Caching Framework for Exabyte-Scale Data Lakes

(COMMUNITY OVER CODE 2023)

Jing Zhao, Uber
Hope Wang, Alluxio

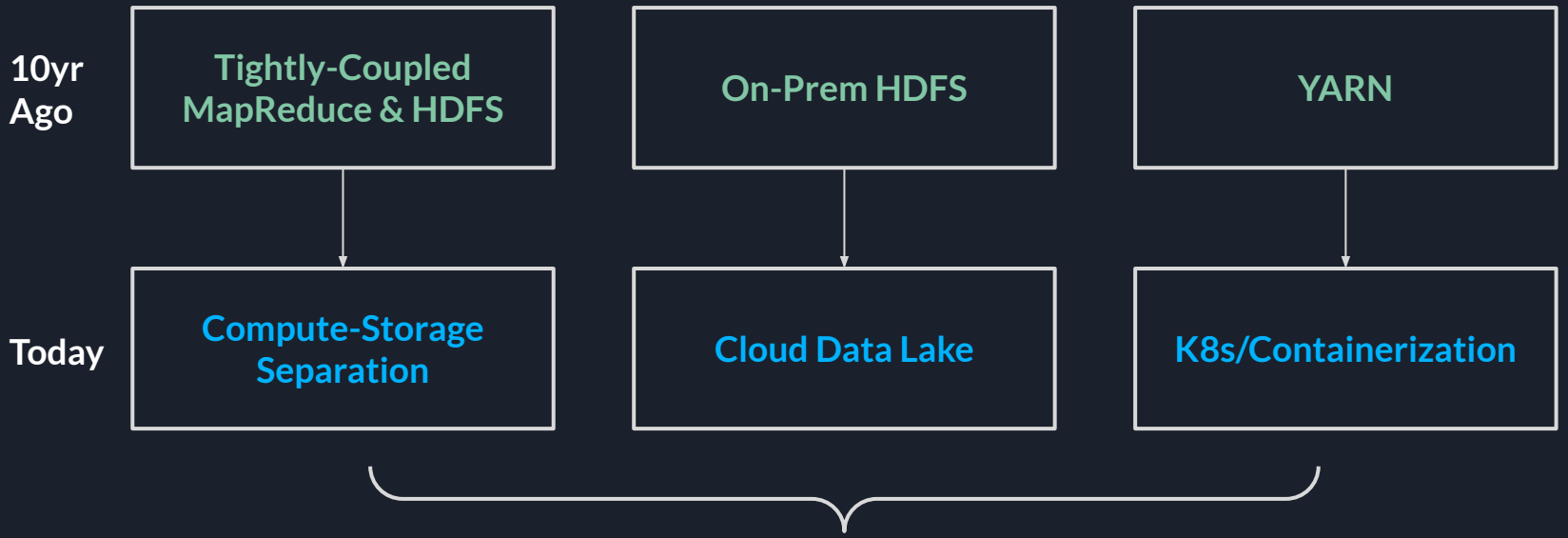
Uber

ALLUXIO

Agenda

- Data locality challenges in large-scale data lakes
- Alluxio: An open-source caching framework
  - Improve both performance and cost efficiency
  - Distributed cache and embedded cache
- Uber: Using cache for exabyte-scale data lake
  - HDFS DataNode local cache for high-density HDD adoption
  - Presto local cache

The Evolution of the Modern Data Stack



More Elastic, Cheaper, Easier to Manage, More Scalable

We are Losing Data Locality

Today

- Compute-Storage Separation
- Cloud Data Lake
- K8s/Containerization

Data locality is missing …

The Implications of Losing Data Locality

Slow & Expensive

Slow and inconsistent data access performance

Fast-growing cloud storage costs, including API and egress costs

High data operation costs when migrating to the cloud

Complex Platform

Data copies, synchronization costs

Multiple APIs necessitate integration and application rewrites

On-premise, cloud, hybrid, multi-cloud environments all have different environment properties

ALLUXIO

10%

Of Your Data is Hot Data

Source: Alluxio

ALLUXIO

Caching's Values

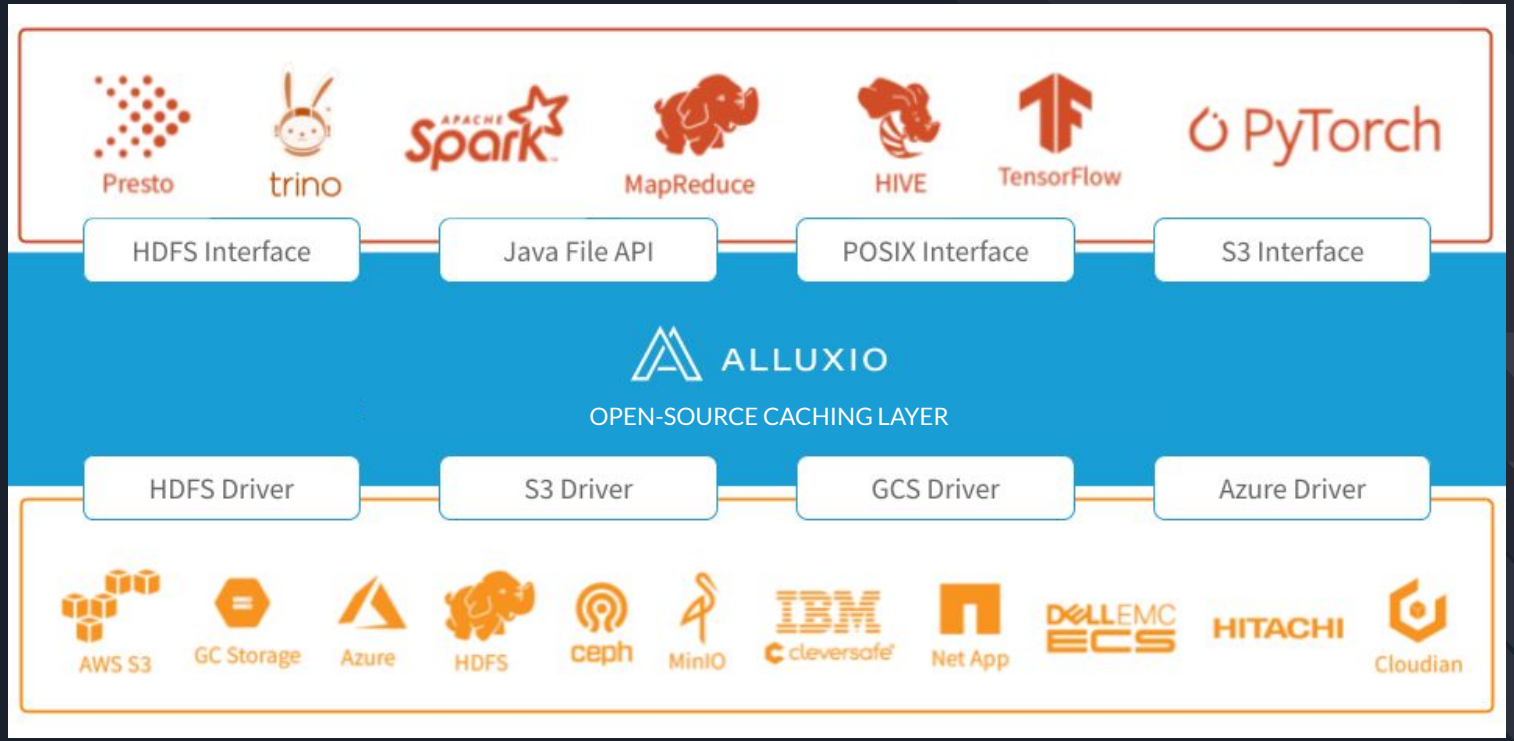
Boost Performance

Prevent Network Congestion

Save Costs

Offload Under Storage

Alluxio: A Critical Caching Framework in the Open-Source Data Stack



Open-Source Started From UC Berkeley AMPLab in 2014

Join the conversation on Slack
alluxio.io/slack

1,200+ contributors & growing

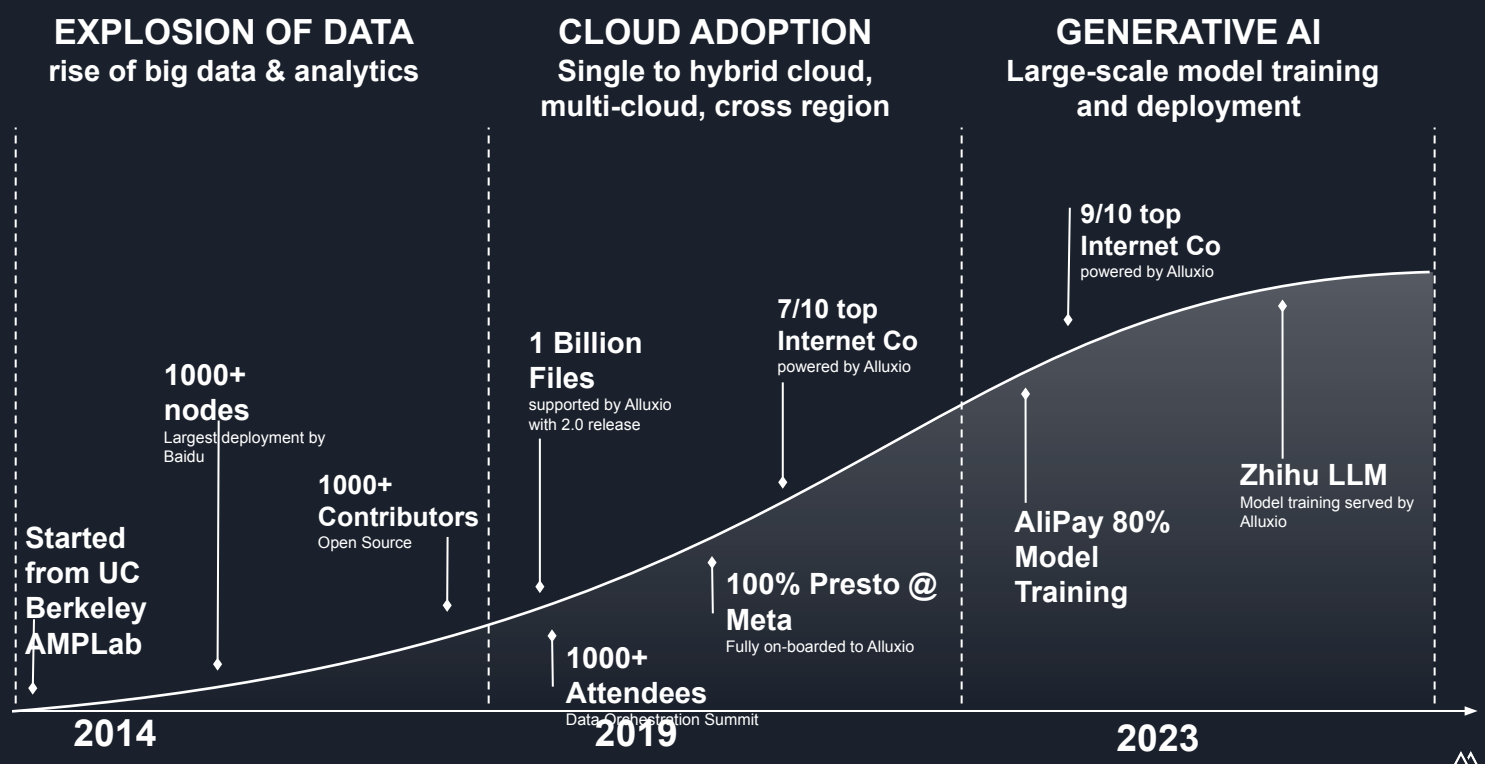
Top 10 Most Critical Java Based Open Source Project

GitHub's Top 100 Most Valuable Repositories Out of 96 Million

11,000+ Slack Community Members

ALLUXIO

Alluxio Technology Journey



ALLUXIO

Powered by Alluxio

TECHNOLOGY

INTERNET

Meta, Uber, expedia group, Alibaba.com, Tencent, JD.COM 京东, amazon, TikTok, Baidu 百度, Rakuten, waze

PUBLIC CLOUD PROVIDERS

aws, Google Cloud, Alibaba Cloud, Microsoft Azure, Tencent Cloud, Baidu Cloud

GENERAL

SAMSUNG, Aunalytics, Microsoft, Adobe, intel, IBM

E-COMMERCE

唯品会 VIPSHOP.COM 一家专门做特卖的网站, 苏宁易购 suning.com, Shopee, Walmart, Myntra

OTHERS

CAESARS ENTERTAINMENT, TalkingData, RYTE, Electronic Arts, 链家, ROBLOX, NetEase Games, yahoo!, nielsen, bazaarvoice, WeRide, GUARDANT HEALTH, Lenovo, eSENTIRE, 网易严选, PERCEPTIN, Ctrip, DiDi, Sogou 搜狗, esri, Lucidworks, CUELOGIC, MOMO, 云知声 Unisound, QINIU 七牛云存储, 去哪儿 Qunar.Com, ARIMO, ORACLE, walkme Digital Adoption Platform

FINANCIAL SERVICES

DBS, ING, BARCLAYS, HUATAI SECURITIES, PayPal, WELLS FARGO, Razorpay, Alipay

TELCO & MEDIA

COMCAST, 虎牙直播 huya.com, HUAWEI, swisscom, 中国移动 China Mobile, China unicom 中国联通, bilibili, 知

LEARN MORE

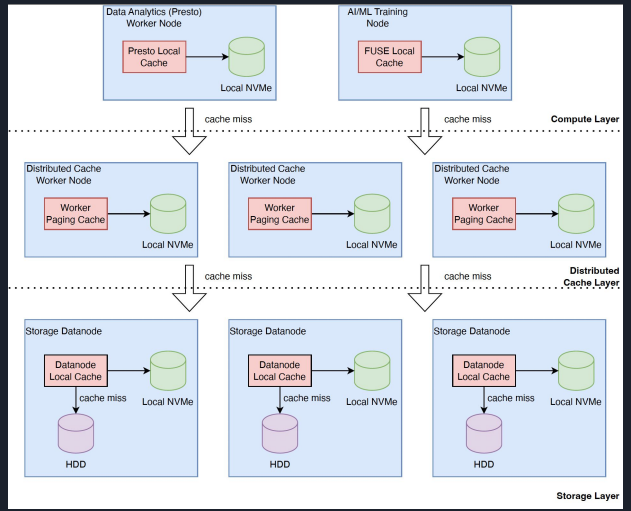
A Caching Framework to Fit Different Needs

Alluxio Embedded/Local Cache

- Run as a library in the application processes (Presto, HDFS DataNode)
- Leverage local disk NVMe or memory
- When the size of hot data fits local disks

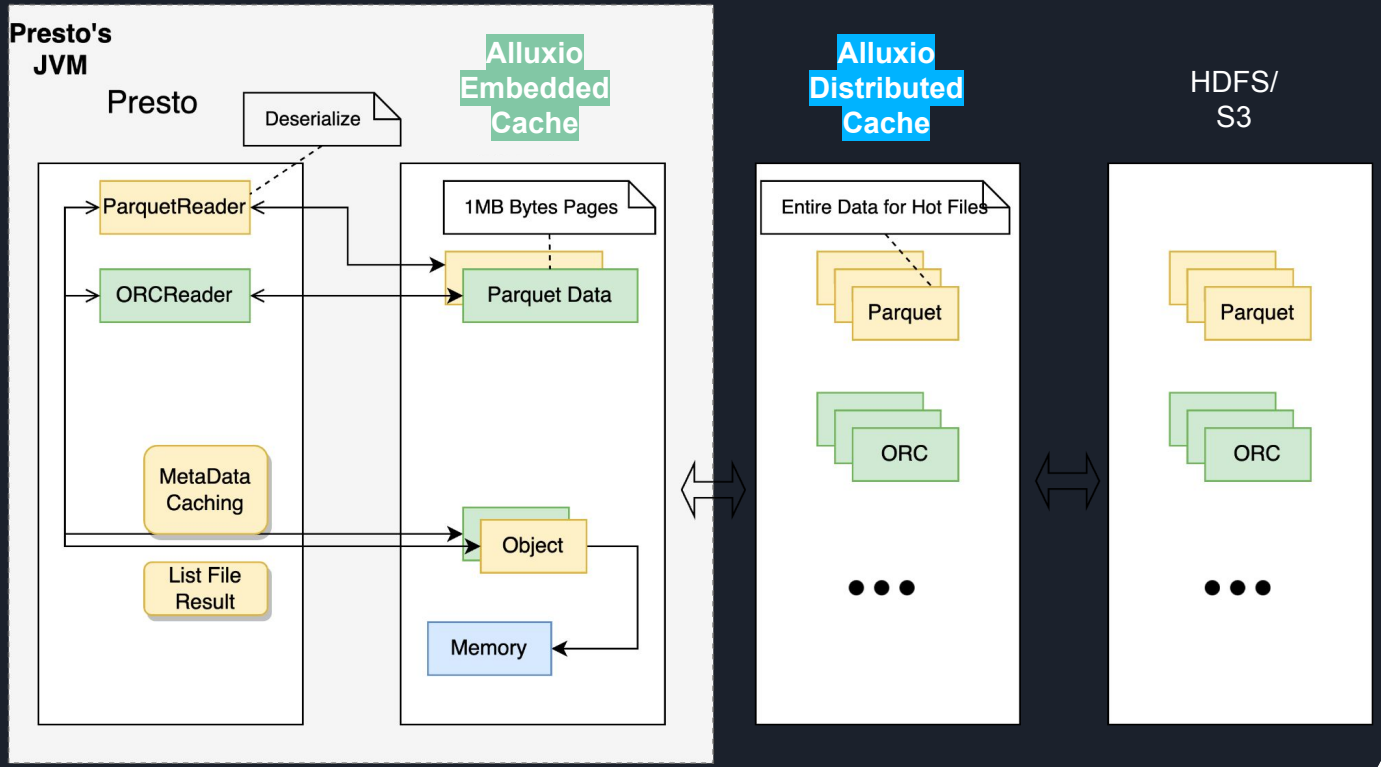
Alluxio Distributed Cache

- Standalone cache service shareable across applications
- Cache capacity scales horizontally



Multi-level Caching

L1 Embedded Cache+L2 Distributed Cache

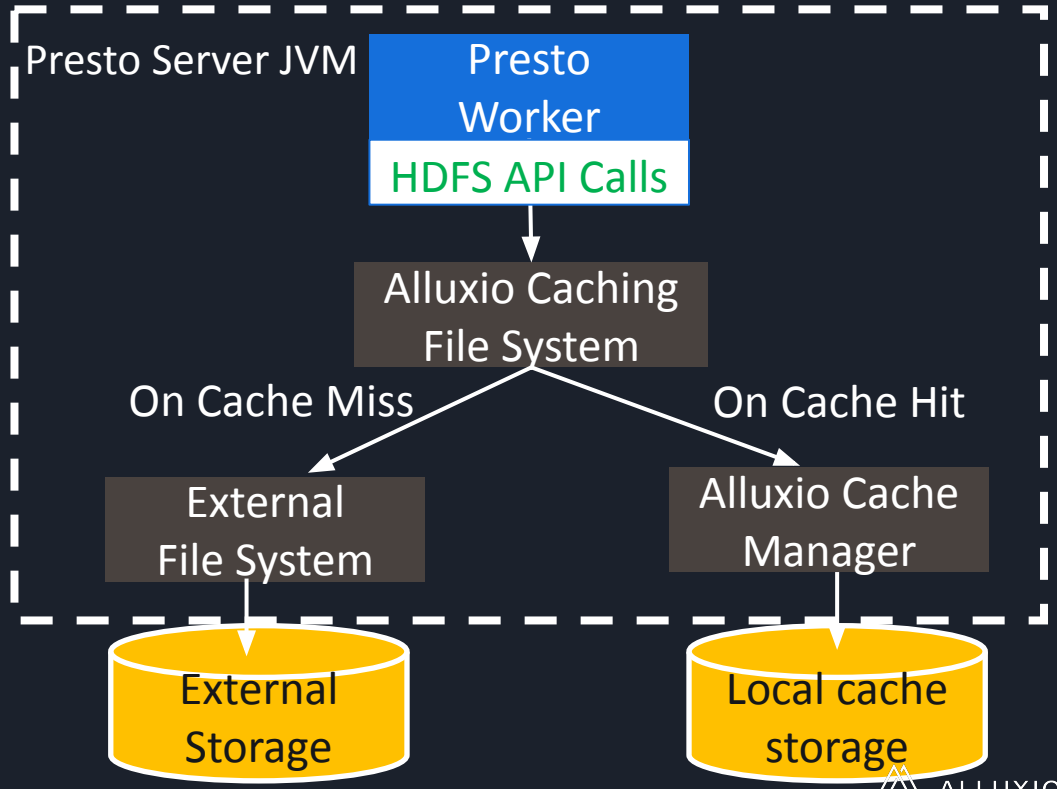


ALLUXIO

Alluxio Embedded Cache in Presto

Embedded Cache

- Battle tested in Uber, Meta, Tiktok and etc.
- Support Iceberg, Hudi, Delta Lake and Hive tables
- Support varied file format such as Paquet, ORC and CSV
- Fully optimized for local NVMe storages



ALLUXIO

Alluxio Embedded Cache Data Management

Embedded Cache

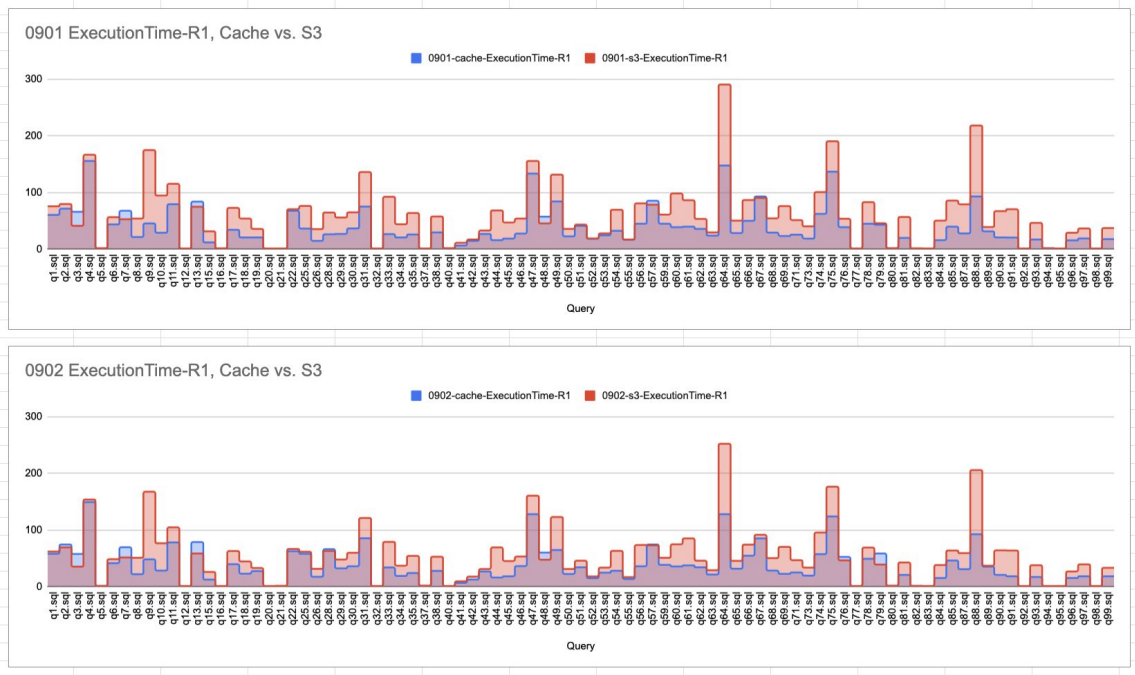
Alluxio embedded cache provides cache eviction and admission

- Support LRU and FIFO cache eviction policy
- Support customized cache admission policy
- Support TTL
- Support data quota

TPC-DS Benchmark of Presto+Embedded Cache

Embedded Cache

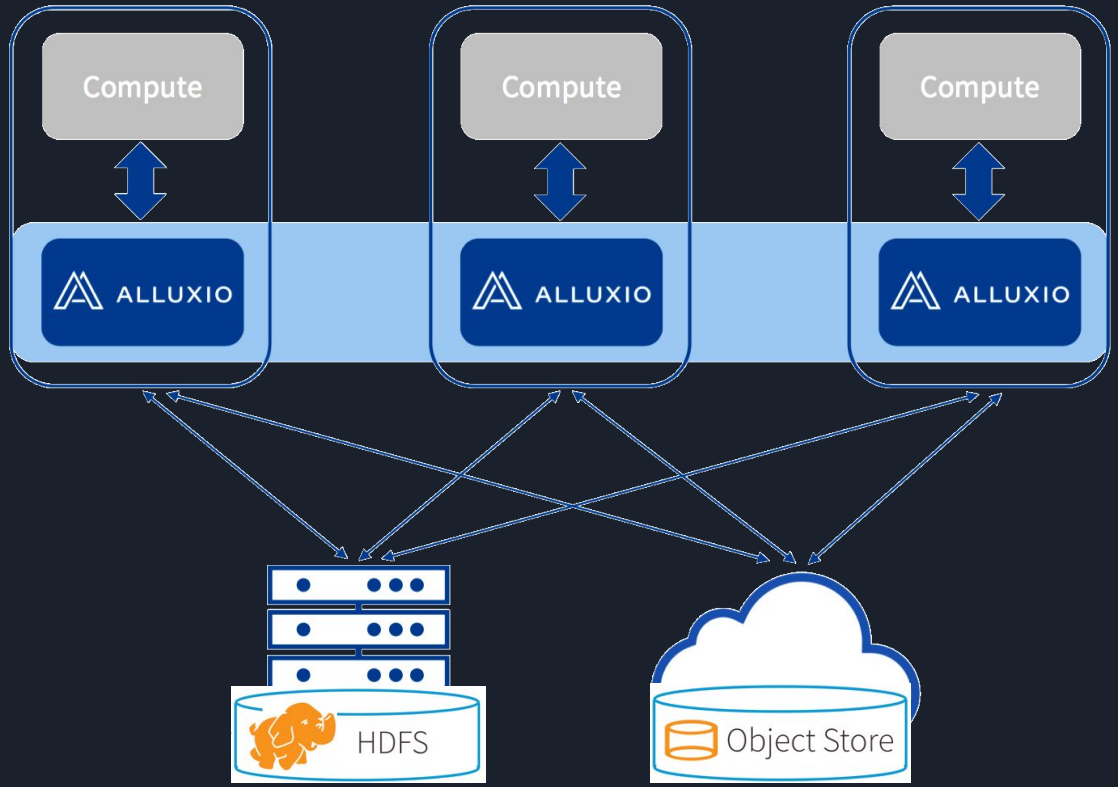
- Accelerate 70 out of 73 queries that lasting > 2s
- Total running time q1-q99 improved by 63% on average



ALLUXIO

Alluxio Distributed Cache

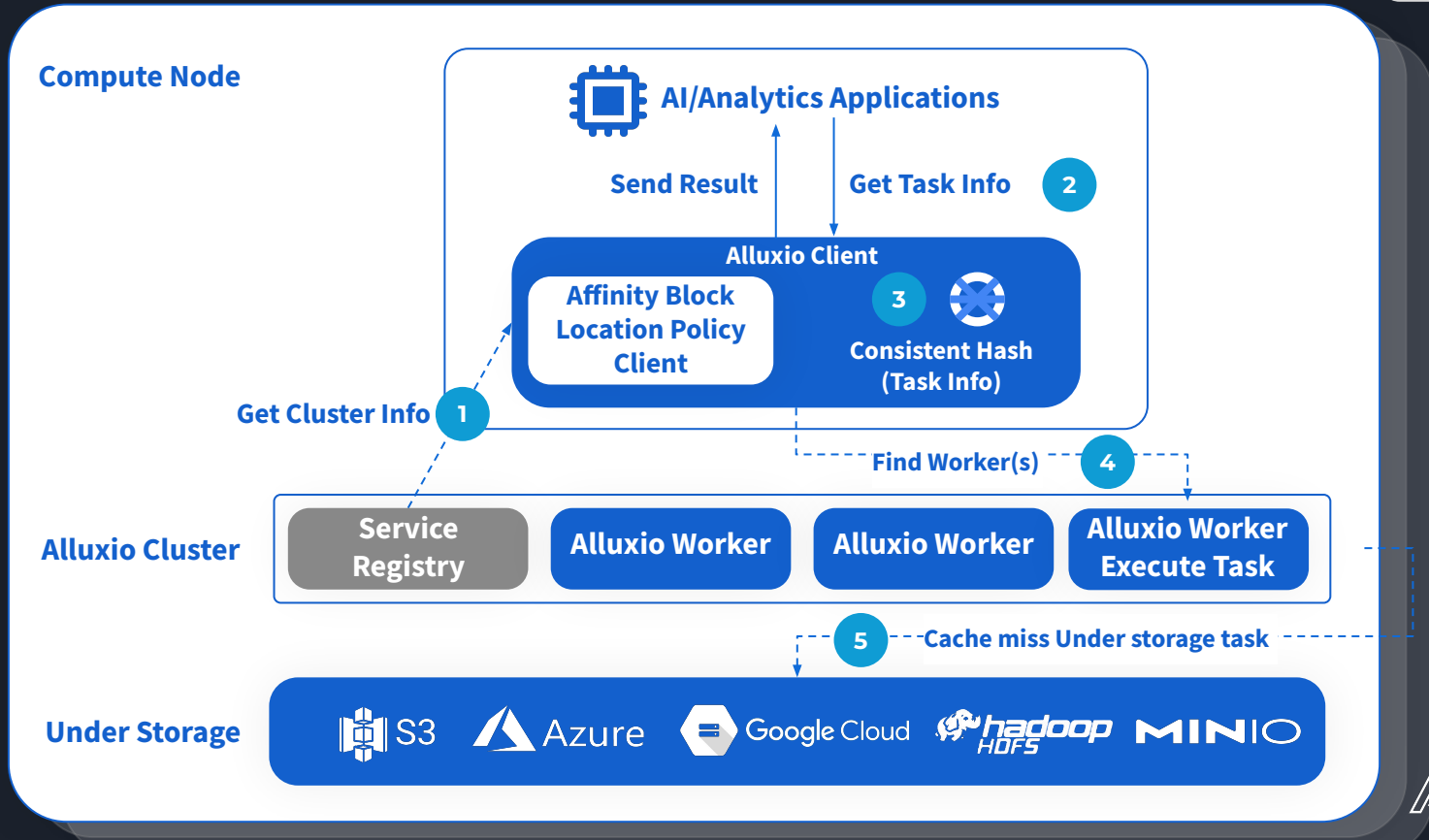
Distributed Cache



ALLUXIO

Alluxio Cluster Architecture

Distributed Cache



ALLUXIO

Uber: Using Cache for Exabyte-scale Data Lake

Uber

Data informs **every decision** at **Uber**

Community Operations

Marketplace Pricing

Eats

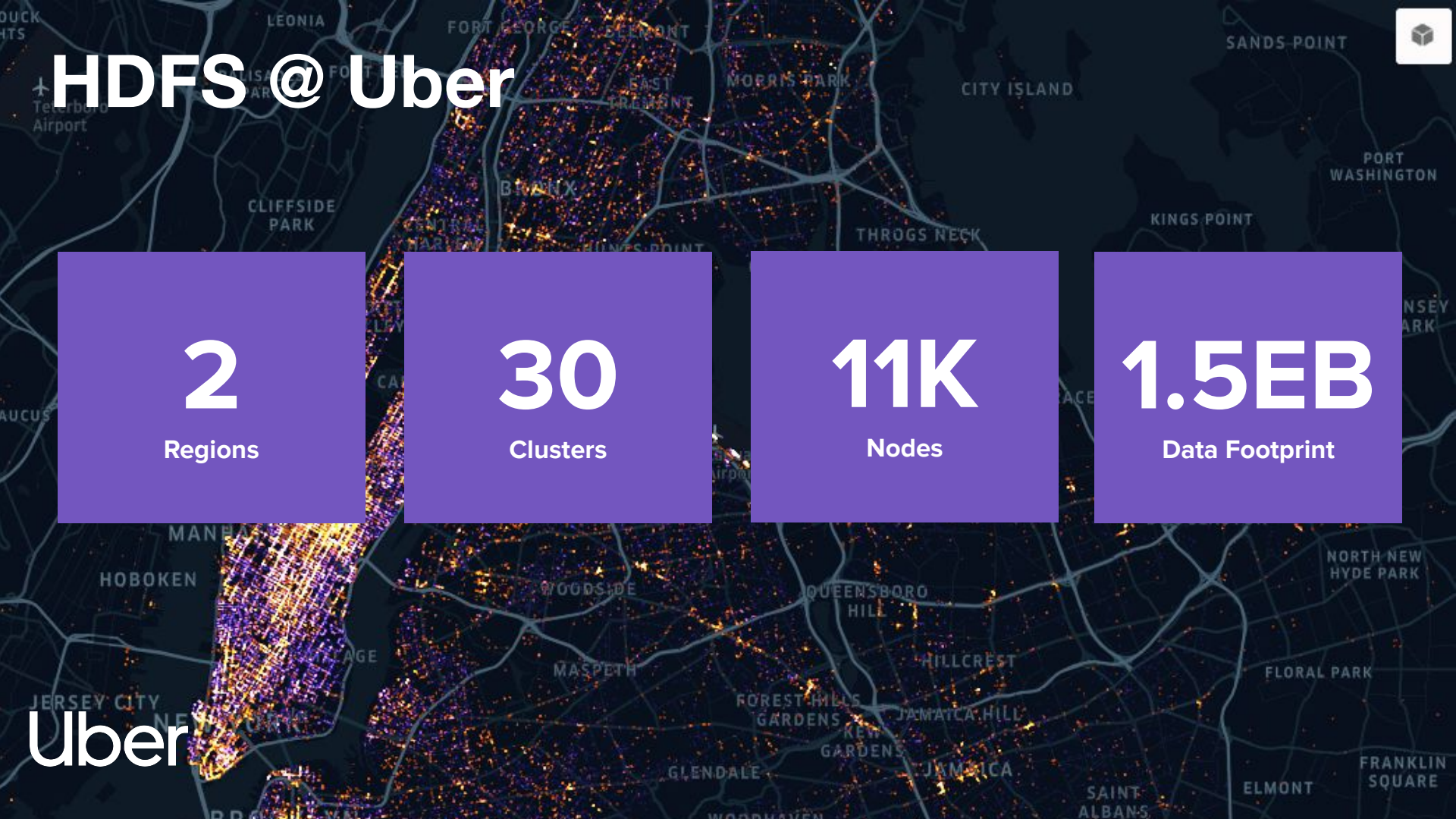
Compliance

Growth Marketing

Data Science

Uber

HDFS @ Uber



Uber

Adopting High-Density HDD in HDFS

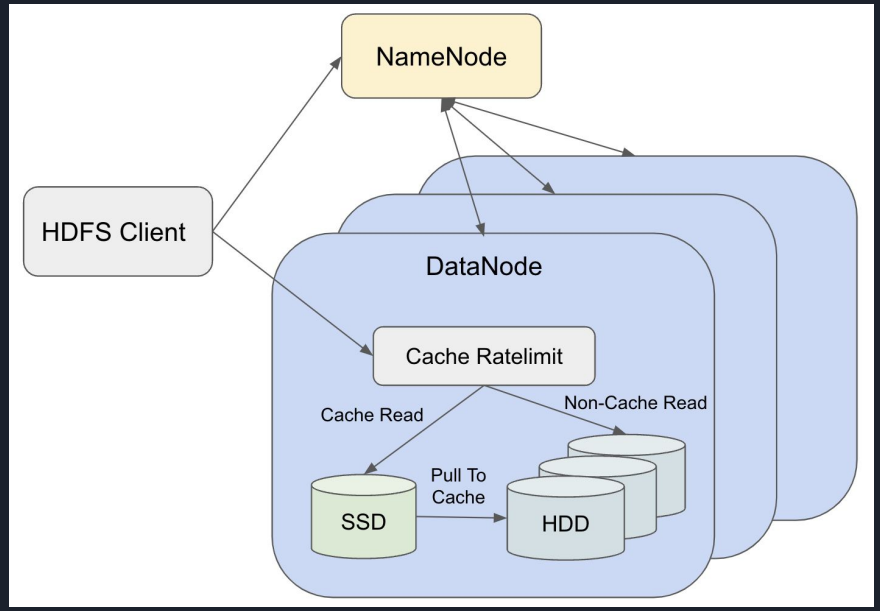
- Capacity per Host: 4TB * 24 → 16TB * 35
- Efficiency: >50% HW cost reduction
- Challenges
  - **DataNode IO performance**
  - HDFS reliability (blast radius)
- Opportunities
  - Traffic focuses on a small group of extremely hot blocks
  - Top 10K blocks attracted >90% read traffic

Host	Total reads	Total writes	Number of Blocks stored in the host	Number of blocks being read	Average Block Size	Capacity Usage	Read traffic on **top 10k** blocks (1 hour time window)
Host1	13.5M	3.3 K	1,074,622	84769	380 MB	77%	89%
Host2	12.8M	4.7 K	633,923	59376	330 MB	79.89%	94%
Host3	11.3M	275 K	479,544	49317	300 MB	79.86%	91%
Host4	8.5M	4.6 K	247,206	31048	300 MB	82.85%	99%
Host5	14.3M	45 K	463,819	79958	160 MB	81.62%	99%

Uber

DataNode Local Cache

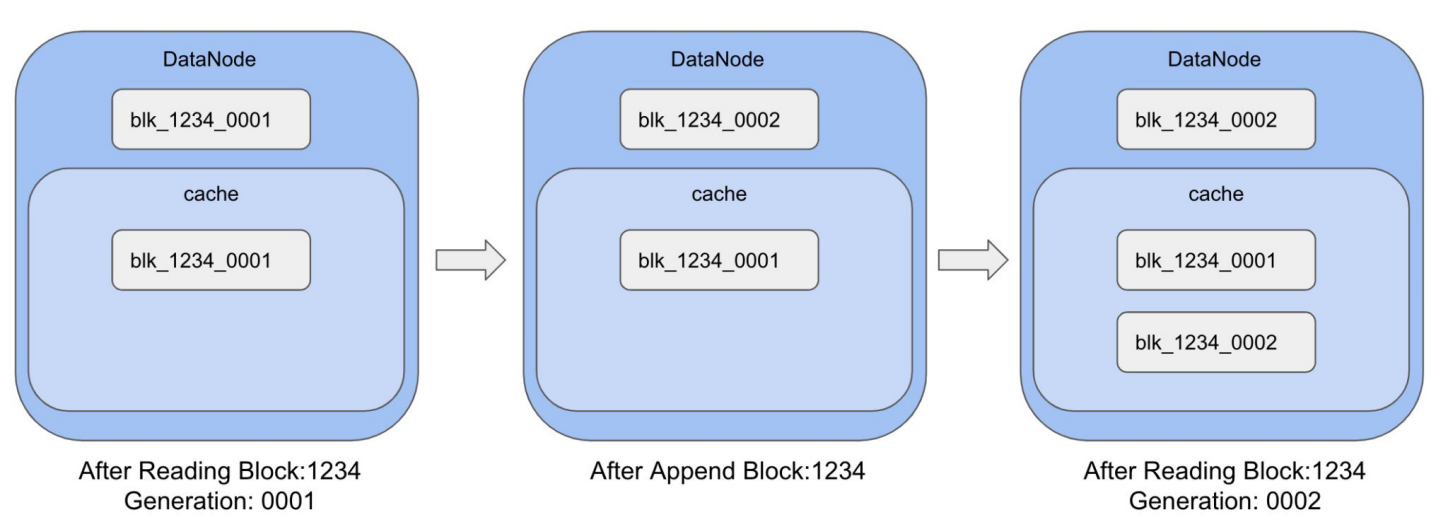
- Build a local cache within DataNode
  - 4TB NVMe SSD disk
  - Based on DataNode local traffic
- Leverage **Alluxio** for cache management
  - Page-level cache
  - 1MB default page size matches traffic pattern
- Exclude Non-Client Reads



Uber

Support Append to a Block

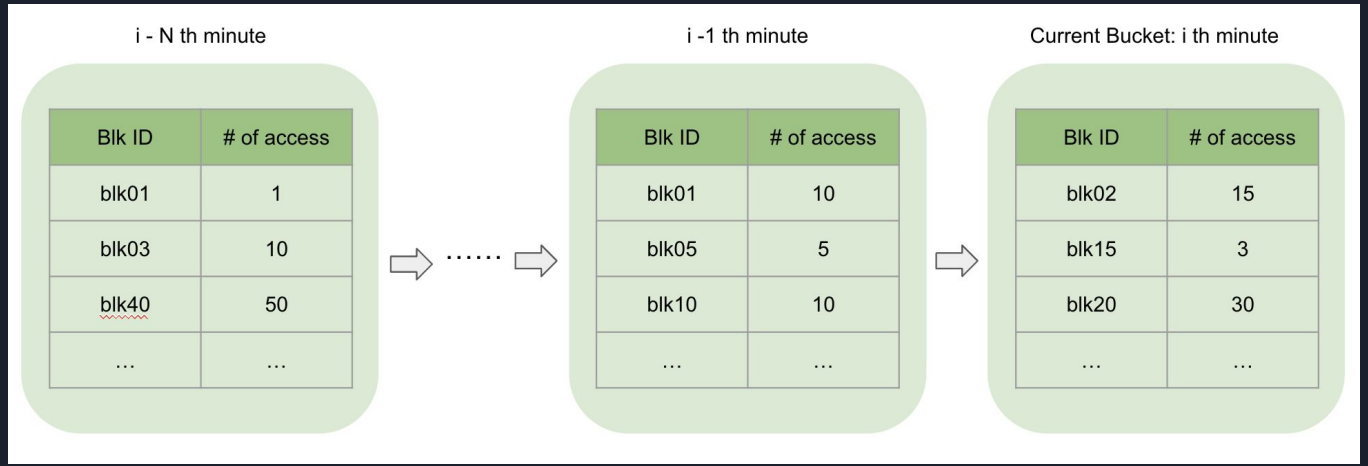
- HDFS blocks may not be immutable due to Append ops
- Leverage HDFS generation stamp to achieve “snapshot isolation”



Uber

Rate Limiter

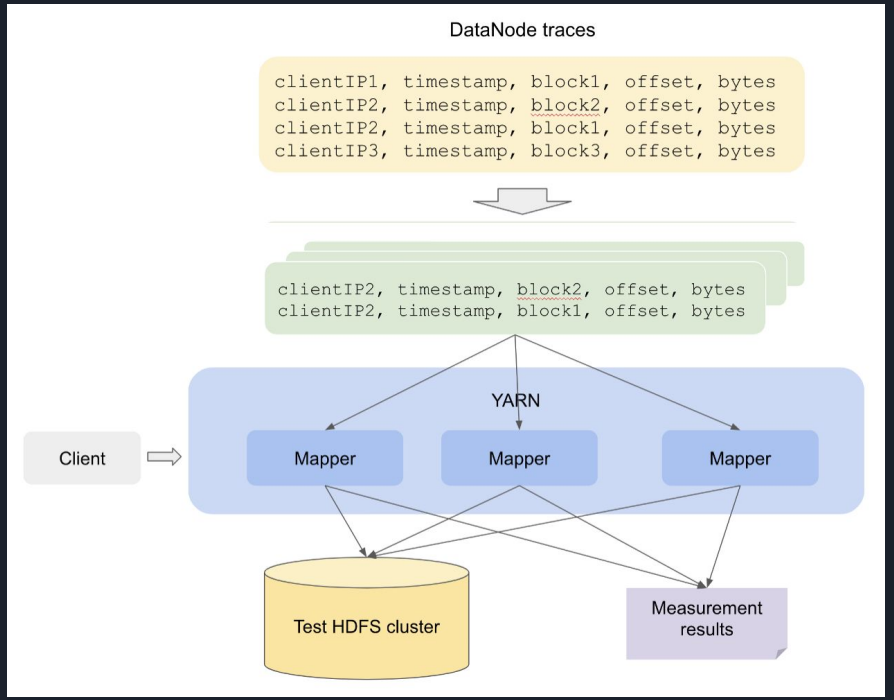
- 1:140 SSD-to-HDD ratio
- Need mechanisms to control what data can be loaded into Cache
  - Cache hit rate
  - SSD write endurance
- Track block access frequency within a sliding time window



Uber

Evaluation and Rollout

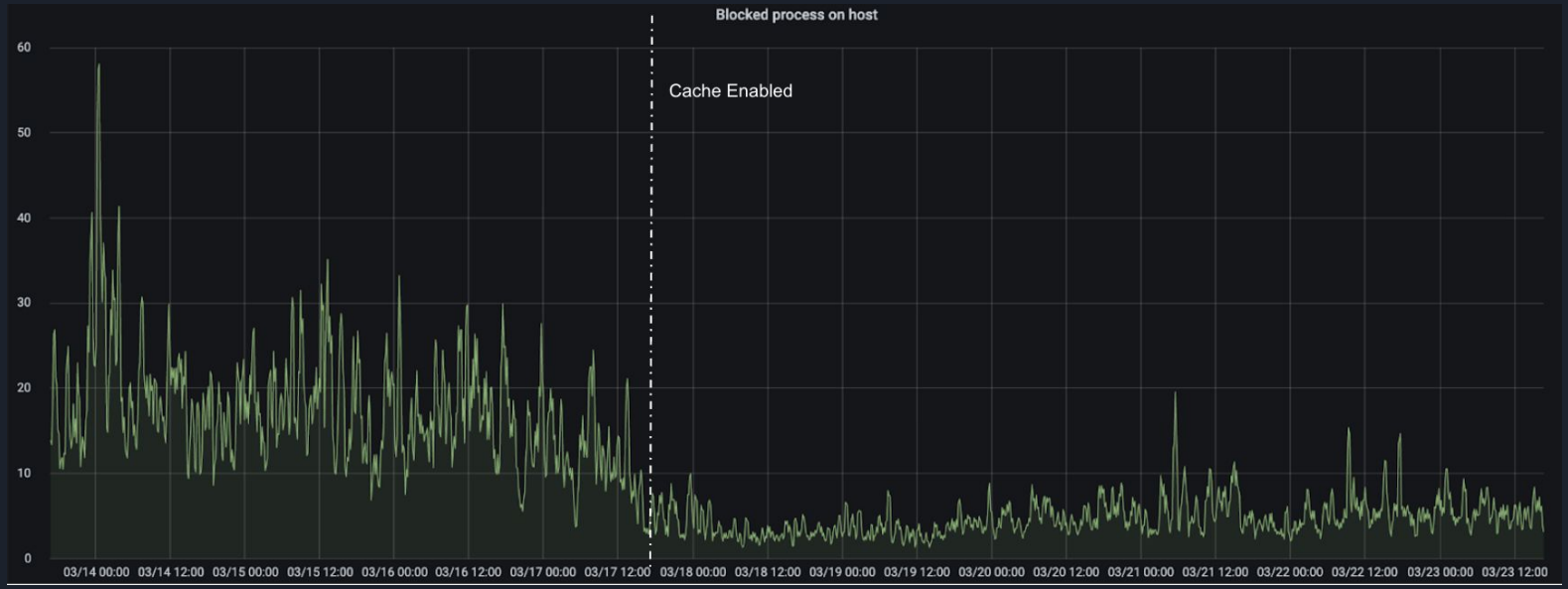
- Performance evaluation: DataNode trace replay
  - Test with scale
  - Reproduce production traffic profile
- Alluxio cache “shadow mode”
  - A simulated cache mode to collect metrics



Uber

Current Status

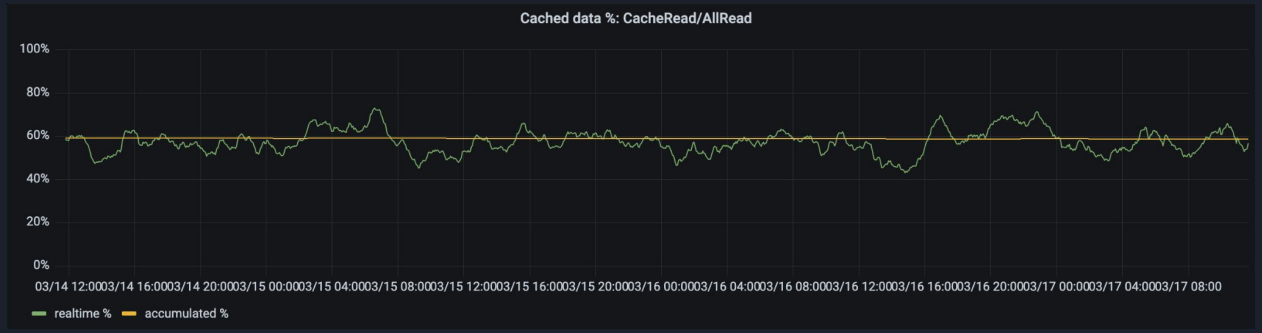
- Deployed in all major production clusters (1200 DataNodes)



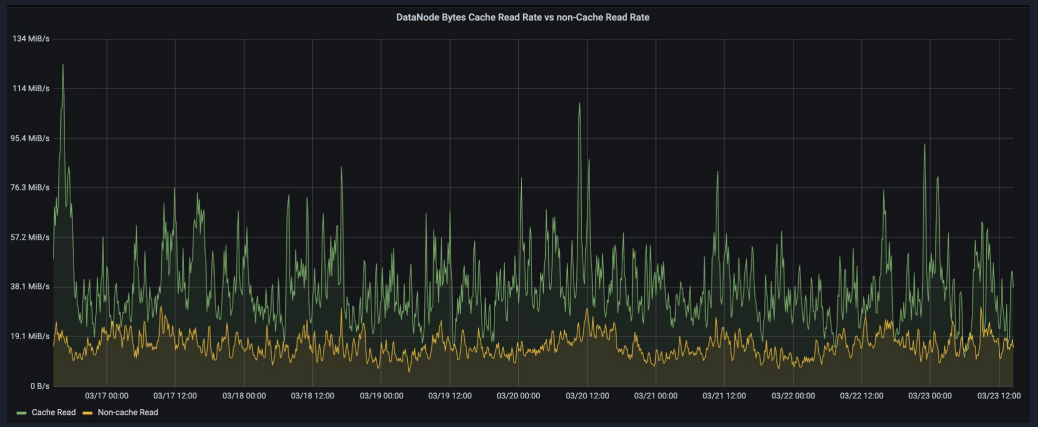
Uber

Effectively reduced the # of processes blocked by IO

Current Status



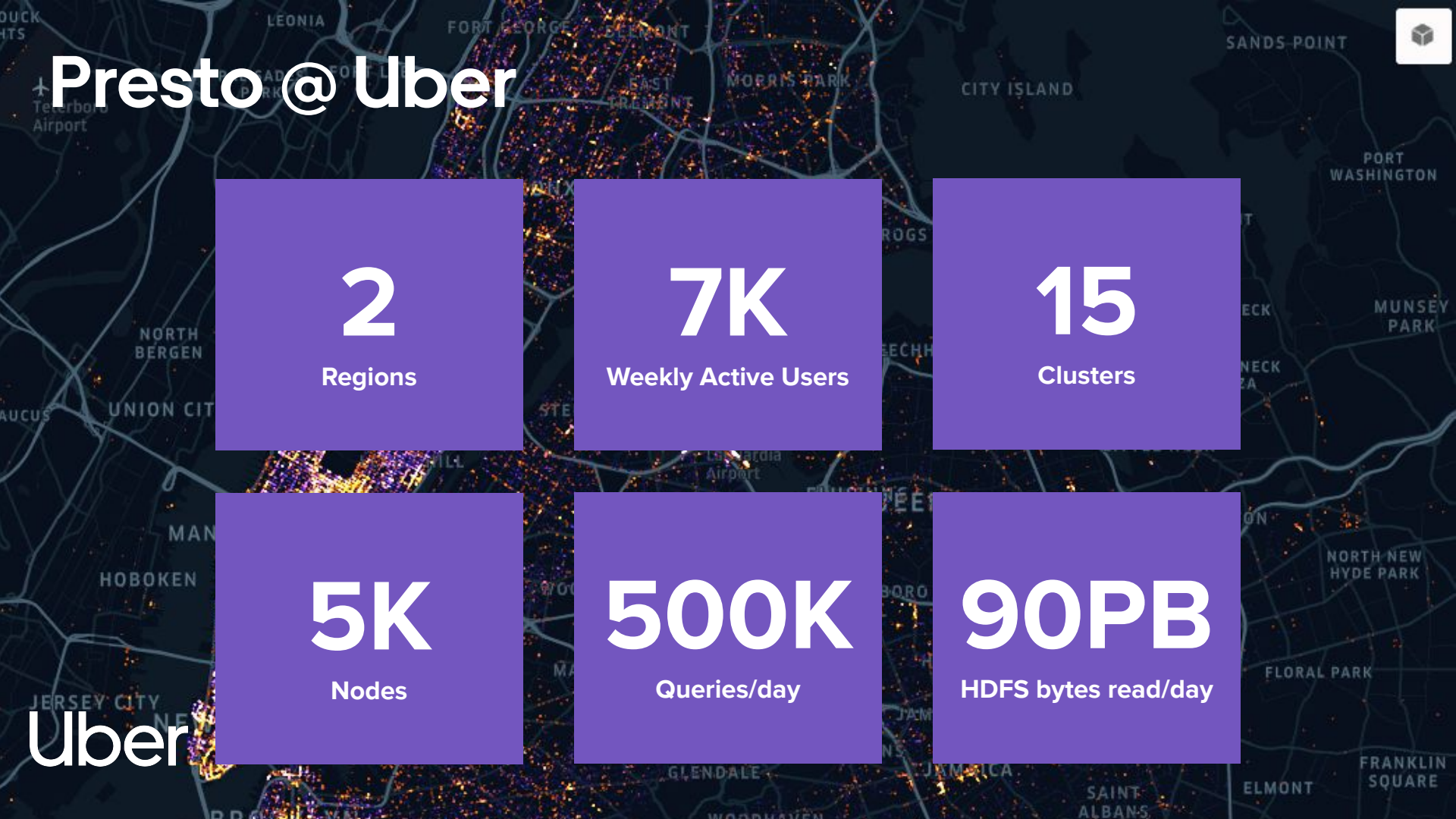
Offload 60% of IO from HDD



Cache read throughput is significantly higher, nearly twice that of non-cache read throughput

Uber

Presto @ Uber

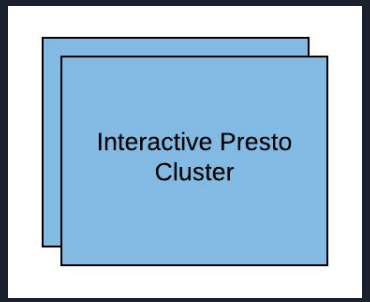


Uber

Workloads

Interactive

Ad hoc queries



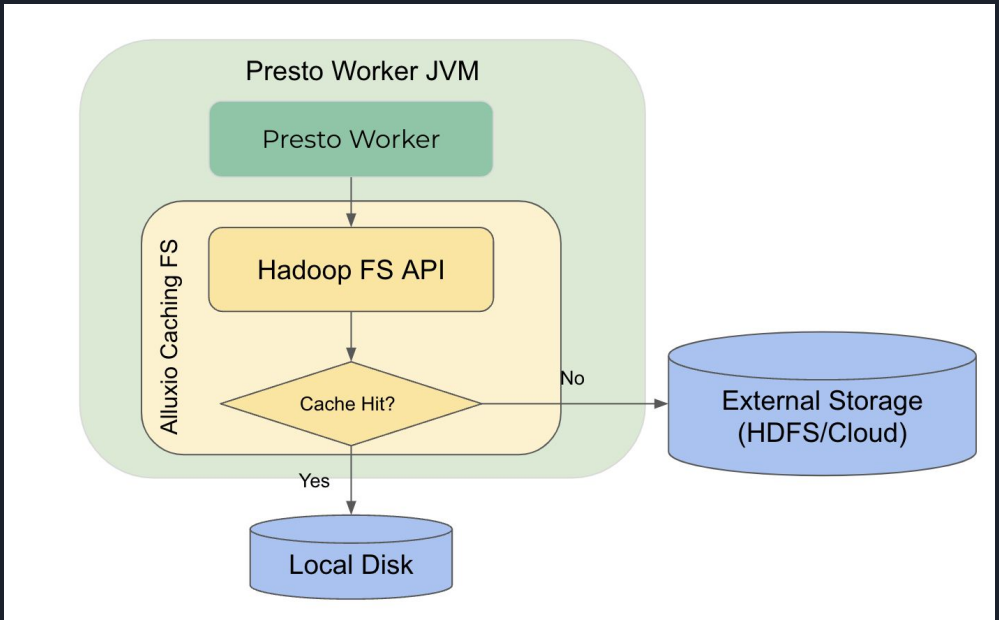
Batch

Scheduled

Uber

Presto Local Disk Cache

- Leverage Presto workers' local NVMe SSD disks
- Selective caching based on table
- Leverage Alluxio for cache management in Presto worker
- Forced cache TTL to meet compliance requirements



Uber

Key Challenges @ Uber and Solutions

- **Realtime Partition Updates**
  - **Problem**: Tables/partitions are constantly changing, naive caching could cause returning stale data.
  - **Solution**: Integrated HDFS file modification time as part of cache key
- **Cluster Membership Change**
  - **Problem:** Presto nodes always leave and join cluster, causing file read to route to wrong node and thus cache miss.
  - **Solution:** Introduced consistent hashing to tie files to certain Presto worker nodes.

Uber

Key Challenges @ Uber and Solutions

- **Cache Size Restriction**
  - **Problem:** Available cache space is limited compared to total data that needs to be scanned.
  - **Solution:** Introduced cache filter to selectively cache only certain hot data (top accessed tables).
- **Tiny Reads**
  - **Problem:** Uber's Presto traffic often involves a series of tiny consecutive reads, leading to degraded behavior in Alluxio.
  - **Solution:** Introduced buffer reads in Alluxio caching.

Uber

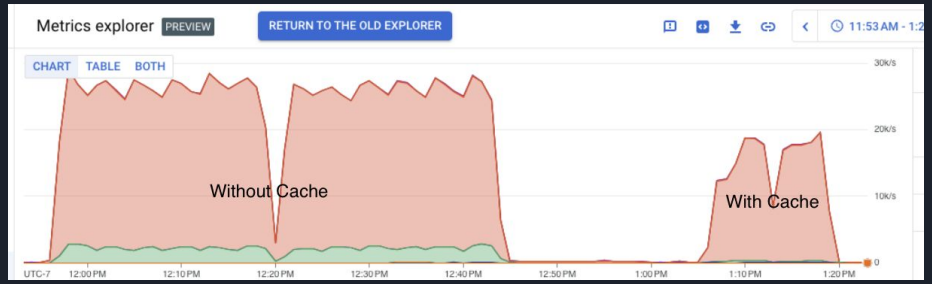
Presto Local SSD Cache Onprem - Today

- **Deployment**
  - Deployed to Presto production since 2022
  - Clusters run with local cache
    - 5 clusters (out of 11 batch clusters)
    - 1500 nodes (43% of all batch nodes in primary region)
- **Improvement**
  - **~13%** Presto batch HDFS read offloaded to cache
  - Input read latency reduced by **~44%** compared to HDFS reads

Uber

Presto Local SSD Cache On Cloud - Initial Evaluation

- Deployed to Presto on cloud for initial evaluation
- Cost reduction AND performance improvement
  - **>80% reduction** of # of read requests to GCS
  - **228 s → 50 s reduction** of P90 query latency



Uber

THANK YOU

Jing Zhao, Uber
Hope Wang, Alluxio