Yellowbrick

OVERVIEW

Engineered for Extreme Efficiency

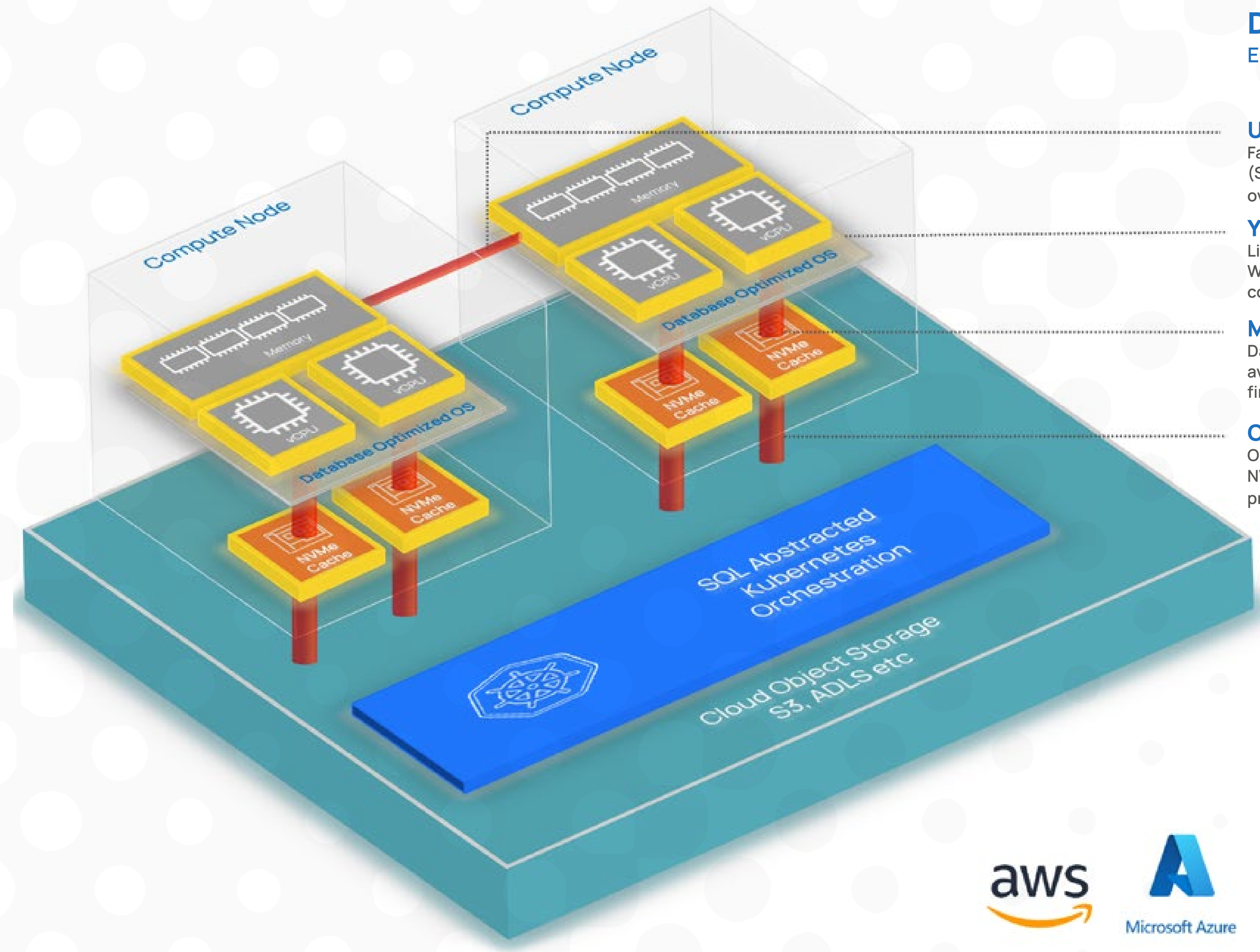
Engineered for Extreme Efficiency

Some of the smartest database minds in the world have been building Yellowbrick's MPP database engine from scratch over the last eight years. The technology underpinning the Yellowbrick Data Warehouse is highly differentiated from competitors – we are not just delivering an MPP layer on top of someone else's database. Yellowbrick's fast, efficient data warehouse processing engine translates into huge resource efficiencies, lower resource and energy consumption, faster queries, and higher workload density – ultimately delivering more productivity and value. The Yellowbrick Data Warehouse unlocks data for every enterprise user with efficiencies driving lower costs.

There are four primary components to the Yellowbrick database engine, all written from scratch by Yellowbrick: the Storage Engine, the Execution Engine, the Workload Manager, and the Query Compiler. In addition, Yellowbrick's engineers have optimized the entire data path and operating system process management, something we call the **Direct Data Accelerator**.

All of these optimizations work silently, behind the scenes whether Yellowbrick is running in the public cloud using commodity compute and storage, or on its optimized hardware appliance, Andromeda.

This overview will walk through some of what makes Yellowbrick technically different.



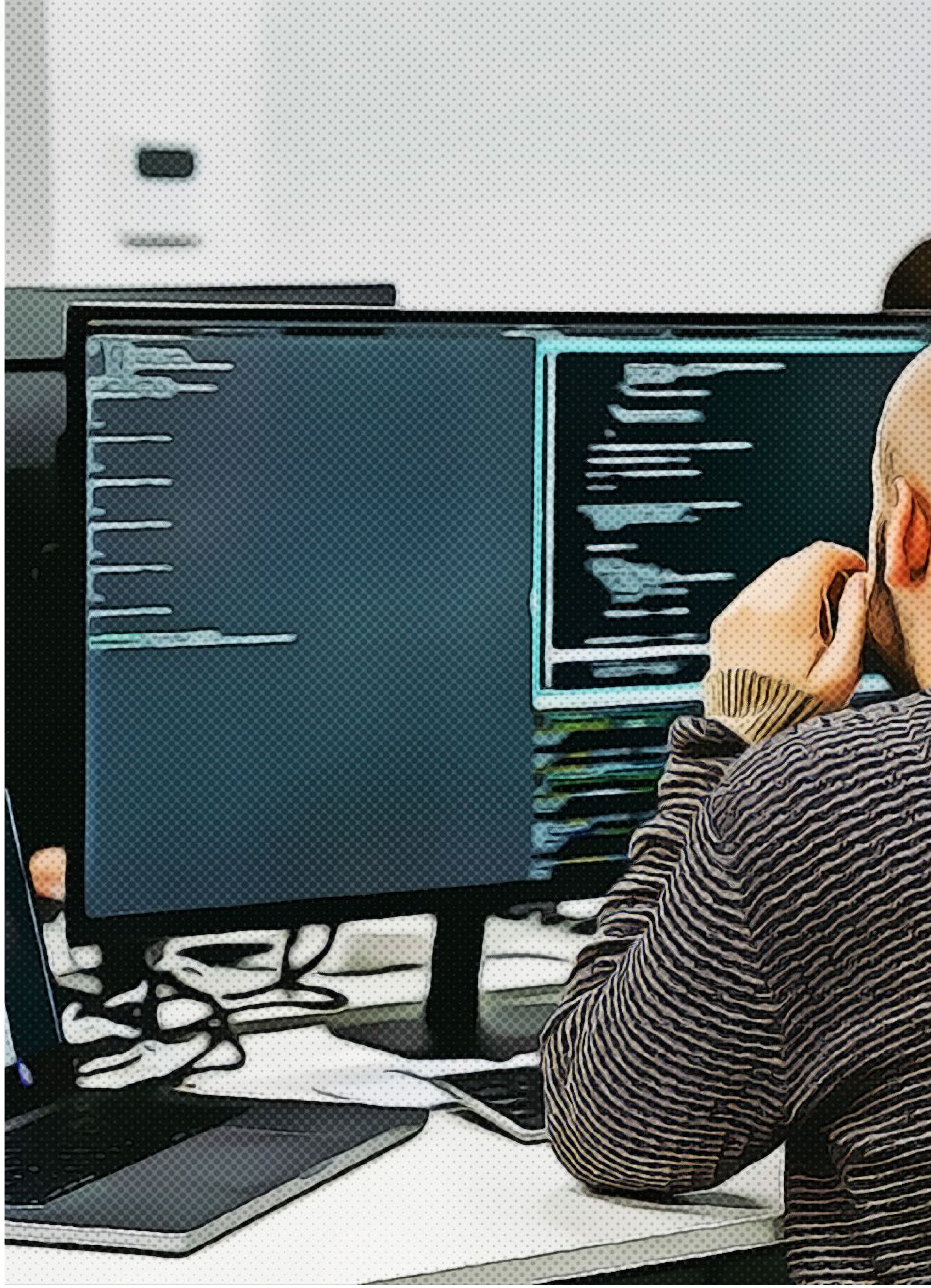
Direct Data Accelerator
Engineered for Extreme Efficiency

Ultra-Low Latency Data Transport
Faster Processing of Expensive Queries (SORT, GROUPBY, JOINS), minimizing CPU overhead, when data moves between nodes

Yellowbrick Database Optimized OS
Linux OS (Unikernel) highly optimized for Data Warehouse workloads, more efficient task completion

Memory Bypass Data Interface
Data ingestion direct to L2/L3 cache on CPU avoiding the need to stage data in main memory first

Object Storage Acceleration
Optimized data ingestion from Object Storage to NVMe Cache with throughput faster than cloud providers' own drivers



Direct Data ***Accelerator***

Yellowbrick's Direct Data Accelerator shrinks or removes bottlenecks in the flow of data all the way from storage through to the CPU, across the network, and back to the client. This requires optimizing operations at a significantly lower level of the technology stack than most vendors would dare to tread in areas of the technology stack buried so deep that they rarely see the light of day.

Re-envisioning the Operating System

Most database platforms run on general-purpose Operating Systems (OS) built to run lots of different workloads together. Yellowbrick has re-envisioned a single-purpose OS, optimized for database workload efficiency, by-passing the OS for task scheduling , device interfaces, and memory management. Co-operative multitasking both within a single node and across distributed compute nodes ensures queries get answered faster.

The result: less time wasted context switching and faster query execution using fewer resources.

By-passing Main Memory Increases Performance and Efficiency

Traditionally database platforms move data from storage to main memory and then start operating on the data based on an outdated assumption that this improves performance. This results in wasted CPU cycles cycling data in and out of this in-memory data cache which wastes valuable main memory. Memory which could be supporting critical calculations not serving database internals. By architecting for modern high-performance NVMe storage and random reads, and ignoring legacy assumptions, data gets to the CPU at memory transfer speeds.

The result: huge efficiency savings, with more memory and CPU resources available for query execution resulting in faster query execution and higher query density.
In-memory performance, without in-memory cost.

Re-imagining Device Interfaces with OS By-pass

In a traditional OS, storage and network interfaces take the CPU away from making progress on work. What's more, these device drivers are designed for general-purpose networking. Yellowbrick has slimmed down and rewritten device drivers to avoid this challenge using reactive programming techniques resulting in higher throughput without using up valuable CPU resources. We've written our own RPC service, ybRPC, and reliable datagram protocol, YRD, based on Intel's DPDK library. Yellowbrick bypasses hundreds of thousands of lines of legacy network stack code to get data directly from processes on one node to another without significant CPU utilization. We've also highly optimized reads from cloud object storage, with an order of magnitude improvements over cloud provider native libraries.

The result: faster internode communications and faster IOPs, with fewer overheads, leading to faster query execution using fewer resources.

Workload Manager

Shares Resources Fairly to Deliver Guaranteed QoS

Some companies choose to throw more resources at complex, high concurrency, or peak workload challenges. It mostly works but it is expensive. Not every query is mission-critical; some can wait a few more seconds. In most platforms, a single badly written ad hoc query can cause resource conflicts and challenges for mission-critical workloads.

Yellowbrick's advanced Workload Manager allocates resources to queries based on policies (out-of-the-box or custom), to deliver consistent, repeatable, predictable performance for queries, streaming data, and data ingest. The Workload Manager ensures critical processes are not starved of resources and can penalty box queries that are taking longer than their Resource Policy allows.

The result: meet SLAs while executing complex workloads, avoid scaling or adding clusters, and avoid excessive costs.

Query Compiler, Workload Manager, Execution Engine, Storage Engine

Turns Query Plans Directly into Efficient Machine Code

Most other MPP databases pass high-level task instructions or SQL snippets to each node. Every node needs to locally turn those instructions into code that can be run, using just-in-time compilation techniques or interpreters. In Yellowbrick, queries are immediately compiled into CPU instructions by a central, scalable compilation service, and then that machine code is passed to each node. These compiled tasks contain all the instructions needed to efficiently execute the query including network and memory tasks.

The result: worker node tasks execute instantly and synchronously without wasting time re-interpreting instructions, increasing execution efficiency and predictability.

Execution Engine

Flows Data Efficiently Through the Query Graph

The Hybrid Execution Engine (EE) ensures optimal query execution and can execute queries in both row and column-oriented fashion. Modeled after packet processing engines, like networks, queries are represented as graphs with SQL operators as nodes and packets flowing over the links between nodes. The graph extends globally across MPP nodes. The EE keeps data core-local or NUMA-node-local whenever possible and exerts flow control over the network to avoid overloading receiving nodes and ensures data stays L1 and L2 cache resident through task completion.

The result: the planet's most efficient query execution engine.

Storage Engine

Streaming and Bulk Operations Without Compromise

The highly efficient column store achieves high compression, with physical data structures organized for optimal query execution. It eschews legacy designs, providing in-built statistics and more granular indexes to optimize data scans – all optimized for SSD storage which excels at random reads. Delete maps with automatic garbage collection make bulk deletes and updates super-efficient. The row store supports streaming and single row transactions with the lowest possible latency with automatic flushing to the column store.

A hybrid row and column store enables Yellowbrick to deal with streaming data without compromise. Queries and transactions transparently maintain ACID transaction semantics across both stores with predicate pushdown further reducing data reads. Data is persisted on cloud object storage (or local storage on-premises).

The result: fresher data, fewer IOPs needed, lower storage costs, and more efficient query scans delivering faster queries.

Summary

Yellowbrick brings together technical excellence into a simple-to-consume data warehouse service in the public cloud and on-premises. Are you considering the future of an existing data warehouse or a new data analytics project? Ask prospective vendors what they are doing to maximize efficiency and minimize costs.

For a more detailed understanding, take a look at our Inside the Yellowbrick Data Warehouse whitepaper.