IDUG

IDUG
2024 NA Db2 Tech Conference

Next Generation Db2 Warehouse: A New Architecture for the Cloud and Lakehouse Era

David Kalmuk, Distinguished Engineer

Venkatesh Gopal, Senior Technical Staff member

IBM

Session Code: CLOUD3| Platform: Db2, Warehouse and Cloud

Agenda

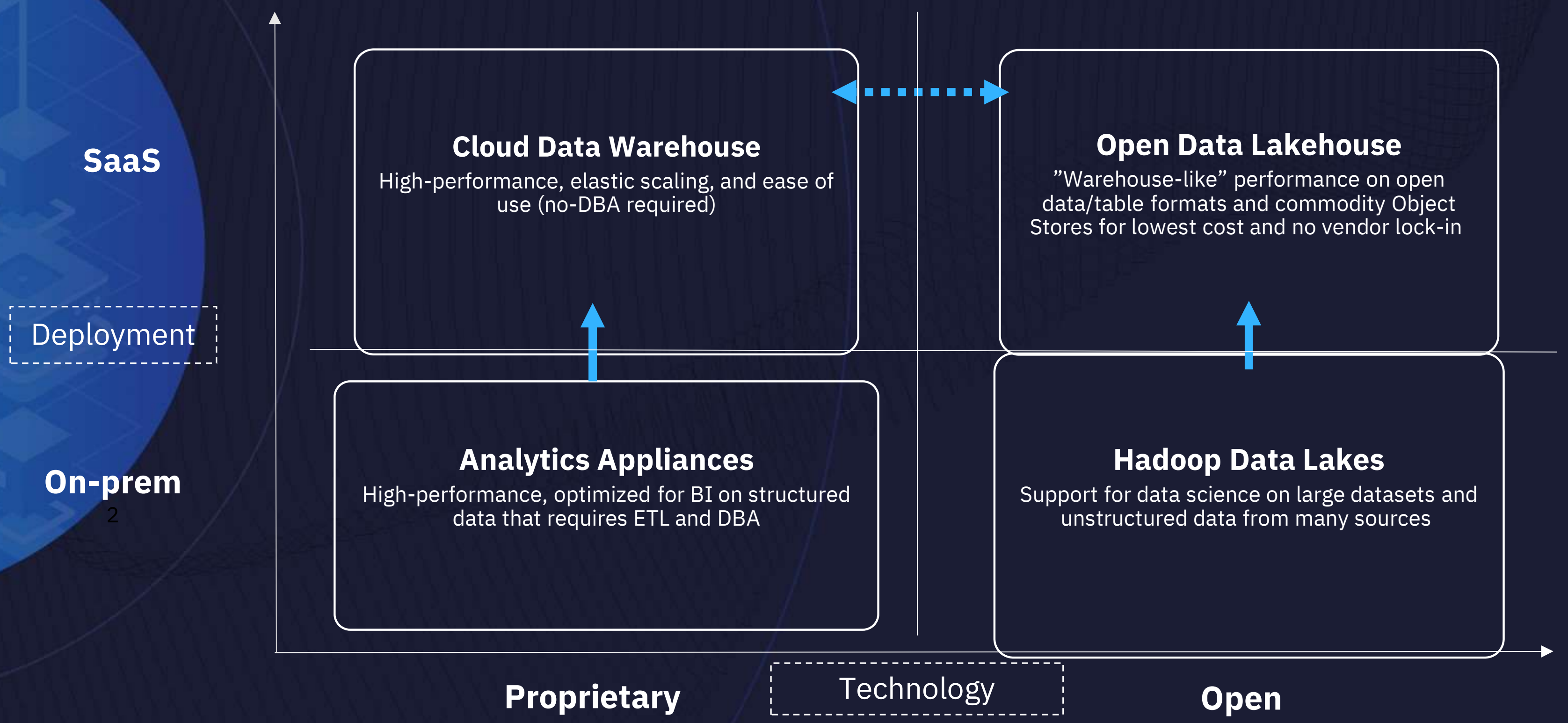
- Overview
- Next Gen Warehouse Architecture
- Cloud SaaS

Overview

Market Dynamics

Major disruptions are driving the growth in the analytics repositories market **from on-prem to SaaS** and **blending proprietary and open technologies**

Analytics Repositories Market Landscape



SaaS

Deployment

On-prem

2

Cloud Data Warehouse
High-performance, elastic scaling, and ease of use (no-DBA required)

Open Data Lakehouse
"Warehouse-like" performance on open data/table formats and commodity Object Stores for lowest cost and no vendor lock-in

Analytics Appliances
High-performance, optimized for BI on structured data that requires ETL and DBA

Hadoop Data Lakes
Support for data science on large datasets and unstructured data from many sources

Proprietary

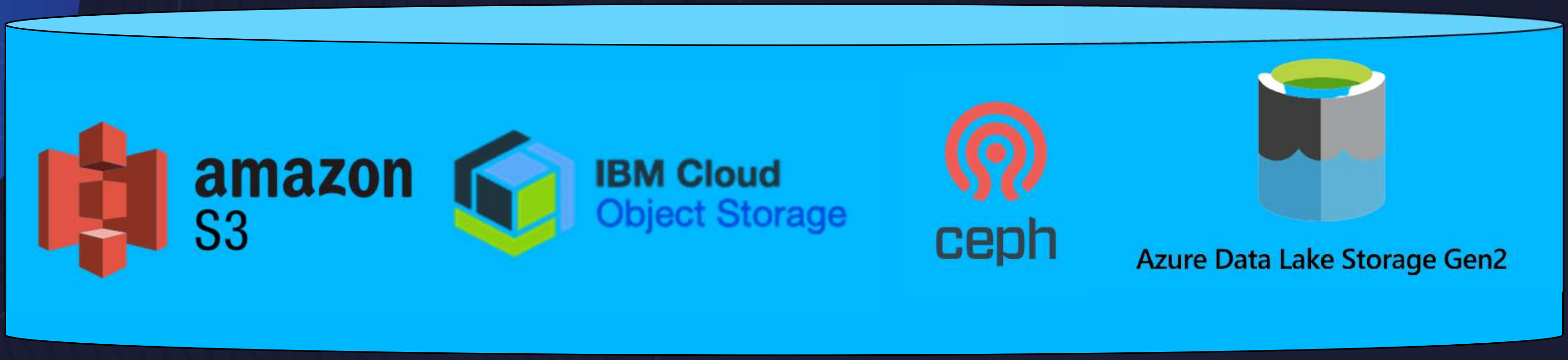
Technology

Open

Sources: IDC Data Management Forecast (November 2021), IDC BDA Forecast (June 2021), MI modeling

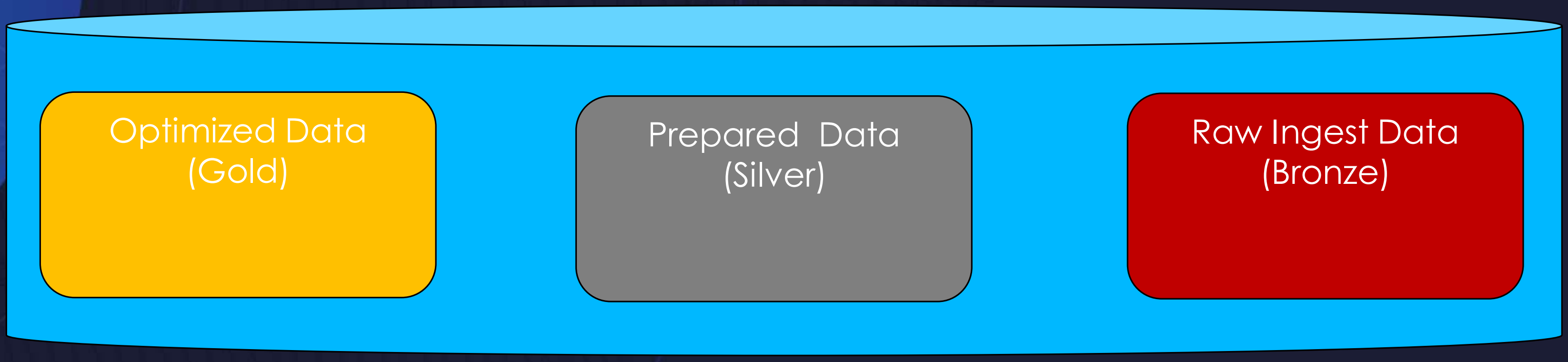
Common Data Lake Storage

- ✓ Low cost
- ✓ Near unlimited scalability
- ✓ Extreme durability + reliability (99.999999999%)
- ✓ High throughput
- 🛑 High latency (but can be compensated for)



Data Lake - Cloud Object Storage

Purpose Optimized Data Formats



Data Lake - Cloud Object Storage

Purpose Optimized Data Formats

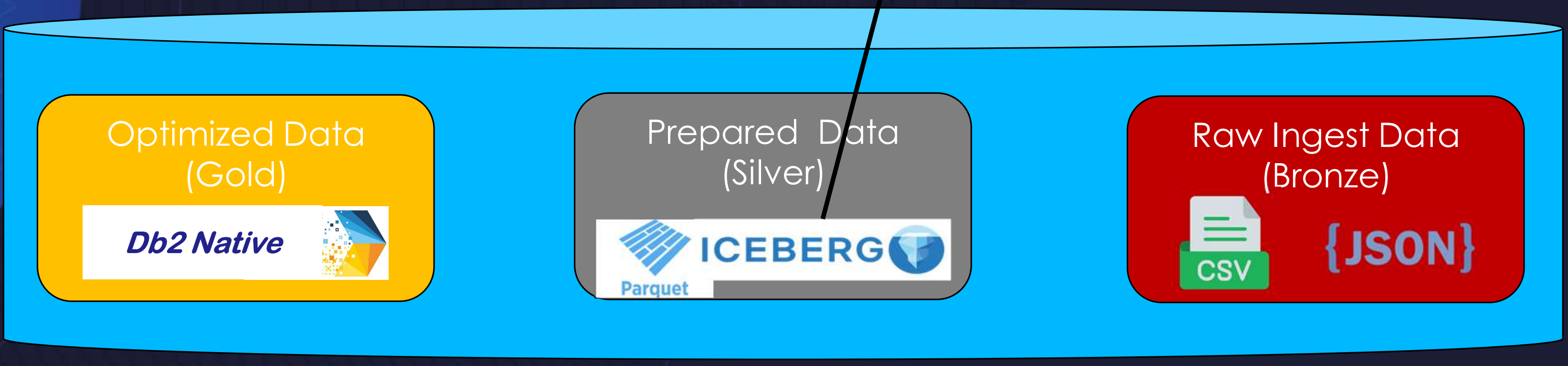
Proprietary / Optimized Format
Well Defined Schema
Highest Performance
High Volume Transactions

Open / Int
Flexible Sc
Medium P
Low Volume Transactions

ormat
ormance
sformation
Not transactional

Iceberg Open Table Format

- *Additional metadata on top of open file formats*
- *Provides ACID consistency for data lake tables*
- *Transforms Lake -> Lake**house***



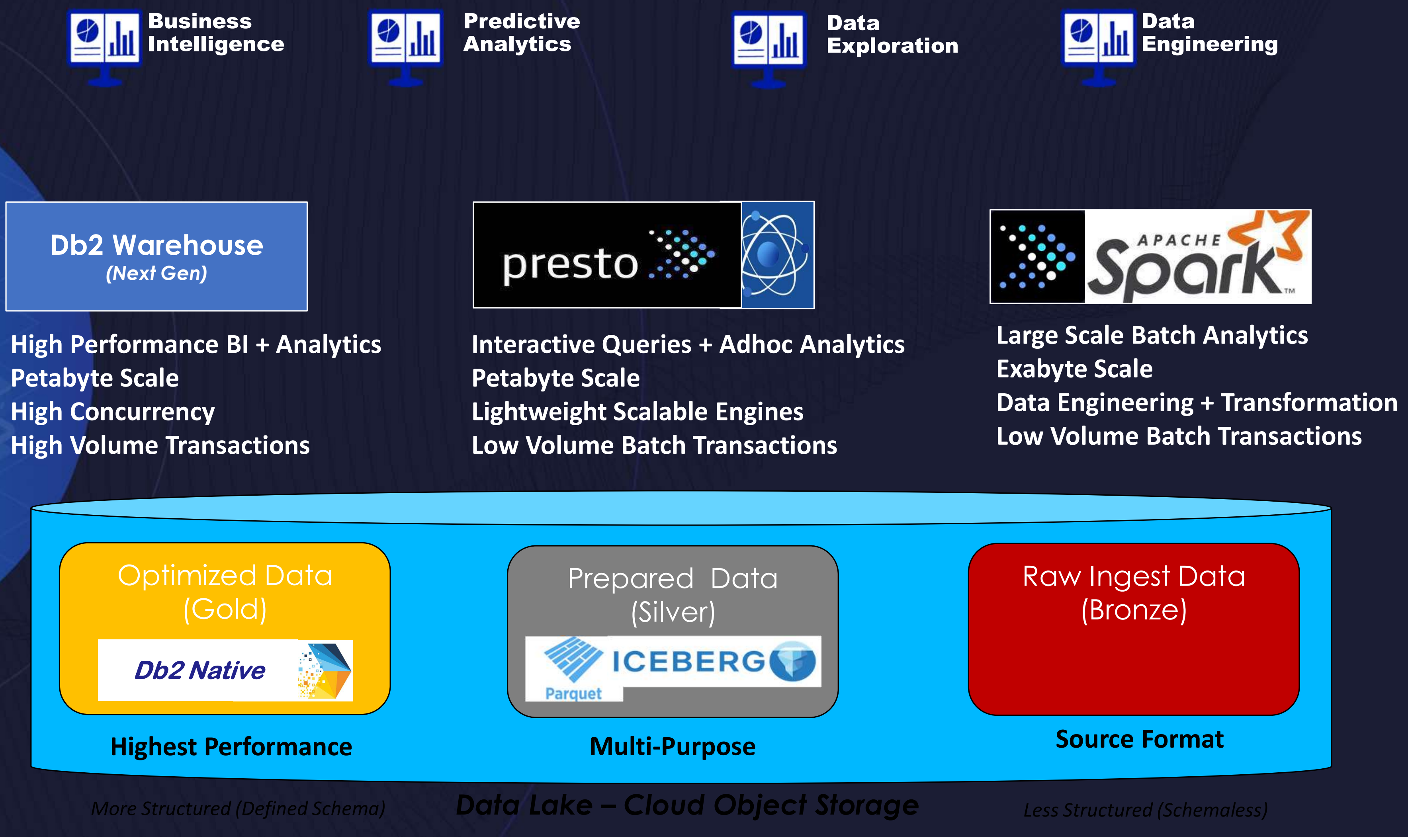
Optimized Data (Gold)

Prepared Data (Silver)

Raw Ingest Data (Bronze)

Data Lake - Cloud Object Storage

Purpose Optimized Engines



Business Intelligence

Predictive Analytics

Data Exploration

Data Engineering

Db2 Warehouse
(Next Gen)

High Performance BI + Analytics
Petabyte Scale
High Concurrency
High Volume Transactions

presto

Interactive Queries + Adhoc Analytics
Petabyte Scale
Lightweight Scalable Engines
Low Volume Batch Transactions

APACHE Spark

Large Scale Batch Analytics
Exabyte Scale
Data Engineering + Transformation
Low Volume Batch Transactions

Optimized Data
(Gold)

Db2 Native

Highest Performance

Prepared Data
(Silver)

ICEBERG
Parquet

Multi-Purpose

Raw Ingest Data
(Bronze)

Source Format

More Structured (Defined Schema)

Data Lake – Cloud Object Storage

Less Structured (Schemaless)

Next Gen Db2 Warehouse on Cloud Object Storage

Full warehousing SQL + performance with tables in cloud object storage

Lowers storage costs and simplifies storage tiering with local NVME caching

Data lake integration with open formats through external tables

Warehouse data stored directly in the “data lake”

Superior query performance with optimized native data format

High performance bulk + streaming IUD with full transactional support

Next Gen Warehouse Architecture

The Advantage of Db2 Warehousing

Performance + Scalability

MPP performance + scalability match the best in the industry

Deep Analytics + Operational Analytics

Column store + row store with indexes provide wider workload support than competitors

Concurrency + Workload Management

Seamlessly handle hundreds to thousands of concurrent jobs

High Speed <u>Transactional</u> Bulk + Streaming Ingest

High speed insert / update / delete for both bulk + trickle feed

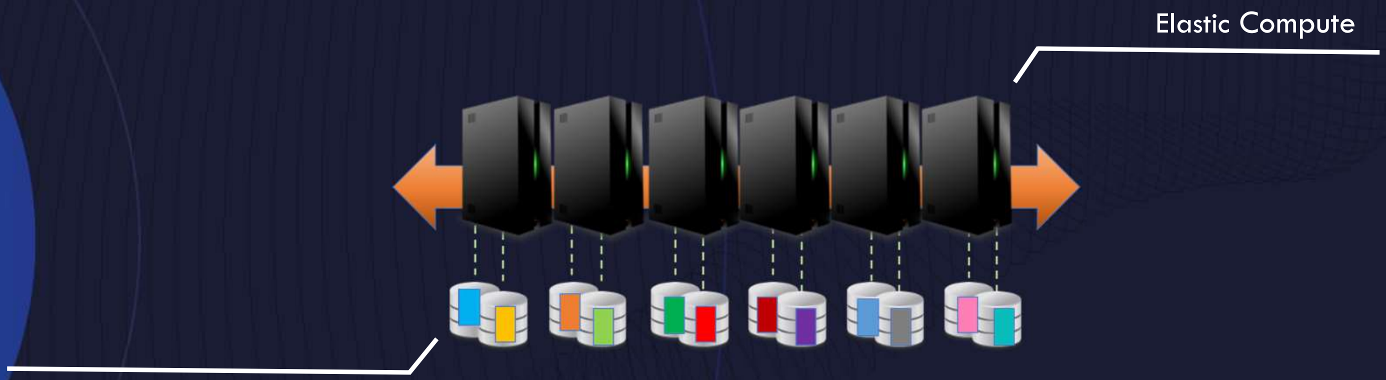
Different Kinds of Workloads – Where Db2 Plays

Db2 Warehousing
(Db2 DPF, Db2 Warehouse, Db2 Warehouse on Cloud, IIAS)

Competitor
Cloud Data Warehouse +
Lakehouse

	OLTP	Mixed Workloads		OLAP / Warehousing			
Workload	**OLTP**	**Operational**	**Extreme Analytics**	**Operational Data Store**	**Operational Analytics**	**Analytics**	**Data Marts**
Use	Transactions	Transactions with operational analytics	All types of analytic workloads and federated sources	Simple queries	Single-record look-up / IUD	Deep analytics – bulk scan	Computationally heavy / mining
Data Types	Traditional structured	Traditional structured + events + JSON + Graph + …	Traditional structured + events + JSON + Graph + …	Traditional structured	Traditional structured	Traditional structured	Traditional structured
Performance	1000s TPS	1000s TPS + 10s QPS	1000 QPS, many complex	100s QPS	10s-100s of S/IUD per second	100s QPS – 1000 QPS	Lower volume long running
Inserts	1000/s	1000/s	Continual Data Ingest	snapshots Seconds delay	Continual Data Ingest (Trickle feed) – OR - Batch - hourly/n-times daily/daily/etc		
Schema	Normalized	Normalized	De-normalized	Normalized	De-normalized	De-normalized	De-normalized
Applications	SQL	SQL + NoSQL + NewSQL	SQL + NoSQL + NewSQL	SQL	SQL	SQL	SQL

Db2 Warehouse Architecture

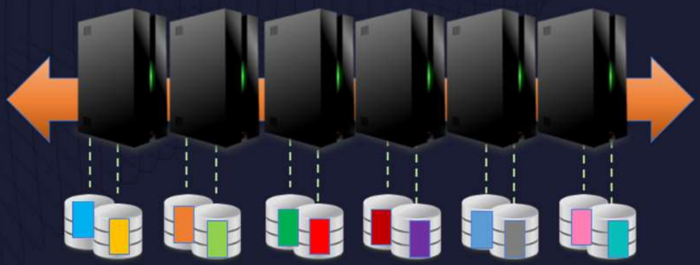


Elastic Compute

Fast Cloud Block Storage

Db2 Warehouse Architecture

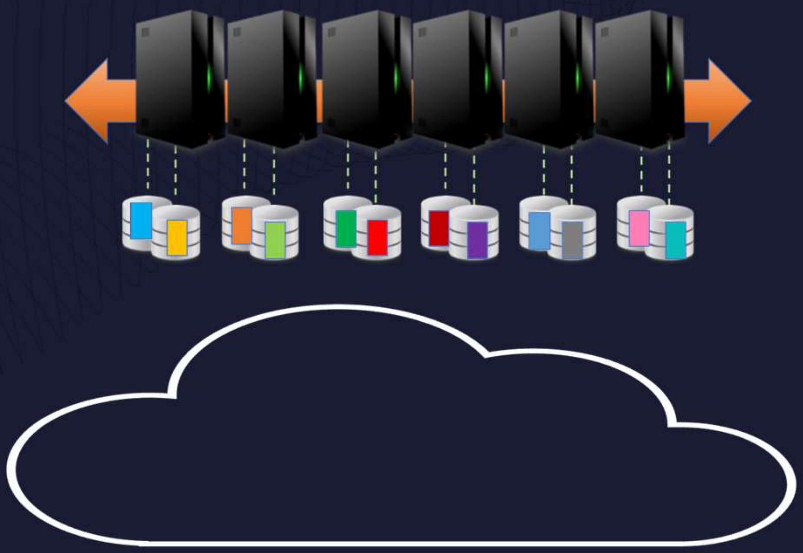
- ✓ *High Performance*
- ✓ *Elastic*
- 🚫 *Monolithic Storage*
- 🚫 *Expensive*



Db2 Warehouse Architecture

Next Generation

- ✓ ***High Performance***
- ✓ ***Elastic***
- 🚫 ***Monolithic Storage***
- 🚫 ***Expensive***



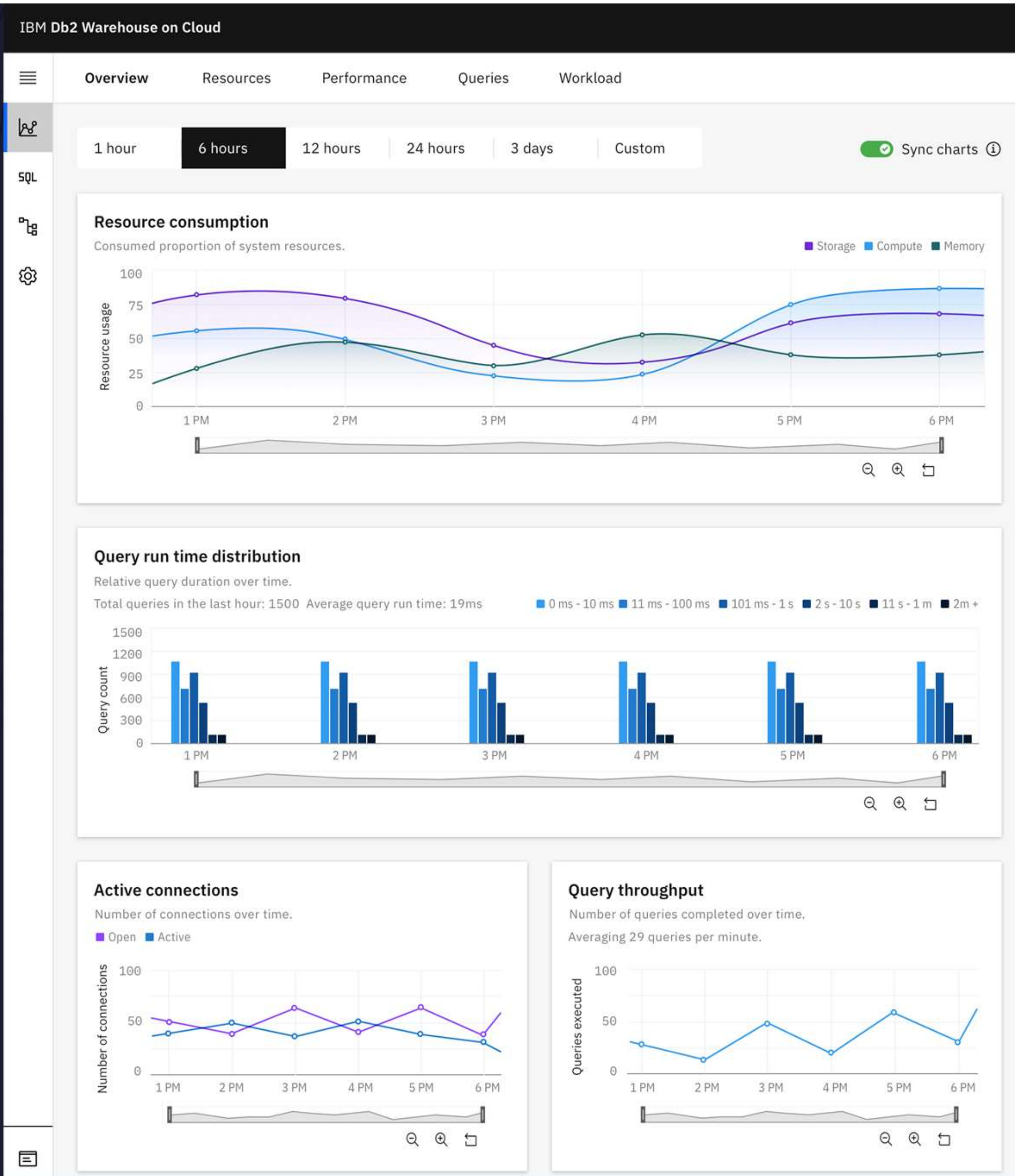
Db2 Warehouse Architecture

Next Generation

- ✓ ***High Performance***
- ✓ ***Elastic***
- 🚫 ✓ ***Monolithic Storage*** ***Decoupled Storage***
- 🚫 ✓ ***Expensive*** ***Cost Efficient***

Introducing the third generation **Db2 Warehouse**, with up to 4x better performance and 34x cheaper storage

- **Key new features:**
  - **Amazon S3 support** for table storage, significantly decreasing the cost of storing data without sacrificing performance
  - **Query, import & export multiple open table & data formats** (Iceberg, Parquet, ORC, CSV and more) leveraging existing compute resources dedicated to the warehouse
  - **Integration with watsonx.data** with sharing of data catalogs and Amazon S3 buckets
- **Other features:**
  - **Fully managed cloud data warehouse** scaling up to 2880 cores (5760 vCPUs) per cluster, multi-petabyte-scale, multi-performant storage
  - **Support for multiple storage options:** Amazon S3 and Block Storage
  - **Storage auto-increase for Block Storage** on set threshold ensuring you never run out of storage for your workloads
  - **Granular, schema-level backup/restore** to S3, restoring only the data you need
  - **Integration with IBM AppID** for easy integration with Azure Active Directory
  - **New APIs** for scaling, updates, backup/restore, logging
- **Soon after GA:**
  - **Cross-region snapshot backup** to Amazon S3 for disaster recovery
  - **Self-service maintenance windows** for product and database engine updates



IBM **Db2 Warehouse on Cloud**

Overview | Resources | Performance | Queries | Workload

1 hour | **6 hours** | 12 hours | 24 hours | 3 days | Custom

Sync charts

Resource consumption

Consumed proportion of system resources.

Query run time distribution

Relative query duration over time.

Total queries in the last hour: 1500 Average query run time: 19ms

Active connections

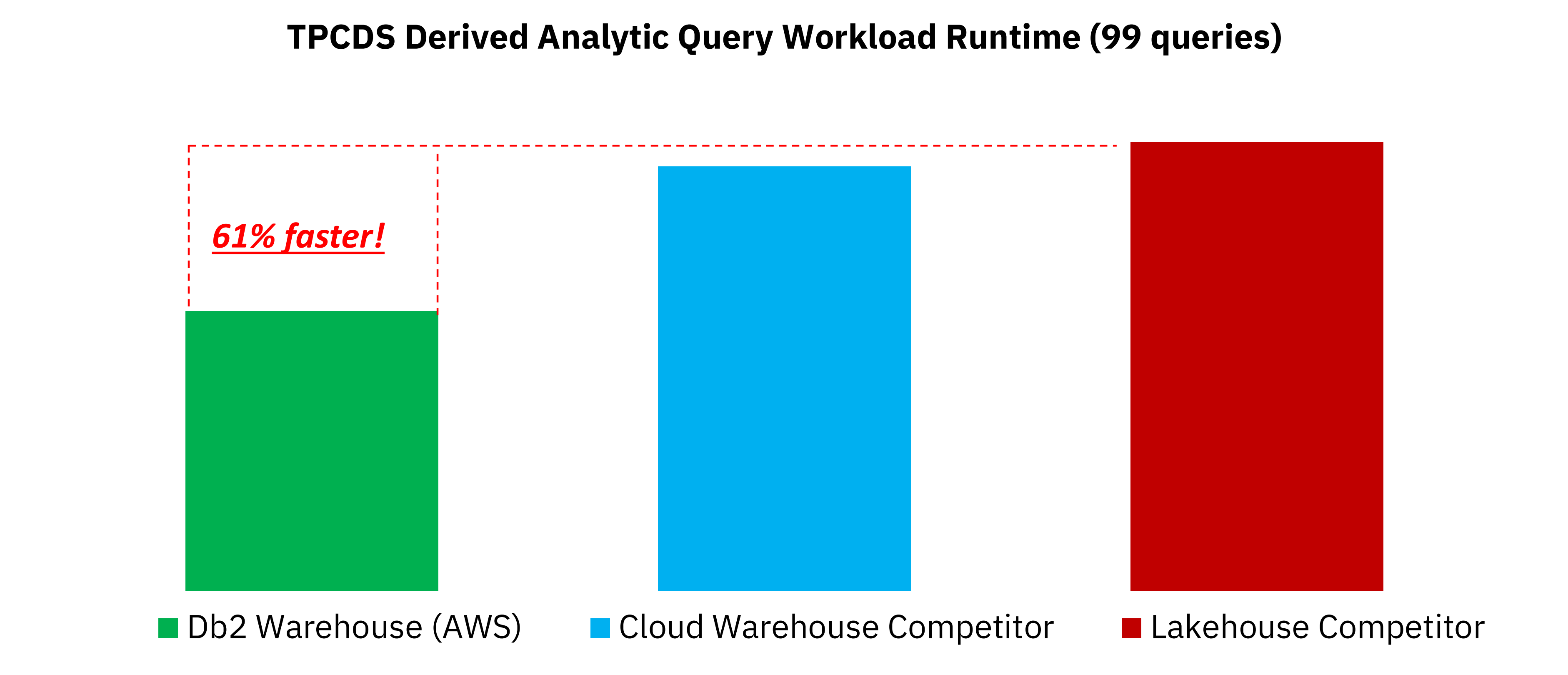
Number of connections over time.

Query throughput

Number of queries completed over time.

Averaging 29 queries per minute.

Blazing Fast Analytics with Db2 Warehouse on Cloud *Gen 3* on AWS



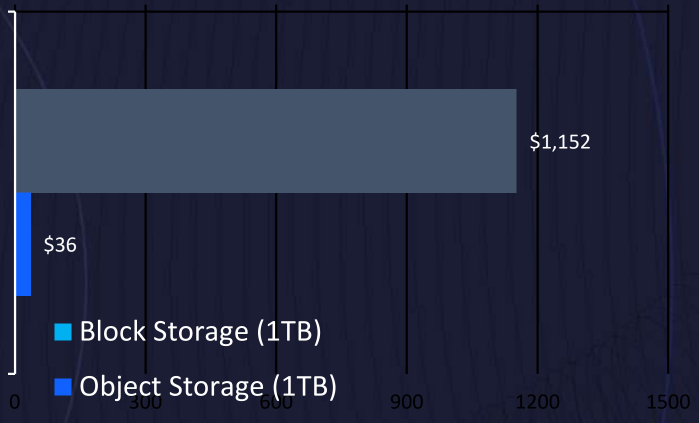
TPCDS Derived Analytic Query Workload Runtime (99 queries)

Db2 Warehouse on Cloud *Gen 3* on AWS

Storage Cost:

34x

Less expensive to host Db2 data on object vs block storage[1]



Cost of Db2 Warehouse block vs. object storage.

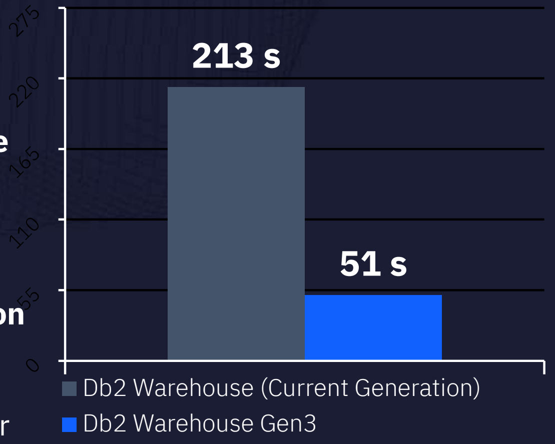
Performance:

4x

Faster query performance

When Gen3 is compared against the prior generation

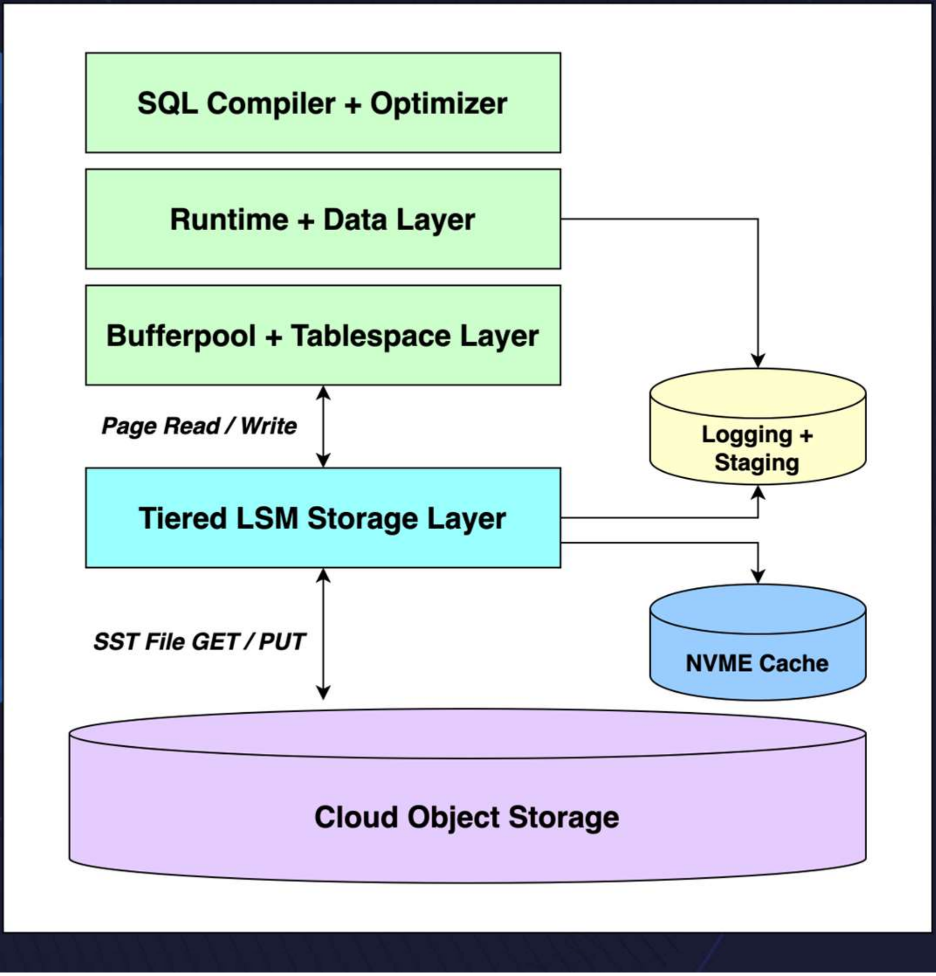
Note: Lower number is better



- IBM Big Data Insight (BDI) Benchmark simulates real-world deep analytics, reporting, and dashboard queries
- 10TB Db2 data warehouse residing either on block storage (current generation) or object storage (Gen3)
- 16 concurrent users running a variety of ML, reporting, and dashboard queries
- Cold cache start for both in-memory buffer pools or the NVMe cache

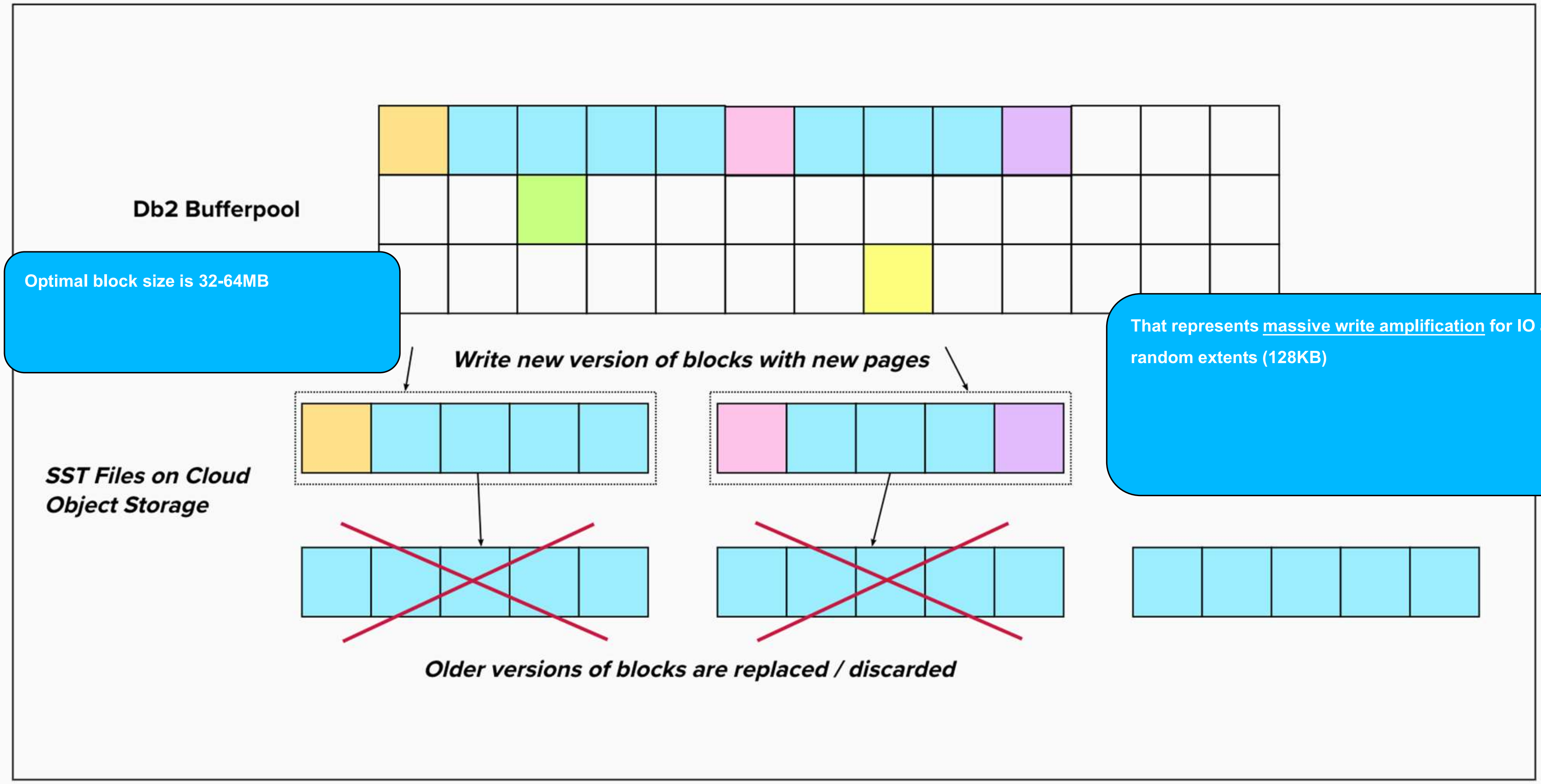
[1] Block vs Object Storage comparison depicts difference between published prices for Amazon EBS 1TB of io1 at 6 IOPS/GB (and additional tiers to support Db2 data) vs Amazon S3. This metric is not an indicator of future storage pricing for Db2 Warehouse Gen 3.

Native Cloud Storage Architecture



- Existing Db2 component stack down through bufferpool + tablespace layer
- New tiered storage layer that sits between tablespace storage and physical storage
- Uses an LSM storage organization to efficiently store page data in SST files on cloud object storage
- Tiered NVME cache enables high performance query processing and bulk ingest
- Existing Db2 logging maintains high performance for trickle feed

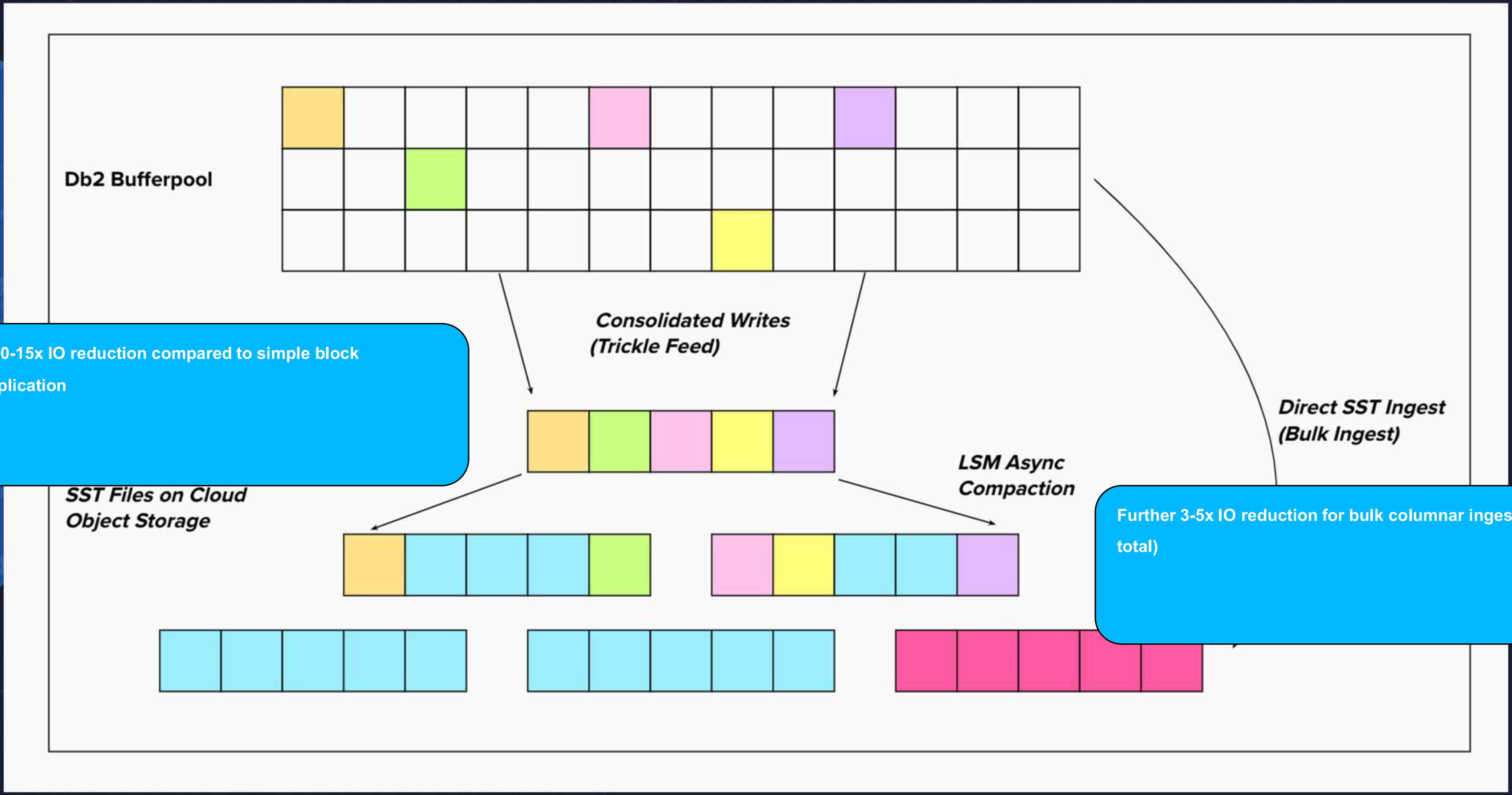
A Naïve Storage Model


Db2 Bufferpool
Write new version of blocks with new pages
SST Files on Cloud Object Storage
Older versions of blocks are replaced / discarded

Optimal block size is 32-64MB

That represents massive write amplification for IO across random extents (128KB)

LSM Based Page IO



~10-15x IO reduction compared to simple block replication

Further 3-5x IO reduction for bulk columnar ingest (~50x total)

DATALAKE Tables

Db2WoC Gen3 now also supports Open Data Formats as DATALAKE tables, allowing for seamless access to other data within the enterprise for integrated workloads.

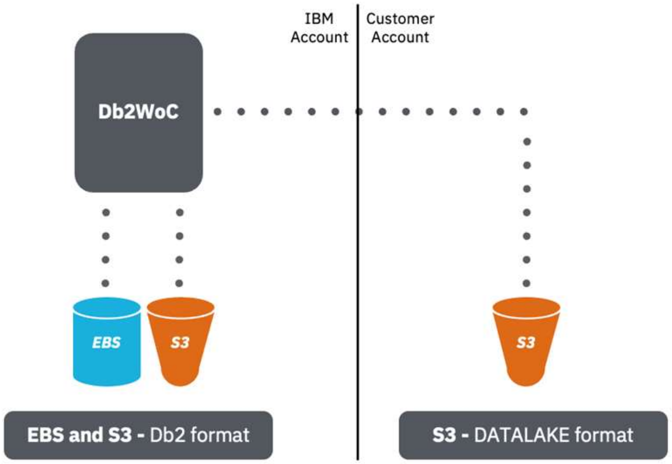
Leverage existing compute resources dedicated to the warehouse

- Facilitate data use to and from the Db2 Warehouse to quickly access a variety of enterprise data
- Leverage the high performance Db2 engine for queries against enterprise data

Db2WoC provides interfaces for customers to leverage their enterprise data residing in object storage as DATALAKE tables

- Supports both regular and Iceberg DATALAKE table types, based on existing data formats or for business/technical requirements such as ACID compliance
- Browse, explore, and query enterprise data in both Db2 and DATALAKE formats, using either the web-based UI, or through SQL
- Access data in place within DATALAKE tables, joining as necessary with Db2 based data for queries
- Access data within DATALAKE tables and import into Db2 formatted tables
- Create new DATALAKE tables in S3 and export from Db2 formatted tables

DATALAKE tables support



```
CREATE DATALAKE TABLE my_datalake_table
  ( id INT,
    name VARCHAR(8)
   )
   STORED AS PARQUET
   STORED BY ICEBERG
   TBLPROPERTIES ('external.table.purge'='true')
   LOCATION 'DB2REMOTE://mybucket-alias//
                         my_datalake_table';
```

DATALAKE Tables

A DATALAKE "Table" is a collection of files serialized using an Open Data File (ODF) format (TEXTFILE, ORC & Parquet) stored on remote storage (S3, COS). Data is not 'owned' by Db2.

The metadata of the table is stored in a metastore server:

- Location, Schema, Manifest lists

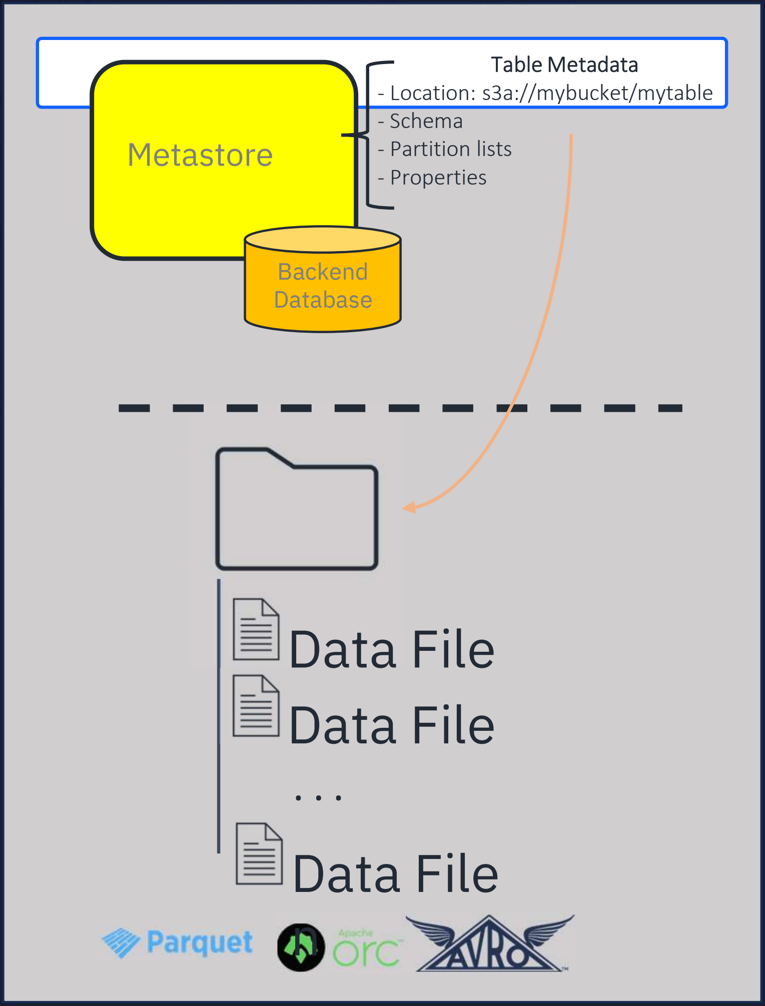
We use Hive as the metastore, a.k.a Hive metastore (HMS)

Db2 must query the metadata first and can proceed to read the data from remote storage

Benefits:

- Interoperability of open data formats
- Data sharing

Restrictions: Db2 Warehouse (MPP only / Db2U intel/amd and ppcle only initially)



DATALAKE Tables – Regular vs. Iceberg

REGULAR (open data formats)

- Default type on CREATE DATALAKE TABLE
- Does not support ACID properties for DML nor even simple transactions.
- Recommended for read-only or append-only tables.
- Formats: best performing (Parquet, ORC), AVRO, JSONFILE and TEXTFILE

ICEBERG (open table format)

- Specify STORED BY ICEBERG on the CREATE DATALAKE TABLE
- Designed* to support ACID properties (does so via snapshots)
- Recommended for scenarios which requires more flexibility on DML operations
- Formats: Parquet, ORC , AVRO

*more details on limitations later

Note: Iceberg and regular DATALAKE Tables are additionally bound by Hive limits for things such as table, schema and column names and other limits.   In addition, see docs for mappings and limits when it comes to data types when creating Regular and Iceberg DATALAKE tables..

DATALAKE Tables – DDL Samples

CREATE - This creates the table in the Db2 catalogs and the HMS. If the data files already exist, pre-existing data can be read. Whether the files pre-exist or not, new data can be appended. See docs for additional limits/features, including partitioning options and datatype support.

```
CREATE DATALAKE TABLE my_datalake_table(id INT, name VARCHAR(8)) STORED AS PARQUET LOCATION
'DB2REMOTE://mybucket-alias//my_datalake_table’
```

DROP - This removes the definition from the db2 catalogs and HMS. See the docs for details on how to control whether the data files are dropped as part of the DROP command or not.

```
DROP DATALAKE TABLE my_datalake_table
```

Some ALTER operations are also supported.

The CREATE/DROP/ALTER DDL are outside the scope of the transaction and rollback/commit at statement level.

For CREATE TABLE AS it acts as 2 individually committed statements (CREATE+INSERT)

DATALAKE Tables – Use Cases

First step is to create the DATALAKE Table which includes ensuring access to the S3 bucket & using a stored procedure to set up the storage access alias. Once created there are 3 common use cases.

1. Queries with DATALAKE Table(s)

DATALAKE tables can be used in Db2 queries like (and along with) Db2 tables.

2. Loading data from DATALAKE Tables into Db2 tables

Data can be loaded into Db2 tables from DATALAKE tables by an INSERT into the Db2 table with the SELECT FROM clause accessing a DATALAKE table.

3. Exporting data from Db2 into DATALAKE Tables

Data can put into DATALAKE Tables by using the INSERT statement into the DATALAKE table with the SELECT FROM clause accessing a Db2 Table.

Limitations

- Anything that would be to related ownership or responding to changing data: CREATE INDEX, REORG, TRIGGERS, ..
- Limited transactional support (iceberg only, single table only)

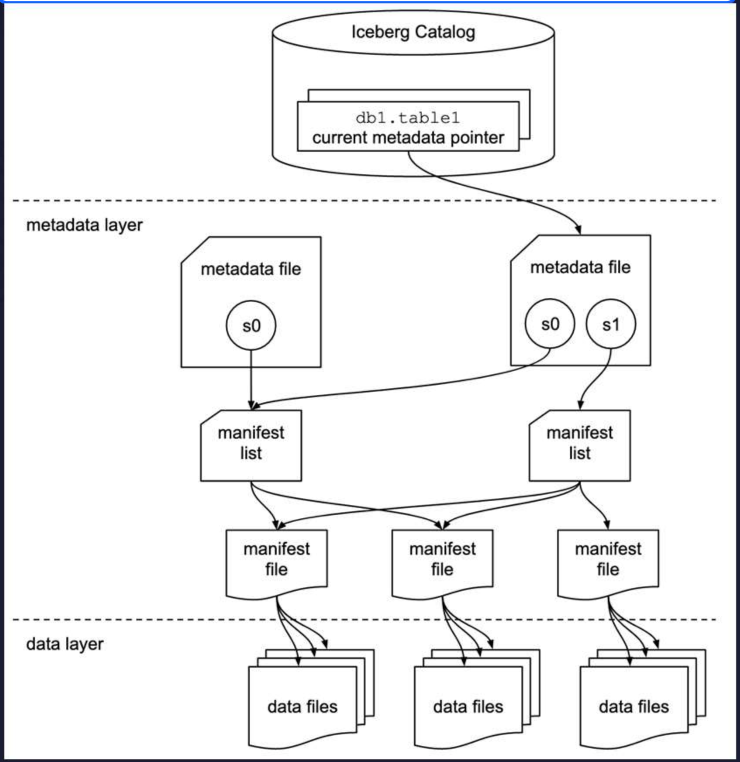
Apache Iceberg – An Open Data Table Format

Full open-source, Open Data Table format, quickly becoming an industry standard

Relies on Open Data File formats for storage, but provides an additional layer of metadata for enhanced capabilities

- Native Encryption
- Full Schema Evolution
- Data Compaction
- Hidden Partitioning
- Integrated Compression
- Time-Travel & Rollback
- ACID Transactions
- Expressive SQL

ICEBERG



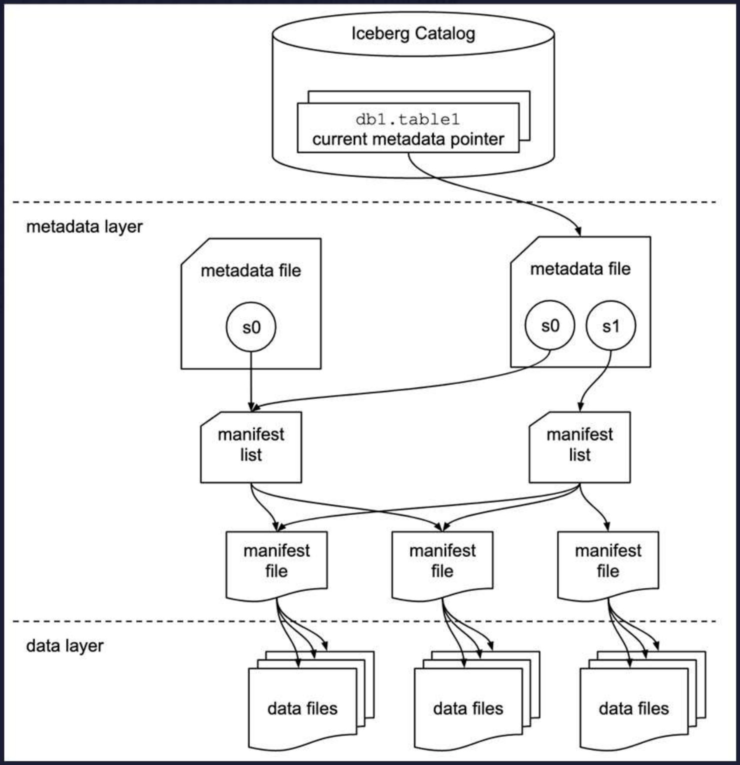
Iceberg Support on Db2 Tables

Support today for:

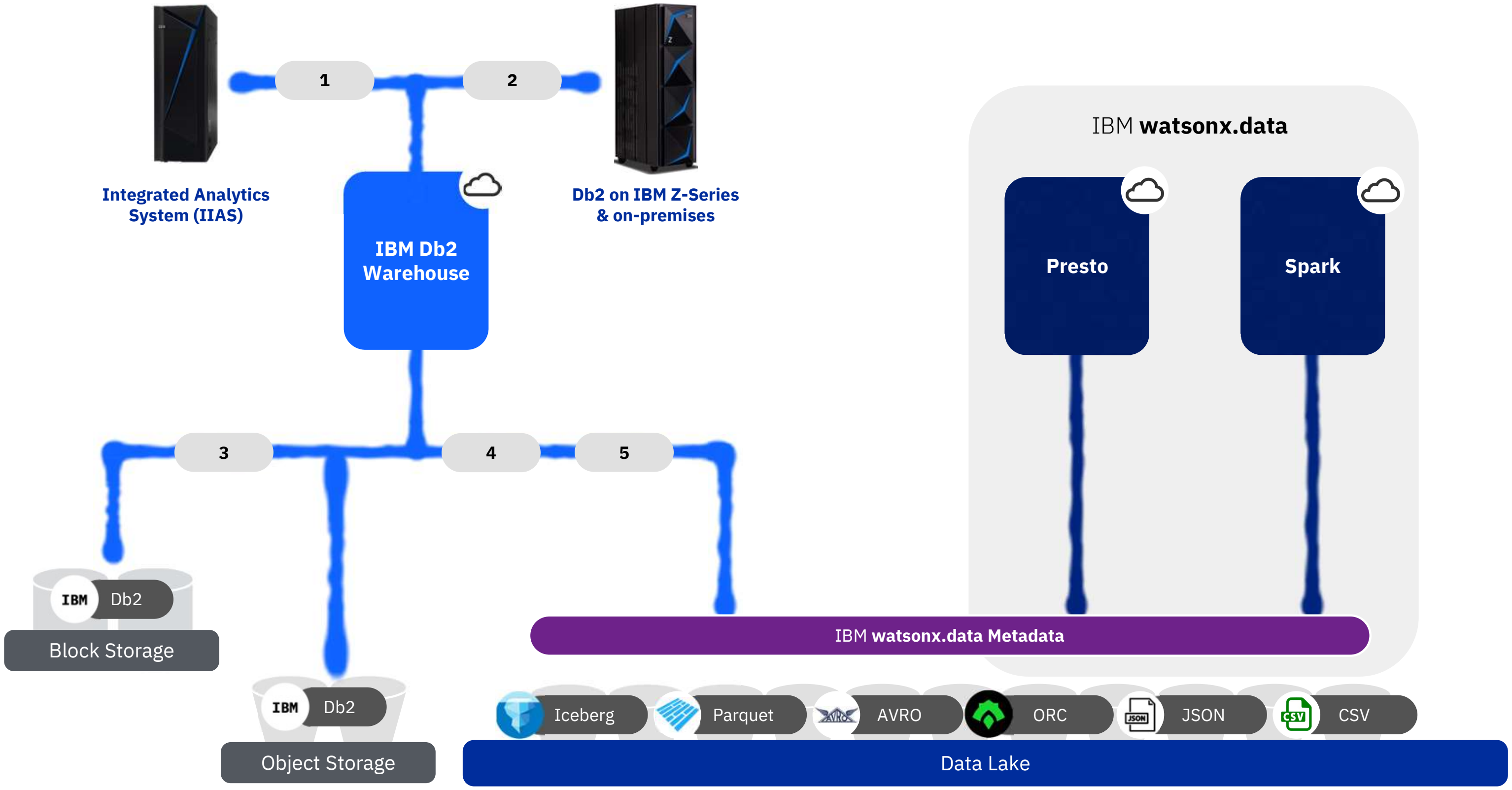
- CREATE (DDL or GUI)
  - Includes partitioning support
- SELECT (aka READ)
- INSERT
- Snapshot Rollback
- Most schema evolution

In Future:

- UPDATE / DELETE
- Time travel



Journey from Db2 Warehouse to **watsonx.data** and back



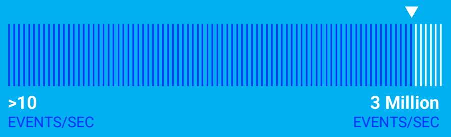
1. **Modernize the warehouse**
   Upgrade from your existing on-premises analytics appliance to the cloud
2. **Replicate data from Db2 on Z to Db2 Warehouse with live twinning**
   Share data from Db2 on Z with your Db2 Warehouse & data lake
3. **Modernize cloud data warehouse storage**
   Take advantage of inexpensive, highly reliable object storage and tiered NVME caching
4. **Share warehouse data with the data lake**
   Join data from your warehouse with your data lake for new insights
5. **Promote data lake data to the warehouse**
   Easily promote curated data lake data to Db2 Warehouse

When to bring Db2 Warehouse into a Lakehouse Strategy

1 Fast Ingest

Real-Time Applications

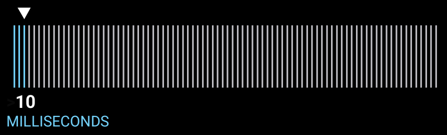
Millions of events per second, fully consistent and immediately available



2 Low Latency

SLA Sensitive Apps

Sub-second latency for high volume operational queries



3 Fast Analytics

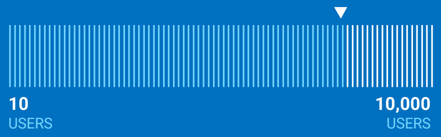
Business Apps

Fastest performance in the industry with Db2 BLU Acceleration on S3 storage

4 High Concurrency

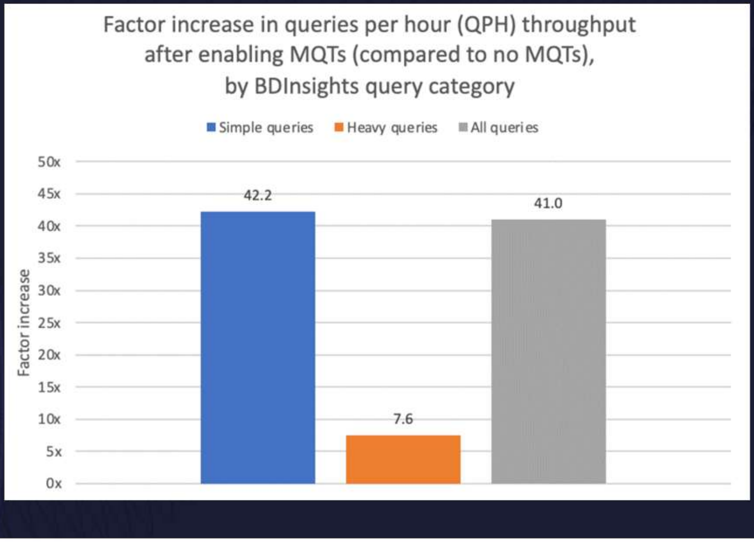
Highly Concurrent App

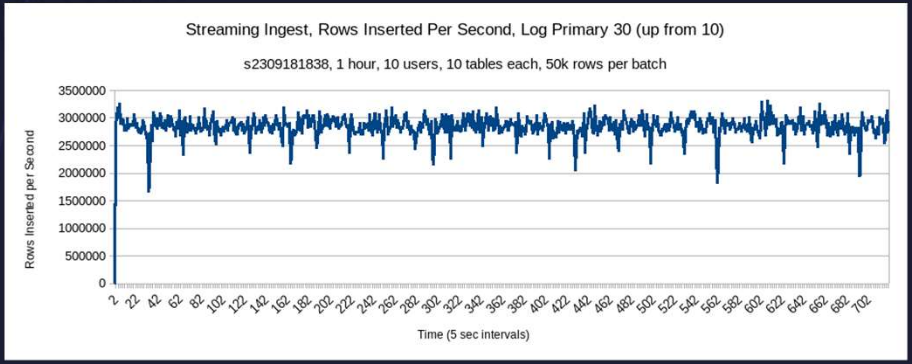
Support thousands of users automatically managed with Db2's Adaptive Workload Mgmt



New Use Cases for Db2 Warehouse with Native COS

- Acceleration of queries on open data format tables (DATALAKE) by +40X using MQTs
  - NCOS MQTs benefit from the built-in multi-tier cache
  - Ability to query both native tables and ODF / Datalake tables
- Continuous high-volume streaming
  - NCOS enables the ability to store the kind of data volumes generated by this use case
    - 3 million rows per second results in 3.78 PB of data per year (40-byte rows, $128K storage per year vs $4.4 million in block)
  - New MEMORY_TABLE UDF enables the batching necessary for this use case


Factor increase in queries per hour (QPH) throughput after enabling MQTs (compared to no MQTs), by BDInsights query category


Streaming Ingest, Rows Inserted Per Second, Log Primary 30 (up from 10)

s2309181838, 1 hour, 10 users, 10 tables each, 50k rows per batch

Cloud SaaS Architecture

Db2 Warehouse on NextGen product tiers AWS and IBM*

	Starter Model	**Middle-tier model**	**Large-scale Model**
	Storage Optimized An MPP data warehouse ideal for cost-effective data warehousing, dev/test environment, infrequently accessed data. Support for Object Storage + DataLake tables	**Compute Optimized** An MPP data warehouse ideal for high-performance and production workloads Support for Object Storage + DataLake tables	**Compute Optimized** An MPP data warehouse ideal for high-performance and production workloads and very large systems Support for Object Storage + DataLake tables
Architecture	**MPP** (Massively Parallel Processing)	**MPP** (Massively Parallel Processing)	**MPP** (Massively Parallel Processing)
Configuration	Dedicated compute & storage	Dedicated compute & storage	Dedicated compute & storage
Cloud Platform Support	Amazon Web Services	Amazon Web Services	Amazon Web Services
Relative Cost	$$$	$$$$	$$$$$
Independently scale storage & compute	Yes	Yes	Yes
Storage for Table Data	Up to 80 TB (Block + Object Storage)	Up to 240 TB (Block + Object Storage)	Up to 600 TB (Block + Object Storage)
Compute Range	16 – 128 cores	48 – 576 cores	288– 2880 cores
Private Endpoint	Available	Available	Available

Db2 Warehouse NextGen Platform Architecture

Delivered on **HDM Common Services Platform**

- Common Platform for many SaaS services, Db2 on cloud, Db2 Warehouse, DVaaS and now Watsonx.data
- A similar deployment and management experience for our Cloud Operations team
- Extensive use of the Kubernetes Operator Framework

Control Plane provides all platform level interactions

- Flow From IBM Cloud Catalog to Broker for provisioning
- Billing Flow to send billing to BSS
- Take a customer request and turn it into a provisioning request
- Can run on any cloud platform, currently hosted on IBM Cloud

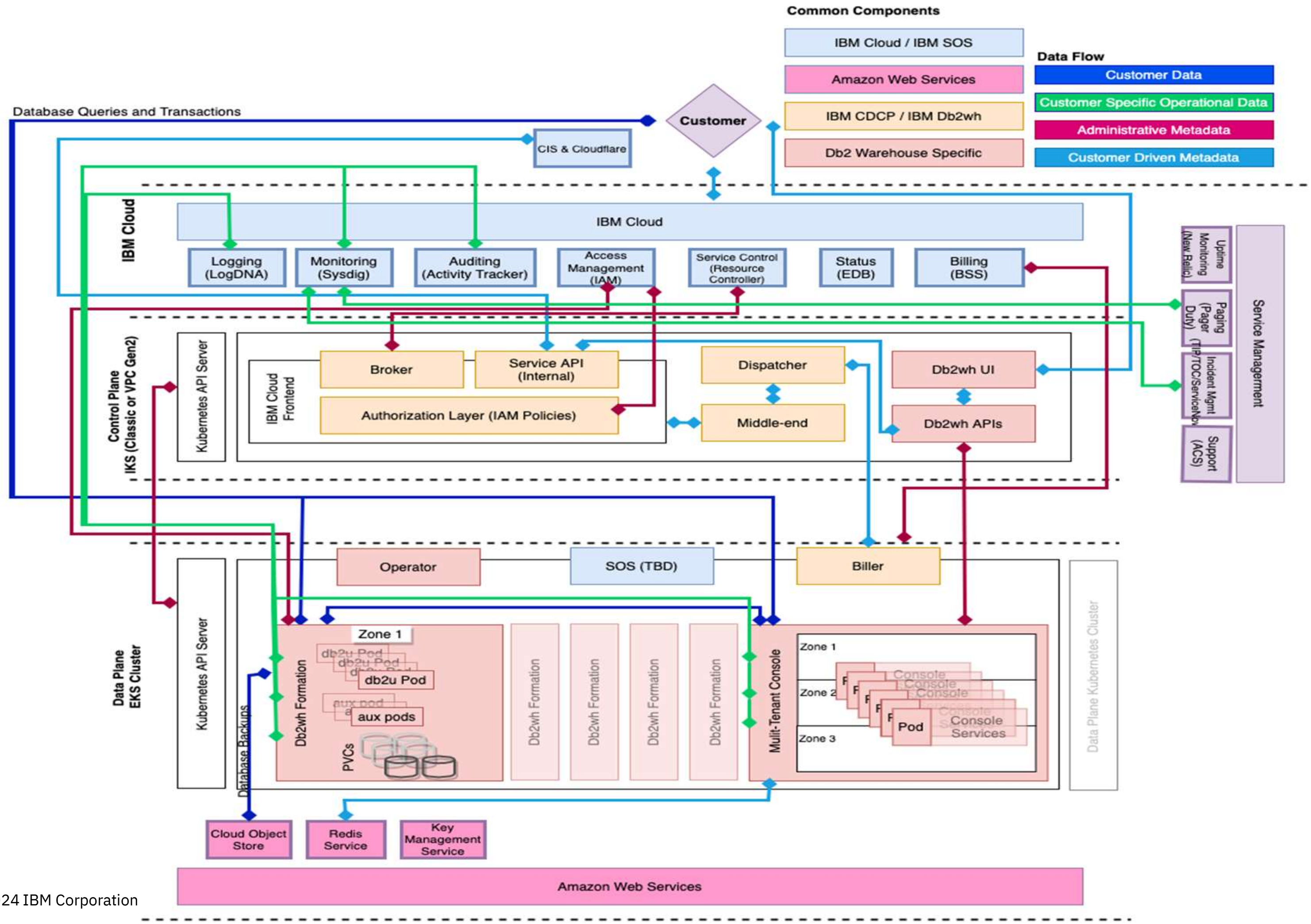
Data Plane

- Deployed on the native public cloud platform (in this case AWS EKS) on an **IBM owned account**
- All customer provisioned components are part of this Data plane
- Includes the multi-tenant console

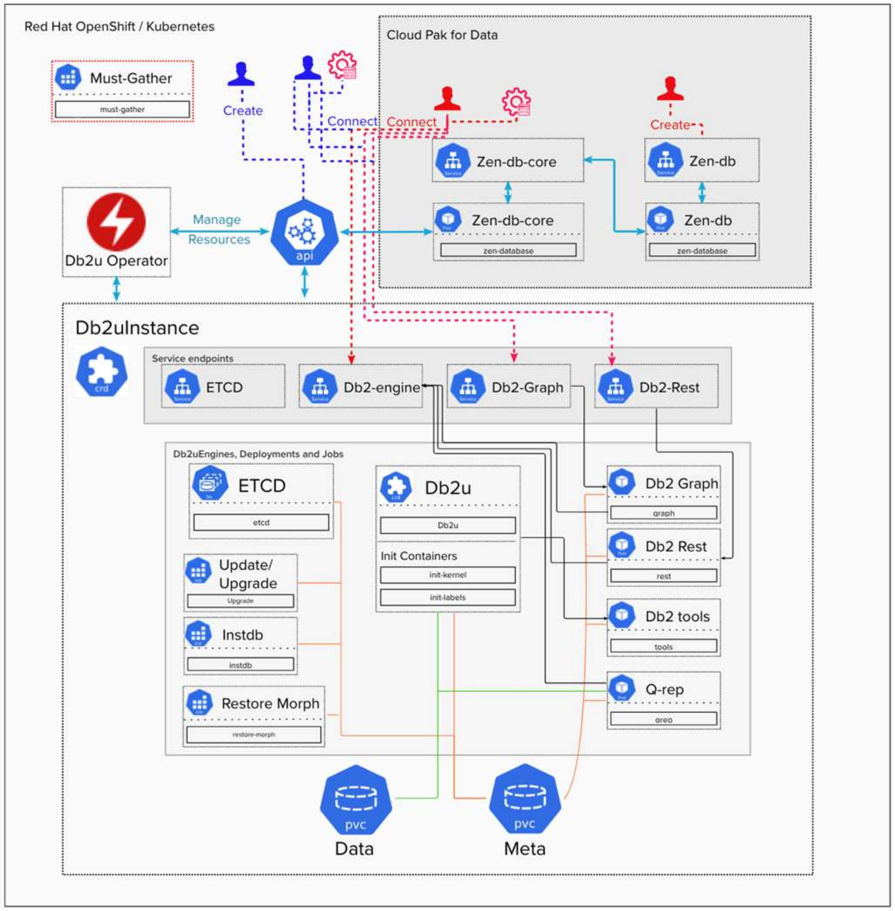
Runs on Db2u

- Db2u containerization provides the underlying Db2 infrastructure (microservice architecture, container images, orchestration – including deployment and scale)
- Decoupling of compute and storage

Db2 Warehouse NextGen – Platform architecture diagram



Db2u



Db2u is the base for all our deployments

– Provides the container images and management infrastructure

Managed via k8s Custom Resource (CR)

– Kind: `Db2uInstance`

– Short name: `db2`

Underlying Kubernetes resource model:

- Db2 Engine Pod lifecycle managed using a NEW `Db2uEngine` CR, which manages a collection of Db2 Pods.
- Onetime tasks managed via a `Job` resource
- In-pod HA to recover Db2 failures, avoiding a pod lifecycle event. This built-in HA leverages ETCD for state information
- Lifecycle of (stateless) Add-Ons (REST, Graph, Q-rep, etc.) managed via `Deployment` resources

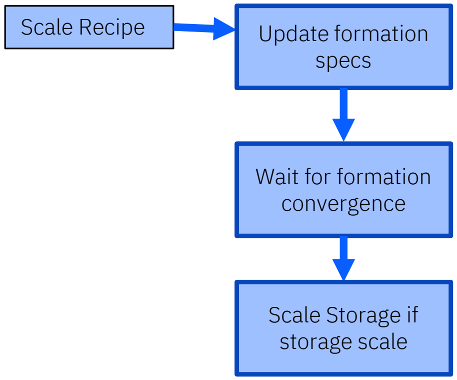
Db2 Warehouse NextGen Deployment Model

- DB2u containers are the base.

- Every customer deployment maps to a Formation. A formation is realized by the Operator model (internally using the db2u layer) and deploys a complete cluster
  - Gets storage provisioned (Includes block storage and Object Storage buckets)
  - Sets up NVMe for Object Storage caching
  - Gets N/W components created (we create both Private and Private endpoints at this stage automatically)
  - Creates container/pod configurations based on formation specs
  - Db2u process sets up a db2 database with the storages and sets up tablespaces with them
  - Customer gets two endpoints, one to the Console and the other to the database.

- Recipes
  - Used for all Operations like scale, backup, restore, network configuration, addition of users etc.

Db2 Warehouse on Cloud NextGen Scaling


Scale Recipe
Update formation specs
Wait for formation convergence
Scale Storage if storage scale

- Scale using the Console or APIs
  - The platform provides mechanics to initiate a scale (we create a scale recipe)
  - In the scale recipe, we provide the new core and storage values.
- The operator framework will
  - Gracefully shutdown the engine and scale down the formation
  - Bring it up in the new spec (appropriately setting the correct MLN count per pod)
  - As the new spec is brought up, new nodes are automatically added by the Cluster Auto Scaler
  - If it is a shrink operation, unused nodes are automatically returned by the Cluster Auto Scaler
- Compute Expansion / Shrink incurs a Downtime
- Storage scaling works the exact same way (recipe mechanism)
  - Storage scale up is a totally online operation
  - Block Storage shrink (support date TBD) will be offline in nature

Db2 Warehouse on Cloud NextGen - High Availability (HA)

- When a Db2 node goes down,
  - Kubernetes will automatically schedule the pod to run on a free node from the pool
  - Wolverine will wait for the pod to come back up and take the engine through a recovery process
  - Engine is back running in full capacity within a short time.

- When a Db2 pod goes down
  - Kubernetes will restart the pod
  - Wolverine will wait for the pod to come back and take the engine through a recovery process.

- Multi prong, provided by
  - Wolverine (our HA component) - Core Db2u container
  - Kubernetes - non Db2 containers
- Wolverine responsible for and provides
  - System recovery (when the pods fail)
  - Overall System health and status recording in ETCD
  - Device Manager framework
  - HA loop (for monitoring and handling HA for processes within core Db2 Container)
  - API endpoints
- Kubernetes
  - Pod level HA based on liveliness and readiness probes

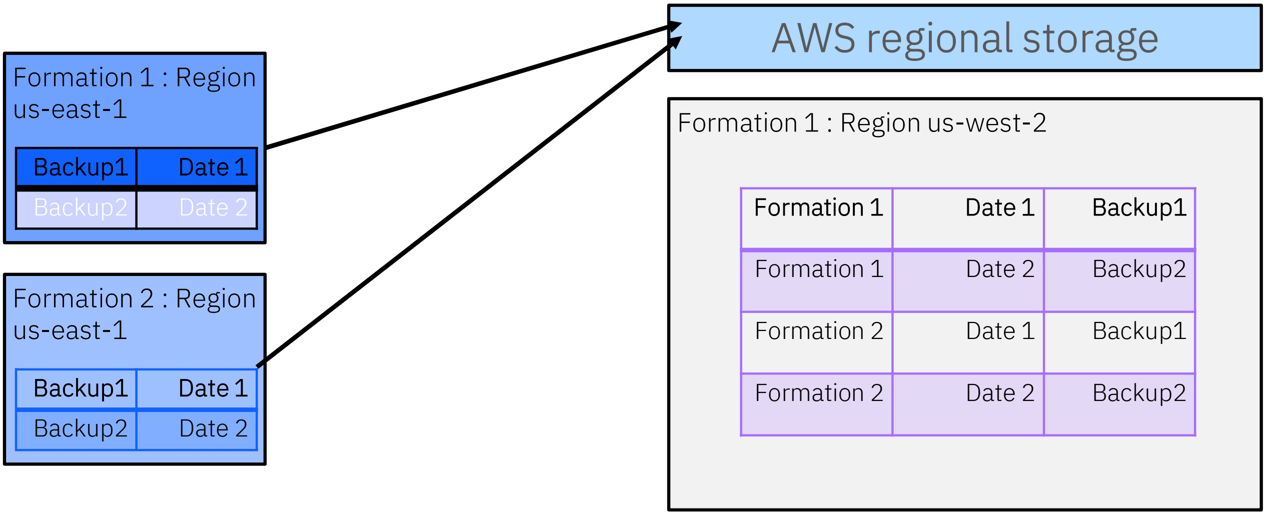
Db2 Warehouse on Cloud NextGen Backup & Restore

- Backups - a combination of Snapshot Backups and S3 backup
  - Scheduled every 24 hour period or on demand
  - Scheduled or on-demand both internally create a Recipe
  - The Recipe will go through the following phases
    - Do a db2 write suspend (including S3 deletes off)
    - Create volume snapshots (Block)
    - Initiate S3 backup process (Object)
    - Backup metadata
    - Write resume (but leave S3 delete off)
    - When S3 backup completes (tag backups and turn on S3 deletes)

- Restore process, On-demand, internally creates a Recipe.
- The recipe will go through the following phase
  - Stop Db2
  - Initiate meta data restore
  - Initiate S3 restore
  - Initiate volume snapshot restore
  - When S3 restore is ready to use, scales all pods up

Db2 Warehouse on Cloud NextGen
self serve DR Options

- Allows the customer to setup a region of their choice for DR recovery
- Both S3 Backups and Snapshots get copied to the DR region as soon as they complete on the Source region
- Customer can restore any backups on the DR region as long as plans match.



Formation 1 : Region us-east-1

Backup1	Date 1
Backup2	Date 2

Formation 2 : Region us-east-1

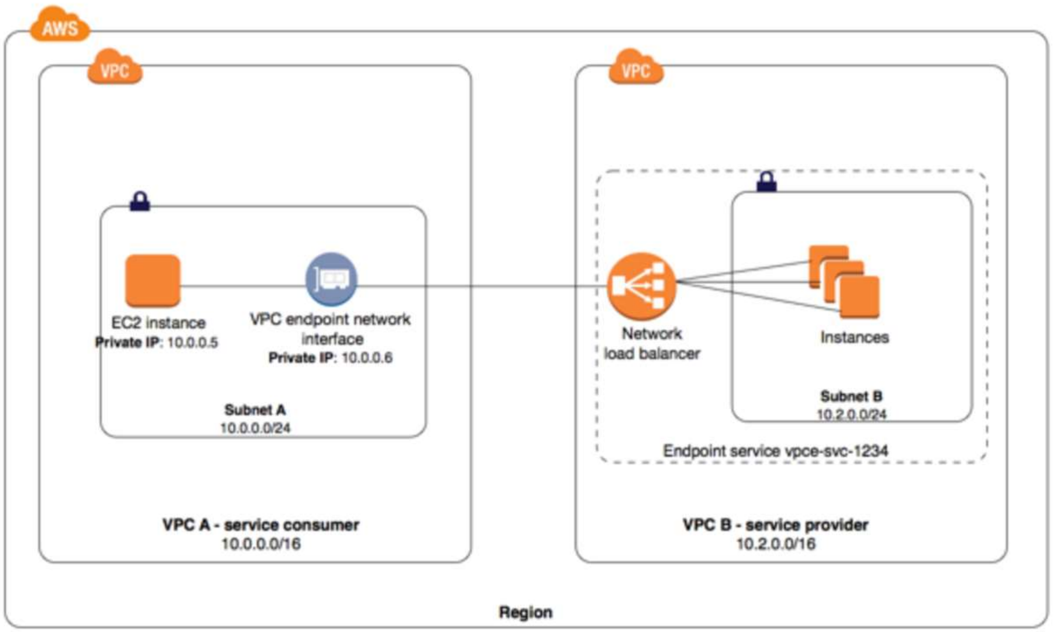
Backup1	Date 1
Backup2	Date 2

AWS regional storage

Formation 1 : Region us-west-2

Formation 1	Date 1	Backup1
Formation 1	Date 2	Backup2
Formation 2	Date 1	Backup1
Formation 2	Date 2	Backup2

Db2 Warehouse NextGen
Networking Connectivity



- Both public and private connectivity models provided
- Private Connections provided using AWS Private Link
  - Allows the customer’s connections from their AWS VPC to the Db2WoC VPC managed by IBM
- Completely self service in nature
  - Customers can enable public/private themselves

IDUG

2024 NA **Db2** Tech Conference

IDUG

Next Generation Db2 Warehouse: A New Architecture for the Cloud and Lakehouse Era

Speakers

David Kalmuk (dckalmuk@ca.ibm.com)
Venkatesh Gopal (gopalv@us.ibm.com)

Session Code: CLOUD3

Please fill out your session evaluation!

@IDUGDb2
#IDUG_NA24