Gen II Time-Series DB

Background

Insight RMD currently lacks a time-series data store that is appropriately distributed, scalable and fault tolerant. A variety of pilots and POC's have been run, and we are treating Cassandra DB as our preferred alternative for a data store. This document outlines high level architecture and a phased approach to migrating to CassandraDB from Proficy Historian.

Goals

This document outlines a multi-step plan to migrate to a new time-series datastore technology, and make appropriate changes to the presentation layer. Goals from this activity include:

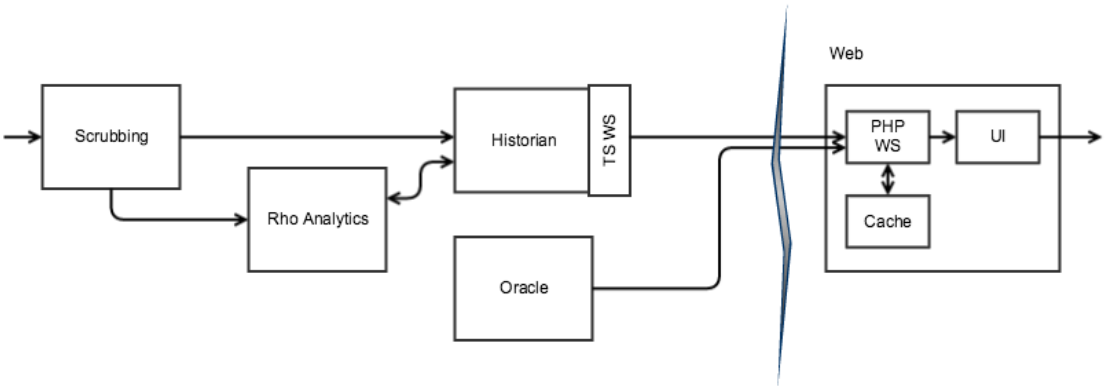
- Movement to "cloud ready" technologies, allowing cloud hosting and lower TCO.
- Distributed, fault-tolerant environment.
- No direct licensing cost for tags or storage.
- Higher performance for more concurrent users.
- Horizontal scalability for huge data sets.
- More effective replication for Stage pre-release testing.
- Tighter integration of the UI with the data store - fewer points of failure.
- Elimination of known reliability issues - Historian API and memcached.
- Highly distributed foundation for integrating analytics and scrubbing tasks.

Architectural Evolution

As-Is State

Key Characteristics

- Time series data exists only in Historian.
- Asset model metadata exists in Oracle.
- Time series data is accessible via standard WS API.
- memcached is used to accelerate metadata access.

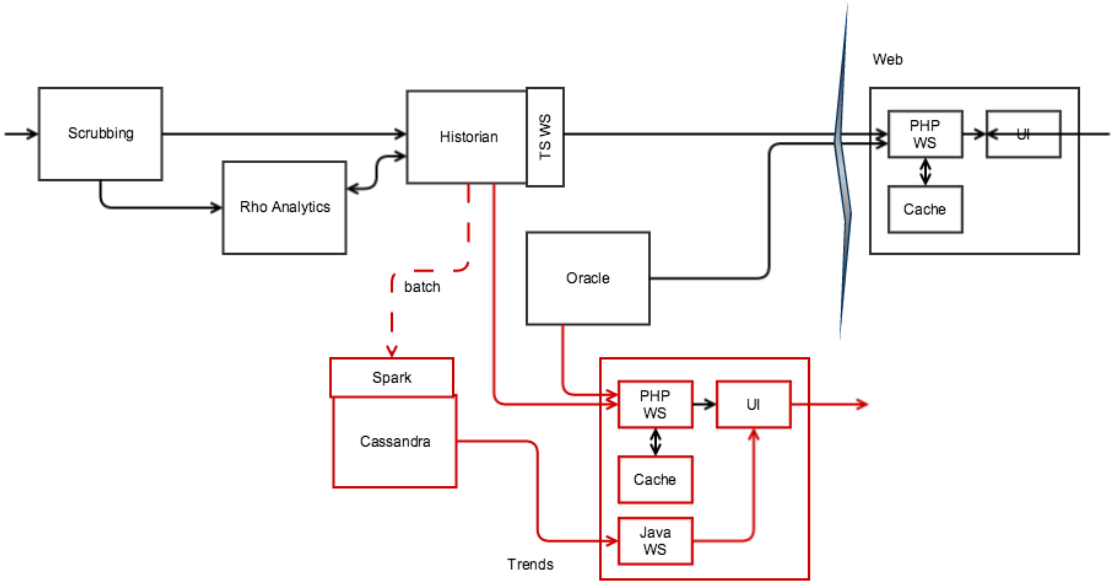


Evaluation Phase

This phase seeks to validate the Cassandra approach without impacting the Prod Insight environment, as quickly as possible. Specific emphasis is given to the Trends screen 7 and 14 days trends, and the Details tab.

Key Characteristics

- Internal web node for demonstration/testing.
- 5 node Cassandra DB cluster for time series data.
- Apache Spark used for hourly and daily roll-ups.
- Time series data queried by UI using Cassandra drivers (binary) not TS WS layer as with Historian.
- Data moved from Historian to Cassandra in batch process (manually).
- Java WS added to Presentation layer to utilize connection pooling.
- Some asset model data sync'd to Cassandra to facilitate Spark jobs.
- Time series data stored in Cassandra in UTC and storage units. (As in Historian)



Production Release 1

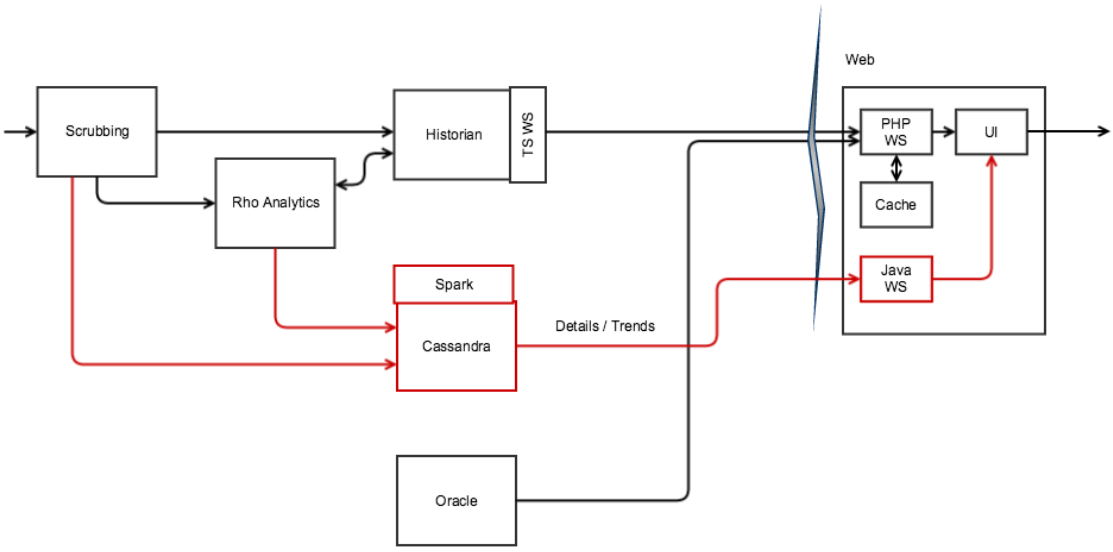
This phase represents the minimum integration required for a viable product release.

Key Characteristics

- Time-series data duplicated in real-time between Historian and Cassandra by Scrubbing and Rho processes.
- Spark implementation of Health roll-up.
- Details/Trends handled by new Java WS (Tomcat).
- Dashboard services collapsed into simpler Cassandra based lookups.
- Both raw and aggregate data in Cassandra stored in storage, display and standard units.
- Both raw and aggregate data in Cassandra stored in UTC and site timezones.

Risks / Questions

- Should the Insight UI use the common TS API web services, or be directly table-aware and use Cassandra binary drivers?
- Impact to scrubbing performance duplicating writes?
- Cassandra support.
- Spark support.

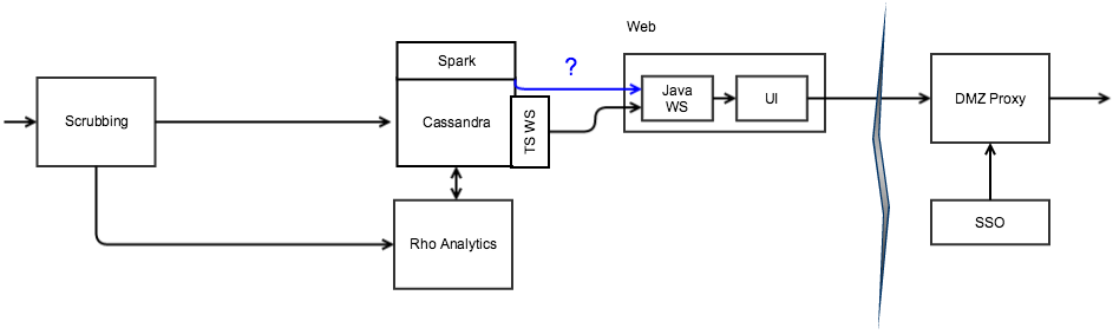


Historian Removal + Web Restructuring

This phase represents complete detachment from Historian and Oracle as dependencies. It also restructures the Presentation layer to provide flexibility and scaleability.

Key Characteristics

- All time-series UI queries redirected to Cassandra.
- All asset-model data migrated to Cassandra.
- Calc Collector removed as data source.
- PHP WS and Memcached decommissioned (ported to Java) - Optional
- Time-Series WS added to Cassandra to provide common API for all clients.
- Internal web nodes deployed to reduce costs and increase responsiveness.
- DMZ web nodes de-scoped to authentication and request forwarding only.

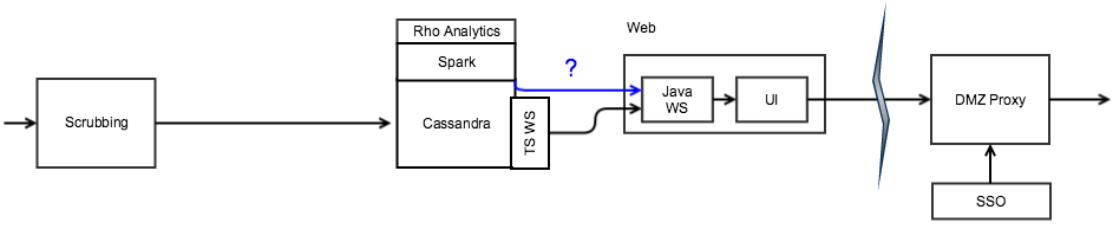


Distributed Analytics

This phase harnesses the distributed compute power of Apache Spark to improve the performance and scaleability of Rho analytics.

Key Characteristics

- Rho analytics (python) deployed across Spark cluster.


Scrubbing
Rho Analytics
Spark
Cassandra
TS WS
?
Web
Java
WS
UI
DMZ Proxy
SSO