Streambased Reference Architecture

Table of Contents

Streambased has a very flexible deployment model, leveraging this flexibility can have a drastic effect on the scalability, time to value and futureproofing of a given deployment. For short term, tactical projects an architecture similar to the one described in “Personal/Team Cluster Reference Architecture” can offer extremely fast delivery times to have your analysts up and running in hours rather than weeks. Conversely, for longer term cases, the “Multi-tenant Cluster Reference Architecture” offers ultimate scalability for deployments suitable for serving entire organisations.

This white paper provides a reference for system architects, data engineers and analysts that are planning to leverage Streambased technology. We provide an overview of the essential components that make up a Streambased cluster and discuss guidelines for compute and storage resources to allocate to a new cluster.

Streambased Architecture

Streambased has two core goals that it must fulfil:

1. Accelerate access to data stored within Apache Kafka for analytical purposes
2. Present the accelerated access in a form that is usable and comfortable for analysts, data scientists and AI/ML engineers.

Streambased achieves these goals by utilising 2 components

Streambased Indexer

Streambased achieves up to 30x accelerated read performance from Kafka by leveraging techniques common in the analytical realm but not in the event streaming world, these include indexes, pre-aggregation and statistics.

The Streambased Indexer consumes historical data from Kafka topics and processes this data into the structures needed for acceleration. These structures are then written back into Kafka to accelerate data read later.

This metadata does not contain the data itself and is typically 50-100X smaller in size than the data it represents. The Streambased Indexer serves this data to applications that should be accelerated. This is typically only the Streambased Server but also could be custom made applications using the Streambased Consumer.

Note: The Streambased Indexer interacts with Kafka via Kafka’s consumer groups feature. This means the number of Indexers running at any given time can be elastically scaled up and down.

Streambased Server

The Streambased server offers Kafka data to the outside world in the well understood JDBC (Java Database Connectivity) and ODBC (Open Database Connectivity) protocols. This enables typical BI and analytics technologies such as DBT, Tableau, Qlik, PowerBI and

Superset to connect and explore Kafka data using SQL. Streambased Server supports a rich ANSI SQL interface to allow all of the usual interactions expected of database systems.

Streambased Server works in conjunction with the Index Server to provide accelerated access meaning queries that would execute in “batch” time periods (i.e. hours to overnight) to execute in “human” periods (i.e. seconds to minutes). This allows Kafka to become a system that can be explored and utilised interactively.

Note: The Streambased Server is built on top of the Trino (https://trino.io/) distributed query engine. This means it can be scaled horizontally to address higher query volume and complexity when required.

Streambased Consumer

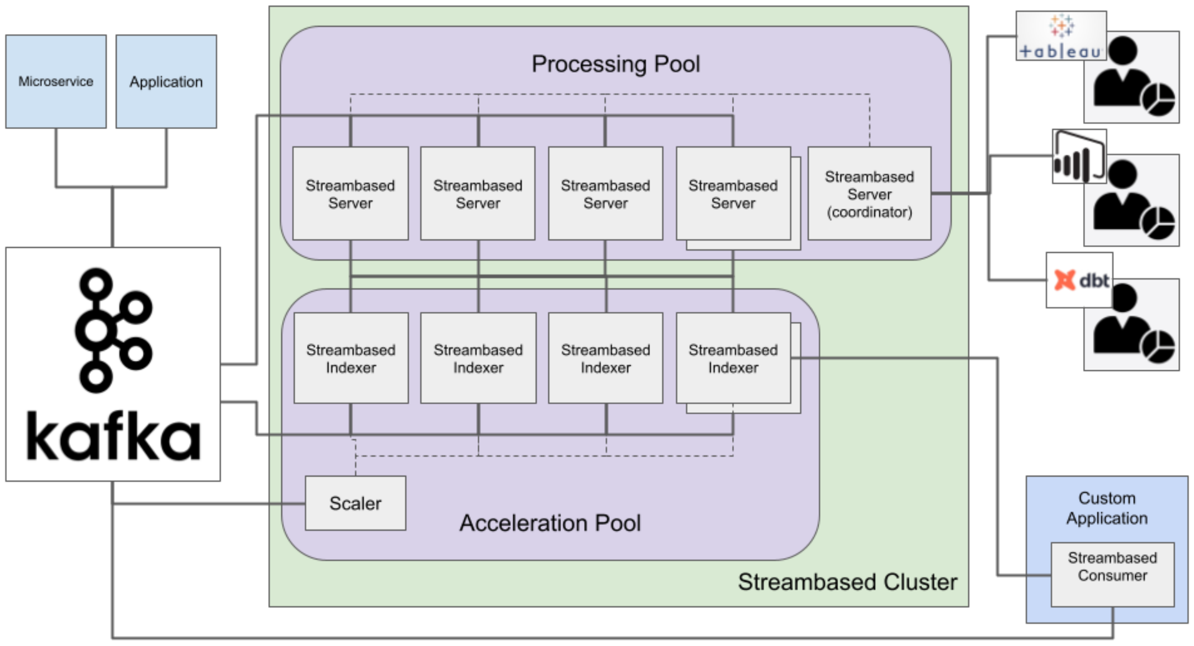
To enable custom projects to leverage acceleration, Streambased has developed an implementation of the Kafka Consumer (https://kafka.apache.org/26/javadoc/org/apache/kafka/clients/consumer/Consumer.html) interface.

This component works in conjunction with an Index Server to accelerate reads based on a number of filtering parameters that are provided programmatically.

Note: The Streambased Consumer does not require a Streambased Server as it communicates via Kafka protocol and not SQL.

Multi-tenant Cluster Reference Architecture

Taking all of the above into consideration a multi-tenant Streambased architecture may look like the below:



In the above, the cluster is split into 2 distinct scalable units, the Processing Pool and the Acceleration Pool. The Acceleration Pool is responsible for reading historical data and preparing the metadata required to accelerate reads from Kafka. The Processing Pool is responsible for collating the data required to satisfy queries submitted by clients and processing it. Both are horizontally scalable by adding new nodes.

Note: Cluster sizing above is arbitrary and components that scale horizontally are denoted with shadowed boxes.

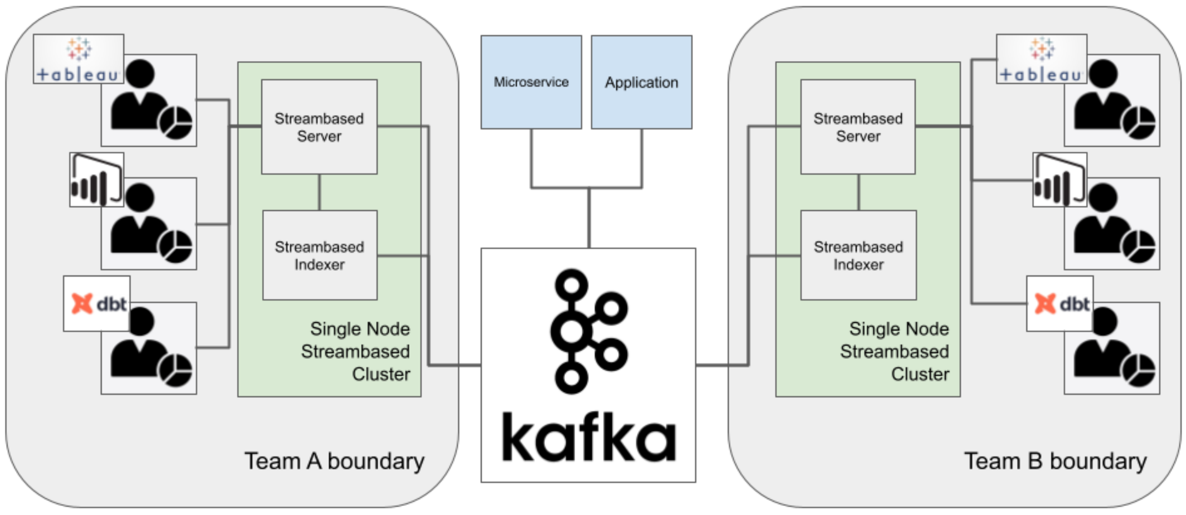
Some components above warrant further explanation as they are not covered by the core Streambased component descriptions above:

Scaler - This component watches the consumer lag of the Streambased Indexers in the Acceleration Pool and, should lag be increasing, will provision new indexers to help. This component is dependent on the provisioning practices of your organisation and so is developed in conjunction with Streambase professional services.

Streambased Server (coordinator) - Streambased Servers can have one of two roles, “Worker” or “Coordinator”. Whilst all nodes can perform processing work associated with SQL queries, coordinators take on the additional responsibility of orchestrating this processing. They will communicate with other Streambased servers in the Processing Pool to ensure that work is distributed evenly and the cluster is making best use of the resources available to it. By convention client access is also facilitated by this node.

<u>Personal/Team Cluster Reference Architecture</u>

The above architecture represents a very different deployment of Streambased targeting a fast time to value, tactical use case such as exploratory data analysis for a data science team or a P.O.C. before deploying the multi-tenant architecture.



This architecture focuses around a number of single node Streambased deployments (typically one per project/team). Each deployment is limited in scope to providing access to only the data required by a specific team or project. In this way an individual cluster typically does not require distributed scale and instead the analytical capability is scaled by adding more single node clusters.

Note: This architecture is intended as a stepping stone to the multi-tenant architecture and we recommend that the number of isolated Streambased clusters is minimised before a larger deployment is considered.

<u>Capacity Planning</u>

Storage

Streambased does not require any persistent storage for its services (everything permanent is persisted back to Kafka). However, we recommend that some storage is provided to the services to facilitate processing of large datasets (Streambased Indexer) and complex queries (Streambased Server). We recommend the following:

Component	**Disk Type**	**Disk Size**
Streambased Indexer	Basic	100GB
Streambased Server	SSD	50GB

Memory

Streambased Indexer progressively consumes and processes records and so memory requirements are low. We recommend a higher memory limit for Streambased Server to facilitate fast exchange and sorting of data as required by common SQL queries.

Component	**Memory**
Streambased Indexer	4GB
Streambased Server	20GB

CPU

Both Streambased Server and Streambased Indexer have moderate CPU requirements to facilitate decoding and processing of Kafka records:

Component	**Cpu Cores**

Streambased Indexer	4
Streambased Server	4

Network

Network provisioning generally follows the Kafka cluster that the Streambased cluster will be accessing. Large-scale Kafka deployments using 1GbE typically become network-bound. If the underlying Kafka cluster is scaled to over 100MB/s, you must provision a higher-bandwidth network.

<u>Conclusion</u>

This white paper shares a starting point for configuring and deploying a Streambased cluster. All cases are different so this should be used only as a “jumping off” before tuning your Streambased cluster for your specific needs.

Considerations such as workload characteristics, access patterns, and SLAs are very important, but they are too specific to be included in this document. For help in choosing the right deployment strategy for your specific use cases, we recommend that you engage with Streambased Professional Services for architecture and operational review.