez

App and resource management
- Yarn
- Mesos

Storage, streams
- HDFS
- HBase
- Kafka
- …

Expressive and Intuitive APIs

Word-POJO:

```
case class Word (word: String, frequency: Int)
```

Scala Batch Processing:

```
val lines: DataSet[String] = env.readTextFile(...)
lines.flatMap(line => line.split(" "))
  .map(word => Word(word,1))
  .groupBy("word")
  .sum("frequency")
  .print()
```

Scala Stream Processing:

```
val lines: DataStream[String] = env.fromSocketStream(...)
lines.flatMap(line => line.split(" "))
  .map(word => Word(word,1))
  .keyBy("word")
  .window(Time.of(5,SECONDS))
  .every(Time.of(1,SECONDS))
  .sum("frequency")
  .print()
```

Further APIs: Python, Java, Gelly (Graph Processing), Table (SQL-Like decararity) , ML Library

Stay tuned!

Mailing List:
news@flink.apache.org

flink.apache.org/blog

@ApacheFlink