

Spark and Cassandra working together to provide real-time node performance metrics

Overview

At Instacluster we too have a big data challenge that we are solving with Apache Cassandra® and Apache Spark®. Instametrics provides us with the perfect opportunity to dogfood the Instacluster technology stack.

As a managed service company we have been closely monitoring thousands of nodes that we have had under management over an extended period of time. We have been collecting an extensive number of metrics on each of these nodes, that we then feed to a monitoring stream engine (Riemann), which in turn alerts our support team as soon as any issue is detected.

So while we want to process these alerts in real-time, we also want to keep recent raw metrics for a period of time. Keeping the detailed data for an initial period of time is important to us for two specific reasons. Firstly, we need to keep information for several days so that our support team can perform analysis as required for recent history of a node and cluster, and secondly to provide a rich set of recent data for either display through our dashboard or to expose the metrics through our monitoring API.

Use Case:

Big Data Management

Technology:

Open Source
Apache Cassandra® and
Apache Spark™

Website:

<https://instacluster.com/>

However, in addition to our short-term storage needs, it is also necessary for us to calculate and store summary statistics for the longer term to provide us with a historic view of our operating environment and to enable us to identify long term trends. This longer-term view enables us to apply advanced analytic processing to predict failures before they occur and to manage our underlying infrastructure more effectively. With our Instametrics solution, we are building a big data analytics environment to continually improve our services and capabilities for our customers.

Challenge

The challenge that we face with this capability is very similar to those faced by many of our customers.

Throughput

Each of our nodes under management generates data for somewhere between one thousand and several thousand different metrics that we capture. Each individual metric is collected every 10 seconds on each individual node.

At the current time we are collecting and storing data and then processing and alerting on over 50,000 metric events per second. Naturally, we want to store this stream back in a data store cluster. This represents a very large and sustained number of writes per seconds that few database systems have the ability to ingest with required performance.

Data Volume

With more than 50,000 metrics generated per second we need to be capable of writing towards a Terabyte per 24 hour period to disk. We need a data-store that can efficiently store this volume in a way that supports slice queries on time range, as this represents our most common read query pattern.

We also need a storage engine with built in TTL (Time to Live) for an easier data lifecycle management, ensuring that data gets automatically removed to make way for fresh data.

Data Processing

We need much more than just a simple data store. We also need a storage layer that will allow fast and sustained processing in order to calculate summary statistics on all our data, providing another data set that is more suitable for long term storage.

Another requirement is that we have to be capable of running complex analytic queries

across the whole data set to more efficiently detect unhealthy nodes and clusters.

Availability

Instametrics is mission-critical for us. Not only does it provide information and data to our customers about the health of their nodes and clusters, it also provides the backbone for our alerting and monitoring system, which is a key feature of any managed service.

Solution

You can call it dogfooding, or you can say that we are a little biased, but the truth is that the power of Apache Spark™ on Apache Cassandra® is the perfect fit for our requirements. Probably one of the most typical use cases for Cassandra is to store times series data, which is exactly what we produce through the flow of metrics from our nodes under management.

We opted for the recent DTCS (DateTieredCompactionStrategy), which ensures that data created close in time are stored in the same SSTable. This is particularly useful for Instametrics and our use case as our typical read pattern involves sliced queries on a time range.

To manage the data size we use a TTL of 72 hours. A convenient characteristic of DTCS is that because SSTables contains data of the same age, the SSTable file gets deleted when the data it contains is older than the TTL, which in turn provides a saving on the CPU intensive compaction/tombstone cleanup activities. We have moved our deployment to the recently supported m4.xl-800 node size on AWS.

This solution is cost effective and provides ample disk space for us to store the data. This allows us to use a large 12 node cluster, which provides us with over 9TB of storage with a total of 48 CPUs and 192GB of memory across the cluster, which in turn is exactly what the distributed analytic engine provided by Apache Spark is looking for.

We use Apache Spark to roll-up the data in five minute, two hour, and one day intervals. This is only the beginning of our plans as we envisage much more extended and deep analytics as our capability continues to grow.

Finally, maybe one of the most interesting features of Cassandra for us is that as our business continues to grow, and the number of nodes under management increases, we will naturally add more nodes to our own cluster to keep the processing capabilities the same.



About Instaclustr

Instaclustr helps organizations deliver applications at scale through its managed platform for open source technologies such as [Apache Cassandra®](#), [Apache Kafka®](#), [Apache Spark™](#), [Redis™](#), and [OpenSearch®](#).

Instaclustr combines a complete data infrastructure environment with hands-on technology expertise to ensure ongoing performance and optimization. By removing the infrastructure complexity, we enable companies to focus internal development and operational resources on building cutting edge customer-facing applications at lower cost. Instaclustr customers include some of the largest and most innovative Fortune 500 companies.

Apache Cassandra®, Apache Spark™, Apache Kafka®, Apache Lucene Core®, Apache Zeppelin™ are trademarks of the Apache Software Foundation in the United States and/or other countries. Elasticsearch and Kibana are trademarks for Elasticsearch BV, registered in the U.S. and other countries. Postgres®, PostgreSQL® and the Slonik Logo are trademarks or registered trademarks of the PostgreSQL Community Association of Canada, and used with their permission. OpenSearch® is a registered trademark of Amazon Web Services.