Esper EQC

Horizontal Scale-Out
for Complex Event Processing
Esper EQC - Introduction

- Esper query container (EQC) is the horizontal scale-out architecture for Complex Event Processing with Esper and EsperHA
- EQC allows adding more nodes to and removing nodes from a system, such as to achieve:
  - Distributed processing
  - Fault tolerance
  - Automated and fast fail-over
  - Replication and reprocessing
  - Rebalancing load dynamically
  - No-downtime rolling deployments
  - Support for multiple data centers
  - Dynamic discovery
Esper EQC - Architecture

- Builds on proven components
  - Zookeeper
  - Kafka and Kafka Streams

- EQC is part of Enterprise Edition
  - Esper for core query execution
  - EsperHA for state management

- EQC releases follow the Kafka release schedule
- EQC has general-availability with version 7.0.0 of Enterprise Edition
- EQC has a web application GUI for cluster management
Key Concepts

- **Kafka topics**
  - Have a very high write and read performance since they are append-only
  - Can be compacted offline in the background keeping the last entry
  - Provide excellent scalability by partitioning

- **EsperHA**
  - Organizes data into keyed pages thus avoiding inefficient per-data read/write
  - Pages are units that efficiently serialize and de-serialize
  - Can remove pages from memory and read them back in when needed
  - Can manage memory thus a one-month time window is not a problem
  - Can utilize a local disk for fastest write and read performance
  - Writes to local disk are append-only
Kafka Streams and EsperHA

- For EQC and Kafka Streams, a Kafka partitioned and compacted topic is the source-of-truth
  - Also known as the “changelog”
  - It contains messages that are EsperHA pages
  - Kafka compacts and keeps the last version of EsperHA pages
  - Offline compaction lazily deletes old EsperHA pages
  - Upon restore or rebalance the “changelog” topic provides the current EsperHA pages for a given partition

- EsperHA local disk acts as a cache only
  - Allows EsperHA to remove all CEP engine state from memory if needed
This Architecture is Optimal Because… (1/3)

- All writes are append-only
  - Write to Kafka broker “changelog” topic is over the network, and the Kafka broker appends to its own local file system.
  - Write to local file system are append-only
- EsperHA pages allow bulk operation with configurable granularity
- It is Masterless and servers don't need to see each other (Kafka brokers do)
- Offline compaction means automatic and cheap cleanup
This Architecture is Optimal Because… (2/3)

- Kafka, Kafka Streams and EQC are a better approach compared to...
  - Spark Streaming can only do batches and introduces high latency
  - HDFS distributed file system is not designed for frequent writes and does not have the append-only performance of local file system writes (Flink can only do HDFS and FS, and can only manage state per data key)
  - Distributed caches require nodes that see each other and lots of memory and network bandwidth (such as Ignite or Hazelcast)
- EQC has minimal latency due to event-at-a-time processing without batching
- EPL is SQL-standard compliant and full-featured while other SQL implementations are minimal
- EPL allows management of queries at runtime
This Architecture is Optimal Because... (3/3)

- Simple to set up and manage
  - Requires only Zookeeper and Kafka brokers

- Give you control over how you deploy
  - Compatible with any JVM application deployment option
  - Easy to use with Docker
  - No need for a fat jar or uber jar and long builds, and no need to learn a complex framework API

- Shorter development since what you test is what you deploy
  - Local and distributed is the exact same development and code
  - Since Esper and EsperHA are the same libraries for local development, with Kafka and EQC managing the distributed operation seamlessly
EQC Cluster Web Application Overview

- Manages EQC cluster
- Act across multiple EQC worker processes and partitions.

Functionality:
- View EQC cluster information
- View EQC per-worker metrics
- View EQC per-partition metrics
- View and search EPL deployments across partitions
- Deploy EPL to multiple or all partitions
- Un-deploy EPL from multiple or all partitions
- Analyze differences between EPL deployments of different partitions
- Define worker and partition metric views
### Worker Metrics

**View Selection:** Worker JVM Default View

<table>
<thead>
<tr>
<th>Worker</th>
<th>heap-used</th>
<th>heap-free</th>
<th>heap-total</th>
<th>heap-max</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.103.8091</td>
<td>199353</td>
<td>67554</td>
<td>200880</td>
<td>451584</td>
</tr>
<tr>
<td>192.168.103.8092</td>
<td>47005</td>
<td>134878</td>
<td>192784</td>
<td>451584</td>
</tr>
<tr>
<td>192.168.103.8093</td>
<td>81599</td>
<td>97360</td>
<td>169990</td>
<td>451584</td>
</tr>
<tr>
<td>192.168.103.8094</td>
<td>69572</td>
<td>104553</td>
<td>174592</td>
<td>451584</td>
</tr>
</tbody>
</table>

### Partition Metrics

**View Selection:** Partition Esper Default View

<table>
<thead>
<tr>
<th>Partition</th>
<th>num-evaluated</th>
<th>routed-internal</th>
<th>routed-external</th>
<th>current-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition 0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
<tr>
<td>Partition 8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1999-12-31T19:00:00:000</td>
</tr>
</tbody>
</table>
EQC Cluster Web App Screenshots (2 of 4)
EQC Cluster Web App Screenshots (3 of 4)
EQC Cluster Web App Screenshots (4 of 4)
Thank you

http://www.espertech.com

info@espertech.com