Data Processing & Data Integration Platform

STREAM HORIZON

Big Data Analytics - Accelerated

Legacy ETL platforms & conventional Data Integration approach

- **Unable to meet latency & data throughput demands of Big Data integration challenges**
  - Based on batch (rather than streaming) design paradigms
  - Insufficient level of parallelism of data processing
  - Require specific ETL platform skills

- **High TCO**
  - Inefficient software delivery lifecycle
  - Require coding of ETL logic & staff with niche skills
  - High risk & unnecessarily complex release management
  - Vendor lock-in
  - Often require exotic hardware appliances
  - Complex environment management
  - Repository ownership (maintenance & database licenses)

StreamHorizon's "adaptiveETL" platform - Performance

- **Next generation ETL Big Data processing platform**
- **Delivering performance critical Big Data projects**
  - Massively parallel data streaming engine
  - Backed with In Memory Data Grid (Coherence, Infinispan, Hazelcast, any other.)
  - ETL processes run in memory & interact with cache (In Memory Data Grid)
  - Unnecessary Staging (I/O expensive) ETL steps are eliminated
  - Quick Time to Market (measured in days)

StreamHorizon's "adaptiveETL" platform - Effectiveness

- **Low TCO**
  - Fully Configurable via XML
  - Requires no ETL platform specific knowledge
  - Shift from 'coding' to 'XML configuration' reduces IT skills required to deliver, manage, run & outsource projects
  - Eliminated 90% manual coding
  - Flexible & Customizable – override any default behavior of StreamHorizon platform with custom Java, OS script or SQL implementation
  - No vendor lock-in (all custom written code runs outside StreamHorizon platform-no need to re-develop code when migrating your solution)
  - No ETL tool specific language
  - Out of the box features like Type 0,1,2, Custom dimensions, dynamic In Memory Cache formation transparent to developer

AdaptiveETL enables:

- **Increased data velocity & reduced cost of delivery:**
  - Quick Time to Market – deploy StreamHorizon & deliver fully functional Pilot of your Data integration project in a single week
  - 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

- **Big Data architectures supported:**
  - Hadoop ecosystem - Fully integrated with Apache Hadoop ecosystem (Spark, Storm, Pig, Hive, HDFS etc.)
  - Conventional ETL deployments - Data processing throughput of 1 million records per second (single commodity server, single database table)
  - Big Data architectures with In Memory Data Grid acting as a Data Store (Coherence, Infinispan, Hazelcast etc.)

AdaptiveETL enables:

- **Achievable data processing throughput:**
  - Conventional (RDBMS) ETL deployments - 1 million records per second (single database table, single commodity server, HDD Storage)
  - Hadoop ecosystem (StreamHorizon & Spark) - over 1 million records per second for every node of your cluster
  - File system (conventional & Hadoop HDFS) - 1 million records per second per server (or cluster node) utilizing HDD Storage

- **Supported Architectural paradigms:**
  - Lambda Architecture - Hadoop real time & batch oriented data streaming/processing architecture
  - Data Streaming & Micro batch Architecture
  - Massively parallel conventional ETL Architecture
  - Batch oriented conventional ETL Architecture

AdaptiveETL enables:

- **Scalability & Compatibility:**
  - Virtualizable & Clusterable
  - Horizontally & Vertically scalable
  - Highly Available (HA)
  - Running on Linux, Solaris, Windows, Compute Clouds (EC2 & any other)

- **Deployment Architectures:**
  - Runs on Big Data clusters: Hadoop, HDFS, Kafka, Spark, Storm, Hive, Impala and more…
  - Runs as StreamHorizon Data Processing Cluster (ETL grid)
  - Runs on Compute Grid (alongside grid libraries like Quant Library or any other)

Targeted Program & Project profiles

Greenfield project candidates:

- Data Warehousing
- Data Integration
- Business Intelligence & Management Information
- OLTP Systems (Online Transactional Processing)

Brownfield project candidates:

Quickly enable existing Data Warehouses & Databases which:

- Struggle to keep up with loading of large volumes of data
- Don't satisfy SLA from query latency perspective

Latency Profile:

- Real Time (Low Latency - 0.666 microseconds per record - average)
- Batch Oriented

Delivery Profile:

- <5 days Working Prototypes
- Quick Time to Market Projects
- Compliance & Legislation Critical Deliveries

Skill Profile

- Low-Medium Skilled IT workforce
- Offshore based deliveries

Data Volume Profile

- VLDB (Very Large Databases)
- Small-Medium Databases

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, Treatment Profiling

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

The Problem

Complexity

- Data Integration = set of large (in numbers), interdependent & usually trivial units of ETL transformations

Performance

- Long loading time windows
- Frequent SLA breaks
- 'Domino Effect' execution & dependencies (Waterfall paradigm)

...above are consequence of batch oriented rather than 'Real Time' & 'Data Streaming' design paradigms

Query Latency

- Longer than expected query response times
- Inadequate Ad-Hoc query capability

Our Solution (Example: RDBMS based Big Data Streaming Platform)

Data Throughput of StreamHorizon platform enables:

- Indexing of Data Warehouses to extent previously unimaginable (4+ indexes per single fact table)
- Extensively indexed database delivers load throughput of 50-80% compared to model without indexes, however, it reduces query latency (intentional sacrifice of load latency for query performance)
- StreamHorizon **fully supports OLAP integration** (please refer to FAQ page). OLAP delivers slice & dice and drill-down (data pivoting) capability via Excel or any other user front end tool.
- No need to utilize OLAP cubes or equivalents (In-Memory solutions) acting as 'query accelerators'. Such solution are dependent on available memory and thereby impose limit do data volumes system can handle.
- Horizontal scaling of In-Memory software comes with a price (increased latency) as queried data is collated from multiple servers into single resultset
- No need to purchase exotic or specialist hardware appliances
- No need to purchase In-Memory/OLAP hardware & licence
- Simple software stack:

StreamHorizon

+

Vanilla Database

VS.

ETL Tool

+

Database (usually exotic)

+

OLAP/In-Memory solution (usually clustered)

Our Solution - (continued)

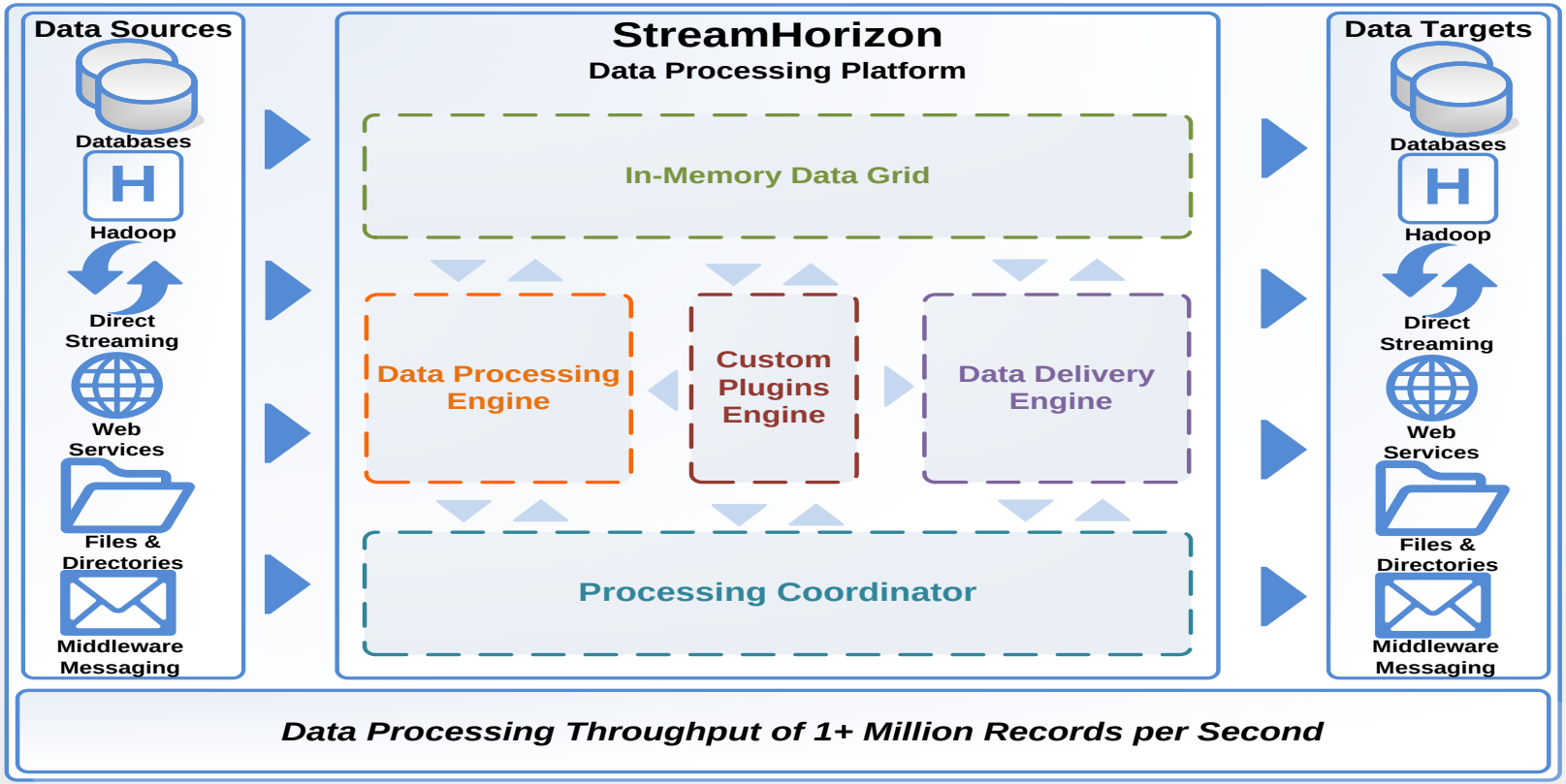
- Time to market of project delivery - measured in days rather than months.
- IT Skills required for development are reduced to basic IT knowledge.
- Manage data volumes typical only for Financial Exchanges, Telecom blue chips and ISP's.
- Desktops running with StreamHorizon platform have more data processing bandwidth than commodity servers with state of the art (read complex and expensive) ETL tools and dozen of CPU's.

- Generic & Adaptable ETL platform
- Simple to setup (XML configuration)
- Platform geared to deliver projects of high complexity (low latency or batch oriented fashion)

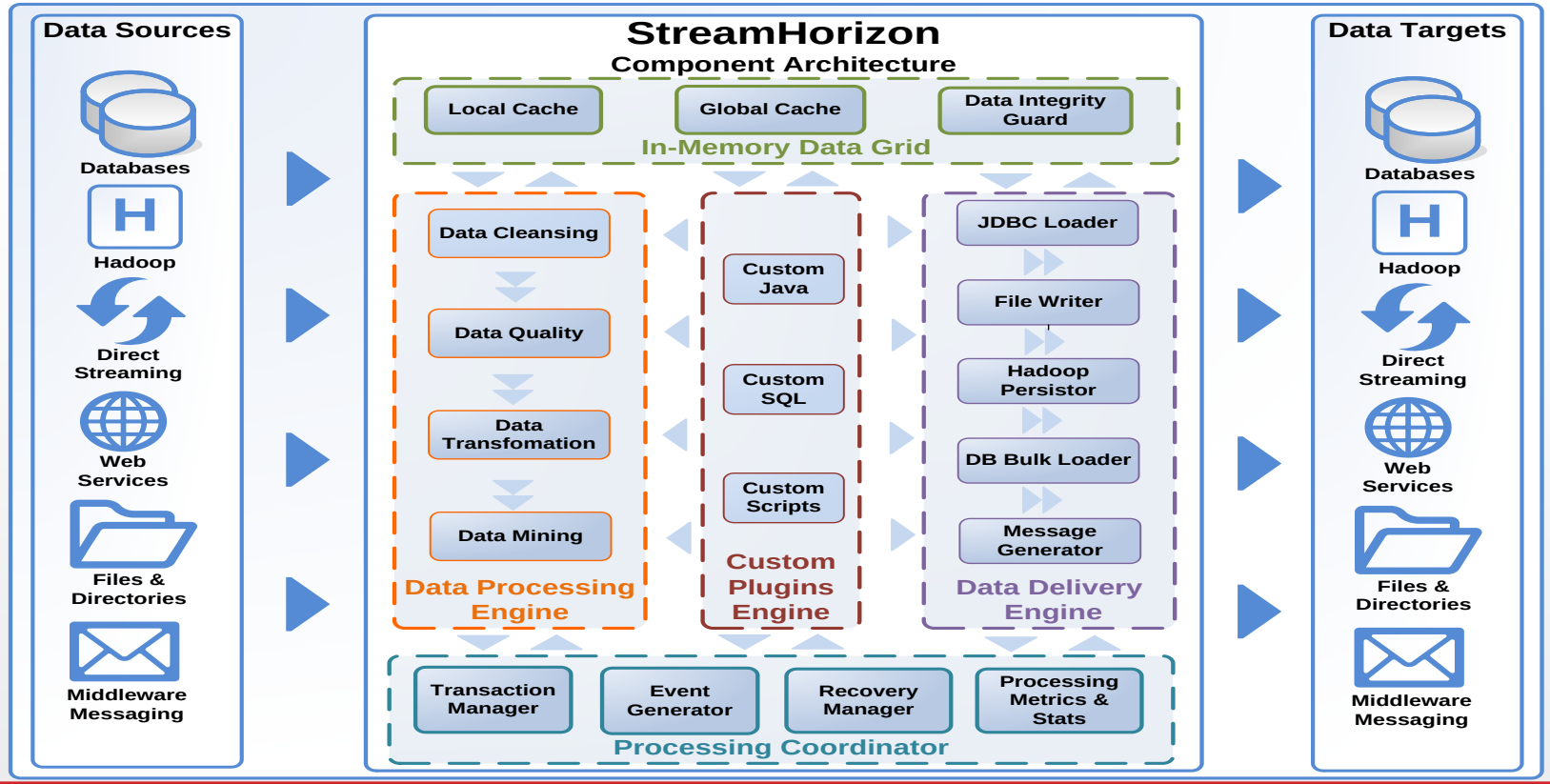
STREAM HORIZON

Architecture

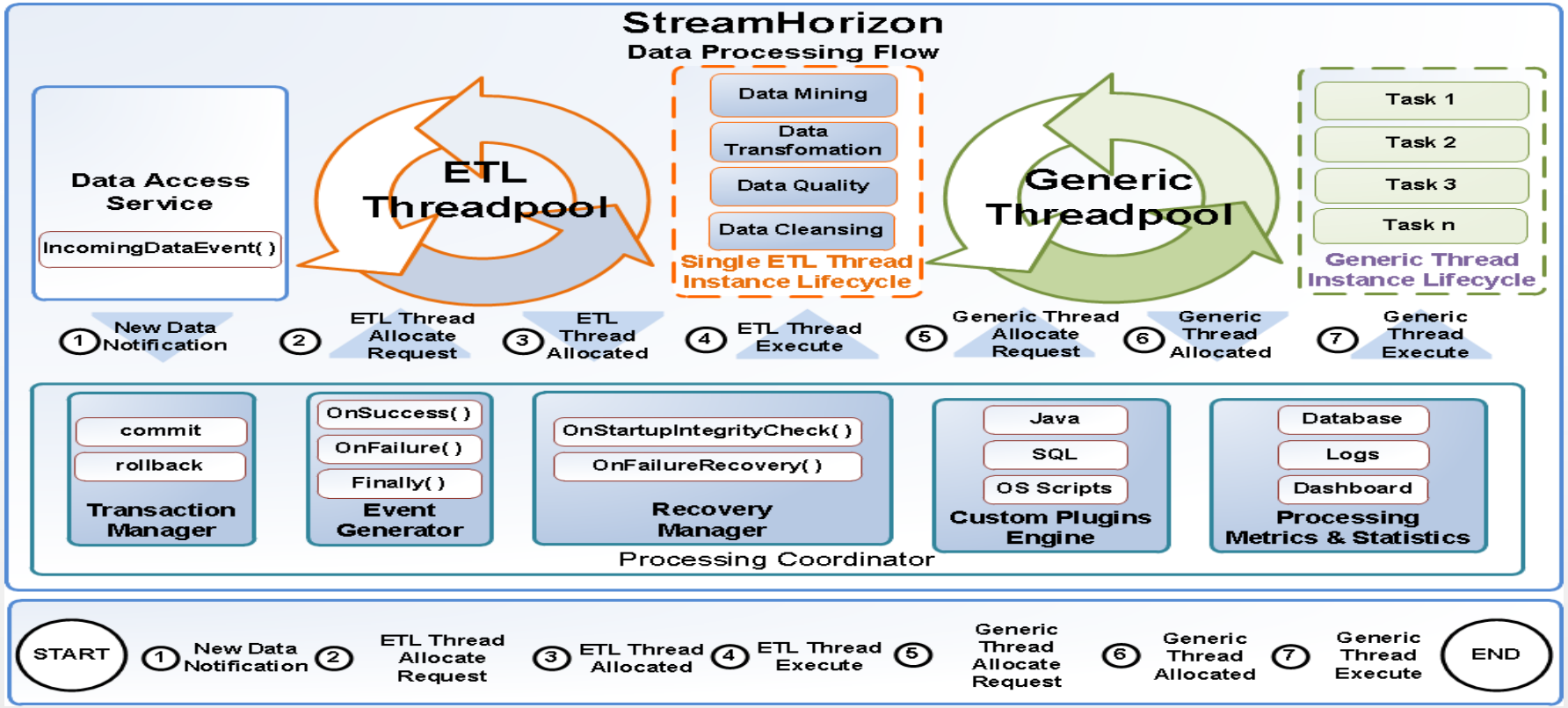
High Level Architecture



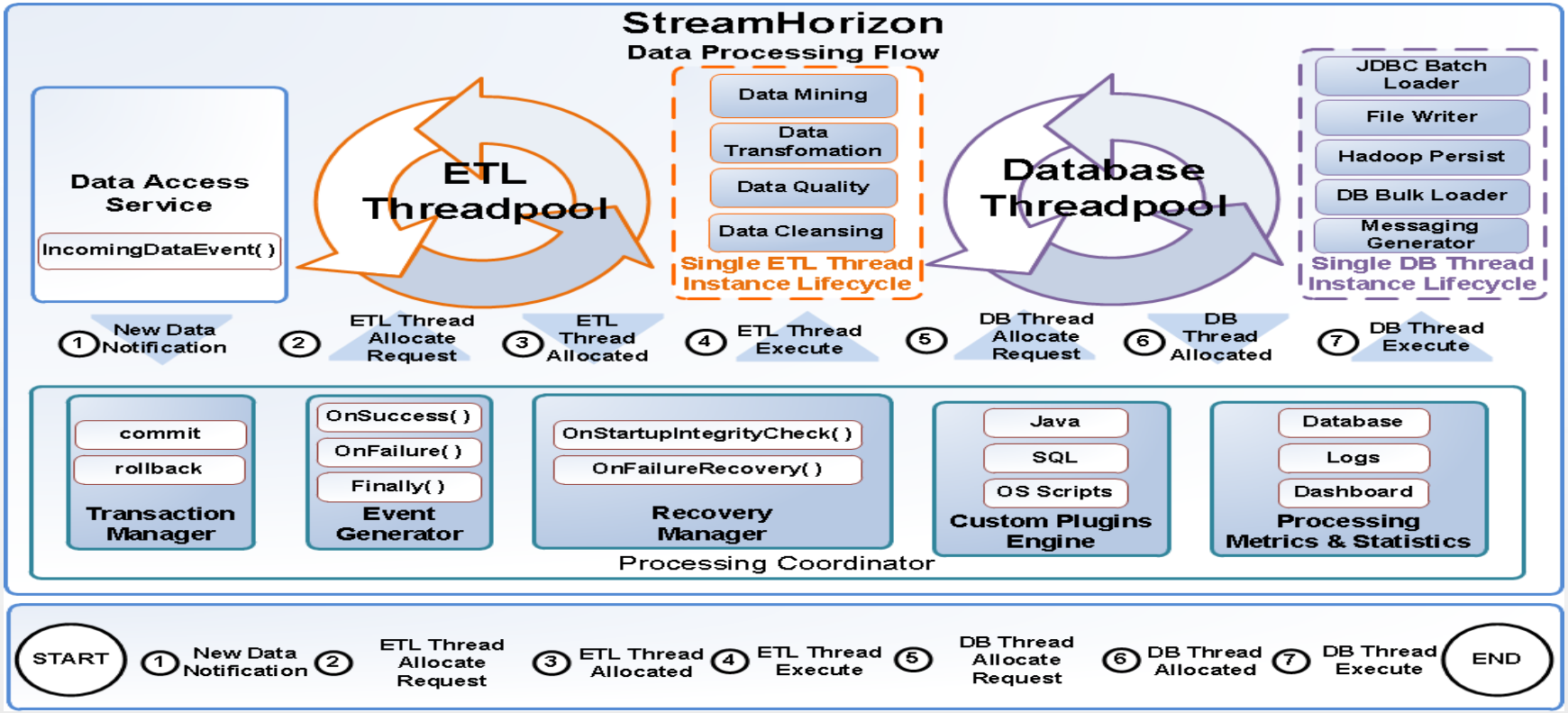
Component Architecture



Data Processing Flow (Generic Target)



Data Processing Flow (RDBMS Database as a Target)



STREAM HORIZON

Benchmarks

Benchmarks

	Reading Method	Writing Method	Database Load Type	Throughput	Deployment Method	Instances & Threadpools	Comment

Benchmark 1	Buffered File Read (JVM Heap)	Off	Off	**2.9** million/sec	Local Cluster (single server)	3 instances x 12 ETL threads per instance	• This is theoretical testing scenario (as data is generally always persisted)
							• Shows StreamHorizon Data Processing Ability without external bottlenecks
							• Eliminates bottleneck introduced by:
							  • DB (DB persistence is switched Off)
							  • I/O (Writing Bulk files is switched Off) and I/O (Reading Feed files is switched Off)

File to File	Read File (SAN)	Write Bulk File or Any other File (feed)	Off	**1.86** million/sec	Local Cluster (single server)	3 instances x 12 ETL threads per instance	• Throughput for feed processing with StreamHorizon Local Cluster running on a single physical server • This test shows ability to create output feed (bulk file or any other feed (file) for that matter by applying ETL logic to input file (feed)
BULK Load	Read File (SAN)	Write Bulk File (Fact table format)	Bulk Load	**1.9** million/sec	Single Instance (single server)	1 instance x 50 DB Threads	• Throughput for DB Bulk Load (Oracle – External Tables) with single StreamHorizon instance running on a single physical server
JDBC Load	Read File (SAN)	Off	JDBC Load	**1.1** million/sec	Local Cluster (single server)	3 instances x 16 ETL threads	• Throughput for JDBC Load (Oracle) with StreamHorizon Local Cluster running on a single physical server

Benchmarks (setup & dependency)

Benchmarks performed with:

- 200 million loaded records
- Individual file size of 100K records
- Each record :

  minimum 200 bytes / 20 attributes
- Target schema is Data Mart
- Kimball design methodology
- Fact table contains:
  - 6 Dimensions & 3 Measures
  - Single dimension cardinality up to 1200
- Data generated by Quant Library (meaning 'partially' sorted)
- Performing housekeeping tasks

Housekeeping tasks performed by StreamHorizon:

- Files moved from source to archive directory
- Bulk files created and deleted during processing
- Metrics gathered (degrades performance on average by approx 20%)
- Files moved from source to error directory (corrupted data feeds)

STREAM HORIZON

Cost - Effectiveness

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 (Exotic Hardware)	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

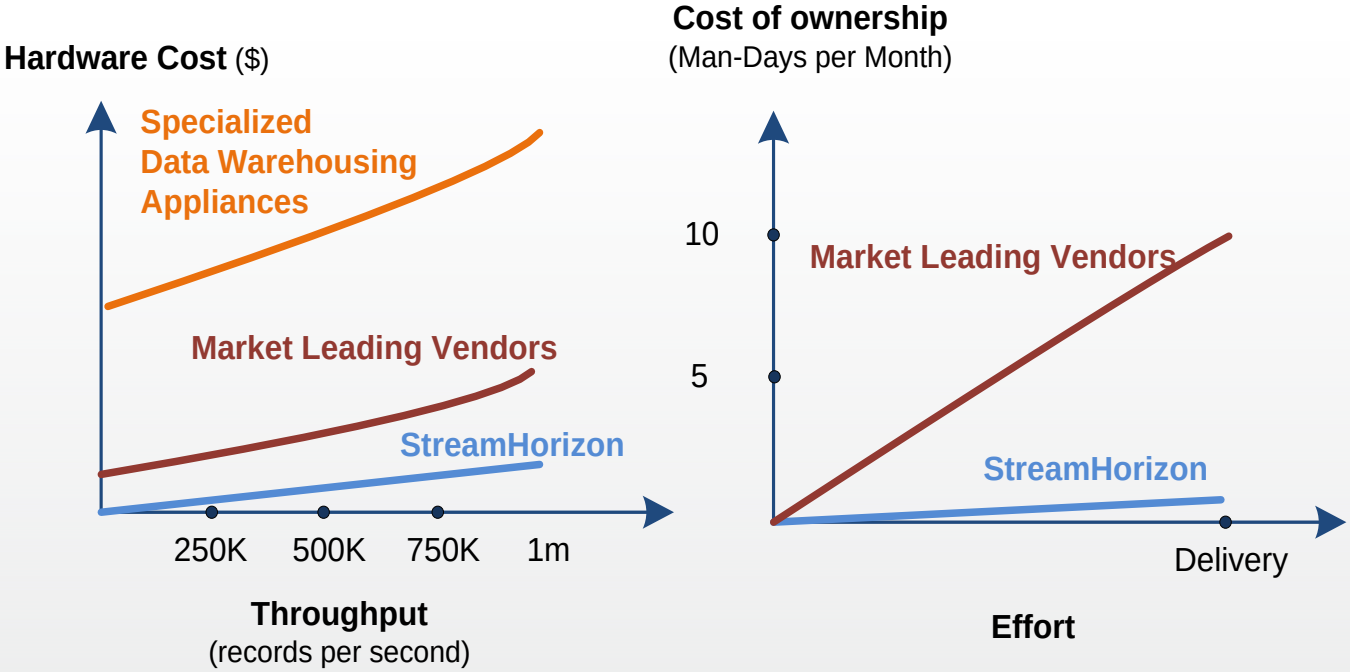
Cost-Effectiveness Analysis - I

Target Throughput of 1million records per second	StreamHorizon	Market Leaders
Hardware Cost	1 unit	3 - 20 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

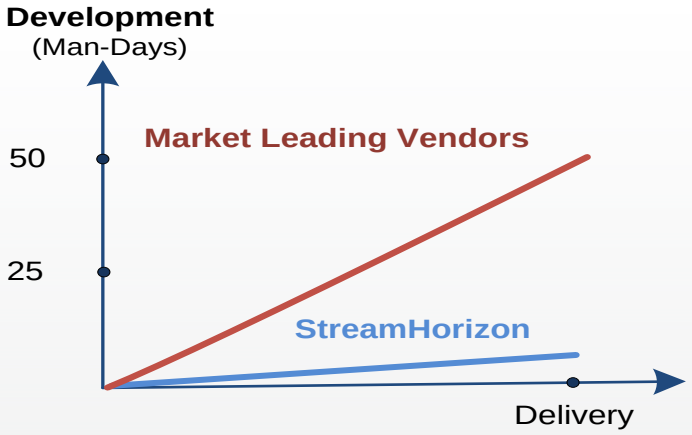
* - Assuming that file data feeds to be processed by StreamHorizon (project dependency) are available at the time of installation

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

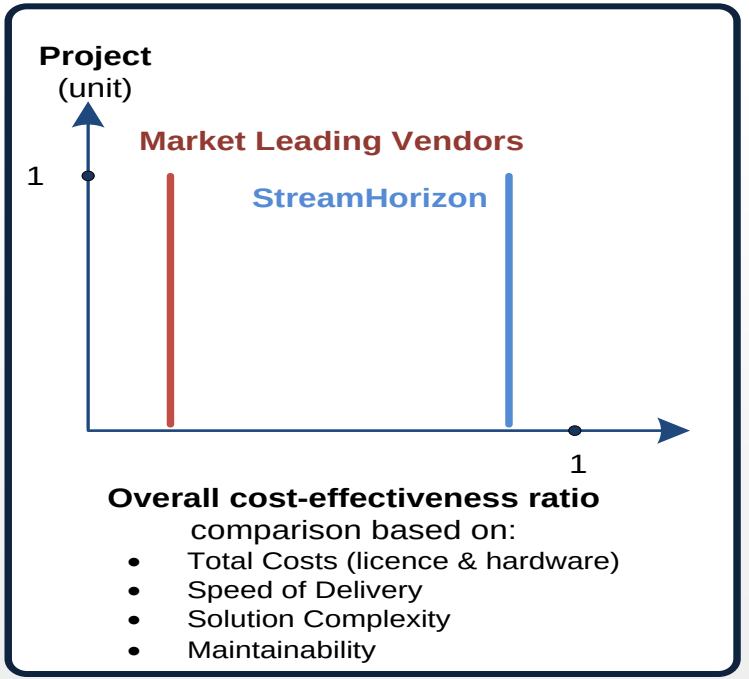
Cost-Effectiveness Analysis - II



Cost-Effectiveness Analysis - III



Time To Market
(Time from Project inception to Project Delivery)



Overall cost-effectiveness ratio
comparison based on:

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

STREAM HORIZON

Use Case study

Use Case Study

Delivering Market Risk system for Tier 1 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).

STREAM HORIZON

ROI & Risk Management

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

- Install at one location – run everywhere
- 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

- **Infraction** - 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)

Deployment Topologies

StreamHorizon Connectivity Map

Relational Databases

- ORACLE
- MSSQL
- DB2
- Sybase
- SybaseIQ
- Teradata
- MySQL
- H2
- HSQL
- PostgreSQL
- Derby
- Informix
- Any Other JDBC compliant...

Non-Relational Data Targets

- HDFS
- MongoDB
- Cassandra
- Hbase
- Elastic Search
- TokuMX
- Apache CouchDB
- Cloudata
- Oracle NoSQL Database
- Any Other via plugins...

Messaging

- JMS (HornetQ, ActiveMQ)
- AMQP
- Kafka
- Any Other via plugins...

Hadoop ecosystem

- Spark
- Storm
- Kafka
- TCP Streams
- Netty
- Hive
- Impala
- Pig
- Any Other via plugins...

Non-Hadoop file systems

- Any file system excluding mainframe

Acting as

- Hadoop 2 Hadoop Data Streaming Platform
- Conventional ETL Data Processing Platform
- Hadoop 2 Non-Hadoop ETL bridge
- Non-Hadoop 2 Hadoop ETL bridge

StreamHorizon OLAP Integration

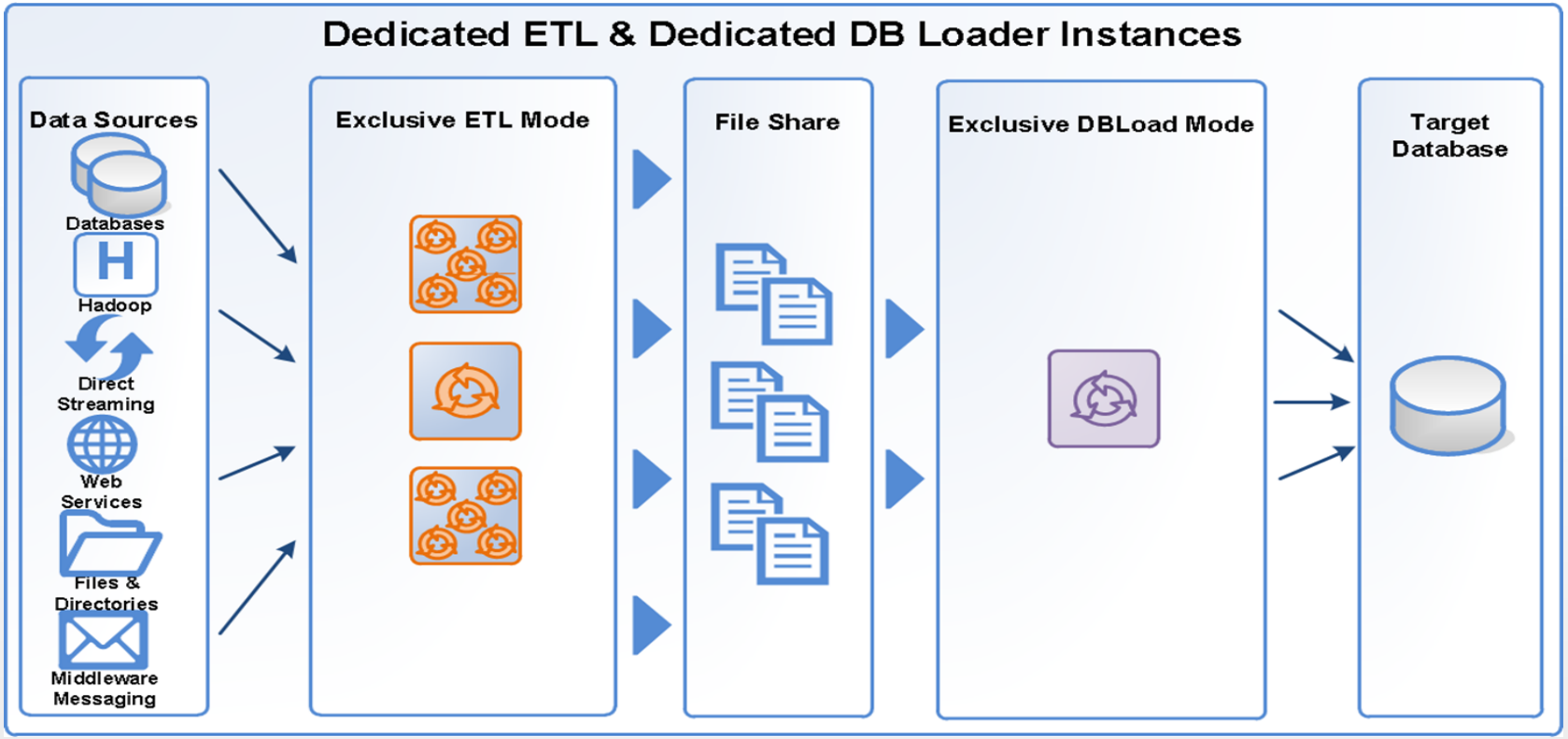
- Seamless integration with OLAP & In-Memory servers

- Supported integration modes:
    - **Real Time** – Data uploaded to the OLAP server immediately after every successful completion of ETL or DB task by StreamHorizon
    - **Batch Oriented** – Data uploaded to the OLAP server in batches rather than immediately after processing of each data entity (file, message, SQL query)

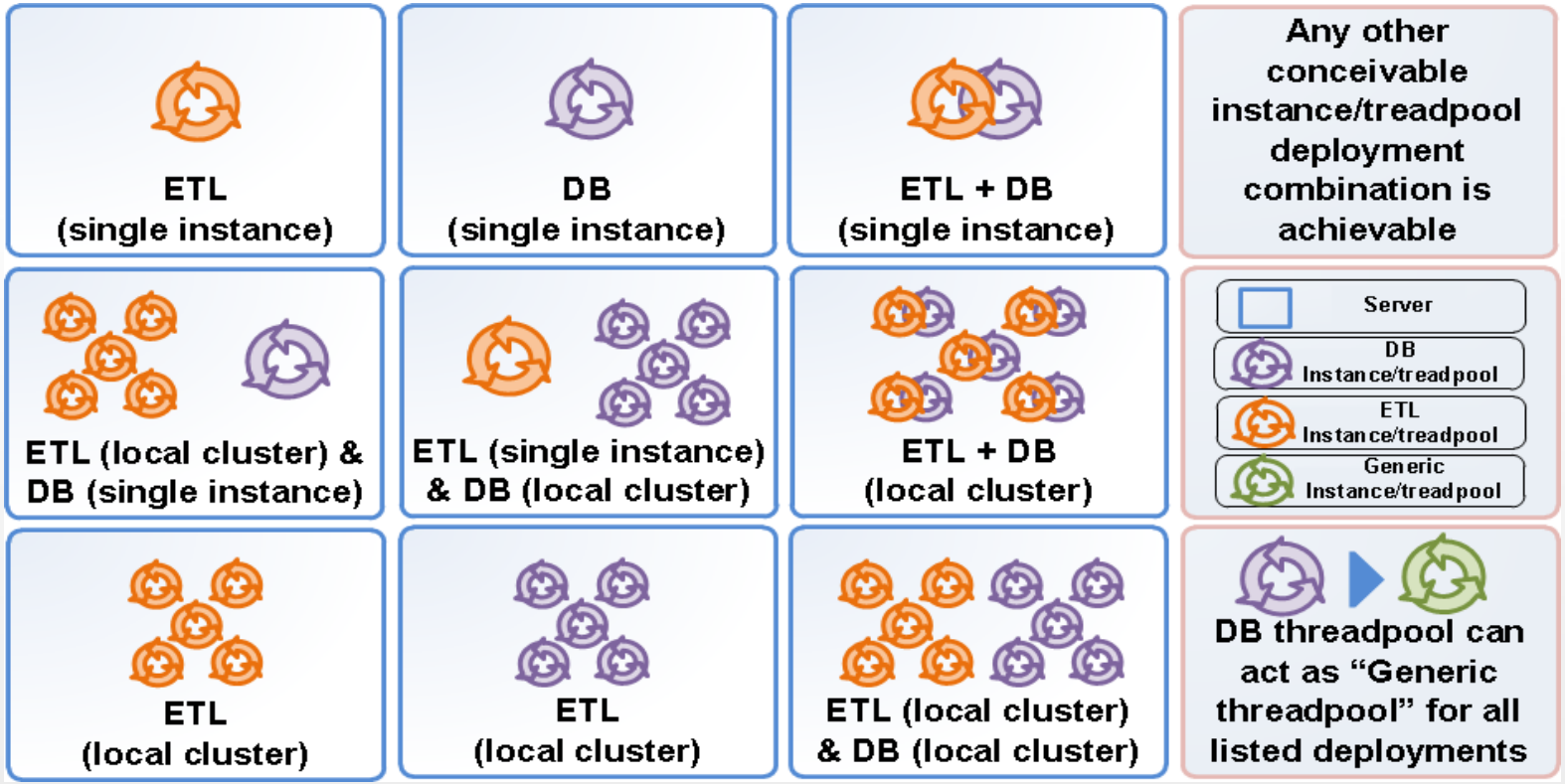
- OLAP integration is available in all StreamHorizon operational modes
    - JDBC
    - Bulk Load
    - Thrift
    - Hadoop
    - Messaging
    - Any other connectivity mode supported by StreamHorizon

- More information is available on StreamHorizon FAQ page

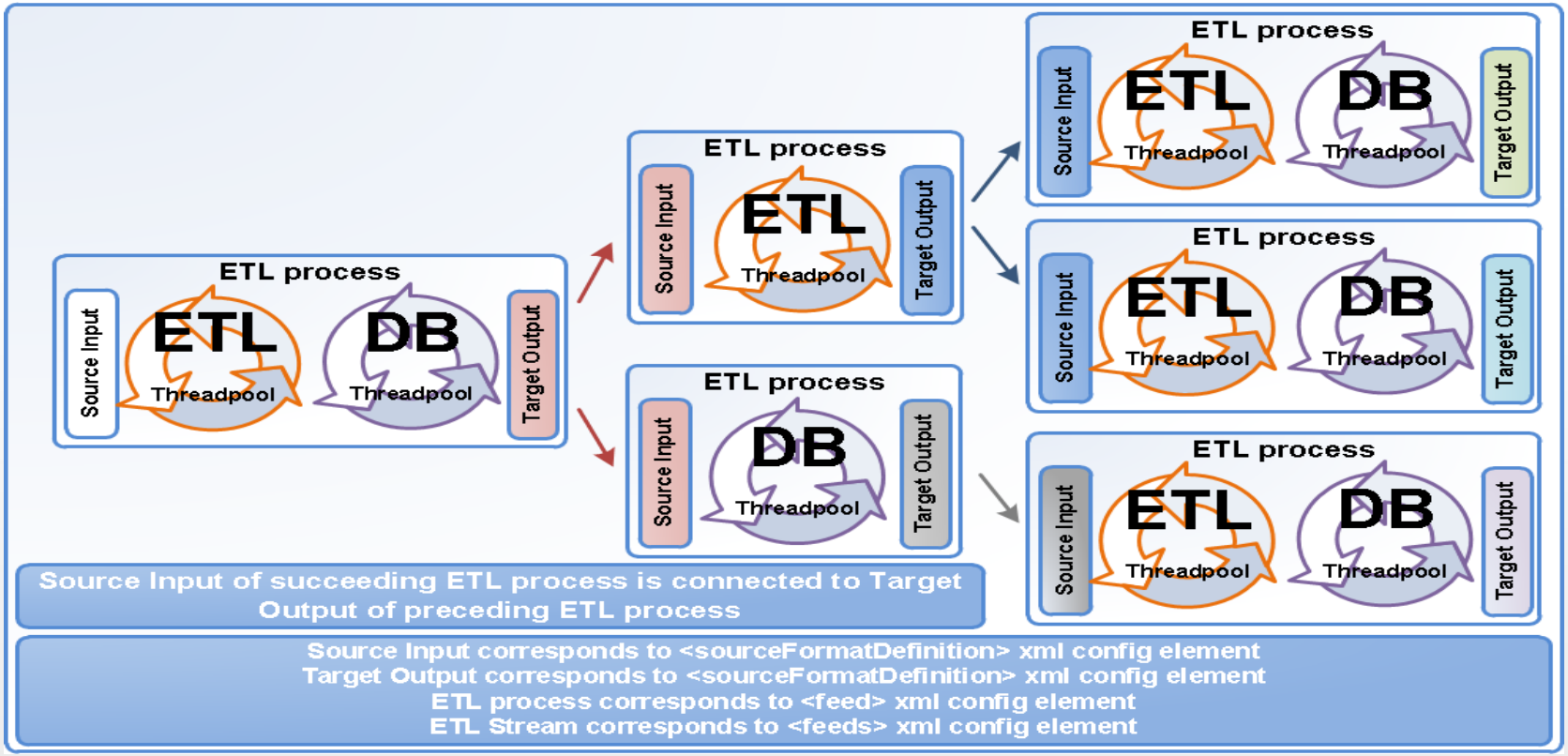
Deployment – Biased



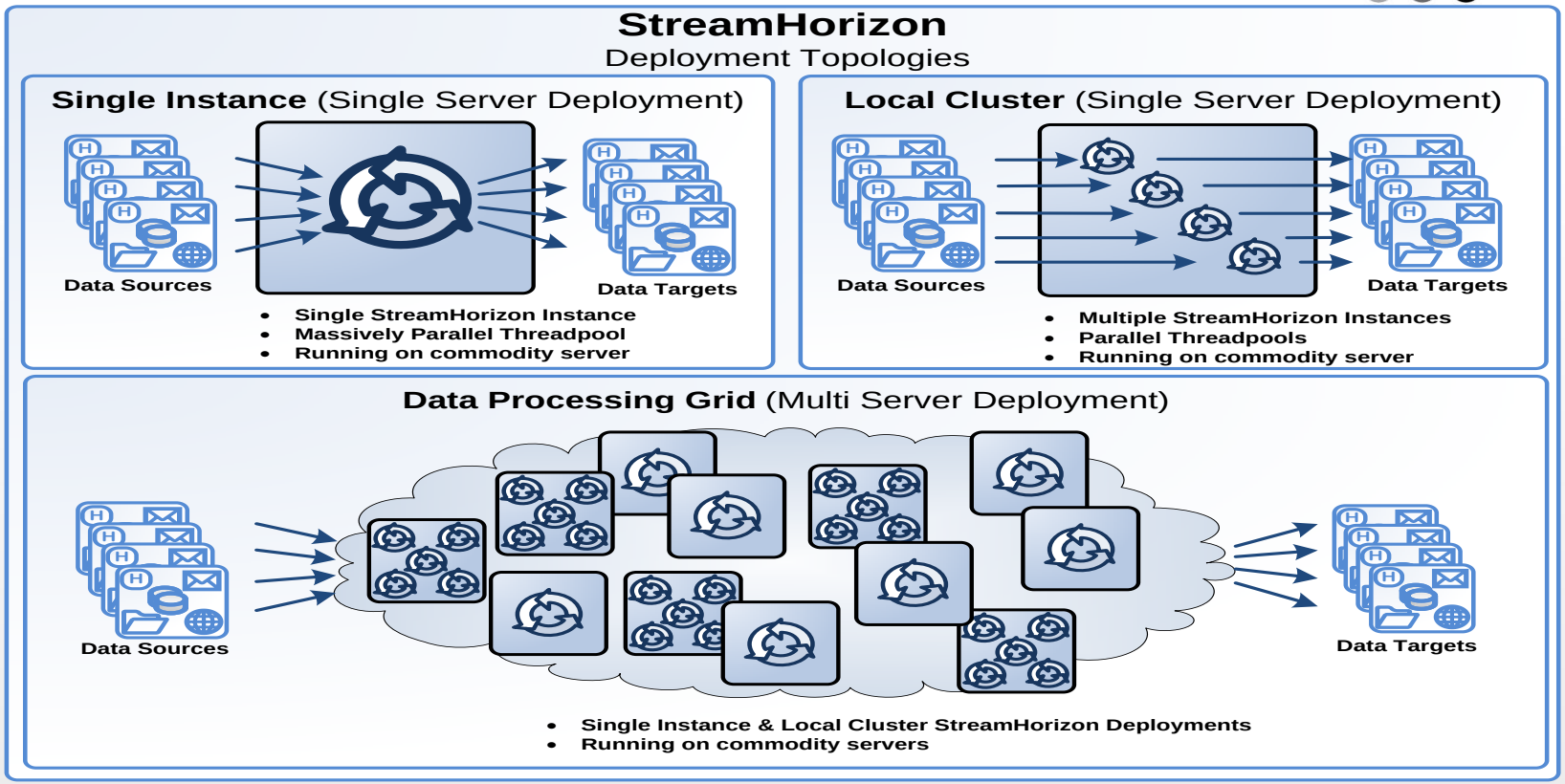
Configuration Options



ETL Streams – Complex & Integrated ETL Flows



Deployment – Vanilla & Grid



Data Processing Grid - ETL cluster

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 & XML

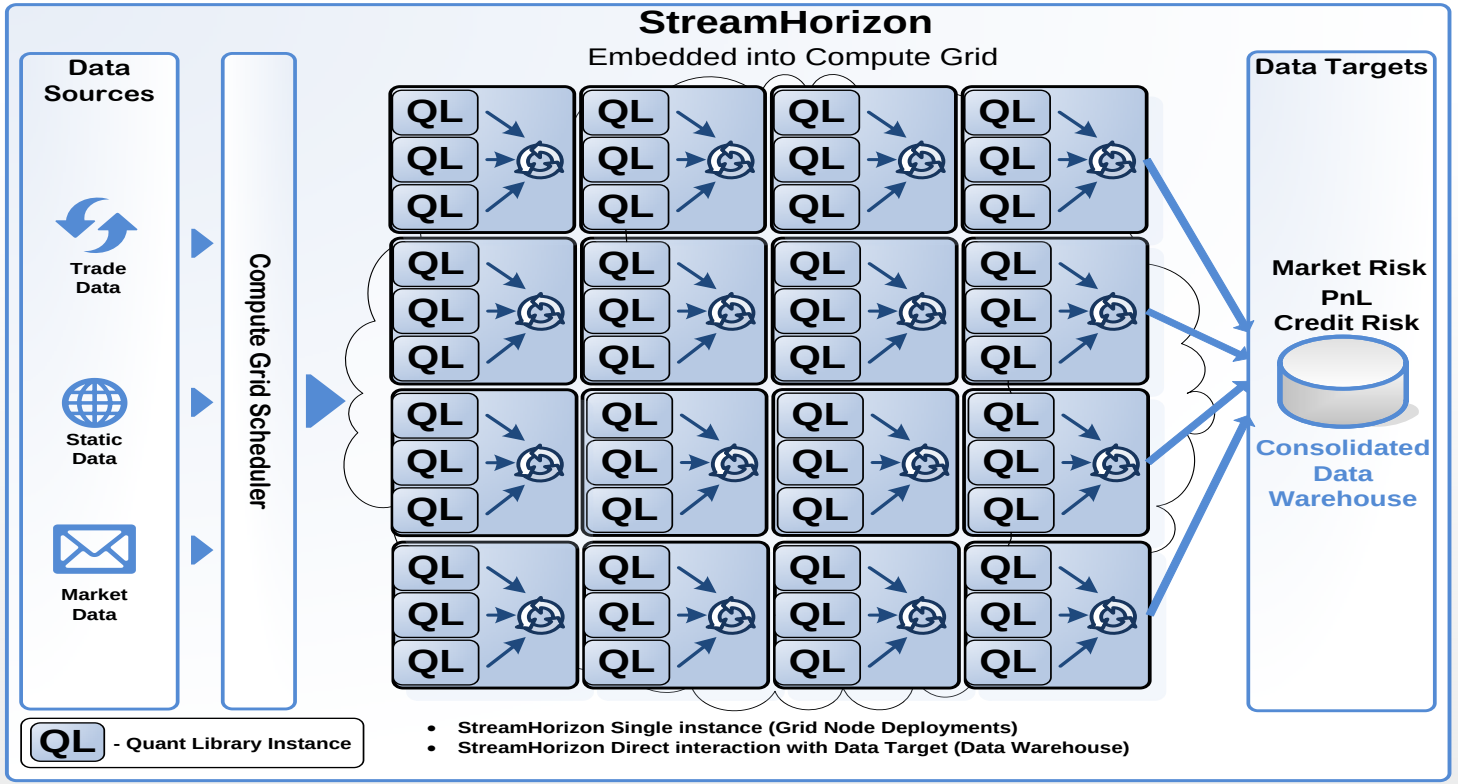
Data Volumes Shipped via Network vs. XML (performance vs. readability)

- 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

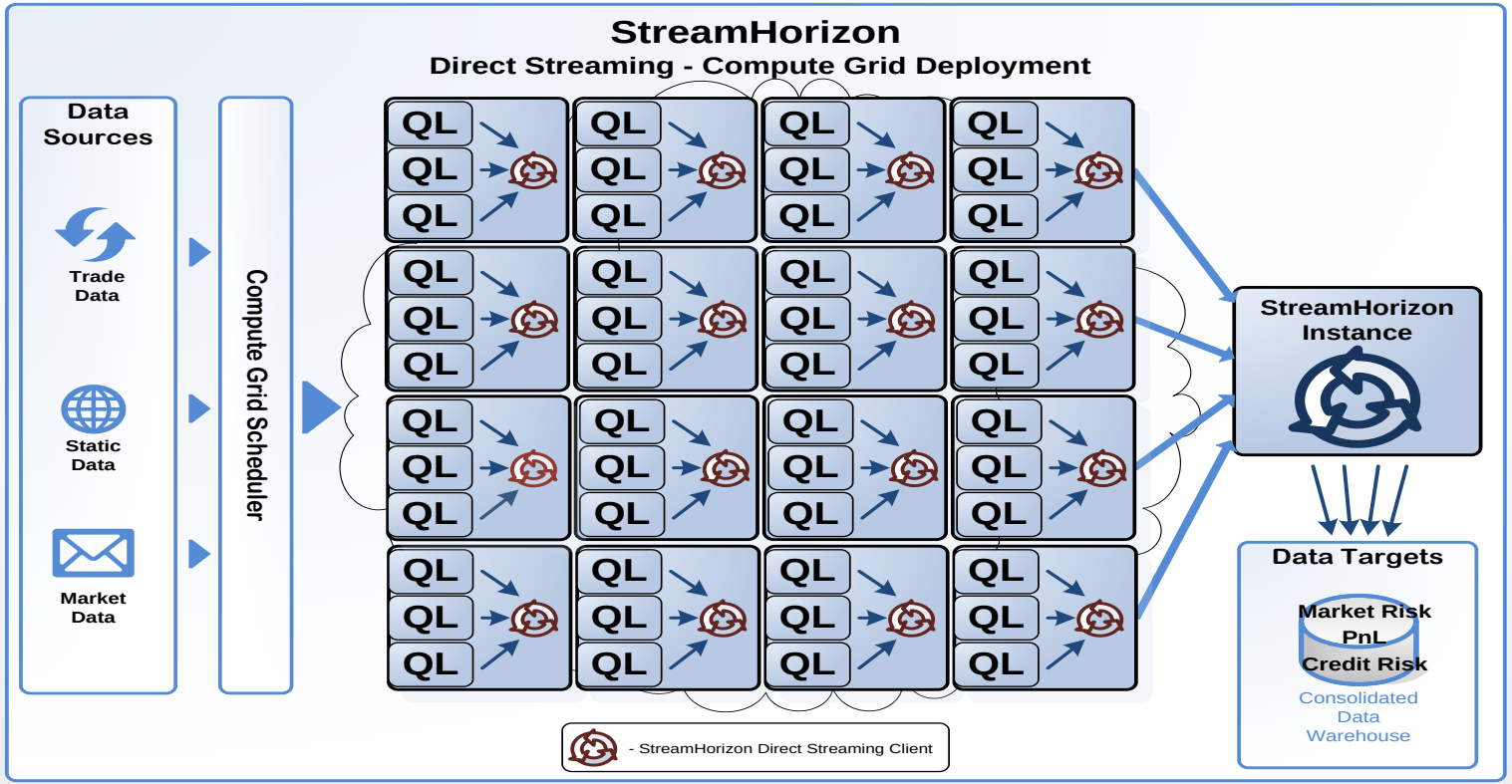
The Solution – Best of both worlds

- 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)



Compute Grid – Data Streaming (Example: Finance Industry)



Big Data Analytics - Accelerated

StreamHorizon & Big Data

Integrates into your Data Processing Pipeline...

- Seamlessly integrates at any point of your your data processing pipeline
- Implements generic input/output HDFS connectivity.
- Enables you to implement your own, customised input/output HDFS connectivity.
- Ability to process data from heterogeneous sources like:
  - Storm
  - Kafka
  - TCP Streams
  - Netty
  - Local File System
  - Any Other...

Accelerates your clients...

- Reduces Network data congestion
- Improves latency of Impala, Hive or any other Massively Parallel Processing SQL Query Engine.
  - *Impala*
  - Hive
  - HBase
  - Any Other...

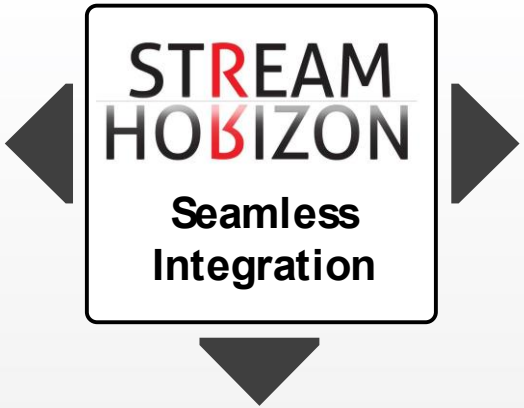
And more...

- Portable Across Heterogeneous Hardware and Software Platforms
- Portable from one platform to another

StreamHorizon - Flavours – Big Data Processing

Storm - Reactive, Fast, Real Time Processing

- Guaranteed data processing
- Guarantees no data loss
- Real-time processing
- Horizontal scalability
- Fault-tolerance
- Stateless nodes
- Open Source



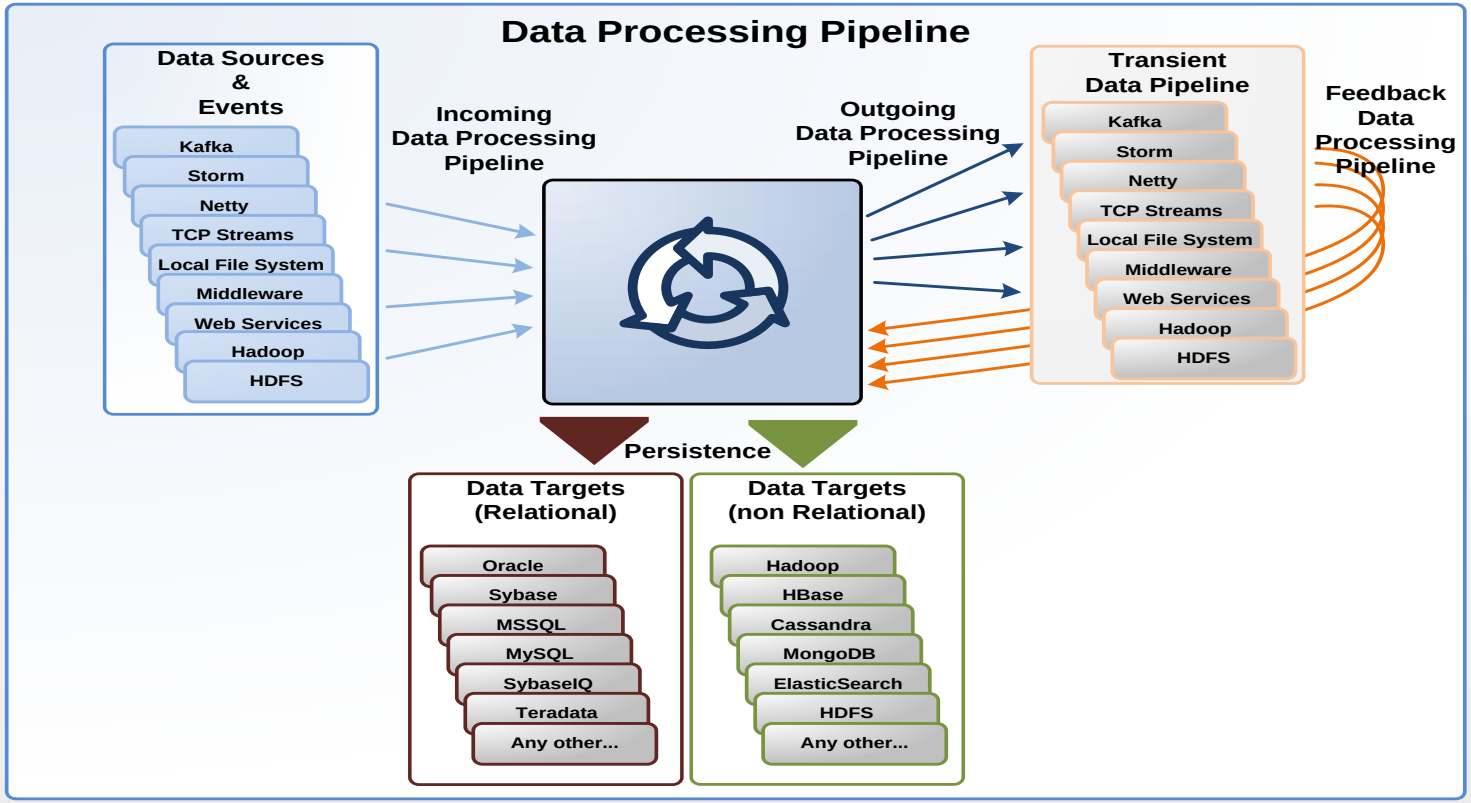
Hadoop – Big Batch Oriented Processing

- Batch processing
- Jobs runs to completion
- Stateful nodes
- Scalable
- Guarantees no data loss
- Open Source

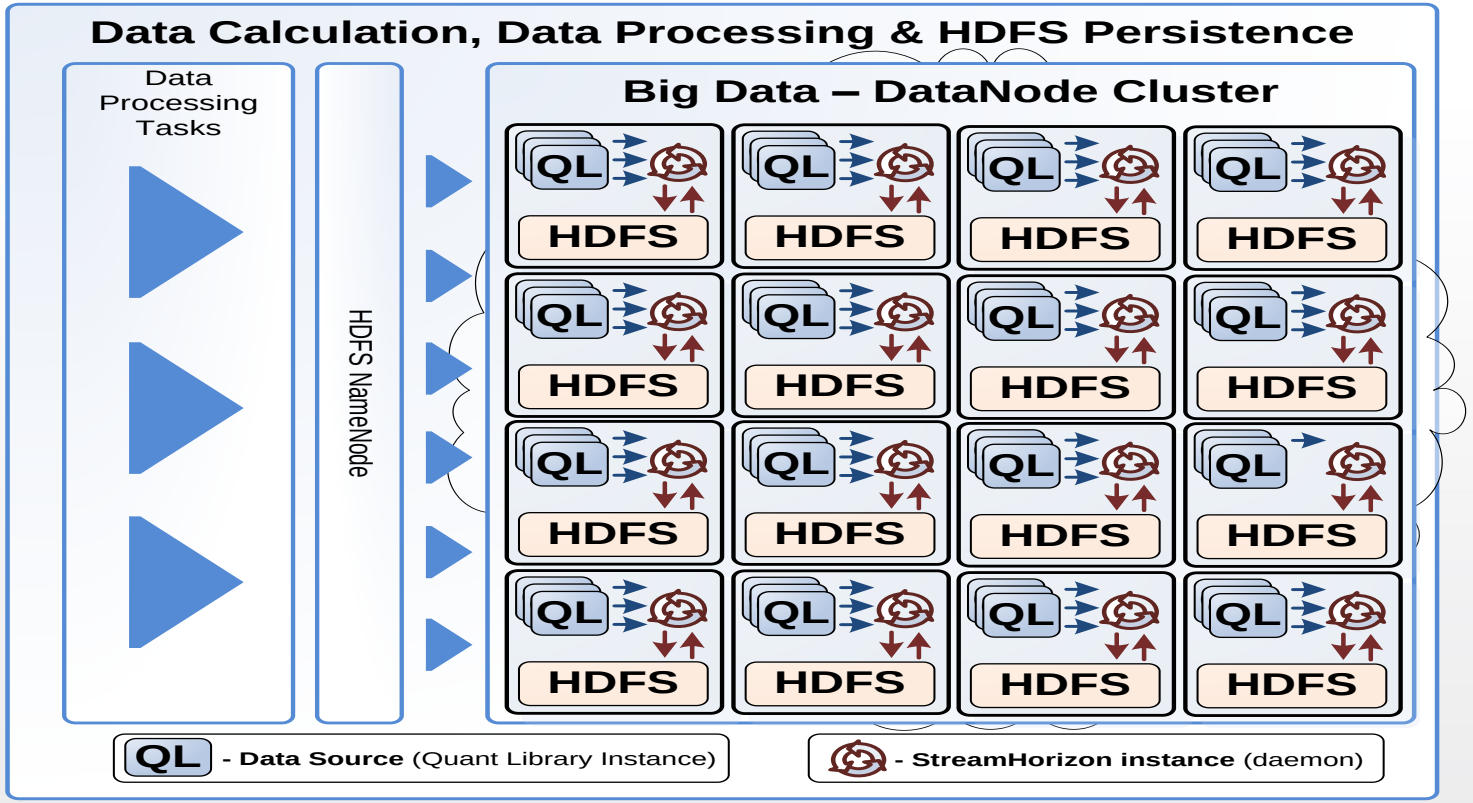
Kafka

- Designed for processing of real time activity stream data (metrics, KPI's, collections, social media streams)
- A distributed Publish-Subscribe messaging system for Big Data
- Acts as Producer, Broker, Consumer of message topics
- Persists messages (has ability to rewind)
- Initially developed by LinkedIn (current ownership of Apache)

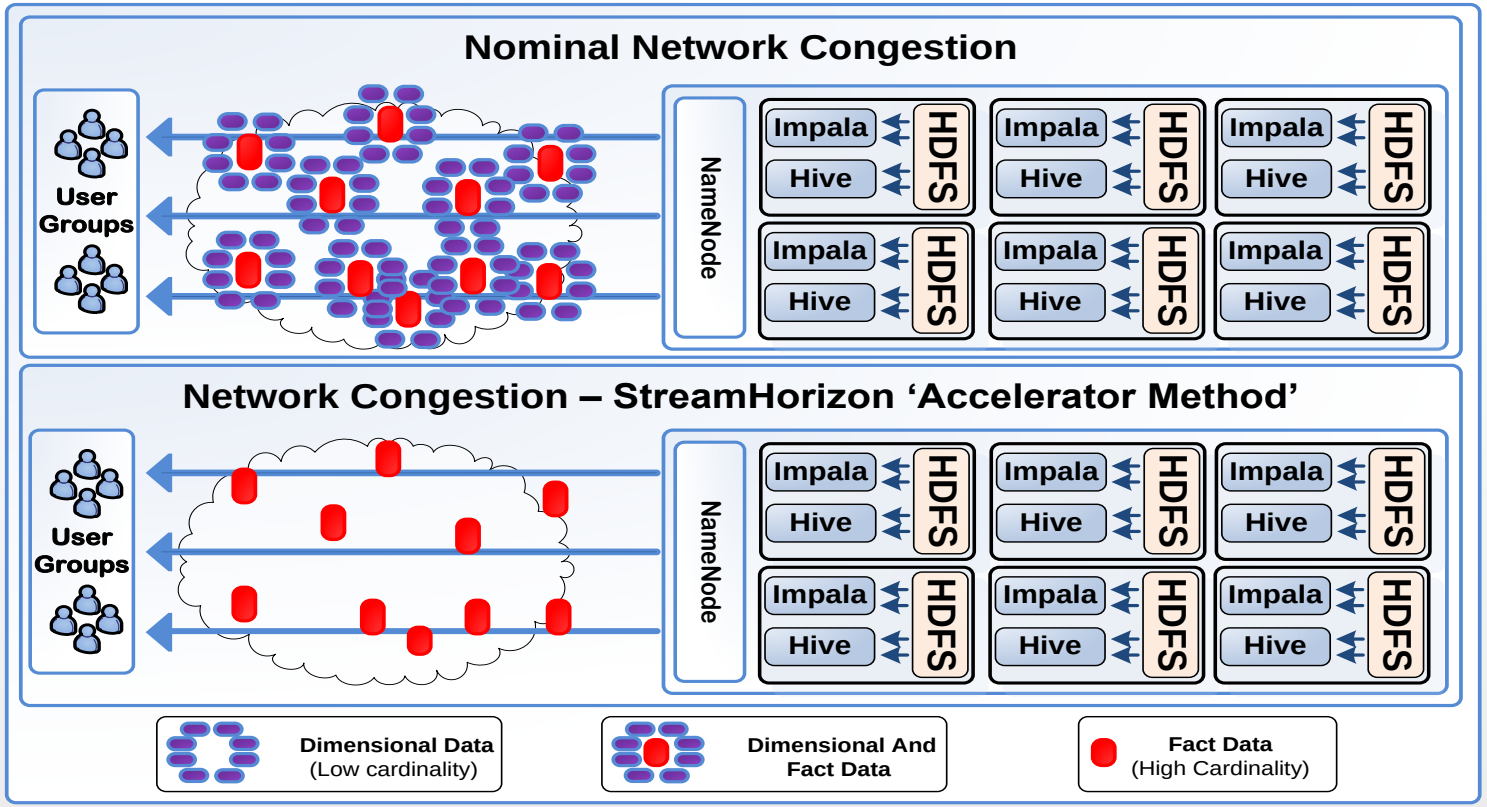
StreamHorizon – Big Data Processing Pipeline



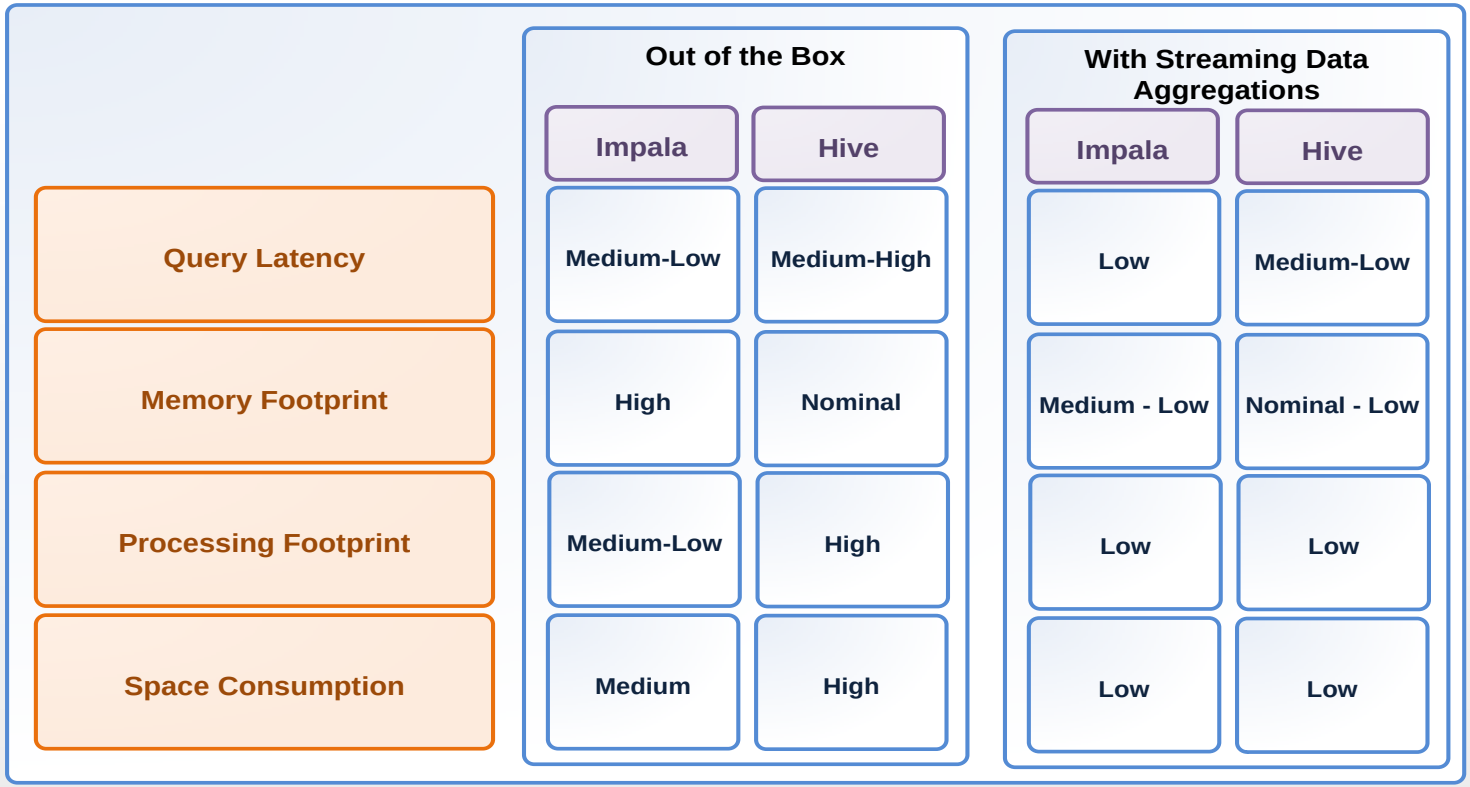
Big Data & StreamHorizon – Data Persistence (Example: Finance Industry)



Big Data & StreamHorizon – Data Retrieval (Example: Finance Industry)



Streaming Data Aggregations – impact on Big Data Query Latency



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

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 2.1 GHz 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)

STREAM HORIZON

Q&A