SDC 20

BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

Accelerate Big Data Workloads with HDFS Persistent Memory Cache

Feilong HE
Jian Zhang
Intel Corporation

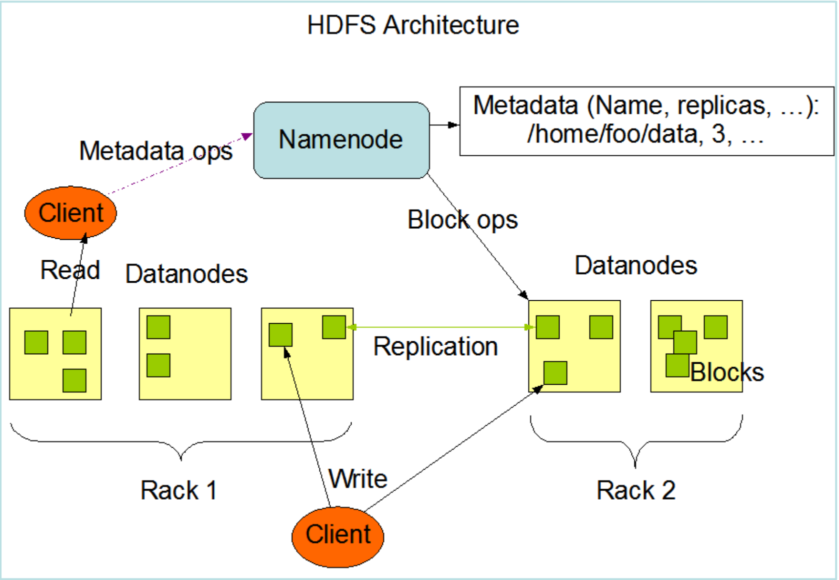
Agenda

- HDFS Introduction
- PMem Introduction
- HDFS PMem Read Cache
- HDFS PMem Performance Test
- Summary

HDFS Introduction

HDFS Introduction

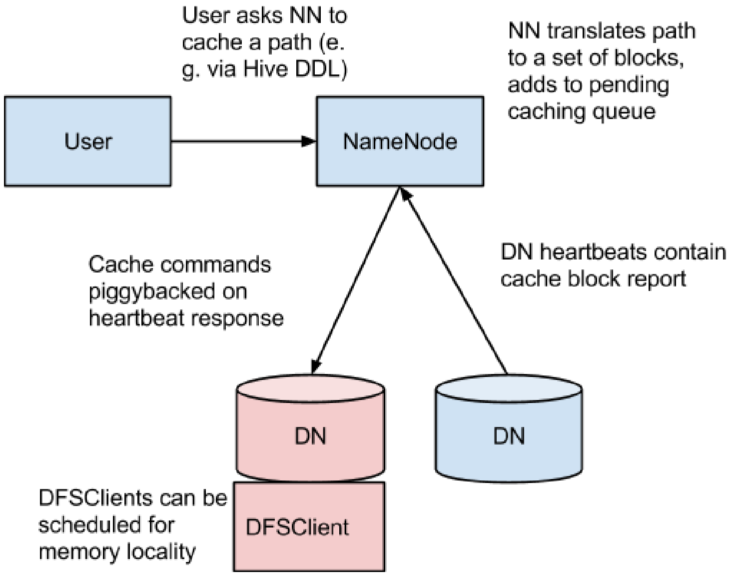
- **Fault Tolerance**
  HDFS is a distributed file system that is fault tolerant, by storing redundant replicas or storing data by erasure coding accross nodes.
- **Good Scalability**
  HDFS is scalable and easy to scale out. It is common to deploy it on thousands of nodes in product environment.
- **High Throughput**
  HDFS can provide very high read/write throughput to application and is suitable for applications with large dataset.
- **Widely Used**
  HDFS is the primary distributed storage used in Hadoop ecosystem.



HDFS Centralized Read Cache

- **Easy to Use**
  To cache data, user just needs to specify its path, number of replicas.
- **Scale Out**
  Each DataNode can cache data, easily to scale out overall cache capacity of the storage system.
- **No Eviction**
  Cached data is held in locked memory. Currently, cache eviction is not supported.
- **Cache Locality**
  Upper applications can optimize their task scheduling by considering cache locality exported by HDFS.

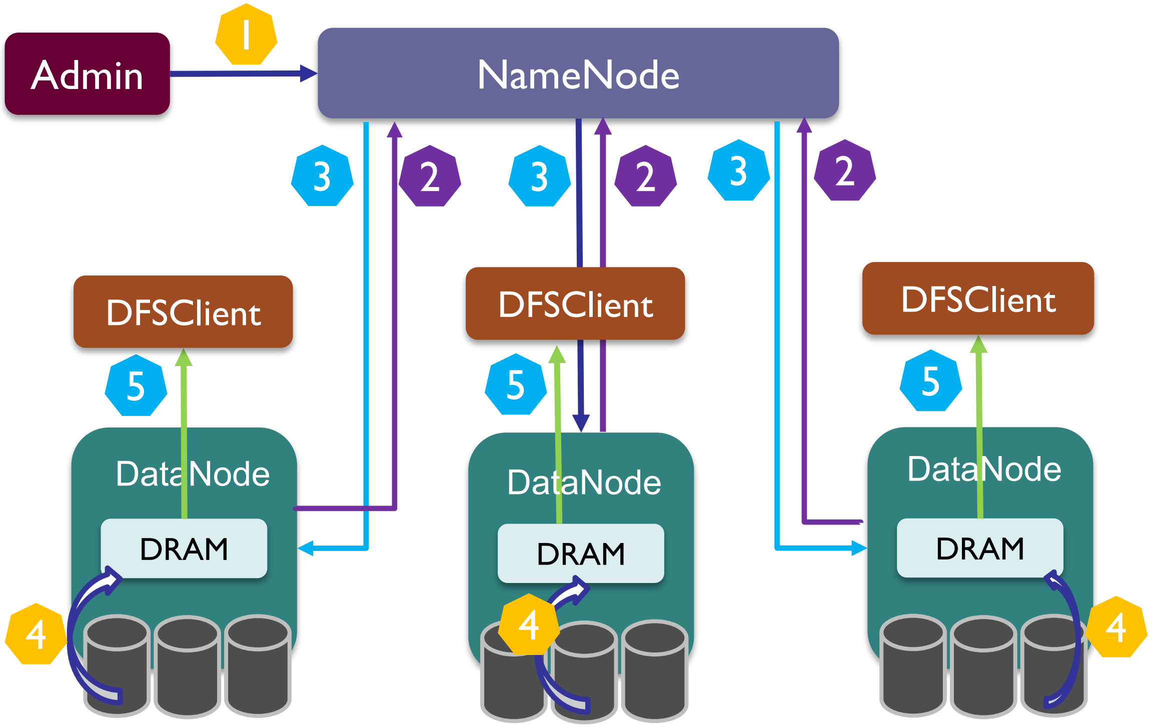
- **Short-circuit Read**
  Short-circuit read is supported, i.e., bypassing send-receive fashion of IO between client process and server process.



Pic. source: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/CentralizedCacheManagement.html

HDFS Centralized Read Cache

1. Send command to NN to cache a path
2. Heartbeat
3. Cache block command
4. Cache data to memory
5. Read directly from memory (zero copy short-circuiting read)

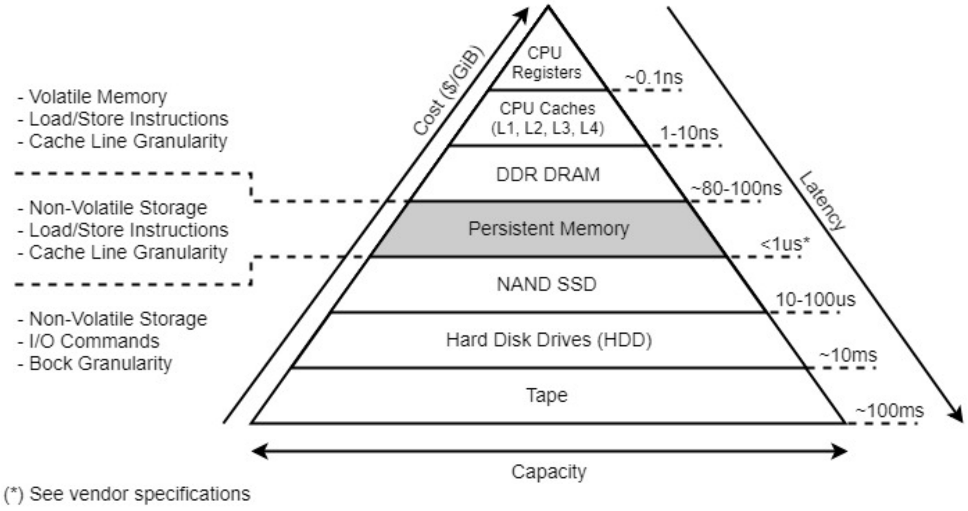


PMem Introduction

PMem Introduction

Persistent Memory (PMem)

- Lays between SSD and DRAM in terms of capacity, unit cost or IO latency.
- Larger capacity than DRAM.
- Support non-volatile mode.



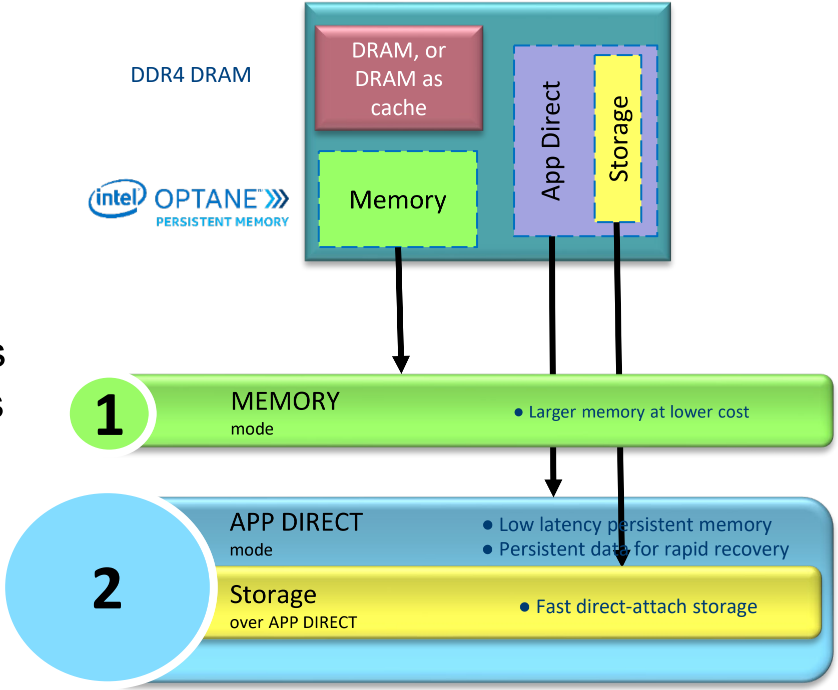
Memory-Storage Hierarchy with Persistent Memory Tier *

* Source: https://docs.pmem.io/persistent-memory/getting-started-guide/introduction

PMem Introduction

Intel Optane™ PMem*.

- **Memory Mode**
  Volatile storage. Provides higher memory capacity. Cache management is handled by processor's integrated controller.
- **APP Direct Mode**
  Non-volatile storage. Application manages its cache data by itself. Read/write bypass page cache.
- **Cost**
  Cost/GB is lower than DRAM.
- **Single Capacity**
  128GB, 256GB, 512GB

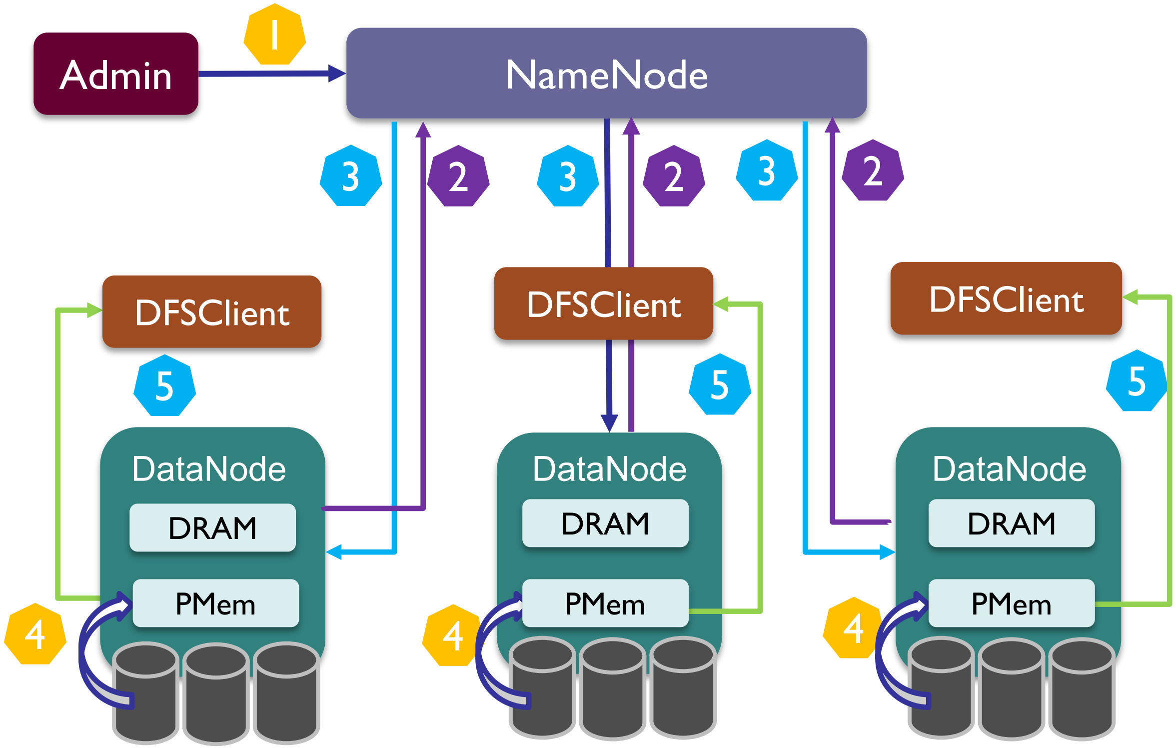


* https://www.intel.com/content/www/us/en/architecture-and-technology/optane-dc-persistent-memory.html

HDFS PMem Read Cache

HDFS PMem Read Cache

1. Send command to NN to cache a path
2. Heartbeat
3. Cache block command
4. Cache data to PMem
5. Read directly from PMem



HDFS PMem Read Cache Implementation

Introduced Two Implementations to HDFS:

I. **Non-PMDK implementation**
   - Not dependent on any native libs.
   - Code portable to different platform.

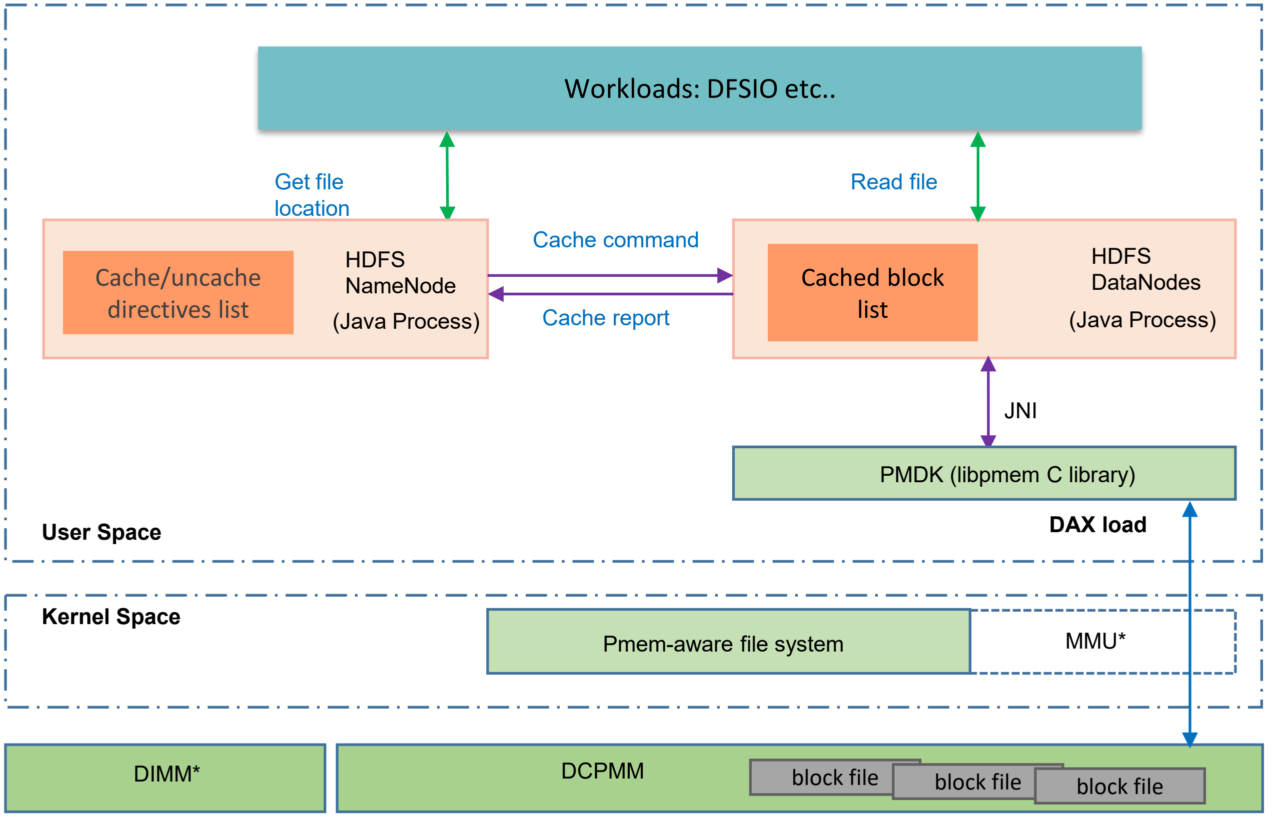
II. **PMDK based implementation**
   - Use native PMDK (persistent memory development kit) to load cache to Pmem. It's a collection of libraries and tools for development on persistent memory.
   - Better IO performance.

pmem_map_file(path, length, …)
Create a read/write mapping for a file, return an address on PMem.

pmem_unmap(address, length, …)
Delete the mapping on PMem.

pmem_memcpy(address, srcBuf, length, …)
Copy data in buffer to PMem.

HDFS PMem Read Cache with PMDK



* MMU: Memory Management Unit
* DIMM: Dual Inline Memory Module

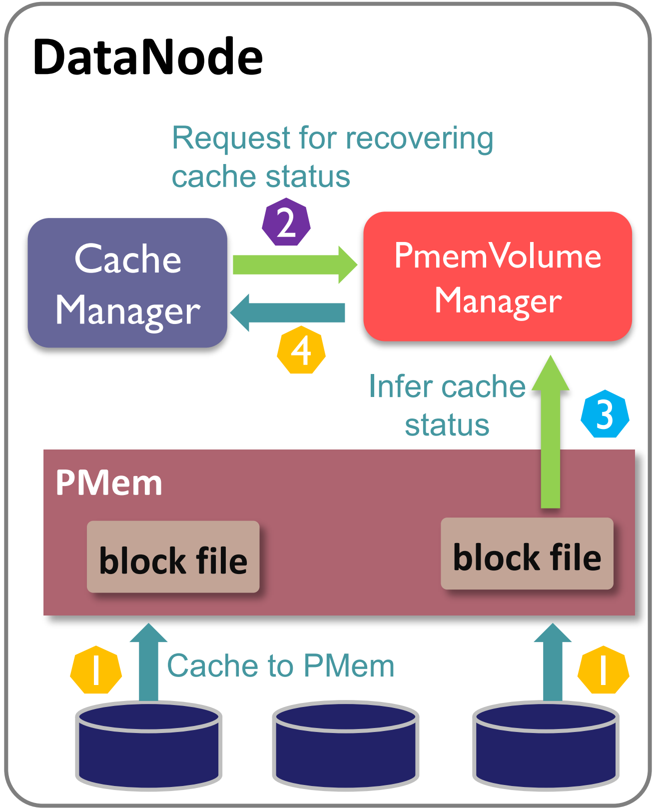
HDFS PMem Read Cache Recovery

Hierarchical Cache Storage

- The cached block is named by corresponding block ID and it is stored hierarchically inside a dir named by block pool ID. 1
- Block pool ID & block ID form a unique identity.

Cache Recovery

- When DN restarts, Cache Manager asks PVM* to scan PMem cache volume. 2
- PVM speculates cache block's block pool ID & block ID according to its path and name. 3
- Then, update the cache status in DN. And report to NameNode via heartbeat. 4



* PmemVolumeManager

Enable HDFS PMem Read Cache

Non-PMDK implementation

Java common file API is used to cache data and read the cached data.

mvn clean package -Pdist,native -DskipTests –Dtar

PMDK based implementation

1) **Install PMDK**

   Has system requirements. Reference link: https://github.com/pmem/pmdk

2) **Build with PMDK**

   Build Apache Hadoop with PMDK. In running time, if this native lib is available, PMDK based implementation will be picked to use.

   mvn clean package -Pdist,native -DskipTests -Dtar -Drequire.pmdk -Dpmdk.lib=/usr/lib64

HDFS PMem Read Cache Usage

Configuration

Only one configuration is needed in a HDFS config file, hdfs-site.xml.

```
<property>
    <name>dfs.datanode.pmem.cache.dirs</name>
    <value>/mnt/pmem0,/mnt/pmem1</value>
</property>
```

Choose Policy

Round robin policy is used to choose an available PM if more than one PM is configured.

Cache Directives

All HDFS cache directives keep unchanged.

```
hdfs cacheadmin -addDirective -path <path> -pool <pool-name> -replication <replica-num>
```

Performance Test Configuration

Performance Test Configuration

	DRAM	HDD (no cache)	Intel Optane PMem
System	CLX-2S	CLX-2S	CLX-2S
CPU	CLX 6240, HT on	CLX 6240, HT on	CLX 6240, HT on
CPU per node	18core/socket, 2 sockets, 2 threads per core	18core/socket, 2 sockets, 2 threads per core	18core/socket, 2 sockets, 2 threads per core
Memory	DDR4 dual rank 768GB = 24 * 32GB	DDR4 dual rank 192GB=12 * 16GB	DDR4 dual rank 192GB=12 * 16GB Intel PMem 8 * 128GB ES2
Network	10GbE	10GbE	10GbE
Storage Type	1x SATA SSD for OS 1TB SATA SSD for name node 2 P4500 for Spark Shuffle 6x 1TB HDD on datanode	1x SATA SSD for OS 1TB SATA SSD for namenode 2 P4500 for Spark Shuffle 6x 1TB HDD on datanode	1x SATA SSD for OS 1TB SATA SSD for namenode 2 P4500 for Spark Shuffle 6x 1TB HDD on datanode
BIOS	SE5C620.86B.02.01.0008.031920191 559	SE5C620.86B.02.01.0008.0319201915 59	SE5C620.86B.02.01.0008.031920191559
OS/Hypervisor/S W	OS: Fedora 29 Java 1.8, Hadoop 3.1.2 , Mysql 5.7	OS: Fedora 29 Java 1.8, Hadoop 3.1.2 , Mysql 5.7	OS: Fedora 29 Java 1.8, Hadoop 3.1.2 , Mysql 5.7
WL Version	DFSIO Decision Support Workloads	DFSIO Decision Support Workloads	DFSIO Decision Support Workloads
Input Data Set	Total data set 1TB	Total data set 1TB	Total data set 1TB
Special Patches	HDFS/cache replication factor=2	HDFS/cache replication factor=2	HDFS/cache replication factor=2

HDFS PMem Cache Persistence Test

HDFS PMem Cache Persistence Test

Persisted Cache Recovery

An enhancement for HDFS PMem cache. Recover cache status by speculating according to persisted cache's metadata.

Recache time vs. Cache Recovery time

With taking advantage of PMem's persistent characteristic, a great deal of time can be saved by reloading HDFS persistent cache, especially in large dataset case.

Dataset Size (GB)	Recache Time (s)	Cache Recovery Time (s)	Speedup
96	200	0.02	10000
512	1400	0.08	17,500
800	1690	0.106	15,943
920	2113	0.127	16,637

Performance results are based on testing as of 12/06/2019 and may not reflect all publicly available security updates. See configuration disclosure on slide for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. Configurations refer to slide titled Performance Test Configuration.

DFSIO Test

DFSIO Test

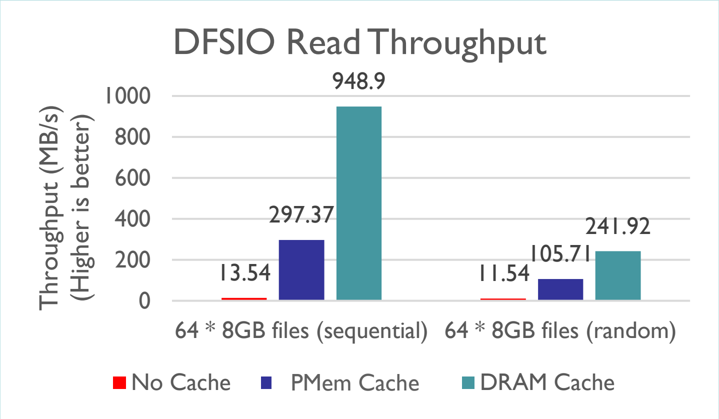
DFSIO

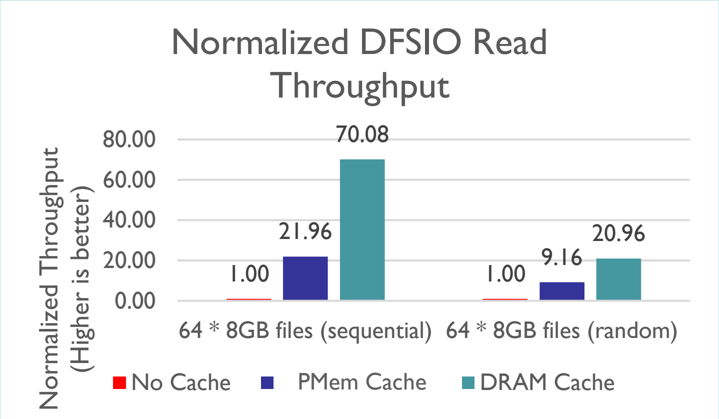
A commonly used workload to benchmark IO performance for HDFS. The workload can trigger a given number of parallelly running Map-Reduce tasks to purely read HDFS data.

Hardware Configuration

For fair comparison, the HW cost for DRAM test env. is as same as that for PMem test env.

DFSIO Performance (512GB)





Workload:
DFSIO Random & Sequential Read
Total Data Set:
64 * 8GB = 512GB

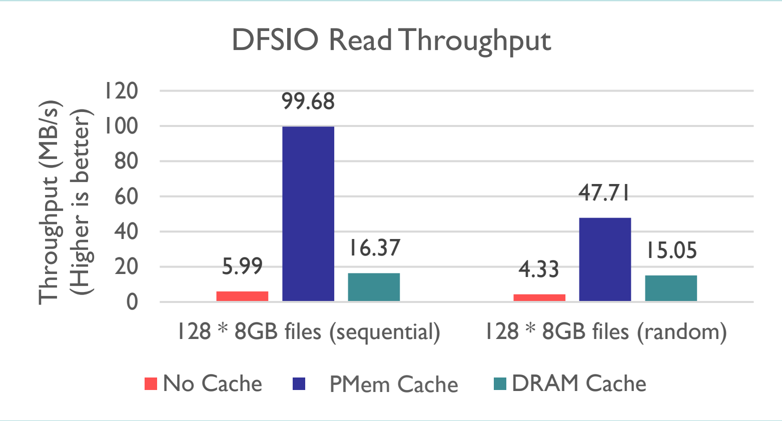
PMem capacity: 918G (fully cached)
DRAM capacity: 560GB (fully cached)
Metrics: Throughput (MB/S)
Baseline: 6 * HDD (no cache)

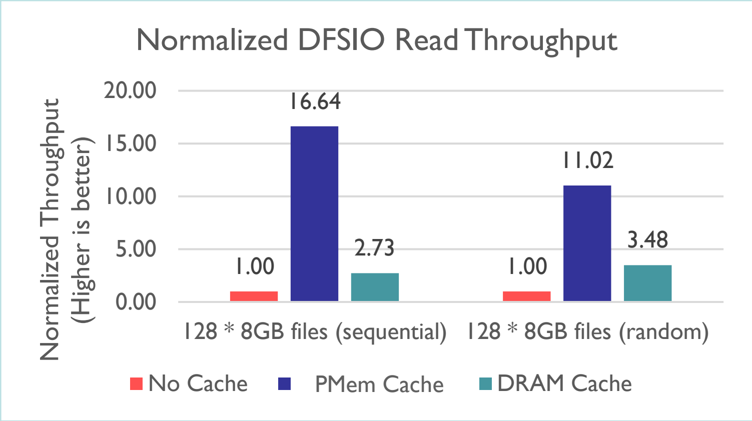
PMem Cache vs. DRAM Cache

- PMem Cache is 69% lower for sequential read and 56% lower for random read in compared with DRAM cache.
- Data can be fully cached into DRAM and PMem, need larger dataset tests.

Performance results are based on testing as of 12/06/2019 and may not reflect all publicly available security updates. See configuration disclosure on slide for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. Configurations refer to slide titled Performance Test Configuration.

DFSIO Performance (1TB)





Workload:
DFSIO Random & Sequential Read
Total Data Set:
128 * 8GB=1TB

PMem capacity: 918G (near fully cached)
DRAM capacity: 560GB (partially cached)
Metrics: Throughput (MB/S)
Baseline: 6 * HDD (no cache)

PMem Cache vs. DRAM Cache

- PMem Cache delivers up to 3.16x speedup for random read and 6.09x speedup for sequential read in compared with DRAM cache.

Performance results are based on testing as of 12/06/2019 and may not reflect all publicly available security updates. See configuration disclosure on slide for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. Configurations refer to slide titled Performance Test Configuration.

Decision Support Workload Test

Decision Support Workload Test

Decision Support Workload

A classic SQL benchmark, with IO-intensive queries included. In our test, HDFS is used as storage backend and Spark SQL serves as SQL engine.

Hardware Configuration

For fair comparison, the HW cost for DRAM test env. is as same as that for PMem test env.

Decision Support Workload (2TB raw data)

Workload

Decision Support Workloads, 54 selected queries, 2TB raw data.

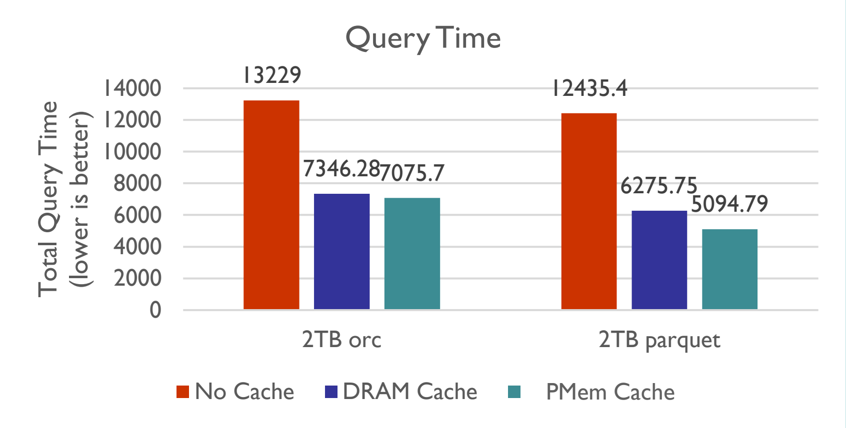
Data format	Actual dataset size (GB)	No. of table cached to PMem	No. of table cached to DRAM
Parquet	816	24	21
ORC	706	24	21

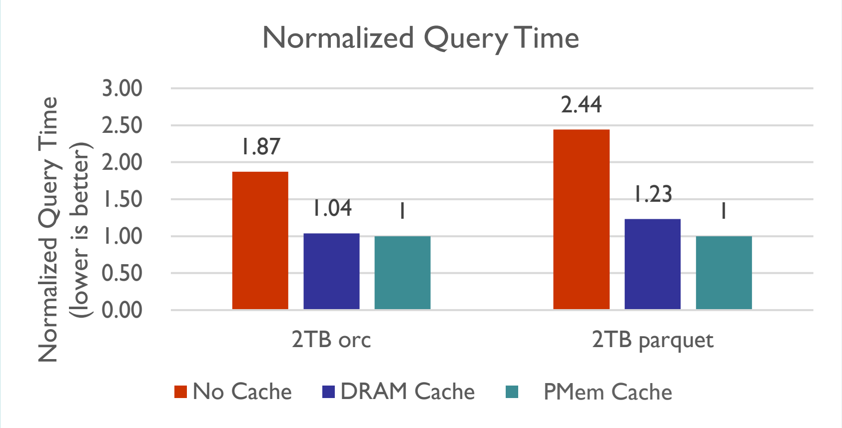
Metric

Query time

Storage Type	Cache Capacity (GB)	Parquet format case	ORC format case
PMem	918	Fully cached	Fully cached
DRAM	560	Partially cached	Partially cached
HDD (baseline)	--	--	--

Decision Support Workload (2TB raw data)





PMem Cache vs. DRAM Cache

- PMem Cache provides 2.44x (Parquet), 1.87x (ORC) speedup over no cache
- PMem Cache provides 1.23x (Parquet), 1.04x (ORC) speedup over DRAM
  PMem cached all the tables, DRAM tried best (DRAM cached more tables on ORC than on Parquet)

PMem provides higher performance boost for Decision Support Workloads on larger data set.

Performance results are based on testing as of 12/06/2019 and may not reflect all publicly available security updates. See configuration disclosure on slide for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. Configurations refer to slide titled Performance Test Configuration.

Summary

Summary

HDFS Cache

- **Centralized Cache Management**
  Supports user to explicitly (un)cache data by specifying its path and the number of replicas.
- **Cache Locality**
  Cache locality is available to upper application, so facilitates task scheduling optimization for apps.

HDFS PMem Read Cache

- **Compatibility**
  Cache Directives/API are as same as that for DRAM cache. No code change needed for upper apps.
- **Upstream Status**
  All patches have been merged to upstream. Support versions: Apache Hadoop 3.1.4, 3.2.2, 3.3.0.

HDFS PMem Read Cache Performance

- **Larger Capacity, Better Performance**
  Under same HW cost config, PMem provides larger cache capacity than DRAM. Suit large dataset.
- **Data Persistence**
  Non-volatile storage, cache warm-up time (the cache time needed after cluster restarts) is reduced significantly.

Thanks!

Credits

Thanks for Tao He for the contribution in the test content!

Please take a moment
to rate this session.

Your feedback matters to us.

Legal Information: Benchmark and Performance Disclaimers

Performance results are based on testing as of Dec. 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information, see Performance Benchmark Test Disclosure.

Configurations: see performance benchmark test configurations.

Notices and Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel, the Intel logo, Xeon, Optane, Optane PMem are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

© Intel Corporation.