UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

BIG DATA STREAMING

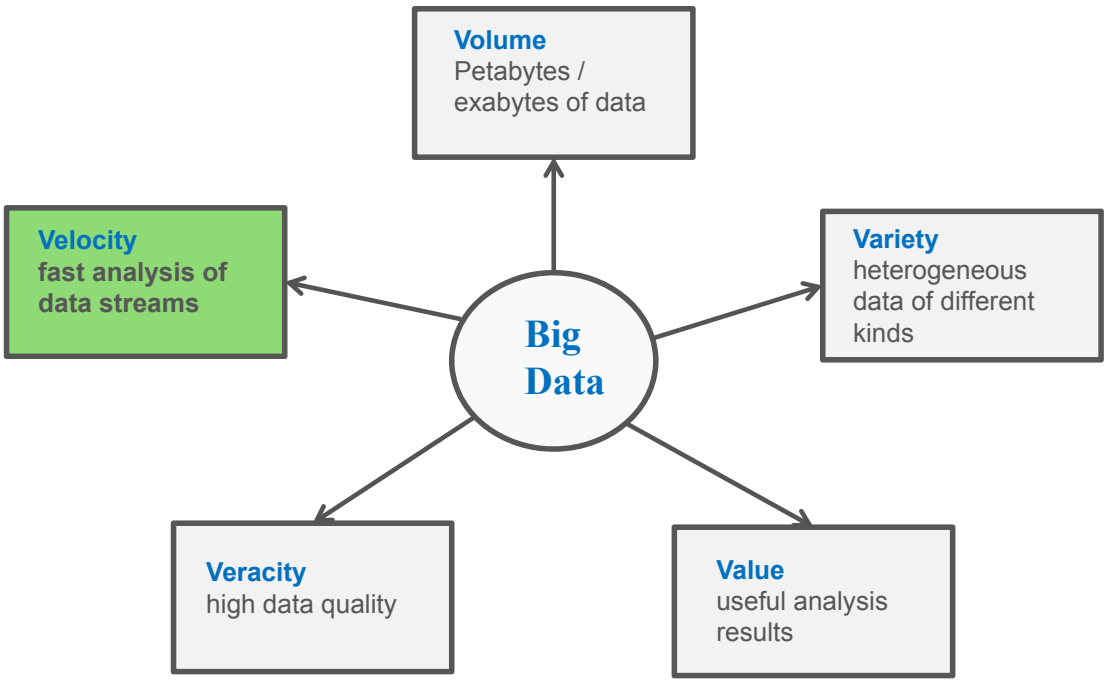
PROF. DR. E. RAHM
UND MITARBEITER

WS 2016/17

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

BIG DATA CHALLENGES



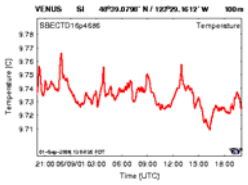
ScaDS DRESDEN LEIPZIG

DATA STREAMS

data stream: sequence of data that is too large to be stored in available memory

Examples:

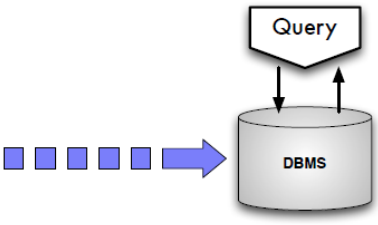
- click streams
- messages in social networks (Twitter, etc.)
- music/video streaming (Spotify, Netflix, etc.)
- financial data (stock changes, credit card tx)
- traffic control (networks, highways)
- energy generation / consumption
- sensor networks
- weather monitoring
- monitoring host intrusion ...



ScaDS DRESDEN LEIPZIG

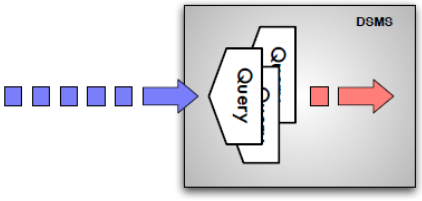
PROCESSING MODELS

- Batch Processing = Store & Process



- Traditional databases
- MapReduce

- Online Processing = Continuous Query Processing (CQP)



- Data Stream Management Systems (DSMS), Stream Processing Engines (SPE)

source: K. Sattler | TU Ilmenau

ScaDS DRESDEN LEIPZIG

STREAM PROCESSING: BASIC PRINCIPLES

- data stream: $<o_i, o_{i+1}, o_{i+2}, ...>$
  - stream element o_i=(*data items*, ***timestamp***)
- operators: filtering, aggregation, joins, ...
- challenge: theoretically unbounded data streams, but limited resources (CPU, memory,...)
- solution: define a window (subset) over a data stream

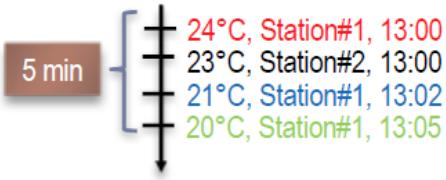
- different possible semantics
  - time-based: last 5 minutes
  - count-based: 100 items
  - data-driven: web session

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

COMPLEX EVENT PROCESSING

- detecting complex events in streams of data
  - event = stream element
  - complex event = sequence of events (not necessarily consecutive)
  - evaluation of logical and temporal conditions (on data values/timestamps)



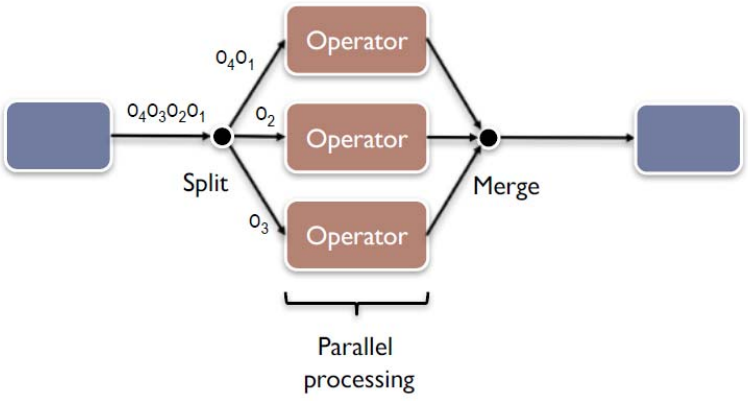

```
SEQ(A, B, C) WITH
  A.Temp > 23°C &&
  B.Station = A.Station && B.Temp < A.Temp &&
  C.Station = A.Station && A.Temp-C.Temp > 3
```

source: K. Sattler | TU Ilmenau

ScaDS DRESDEN LEIPZIG

SCALABILITY: DATA-PARALLEL SYSTEMS

- high volume of data, high arrival rates
  - parallel processing (multi-core servers, clusters)
  - requires partitioning of data streams



source: K. Sattler | TU Ilmenau

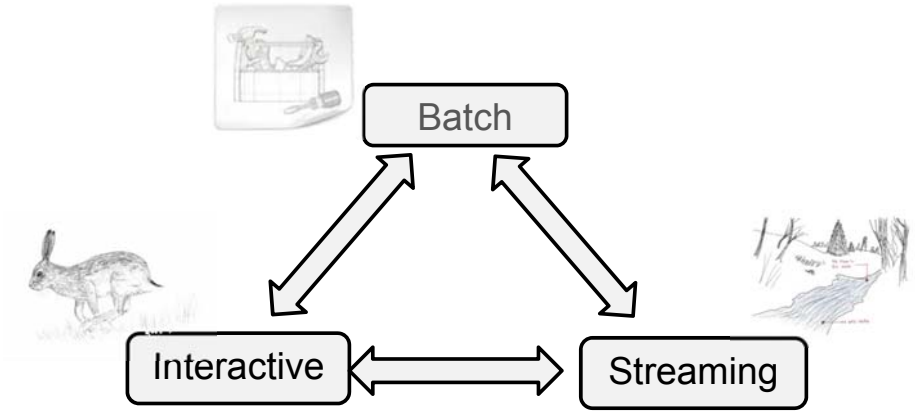
ScaDS DRESDEN LEIPZIG

STREAM PROCESSING SYSTEMS

- Data Stream systems / CEP engines
  - StreamBase, Esper, Software AG Apama, IBM ...
- Dedicated streaming systems for Big Data
  - Storm, Trident; Heron: Twitter
  - Samza (Linkedin)
  - Apache Beam, Millwheel
  - Research prototypes: Kineograph
- General pupose dataflow systems with support for streaming
  - Apache Spark / Flink

ScaDS DRESDEN LEIPZIG

DIFFERENT KINDS OF DATA ANALYSIS



- need to combine ***batch***, ***streaming***, and ***interactive***e computations
- can be achieved general-purpose platforms like Apache Spark and Flink

Spark

Flink

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

COMPARISON (1)

	Storm (Core)	Spark Streaming	Flink Streaming
API	Low-Level	High-Level	High-Level
Language Support	Any	Java, Scala, Python	Java, Scala, Python
Windowing	Not built-in	Only Time-based	Time-based, Row-based, Data-driven with some limitations
Resource Managers	YARN, Mesos, Built-in	YARN, Mesos, Built-in	YARN, Built-in
Computation Model	Tuple-wise	Micro-batch	Tuple-wise
Latency due to Model	Low	Medium	Low
Stream Primitive	Tuple	DStream	DataStream
Fault Tolerance Mechanism	Record ACKs	Micro-batching	Distributed Snapshots
Guarantee	At least once	Exactly once	Exactly once
Stateful Operations	No	Yes	Yes
Flow control	Problematic	Problematic	Natural

source: Hagedorn, Goetze, Saleh, Sattler: *Stream Processing Platforms for Analyzing Big Dynamic Data* it-Information Technology, 2016

ScaDS DRESDEN LEIPZIG

COMPARISON (2)

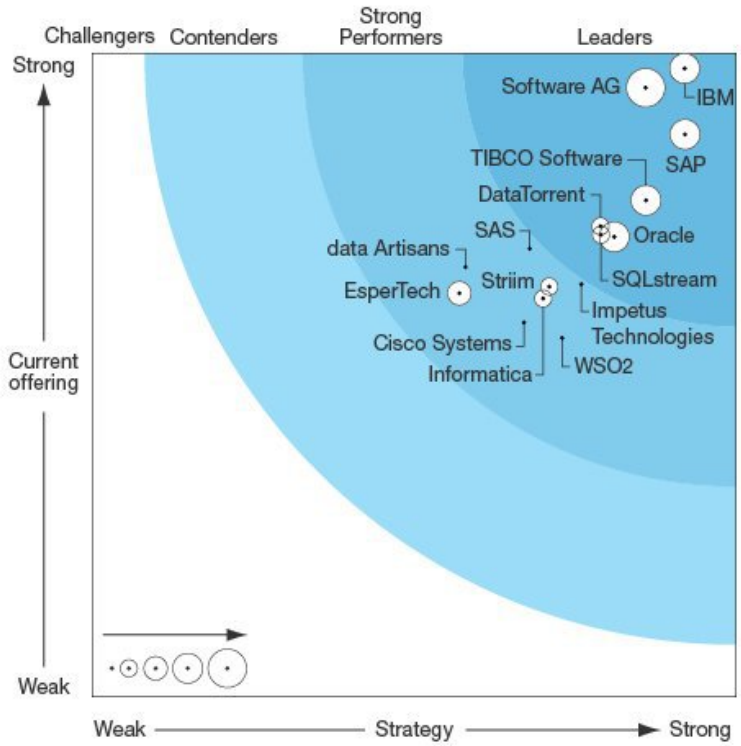
	Flume	NiFi	Gearpump	Apex	Kafka Streams	Spark Streaming	Storm	Storm + Trident	Samza	Flink	Ignite Streaming	Beam (*GC DataFlow)
Current version	1.6.0	0.6.1	incubating	3.3.0	0.9.0.1* (available in 0.10)	1.6.1	1.0.0	1.0.0	0.10.0	1.0.2	1.5.0	incubating
Category	DC/SEP	DC/SEP	SEP	DC/ESP	ESP	ESP	ESP/DEP	ESP/DEP	ESP	ESP/DEP	ESP/DEP	SDK
Event size	single	single	single	single	single	micro-batch	single	mini-batch	single	single	single	single
Available since (incubator since)	June 2012 (June 2011)	July 2015 (Nov 2014)	(Mar 2016)	Apr 2016 (Aug 2015)	Apr 2016 (July 2011)	Feb 2014 (2013)	Sep 2014 (Sep 2013)	Sep 2014 (Sep 2013)	Jan 2014 (July 2013)	Dec 2014 (Mar 2014)	Sep 2015 (Oct 2014)	(Feb 2016)
Contributors	26	67	19	53	160	936	207	207	48	159	56	80
Main backers	Apple Cloudera	Hortonworks	Intel Lightbend	Data Torrent	Confluent	AMPLab Databricks	Backtype Twitter	Backtype Twitter	LinkedIn	dataArtisans	GridGain	Google
Delivery guarantees	at least once	at least once	exactly once at least once (with non-fault-tolerant sources)	exactly once	at least once	exactly once at least once (with non-fault-tolerant sources)	at least once	exactly once	at least once	exactly once	at least once	exactly once*
State management	transactional updates	local and distributed snapshots	checkpoints	checkpoints	local and distributed snapshots	checkpoints	record acknowledgements	record acknowledgements	local snapshots distributed snapshots (fault-tolerant)	distributed snapshots	checkpoints	transactional updates*
Fault tolerance	yes (with file channel only)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes*
Out-of-order processing	no	no	yes	no	yes	no	yes	yes	yes (but not within a single partition)	yes	yes	yes*
Event prioritization	no	yes	programmable	programmable	programmable	programmable	programmable	programmable	yes	programmable	programmable	programmable
Windowing	no	no	time-based	time-based	time-based	time-based	time-based count-based	time-based count-based	time-based	time-based count-based	time-based count-based	time-based
Back-pressure	no	yes	yes	yes	N/A	yes	yes	yes	yes	yes	yes	yes*
Primary abstraction	Event	FlowFile	Message	Tuple	KafkaStream	DStream	Tuple	TridentTuple	Message	DataStream	IgniteDataStreamer	PCollection
Data flow	agent	flow (process group)	streaming application	streaming application	process topology	application	topology	topology	job	streaming dataflow	job	pipeline
Latency	low	configurable	very low	very low	very low	medium	very low	medium	low	low (configurable)	very low	low*
Resource management	native	native	YARN	YARN	Any process manager (e.g. YARN, Mesos, Chef, Puppet, Salt, Kubernetes, ...)	YARN Mesos	YARN Mesos	YARN Mesos	YARN	YARN	YARN Mesos	integrated*
Auto-scaling	no	no	no	yes	yes	yes	no	no	no	no	no	yes*
In-flight modifications	no	yes	yes	yes	yes	no	yes (for resources)	yes (for resources)	no	no	no	no
API	declarative	compositional	declarative	declarative	declarative	declarative	compositional	compositional	compositional	declarative	declarative	declarative
Primarily written in	Java	Java	Scala	Java	Java	Scala	Clojure	Java	Scala	Java	Java	Java
API languages	text files Java	REST (GUI)	Scala Java	Java	Java	Scala Java Python	Scala Java Clojure Python Ruby	Java Python Scala	Java	Java Scala Python	Java .NET C++	Java*
Notable users	Meebo Sharethrough SimpleGeo	N/A	Intel Levi's Honeywell	Capital One GE Predix PubMatic	N/A	Kelkoo Localytics AsiaInfo Opentable Fairndata Guavus	Yahoo! Spotify Groupon Flipboard The Weather Channel Alibaba Baidu Yelp WebMD	Klout GumGum CrowdFlower	LinkedIn Netflix Intuit Uber	King Otto Group	GridGain	N/A

source: www.quora.com/What-are-the-differences-between-Apache-Spark-Storm-Flink-Beam-Apex

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

Figure 2: The Forrester Wave™: Big Data Streaming Analytics, Q1 '16



UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

SEMINAR

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

SEMINARZIELE

- Beschäftigung mit einem praxis- und wissenschaftlich relevanten Thema
  - kann Grundlage für Abschlussarbeit oder SHK-Tätigkeit sein
- Erarbeitung + Durchführung eines Vortrags unter Verwendung wissenschaftlicher (englischer) Literatur
- Diskussion
- Schriftliche Ausarbeitung zum Thema
- Hilfe und Feedback durch zugeteilten Betreuer

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

SEMINAR: ANRECHNUNGSMÖGLICHKEITEN

- Masterstudium, insbesondere für Schwerpunkt „Big Data“
  - Teil der Module Moderne Datenbanktechnologien
  - Seminarmodul
- Bachelorstudium
  - Seminarmodul
  - evtl. Realisierung von IS

UNIVERSITÄT LEIPZIG

ScaDS DRESDEN LEIPZIG

SCHEINVERGABE / MODULPRÜFUNG

- selbständiger Vortrag mit Diskussion (ca. 45 Minuten)
  - Abnahme der Folien durch Betreuer
- schriftliche Ausarbeitung (ca. 15 Seiten)
  - Abnahme der Ausarbeitung durch Betreuer
  - Ausarbeitung soll zum Vortragstermin vorliegen (Vorträge ab Januar 2017)
- aktive Teilnahme an allen Vortragsterminen
- Modul-Workload: 30h Präsenzzeit,
  120 h Selbststudium

ScaDS DRESDEN LEIPZIG

SEMINAR (3)

- Themenzuordnung
  - Koordinierungstreffen mit Betreuer bis spätestens 4.11.2016
  - ansonsten verfällt Seminaranmeldung
  - freiwilliger Rücktritt auch bis max. 4.11.2016

- Vortragstermine
  - 5x freitags, Ritterstr, ab 6. 1. 2017
  - max. 2 Doppelstunden ab 13:30 Uhr

Themen	Betreuer	max. #Themen	Termin	Studenten
Einführung / Anforderungen	Lin / Kricke	2-3	6.1.	
Systeme S-Store MillWheel Kineograph Twitter Heron Apache Beam	Groß Nentwig Grimmer Peukert Kricke	5	13.1.	
Stream Mining Stream clustering (swarm intell.) Similarity search in tweets (LSH) Monitoring (distance) outliers Taxi demand prediction	Christen Groß Sehili Christen	4	20.1.	
Graph Algorithms on Streams/Dynamic Graphs Partitioning dynamic graphs Event pattern matching Frequent subgraph mining (2x) GraphTau: graph snapshots Anomaly detection	Junghanns Junghanns Petermann Nentwig Grimmer	6	27.1.	
Weitere Aspekte Streaming similarity self join Elastic scaling	Sehili Peukert	2	20.1.	