Yellowbrick

TECHNICAL OVERVIEW

Yellowbrick Technical Overview

We've re-imagined MPP analytic database architecture to build the fastest, most flexible data warehouse on the planet.

Yellowbrick Data Warehouse provides near-real time answers from data at any scale, for any workload, in any environment. Its uniquely efficient use of resources (processor, memory, storage, and network) in an adaptive "cut-through" architecture helps you solve data processing-intensive problems you couldn't solve before while getting the best possible performance value from the infrastructure available. Yellowbrick also offers flexibility for enabling distributed clouds, with deployment options including on Andromeda optimized instances in private clouds, as a service or instance in public clouds, and on K8S containers at the edge, with all data warehouse managed via a single, unified control plane (Yellowbrick Manager).

Yellowbrick was conceived with the goal of optimizing price/performance. New SQL analytics use cases are emerging all the time, and more concurrent users are consuming more ad hoc analytics. That requires more performance per dollar spent, and Yellowbrick architecture leapfrogs the industry in this respect.

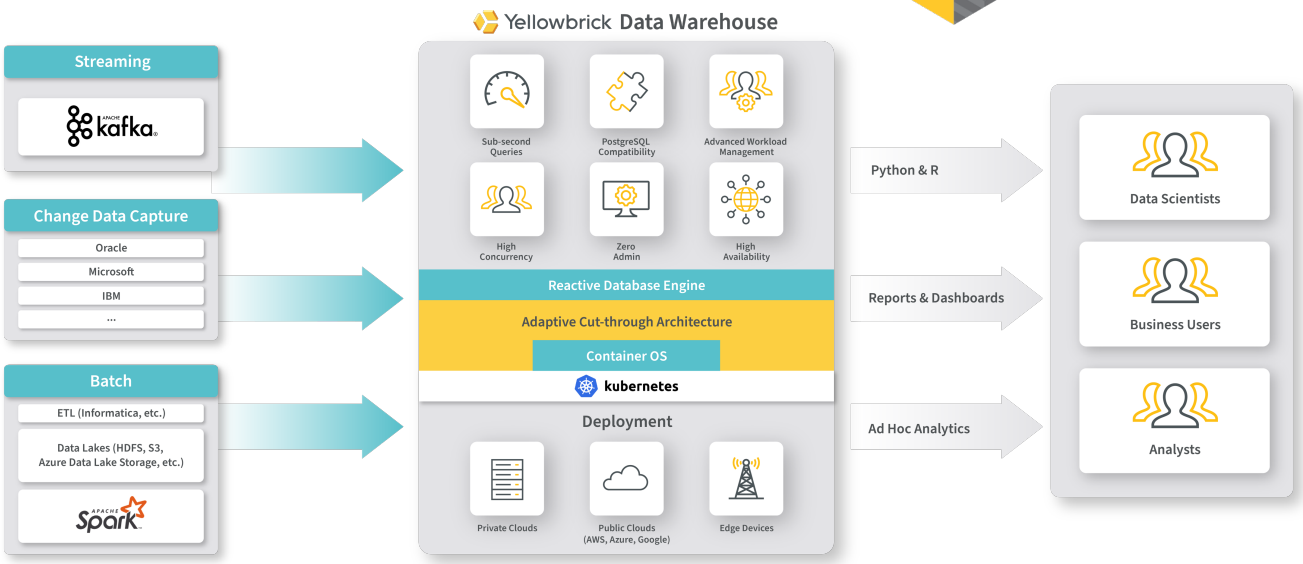
In the cloud, Yellowbrick's efficiency brings big savings compared to rivals combining unremarkable performance with consumption pricing. It's not uncommon for 16 nodes of Yellowbrick to outperform 128 nodes of our closest competitor on both performance and concurrency, and by healthy margins. This means you will solve the same problem in a fraction of the time, and at a fraction of the cost.

Yellowbrick also addresses workload needs with incredible flexibility. Inside Yellowbrick, data is actively managed in a hybrid row and column store. The front-end row store minimizes commit latency for real-time inserts, while the back-end column store handles massive ad hoc queries, giving customers the ability to address the most demanding workloads with ease.

To follow are examples of some key Yellowbrick innovations. For a comprehensive overview of Yellowbrick data warehouse architecture, see our technical white paper at **yellowbrick.com/whitepaper**.

Key features

- Effective storage scan rates of TBs per second, backed by 200Gb/sec RDMA networking
- Linear scalability to 6PB across 40 nodes with no aggregators/leader
- No partitions, cubes, or user-created indexes required
- Native real-time streaming (millions of rows/sec) and high-volume batch insert (10TB/hour)
- Supports queries and joins of external tables in S3, Azure Data Lake Storage v2, and NFS
- ANSI SQL-compliant syntax with PostgreSQL dialect
- Works out of the box with common BI, data motion, and ML tools
- Connectors for ODBC, JDBC, ADO.NET, Kafka, and Spark Streaming



Yellowbrick Kernel

To avoid performance limitations in Linux, we built a new Linux-based OS kernel from scratch that achieves maximum CPU efficiency by keeping memory pinned to the correct core until queries complete. Our optimized non-blocking thread management ensures the same data remains in the CPU caches while queries run. The Yellowbrick Kernel is also deployment-aware and will select which code and drivers to use based on the environment.

Optimized data path/cache-less design

While most databases need to keep hot data in a memory buffer cache for performance, Yellowbrick never caches data; rather, its optimized data path provides the experience of *all* data living in cache *all* the time. Data is read directly from primary storage into the CPU cache, so no cache rewarming is ever needed.

Custom drivers

Yellowbrick borrows concepts from high-performance computing, high-frequency trading, and public cloud stacks to offer the most efficient data warehouse drivers. Our networking drivers are 20X more efficient than standard Linux drivers: We've clocked a single CPU core sending and receiving 16GB/sec!

Cluster parity filesystem

To lower costs while improving availability and reliability, the Yellowbrick cluster filesystem, ParityFS, implements n+2 erasure coding. Data is reconstructed on the fly, and when nodes are replaced, the original data is rebuilt automatically. Non-HA instances, or those with ephemeral storage in public clouds, use Yellowbrick Cloud Mirror technology. This approach, along with its TCO benefits, performance benefits, and integration into MPP database processing, is unique to Yellowbrick.

250 Cambridge Avenue, Suite 300, Palo Alto, California 94131 I USA I 1.650.687.0896