QTopology

Node.js platform for running stream processing

Created by QMiner community (@qminer)

Features

- Storm-like design - bolts and spouts
- All nodes run in process
- Separate topologies run in separate processes
- Distributed execution via worker
    - With no central coordinator

What are bolts and spouts?

In stream processing, we:

- pump data coming into the system
- process, transform, aggreate the data
- write the data to some storage

In QTopology we have:

- Spouts pump data into the system at proper pace
- Bolts process and transform the data

QTopology does the wiring and routing between the nodes

Bolt

- Receives messages (with stream id)
- Can emit 0, 1 or multiple new messages on multiple streams
- Must acknowledge the input tuple

Spout

- Receives calls to `next()`
- Can emit single data tuple
  - null means no data, this spout wont be queried in the next 1000msec

Heartbeat

- QTopology sends heartbeat signals to all nodes
- Time interval set in topology config
- Bolts can emit new data

Common intialization and shutdown

- Optional code to be executed at start and shutdown
- Can create context object
    - Completely custom
    - Can contain common settings, db connection etc.
- Context is passed to all nodes

Local vs Distributed setting

- **Local**: QTopology runs inside the process. Only single topology is executed
- **Distributed**: Each machine runs it's own worker, which:
  - connects to common storage
  - receives topologies to execute
  - runs each topology in separate process

Thank you

- Architecture is subject to future changes

> This presentation can be viewed using Marp