Gresham

INNOVATION LABS PAPER

Taming the Fire-Hose Faster than Google!

October 2018

Rob Yeates
R&D Lead Developer

Big data is often conceptualised as a firehose with massively high volumes and very high flow rate of events. These events are generally inspected and analysed before being aggregated, filtered or mapped in some way. Every time the events are touched CPU cycles are used and memory is consumed. When they are transferred network bandwidth is used. To store these events in some form requires disk capacity to be consumed. Clearly then we want these events to be as small as possible and be as cheap as possible to interact with.

At Gresham, we understand the challenges that big data presents to your infrastructure— as data volumes increase systems slow down and expensive infrastructure has to be upgraded. The Clareti platform includes technology that can help.

Preon is a key component of the Clareti Adapter messaging suite that accelerates data transfer rates and reduces disk and bandwidth requirements. Developed in house by Gresham, Preon is proprietary software offering the highest levels of efficiency when working with data. To achieve this it employs sophisticated serialisation technologies.

Data compression—Doing more with less

With Preon data compression these costs can be avoided. Compressed data uses less resources in terms of memory used, disk space consumed and network bandwidth utilised. This means you can process more data and spend less.

Google have two main offerings in this space ProtocolBuffers and FlatBuffers. There are a number of Open Source alternatives that are quite popular including Avro and MsgPack.

Functional differences

Schema vs schemafree

Preon makes strong guarantees about the data is contains. It stores the description of the data alongside the data. This allows for fast interactions since the data can be examined without external description. As the benchmark shows the next fastest offering FlatBuffers uses a similar approach.

Language interoperability

Preon is available in both Java and Scala providing excellent compatibility with JVM languages. Other tools focus on providing interoperability with more languages but as can be seen in the results potentially to the cost of performance.

Ready to use complex message adaptors

FiX, FpML, SWIFT and ISO2022 are just some of the ready to use converters that can be used as Preon objects.

Functional differences

Preon is the only tool here that is provided with automated UI tooling. For the other choices we had to manually create schema, referring to data type tables to find how to express data types.

The Clareti Adaptors UI can consume source data and automatically creates a model to work from. This increases development momentum and reduces iterations.

Type safety

For integrating systems providing a strong type system allows for stronger guarantees to be made about the data consumed and produced.

Tool	Expression
Preon	Date settlementDate
Flatbuffers	settlementDate:ulong
Protocol Buffers	int64 settlementDate
Avro	{“name”: “settlement_date”, “type”: “long”, “logicalType”: “date”}
Colfer	settlementDate uint64

How do we compare?

How fast can Preon messages be processed compared to the other tools? To measure this we created full-sized objects and compressed them using different technologies. We then measured how fast we could get access to the underlying data.

How do we really compare?

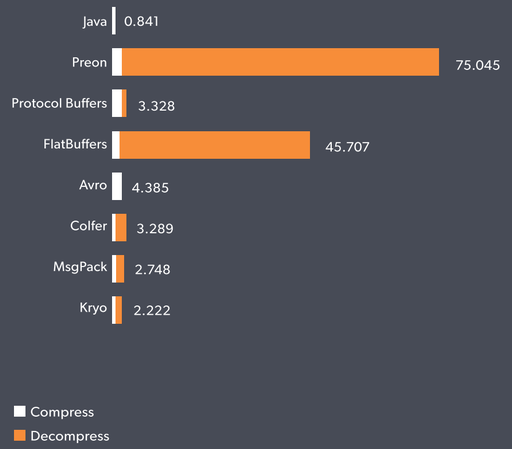
Measuring real world use cases for serialisation technology can be challenging. There are multiple aspects to consider, for example it may be possible to compress an object 100 times and take 100 CPU ticks to do that. However this is less efficient relatively than being able to compress 50 times and take 20 CPU ticks to do that.

As such these snapshot benchmarks don’t always show the full picture. We wanted to see how Preon performed in a realworld benchmark. From consumption through transmission and persistence we wanted to understand the combination of performance and efficiency offered by each tool.

We leveraged big data tools including Spark and Flink and messaging tools including Kafka and gRPC. We included an aggregation task backed by Redis and a mutation task backed by Ignite.

What is clear is that we had to measure every aspect of the flow through our benchmark system to understand how Preon compares to the alternatives. Using tools such as Prometheus and Grafana allow us to measure every metric that makes up this benchmark. By considering CPU cycles, Memory usage, Network bandwidth and Disk usage we can determine the fastest and most efficient serialisation tool set.

Throughput — Operations per second



The results

Compatibility with big data tooling

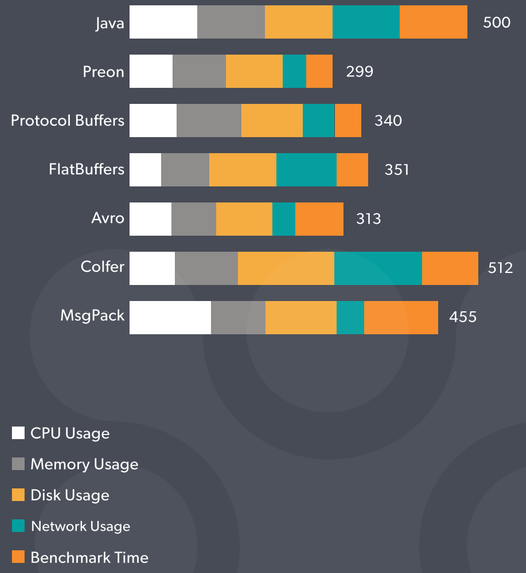
The use-cases for such powerful technologies are myriad. By using the latest technologies such as gRPC and Flink we demonstrate the compatibility, performance and efficiency of Preon with BigData platforms. The network results captured using Kafka show the dramatic improvement in network bandwidth utilisation. Spark worked well with Preon, as did Hadoop. It’s clear that this technology has excellent interoperability

Conclusion

We’ve demonstrated that Preon is faster and more efficient than other serialisation technologies. By using Preon we are able to ingest, process and transfer more data using less hardware resources.

Preon is continuously invested in to maintain this best in class performance. Enterprise ready and Production proven, Preon continues to be a core technology for Gresham.

Usage as % of Native Java Status



Appendix A—Software versions

Library	Version	Link
Avro	1.8.2	avro.apache.org
Colfer	1.11.2	github.com/pascaldekloe/ colfer
Flatbuffers	1.8.0.1	google.github.io/flatbuffers
MsgPack	0.6.12	msgpack.org/index.html
C24 Preon	4.12.0	github.com/C24-Technologies
ProtoBuf	2.9.6	developers.google.com/ protocol-buffers

Benchmark tools used

Library	Version	Link
Redis	4.0.11	redis.io
Ignite	2.6.0	ignite.apache.org
Flink	1.6.0	flink.apache.org
Spark	2.3.1	spark.apache.org
gRPC	1.13.1	grpc.io
Kafka	2.11-1.1.0	kafka.apache.org
Hadoop	2.7.7	hadoop.apache.org

UK
+4420 7653 0222

Luxembourg
+352 278 537 739

North America
+1 646 943 5955

Singapore
+65 6832 5166

Sydney
+612 8514 7007

greshamtech.com
@greshamtech