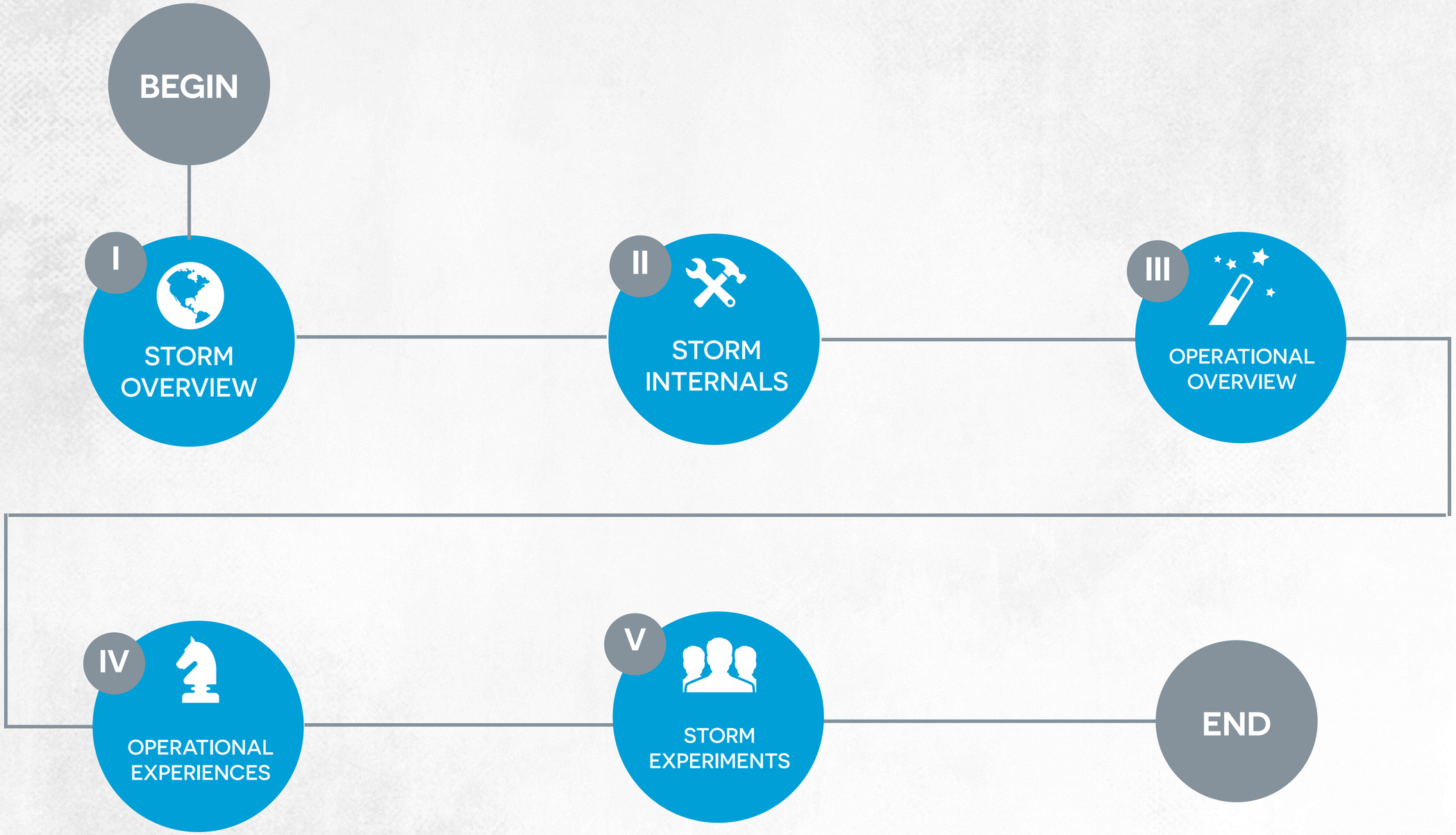
Storm @Twitter

KARTHIK RAMASAMY
@KARTHIKZ

#TwitterAtSigmod
#TwitterDataStorm

Ankit Toshniwal, Siddarth Taneja, Amit Shukla, Jignesh Patel, Sanjeev Kulkarni
Jason Jackson, Krishna Gade, Maosong Fu, Jake Donham, Nikunj Bhagat
Sailesh Mittal and Dmitriy Ryaboy

TALK OUTLINE



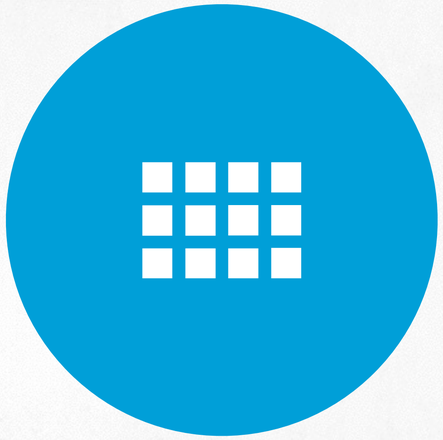
OVERVIEW

I

WHAT IS STORM?

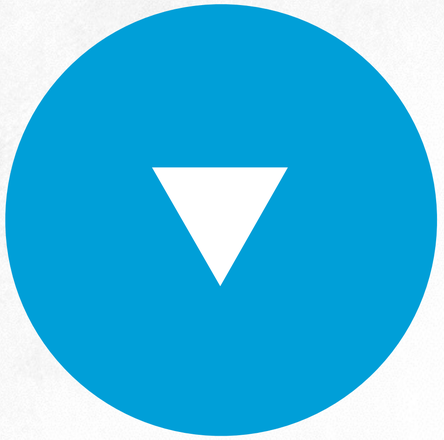
Streaming platform for analyzing **realtime data** as they arrive, so you can react to data **as it happens**.

GUARANTEED MESSAGE PROCESSING

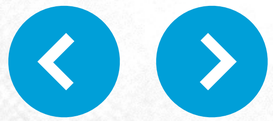


HORIZONTAL SCALABILITY

ROBUST FAULT TOLERANCE



CONCISE CODE- FOCUS ON LOGIC



STORM DATA MODEL

TOPOLOGY

Directed acyclic graph

Vertices=computation, and edges=streams of data tuples

SPOUTS

Sources of data tuples for the topology

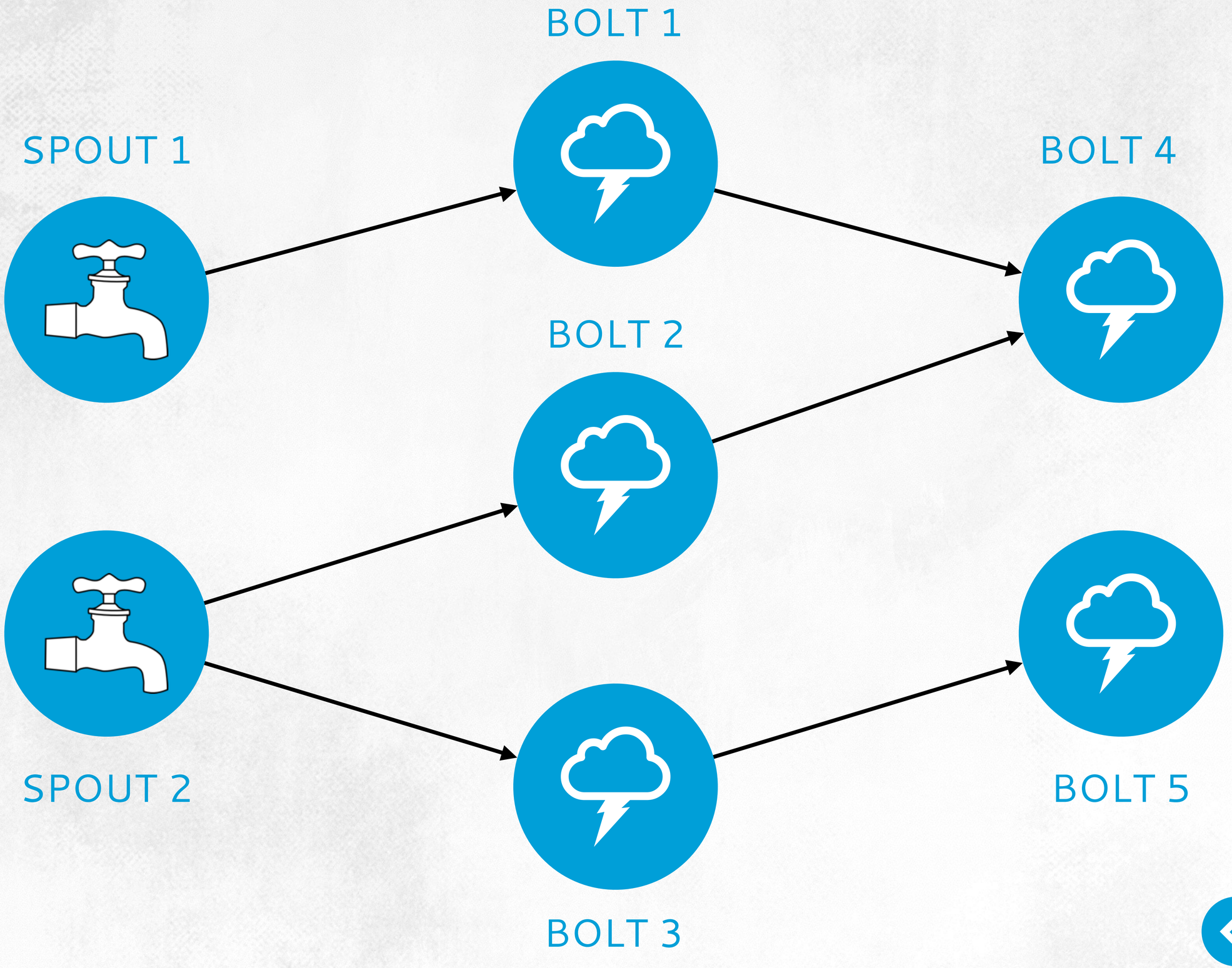
Examples - Kafka/Kestrel/MySQL/Postgres

BOLTS

Process incoming tuples and emit outgoing tuples

Examples - filtering/aggregation/join/arbitrary function

STORM TOPOLOGY



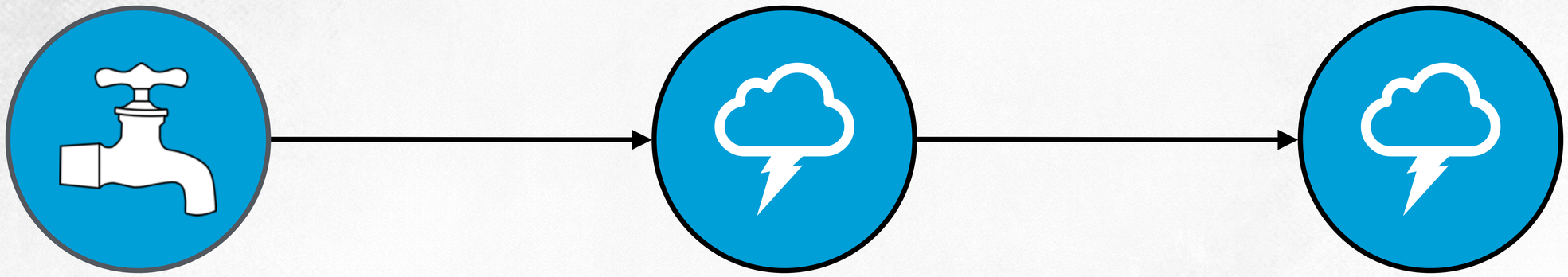
WORD COUNT TOPOLOGY

Live stream of Tweets

#worldcup : 1M

soccer: 400K

....

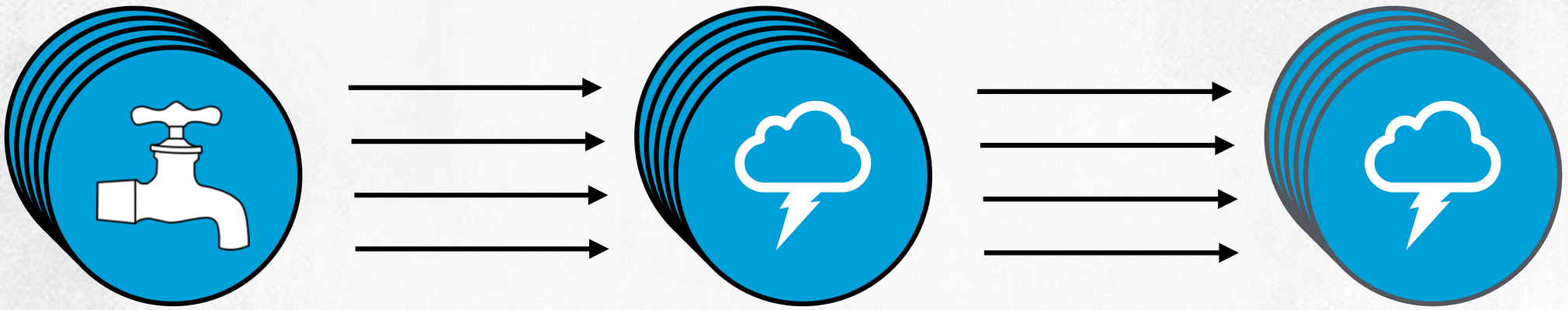


TWEET SPOUT

PARSE TWEET BOLT

WORD COUNT BOLT

WORD COUNT TOPOLOGY



TWEET SPOUT TASKS

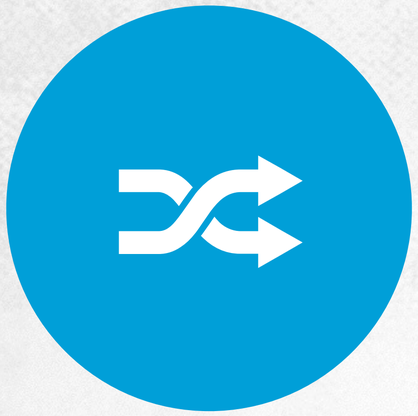
PARSE TWEET BOLT TASKS

WORD COUNT BOLT TASKS

When a parse tweet bolt task emits a tuple which word count bolt task should it send to?

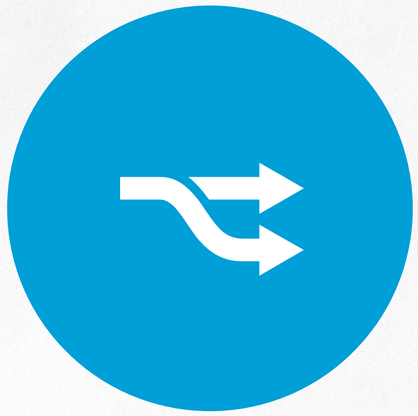
STREAM GROUPINGS

SHUFFLE GROUPING



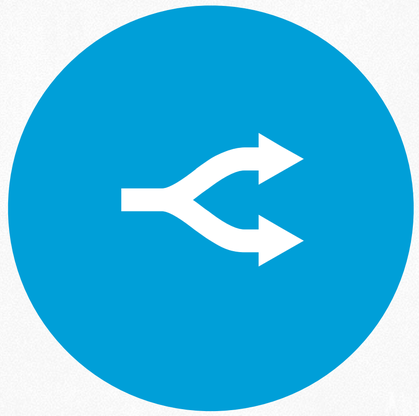
Random distribution of tuples

FIELDS GROUPING



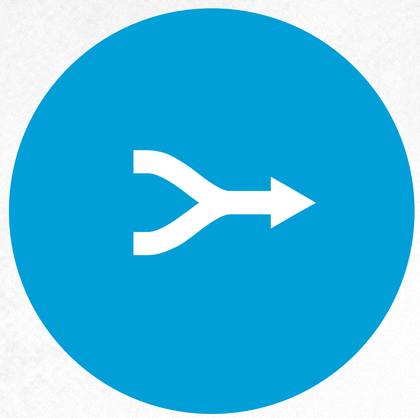
Group tuples by a field or multiple fields

ALL GROUPING



Replicates tuples to all tasks

GLOBAL GROUPING

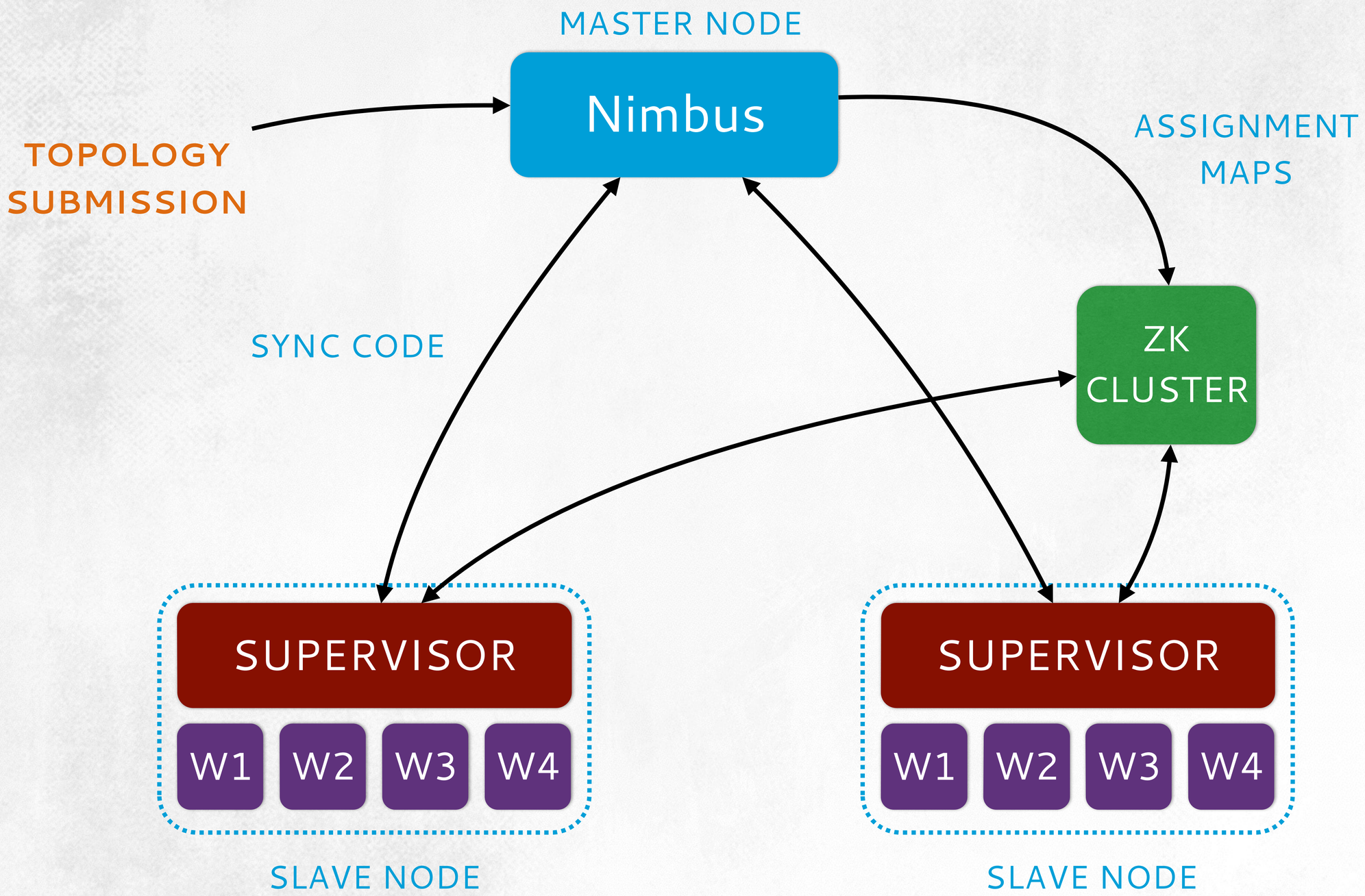


Sends the entire stream to one task

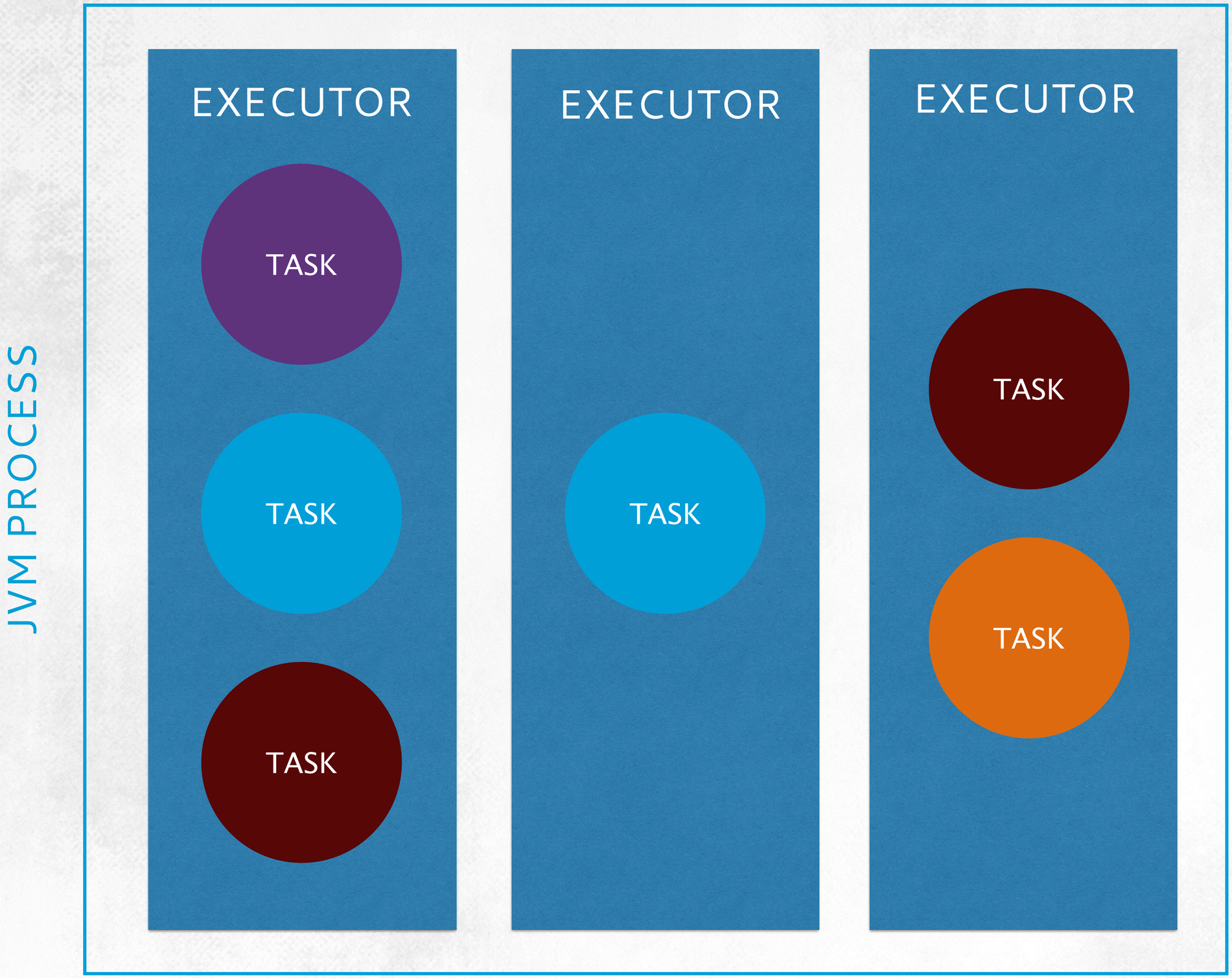
II

STORM INTERNALS

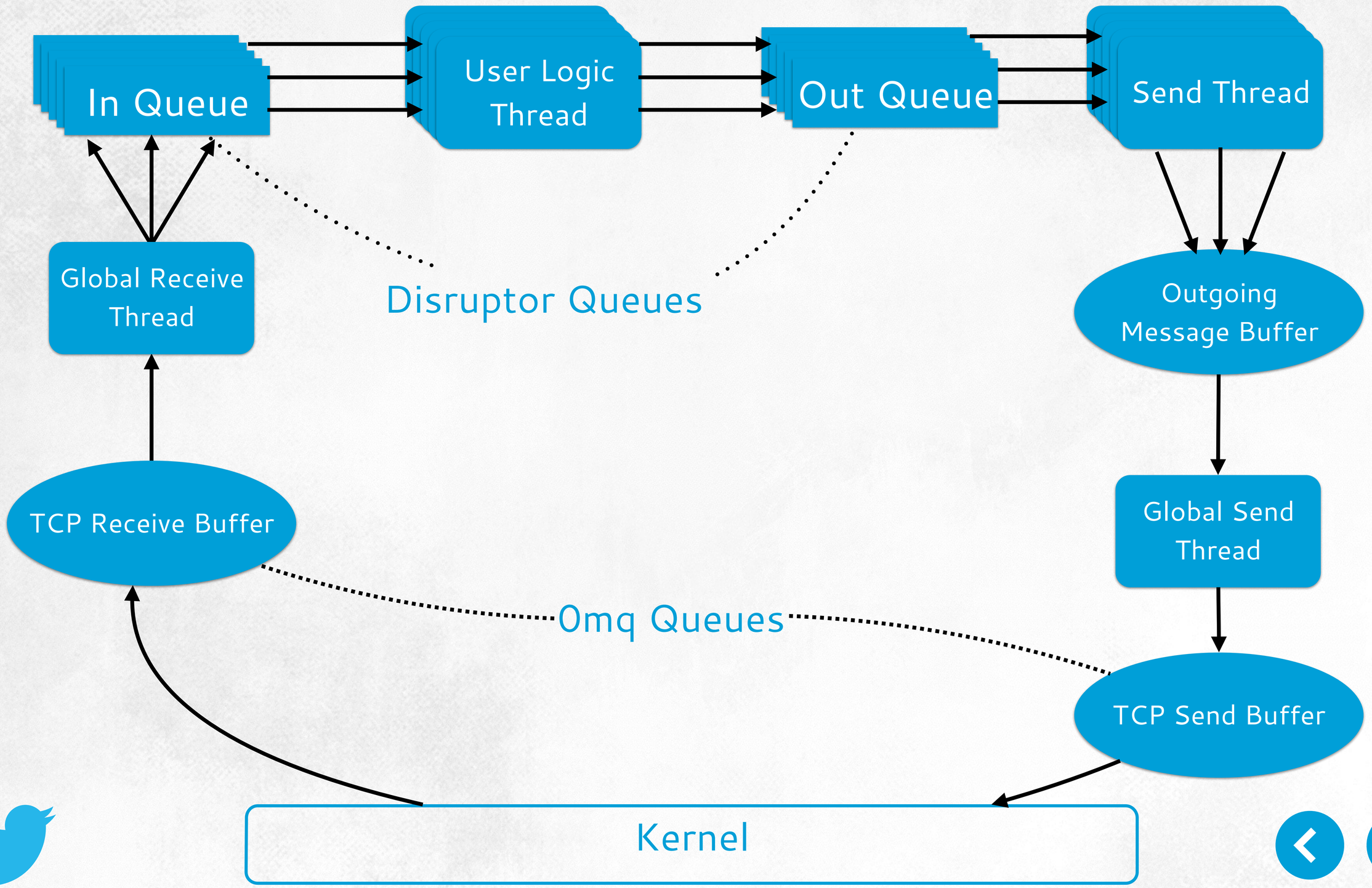
STORM ARCHITECTURE



STORM WORKER



DATA FLOW IN STORM WORKERS



III

OPERATIONAL OVERVIEW

STORM METRICS

1

SUPPORT AND TROUBLE SHOOTING

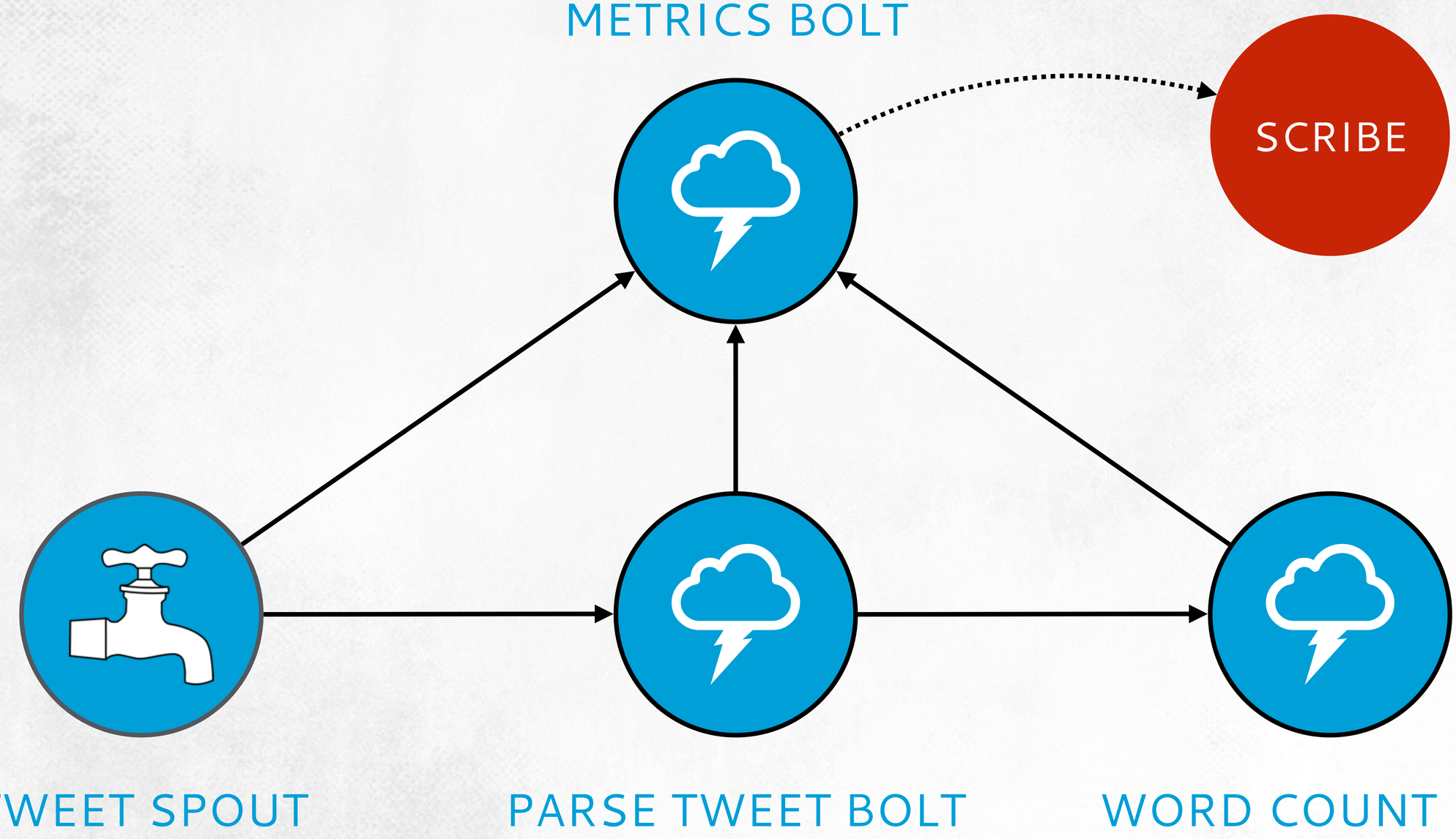
2

CONTINUOUS PERFORMANCE

3

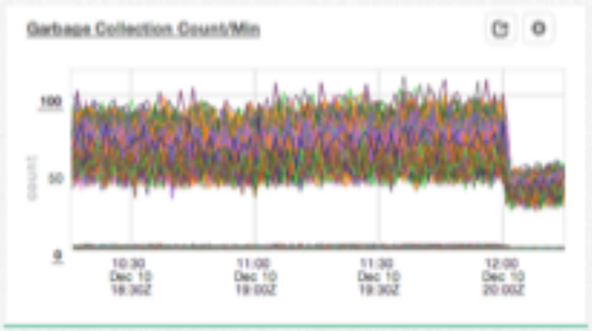
CLUSTER AVAILABILITY

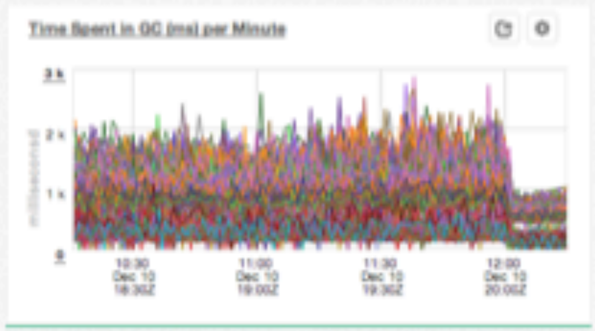
COLLECTING TOPOLOGY METRICS

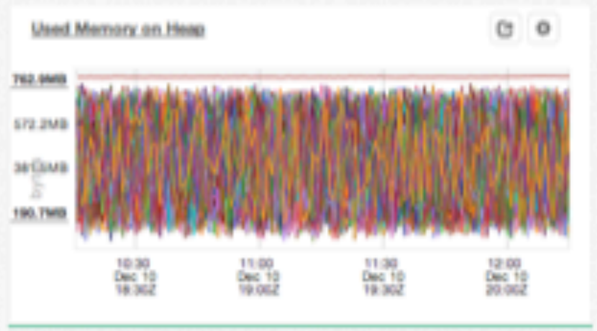


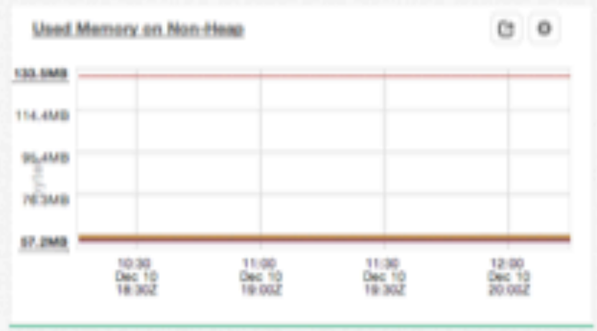
SAMPLE TOPOLOGY DASHBOARD

Workers (aka JVM Processes)

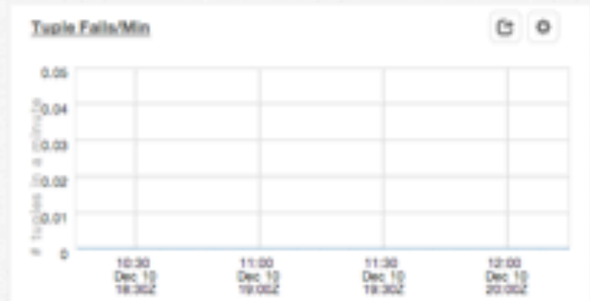


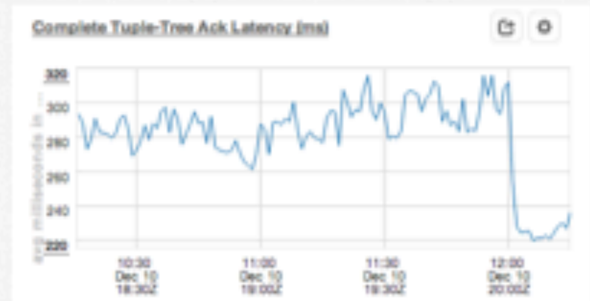




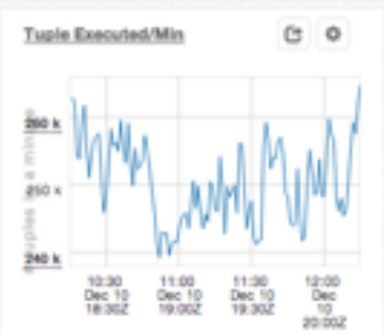


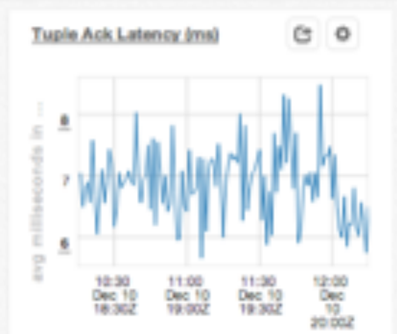
Spout: Tail-FlatMap-Source

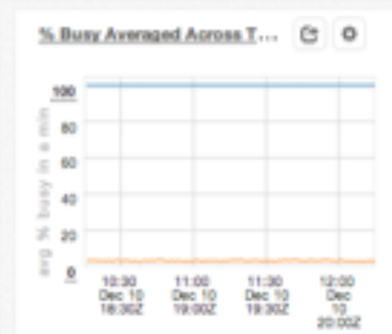




Bolt: Tail-FlatMap





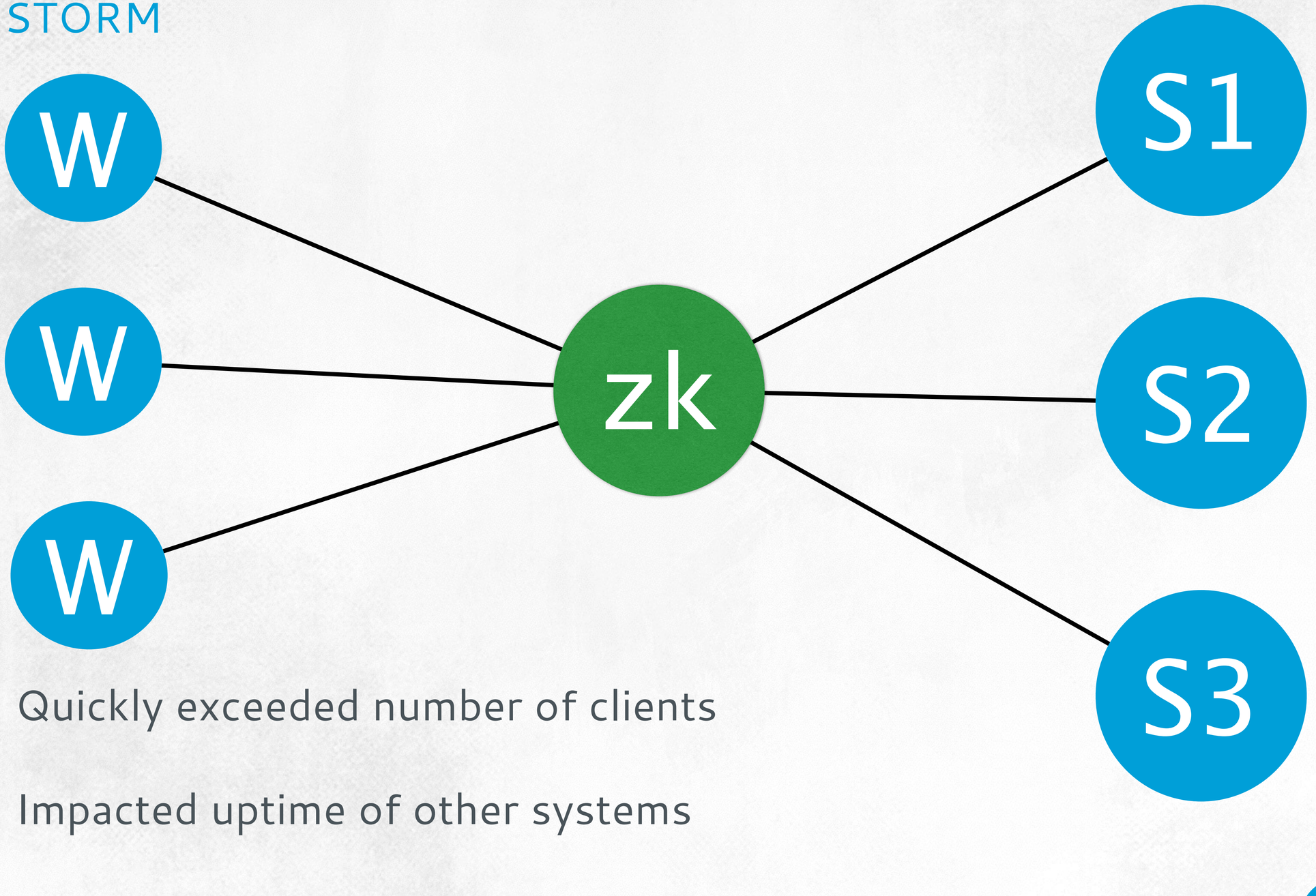


IV

OPERATIONAL EXPERIENCES

OVERLOADED ZOOKEEPER

Shared configuration

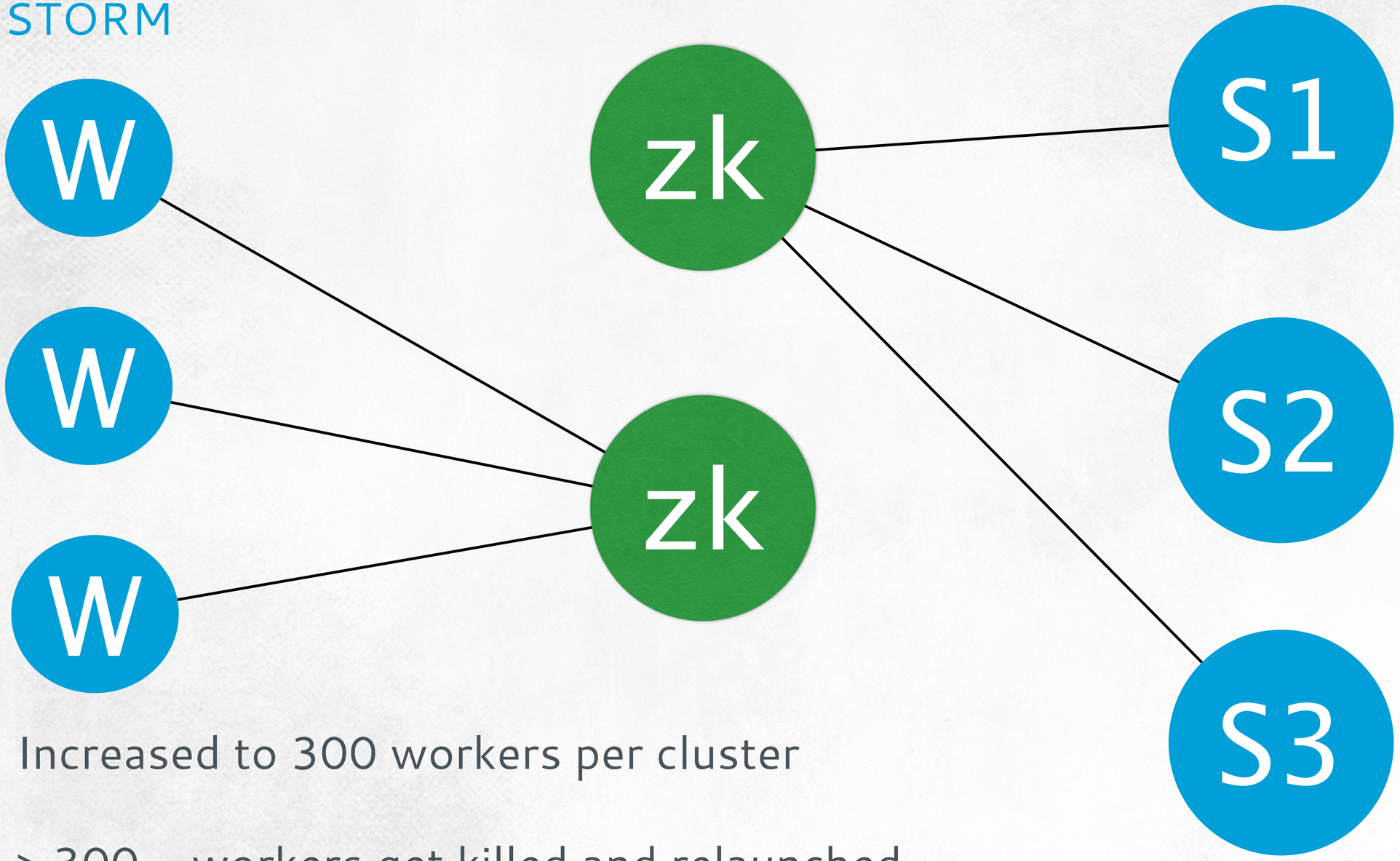


Quickly exceeded number of clients

Impacted uptime of other systems

OVERLOADED ZOOKEEPER

Detached configuration



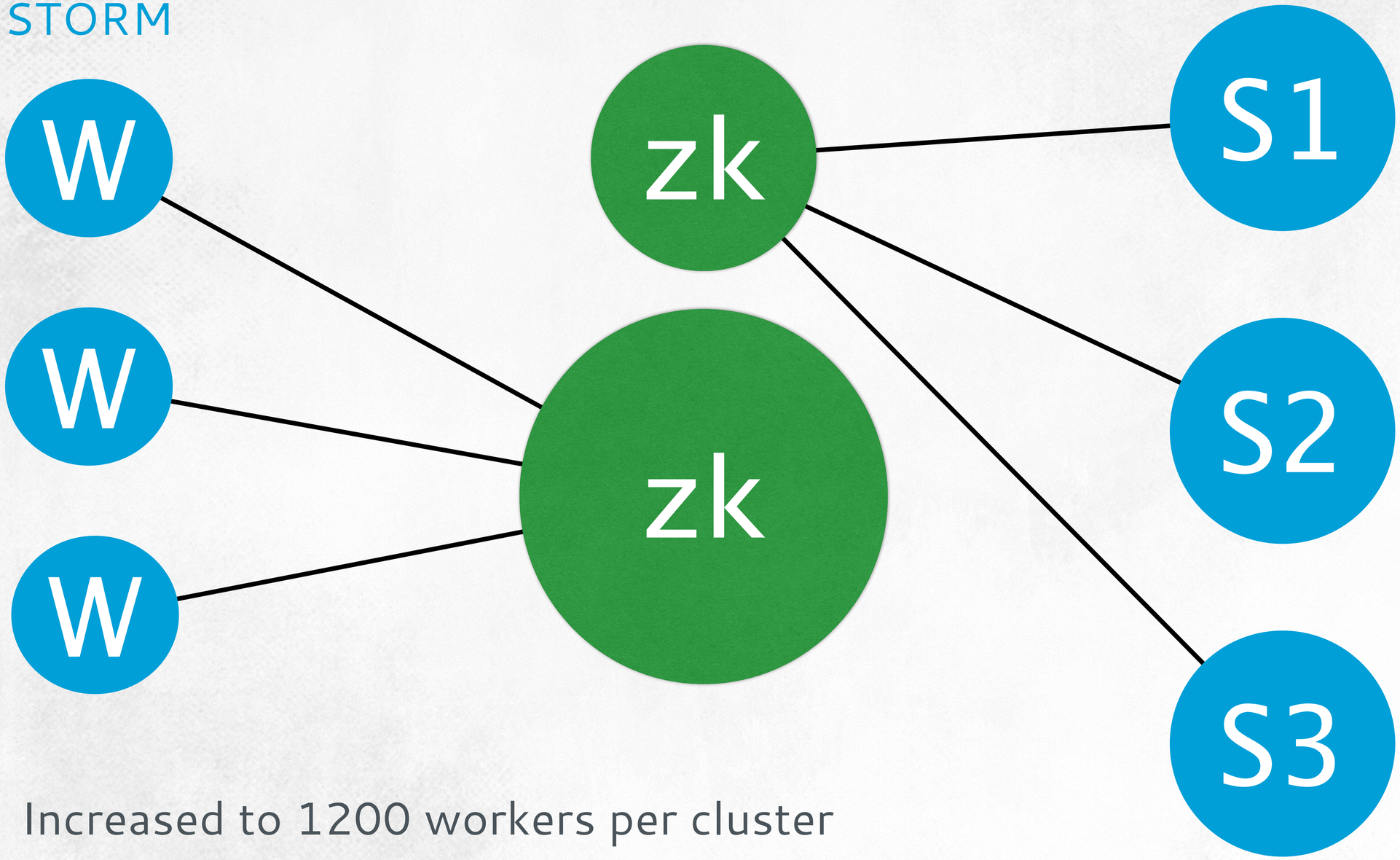
Increased to 300 workers per cluster

> 300 – workers get killed and relaunched

Worker heart beats written to zknode every 15 secs

OVERLOADED ZOOKEEPER

Scale up



Increased to 1200 workers per cluster

OVERLOADED ZOOKEEPER

Analyzing zookeeper traffic

67%

KAFKA SPOUT

Offset/partition is written every 2 secs

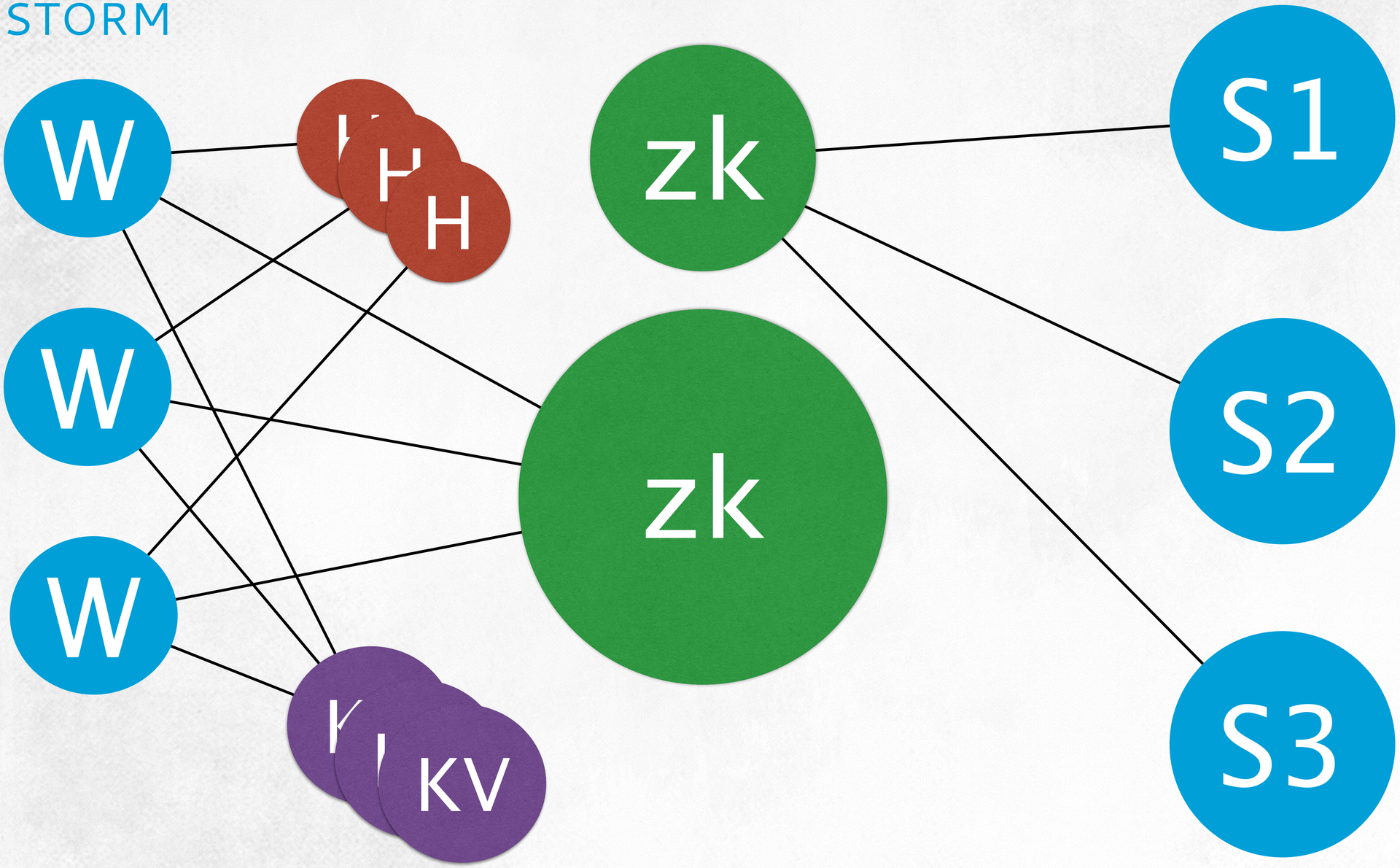
33%

STORM RUNTIME

Workers write heart beats every 3 secs

OVERLOADED ZOOKEEPER

Heart beat daemons



5000 workers per cluster and still growing!

STORM OVERHEADS

EXPT 1

JAVA PROGRAM

Read from Kafka cluster and deserialize in a "for loop"

Sustain input rate of 300K msgs/sec from Kafka topic

EXPT 2

1-STAGE TOPOLOGY

No acks to achieve at most once semantics

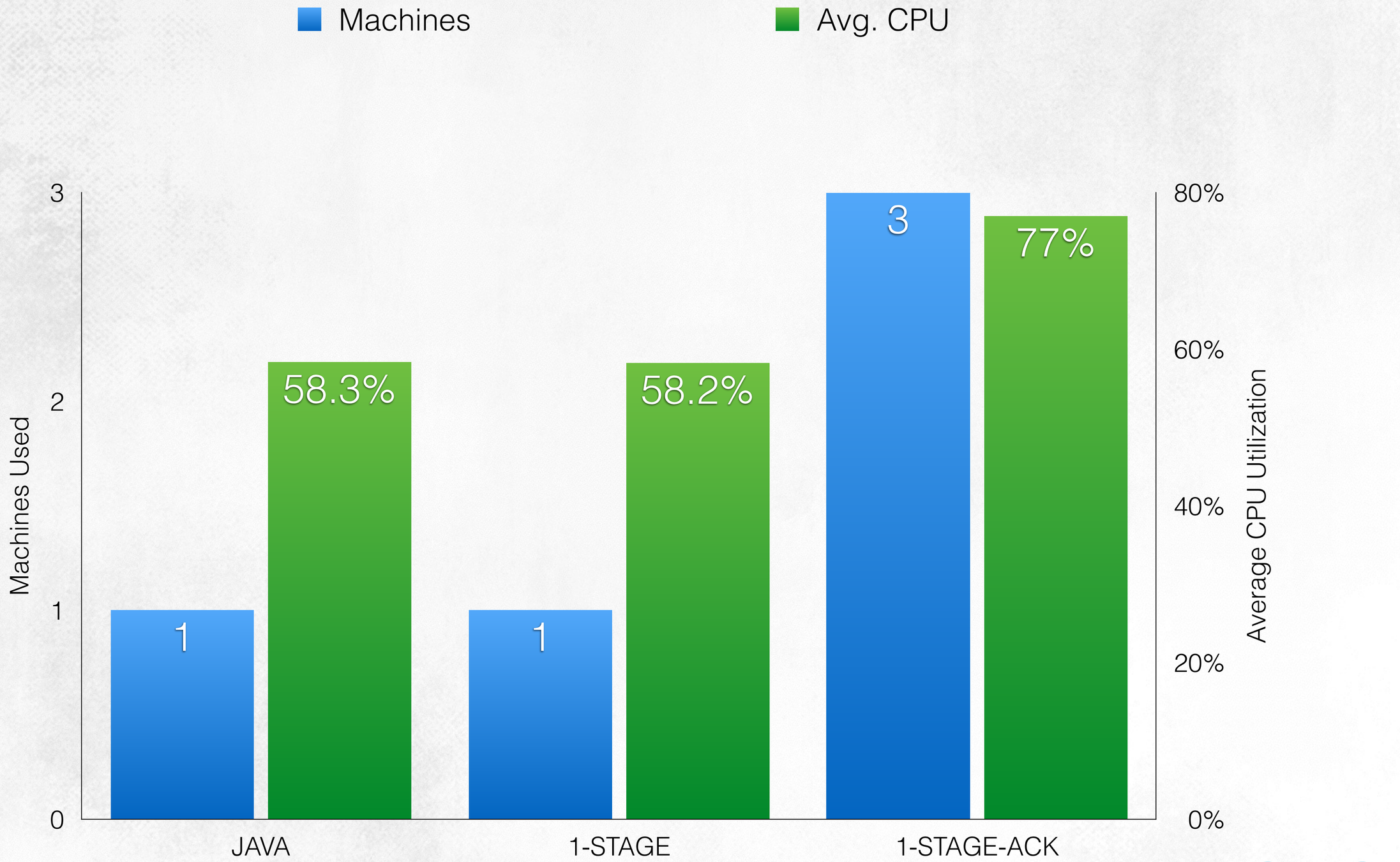
Storm processes were co-located using isolation scheduler

EXPT 3

1-STAGE TOPOLOGY WITH ACKS

Enable acks for at least once semantics

STORM OVERHEADS

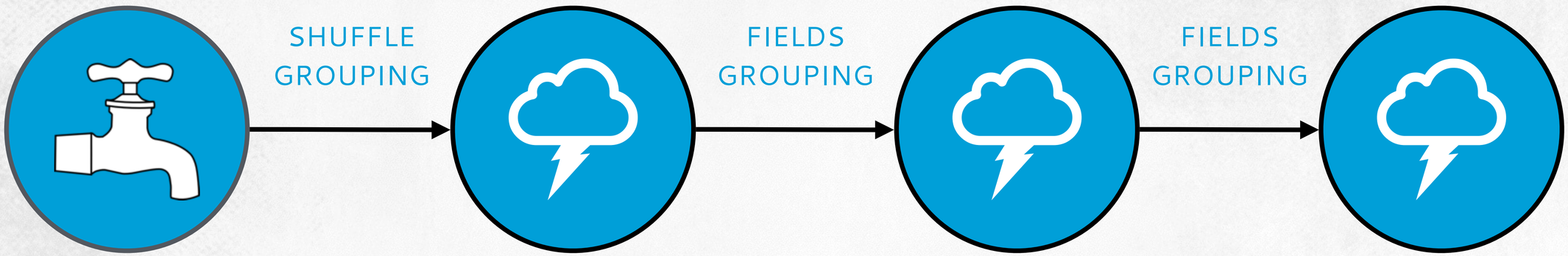


STORM EXPERIMENTS

V

STORM EXPERIMENTS

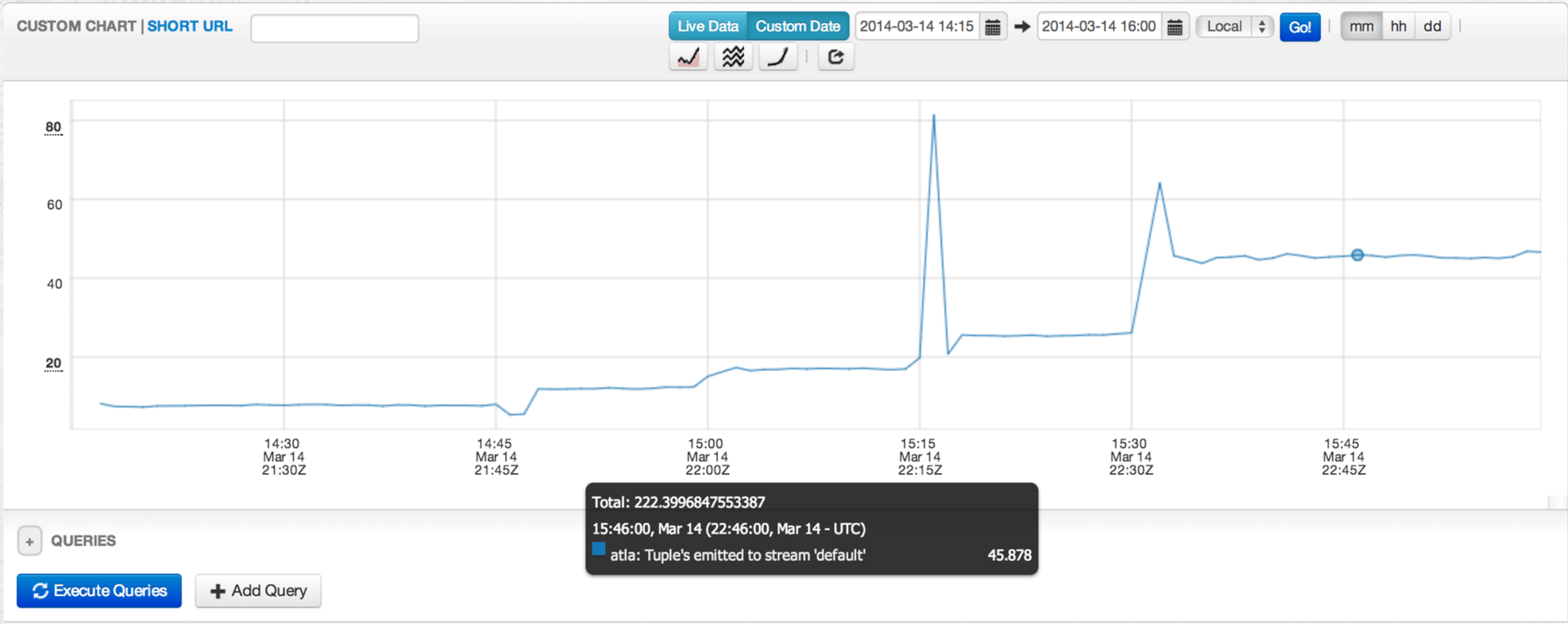
Examine resiliency and efficiency during machine failures



COMPONENTS	# TASKS
client event spout	200
distributor bolt	200
user count bolt	300
aggregator bolt	20

STORM THROUGHPUT

STORM LATENCY



#ThankYou

FOR LISTENING

QUESTIONS AND ANSWERS

Go ahead. Ask away.