aquenos
software & more

The Cassandra Archiver

A Scalable Solution for Archiving EPICS Channels

Sebastian Marsching

EPICS Collaboration Meeting, May 1st – 3rd, 2013
Rutherford Appleton Laboratory, Didcot, UK

Motivation

- Two traditional archivers for EPICS:
  - Channel Archiver
    - High Performance
    - Limited to a single node
    - Proprietary data-store engine
  - RDB Archiver
    - Standard data-store engine (MySQL, PostgreSQL, Oracle)
    - Limited Performance
    - Multi-node capabilities depending on the used database
- Wanted:
  - High Performance
  - Multi-node capabilities

Idea

- Use an existing key-value store that is
  - Distributed (multi-node operation),
  - Optimized for high write performance and
  - Offers high-availability through data-replication across multiple nodes.
- Apache Cassandra offers all these features.

Cassandra Archiver

- Project started in 2011.
- Based on the CSS archiver framework.
- Uses Apache Cassandra to store archived data and configuration.
- Multiple archive engines write to the same archive
- The archive reader sees one single archive.
- Store compressed / aggregated samples to speed up data retrieval for a long period (e.g. months or years).

Apache Cassandra

- Started at Facebook, released to the public in 2008.
- Apache top-level project since 2010.
- Data-model is similar to Google Bigtable.
- Used by
  - CERN
  - eBay
  - Hewlett Packard
  - IBM
  - Netflix
  - Spotify
  - Twitter

Apache Cassandra (Continued)

- Data is stored locally
  - No expensive SAN
  - No bottlenecks
- Replication is used for high-availability and improved performance.
- All nodes are equal, thus there is no single-point of failure.
- Use cluster of cheap PCs for building a reliable and scalable data-store.
  - Typical Configuration: quad-core processor, 32 GB memory, 2-4 hard disks, software RAID 0

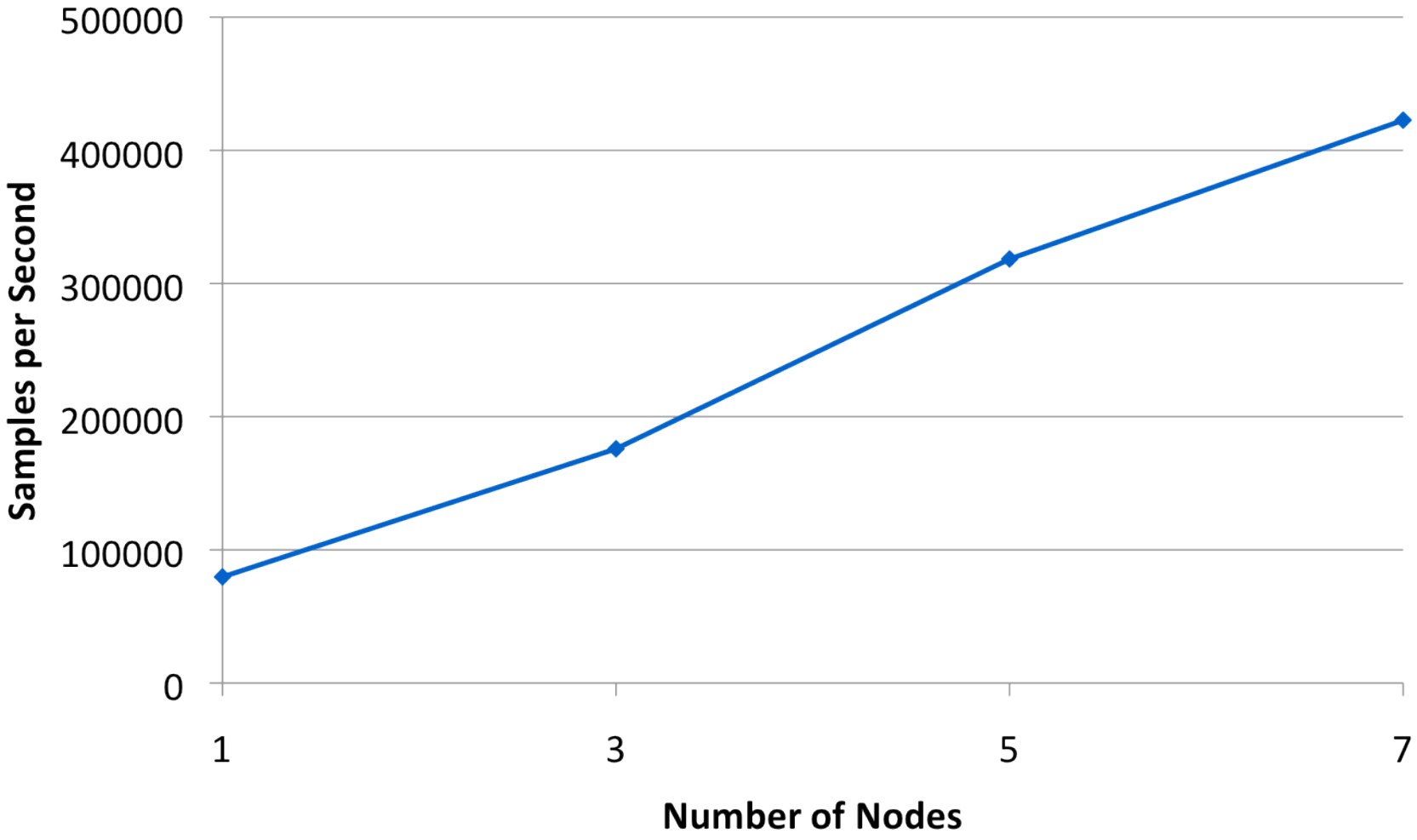
Benefits of the Storage Model

- All write operations are sequential.
  - There are no random writes.
  - Data is aggregated in big files (from about 50 MB up to several hundred GB).
- Reads are fast, because typically data read together is stored together.
- I/O throughput is increased dramatically.

Cassandra Archiver vs. RDB Archiver

- RDB Archiver tested with MySQL database (MyISAM engine)
  - This setup is supposed to have the best performance. PostgreSQL and Oracle supposedly are slower.
- Single-node setup (MySQL cannot easily be used in a distributed environment).
- RDB Archiver: ~5,500 samples/s
- Cassandra Archiver: ~80,000 samples/s

Cassandra Archiver Scalability



Summary

- The Cassandra Archiver has been designed with performance and scalability in mind.
- The Apache Cassandra database provides a high-performance, scalable and highly available data store.
- The Cassandra Archiver outperforms the RDB Archiver in single-node operation and can be scaled further by adding more nodes.

More Information

- http://oss.aquenos.com/epics/cassandra-archiver/
- Contact the speaker at sebastian.marsching@aquenos.com

Compressed Samples

- The Cassandra Archiver can automatically generate compressed samples for defines time intervals.
- Example:
  - Original data changes every second.
  - Compressed samples are calculated for the time intervals of 30 seconds, five minutes, one hour and twelve hours.
- Data for extended periods (e.g. years) can be viewed quickly by retrieving appropriate compressed samples.
- Original samples and compressed samples for short time periods can be deleted after a retention period to save disk-space.

Ideas for Future Versions

- Replace CSS archiver framework.
- Run archiving close to actual data storage for the respective channel.
- Use Apache ZooKeeper for coordinating channel archiving, providing automatic failover if an archive engine fails.

Data Layout on Disk