Ztreamy :
A middleware for publishing semantic streams on the Web*

Contiu Stefan, IVA, Anul I

UTCN

* Original article by: Jesus Arias Fisteus, Norberto Fernandez Garcia, Luis Sanchez Fernandez, Damaris Fuentes-Lorenzo

Semantic Data Streams

- There are well-known best practices for publishing *static* linked data on the Web, but not for time-dependent data about the physical world, processed in the form of streams.
- **Data Stream**: real-time, continuous, ordered sequence of items.
- Data Streams vs Stored data:
  - Can not be stored in their entirety.
  - Order in which the data is received cannot be controlled.
- To facilitate data integration:
  - Wrap sensor data into semantic formats like RDF.
  - Extend it with semantic annotations.
- Applications can exploit information by integrating sensor data and linking to sources of static information.
- This platform is called the ***semantic sensor Web***.

The Ztreamy Platform

- A middleware for publishing streams of semantically-annotated data on the Web.
- Data sources can publish their streams, and applications can consume them.
- Provides operations for : Mirroring, joining, splitting and filtering streams.
- Similar frameworks: DataTurbine, Linked Stream Middleware(LSM), ZeroMQ.
- Ztreamy unique features: scalability and use of HTTP.

Requirements of Ztreamy (1)

1. Ability for applications **to consume** the streams of data they are interested in.
   - Providers should be able to logically group data into separate streams.
   - Client applications would subscribe just to the streams they are interested in.
2. Ability for any party **to publish** value-added streams by merging, splitting, filtering, processing, enriching with external data.
3. Ability **to scale** with the number of consumers and the data rate.

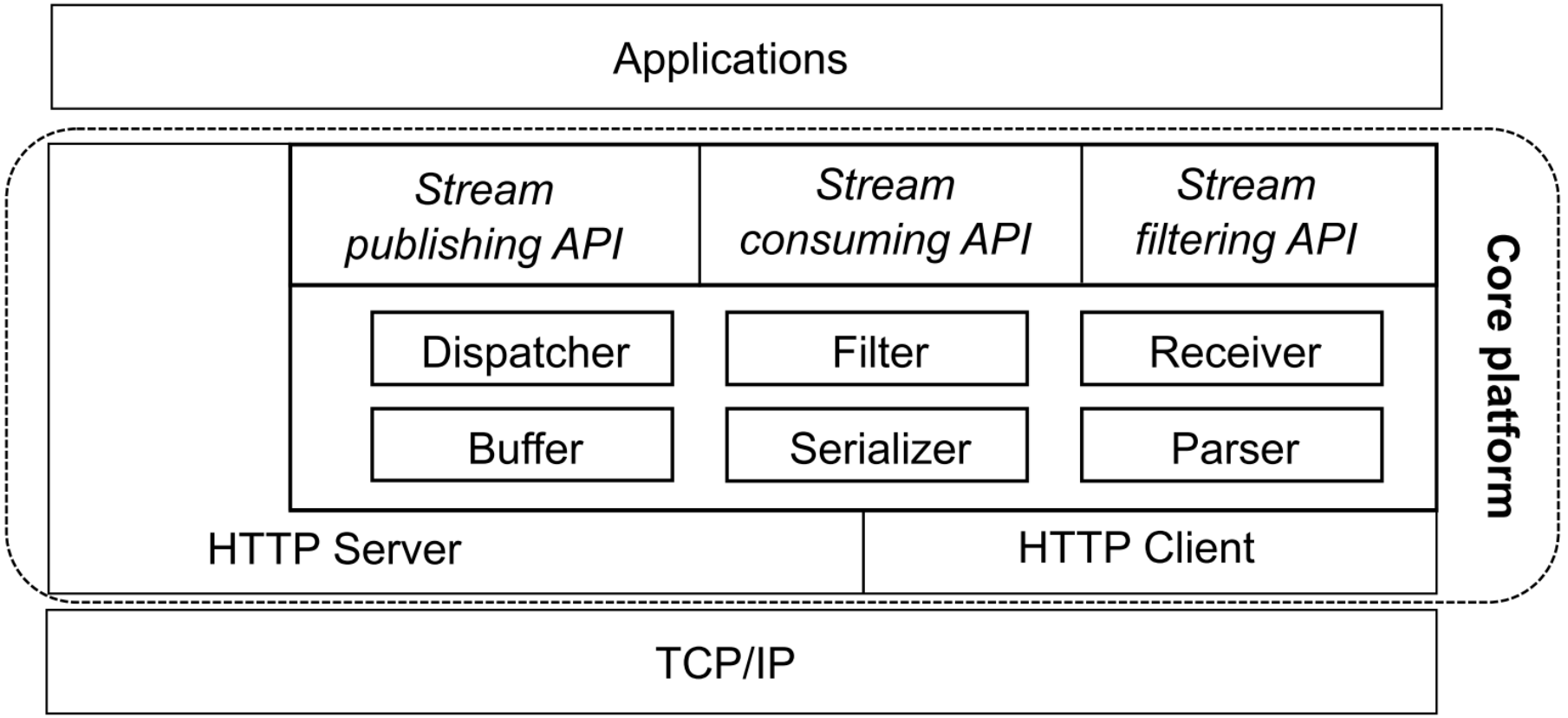
Requirments of Ztreamy (2)

4. **Accessibility** from a wide range of application environments.
   - applications for desktops, smartphones, Web browsers, etc. in almost any programming language.
5. Configurable **quality of service**, such as delivery reliability and latency.
   - consumers should not miss current items in the stream.
   - distribute data as quickly as possible.
6. Flexibility in **network layout**.
   - from simple client–server layouts to complex treelike distribution
7. **Semantic annotation**.
   - necessary for data interoperability and integration
   - the platform should support its use and provide facilities based on those semantics

Ztreamy Arhitecture

- Ztreamy acts as a middleware library for applications.
- APIs exposed for main functions:
  - Consuming streams
  - Publishing streams
  - Filtering streams
  - Sending items for publication in a stream
- All operations performed through HTTP (or HTTPS if security requirements).

Arhitecture Diagram



Architecture Details

- Making use of **Http Streaming** (long-lived requests) :
  - server sends an HTTP response in chunks but never finishes the response neither closes the connection.
  - Only one underlying TCP connection is used.
  - Only one HTTP request needs to be processed for a long period of time
- Improved performance by using **Buffering** :
  - Items stored in a buffer, periodically dumped to the network.
  - Reduces CPU consumption, improves compression ratios and reduces network load.
- Conventional **load balancing** techniques(due to use of HTTP)
- Using **stream compression** techniques to reduce the bandwidth usage on the server.

Performance Evaluation (1)

- Series of experiments with Ztreamy and other systems under various **number of clients** and various **data rates**.
- Performance Indicators:
  - CPU use : absolute amount of CPU time the server needed to process a given load.
  - Delivery delay : time between the source is ready to send an item and the client parses it.
- Timing of the stream items followed a Poisson process.
- Dataset consisted of posts captured from the micro-blogging site (Identica).
- Each experiment was repeated 10 times. The 95% confidence interval was computed for each data point.

Performance Evaluation (2)

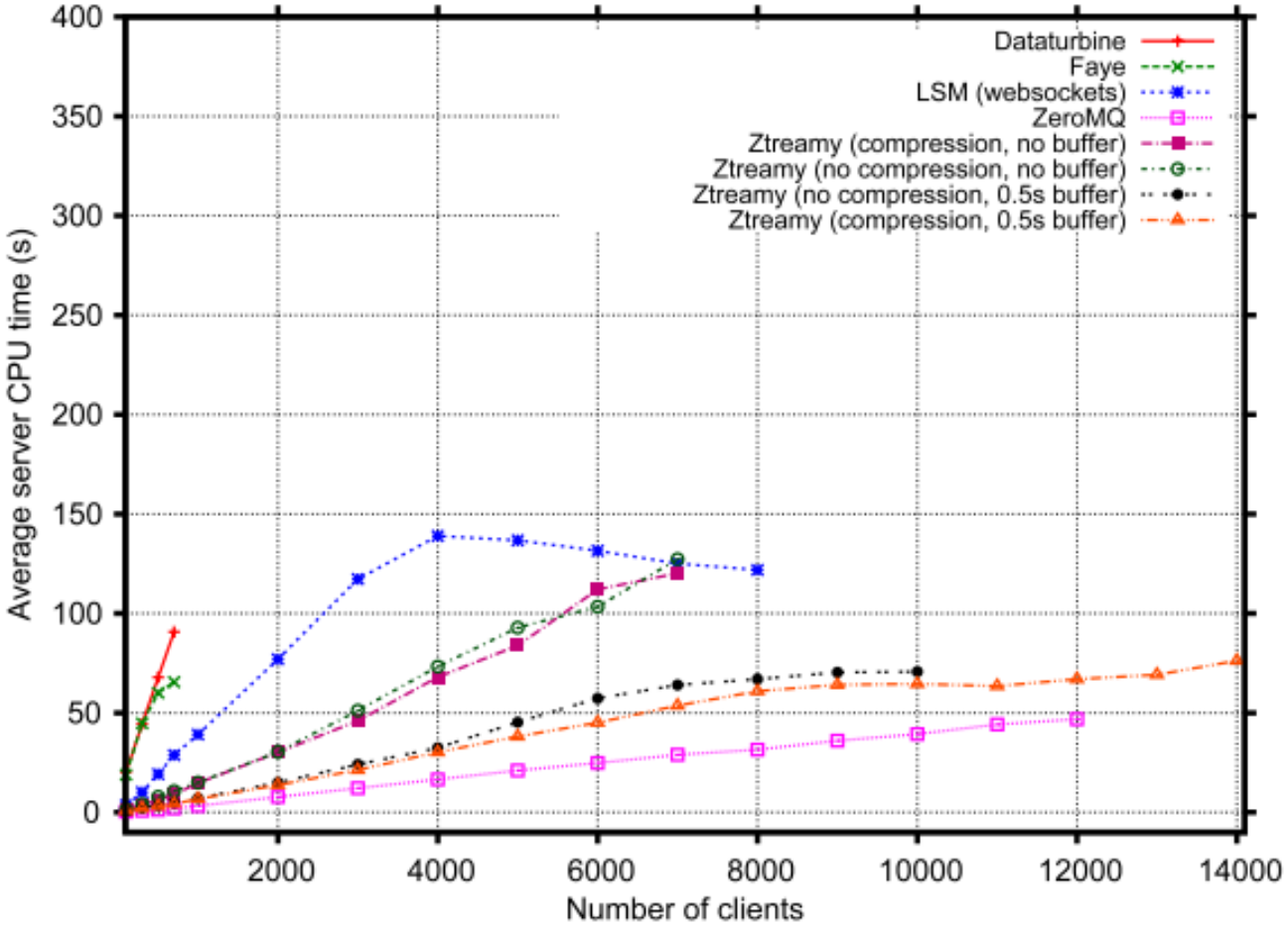
- Other systems choosed for comparison:
  - Dataturbine : a leading free software for sensor data delivery
  - Faye : HTTP publish-subscribe framework
  - ZeroMQ : messaging system used to distribute streams through the cluster(maintained by Twiter)
  - LSM : websockets-based publishing on top of Apache Tomcat server.
- Ztreamy run in four configurations:
  - with and without buffering
  - with and without compression

Perf Eval - Scenario #1

- **4 items/s stream served to a variable amount of clients.**

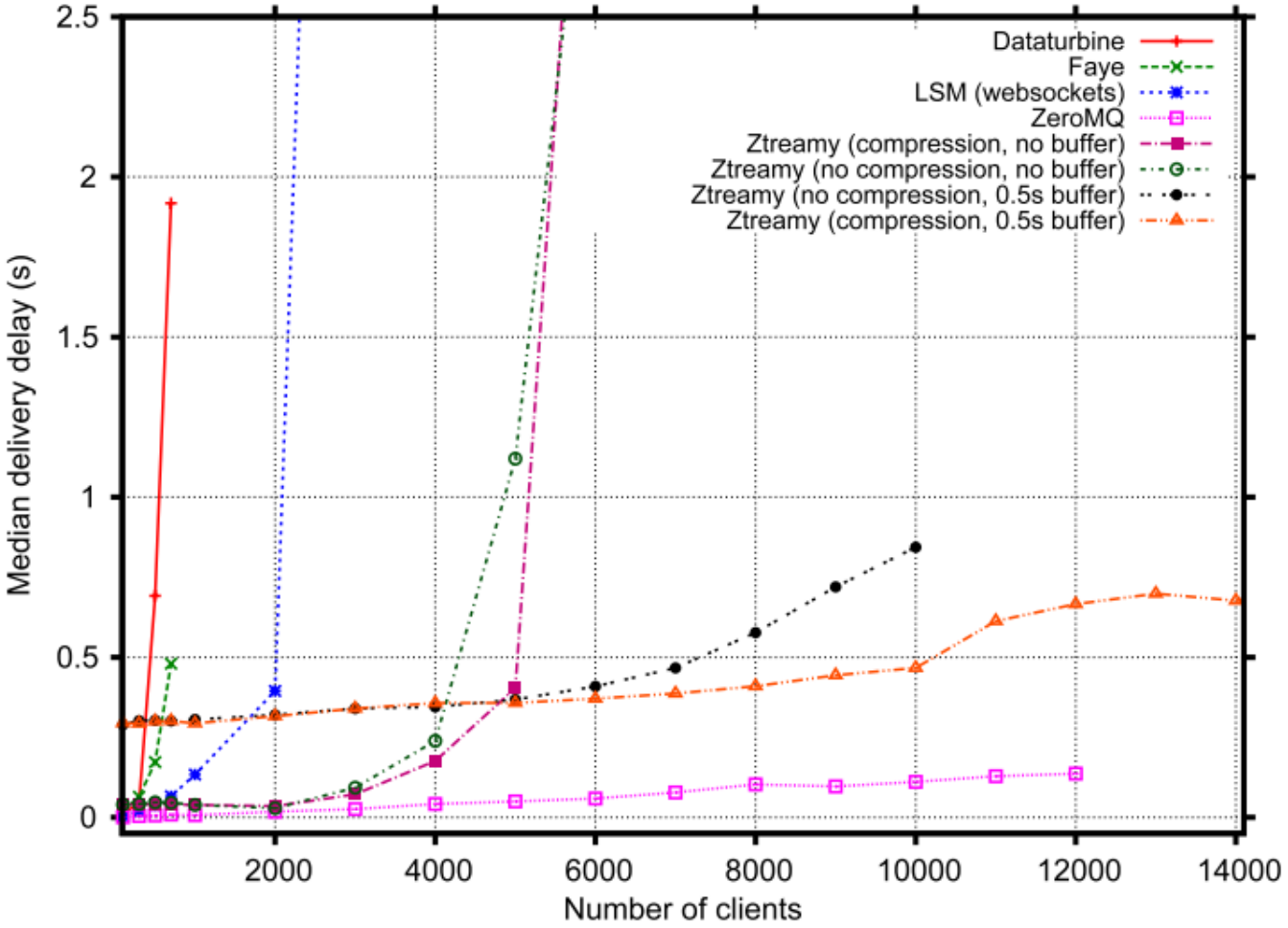
- DataTurbine and Faye stopped working properly at 700 clients.
- LSM suffered high delays beyond 2,000 clients, loosing data beyond 5,000 clients.
- ZeroMQ was unable to deliver about 8% of the items one minute after the source sent the data. (at 12,000 clients).
- **Ztreamy still delivered the stream at 40,000 clients with less than 3s delay and 90% CPU use.**

Scenario #1 : CPU Time



(a) CPU time needed to send 400 items at average 4 items/s to a variable number of simultaneous clients.

Scenario #1 : Delivery delays



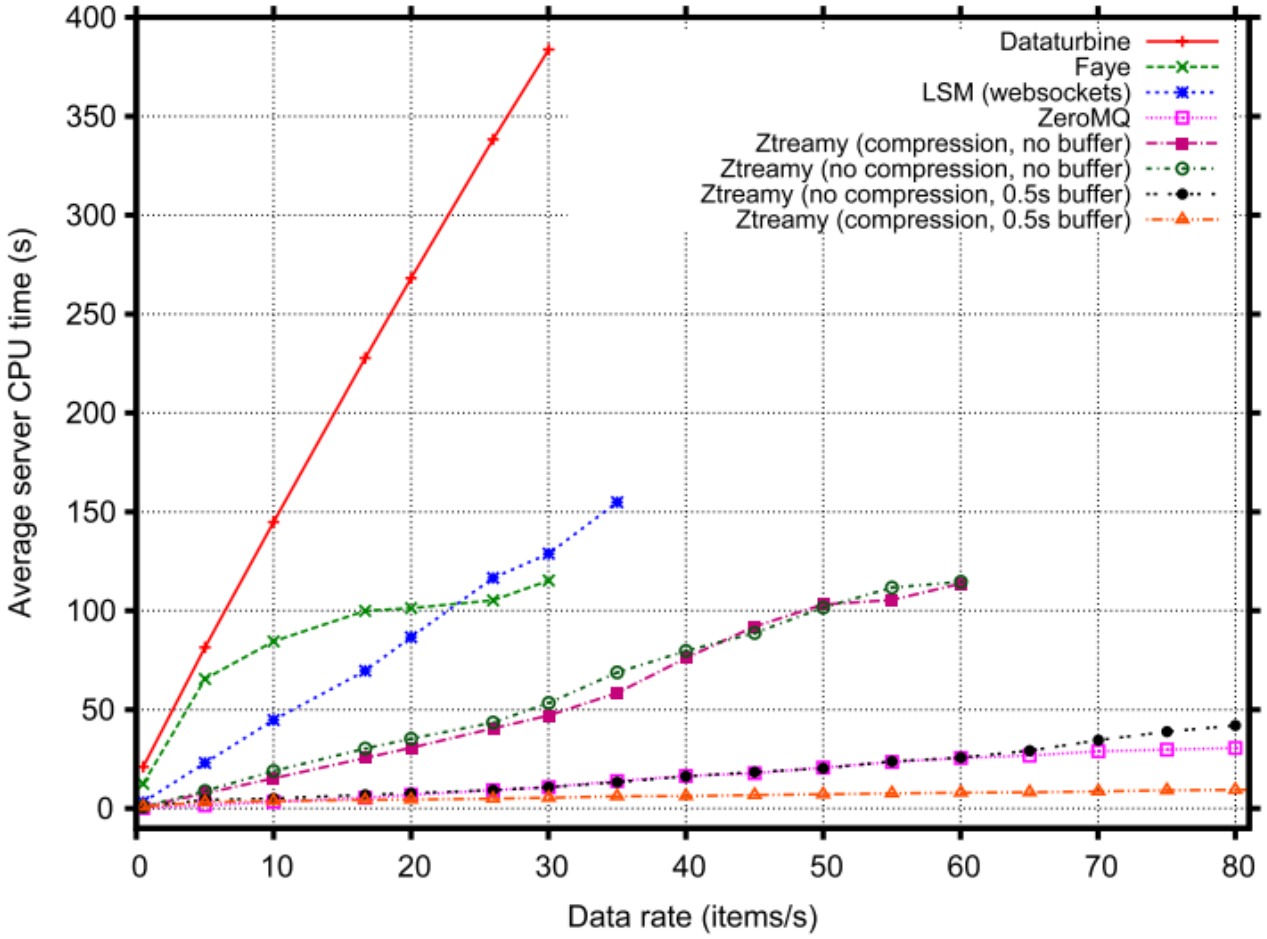
(c) Delivery delays when sending 400 items at average 4 items/s to a variable number of simultaneous clients.

Perf Eval - Scenario #2

- **500 clients and a variable data rate.**

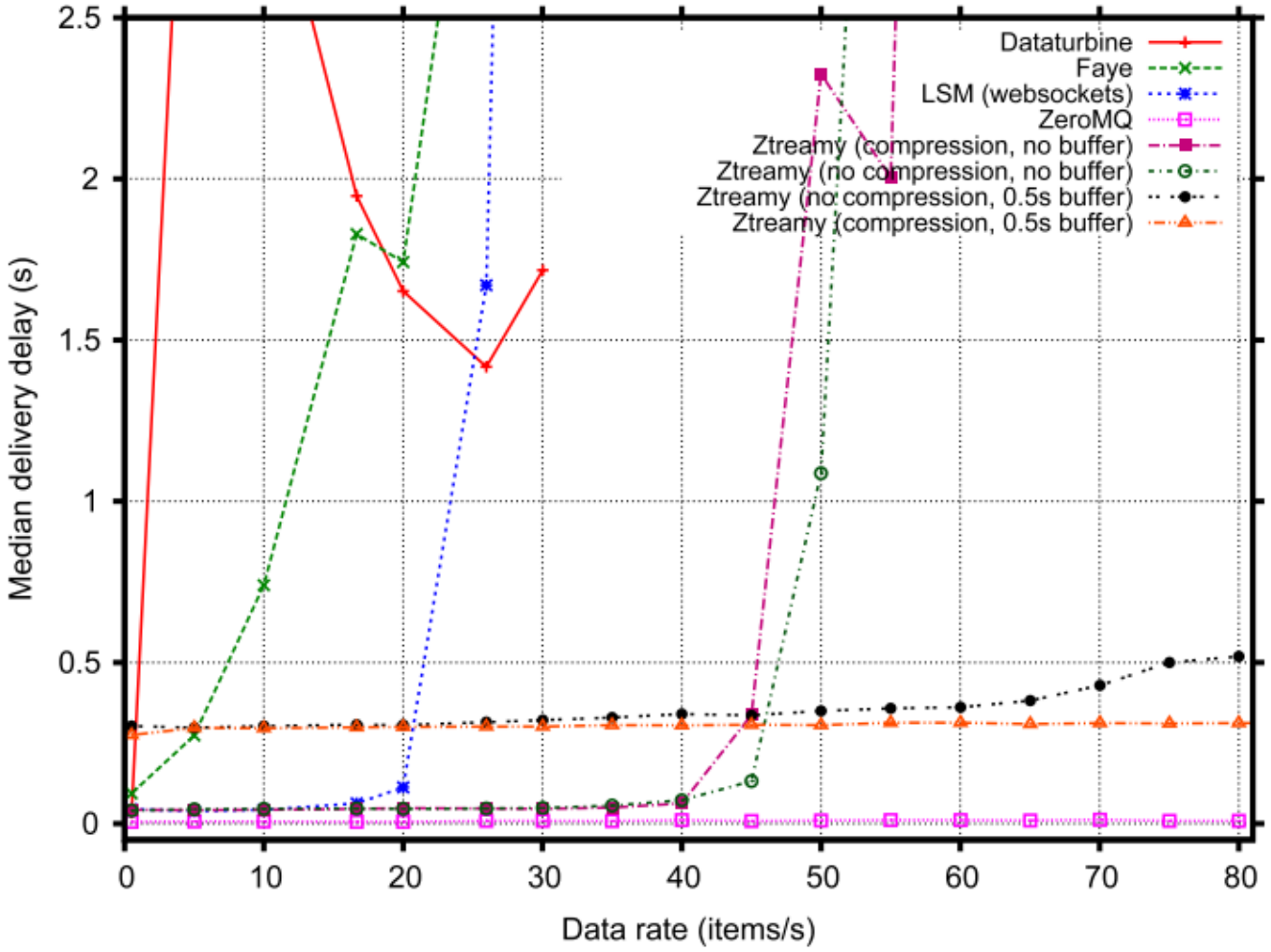
- DataTurbine reached a 2.5% data loss at about 30 items/s.
- Faye delivered all the data even at its saturation point, but suffered from massive delays.
- LSM suffered high delays beyond 25 items/s.
- ZeroMQ begun to drop data, with a 6% data loss at 80 items/s.
- **Ztreamy maintained its performance until aprox. 250 items/s.**

Scenario #2 : CPU Time



(b) CPU time needed to send items at different rates to 500 simultaneous clients.

Scenario #2 : Delivery delays



(d) Delivery delays when sending items at different rates to 500 simultaneous clients.

Performance Discussion

- Ztreamy is able to handle many more clients and much higher data rates.
- Improvements Reason :
  - Buffering and compression are not implemented in any of the other analyzed systems.

Conclusions

- Ztreamy is a scalable platform for publishing semantic streams on the Web.
- Streams can be easily duplicated, aggregated and filtered.
- Applications written in diverse application environments can consume and publish streams by accessing platform server through HTTP.
- The experiments show that Ztreamy outperforms other solutions in a single server installation.