REFERENCE ARCHITECTURE

Scalable Lakehouse Analytics with Pure Storage and Starburst

From Hadoop sprawl to governed lakehouse:
performance, cost, and operational wins

Contents

Introduction

Organizations are increasingly adopting data lakehouse architectures to combine the flexibility of data lakes with the performance and structure of data warehouses. However, implementing data lakehouses involves significant challenges. These environments must manage immense volumes of disparate data while ensuring data quality, integrity, and acceptable query performance. They require substantial infrastructure scale to store, process, and analyze large data sets in a timely manner, necessitating both scalable infrastructure and efficient processing tools.

Traditional data lake deployments often introduce complexity, management overhead, and performance limitations. As data volumes grow, conventional approaches struggle to deliver consistent performance, especially for concurrent analytics workloads. Organizations face additional operational challenges, including maintaining system reliability, ensuring regulatory compliance, managing infrastructure costs, and implementing comprehensive security measures.

Solution Overview

Modern analytics workloads require a data platform that can deliver high performance at scale while minimizing complexity, cost, and operational overhead. This reference architecture demonstrates how the combination of the Pure Storage® and Starburst Enterprise platforms provides a powerful and efficient foundation for building a scalable data lakehouse.

In rigorous testing using the industry-standard TPC-DS benchmark at both 100GB and 1TB data set scales, this solution consistently outperformed traditional Hadoop distributed file system (HDFS)-based architectures in query throughput, infrastructure efficiency, and scalability. Pure Storage FlashBlade® delivers superior performance in both high-concurrency and sustained analytics scenarios while also reducing infrastructure sprawl and simplifying day-to-day operations.

Organizations adopting this architecture can expect:

- High-performance SQL analytics across federated data sources
- Transactional consistency and time-travel queries delivered by the Iceberg open table format, which removes Hive Metastore bottlenecks
- Significant reductions in physical footprint and power consumption
- Simplified operations through unified storage and intelligent orchestration
- A future-ready platform capable of scaling to meet growing data demands

This document provides a complete technical guide for deploying Starburst on Red Hat OpenShift with FlashBlade and Portworx®, including validated performance benchmarks, design considerations, and milestone-based deployment steps.

This reference architecture (Figure 1) combines the Starburst Enterprise platform with FlashBlade Object Storage to create a high-performance, scalable data lakehouse. The platform delivers faster time to insight while simplifying infrastructure management. Iceberg tables give the lakehouse ACID transactions; hidden partitioning; schema evolution; and efficient time-travel queries, all while remaining completely open and vendor-neutral.

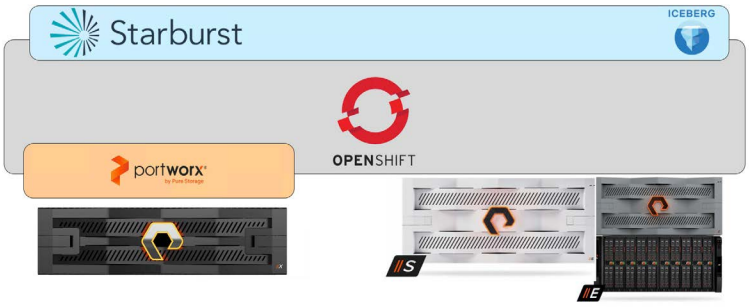

FIGURE 1 The Pure Storage and Starburst Enterprise unified lakehouse.

Solution Benefits

This solution addresses key challenges organizations face when implementing and scaling data lakehouse architectures:

- **Simplified infrastructure**: FlashBlade and Starburst together eliminate the sprawl of traditional data lake deployments that rely on distributed storage across multiple servers, significantly reducing data center footprint, power consumption, and management complexity.
- **Superior performance density**: FlashBlade delivers exponentially higher performance per rack unit than legacy approaches, dramatically reducing data center footprint, power consumption, and cooling requirements. This density advantage, combined with Starburst's efficient query processing, enables organizations to run demanding analytics workloads in a fraction of the physical space required by traditional distributed storage systems, addressing the power, space, and cooling challenges inherent in legacy data lake infrastructures.
- **Consistent, predictable performance**: FlashBlade high-performance object storage combined with Starburst intelligent query optimization ensures reliable analytics performance that scales with growing data volumes, eliminating the bottlenecks common in traditional environments.
- **Independent scaling**: The solution allows separate scaling of compute (Starburst workers) and storage (FlashBlade) based on actual requirements, optimizing infrastructure investments and enabling precise resource allocation for varying workload patterns.
- **Unified data access**: Starburst query federation capabilities combined with FlashBlade consolidated storage create a comprehensive data access layer, enabling SQL analytics across multiple sources while maintaining FlashBlade as a high-performance foundation.
- **Accelerated modernization:** This combination streamlines migration from legacy data lakes because Starburst can query existing sources while organizations transition to the FlashBlade foundation, ensuring business continuity throughout the modernization journey.
- **Operational simplicity**: The solution dramatically simplifies day-to-day operations through intuitive management interfaces, automated maintenance, and reduced infrastructure complexity. This simplicity eliminates the need for specialized expertise required by traditional distributed storage systems, freeing IT resources and accelerating time to insight.
- **Future-proof scalability**: FlashBlade exabyte-scale capacity combined with Starburst elastic worker scaling ensures seamless growth with business demands, eliminating performance ceilings and disruptive upgrades while supporting the most demanding analytical workloads.

Technology Overview

This reference architecture implements several interconnected technology components to deliver a high-performance, scalable data lakehouse. Each component's connectivity and functionality are validated to work together seamlessly, creating a platform that addresses the performance, scalability, and management challenges found in modern analytics environments.

Pure Storage FlashBlade

FlashBlade empowers organizations with a simple, adaptable, and scalable storage infrastructure that effortlessly meets the demands of modern data analytics. The innovative blade architecture prioritizes speed, efficiency, and scalability, making FlashBlade ideal for the most demanding and intensive workloads, such as rapid recovery, analytics, AI, and machine learning.

FlashBlade includes the following key capabilities:

- **Distributed, scale-out architecture**: FlashBlade employs a distributed, scale-out design that enables seamless growth from terabytes to exabytes without disruption or performance degradation.
- **High-performance object storage**: Purpose-built for analytics workloads, FlashBlade delivers consistent low-latency access regardless of workload size or complexity.
- **Unified management**: FlashBlade simplifies storage operations through a single management plane for all data services.
- **Massive parallelism**: The architecture supports highly concurrent access patterns common in data analytics workloads.

The FlashBlade product line (Figure 2) includes the following models:

- **FlashBlade//S™:** the latest evolution in enterprise scale-out storage, offering a blend of high density, capacity, performance, and scalability to meet the demands of modern applications
- **FlashBlade//E™**: a cost-efficient storage platform that offers effective performance, uncompromising reliability, and all-flash capabilities to a broader audience, delivering 40% lower total cost of ownership for an acquisition cost comparable to hard disk drives

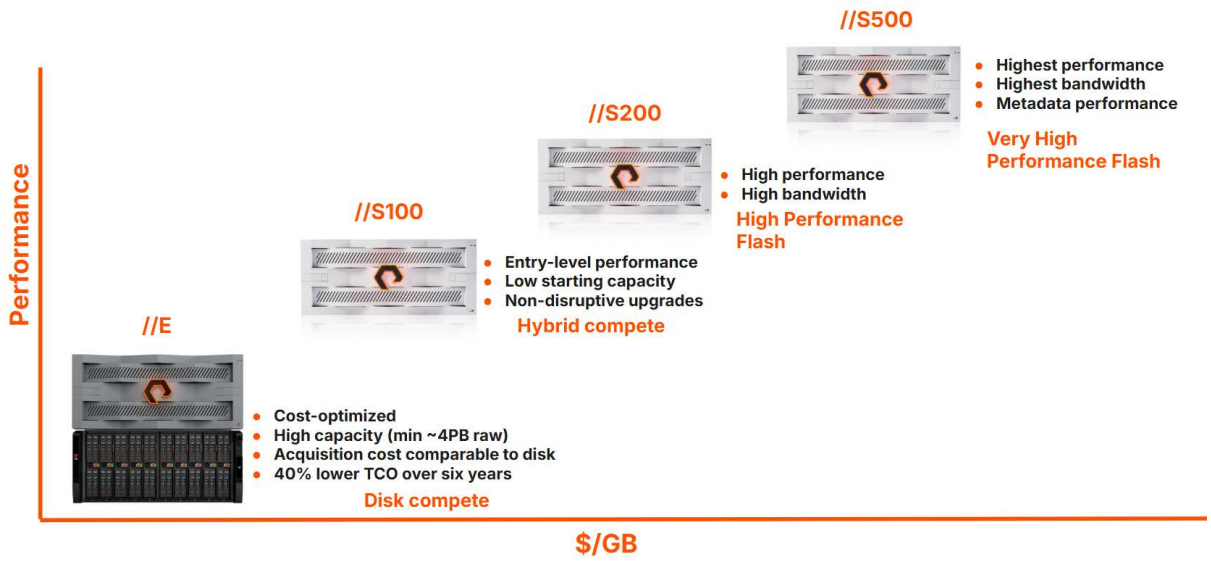

FIGURE 2 The FlashBlade product family.

Starburst Enterprise

Starburst Enterprise is an enterprise-grade distributed SQL query engine that enables organizations to analyze data across multiple sources without having to move or copy it. Based on the open source Trino project, Starburst allows users to run fast SQL queries against disparate data sources—including data lakes, data warehouses, and databases—regardless of location, be that on-premises, cloud, or cross-cloud.

The platform features a coordinator-worker architecture (Figure 3) that intelligently distributes query processing across multiple nodes for optimal performance. When paired with FlashBlade high-performance object storage, Starburst creates a powerful analytics foundation that delivers consistent performance even as data volumes and query complexity grow.

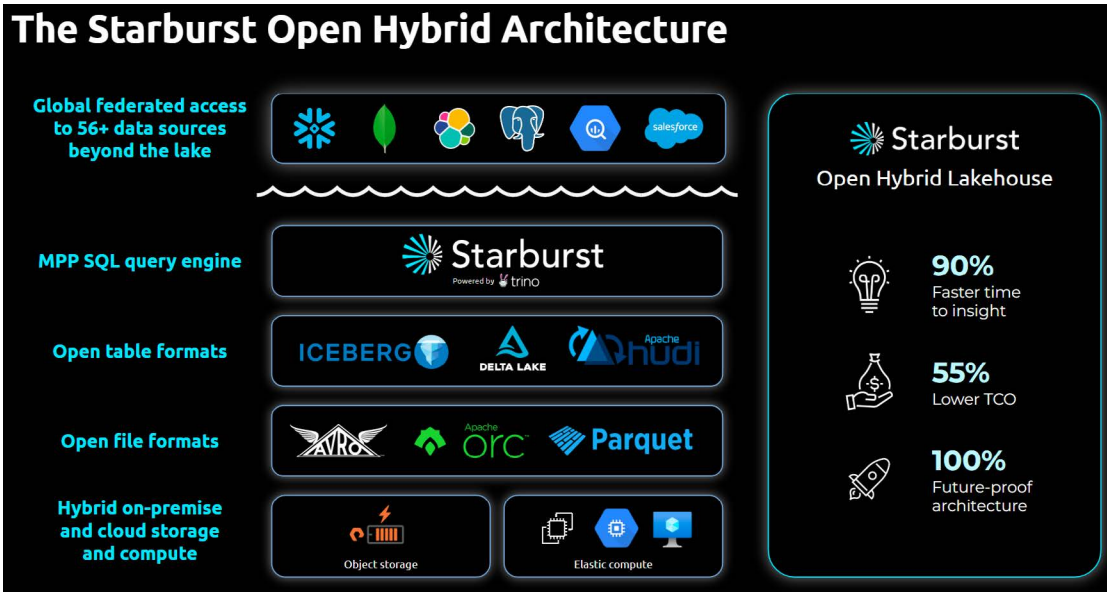

FIGURE 3 The Starburst open hybrid architecture.

Apache Iceberg Table Format

Apache Iceberg brings database-grade consistency to object storage and scales metadata to petabyte levels, offering capabilities such as:

- **Atomic commits:** Readers never see partial writes, so long-running queries remain stable.
- **Hidden partitioning:** Partition logic lives in metadata rather than directory names, avoiding costly file system scans.
- **Schema evolution:** Columns can be added, reordered, or renamed without rewriting data.
- **Snapshot-based time travel:** Analysts can reproduce results by querying any historical table state.

Starburst accesses Iceberg through its native connector. FlashBlade supplies fast Simple Storage Service (S3) storage for data files, while Iceberg catalog databases reside on Pure Storage FlashArray™ block volumes provisioned by Portworx.

Red Hat OpenShift

Red Hat OpenShift is an enterprise Kubernetes container platform that provides a foundation for deploying and managing containerized applications at scale. In this reference architecture, Red Hat OpenShift serves as the container orchestration environment for the Starburst Enterprise platform, offering production-grade infrastructure with built-in security, monitoring, and automation capabilities.

Red Hat OpenShift extends Kubernetes with developer-friendly tools and operational features that simplify deployment and lifecycle management. Its robust security model includes integrated identity management, Role-Based Access Control (RBAC), and container isolation. These capabilities create a secure, reliable foundation for running Starburst's distributed SQL query engine in production environments.

The platform's resource management and auto-scaling features enable efficient allocation of compute resources based on actual workload demands, complementing the independent scaling of FlashBlade storage resources. This flexibility allows organizations to optimize their infrastructure as analytics requirements evolve.

Portworx Enterprise

Portworx Enterprise is a cloud-native storage platform designed for Kubernetes environments (Figure 4). In this reference architecture, Portworx provides persistent storage for the stateful services within the Starburst deployment, specifically for the PostgreSQL databases that support Starburst's metadata and operational functions.

As a Kubernetes storage solution, Portworx integrates with Red Hat OpenShift to create, manage, and protect storage volumes across the container environment. It leverages Pure Storage FlashArray to deliver high-performance block storage, ensuring reliable data persistence for critical database operations.

Portworx offers enterprise features including data replication, snapshots, and backup capabilities that enhance overall solution reliability. Its storage orchestration helps maintain high availability for Starburst's stateful components, ensuring system resilience even during infrastructure changes or failures.

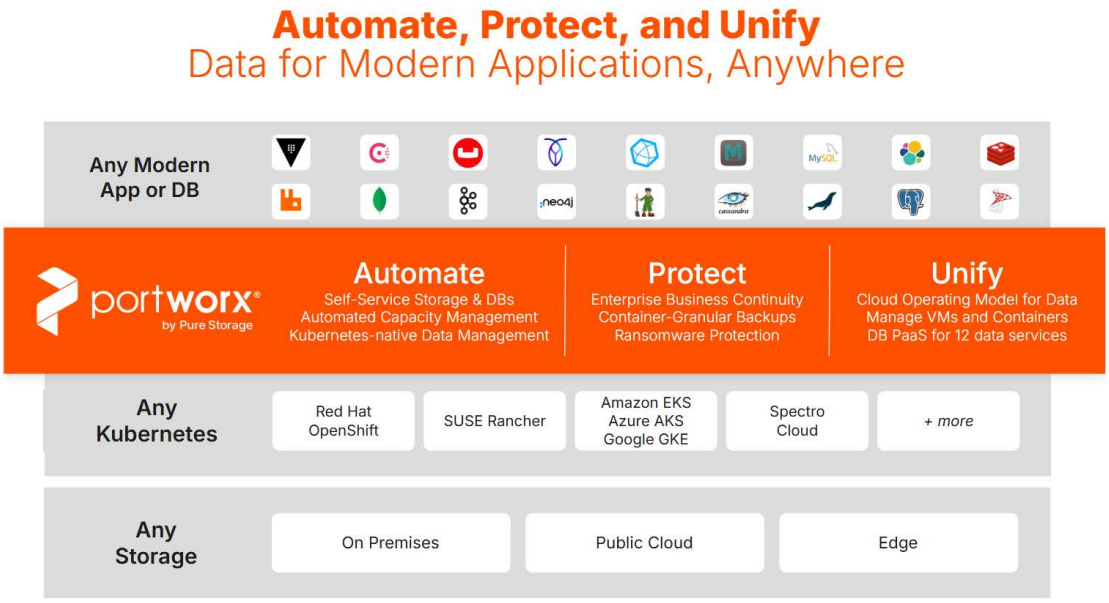

FIGURE 4 Portworx, the storage platform for Kubernetes.

PostgreSQL

PostgreSQL provides the relational database foundation for Starburst Enterprise's metadata and operational data (Hive Metastore).

In this reference architecture, PostgreSQL instances run within the Red Hat OpenShift environment with persistent storage provided by Portworx. These databases maintain critical information for the Starburst platform, including metadata about data sources, user session details, and query history. As an open source, enterprise-grade database system, PostgreSQL delivers the reliability and performance required for Starburst's core functions while integrating seamlessly with the containerized environment.

Technical Solution Design

The technical solution design in this reference architecture focuses on implementing a high-performance, cost-effective data lakehouse using Starburst Enterprise and Pure Storage FlashBlade Object Storage.

In this design (Figure 5), Starburst is deployed on a Red Hat OpenShift cluster using bare metal hosts with persistent Kubernetes block storage provided via Portworx from a back-end FlashArray. FlashBlade serves as the object storage foundation, delivering high-performance S3 storage for the data lakehouse layer. All components communicate over an Ethernet network optimized for both performance and availability.

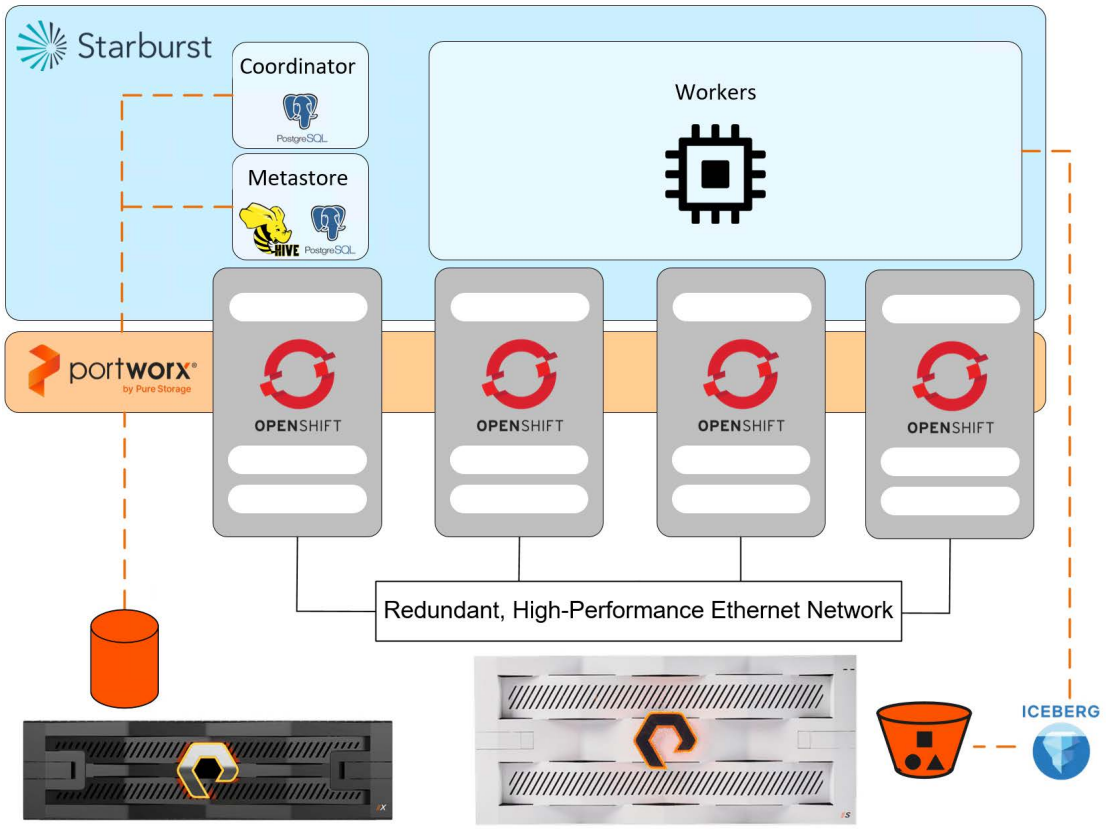

FIGURE 5 Technology component interconnects and architecture.

Infrastructure Layer

Compute

This reference architecture utilizes Red Hat OpenShift as the container orchestration platform managing the compute resources for the Starburst Enterprise platform. Red Hat OpenShift provides automated scheduling, deployment, and lifecycle management of the containerized analytics environment.

The underlying infrastructure can leverage either x86-64 or ARM-based server architectures, providing flexibility in hardware selection based on performance requirements, power efficiency goals, or existing infrastructure standards.

While the reference implementation showcases a robust nine-node deployment, the minimum viable configuration for a production-ready environment would include:

- 3 Red Hat OpenShift nodes (to maintain cluster quorum)
- 16 cores per node
- 128GB RAM per node
- Network cards with at least 10GbE ports

The recommended starting configuration (for moderate workloads) would include:

- 4 to 6 Red Hat OpenShift nodes
- 32 cores per node
- 256GB RAM per node
- Network cards with at least 25GbE ports

This flexible approach allows organizations to start with a smaller footprint and scale as analytics demands grow. Red Hat OpenShift's intelligent scheduler ensures optimal placement and resource allocation for Starburst's coordinator and worker containers across the available infrastructure.

Network

The network considerations for this reference architecture align with enterprise IT infrastructure best practices, focusing on availability, performance, and extensibility. The Red Hat OpenShift nodes connect to a compatible TCP/IPv4 or TCP/IPv6 network infrastructure with a recommended minimum network speed of 10GbE per node.

For optimal performance and reliability, this technical solution design recommends implementing the following network design principles for each Red Hat OpenShift node:

- Dual switches configured with interconnects to eliminate single points of failure
- Network port bonding on nodes using either Active Load Balancing (Mode 6) for highest performance or 802.3ad/LACP (Mode 4) for highest performance and availability
- Direct connectivity between Red Hat OpenShift nodes and storage through the same switches, eliminating network hops that could introduce latency

FlashBlade networking leverages link aggregation groups, subnets, and virtual network interfaces (VIFs) to optimize performance. To avoid bottlenecks, the FlashBlade link aggregation group should provide aggregate bandwidth at least equal to the total compute node connectivity.

Data Storage Layer

The storage infrastructure layer of this reference architecture consists of two complementary systems that provide the foundation for the data lakehouse: Pure Storage FlashBlade for object storage and Pure Storage FlashArray with Portworx for container-native block storage.

FlashBlade Object Storage

Pure Storage FlashBlade serves as the primary data repository for the data lakehouse, providing S3-compatible object storage optimized for analytics workloads.

Key design considerations for FlashBlade in this architecture include:

- S3 buckets configured with appropriate access controls to store the data lakehouse data sets
- A dedicated bucket provisioned for Starburst fault-tolerant execution spool to enable resilient query execution
- Consistent bucket naming conventions across the environment to avoid conflicts

Portworx with FlashArray

Portworx Enterprise provides persistent block storage for stateful services running within the Red Hat OpenShift environment, particularly the PostgreSQL databases that support Starburst's operation. As a container-native storage solution, Portworx integrates with Kubernetes to provide dynamic provisioning and data protection for these critical components.

In this architecture, Portworx leverages Pure Storage FlashArray as its back-end storage system, connecting through iSCSI to provide high-performance persistent volumes for:

- PostgreSQL databases containing Starburst Insights data
- The Hive Metastore database maintaining schema and table metadata
- Other stateful services required by the platform

Query Engine Layer

Starburst Enterprise functions as the distributed SQL query engine in this architecture, enabling high-performance analytics across data stored in FlashBlade Object Storage. Deployed as containerized workloads on the Red Hat OpenShift cluster, Starburst, in addition to the data access function, is the secured and governed data management layer, which is critical for business intelligence, ML, and AI solutions and AI agent workflows.

The Starburst deployment consists of several key components:

- **Coordinator node**: a single server that handles incoming queries; provides query parsing, analysis, scheduling, and planning; and distributes processing to worker nodes
- **Worker nodes**: multiple servers that execute tasks as directed by the coordinator, retrieving and processing data in parallel
- **Hive Metastore:** maintains metadata about tables, schemas, and data organization stored in FlashBlade Object Storage

The deployment scales by adding or removing worker nodes. Starburst workers read and write Iceberg tables stored on FlashBlade Object Storage through the Iceberg connector, which bypasses Hive directory listings and delivers consistent performance as table size grows.

Metadata Management Layer

PostgreSQL databases provide the metadata management foundation for the Starburst Enterprise platform. These databases are deployed within the Red Hat OpenShift environment with persistent storage provided by Portworx on FlashArray. The metadata layer consists of two critical components:

- **Insights database**: stores back-end information for the Starburst user interface (UI), maintains user session data, houses RBAC system tables, stores privilege mappings, and records comprehensive query history and audit logs
- **Hive Metastore database**: functions as a centralized repository for metadata, including catalog configurations, schema definitions, table properties, storage locations, and partitioning information

This metadata layer bridges the unstructured nature of object storage and the relational query engine. Iceberg catalogs are stored in PostgreSQL or Hive Metastore databases that sit on FlashArray volumes managed by Portworx, giving low-latency commits and reliable recovery.

Data Access Layer

The data access layer provides interfaces for application and user interaction with the data lakehouse. Starburst Enterprise offers multiple access methods:

- **JDBC/ODBC connectors**: enable integration with business intelligence tools, custom applications, and other SQL-based systems
- **Web-based UI:** provides an intuitive interface for ad hoc querying, query management, and system administration
- **REST API:** supports programmatic access for automation and integration with external systems

These interfaces create a unified access point for analytics across the entire data estate, with security controls enforced consistently regardless of the access method.

Design Validation

This reference architecture was evaluated by establishing the performance and scalability benefits of deploying Starburst Enterprise with Pure Storage FlashBlade as the foundation for data lakehouse analytics, with particular focus on establishing the storage performance baseline compared to aggregated storage (HDFS).

Workloads and Methodology

The primary objective of this validation was to establish a clear storage performance baseline by comparing Pure Storage FlashBlade against a traditional HDFS cluster when running Starburst Enterprise on Iceberg tables (see Table 1 for component details). All TPC-DS data sets were generated as Iceberg tables; tests compared partitioned and non-partitioned layouts to measure the benefit of Iceberg hidden partitioning.

Testing focused on two key aspects:

- The performance difference between partitioned and non-partitioned Iceberg data sets at SF1000 scale on FlashBlade
- The scalability comparison between FlashBlade and HDFS with increasing concurrency and worker nodes at SF100 scale

The TPC-DS benchmark was selected because it represents complex analytical queries that exercise multiple aspects of the data platform and is widely accepted as an industry standard for evaluating analytical database performance.

Component	Detail
Red Hat OpenShift compute nodes	**Specification:** 9 mid-range performance servers with dual Intel Xeon Gold 6342 CPUs running at 2.80GHz **Memory:** 756GiB of RAM per node **Network:** dual-port ConnectX-6 100GB network adapters
Storage array	**FlashBlade//S** **Configuration:** FlashBlade//S200 with 10 blades **Capacity:** (2) 24TB (285TiB total) direct flash modules per blade **FlashBlade//E** **Configuration:** 2 chassis **Capacity:** (40) 48TB (4PB total) direct flash modules **Software:** Purity//FB 4.5.6 **Network connectivity:** 8 network interfaces in link aggregation group providing 800Gbps total bandwidth
HDFS cluster	**Server configuration:** 8 mid-range performance servers as HDFS nodes with 6 SSDs per node (98TB usable) **HDFS version:** 3.3.6 **Support services:** 3 additional mid-range performance servers hosting NameNode, Hive Metastore, and ZooKeeper services
Starburst Enterprise platform	**Version:** 472.0.0 **SF1000 test setup:** **Coordinator:** 60vCPUs, 240GB RAM **Workers:** 8 workers with 90vCPUs each **SF100 test setup:** **Coordinator:** 31vCPUs, 240GB RAM **Workers:** 4 to 16 workers with 31vCPUs and 240GB RAM each **PostgreSQL back end:** database for Insights and Hive Metastore **S3 connector:** configured for FlashBlade Object Storage
Networking	**Switches:** (2) 100GbE network switches configured as multi-chassis link aggregation for high availability
TPC-DS data sets	**SF1000 (1TB) data set:** generated in both partitioned and non-partitioned formats; used to evaluate the performance impact of partitioning on query execution time **SF100 (100GB) data set:** used for concurrency and scalability testing; used to evaluate system behavior under increasing load
Test parameters	**SF100 comparative testing (FlashBlade vs. HDFS)** Varied worker count: 4, 8, and 16 workers Varied concurrency: 8 to 128 concurrent sessions 1 loop per configuration Metrics: time to complete, error rates under increasing load **SF1000 comparative testing (FlashBlade vs. HDFS)** 8 workers 8 concurrent sessions 1 loop per configuration Metrics: time to complete, error rates under increasing load **SF1000 testing on FlashBlade** 5 loops with a concurrency of 5 for both partitioned and non-partitioned data sets 25 total iterations per configuration Metric: total elapsed time for all 99 TPC-DS queries

TABLE 1 Testing component details.

Storage Baseline Comparison: FlashBlade vs. HDFS

To evaluate the impact of storage architecture on Starburst query performance, we conducted a series of baseline tests comparing FlashBlade and HDFS using the industry-standard TPC-DS benchmark. Two data set scales were selected to represent different workload profiles:

- **SF100** (100GB): designed to simulate high-concurrency query execution with significant metadata activity and mixed I/O patterns
- **SF1000** (1TB): intended to stress sustained analytical throughput with large scans, aggregations, and heavier sequential I/O demands

The objective of this testing was to isolate and measure the effect of the underlying storage system on Starburst query performance across two distinct workload types. Results are normalized and presented as query throughput (queries per second), allowing for a direct comparison of efficiency and scalability between FlashBlade (//S and //E models) and a traditional HDFS deployment.

SF100 Performance Comparison

The primary focus of this validation was to measure query throughput at a given Starburst cluster scale and compare the performance of FlashBlade storage with a traditional HDFS cluster. The TPC-DS-SF100 benchmark was used to evaluate how each storage platform (FlashBlade//E, FlashBlade//S, and HDFS) influenced throughput. The test environment was scaled to 16 worker nodes and 128 concurrent sessions to simulate high-concurrency workloads and determine the number of queries completed over time.

This configuration places sustained pressure on several aspects of the storage subsystem:

- I/O bandwidth and IOPS, as multiple queries access storage simultaneously
- Metadata handling, driven by frequent small reads and seeks
- Cache efficiency, based on how well the system manages concurrent access patterns
- Lock contention across multiple threads
- Network saturation under sustained load

As shown in Figure 6, both FlashBlade//S and FlashBlade//E achieved higher normalized query throughput than HDFS. This demonstrates the advantage of an all-flash, scale-out architecture in delivering greater efficiency and responsiveness for concurrent analytical workloads.

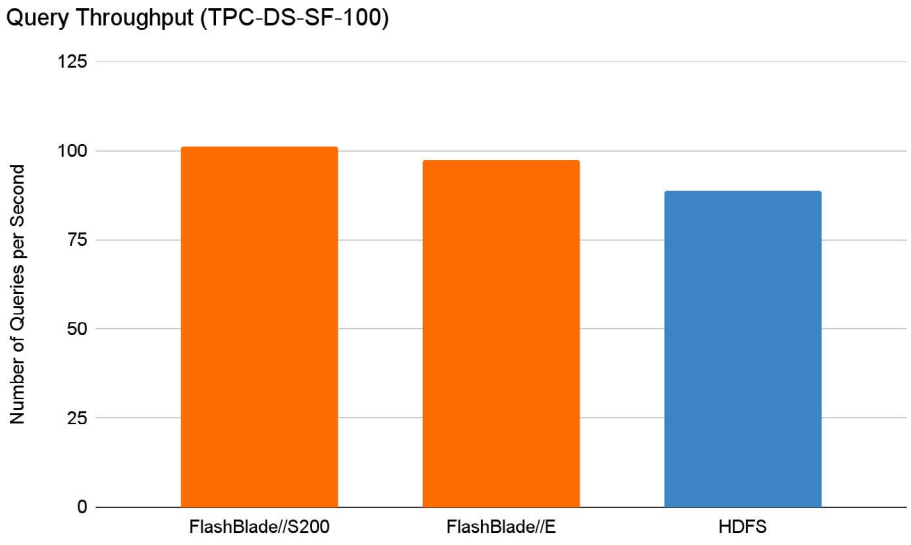

FIGURE 6 Query throughput comparison with TPC-DS-SF100.

SF1000 Performance Comparison

The primary focus of this validation was to assess sustained analytical workload performance with minimal influence from compute or cache. The TPC-DS-SF1000 benchmark was executed at a concurrency level of 8 with a single execution loop across FlashBlade//S200, FlashBlade//E, and a traditional HDFS storage environment. This workload reflects scenarios where query patterns are dominated by large scans and aggregations rather than high concurrency.

This configuration stresses several key aspects of storage performance:

- Sequential throughput capacity, driven by large-scale data scans and aggregations
- Latency under load, with fewer but larger I/O operations
- Sustained bandwidth, maintained over longer query execution times

This workload is focused on testing the depth of the I/O pipeline, highlighting how well each storage system handles prolonged, bandwidth-intensive access patterns. As shown in Figure 7, FlashBlade//S delivered the highest sustained throughput, outperforming both FlashBlade//E and HDFS. These results illustrate the architectural advantage of all-flash storage for large-scale, scan-heavy analytics typical in data lakehouse environments.

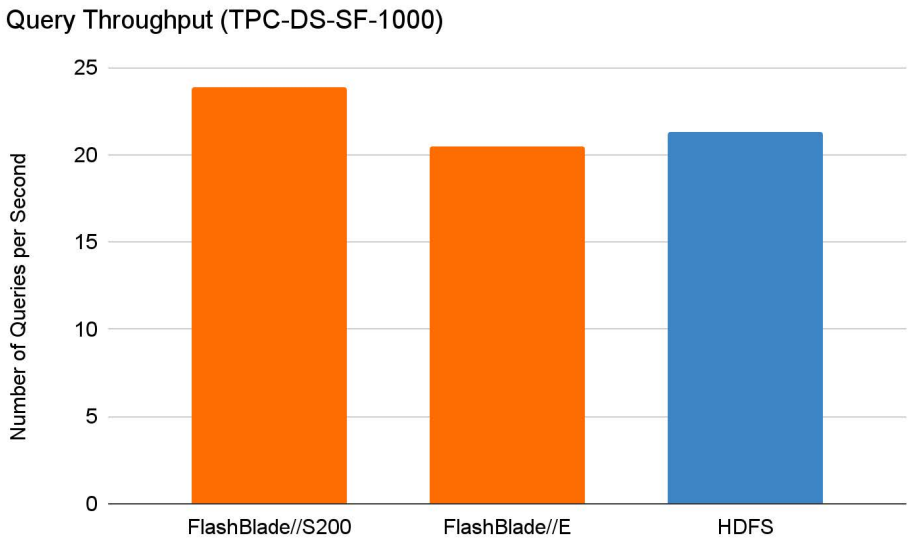

FIGURE 7 Query throughput comparison with TPC-DS-SF1000.

Storage Efficiency Analysis

Beyond raw query performance, the evaluation also considered storage efficiency, infrastructure footprint, and operational complexity. As shown in Figure 8, the FlashBlade platforms delivered significantly higher usable capacity per rack unit compared to the HDFS deployment used in testing:

- FlashBlade//E provided 54 times the usable space per rack unit compared to HDFS.
- FlashBlade//S provided 14 times the usable space per rack unit compared to HDFS.

These improvements in storage density support data center consolidation by reducing the physical footprint required to support large-scale analytics workloads.

Infrastructure consolidation: The HDFS cluster required 11 physical servers, including eight data nodes and three service nodes. Each consumed rack space, power, and cooling. In contrast, FlashBlade delivered a fully integrated storage system with a much smaller footprint. This reduction in infrastructure translates directly into lower energy consumption, cooling needs, and space utilization.

Operational overhead: The HDFS environment involved managing multiple distributed services across separate nodes. FlashBlade offered a unified platform that simplified administration and reduced the complexity typically associated with managing large-scale storage systems.

Scalability characteristics: Both storage platforms showed linear performance scaling as worker nodes were added. However, FlashBlade maintained more consistent performance at high concurrency, particularly when running the most complex queries in the benchmark.

Resource utilization: Throughout testing, FlashBlade was never the limiting factor for performance. Even under the highest concurrency levels, storage remained responsive, and the primary constraint shifted to compute capacity. This indicates that FlashBlade can support further scaling without requiring additional storage infrastructure.

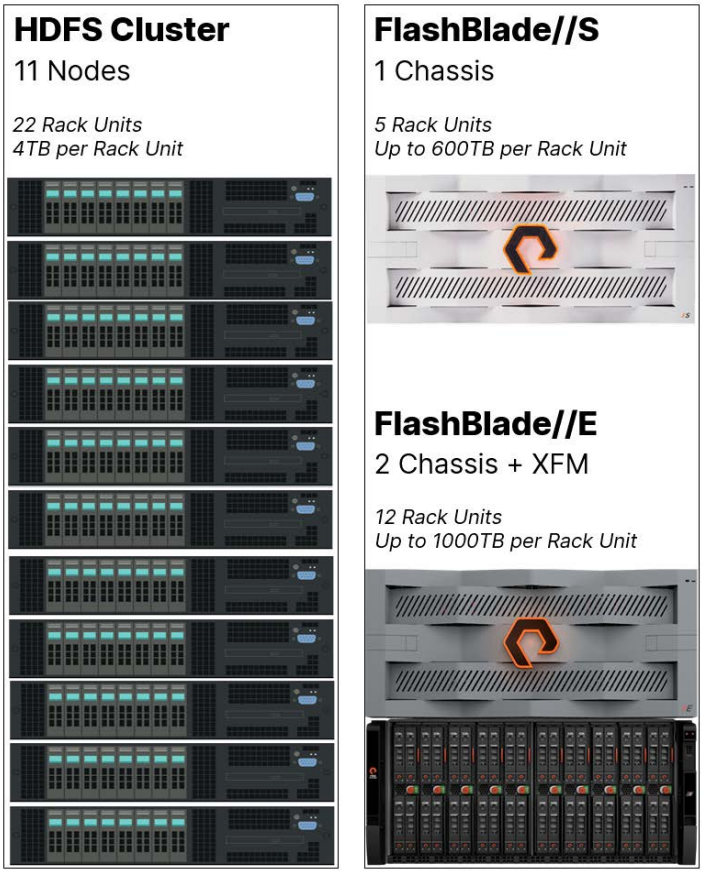

FIGURE 8 Storage and rack efficiency.

Deployment Guidance

This section provides milestone-based deployment guidance for implementing Starburst Enterprise with Pure Storage FlashBlade. Each milestone includes key objectives, references, and detailed documentation for step-by-step execution.

This deployment guide assumes the following components are already set up and in this state:

- Red Hat OpenShift: operational cluster with administrator access
- FlashBlade: deployed and network-accessible from Red Hat OpenShift nodes
- FlashArray: deployed and configured for Portworx integration
- Portworx: installed and configured on the Red Hat OpenShift cluster

Milestone 1: Provision FlashBlade Object Storage

Objective: Set up FlashBlade as the foundation for the data lakehouse.

- Configure network settings including link aggregation groups, subnets, and virtual interfaces.
- Create S3 buckets and user accounts with appropriate access credentials.
- Validate S3 connectivity and performance from client systems.

Milestone 2: Prepare Red Hat OpenShift Environment

Objective: Deploy and configure Red Hat OpenShift as the container platform for Starburst.

- Implement network security policies for secure communication.
- Deploy and configure Portworx with FlashArray for stateful services.
- Use the Portworx pre-created storage classes or set up new storage classes for persistent volume claims.

Milestone 3: Deploy PostgreSQL for Starburst Metadata

Objective: Set up the database foundation for Starburst metadata services.

- Deploy PostgreSQL database service with high availability configuration or PostgreSQL directly on FlashArray.
- Create required databases for Starburst Insights and Hive Metastore(s).
- Configure database backup and recovery procedures.

Milestone 4: Deploy Starburst Enterprise

Objective: Implement Starburst as the query engine for the data lakehouse (refer to official Starburst documentation).

- Configure Helm repositories and authentication for Starburst.
- Deploy Starburst coordinator and worker nodes with Helm charts.
- Connect Starburst to PostgreSQL databases.
- Set iceberg.catalog properties so Starburst points to the FlashBlade S3 endpoint.
- Create an Iceberg catalog in Hive Metastore or REST mode.
- Deploy Hive Metastore for table metadata.
- Configure Starburst fault-tolerant execution using FlashBlade S3.
- Implement data partitioning strategies for optimal query performance.

Refer to the Pure Storage Knowledge Base: Starburst implementation guide for details on a validated environment.

Conclusion

This reference architecture validates that Pure Storage FlashBlade combined with Starburst Enterprise is a highly effective solution for modern data lakehouse analytics and AI workflows. Across both high-concurrency and large-scale query workloads, the solution delivered consistently higher performance and resource efficiency than traditional HDFS-based environments.

FlashBlade provided exceptional throughput and operational simplicity, eliminating the infrastructure sprawl and tuning overhead associated with distributed storage clusters. When paired with Starburst's distributed SQL engine and Red Hat OpenShift's container orchestration, the result is a highly scalable, production-ready platform that simplifies deployment and management of large-scale analytics workloads.

By consolidating storage, improving performance per rack unit, and enabling elastic compute scaling, organizations can accelerate insights, reduce costs, and future-proof their analytics infrastructure.

To learn more or begin deploying this solution, visit purestorage.com/partners/starburst.

CM1852-01-en 06/25