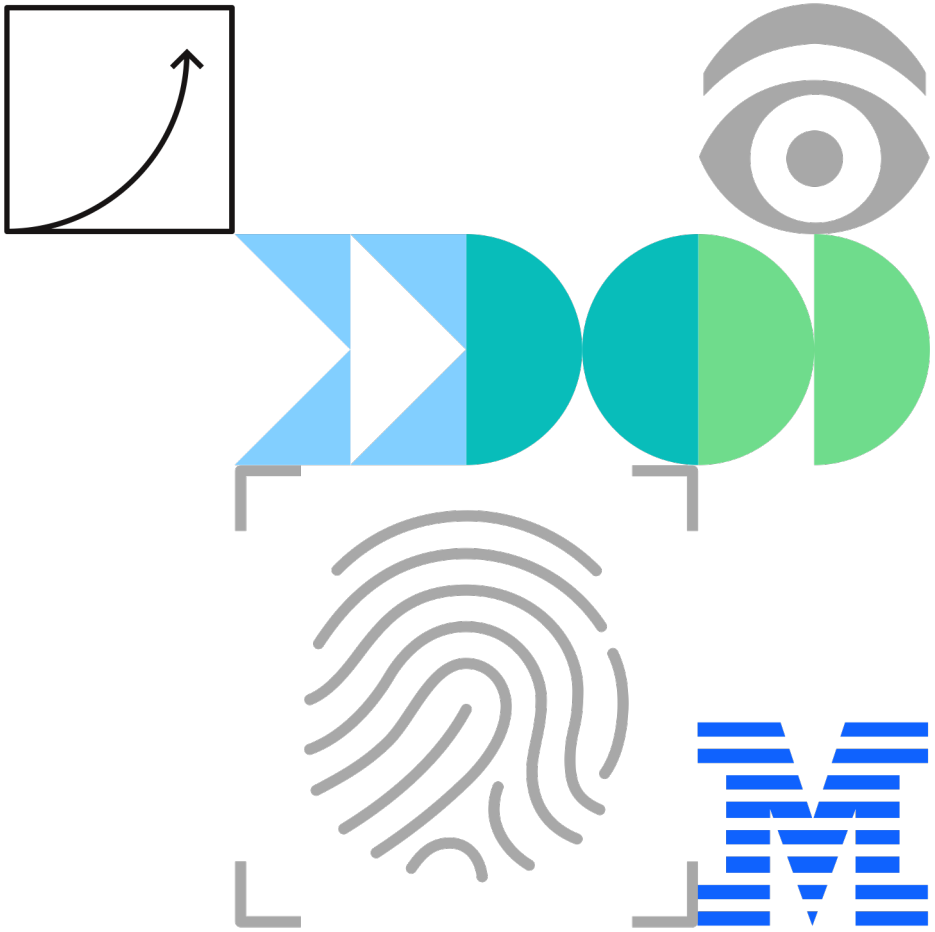
IBM **TechXchange**

Why is Db2 Warehouse Faster with Cloud Object Storage?



Christian Garcia-Arellano
STSM, Db2 OLTP Architect and Master Inventor

March 28, 2024

IBM

Notices and disclaimers

This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.

Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM.

Not all offerings are available in every country in which IBM operates.

Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.

IBM, the IBM logo, and ibm.com are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at: www.ibm.com/legal/copytrade.shtml.

Certain comments made in this presentation may be characterized as forward looking under the Private Securities Litigation Reform Act of 1995.

Forward-looking statements are based on the company’s current assumptions regarding future business and financial performance. Those statements by their nature address matters that are uncertain to different degrees and involve a number of factors that could cause actual results to differ materially. Additional information concerning these factors is contained in the Company’s filings with the SEC.
Copies are available from the SEC, from the IBM website, or from IBM Investor Relations.

Any forward-looking statement made during this presentation speaks only as of the date on which it is made. The company assumes no obligation to update or revise any forward-looking statements except as required by law; these charts and the associated remarks and comments are integrally related and are intended to be presented and understood together.

Agenda

01 Cloud Object Storage

02 Evolution of the Storage Architecture

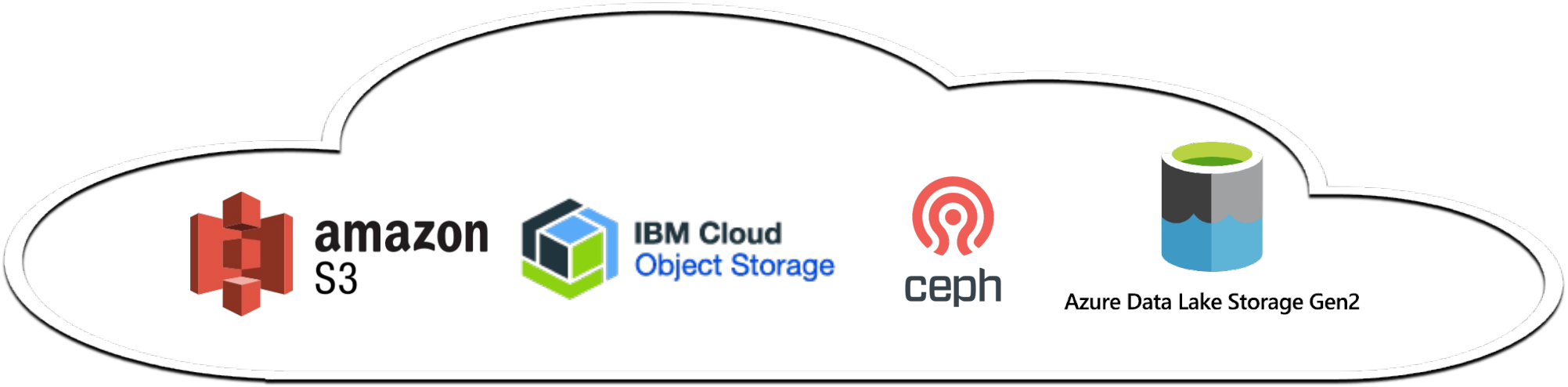
03 Native Cloud Object Storage Architecture

04 Three reasons for the speed up explained

05 User Experience and Out-of-the-box Set up for Native Cloud Object Storage

Cloud Object Storage

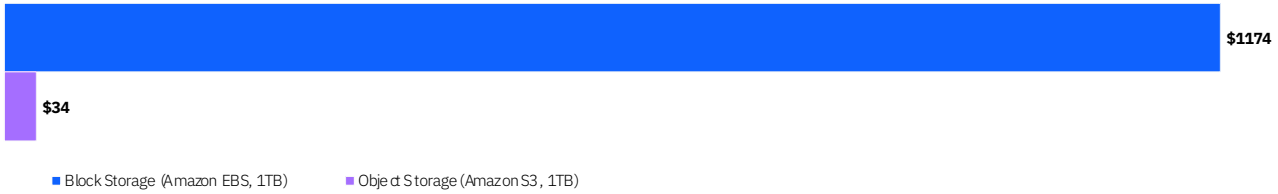
- ✓ **Lower cost**
- ✓ **Near unlimited scalability**
- ✓ **Extreme durability + reliability (99.999999999%)**
- ✓ **High throughput**



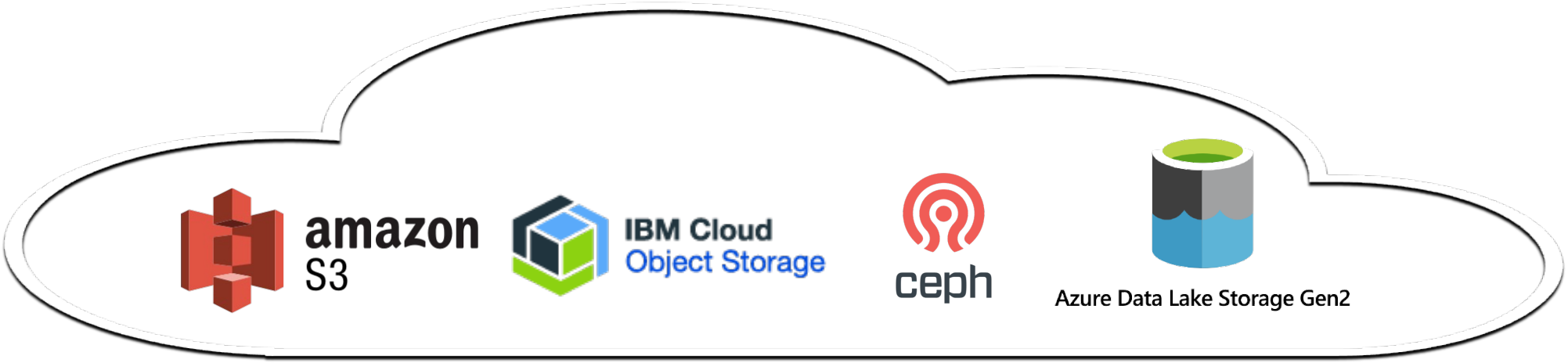
The Main Incentive: Lower Cost

34x

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



Block Storage (Amazon EBS) vs Object Storage (Amazon S3)
Cost reflects Amazon's list price for block storage (various tiers & IOPS levels) required to host an incremental 1TB of Db2 data

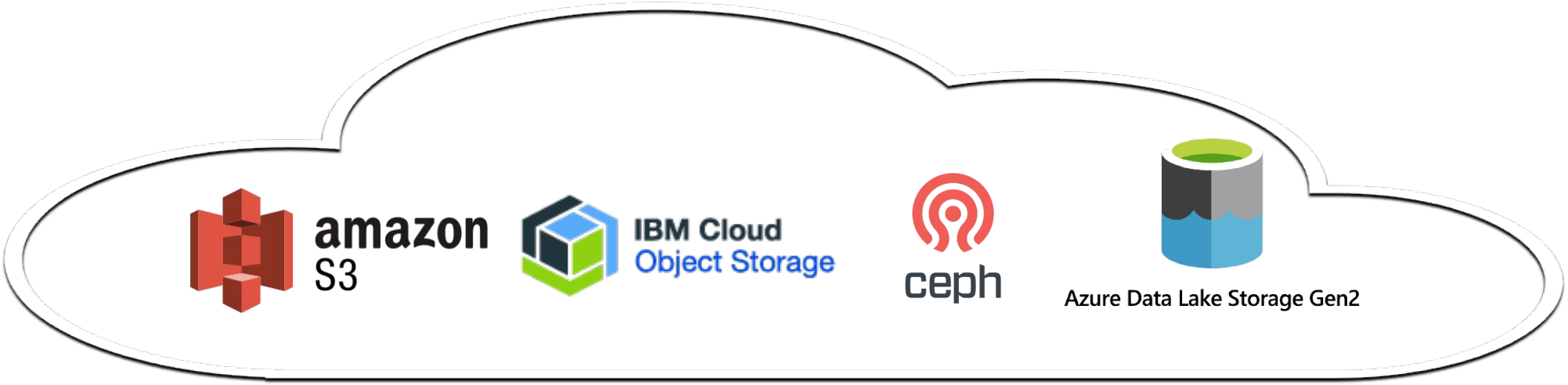


[1] Block vs Object Storage comparison depicts difference between published prices for Amazon EBS 1TB of io1 at 6 IOPS/GB (and additional tiers to support Db2 data) vs Amazon S3. This metric is not an indicator of future storage pricing for Db2 Warehouse Gen 3.

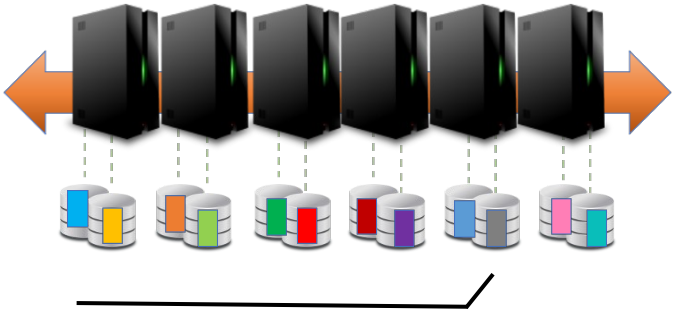
The catch

- ✓ **Lower cost**
- ✓ **Near unlimited scalability**
- ✓ **Extreme durability + reliability (99.999999999%)**
- ✓ **High throughput**

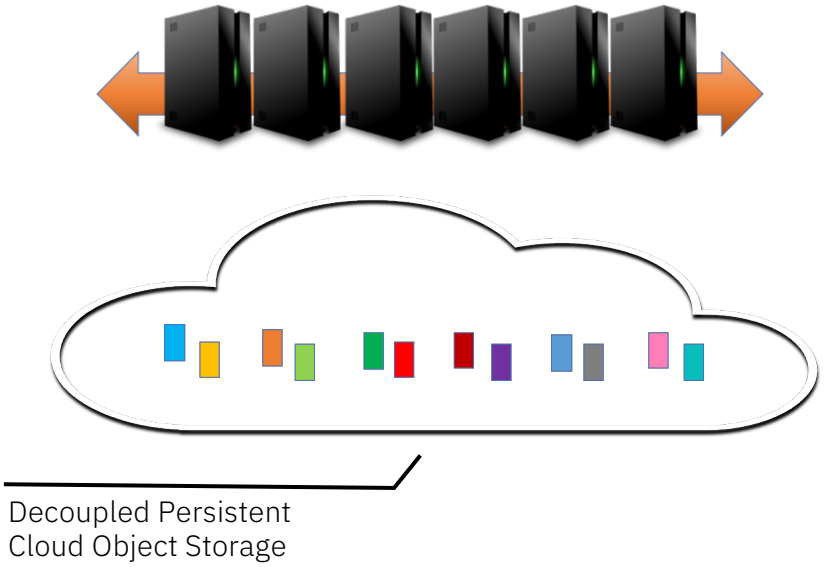
🛑 **High latency**



Evolution of the Storage Architecture



High-Performance
Cloud Block Storage



Decoupled Persistent
Cloud Object Storage

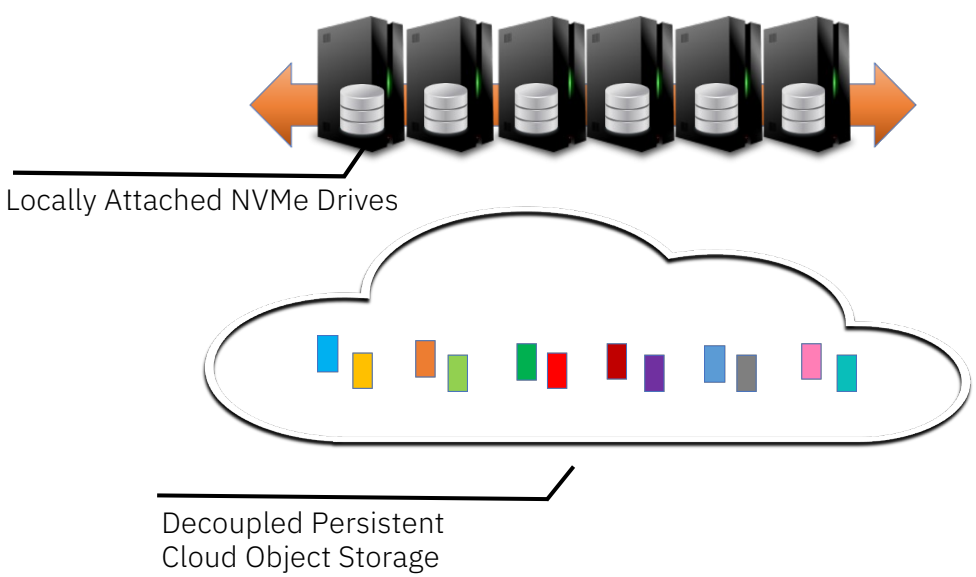
@ 10-30ms latency each (6 IOPS/GB)

@ 100-300ms latency per operation

Next Generation Db2 Warehouse Storage Architecture

- Locally attached NVMe drives
  - No replication
  - Nanoseconds to small single digit microseconds latency

- Cloud Object Storage
  - Throughput limited by network bandwidth
  - @ 100-300ms latency



And with that it was faster

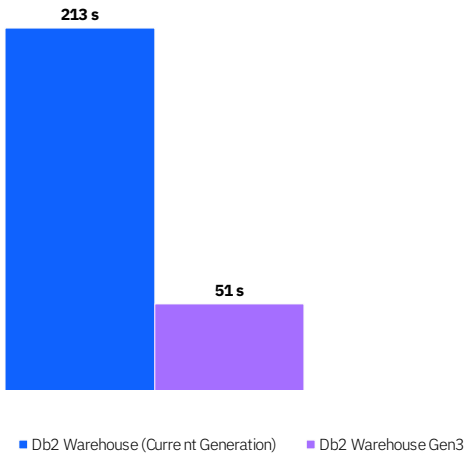
Performance numbers comparing current generation vs Gen3

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

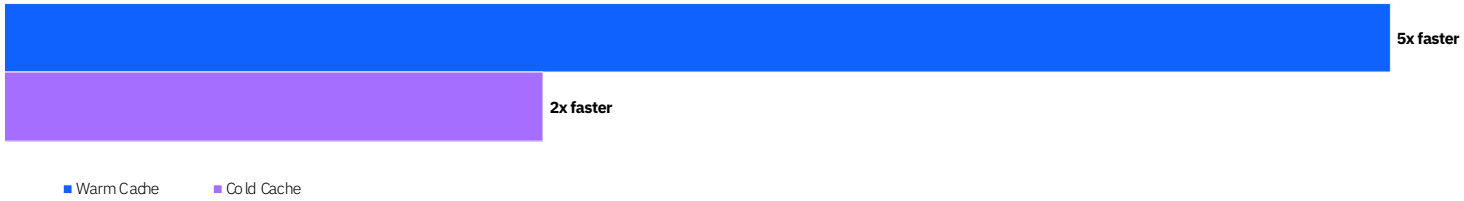
Cold cache start
for both the in-memory buffer pools or the NVMe cache

And with that it was faster

Performance numbers comparing current generation vs Gen3

4.5x

Average query speed-up ratio



TPC-DS benchmark
running industry standard queries

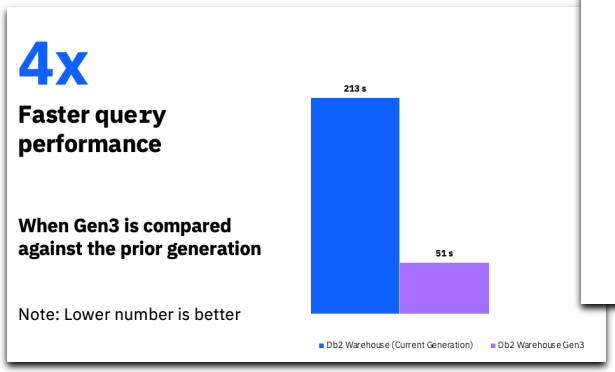
10TB Db2 data warehouse
residing either on block storage (current generation) or object storage (Gen3)

99 queries
serial test running SQL statements sequentially

Multi-temperature test
running queries on both a cold and warm cache

More performance results

- Concurrent query performance is 4X faster than block storage.
- Serial query performance is 4.5X faster than block storage.
- Bulk ingest is 3X faster than block storage
- NCOS is 40X faster in queries than Datalake tables.
- Faster against competitors, both WH and Lakehouse



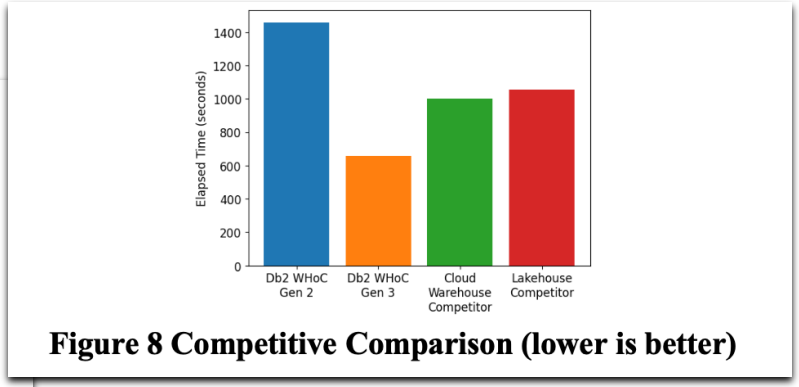


Figure 8 Competitive Comparison (lower is better)

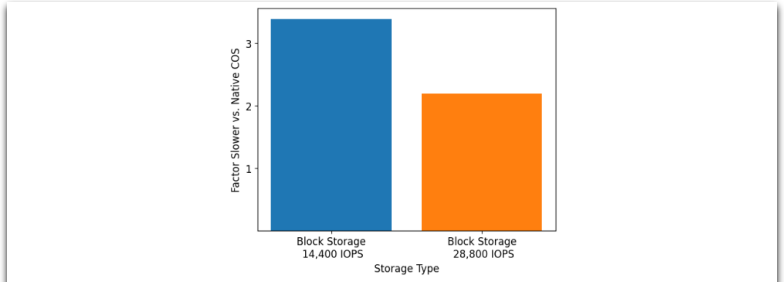
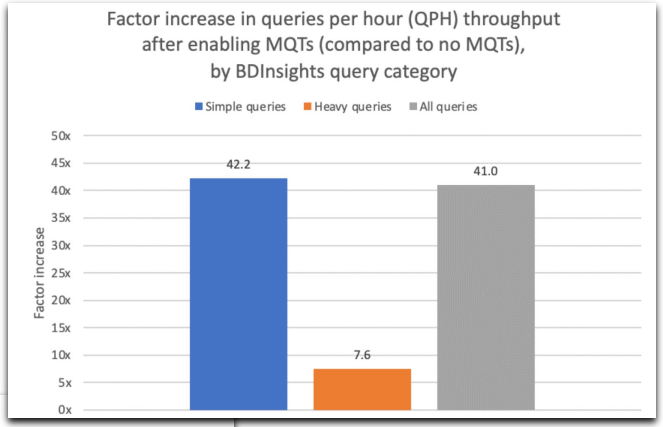
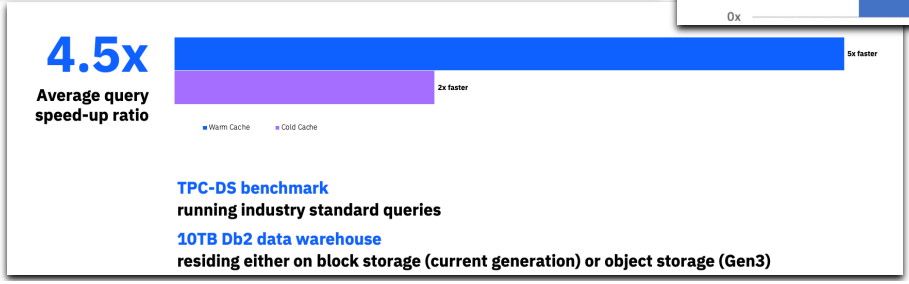


Figure 6 Bulk insert elapsed time for tables on block storage relative to Native COS tables





Agenda

01 Cloud Object Storage

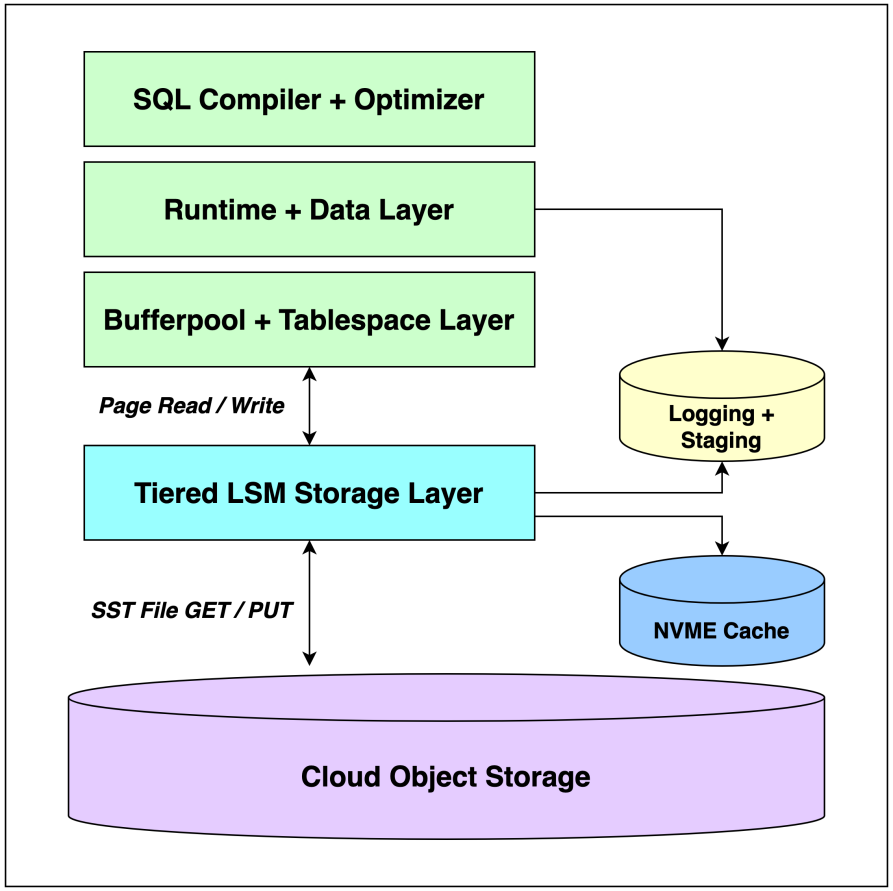
02 Evolution of the Storage Architecture

03 Native Cloud Object Storage Architecture

04 Three reasons for the speed up explained

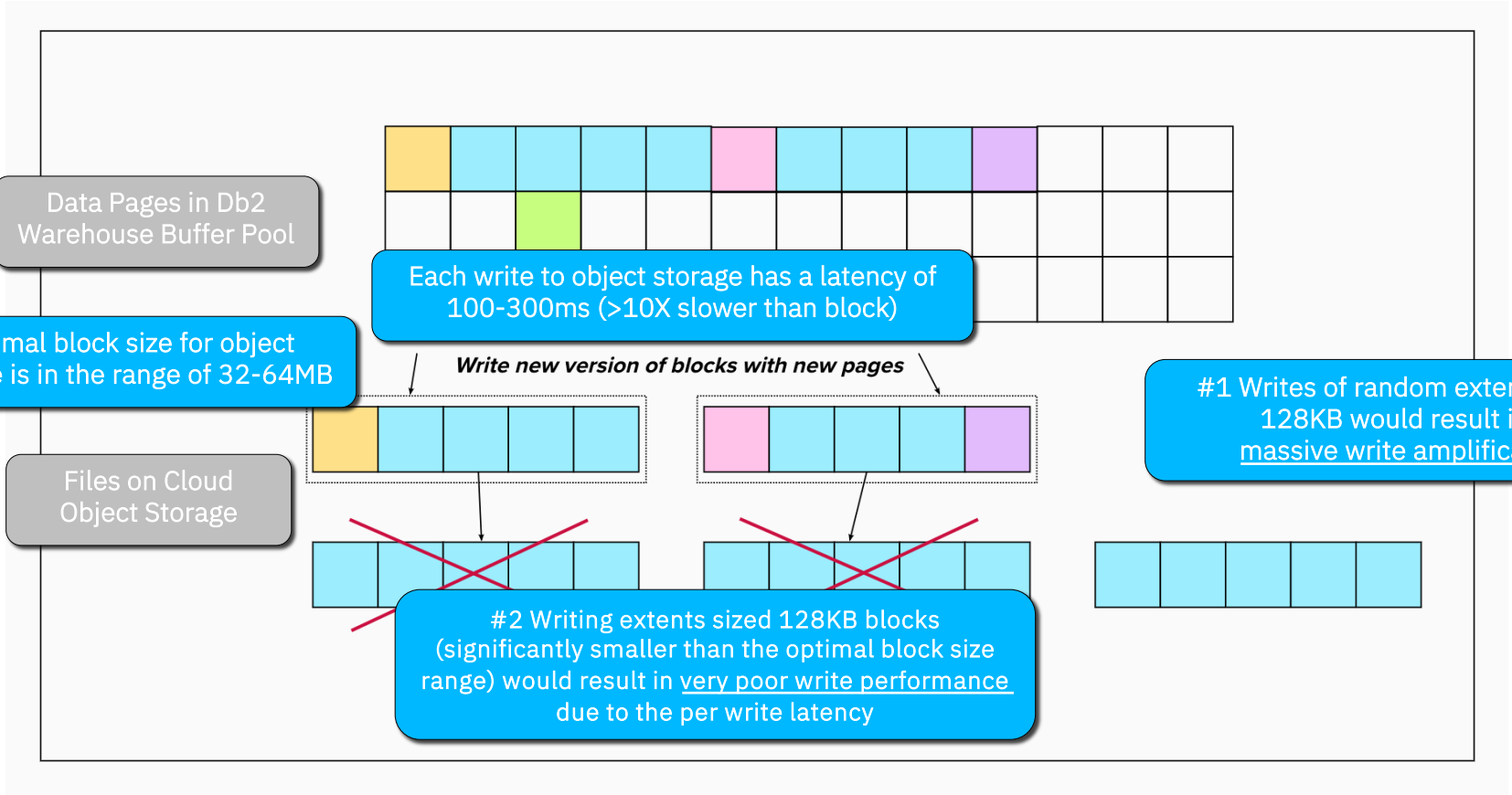
05 User Experience and Out-of-the-box Set up for Native Cloud Object Storage

Native Cloud Object Storage architecture



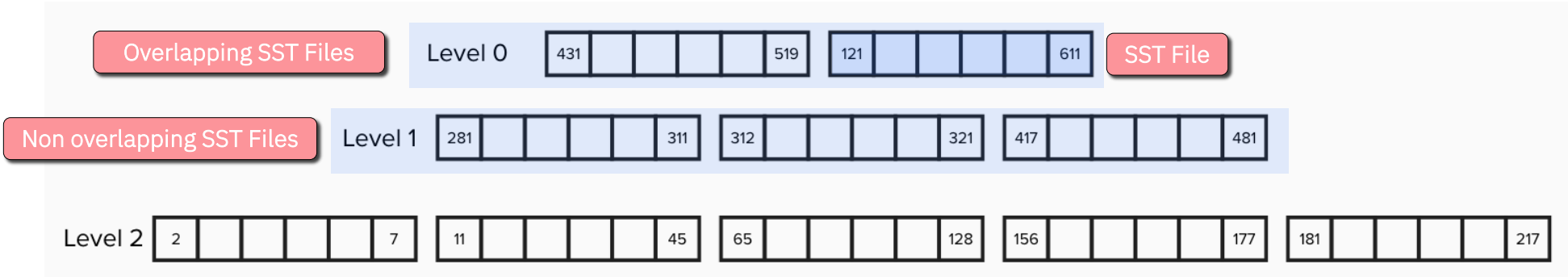
- Existing Db2 component stack down through bufferpool + tablespace layer
- Existing Db2 logging maintains high performance for trickle feed
- Three new elements in new native cloud storage layer:
  1. An LSM tree storage organization to efficiently store Db2 native pages on cloud object storage.
  2. A novel data clustering technique that exploits the self-clustering capabilities of the LSM tree.
  3. A multi-tiered cache that adds a local NVMe component to enable high performance query processing and bulk ingest.

Pitfalls of a naïve storage model



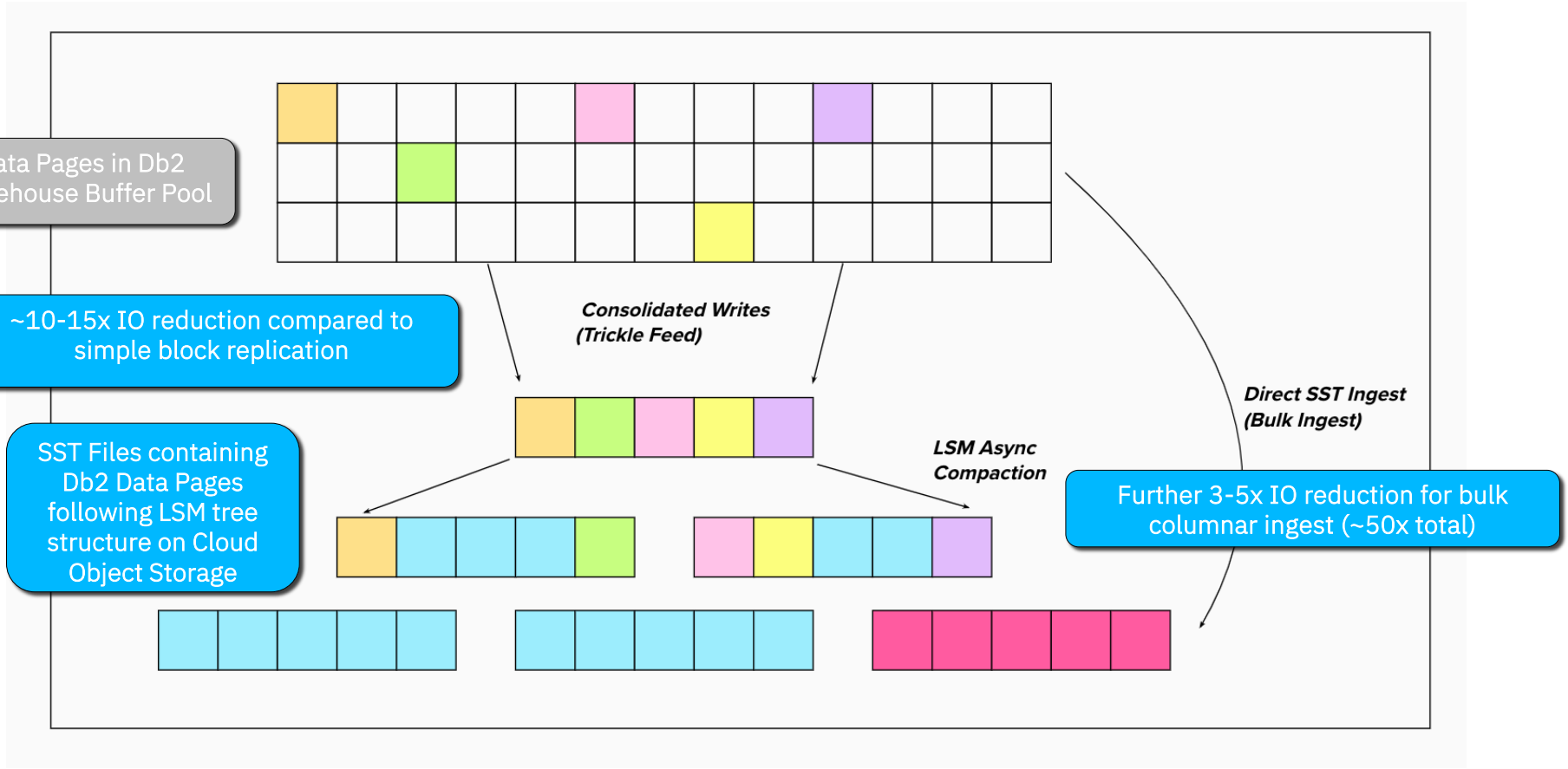
Background On LSM trees

- Log Structured Merge trees (LSM tree) is an index structure designed for on disk low-cost indexing for data with a high insert rate.

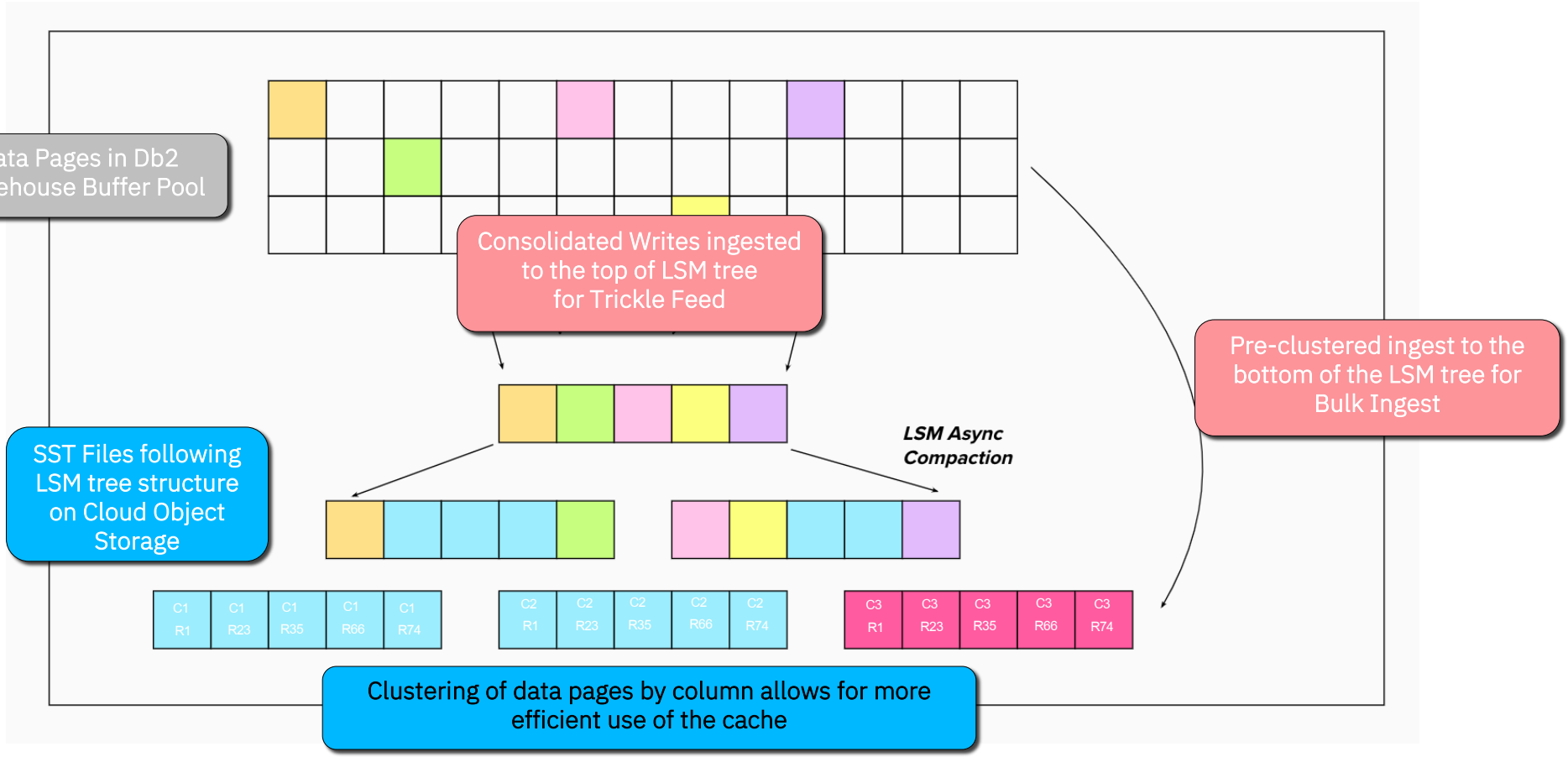


- There are three main characteristics that make it really interesting as a storage model for Db2 Warehouse:
  1. It follows an append-only write mode, where its SST files are only written once, which is ideal for cloud object storage and to simplify cache management.
  2. It is designed for self-optimization, through its background compaction process that moves data through the fully ordered levels.
  3. It is built for a high-volume ingest rate, ideal for data warehouses.

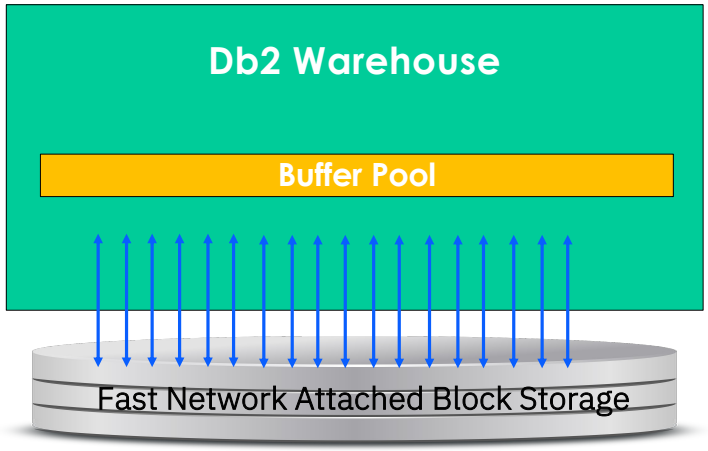
#1 LSM Tree based page IO



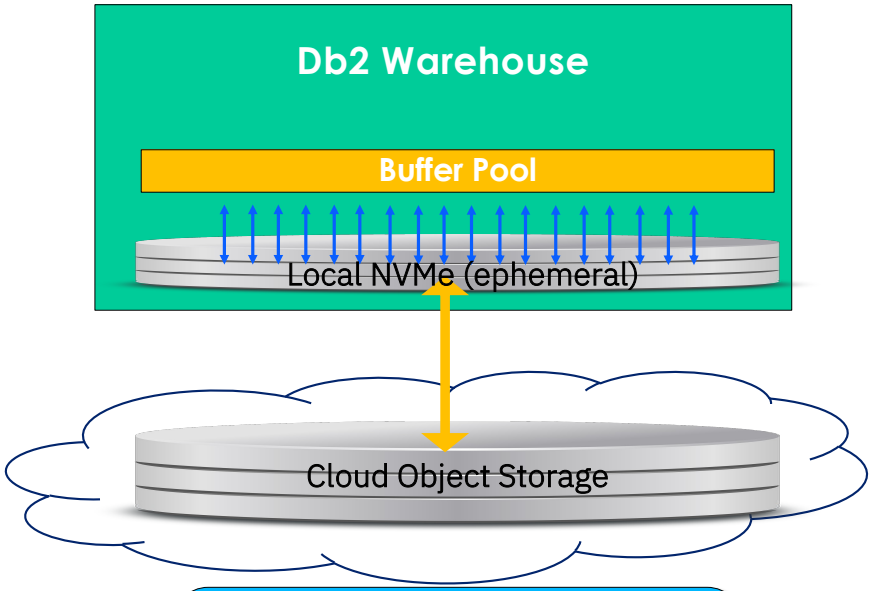
#2 Column Group Clustering within LSM tree



#3 Multi-tiered Cache on Local NVMe drives

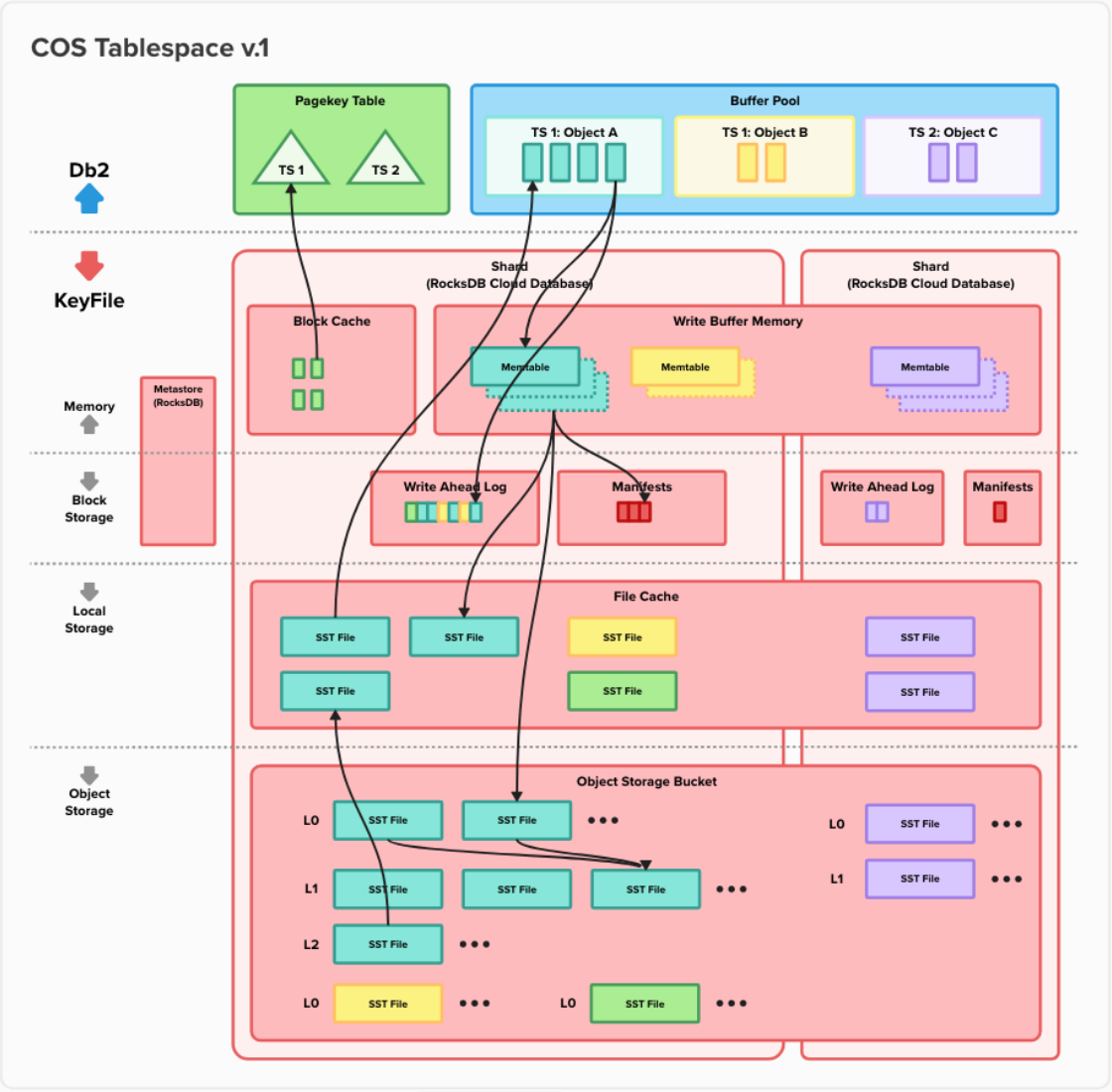


1000 page writes/reads to/from fast network attached block storage @ 10-30ms latency each (6 IOPS/GB)



1000 pages writes/reads to/from local NVME (ephemeral) + 1 PUT/GET to S3 @ 100-300ms latency each

Looking Deeper Under the Hood



Agenda

01 Cloud Object Storage

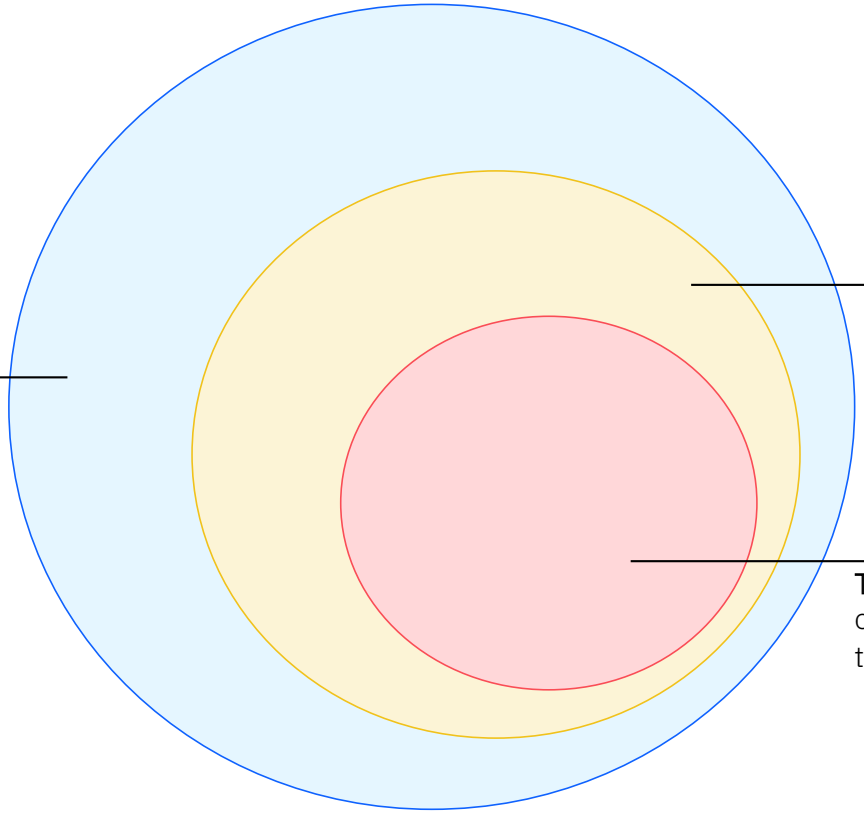
02 Evolution of the Storage Architecture

03 Native Cloud Object Storage Architecture

04 Three reasons for the speed up explained

05 User Experience and Out-of-the-box Set up for Native Cloud Object Storage

Recap of storage hierarchy in Db2



Storage group defines the set of paths where data can be stored

Table spaces are logical storage groupings using storage defined by the storage group

Tables, indexes and large objects are stored within a table space

User Experience with Native Cloud Object Storage Support

1

A remote storage access alias defines an endpoint, path and credentials in cloud object storage.

2

A remote storage group is associated with a remote storage access alias instead of a set of local paths.

3

A remote table space is defined with a remote storage group

4

A column-organized table is created within a remote storage group

User Experience with Native Cloud Object Storage Support in Db2 Warehouse Gen3

1

The remote storage access alias IBMDEFAULTREMALIAS is pre-created using a pre-provisioned AWS S3 bucket.

2

The remote storage group IBMDEFAULTREMSG1 is pre-created.

3

Two remote table spaces OBJSTORESPACE1 and OBJSTORESPACEUTMP1 are pre-created.

4

Tables and DGTTs can be created within the two pre-created remote storage groups for out-of-the-box exploitation of the Native Cloud Object Storage.

Exploring Remote Table Spaces

- To create a column-organized table in the default remote table space, use the following:

```
CREATE TABLE CT1 (c1 INT NOT NULL, c2 INT NOT NULL)
       IN OBJSTORESPACE1
       ORGANIZE BY COLUMN
```

- To create a column-organized Declared Global Temporary table use the following:

```
DECLARE GLOBAL TEMPORARY TABLE GTT1 (c1 int not null, c2 int not null)
    IN OBJSTORESPACEUTMP1
    ORGANIZE BY COLUMN
```

Db2U integration for user-managed environments

Enabling Native COS using Db2UInstance Custom Resource (CR) 1/2

1. Enable Native Cloud Object Storage

```
spec:
   version: s11.5.9.0
   nodes: 2
   addOns:
      advOpts: enableCos: "true"
```

2. Set up the Cloud Object Storage provider if necessary
   - ”aws” for AWS S3 and IBM Cloud Object Storage (default).
   - “self-hosted” for Ceph, MinIO, RHOS Open Data Foundation / ODF

```
spec:
   version: s11.5.9.0
   nodes: 2
   addOns:
      advOpts: enableCos: "true"
      cosProvider: "self-hosted"
```

Db2U integration for user-managed environments

Enabling Native COS using Db2UInstance Custom Resource (CR) 2/2

3. Set up the Local Caching Tier devices
   - The local caching tier requires NVMe drives directly attached to each node for best performance.
   - This NVMe drive is treated as ephemeral, and its contents can be destroyed, if necessary, but result in the need to warm it up. In Db2WHoC this happens automatically on scale-out.
   - The caching tier is configured through Db2U cachingtier setting:

```
storage:
   - name: cachingtier
     spec:
       accessModes:
         - ReadWriteMany
       resources:
         requests:
           storage: 100Gi
       storageClassName: local-device
     type: create
```

Db2U integration for user-managed environments

Setting up Native COS

1. Set up an object storage bucket in the user's cloud object storage provider
   - Db2WHoC: this is pre-provisioned by the cloud infrastructure and configured with role-based authentication and other set up required for backup and restore.

2. Create a remote storage access alias
   - Db2WHoC: IBMDEFAULTREMALIAS is created using AWS role-based authentication.

```
•db2 CALL SYSIBMADM.STORAGE_ACCESS_ALIAS.CATALOG( 'IBMDEFAULTREMALIAS', 'S3',
's3.amazonaws.com', '<user name>', '<password>', 'db2wh-instance1', 'sg00', 'I', '')
```

3. Create a remote storage group associated with the remote storage access alias
   - Db2WHoC: IBMDEFAULTSG1 is created under IBMDEFAULTREMALIAS.

```
•db2 CREATE STOGROUP IBMDEFAULTREMSG1 ON 'DB2REMOTE://IBMDEFAULTREMALIAS/'
```

4. Create remote table spaces using the remote storage group.

```
•db2 CREATE TABLESPACE OBJSTORESPACE1 USING STOGROUP IBMDEFAULTREMSG1
•db2 CREATE USER TEMPORARY TABLESPACE OBJSTORESPACEUTMP1 USING STOGROUP IBMDEFAULTREMSG1
```

Monitoring Remote Table Spaces

The remote table spaces are the only table spaces that have the CACHING_TIER column set to ENABLED.

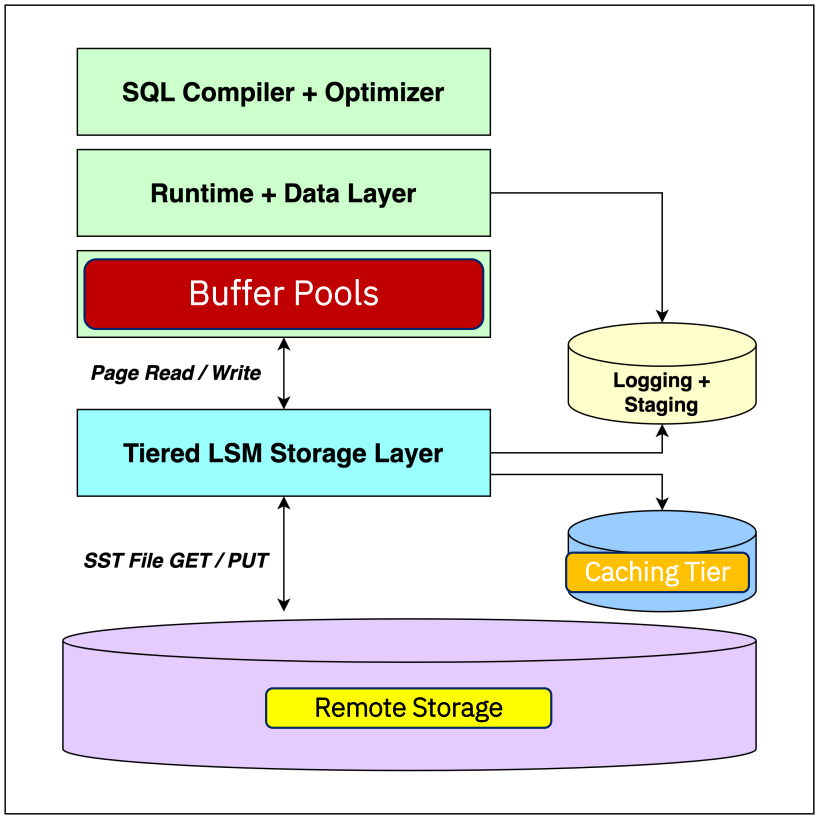
```
SELECT VARCHAR(TBSP_NAME, 30) AS TBSP_NAME,
       MEMBER,
       TBSP_TYPE,
       CACHING_TIER
FROM TABLE(MON_GET_TABLESPACE('',-2)) AS T

TBSP_NAME                      MEMBER TBSP_TYPE  CACHING_TIER
------------------------------ ------ ---------- ------------
…
OBJSTORESPACE1                      0 DMS        ENABLED
OBJSTORESPACEUTMP1                  0 DMS        ENABLED
…
```

Monitoring Remote Table Spaces: Reads

The Native COS read storage hierarchy has 3 levels:

1. A set of buffer pools for in-memory caching of data pages, shared between remote table spaces and non-remote table spaces.
2. A caching tier layer backed by fast locally-attached NVMe drives, for the extended local caching to maintain a larger working set than in-memory and to amortize the cost of accessing remote storage.
   - Note: WAL is not monitored for READS
3. A remote storage layer, in Cloud Object Storage, when reading data pages are not currently cached in either of the two caching layers.



Monitoring Remote Table Spaces: Reads

New monitoring elements were added or changed to expose the additional layers in the storage hierarchy.

Two pairs of examples:

- POOL_COL_LBP_PAGES_FOUND: number of pages read (found) in BP.
- POOL_COL_CACHING_TIER_PAGES_FOUND: number of pages read (found) in caching tier.
- POOL_COL_P_READS: number of pages read from remote storage.

- DIRECT_READ_TIME: this is time spent on direct access to the remote storage, excluding the caching tier.
- CACHING_TIER_DIRECT_READ_TIME: For remote containers, this is the elapsed time in milliseconds required to perform the direct reads serviced using the caching tier.

Monitoring Remote Table Spaces: Reads

Caching tier hit ratios expose the efficiency of the caching tier, for example:

- CACHING_TIER_DATA_HIT_RATIO_PERCENT: for pages that were found in the caching tier without needing to get them from remote storage.

As usual with cache hit ratios, the higher the ratio the better the cache efficiency.

```
SELECT VARCHAR(TBSP_NAME, 30) AS TBSP_NAME,
       MEMBER,
       CACHING_TIER_DATA_HIT_RATIO_PERCENT
FROM SYSIBMADM.MON_TBSP_UTILIZATION

TBSP_NAME                      MEMBER CACHING_TIER_DATA_HIT_RATIO_PERCENT
------------------------------ ------ ----------------------------------
…
OBJSTORESPACE1                      0                             100.00
OBJSTORESPACEUTMP1                  0                             100.00
…
```

Q&A

Christian Garcia-Arellano

STSM, Db2 OLTP Architect and Master Inventor

cmgarcia@ca.ibm.com

Thank You

Christian Garcia-Arellano

STSM, Db2 OLTP Architect and Master Inventor

cmgarcia@ca.ibm.com