Beyond Analytics

The Evolution of Stream Processing Systems

Introduction & Fundamentals

Paris Carbone, Marios Fragkoulis, Vasiliki Kalavri, Asterios Katsifodimos

Slides: **streaming-research.github.io/Tutorial-SIGMOD-2020**

Tutorial overview

- Part I: Introduction & Fundamentals (Vasia)
- Part II: Time, Order, & Progress (Marios)
- Part III: State Management (Paris)
- Part IV: Fault Recovery & High Availability (Marios)
- Part V: Load Management & Elasticity (Vasia)
- Part VI: Prospects (All)

What this tutorial is *not* about

- Synopses, sketches, and streaming algorithms
- Languages, operator semantics, windowing models
- Complex event processing
- Publish/Subscribe and message queues

Why this tutorial?

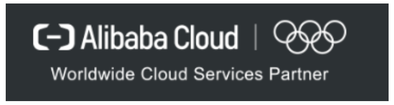
And why now?

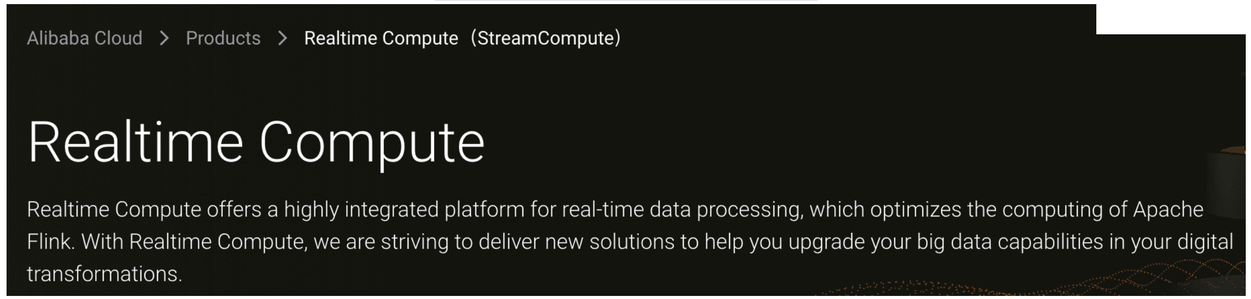
- To review and highlight noteworthy past research findings
- To compare early and modern systems
- To turn our community's attention to recent trends:
  - Streaming use-cases beyond traditional analytics
  - Requirements of emerging applications
  - Open problems

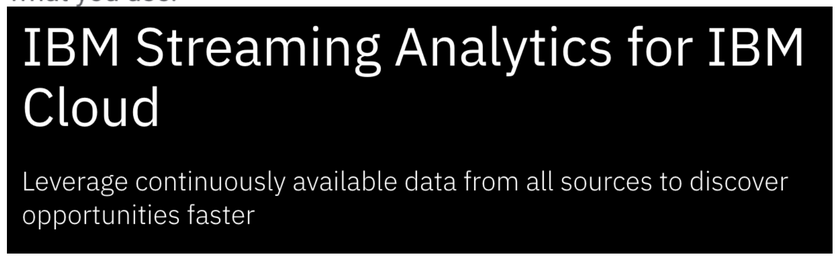
Stream processing is an established technology in the data analytics stack of the modern business

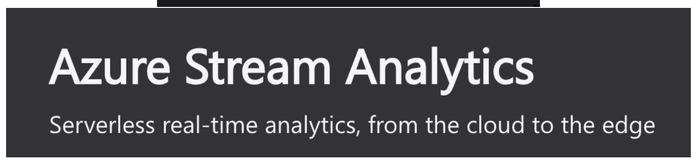
Dataflow

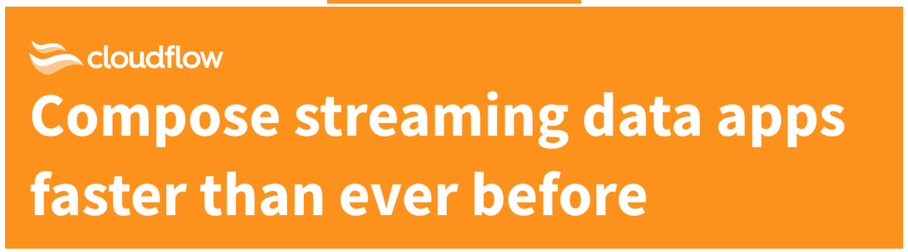
Dataflow is a fully managed streaming analytics service that minimizes latency, processing time, and cost through autoscaling and batch processing. With its serverless approach to resource provisioning and management, you have access to virtually limitless capacity to solve your biggest data processing challenges, while paying only for what you use.











Architecture Open Source

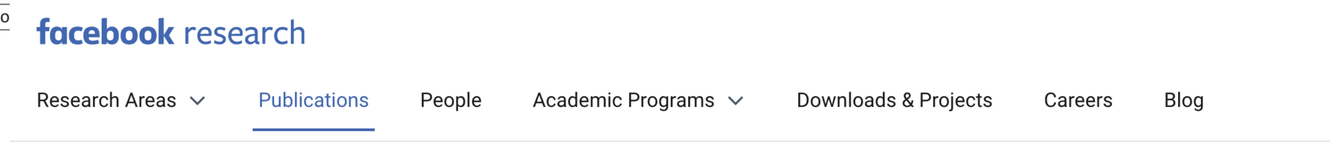
Introducing AthenaX, Uber Engineering's Open Source Streaming Analytics Platform

Haohui Mai, Bill Liu, and Naveen Cherukuri October 9, 2017

Keystone Real-time Stream Processing Platform

Top highlight

Netflix Technology Blog Follo
Sep 10, 2018 · 13 min read



June 25, 2016

Realtime Data Processing at Facebook

ACM SIGMOD

By: Guoqiang Jerry Chen, Janet Wiener, Shridhar Iyer, Anshul Jaiswal, Ran Lei, Nikhil Simha, Wei Wang, Kevin Wilfong, Tim Williamson, Serhat Yilmaz


Flink


APACHE
kafka
A distributed streaming platform


STORM


beam


Spark


samza


APACHE
nifi


HERON

Alibaba City Brain analyzes vehicle locations to:

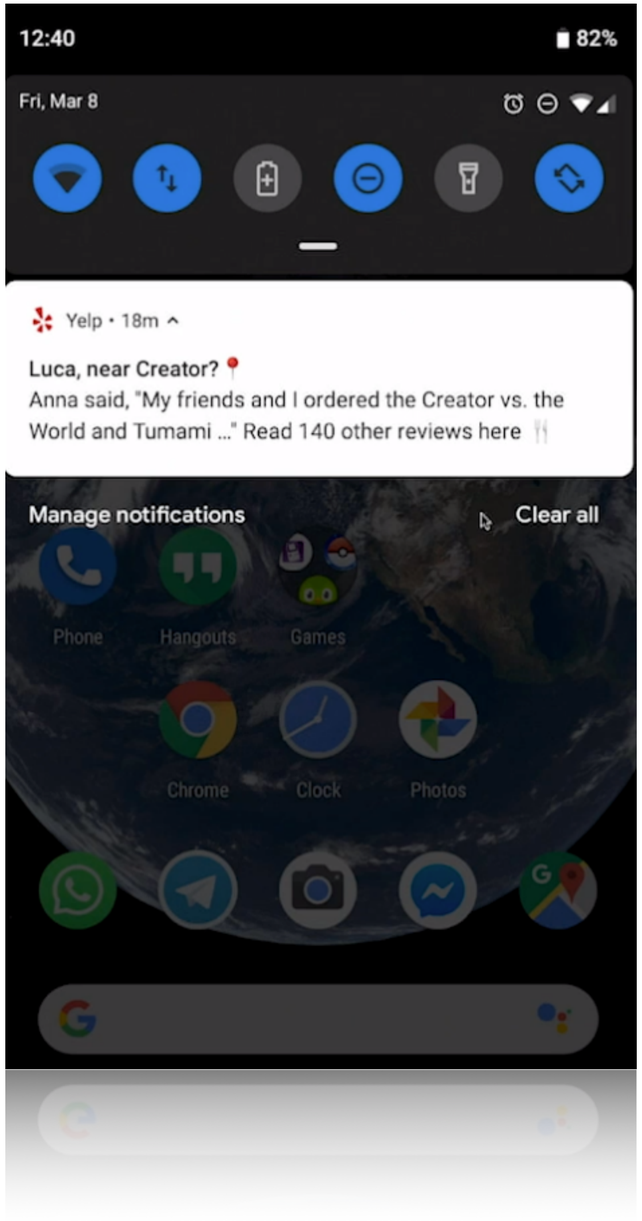
- clear paths for emergency response vehicles
- provide scheduling information for public transport
- recommend alternative routes

Traffic light adjustment in real time

Read more: **https://edition.cnn.com/2019/01/15/tech/alibaba-city-brain-hangzhou/index.html**

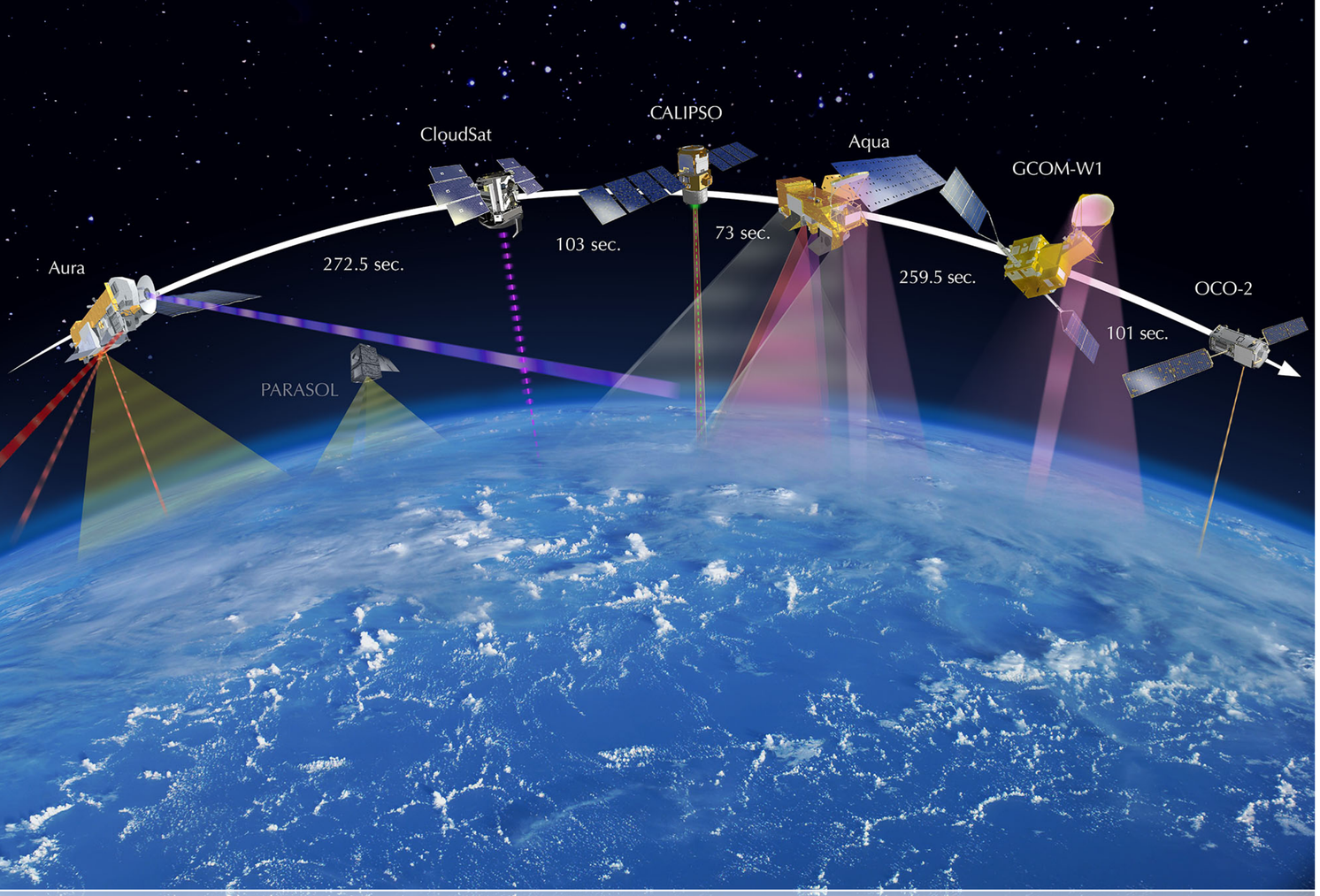
Yelp analyzes geospatial location data from mobile users and historical profile information to:

- suggest the best nearby local businesses
- ensure that any push notification sent to a user is timely and relevant



Store visits prediction at Yelp

Read more: https://www.ververica.com/blog/how-yelp-uses-flink-for-predicting-store-visits-in-real-time



Fault-detection for NASA’s Deep Space Network

Read more: https://www.confluent.io/kafka-summit-san-francisco-2019/mission-critical-real-time-fault-detection-for-nasas-deep-space-network-using-apache-kafka/

NASA’s DSN Complex Event Processing analyzes real-time network data, predicted antenna pointing parameters, and physical hardware logs to:

- ingest, filter, store, and visualize all of the DSN's monitor and control data
- ensure the successful DSN tracking, ranging, and communication integrity of dozens of concurrent deep-space missions

Evolution of Stream Processing Systems

What is a stream?

A data set that is produced incrementally over time, rather than being available in full before its processing begins.

What is a stream?

A data set that is produced incrementally over time, rather than being available in full before its processing begins.

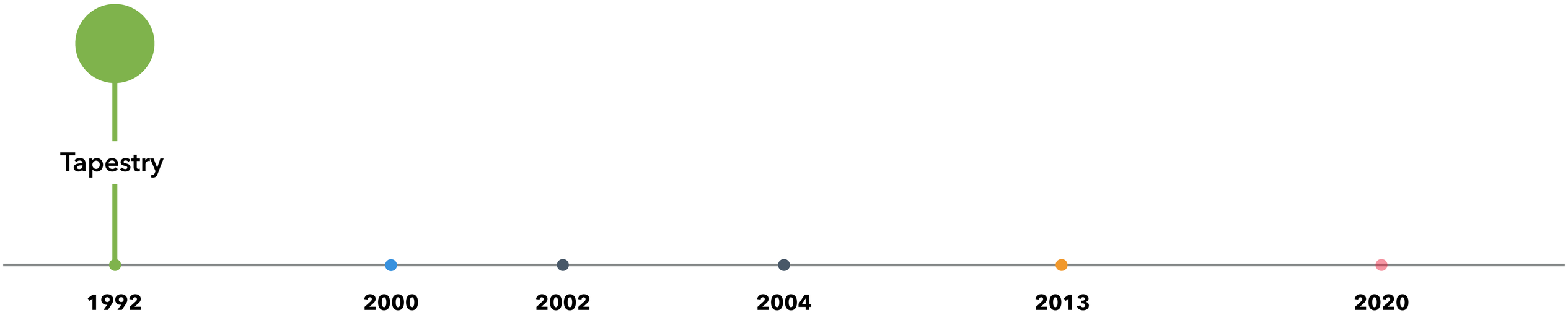
- Data streams have **unknown**, possibly **unbounded length**
- They bear an arrival and/or a generation **timestamp**
- They are produced by external sources, i.e. the system has **no control** over their **arrival order** or the **data rate**

SIGMOD '92

Continuous Queries over Append-Only Databases

Douglas Terry, David Goldberg, David Nichols, and Brian Oki

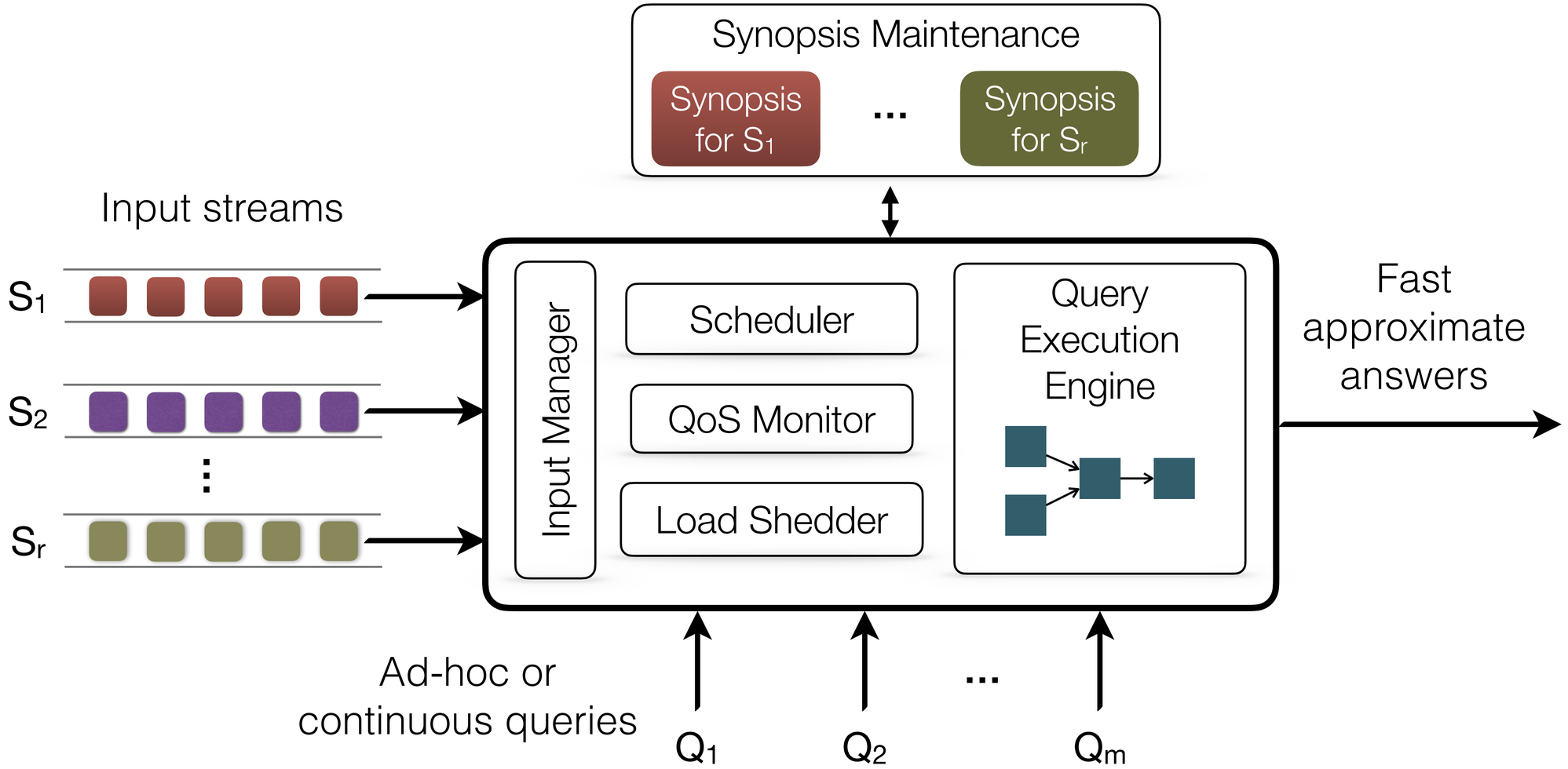
[... A new class of queries, **continuous queries**, are similar to conventional database queries, except that they are issued once and henceforth run "continually" over the database ...]


Tapestry
1992
2000
2002
2004
2013
2020


Tapestry
NiagaraCQ
Aurora
STREAM
TelegraphCQ
Gigascope
1992
2000
2002
2004
2013
2020

Data Stream Management Systems

DSMS architecture



Data Stream Management Systems

Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

Data Stream Management Systems

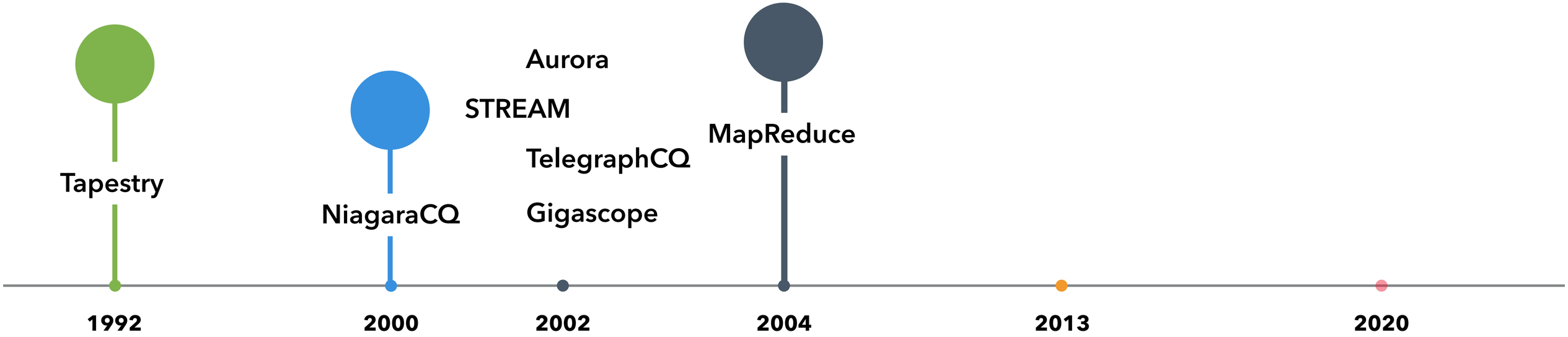
representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling

Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

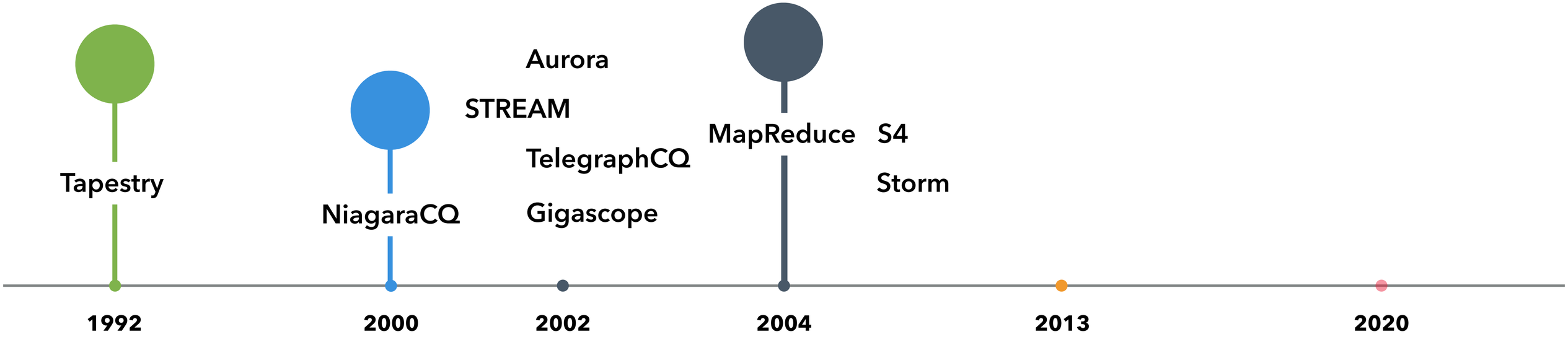
load management
high availability
scheduling



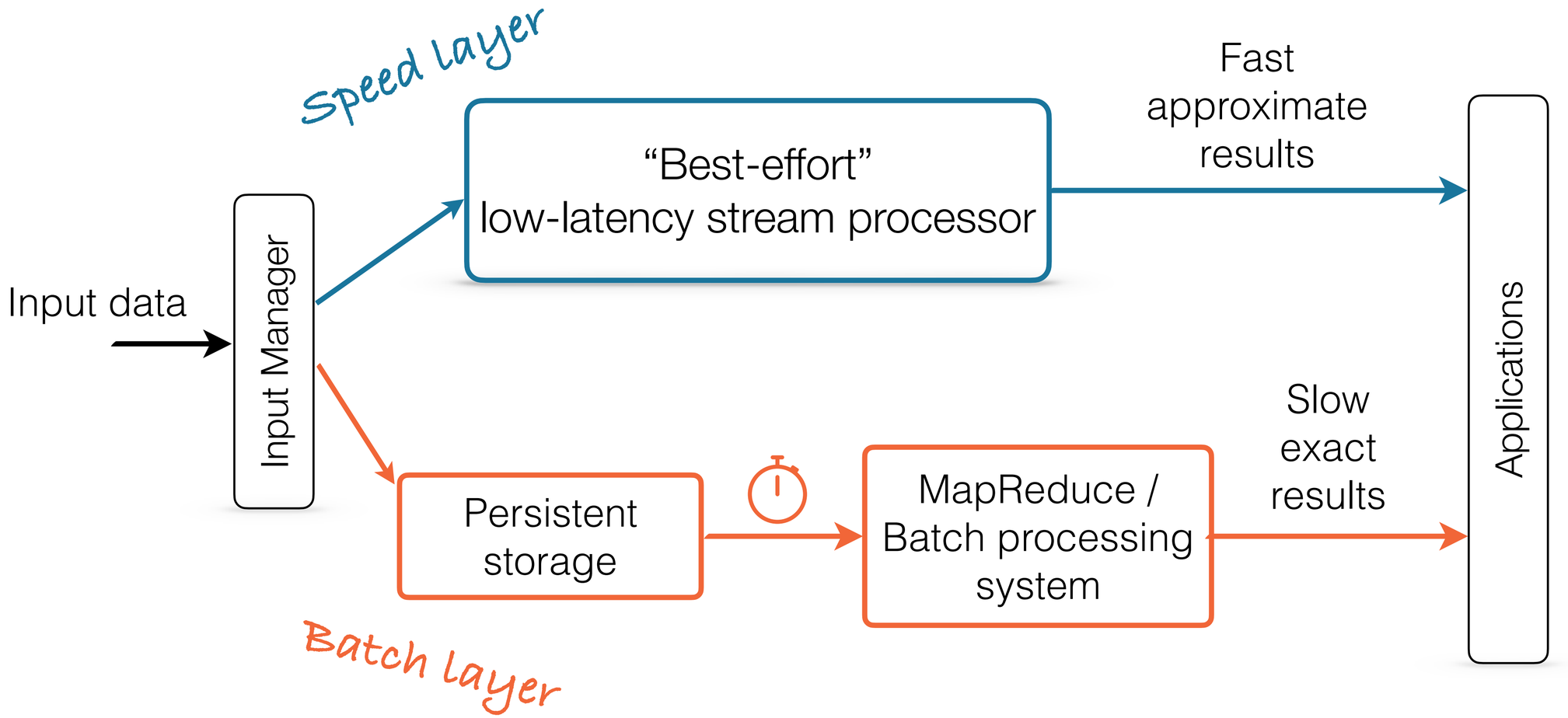
Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling



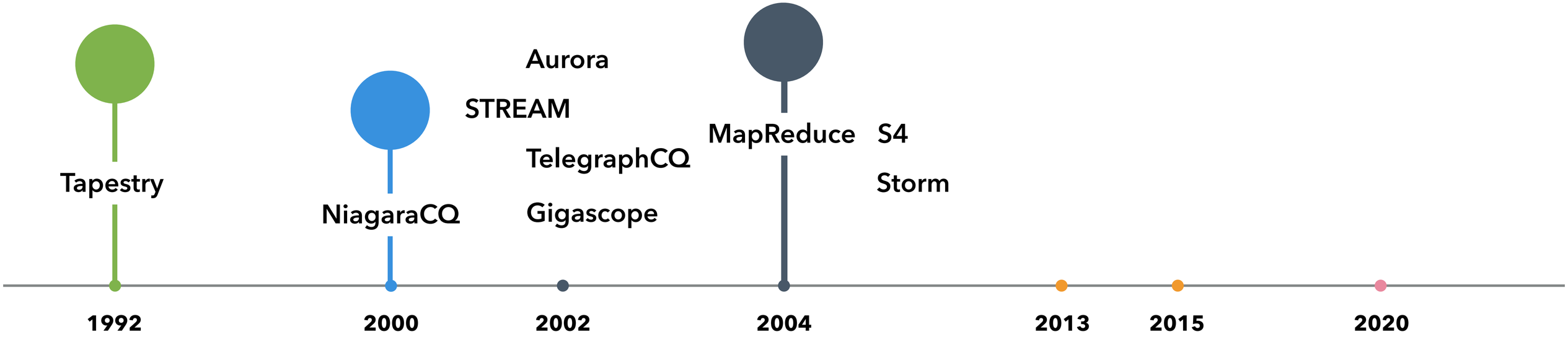
λ-architecture



Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling

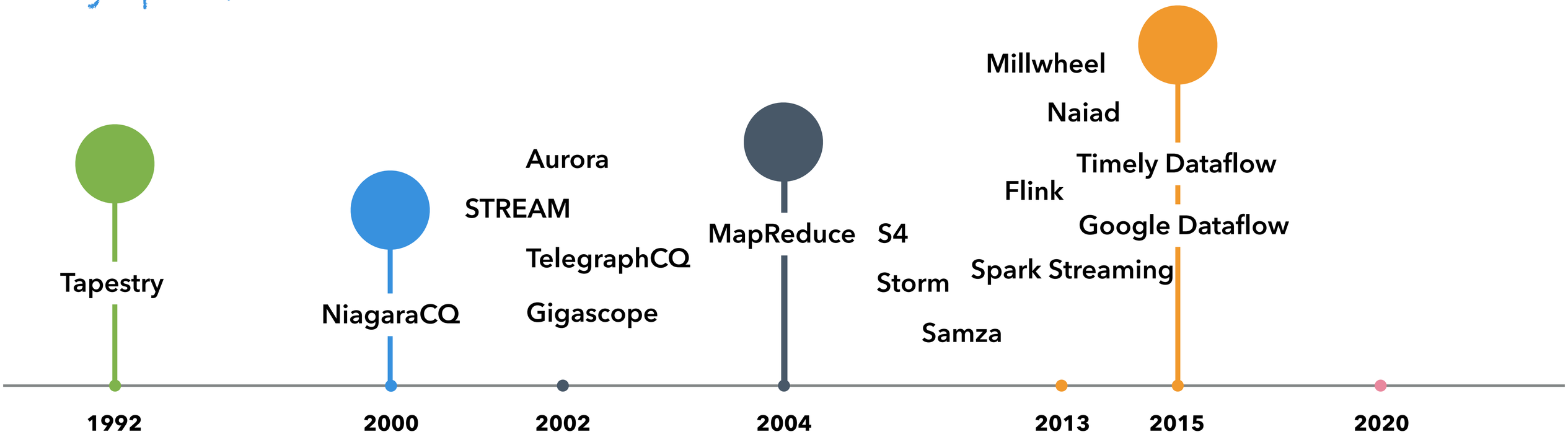


Data Stream Management Systems

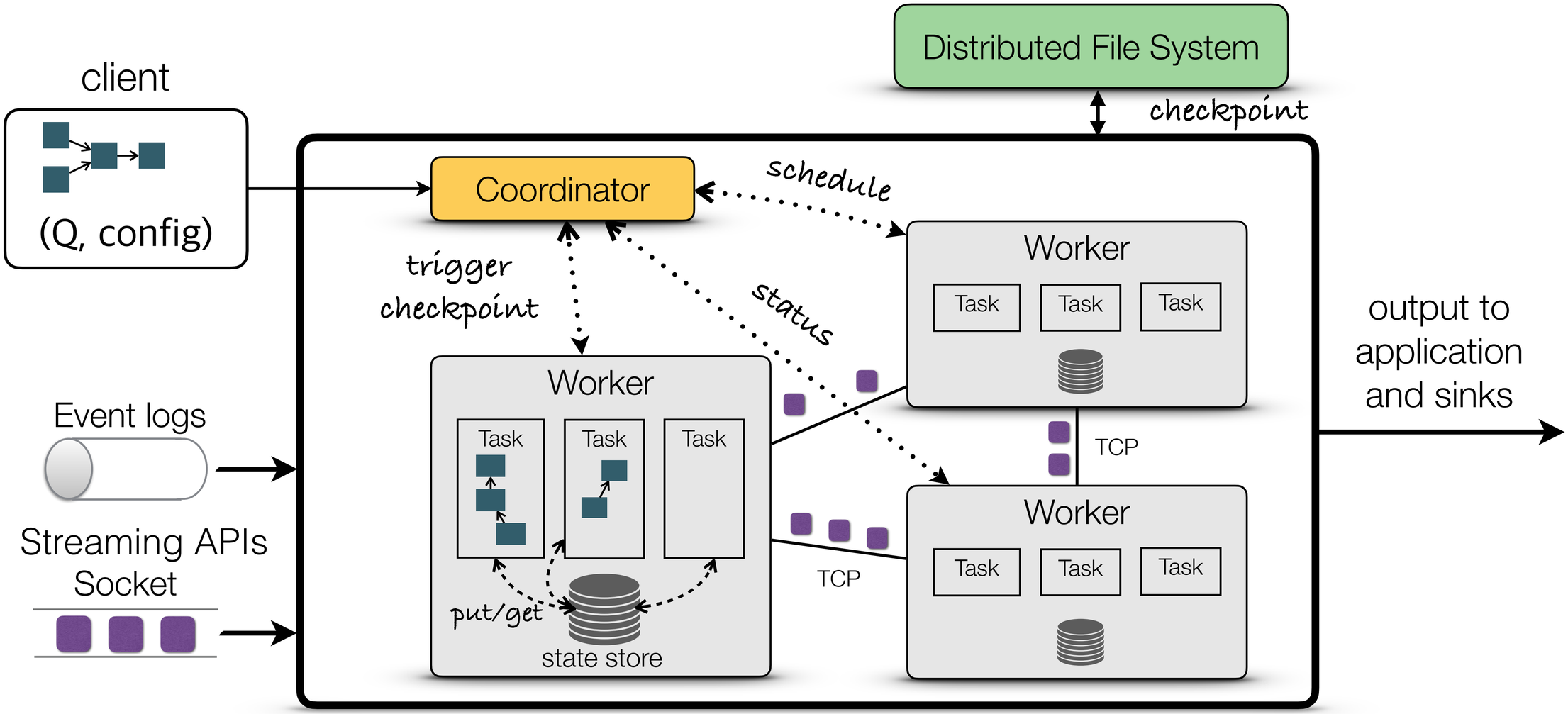
representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling

Distributed Dataflow Systems

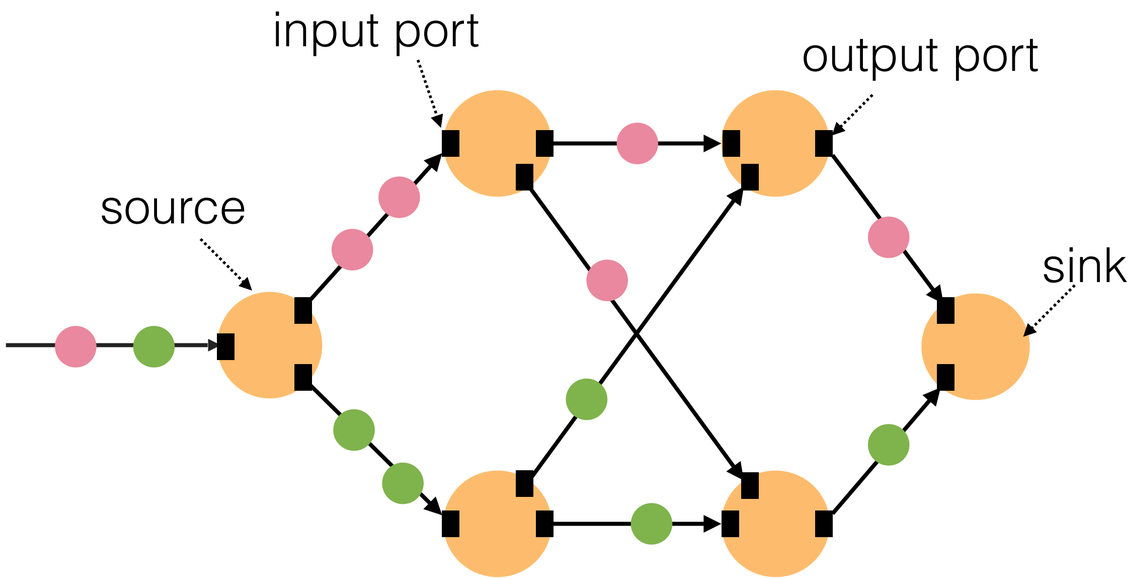


Dataflow System architecture



A series of transformations
on streams in
Stream SQL, Scala, Python,
Rust, Java…

dataflow graph

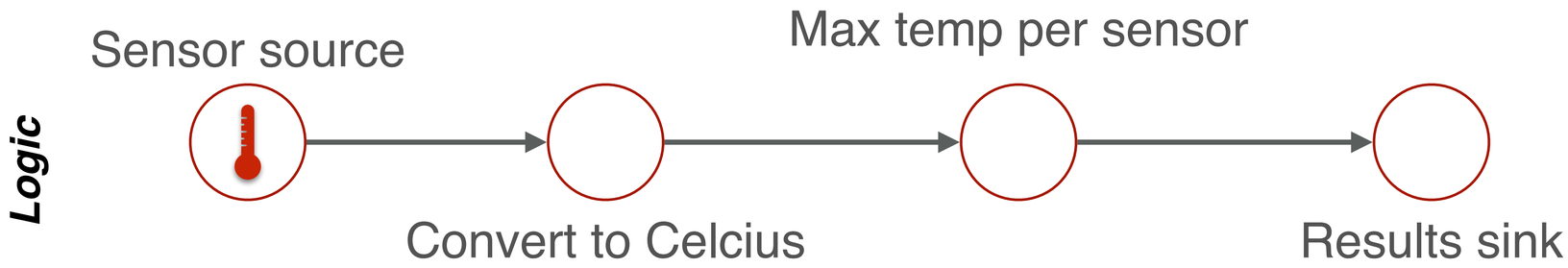


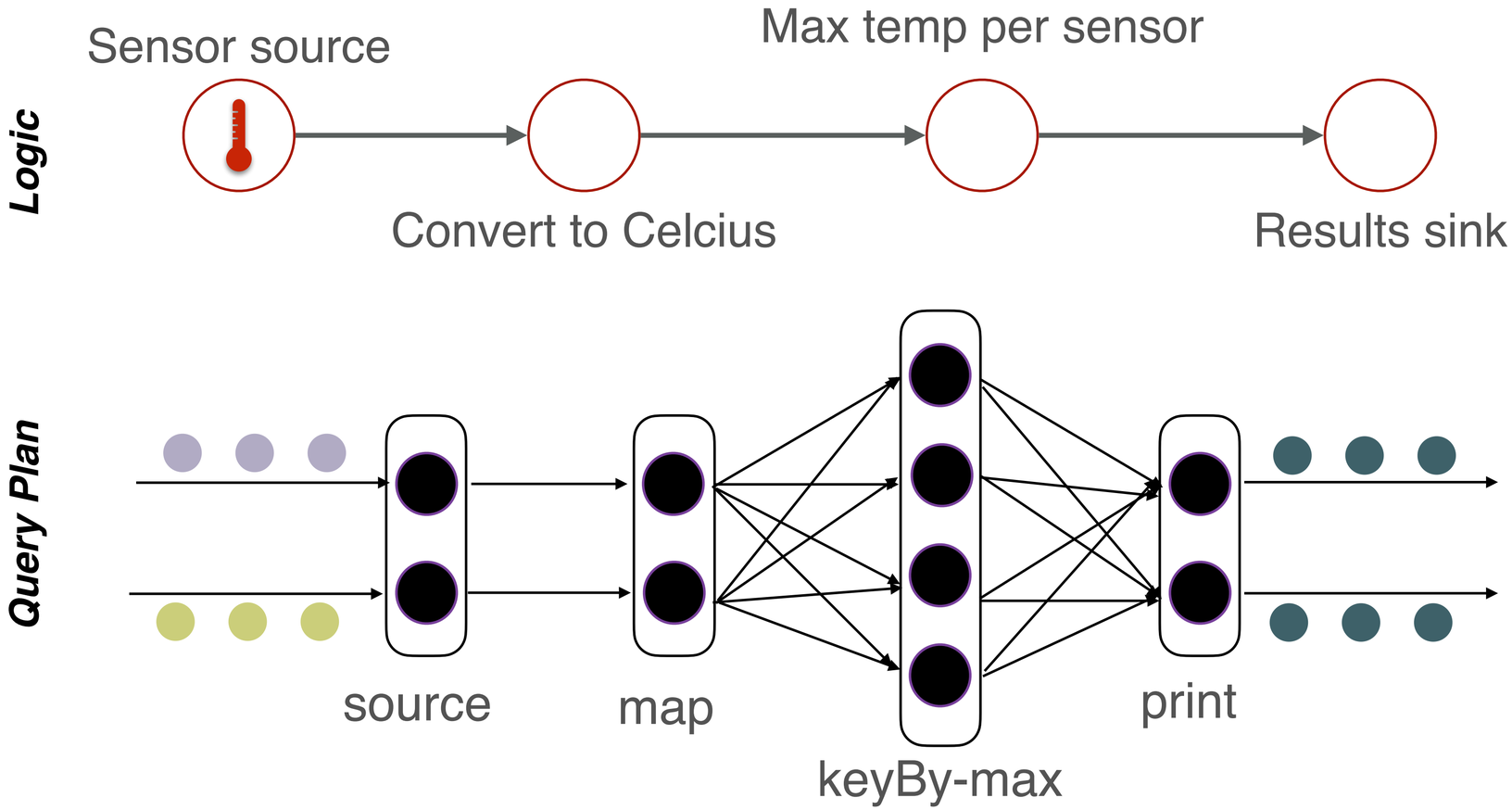
Dataflow graph

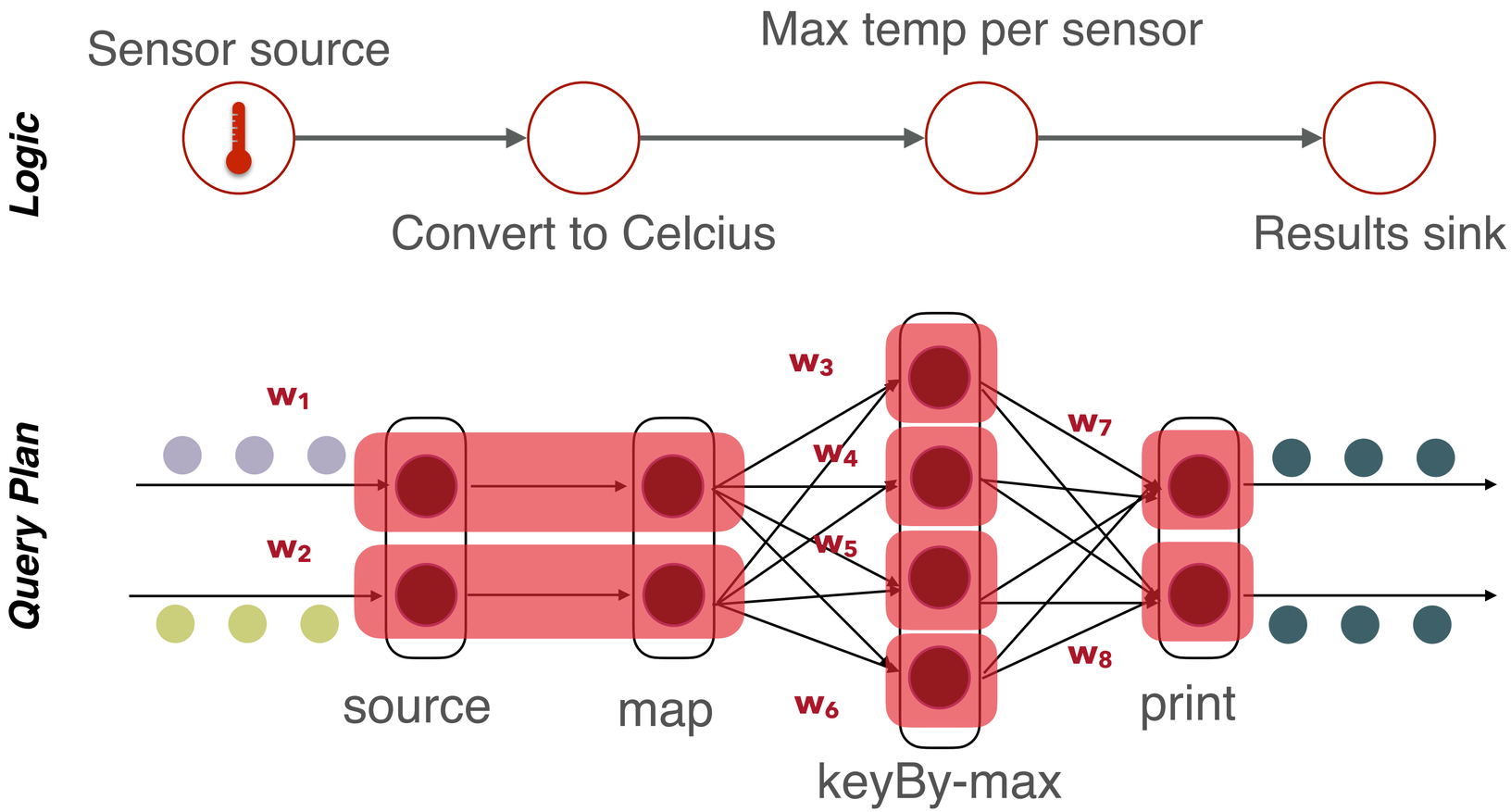
- operators are nodes, data channels are edges
- channels have FIFO semantics
- streams of data elements flow continuously along edges

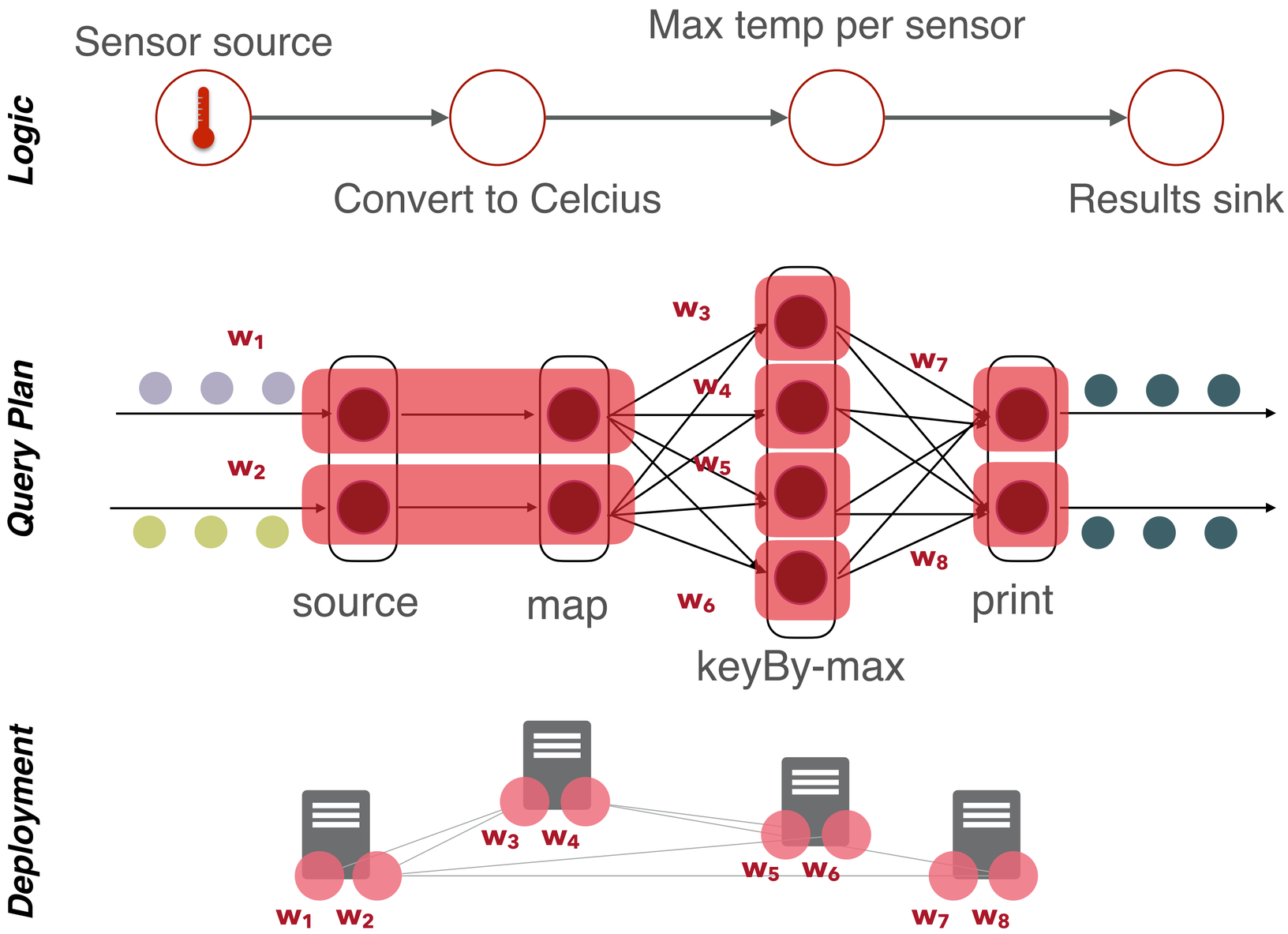
Operators

- receive one or more input streams
- perform tuple-at-a-time, window, logic, pattern matching transformations
- output one or more streams of possibly different type


Logic
Sensor source
Max temp per sensor
Convert to Celcius
Results sink


Logic
Sensor source
Max temp per sensor
Convert to Celcius
Results sink
Query Plan
source
map
keyBy-max
print


Logic
Sensor source
Max temp per sensor
Convert to Celcius
Results sink
Query Plan
w1
w2
w3
w4
w5
w6
w7
w8
source
map
keyBy-max
print

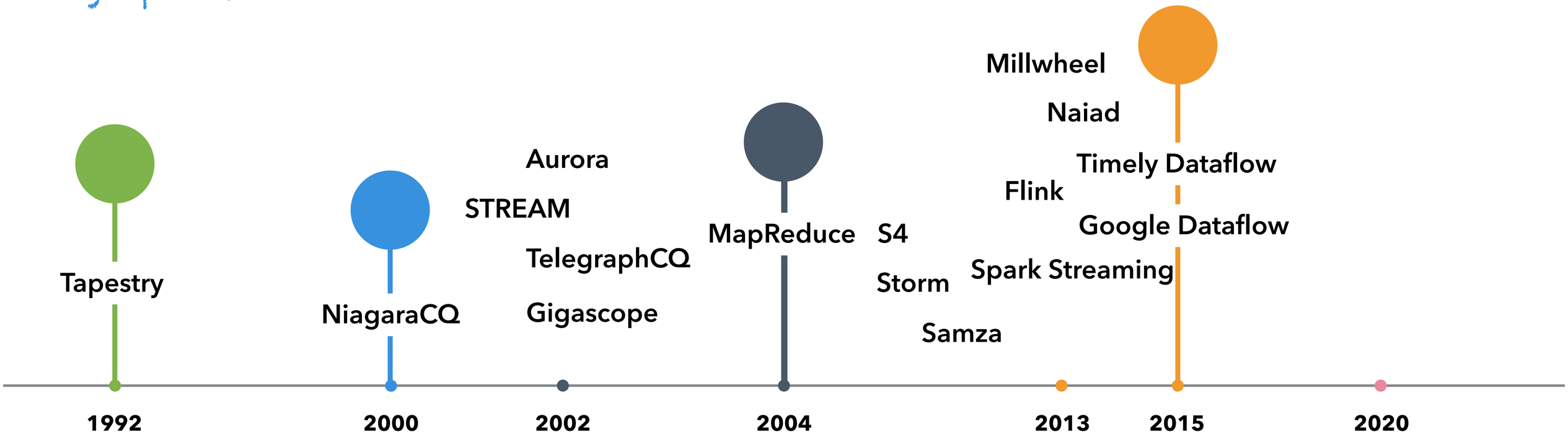

Logic
Sensor source
Max temp per sensor
Convert to Celcius
Results sink
Query Plan
w1
w2
w3
w4
w5
w6
w7
w8
source
map
keyBy-max
print
Deployment
w1 w2
w3 w4
w5 w6
w7 w8

Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling

Distributed Dataflow Systems



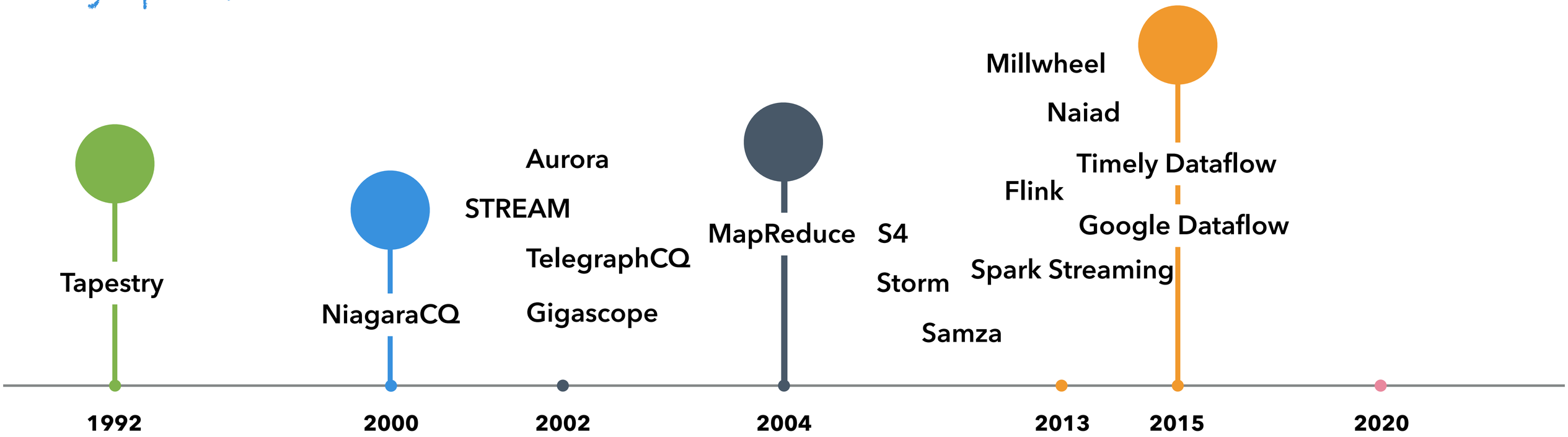
Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling

Distributed Dataflow Systems

data parallelism state management
exactly-once fault-tolerance UDFs
iterations general-purpose languages



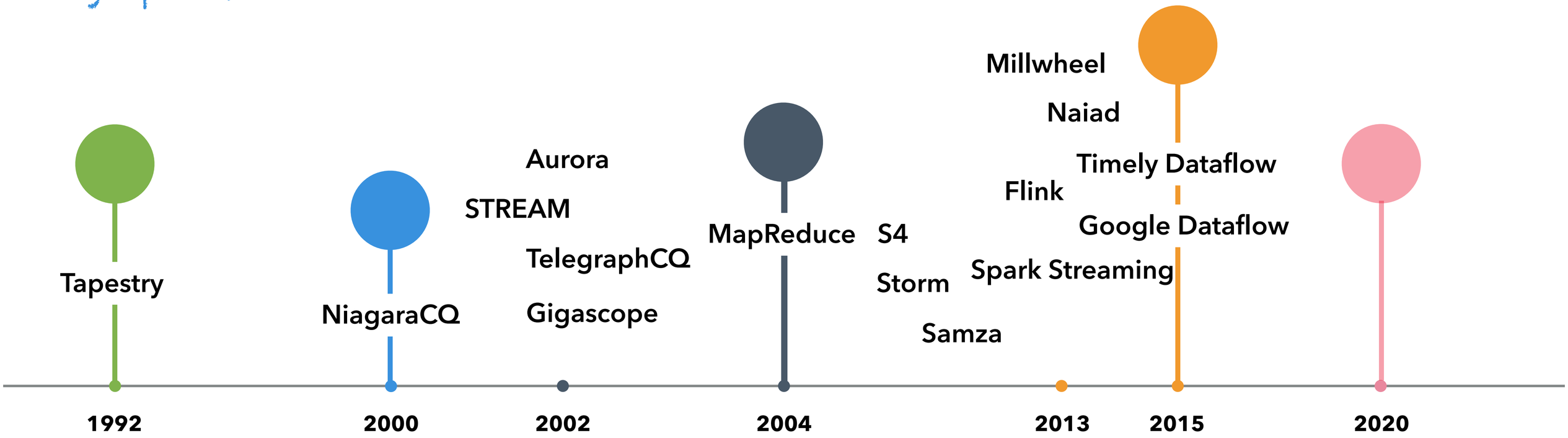
Data Stream Management Systems

representations
operator semantics
event time & progress
synopses & sketches

load management
high availability
scheduling

Distributed Dataflow Systems

data parallelism state management
exactly-once fault-tolerance UDFs
iterations general-purpose languages



Vintage vs. modern

	DSMS	Distributed Dataflow
Input	in-order	out-of-order
Results	exact or approximate	exact
Language	SQL extensions, CQL	Java, Scala, Python, SQL-like
Query plans	global, optimized, with pre-defined operators	independent, with custom operators
Execution	centralized	distributed
Parallelism	pipeline	data, pipeline, task
Time & progress	heartbeats, slack, punctuations	low watermarks, frontiers
State	shared synopses, in-memory	per-query, partitioned, persistent, larger-than-memory
Fault tolerance	HA-focused, limited correctness guarantees	distributed snapshots, exactly-once
Load management	load shedding, load-aware scheduling	backpressure, elasticity

Vintage vs. modern

	DSMS	Distributed Dataflow
Input	in-order	out-of-order
Results	exact or approximate	exact
Language	SQL extensions, CQL	Java, Scala, Python, SQL-like
Query plans	global, optimized, with pre-defined operators	independent, with custom operators
Execution	centralized	distributed
Parallelism	pipeline	data, pipeline, task
Time & progress	heartbeats, slack, punctuations	low watermarks, frontiers
State	shared synopses, in-memory	per-query, partitioned, persistent, larger-than-memory
Fault tolerance	HA-focused, limited correctness guarantees	distributed snapshots, exactly-once
Load management	load shedding, load-aware scheduling	backpressure, elasticity

Beyond Analytics

The Evolution of Stream Processing Systems

Introduction & Fundamentals

Paris Carbone, Marios Fragkoulis, Vasiliki Kalavri, Asterios Katsifodimos

Slides: **streaming-research.github.io/Tutorial-SIGMOD-2020**