Exploring a Path towards Memory-Storage Convergence

Daniel Waddington (daniel.waddington@ibm.com)
Research Staff Member

IBM Research Almaden

Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

- CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR
- ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Memory/Storage Tiers

*random read aggregate

Durable			Programmable	
Cloud / Database	HPC/Enterprise Network-Attached Storage	Local NVMe Storage Array	DDR Memory (Intel Cascade Lake)	CPU L1 Cache
600 MB/s	100 Gbps (12.5 GB/s)	50 GB/s	200 GB/s	> 10 TB/s
> 10ms	10-20 µsec	< 10 µsec	< 0.1 µsec	1ns

500x 2x 100x 100x

A New Breed of Storage Class Memory

Intel/Micron 3D-Xpoint is the first-in-breed “Storage Class Memory” designed for the enterprise storage domain

3DXP is based on a lattice-arranged Phase-Change-Memory (PCM)

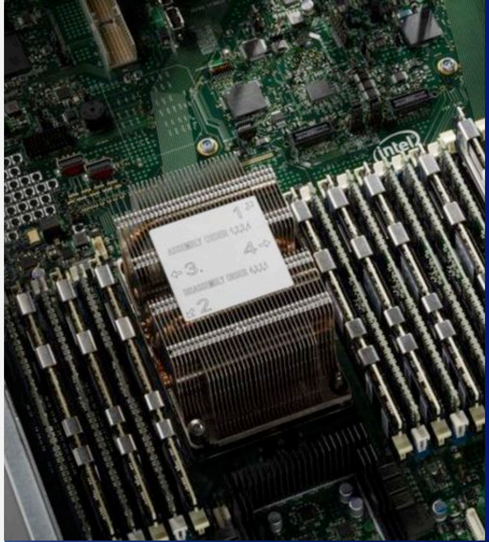
Intel product offerings (under Optane trademark) in NVMe SSD and NVDIMM space

Intel Optane DC *Persistent Memory* Modules offers DIMM modules that is attached to the memory-bus

~3x Slower than DRAM

8x capacity than DRAM, at ½ cost per GB

Expect 12TB capacity in 2U by 2020



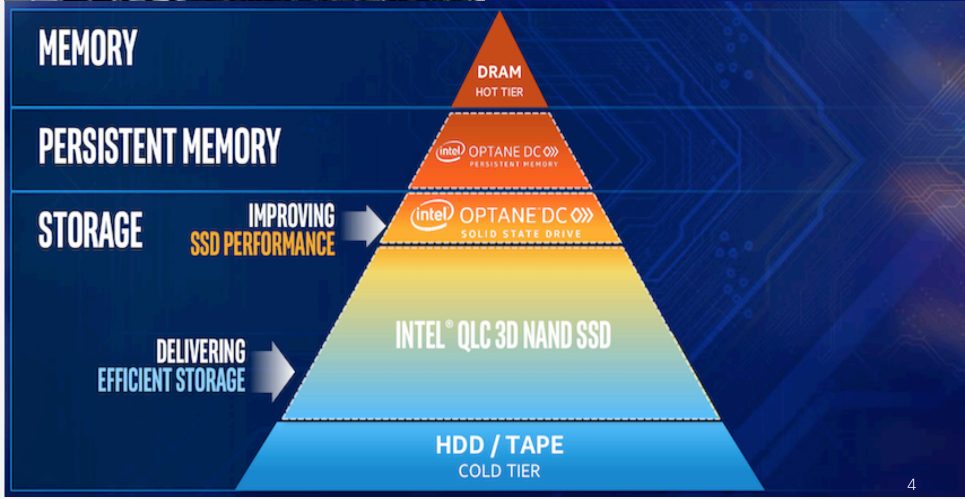
Optane DC PMM:

Load/store addressable (64B)

Up to 512GB DIMMs providing 6TB in 2U

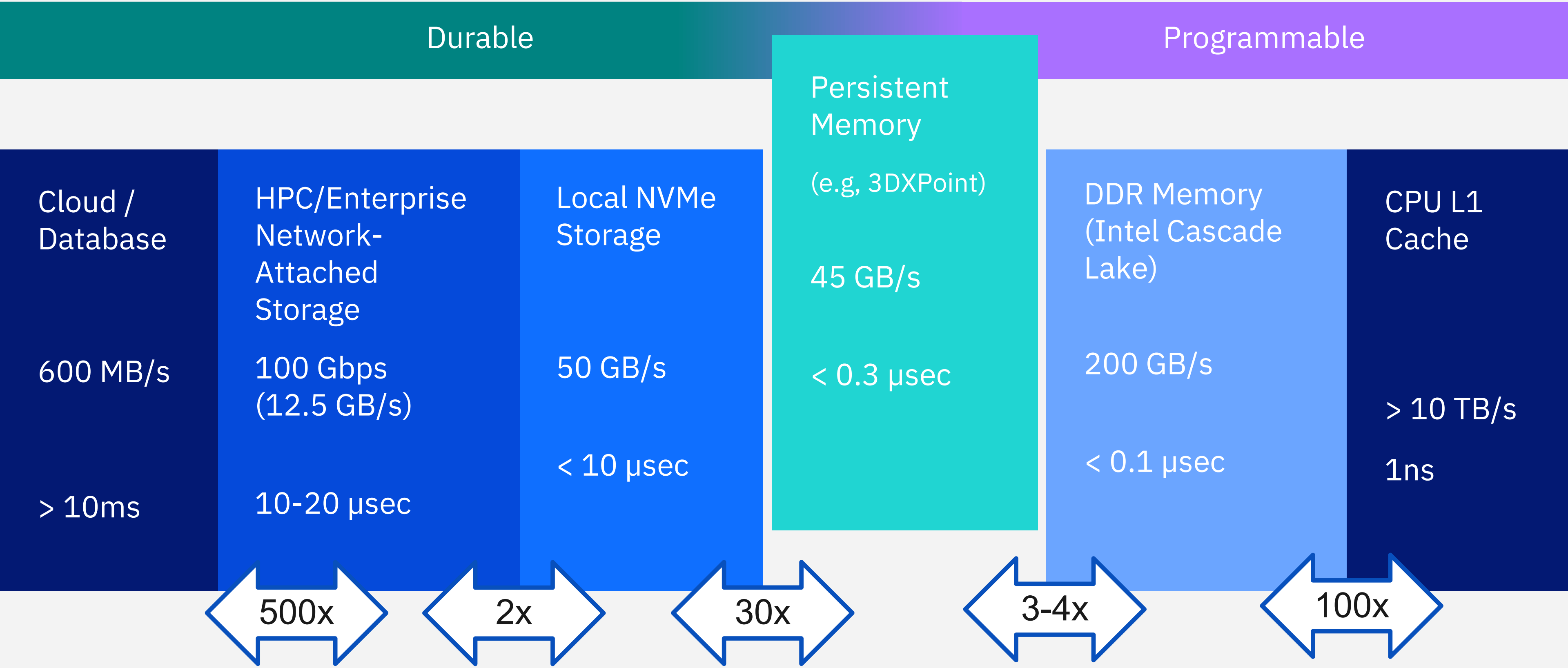
150-300ns access latency (64B)

56GB/s RR, 20GB/s RW



Storage Class Memory

*random read aggregate



Durable

Programmable

Cloud / Database

600 MB/s

> 10ms

HPC/Enterprise Network-Attached Storage

100 Gbps (12.5 GB/s)

10-20 µsec

Local NVMe Storage

50 GB/s

< 10 µsec

Persistent Memory

(e.g, 3DXPoint)

45 GB/s

< 0.3 µsec

DDR Memory (Intel Cascade Lake)

200 GB/s

< 0.1 µsec

CPU L1 Cache

> 10 TB/s

1ns

Changing the Landscape

How does Persistent Memory change our view of the world?

Bringing data closer to the CPU

Data intensive applications with unpredictable access patterns

Data in-memory is now durable against s/w crash and reset events

Reduced recovery data

Near-DRAM low latency access for synchronous CPU load/store

x8-16 capacity and lower cost than DRAM

Potential for *unified* compute and storage data models

Seamless data movement

Memory-Storage Convergence Vision

Durable and programmable data domains the same thing

Unified security protection scheme (i.e. process/user)

Eliminate the need to transform data for storage

Remove need for file and block abstractions?

RDMA/DMA engines allow fast CPU-bypass transfers of data

Convergence Challenges

Different programming languages have different data models

Efficient support for relocatable data structures

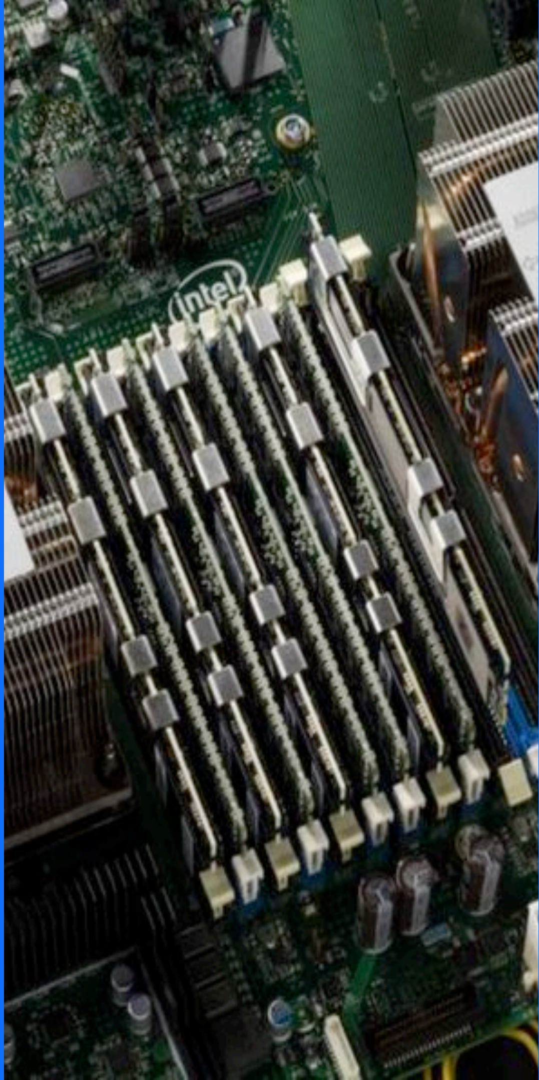
Achieving crash-consistency and transactional boundaries for legacy data structure

Realizing conventional storage services, encryption, de-dupe, erasure coding, versioning, snapshots …

Minimizing the overhead of crash-consistency

Reliability and Serviceability

Data recovery



Road Ahead

Accelerated devices

- 5+ GB/s SSD
- 400Gbps+ network
- In-storage compute (e.g. Samsung SmartSSD)
- In-network compute (e.g. Mellanox Bluefield)

Direct-to-memory accelerators

- PCIe 4.0 and OpenCAPI
- FPGA accelerator cards
- Peer-to-peer device DMA
- Digital and analog AI cores

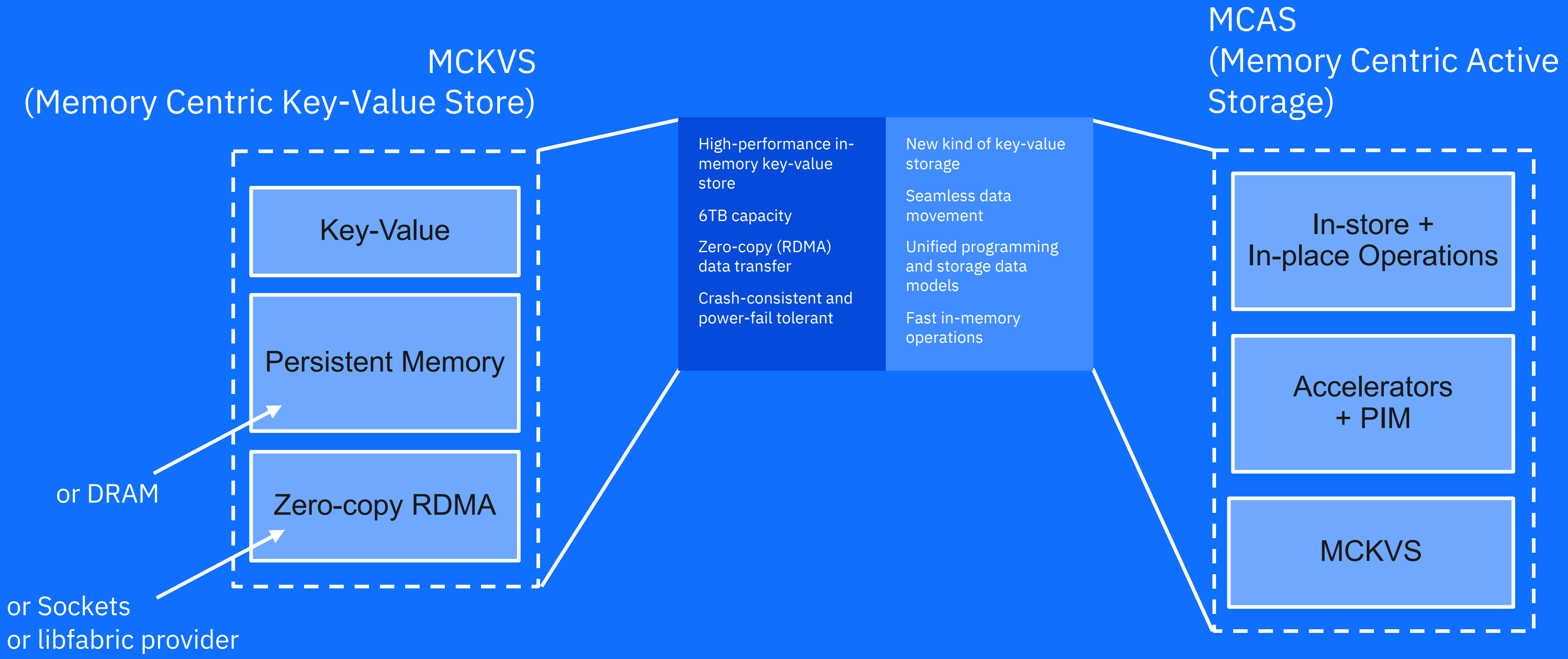
Near-memory accelerators

- CPU domain-specific instructions (e.g., low precision)
- Programmable DIMM-embedded Data Processing Units (e.g., UPMEM)

Processing In-memory

- Phase Change Memory-based embedded logic (e.g. Memristor-Aided Logic MAGIC, FLEX)

IBM Research MCAS Project



MCKVS
(Memory Centric Key-Value Store)

Key-Value

Persistent Memory

Zero-copy RDMA

or DRAM

or Sockets
or libfabric provider

- High-performance in-memory key-value store
- 6TB capacity
- Zero-copy (RDMA) data transfer
- Crash-consistent and power-fail tolerant

- New kind of key-value storage
- Seamless data movement
- Unified programming and storage data models
- Fast in-memory operations

MCAS
(Memory Centric Active Storage)

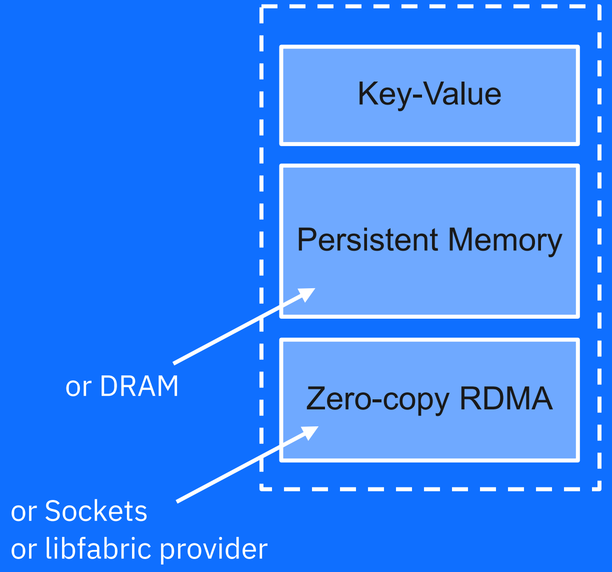
In-store +
In-place Operations

Accelerators
+ PIM

MCKVS

MCKVS

MCKVS
(Memory Centric Key-Value Store)

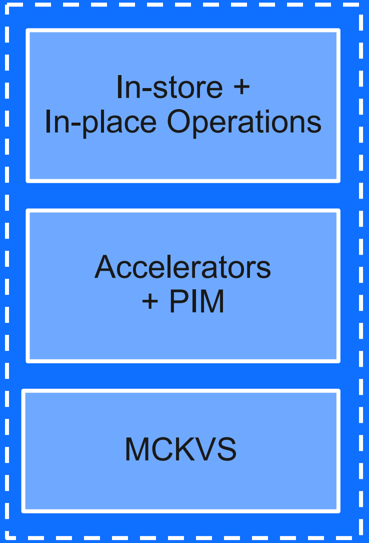


Current prototype

- 20M IOPS (small random gets)
- 10M IOPS (small random puts)
- ~7us round-trip latency synchronous put
- Tested on 100GbE RDMA
- 20GB/s + throughput for larger transfers (2xNICs)
- Based on custom crash-consistent hash table
- GPU-Direct capable
- Does NOT use PMDK (due to RDMA integration)
- C++ and basic Python APIs
- Can be deployed in containers/k8 or VMs
- Open Source (building community)
- Not secure (yet)
- Available at https://github.com/IBM/mcas/

Evolution to MCAS

MCAS
(Memory Centric Active Storage)



In-store +
In-place Operations

Accelerators
+ PIM

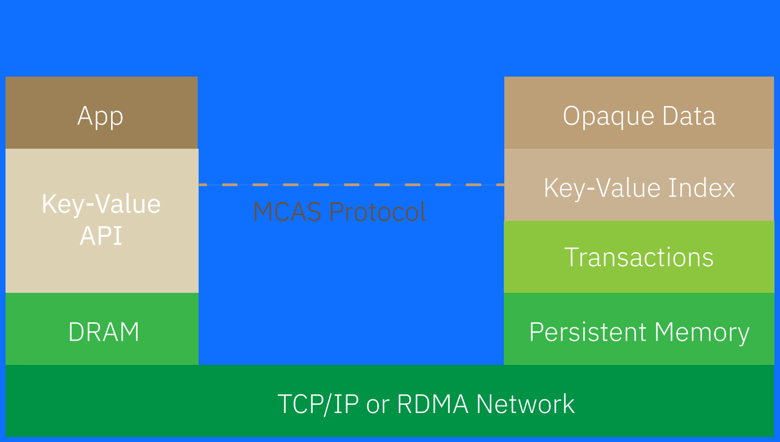
MCKVS

Prototype in development

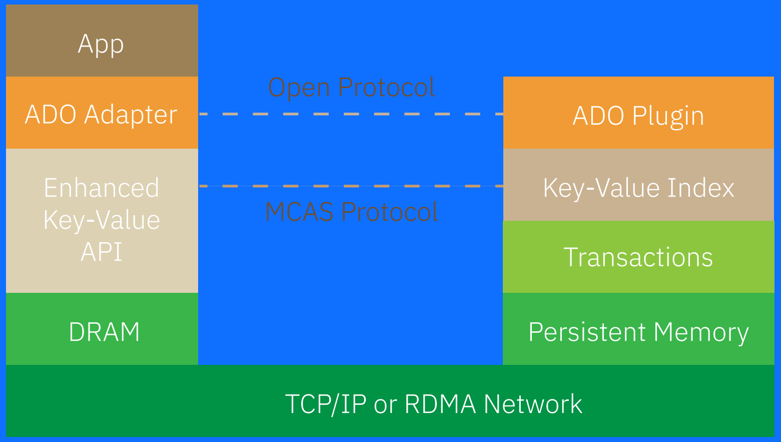
- MCKVS as the core
- Shift to “structured document” key-value store
- Flexible to support arbitrary data structures
- Flexible to support either flat (e.g., JSON) or pointer-based data models (e.g., C++ data structures)
- Designed to allow user-written operations to be safely deployed in the system
- Open protocol allows custom layering of services (e.g., versioning, encryption, logging, data conversion, data summarization, ...)
- Aimed at allowing accelerators based on HW (e.g., FPGA, AI core, Processing-in-Memory)
- Initial release expected 1Q2020

MCAS

Defines an open architecture for layering “services” on top of a in-memory key-value store - more performance, less data movement



Plain key-value

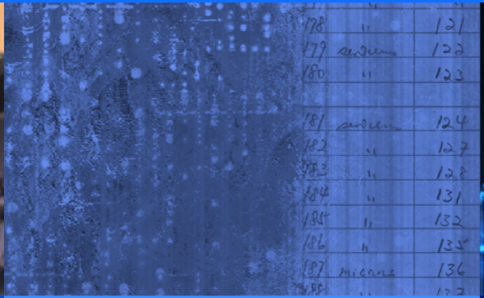


Active data services

Example Use-Cases

Storage Services

- versioning/TTL
- encryption/CRC
- logging
- durability
- event notification



Data Curation

- sorting, filtering
- EDI transform
- real-time compression and decompression
- summary operators

Real-time Data Analytics

- domain specific data structures (e.g., k-d tree)
- core math operations (e.g., mat mul , fft)

Real-time Cognitive

- graph processing
- inferencing
- sparse distributed memory / HTM
- NLP

IBM