Real-time stream processing with Twitter Heron

Adrian Bartnik | Complex and Distributed IT Systems | Seminar: Operating Complex IT-Systems

Quick question

What is the record for tweets per minute?

618,725

During football world cup final between Germany and Argentina in 2014

Currently, on average, arount 6000 tweets per seconds

Real-time stream processing

- Processing of high-volume data steams
  → Trading, social networks, Internet of things, system monitoring, analytics

TUBerlin @TUBerlin · 28. Jan.

#TUBerlin feiert chin. Frühlingsfest am 31. Januar: KungFu, Theater und Musik im Jahr des Feueraffens. bit.ly/1PUP5yk @stadtleben

- Characteristics:
  - a one-at-a-time processing model
  - data are processed immediately upon arrival
  - computations are simple and generally independent
  - low latency (sub-second)

Background - Apache Storm

- Real-time fault-tolerant and distributed stream data processing system

- Developed at BackType in 2011
- Aquired and open sourced by Twitter in 2013

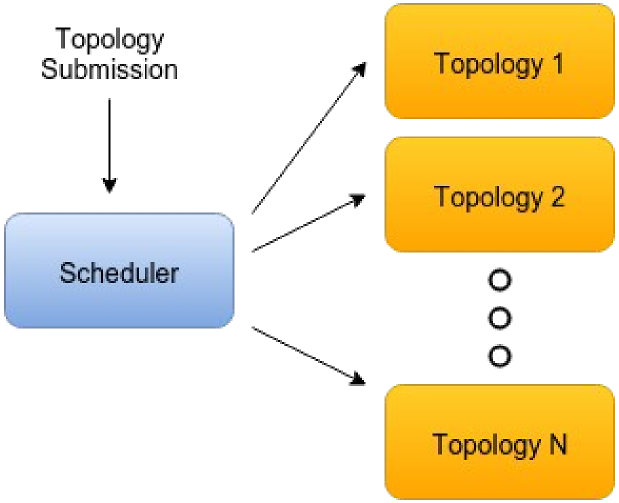
- Characteristics
  - Scalable
  - Resilient
  - Extensible
  - Efficient
  - Easy to administer

Twitter Heron - Architecture

- Evolution from Apache Storm due to problems with scaling, performance, debugging and cluster management
- Compatibility with the Storm API was essential due to existing topologies and ecosystem

- Aurora Scheduler introduces abstraction layer for various other schedulers

- Topology defines a graph of computations



Topology Submission

Scheduler

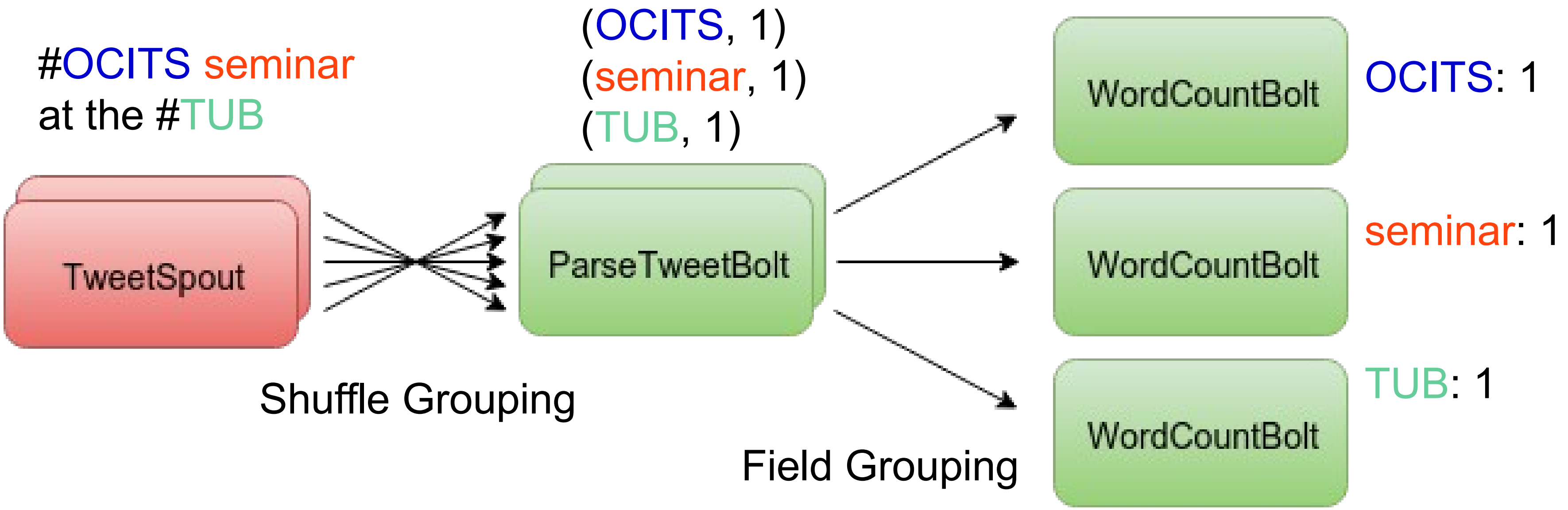
Topology 1

Topology 2

Topology N

Topology – Trending Topics

- Data flow described by directed acyclic graph (DAG)
  - Spouts for input and bolts for computations
  - Different partitioning strategies, e.g. Shuffle, All, Global
  - Topology is a DAG of spouts and bolts



#OCITS seminar
at the #TUB

(OCITS, 1)
(seminar, 1)
(TUB, 1)

OCITS: 1

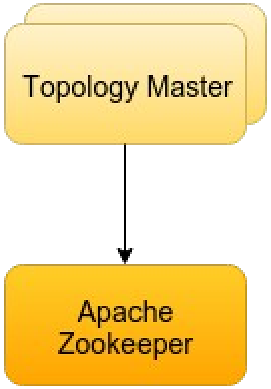
seminar: 1

TUB: 1

Shuffle Grouping

Field Grouping

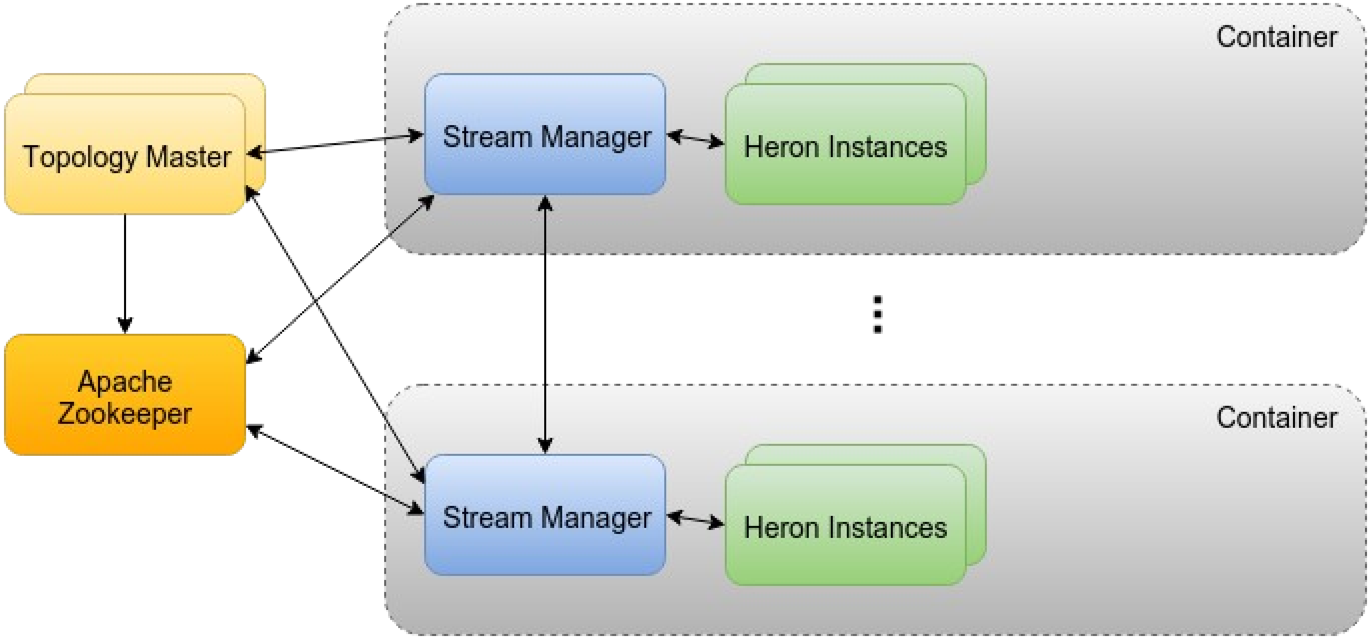
Topology Architecture



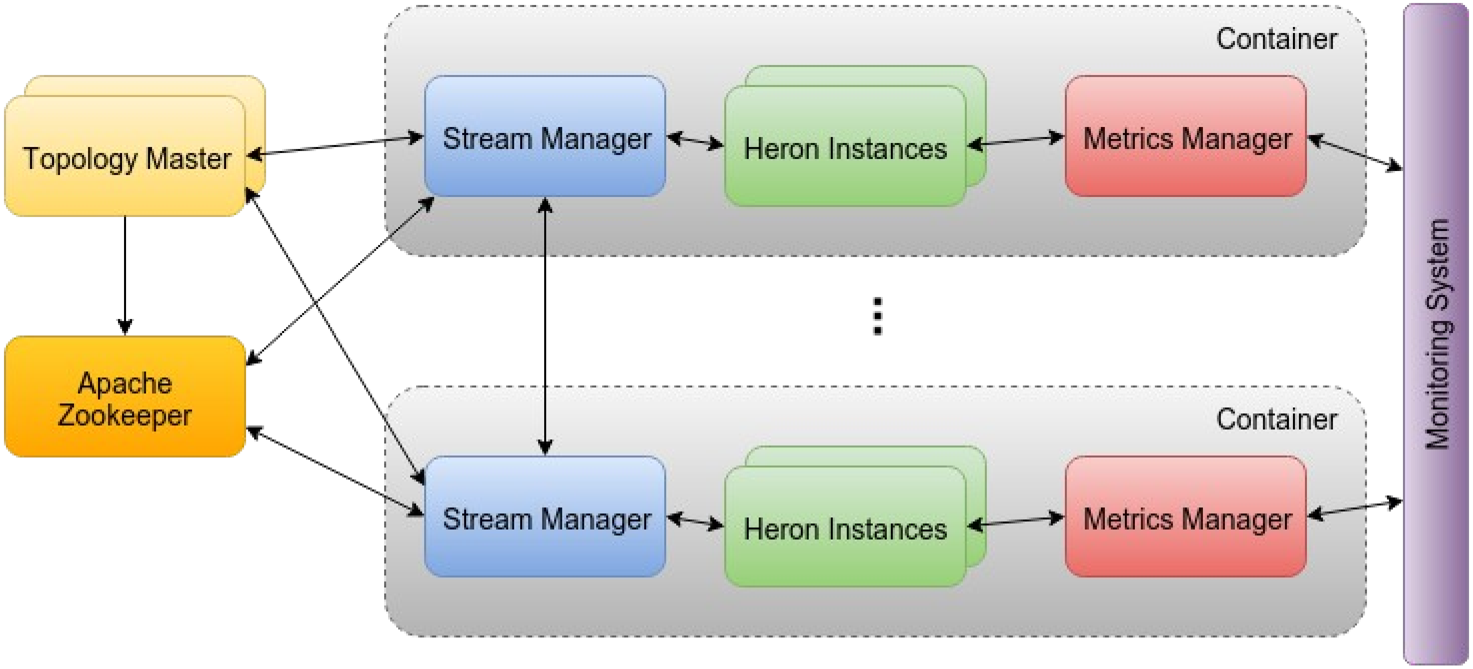
- Topology master
  - Manges lifecylce of a topology
  - Single point of contact for the Scheduler
  - Optional Backup node

- Apache Zookeeper
  - Centralized service and maintains
    - configuration information
    - Naming
    - providing distributed synchronization

Topology Architecture

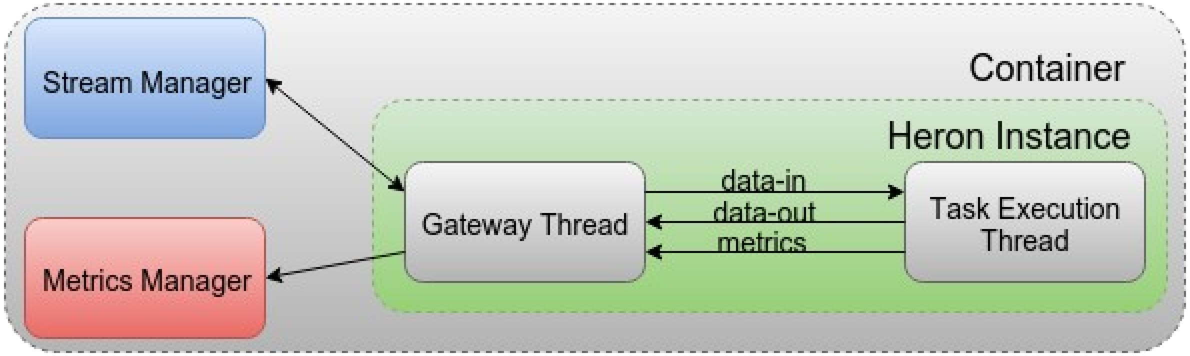


Topology Architecture



Heron Instance

- Runs as a single JVM process
- Is either a spout or bolt
- Gateway Thead communicates with Stream Manager for sending and receiving tuples
- Task Execution Thead
  - Processes the tuples from the data-in queue
  - Emmits metrics for Metrics Manager via the metrics queue

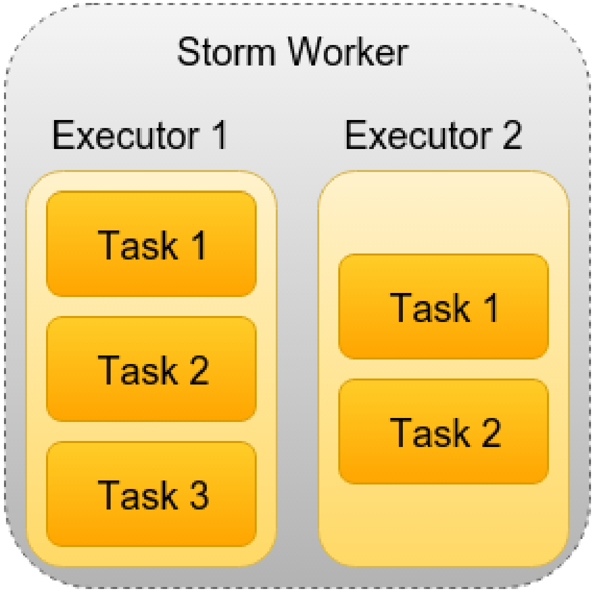


Storm Worker

- Storm worker similar to a container
- Runs parts of a topology, but in one single JVM

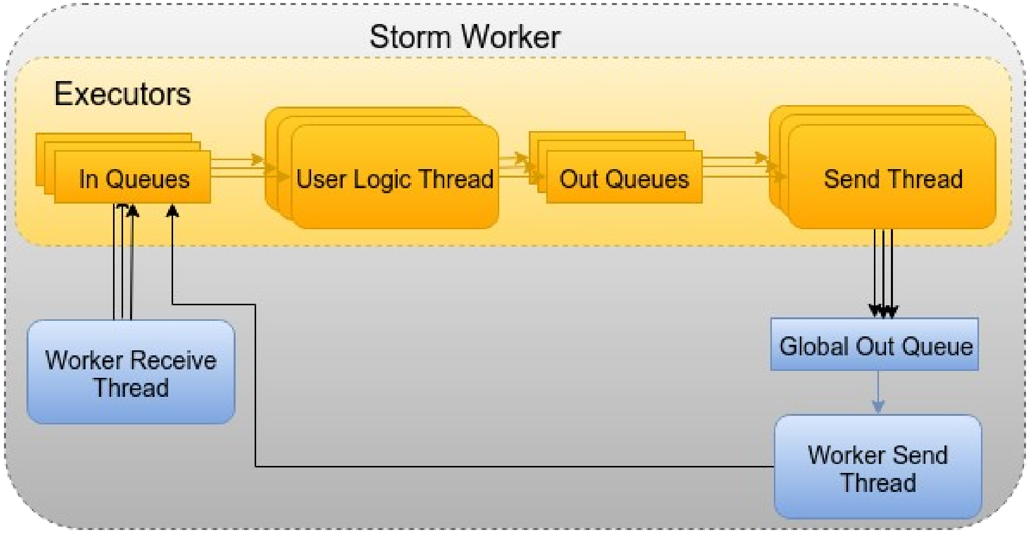
- Executor (≙ Heron Instance)
  - Contains multiple tasks
  - Consists of 2 Threads

- Task
  - Instance of either spout or bolt
  - Assignment of task to executor is static



Storm Worker

- Scheduling on multiple levels
  - Physical machine may hosts several Storm workers
  - Each executor is mapped to 2 threads: User Logic and Send Thread
  - User logic thread may run multiple tasks
- Worker may run disparate tasks
  - Not possible to isolate resource usage
- Each tuples has to pass 4 threads
  - Significant overhead



Storm Worker vs Heron Instance

- Streaming Managers simplify routing

- 1 JVM per Heron Instance
  - Runs only one task (spout or bolt)
  - Simplifies scheduling
  - Easier debugging in case of failure

- Reduction of queues and threads lowers overhead for tuple execution

- Separation of monitoring

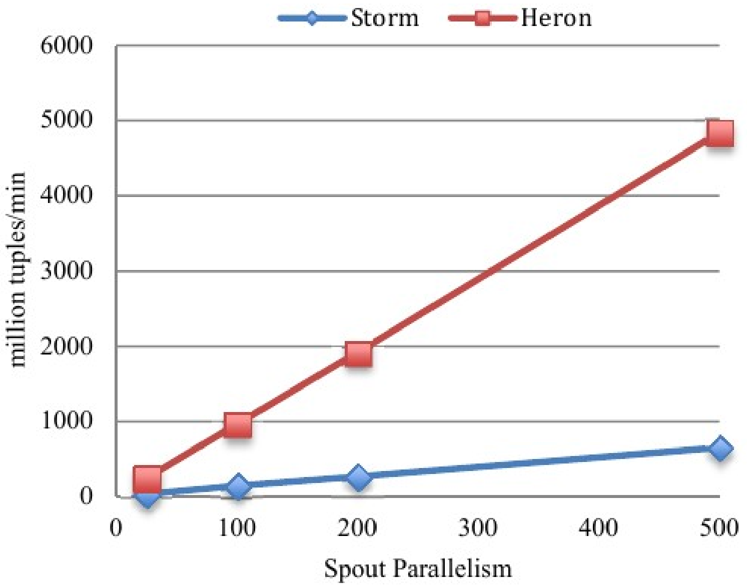


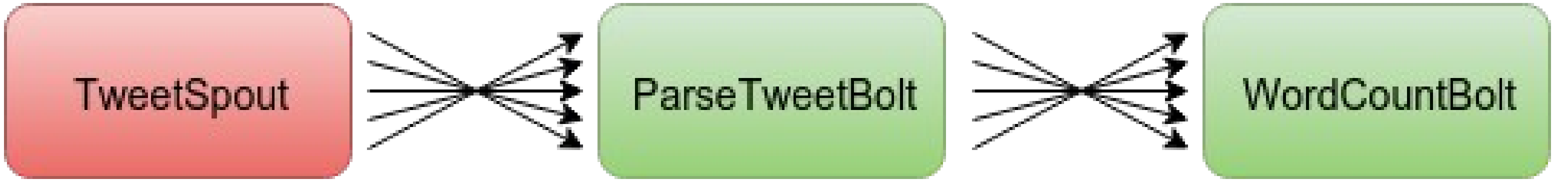
Figure 12: Throughput with acknowledgements disabled

Summary

- Processing of high-volume data steams
  → Trading, social networks, Internet of things, system monitoring, analytics
- Characteristics:
  - a one-at-a-time processing model
  - data are processed immediately upon arrival and „in-stream“
  - computations are simple and generally independent
- Twitter Heron
  - Moved routing logic to Streaming manager
  - Less overhead in Heron Instances due to scheduling

Requirement of a real-time streaming engine

- Michael Stonebraker et al. have defined requirements for a real-time streaming engine
  - Keep the data moving and avoid expensive storage operations
  - Support a high-level “StreamSQL” language
  - Handle stream imperfections
  - Integrate stored and streaming data
  - Process and respond instantaneously


TweetSpout
ParseTweetBolt
WordCountBolt