Data Processing & Data Integration Platform

STREAM HORIZON

High Throughput, Real Time

by Threeglav © ®

THREEGLAV

threeglav.com

About StreamHorizon…

- Data Processing & Data Integration Platform
- Data Processing throughput of **1 Million Records per second** (single commodity server)
- Quick Time to Market – deploy StreamHorizon & **deliver** your Data integration project **in a single week**
- Fully Configurable via XML – making a project ideal outsourcing candidate
- Utilizes skills of your existing IT staff - requires no ETL platform specialized knowledge

- **Total Project Cost / Data Throughput** ratio = 0.2 (**20% of budget required** in comparison with Market Leaders)
- **1 Hour Proof of Concept** – download and test-run StreamHorizon's demo Data Warehousing project

- Horizontally and Vertically scalable, Highly Available, Clusterable
- Deployable as Data Processing Grid (ETL Grid)
- Deployable on existing Compute Grid (alongside grid libraries - Quant Library)

Applicability Domains

Enabling your business to leverage data for

- Data Warehousing (Real Time rather than batch oriented)
- Data Integration (Real Time rather than batch oriented)
- Business Intelligence, Management Information & Client Reporting Projects
- Time Series Analysis
- Scenario Analysis
- Data Mining
- R&D

… while reducing project Time to Market, costs of IT development, hardware & cost of ownership

Industries (not limited to…)

Finance - Market Risk, Credit Risk, Foreign Exchange, Tick Data, Operations

Telecom - Processing PM and FM data in real time (both radio and core)

Insurance - Policy Pricing, Claim Profiling & Analysis

Health – Activity Tracking, Care Cost Analysis, Clinical Outcomes per money spent KPI’s

ISP - User Activity Analysis & Profiling, Log Data Mining, Behavioral Analysis

The Problem

- **Complexity** - Data Integration can be typically characterized as a set of large (in numbers), interdependent and usually trivial units of ETL transformations/code. Above stated spells 'high complexity' for majority of Data Integration, Business Intelligence and Data Processing projects.
- **Performance** - Data Processing commonly faces performance issues, long load times, SLA breaks and long processing windows. Consequence of batch oriented rather than Real Time & Data Streaming design paradigms & thinking.
- **Query Latency -** Business Intelligence users often require much smaller query latencies from the ones delivered.

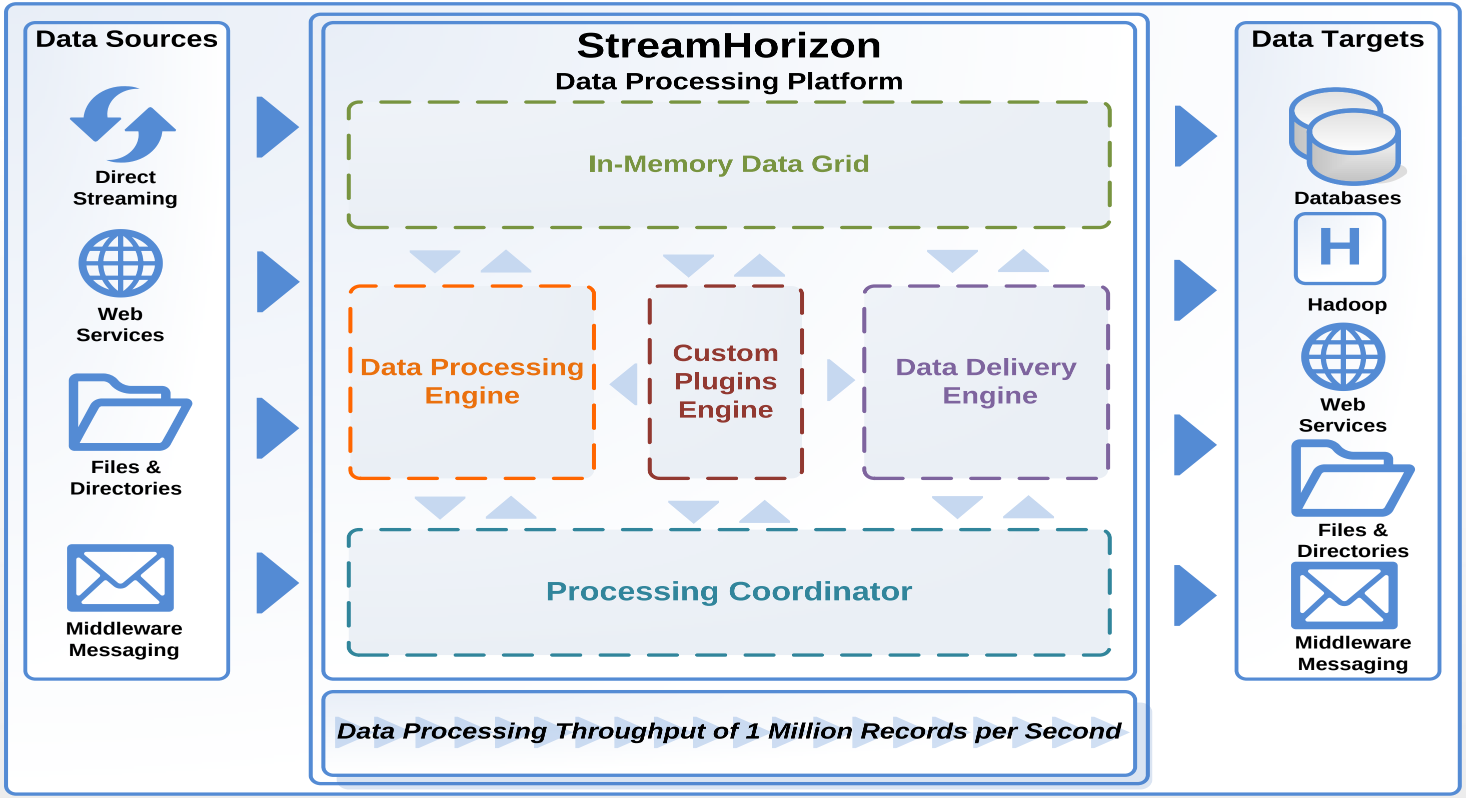
Our Solution

- We have created generic platform which is simple to setup and capable to deliver projects of high complexity. Time to market of project delivery is being measured in day(s) rather than months. Skills required for delivery are reduced almost to a truly basic 101 IT knowledge.

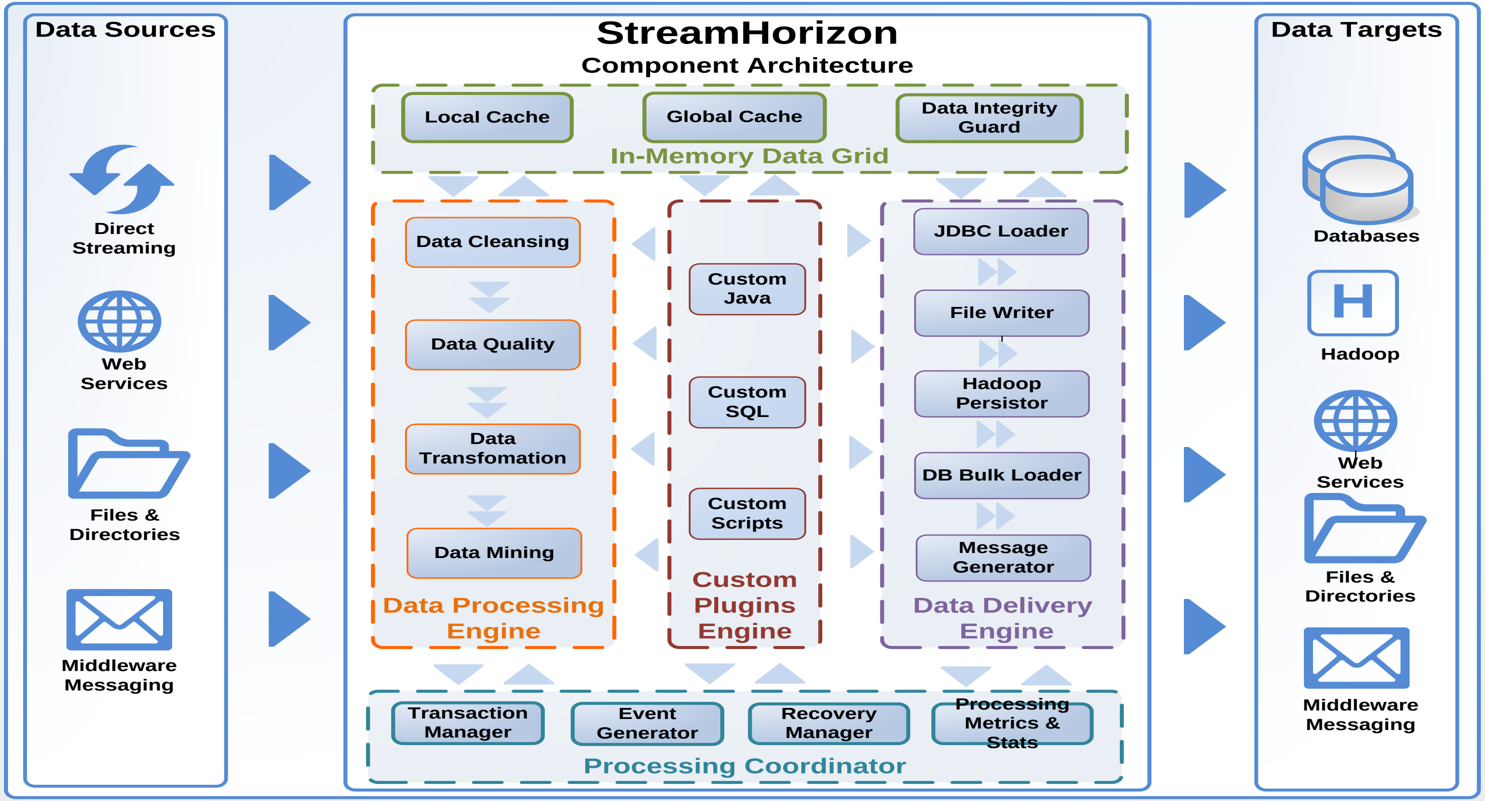
- We have delivered highly performing ETL platform, which eliminates performance issues even for volumes typical only for Financial Exchange. Desktops running on our platform have more processing bandwidth than commodity servers with state of the art (read complex and expensive) ETL tools and couple of dozens of CPU's.

- Performance of our platform (throughput measured in millions of records per second) allows database experts to index their Data Mart to extent previously unimaginable (4+ indexes per single fact table) while still having real time database loading in place (250K-800K records/second). This ability further simplifies Business Intelligence stack by eliminating the need to have MOLAP or other alternatives acting as query accelerators behind relational databases (aggregate tables, materialized views or equivalents). It also reduces complexity of the Business Intelligence stack, budget spent on hardware & licences. Most important is complexity of solution overall (reduction of development man-day and support costs).

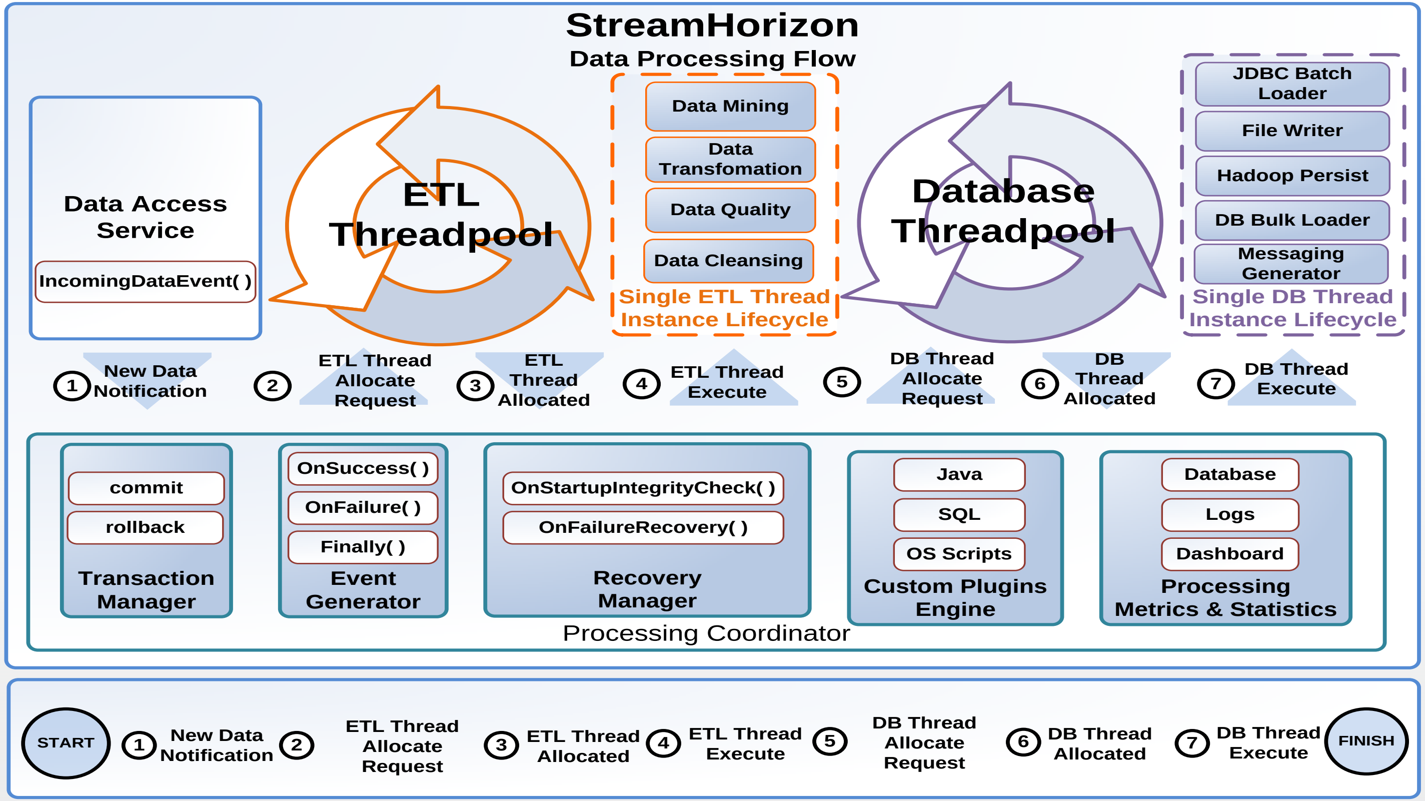
High Level Architecture



Component Architecture



Data Processing Flow



StreamHorizon

Data Processing Flow

Data Access Service

IncomingDataEvent( )

ETL Threadpool

Data Mining

Data Transformation

Data Quality

Data Cleansing

Single ETL Thread Instance Lifecycle

Database Threadpool

JDBC Batch Loader

File Writer

Hadoop Persist

DB Bulk Loader

Messaging Generator

Single DB Thread Instance Lifecycle

1 New Data Notification

2 ETL Thread Allocate Request

3 ETL Thread Allocated

4 ETL Thread Execute

5 DB Thread Allocate Request

6 DB Thread Allocated

7 DB Thread Execute

Transaction Manager: commit, rollback

Event Generator: OnSuccess( ), OnFailure( ), Finally( )

Recovery Manager: OnStartupIntegrityCheck( ), OnFailureRecovery( )

Custom Plugins Engine: Java, SQL, OS Scripts

Processing Metrics & Statistics: Database, Logs, Dashboard

Processing Coordinator

START 1 New Data Notification 2 ETL Thread Allocate Request 3 ETL Thread Allocated 4 ETL Thread Execute 5 DB Thread Allocate Request 6 DB Thread Allocated 7 DB Thread Execute FINISH

Functional & Hardware Profile

	StreamHorizon	Market Leaders
Horizontal scalability (at no extra cost)	🔵	🔴
Vertical scalability	🔵	🔵
Clusterability, Recoverability & High Availability (at no extra cost)	🔵	🔴
Runs on Commodity hardware*	🔵	◐
Exotic database cluster licences	Not Required	Often Required
Specialized Data Warehouse Appliances	Not Required	Often Required
Linux, Solaris, Windows and Compute Cloud (EC2)	🔵	🔵
Ability to run on personal hardware (laptops & workstations)*	🔵	🔴

* - Implies efficiency of StreamHorizon Platform in hardware resource consumption compared to Market Leaders

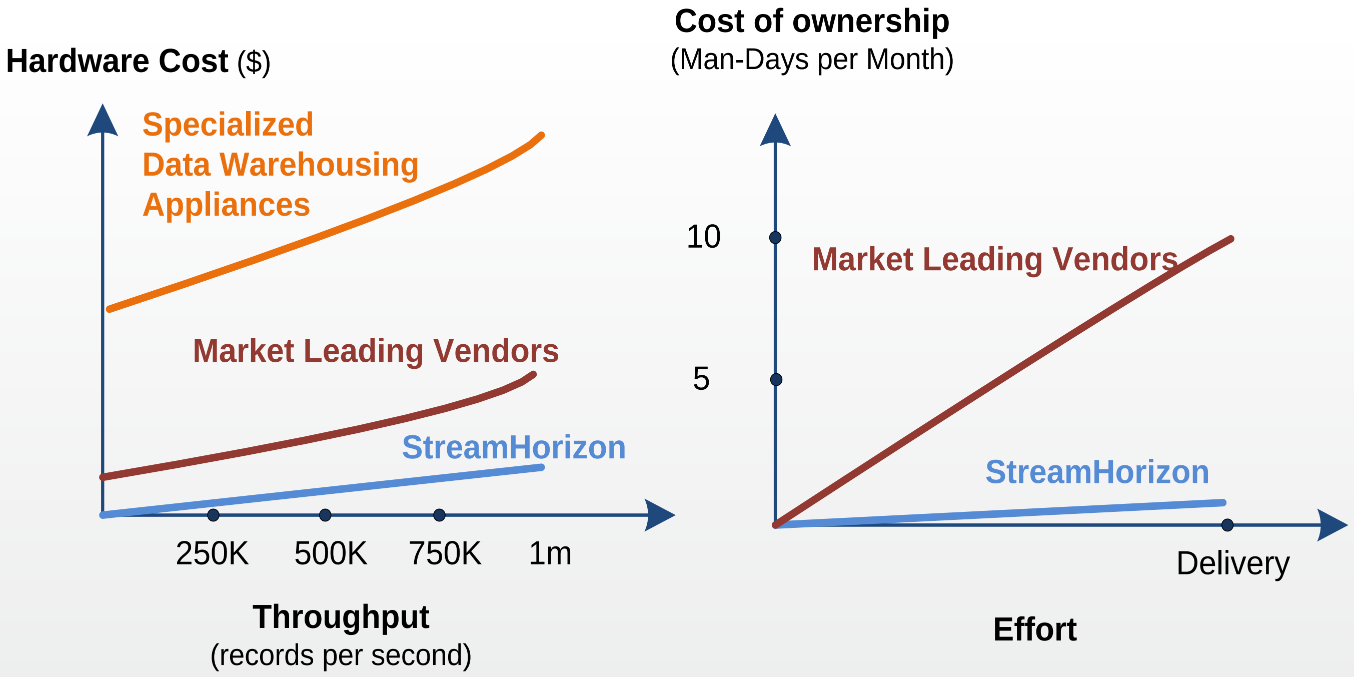
Cost-Effectiveness Analysis - I

Target Throughput of 1million records per second	StreamHorizon	Market Leaders
Hardware Cost	1 unit	3-10 units
Time To Market (installation, setup & full production ready deployment)*	4 days	40+ days
Throughput (records per hour) **	3.69 billion (Single Engine)	1.83 billion (3 Engines)
Throughput (records per second) **	1 million (Single Engine)	510K (3 Engines)
Requires Human Resources with specialist ETL Vendors skill	No	Yes
Setup and administration solely based on intuitive XML configuration	Yes	No
FTE headcount required to support solution	0.25	2

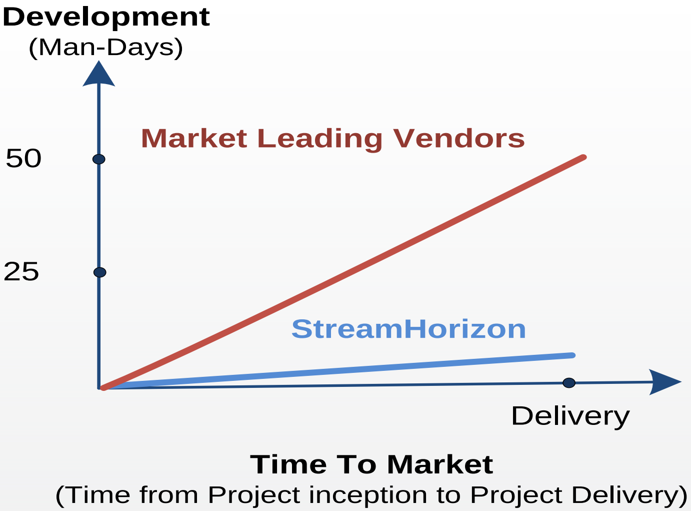
* - Assuming that file data feeds to be processed by StreamHorizon (project dependency) are available at the time of installation

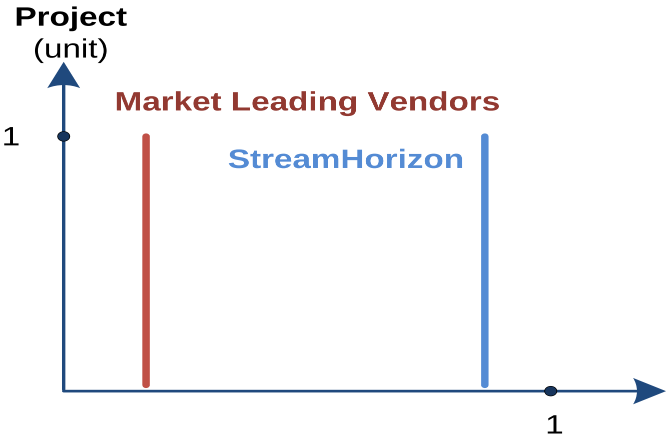
** - Please refer to end of this document for detailed description of hardware environment

Cost-Effectiveness Analysis - II



Cost-Effectiveness Analysis - III





Overall cost-effectiveness ratio
comparison based on:

- Total Costs (licence & hardware)
- Speed of Delivery
- Solution Complexity
- Maintainability

Use Case Study

Delivering Market Risk system for Tier 1 Investment Bank

- Increasing throughput of the system for the factor of 10
- Reducing code base from 10,000+ lines of code to 420 lines of code
- Outsourcing model is now realistic target (delivered solution has almost no code base and is fully configurable via XML)

Workforce Savings:

- Reduced number of FTE & part-time staff engaged on the project (due to simplification)

Hardware Savings:

- $200K of recycled (hardware no longer required) servers (4 in total)

Software Licence Savings:

- $400K of recycled software licences (no longer required due to stack simplification)

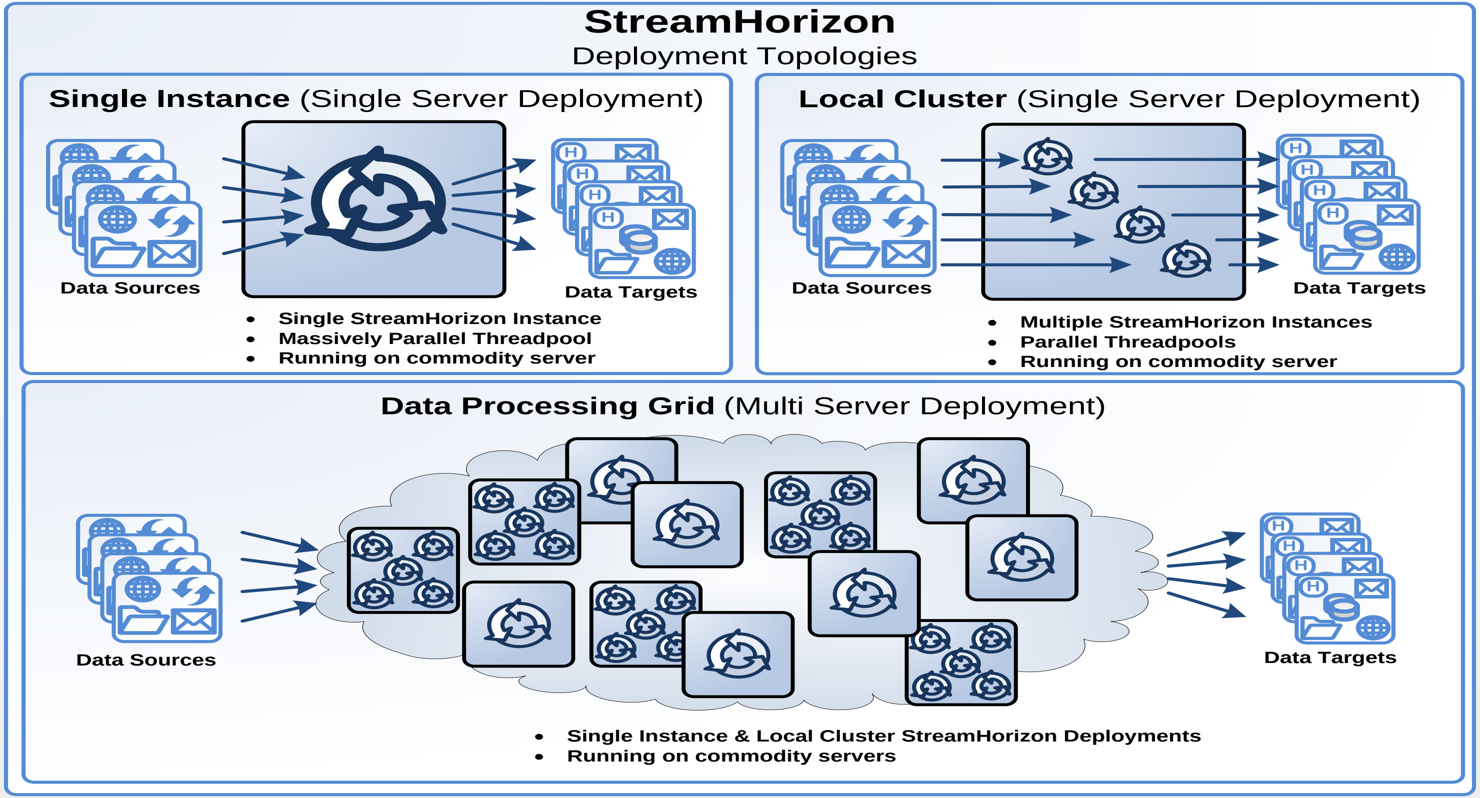
Total

- Negative project cost (due to savings achieved in recycled hardware and software licences)
- BAU / RTB budged reduced for 70% due to reduced implementation complexity & IT stack simplification

Use Case Study - continued

- Single server acts as both StreamHorizon (ETL Server) and as a database server
- Single Vanilla database instance delivers query performance better than previously utilized OLAP Cube (MOLAP mode of operation).
- By eliminating OLAP engine from software stack:
  - User query latency was reduced
  - ETL load latency reduced for factor of 10+
  - Ability to support number of concurrent users is increased
- Tier 1 Bank was able to run complete Risk batch data processing on a single desktop (without breaking SLA).

Deployment



StreamHorizon

Deployment Topologies

Single Instance (Single Server Deployment)

- Single StreamHorizon Instance
- Massively Parallel Threadpool
- Running on commodity server

Local Cluster (Single Server Deployment)

- Multiple StreamHorizon Instances
- Parallel Threadpools
- Running on commodity server

Data Processing Grid (Multi Server Deployment)

- Single Instance & Local Cluster StreamHorizon Deployments
- Running on commodity servers

Data Processing Grid (ETL farm)

Data Processing Grid Deployment

- Enables your organisation to build Data Processing grid, that is, cluster of ETL (Data Processing) servers (analogue to Compute Grid for computation tasks).
- Enables IT infrastructure to reduce total number of ETL servers by approximate factor of 4
- Ability to seamlessly upgrade your Data Processing grid via your compute grid scheduler (Data Synapse or other) or any other management software (Altirs etc.)

Compute Grid Deployment

- StreamHorizon enables you to process data (execute ETL logic) at your compute grid servers as soon as data is created (by Quant Library for example).
- Helps to avoid sending complex/inefficient data structures (like middleware messages, XML or FpML) via network
- Utilize StreamHorizon to process your data at compute grid nodes and directly persist your data to your persistence store
- Eliminates need for expensive In-Memory Data Grid Solutions

Data Processing Grid (ETL farm) - continued

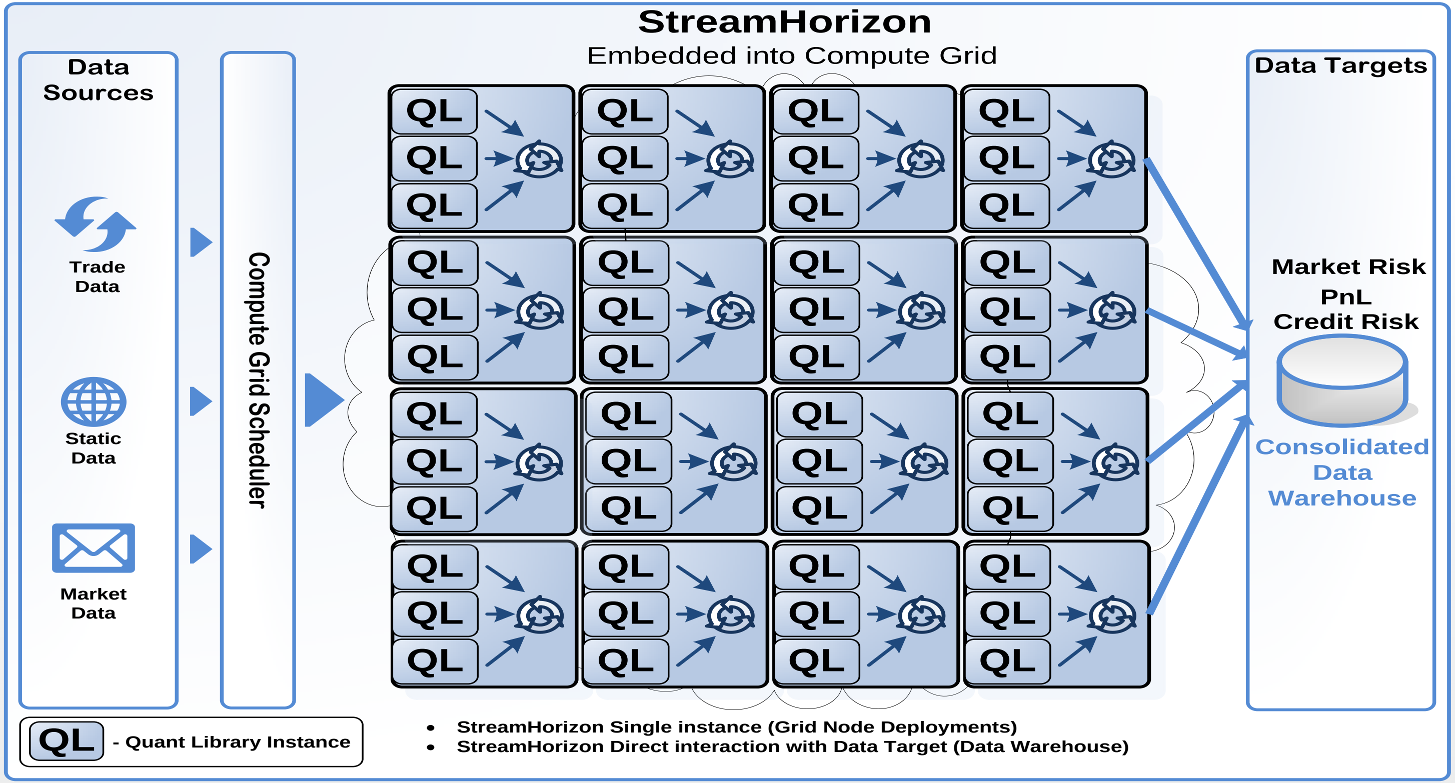
Data Volumes Shipped via Network vs. XML (performance vs. readability)

- Due to challenging volumes numerous architectural solutions gravitate away from XML (self descriptive, human readable data format, but not storage effective)
- XML message size is unnecessarily large, dominated by metadata rather than data itself (over 80%)
- XML does however makes daily job easier for your IT staff and Business Analysts as it is intuitive and human readable as data format

The Solution – Best of both worlds

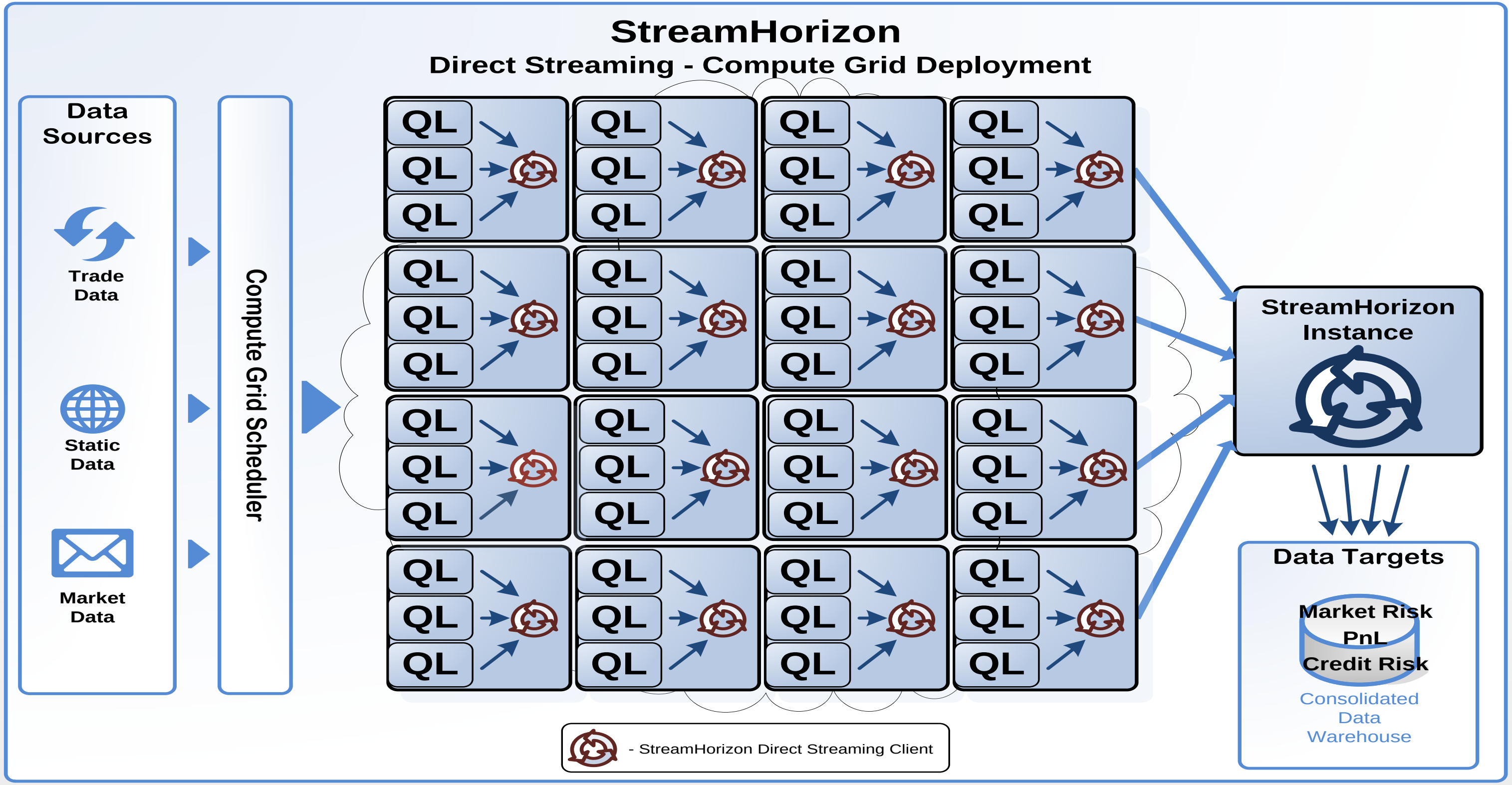
- StreamHorizon deployed at your compute grid nodes enables you to keep XML as inter-process/inter-component message format – without burdening the network traffic.
- StreamHorizon persist directly to your persistence store (most commonly a database) only data, thereby achieving minimal network congestion and latency (as if you where not using XML as cross component communication format).

Compute Grid - Distributed Deployment (Example: Finance Industry)



- StreamHorizon Single instance (Grid Node Deployments)
- StreamHorizon Direct interaction with Data Target (Data Warehouse)

Compute Grid – Data Streaming (Example: Finance Industry)



Reduced Workforce demand

Development

- Reduced development headcount (due to simplicity of StreamHorizon platform)
- Enables utilization of skills of your current IT team
- No need to hire extra headcount with specialized (ETL platform specific) knowledge
- Customizations and extensions can be coded in Java, Scripts and SQL
- Supports agile development methods

Infrastructure & Support

- Not required - Dedicated teams of experts to setup and maintain ETL servers
- Not required - Dedicated teams of experts to setup and maintain ETL platform Repository

BAU / RTB / Cost of ownership

- BAU budgets are usually 20% compared to cost of ownership of traditionally implemented Data Integration project
- Simple to outsource due to simplicity of delivered solution

Environment Risk Management

- Ability to run multiple versions of StreamHorizon simultaneously on same hardware with no interference or dependencies
- Simply turn off old and turn on new StreamHorizon version on the same hardware
- Seamless upgrade & rollback performing simple 'File Drop'
- Backups taken by simple directory/file copy
- Instant startup time measured in seconds
- ESCROW Compliant

Planned platform extensions

- StreamHorizon In-Memory Data Store designed to overcome limitations of existing market leading OLAP and In-memory data stores.
- Feed streaming to support massive compute grids and eliminate data feed (file) persistence and thereby I/O within Data Processing Lifecycle
- StreamHorizon streaming is based on highly efficient communication protocol (analog to ProtoBuffer by Google)

Commodity Server Deployment

(details of StreamHorizon benchmark environments)

Target throughput of 1 million records per second

- Commodity Tier 2 SAN storage (no SSD, no local storage was used)
- Commodity AMD Processors 6 processors (4 cores each)
- Files of size of 100K records (of a minimum single line size of 400 bytes)
- 35 attributes per line which comprise 13 Dimensions (Kimball Star Schema Data Model, fact table has 18 attributes in total)
- Tested with heavily indexed Database tables tuned to respond to queries scanning up to 10 million fact table records within 1.4 seconds (without use of aggregate tables, material views or equivalent facilities. Standard (no RAC, non Parallel Query) Enterprise Edition Oracle database instance)
- Testing Query profile:
  - Aggregate up to 10 million transactional records & return results
  - Return up to 10,000 of transactional records without aggregation
  - All queries execute against single table partition which contains 500 Million records

Desktop Hardware Deployment

(details of StreamHorizon benchmark environments)

Target throughput of 250K records per second

- Target hardware: laptops or workstations (commodity personal hardware)
- Local (not SSD) Hard Disk storage used (single Hard Disk)
- 2 AMD (A8) processors (4 cores)
- Files of size of 100K records (of a minimum single line size of 400 bytes)
- 35 attributes per line which comprise 13 Dimensions (Kimball Star Schema Data Model)

Other Technical details

- Pluggable Architecture - extend and customize StreamHorizon platform without recompilation
- Integrates with RDBMS vendors via JDBC (Oracle, MSSQL, MySQL, SybaseIQ, Sybase, Teradata, kdb+ and others)
- Seamless integration with major programming and script languages (Java, SQL, Shell, Python, Scala, Windows Batch etc.)
- Utilizes leading industry standard software development practices (GIT, Maven, TDD, IMDG...)
- Instant startup time measured in seconds

STREAM HORIZON

Q&A