LA-UR-24-26760

Toward Open, Standardized Object-Based Computational Storage for Big Data Analytics

Woosuk Chung, Director, Memory Systems Research, SK Hynix Inc.

Qing Zheng, Los Alamos National Laboratory

the Future of Memory and Storage

Legal Disclaimer

- The information contained in this document is claimed as property of SK hynix. It is provided with the understanding that SK hynix assumes no liability, and the contents are provided under strict confidentiality.

- This document is for general guidance on matters of interest only. Accordingly, the information herein should not be used as a substitute for consultation or any other professional advice and services.

- SK hynix may have copyrights and intellectual property right. The furnishing of document and information disclosure should be strictly prohibited.

- SK hynix has right to make changes to dates, product descriptions, figures, and plans referenced in this document at any time. Therefore the information herein is subject to change without notice.

Overview

- Goal
  - Rapid insight generation
- Problem
  - Scientific analysis often reads more data than is necessary
- Approach
  - Execute queries closer to data using object-based computational storage
  - Embrace open community analytical database efforts (Spark, Arrow, Parquet, Substrait, DuckDB, Iceberg, etc)

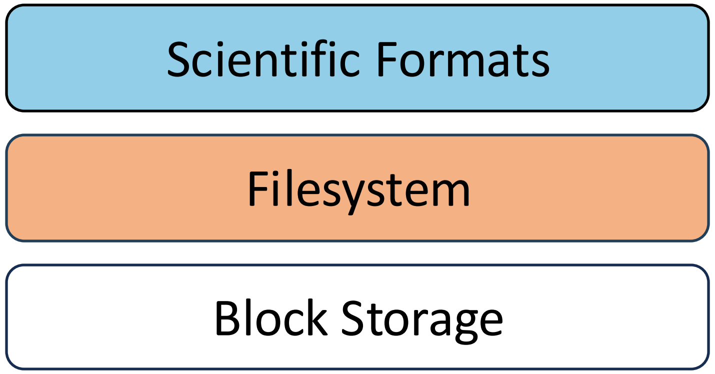
OCS

Object-based Computational Storage

An effort in exploring a standards-based approach to utilizing all levels of storage to process and accelerate queries

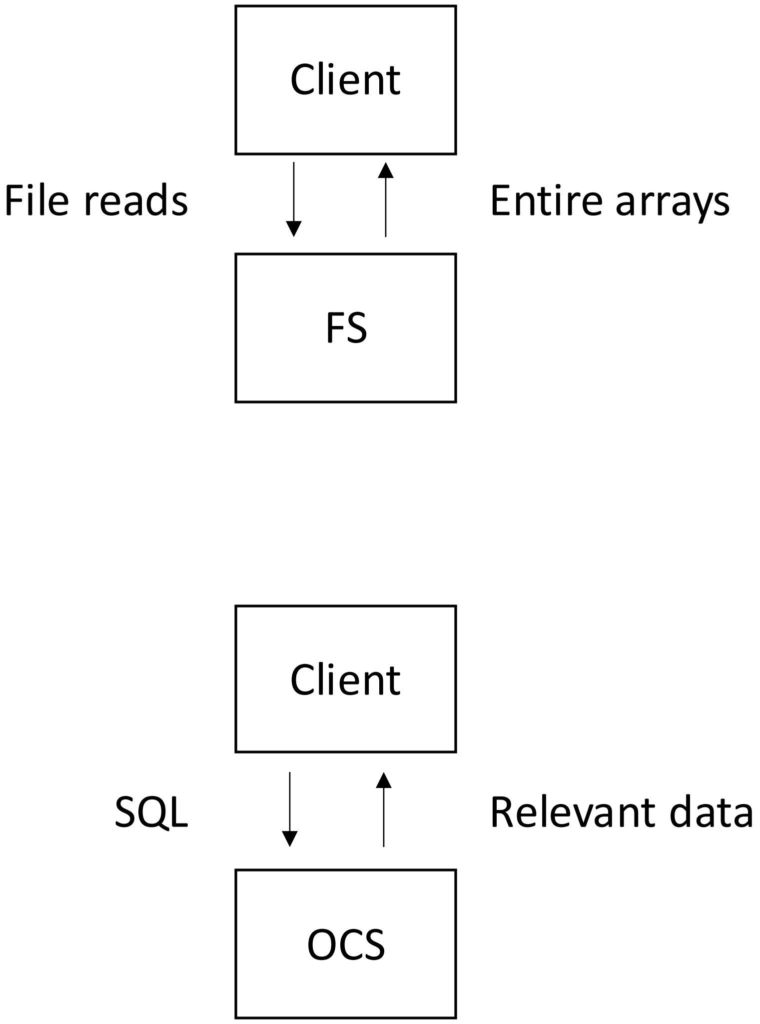
Background: Scientific Storage I/O Stack

- Filesystems over blocks
- Popular data formats: VTK, HDF5, …
  - Self-describing
  - Columnar (data arrays)
  - Geometry data (points, cells)
- Limited support for selective data retrieval
  - Allow retrieving data by array offsets
  - Does not support retrieval by predicates (e.g.: values larger than 0.3)



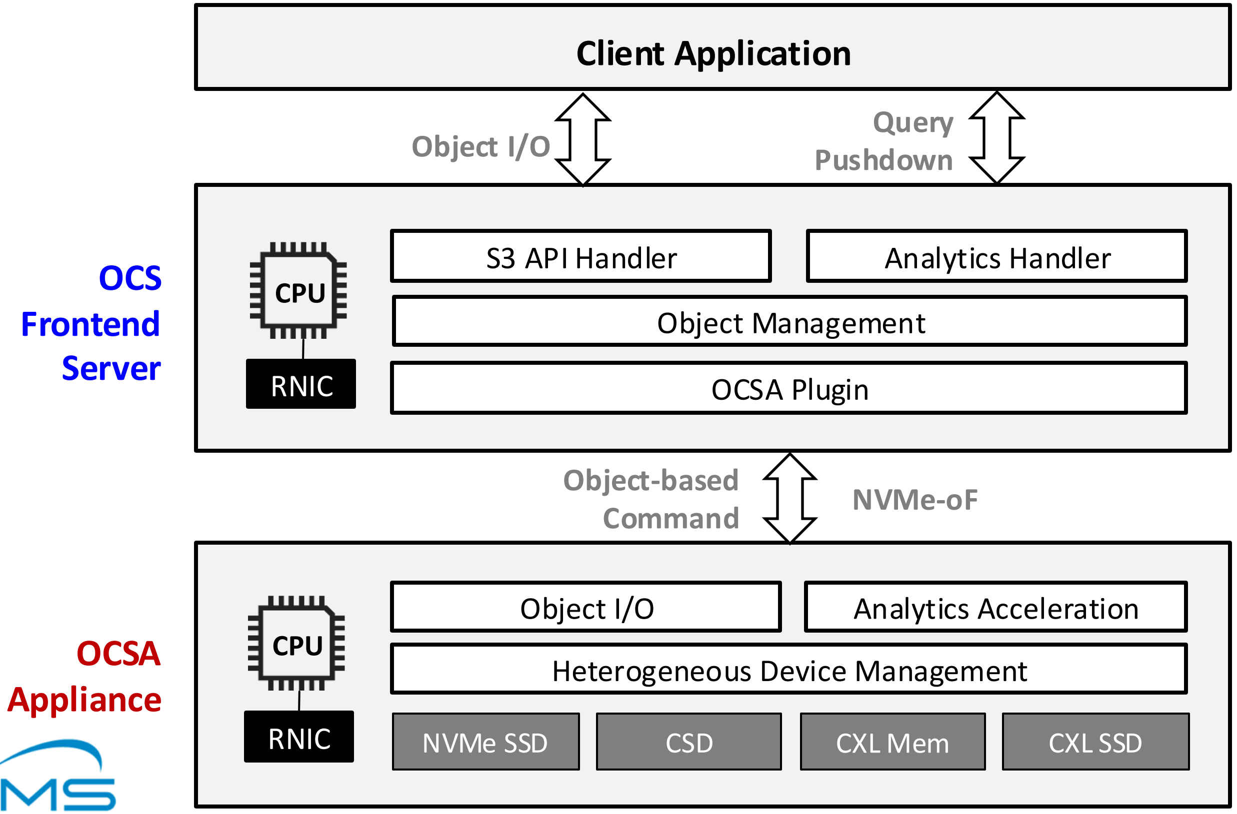
How OCS Makes A Difference

- SQL as the query interface
  - Allow complex queries with predicates
- Query processing offloaded/distributed along the storage I/O path
  - SSDs, storage arrays, servers, clients, ...
  - Minimal data movement
  - Rapid queries
- A standards-based approach
  - SQL/substrait, Parquet, object-based query pushdown API (new), NVMe command sets (new)



OCS System: Architecture

- OCS system consists of OCS frontend server and Object-based Computational Storage Array (OCSA)
  - Disaggregates back-end storage (OCSA) through NVMe over Fabrics
- OCS is vertically optimized distributed analytics system
  - Supports standard-based object interface and analytics features

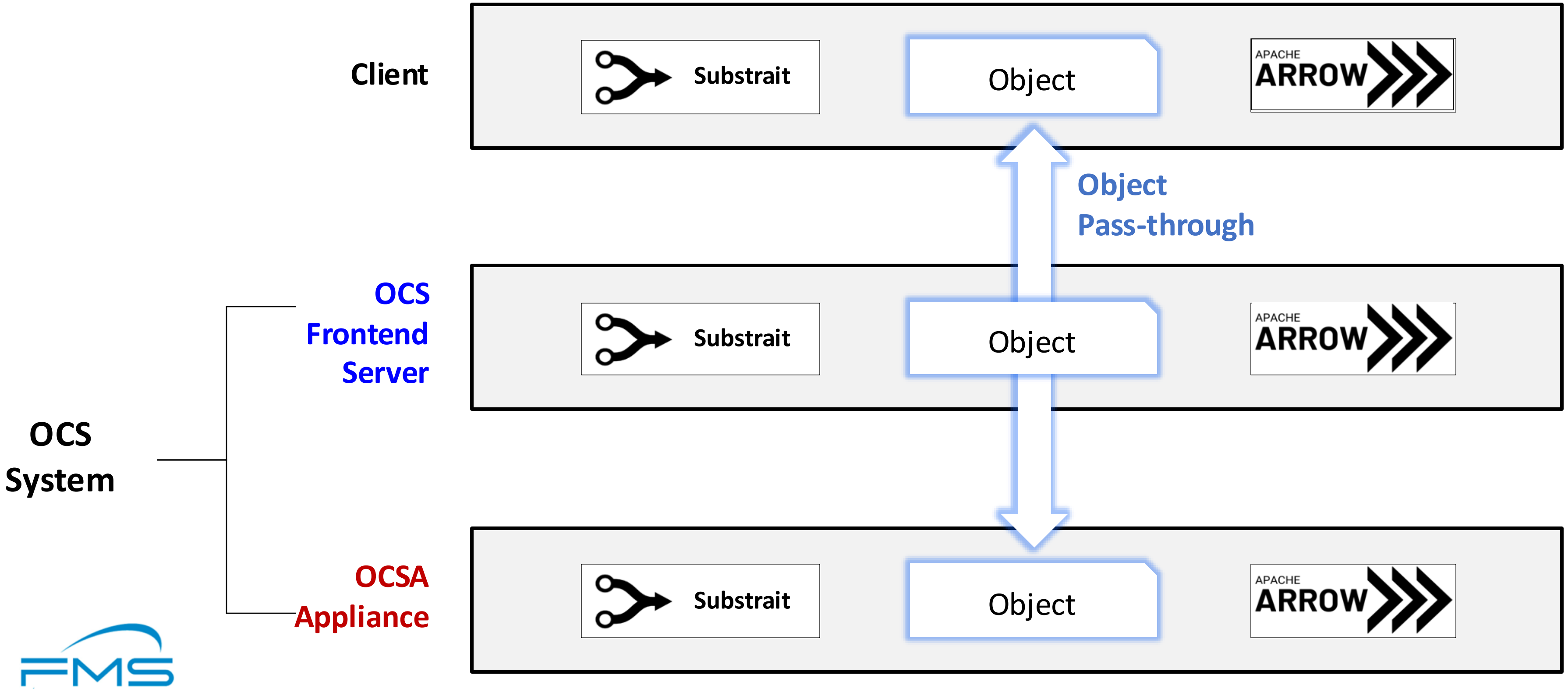


- **Support S3 REST API & open format data**
- **Support query pushdown to OCSA & interface**

- **Object-based NVMe-of command (I/O, Analytics)**
- **Filtering or query processing with its own analytics engine**
- **Utilize heterogeneous devices (plan)**

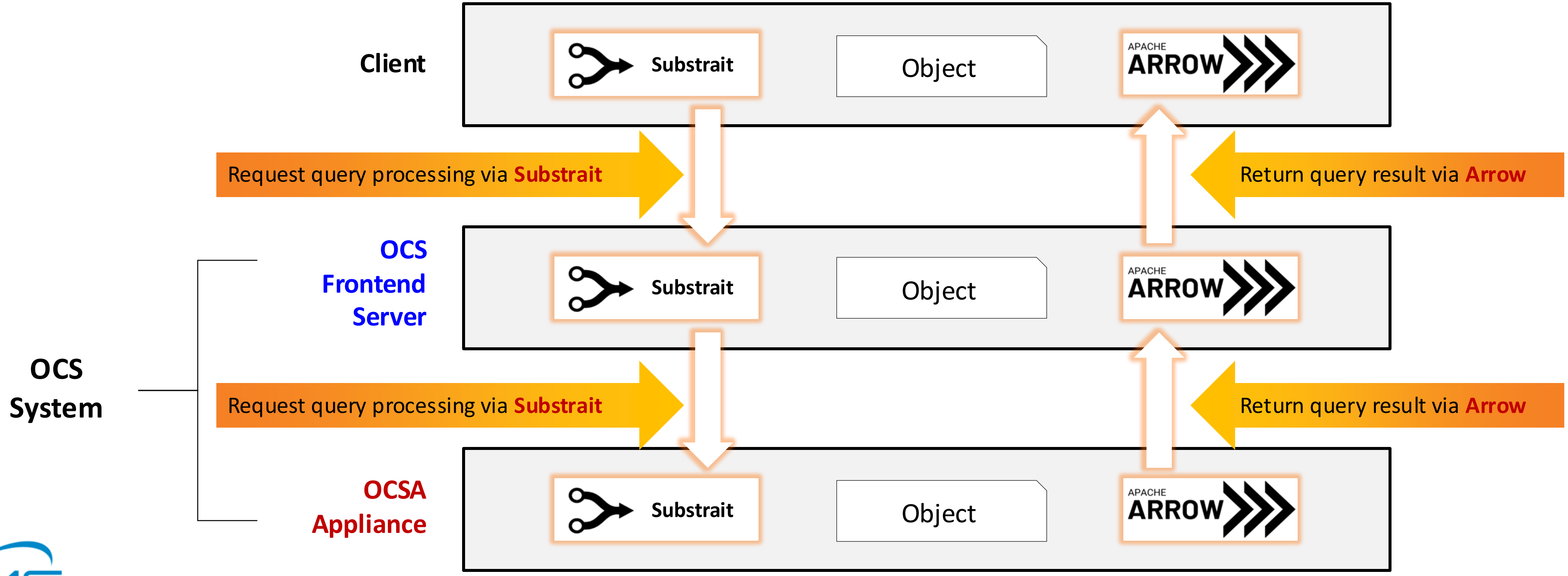
Key Features: Everywhere Compute w/ Object

- OCS supports a object store across all systems and it can provide consistent computational viewpoint everywhere (Client, Storage server, Storage device)
  - Object enables the same view of “analytics chunk” everywhere in the system
  - Data can be processed as the same object view anywhere that query is offloaded



Key Features: Everywhere Compute w/ Analytics Eco.

- Apache Arrow and Substrait enables consistent data analytics processing everywhere
  - Substrait is common method & interface to pushdown operators and it can support query pushdown
  - Apache Arrow is common data format to remove data transformation between different systems
  - OCS is aligned with analytics community Trends



Expected Benefits

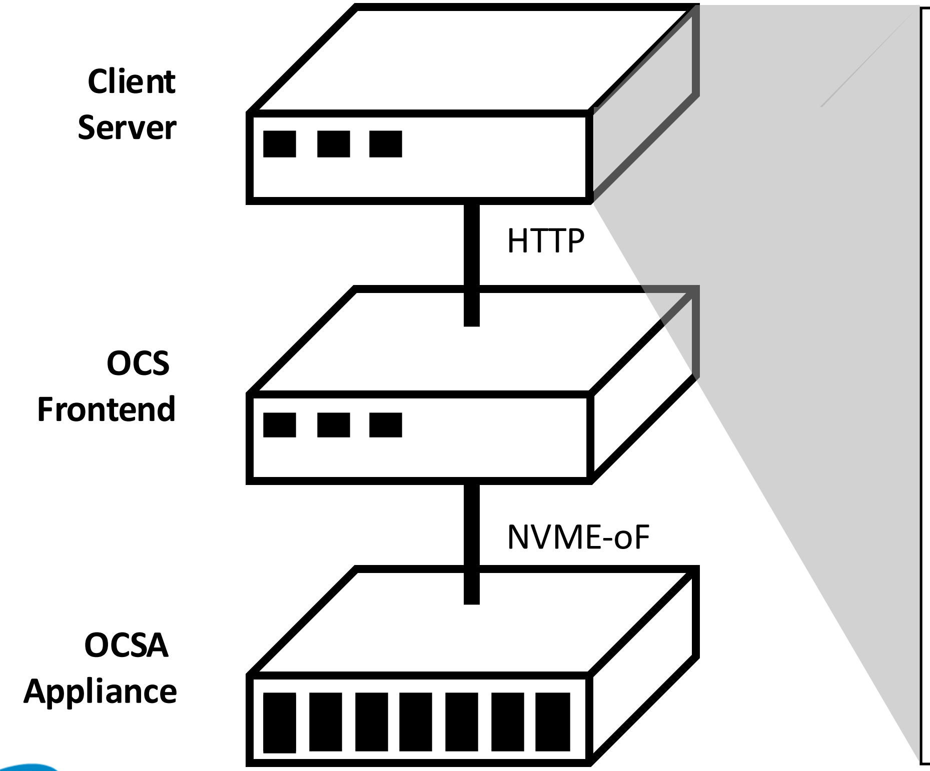
- Reduce data movement between a host system and storage system
  - Performs data analytics where data reside
  - Efficient data movement increase data analytics performance

- Save host system resource
  - Push-down query execution to OCS system, can save host CPU & memory resources

- High interoperability and flexibility
  - Based on standard data format & interface(Apache Arrow & Substrait)
  - Break away from fixed and limited pushdown functionality and allow complex queries with predicates

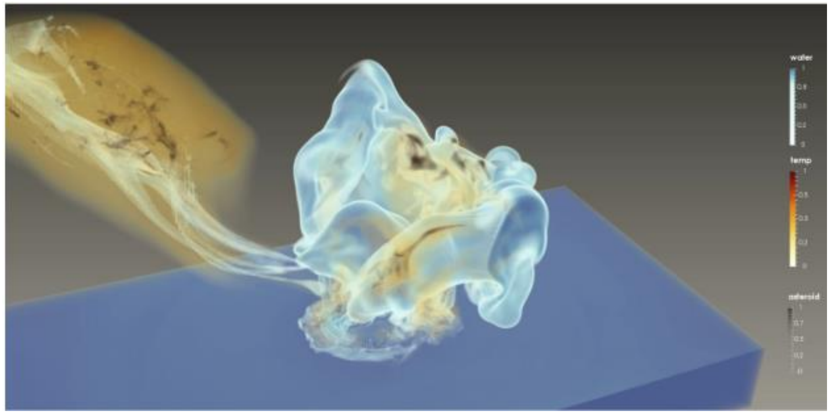
Demonstration: Environments

- Compare performance between **“Base analytics”** and **“OCS Analytics”**
  - Base Analytics (Legacy): Read object data and perform filtering in the client’s analytics engines
  - OCS Analytics: Request query using Substrait to OCS system and get the filtered data



Deep Water Asteroid Impact Simulation

- Real-world scientific simulation
- Select and visualize the factors affecting Earth (Selectivity: 0~18%)



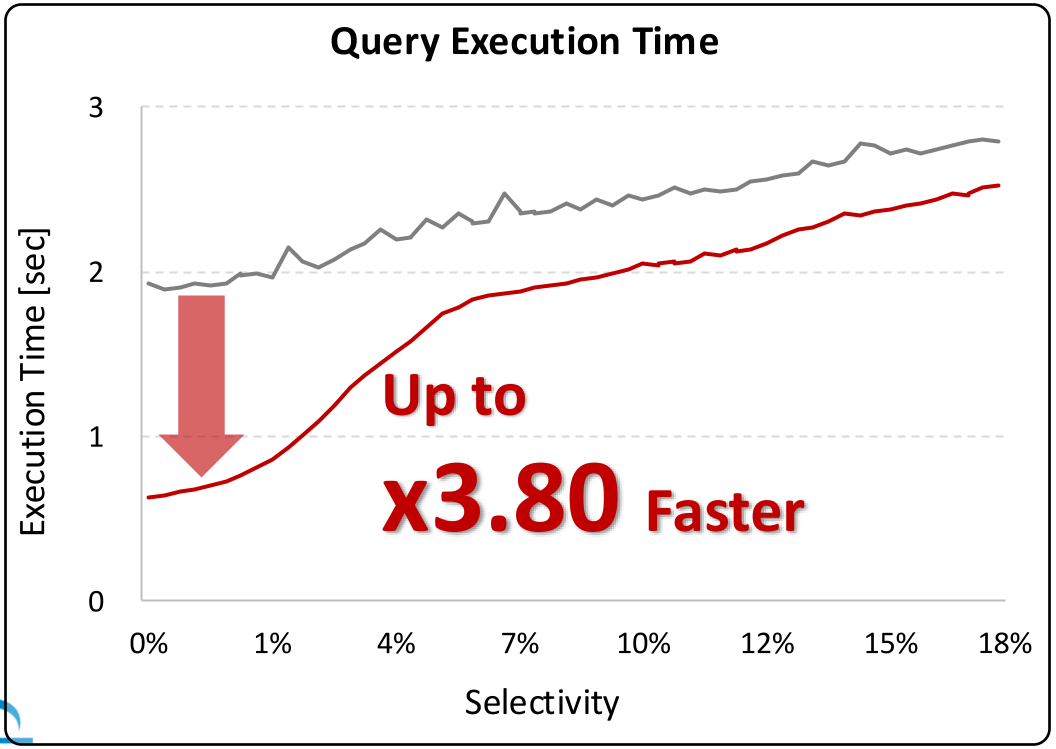
Parquet

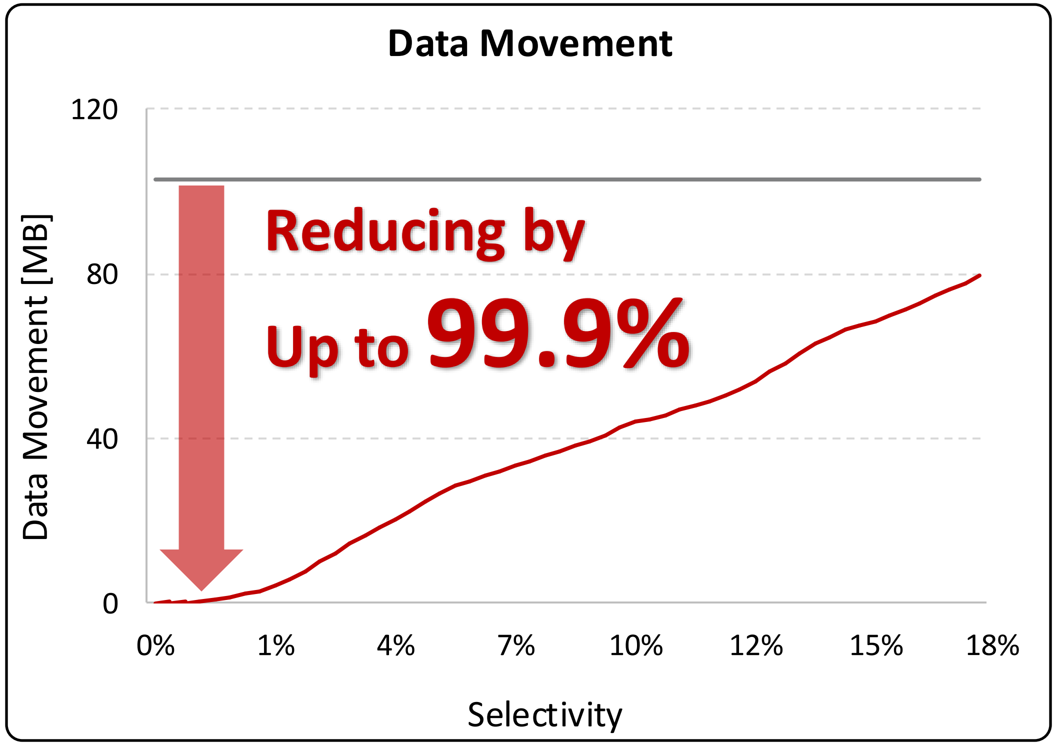
Dataset :

- Apache Parquet Format
- 64 Parquet Files
- Approx. 200MB

Demonstration: Evaluation Result

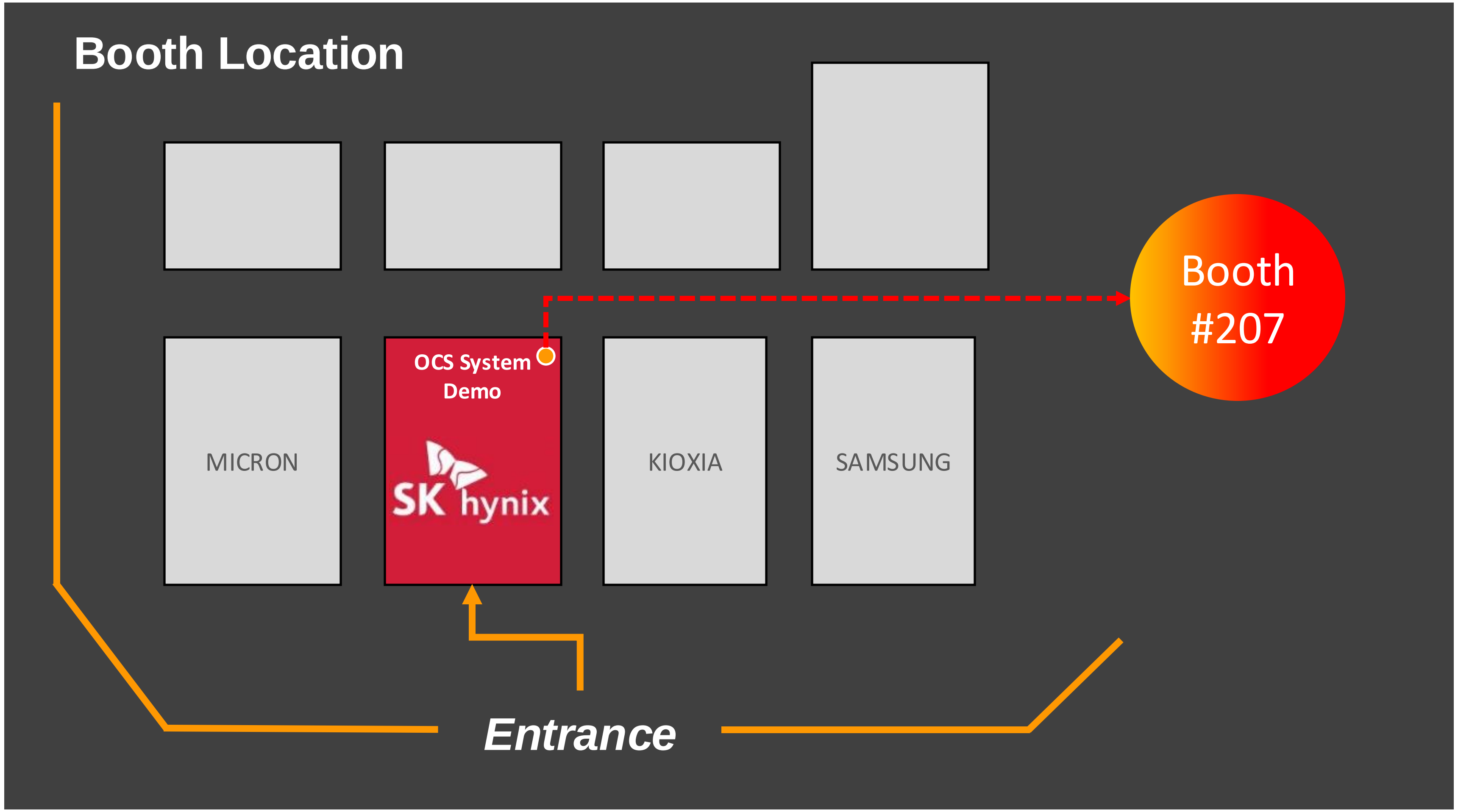
- Enhancing data analytics performance up to 3.8 times
  - Higher selectivity, greater the increase performance
- Removing data movement up to 99.9%
  - Reduce computing and network overhead caused by unnecessary data transfer
- Reducing CPU utilization of client server up to 98%





Base Analytics (Legacy) OCS Analytics

Learn more about SK hynix



Visit Booth #207 and Experience SK hynix products and demos

Thank you