

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 if 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 & others)
- **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 - 0.000000666 seconds 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



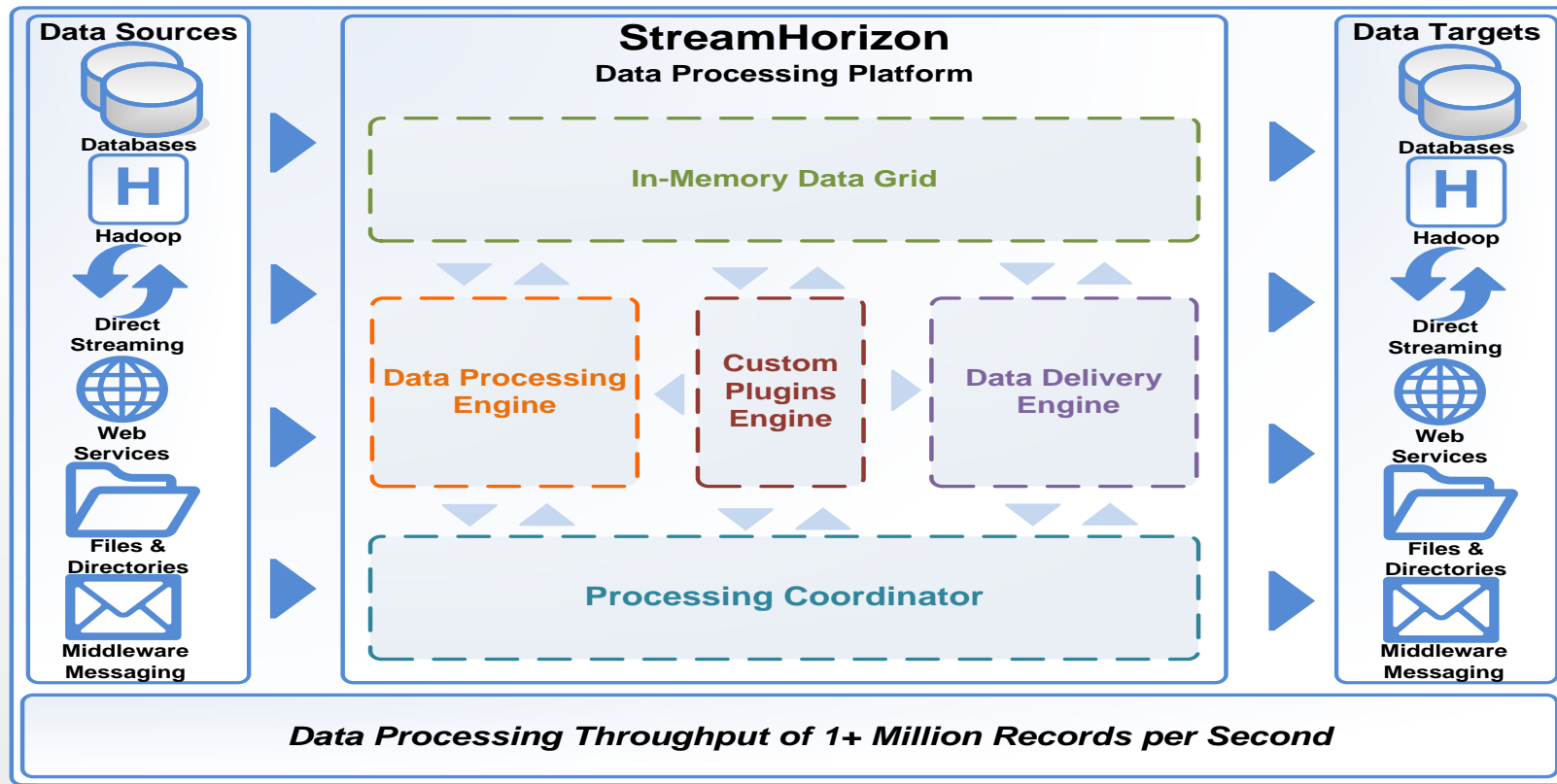
ISP - User Activity Analysis & Profiling, Log Data Mining, Behavioral Analysis

# STREAM HORIZON



## Architecture

# High Level Architecture



# STREAM HORIZON

Three dots are positioned to the right of the 'STREAM HORIZON' text, aligned with the horizontal line that separates the two words.

## 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	<ul style="list-style-type: none"> <li>This is theoretical testing scenario (as data is generally always persisted)</li> <li>Shows StreamHorizon Data Processing Ability without external bottlenecks</li> <li>Eliminates bottleneck introduced by: <ul style="list-style-type: none"> <li>DB (DB persistence is switched Off)</li> <li>I/O (Writing Bulk files is switched Off) and I/O (Reading Feed files is switched Off)</li> </ul> </li> </ul>
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	<ul style="list-style-type: none"> <li>Throughput for feed processing with StreamHorizon Local Cluster running on a single physical server</li> <li>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)</li> </ul>
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	<ul style="list-style-type: none"> <li>Throughput for DB Bulk Load (Oracle – External Tables) with single StreamHorizon instance running on a single physical server</li> </ul>
JDBC Load	Read File (SAN)	Off	JDBC Load	1.1 million/sec	Local Cluster (single server)	3 instances x 16 ETL threads	<ul style="list-style-type: none"> <li>Throughput for JDBC Load (Oracle) with StreamHorizon Local Cluster running on a single physical server</li> </ul>













# 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

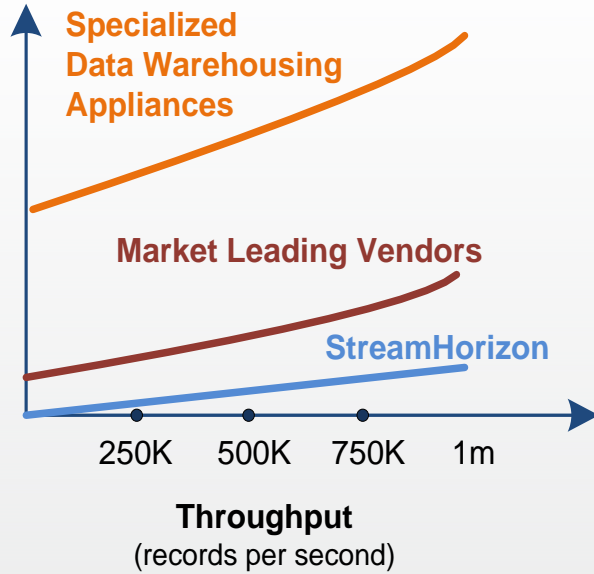
\*\* - Please refer to end of this document for detailed description of hardware environment



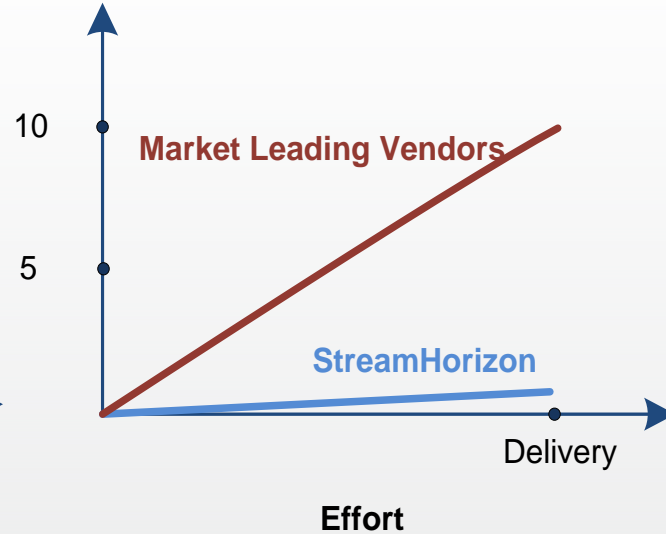
## Cost-Effectiveness Analysis - II



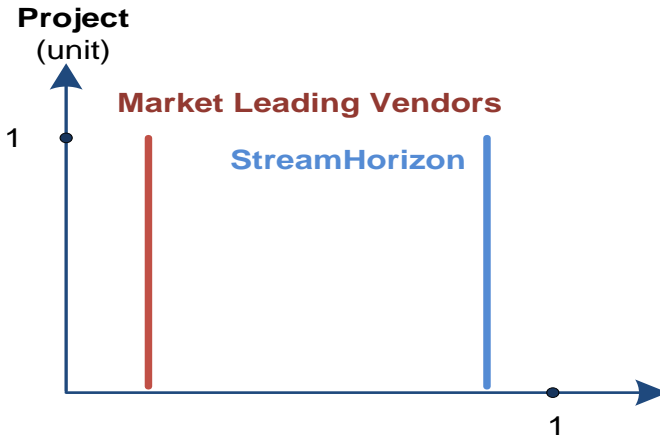
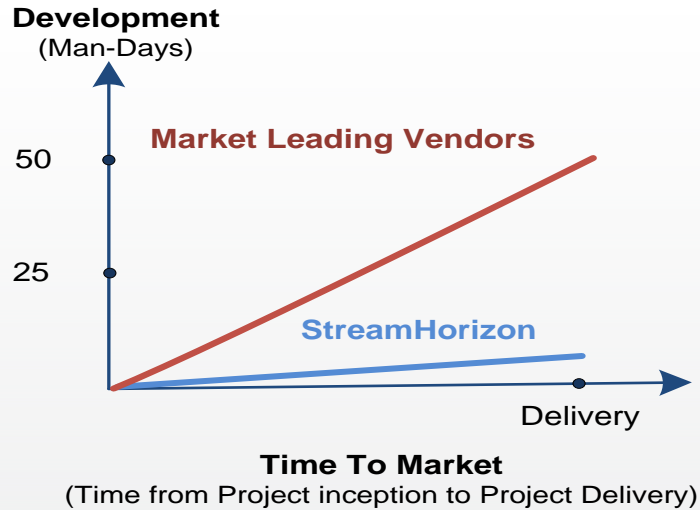
**Hardware Cost (\$)**



**Cost of ownership**  
(Man-Days per Month)



## Cost-Effectiveness Analysis - III



### 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 budget 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



## Q&A