PURESTORAGE®

komprise

TECHNICAL WHITE PAPER

File Migration in High File-count Environments to FlashBlade™ Using Komprise

Host-based file migration from non-Pure Storage® and FlashBlade to FlashBlade.

Contents

Introduction **3**

Data Platform Challenges **3**

FlashBlade **4**

Data Migration **4**

Komprise **5**

Datasets Validated for Migration **7**

Android Open Source Code Dataset 7

Vdbench Workload Dataset 9

SpecSFS_2020 Workload Dataset 11

Test Environments **12**

Test Results and Observations **13**

Conclusion **14**

Introduction

Business organizations always want to release their products and services to market faster. In an effort to drive faster time to market with newer innovations, you need workflows and applications to perform at higher speeds with complete automation. Transitioning legacy and new age applications that use automated and continuous pipelines requires a faster and scalable data platform to store, manage, and protect large volumes of data.

Data Platform Challenges

With the exponential data growth in modern times, data is very quickly crossing hardware boundaries in on-premises data centers. Legacy and modern applications generating and reusing data need to scale in performance beyond the limits of traditional scale-up and scale-out storage platforms. Businesses are looking for scalability and speed in the applications that consume and process data to drive faster time to market and spur new innovations to stay competitive.

Pure Storage® FlashBlade™ provides a data platform for many applications and workloads, over both file and object formats, without any disruption, downtime, or data migration. Data becomes immortal as it scales in capacity and performance while stored in FlashBlade Evergreen Storage™. Using Komprise Intelligent Data Management, migrating data from traditional scale-up and scale-out storage platforms to FlashBlade is simple and fast for on-boarding applications quickly.

This data is stored in silos with limited data continuity among workflows and workloads in traditional data centers. The following challenges also restrict the flexibility of data growth in high file-count environments that consist of millions of files with various file sizes:

- Adding more CPU/cores to the compute layer requires adding more storage although additional capacity is not a requirement.
- Capacity and linear performance scaling with high data growth is restricted with traditional scale-up storage architectures.
- CPU resources required for controller headroom and network bottlenecks are typical challenges with traditional scale-up architectures.
- Traditional scale-out architectures do not provide storage efficiency required for the high file-count datasets to support higher return on investment (ROI).
- Applications are restricted with the file-system size with traditional scale-up and scale-out architectures.

Applications that generate and access large datasets require high concurrency at scale. High metadata access with faster IOPs are the workload signature of applications used in electronic design automation (EDA) and semiconductor chip design companies. Modern silicon-on-chip (SoC) and field-programmable gate arrays (FPGA) are shrinking their form

factors by using field-effect transistors (FinFETs) and gates all around (GAA). FinFETs and GAAs datasets require higher concurrency at scale and larger file-system sizes with small to large file sizes.

FlashBlade

Modern distributed scale-out architectures like Pure Storage FlashBlade address most of the data-platform challenges for applications that generate exceedingly high file count environments. Specifically:

- FlashBlade not only supports an extraordinarily high file-system size, but supports extreme parallel access for high concurrency. Performance scales linearly.
- FlashBlade performance scales linearly with a small form factor compared to traditional scale-up and scale-out storage. This reduces the storage footprint of modern data centers.
- FlashBlade reduces the data footprint with compression where applications can generate more data with less cost. This leads to better ROI.
- Apart from data reduction, FlashBlade also improves storage efficiency by supporting variable block sizes.
- Capacity can grow linearly by adding more blades to the FlashBlade chassis. The file-system auto-balances in the newly added blades, seamlessly.

In the light of benefits that FlashBlade provides to support high file-count environments, EDA and chip-design companies and other industry verticals are quickly moving their new production workloads to FlashBlade. Doing so requires migrating the existing workloads and datasets on traditional scale-up and scale-out storage platform to FlashBlade.

Data Migration

Data migration and data replication are two sides of the same coin but have different business relevance.

- *Data replication* mainly involves copying a production dataset to a homogeneous data platform or to a specific cloud endpoint for disaster recovery (DR) or archival purposes. Applications can use the replicated copy after a failover to the DR site that holds the last replicated copy of the production dataset.
- *Data migration* involves moving a production dataset to heterogeneous data platforms and cloud endpoints to gain competitive edge and for improved scalability, efficiency, and productivity. Data migration allows newer and modern applications to switch over from traditional scale-up and scale-out storage platforms to enable operating on data platforms like FlashBlade at scale.

Data and the information derived from it are powerful and strategic. Newer applications and tools may not run to their full potential if the underlying data platform can't support the speed and the scale required to consume the data. Data migration plays a vital role in moving data from legacy storage to FlashBlade as you're updating and optimizing workflows and execution pipelines for various modern and transitioning workloads.

Network-attached storage (NAS) over Network File System (NFS) have long been in use by applications and tools in production by many EDA vendors and chip-design businesses. Files and objects as different data formats are getting more traction for modern applications. FlashBlade provides the data continuity for applications and tools by supporting

both files (NFS) and objects (S3-compatible) on one data platform. Data migration can happen between on-premises and cloud endpoints. Data migration applies to the following scenarios:

- Consolidating local storage or direct-attached storage (DAS) data silos to FlashBlade
- Moving data from legacy scale-up and scale-out storage to FlashBlade for workload scaling and cost efficiency
- Data tiering and archival from FlashBlade to cloud

Data migration can be time consuming, error prone, labor intensive, and disruptive depending on the size of the dataset and the type of migration. Many open source and commercial tools are available for migration. This paper focuses on using Komprise's capability for data migration from DAS and legacy storage devices to consolidate on FlashBlade to reduce the data-center storage footprint. Timely movement of data to FlashBlade allows applications to:

- Manage large and active datasets
- Scale on demand with respect to capacity
- Perform under massively concurrent workloads

Komprise

Komprise is an intelligent, analytics-driven data management solution for easy, reliable, fast, and efficient NAS data migrations. Komprise eliminates the cost, complexity, low performance, and low fidelity of other approaches. Simply pick one or more NFS or SMB sources and one or more NFS or SMB destinations, and Komprise manages the parallel high performance, high integrity, reliable data migration. Komprise is vendor-agnostic. Its scale-out distributed architecture consisting of VMs called Observers allows for migration at a large scale. The shares to be migrated are distributed across the Observers to enable simultaneous migration of hundreds of shares. The administrative console provides detailed feedback on each migration task including the time taken, the number of files migrated, and any errors that may have occurred. All errors are automatically re-tried to eliminate transient errors. These features make it possible to run large data-center migration tasks without excessive overhead and labor-intensive oversite.

Komprise's migration software provided in version 2.12 (available in October 2019) has been optimized to work with Pure Storage FlashBlade. Komprise uses NFS aliasing to speed up the FlashBlade data ingestion rate, thereby increasing the migration speeds to FlashBlade.

In addition, Komprise eliminates the sunk costs of traditional migration approaches. You can use the full Komprise data-management solution to maximize your use of FlashBlade. Komprise analyzes data growth and usage across your storage to find cold, inactive data, and projects the ROI of moving cold data to secondary storage such as cloud/object. Komprise also moves the data transparently based on customer-defined policies, so users continue to access the moved data as before. Komprise helps organizations reduce over 70% of costs while managing data growth.

Figure 1 shows the architecture of an NAS migration using Komprise. The easy-to-deploy, fault-tolerant, distributed architecture consists of one or more Komprise Observer virtual appliances running at the customer site. The virtual appliances are connected to a Komprise Director that provides the administrative console UI, which can run either as a cloud service or on-premises. Komprise works across on-premises NFS and SMB/CIFS storage and object/cloud storage seamlessly.

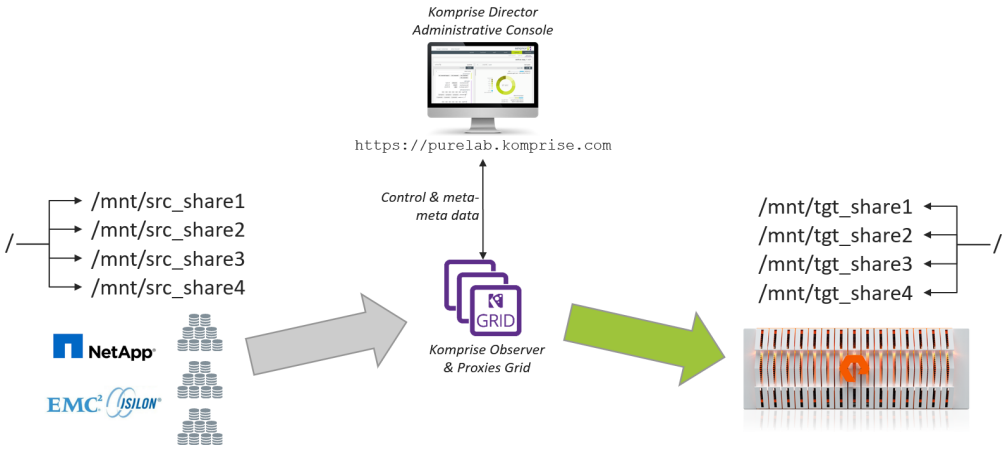


Figure 1: The Komprise data migration architecture

When performing data migrations, Komprise performs automatic retries if network or storage is unavailable. It ensures that all NTFS permissions and file data are fully migrated. Komprise scales the migration workloads adaptively to avoid interfering with active usage of your current environment. You can scale up performance as needed by increasing the number of threads in an Observer or simply adding more Observer virtual machines.

During the validation of the different high file-count datasets, Komprise provides some important optimizations to the Observers to improve the scan and migration speed of small files:

- Removed the checkpoints during migration iterations to reduce overhead. Dormant datasets may not require checkpoints during migration unless there is a loss in the network communication.
- FlashBlade provides multiple TCP connections (data virtual IP addresses, or “data VIPs”) for increased parallelism. Komprise includes the ability to configure multiple FlashBlade data VIPs to enable faster data migrations to FlashBlade (Figure 2).
- Each Observer can use up to 128 threads for improved scalability while performing different meta-operations on the source and target NFS shares during the migration process.

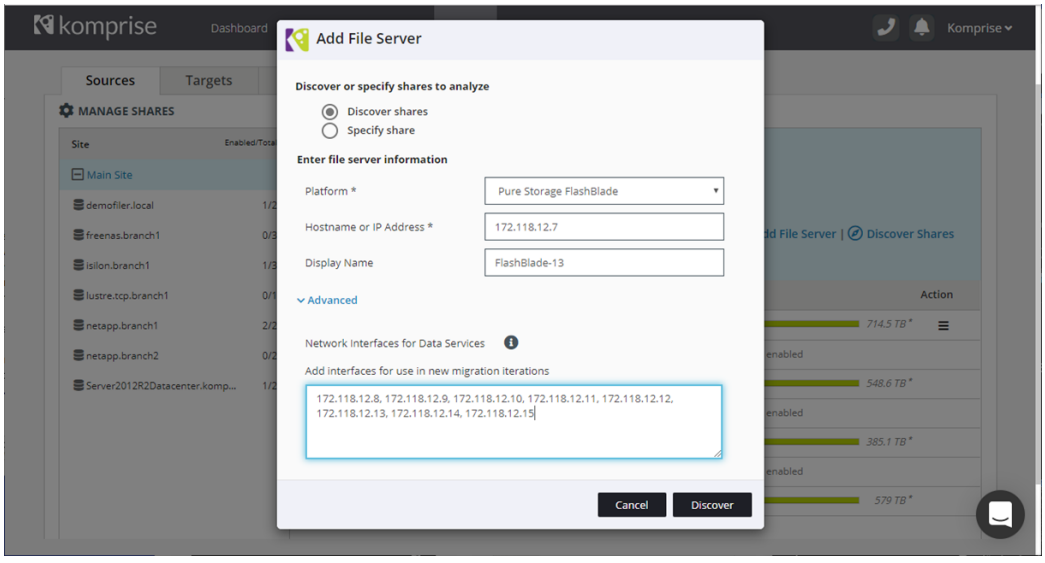


Figure 2: Configuring FlashBlade as a source in Komprise

Changes made to the Komprise system improved the scan and migration speeds for small files by up to 400%. Higher numbers of threads in each Observer enabled an increase in throughput for medium to large files. The migration time and latency reduced by 50%.

Datasets Validated for Migration

Komprise not only performs data migration but provides features like data analytics, data copy, and data tiering capabilities. This paper primarily focuses on data migrations, using different, high file-count datasets from NetApp and FlashBlade sources to a FlashBlade target. We validated three high file-count datasets using the Komprise migration function over NFS shares (Table 1): the Android open source code, a dataset generated using Vdbench, and a dataset generated using SpecSFS2020 for EDA.

	File count/ total dataset size	Directory structure	File sizes	Operational state
Android	9M/0.22TB	6 levels deep; files in every leaf directory	Mixed Small: 500B–100KB Med/Large: 10MB–1.5GB	Active (changes to the FS)
Android	9M/0.22TB	6 levels deep; files in every subdirectory	Mixed Small: 500B–100KB Med/Large: 10MB–1.5GB	Dormant (no changes to FS)
Vdbench	4.2M/0.28TB; 59M/3.9TB	5 levels deep; 500 files in leaf directories	Mixed Small: 500B–100KB Med/Large: 10MB–1.5GB	Dormant (no changes to FS)
SpecSFS	3M/3.39GB	7 levels deep; 50 and 10 files in leaf directories	Small: <1KB–8KB Large: 10MB–100MB	Dormant (no changes to FS)

Table 1: Characteristics of each dataset used in the Pure Storage Lab.

Android Open Source Code Dataset

We downloaded Android open source code from https://source.android.com/setup/build/ to the NFS share mounted from NetApp and FlashBlade source locations respectively in the test environments. The actual size of the Android source code was 74GB. We copied the Android directory multiple times in the same volume/filesystem to increase the dataset size to 0.22TB for the migration testing. “Pack” and “text” are the most common file extensions followed by archive and binaries.

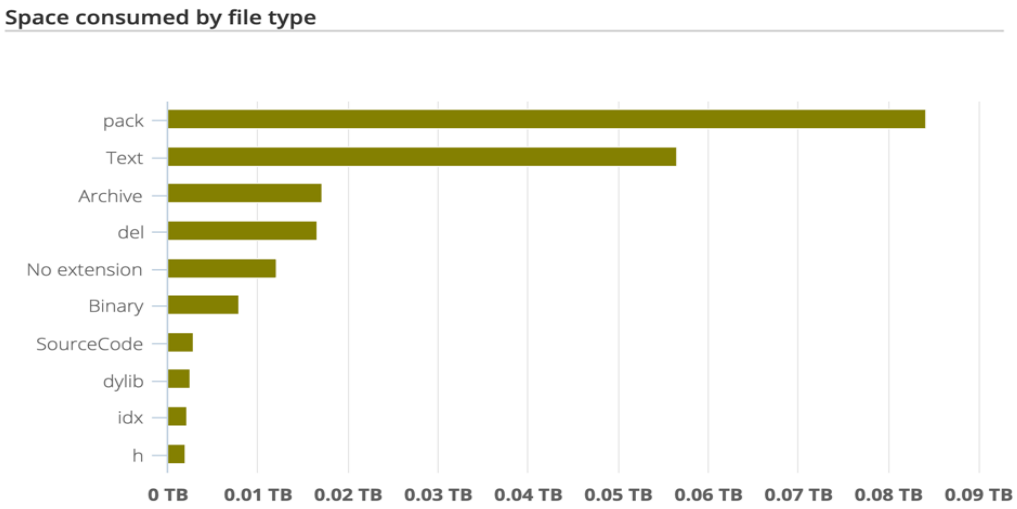
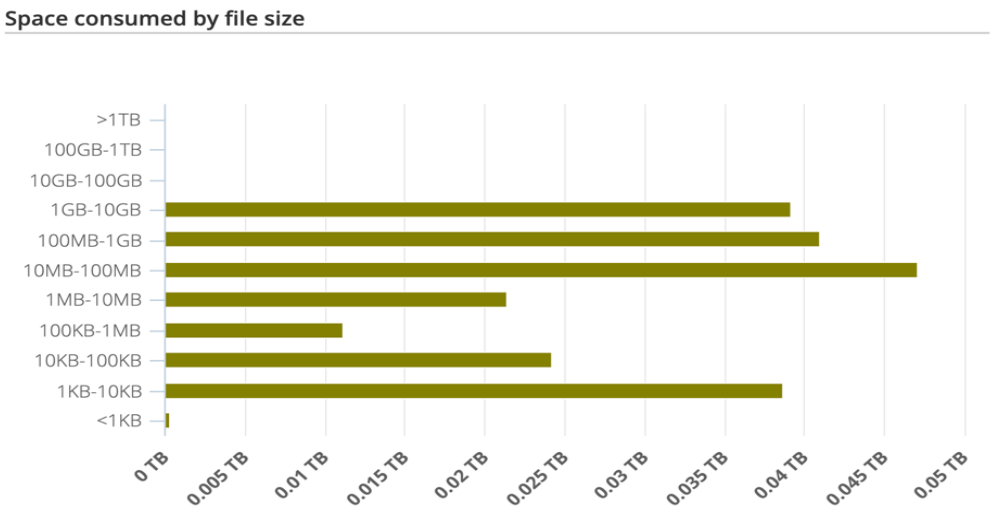


Figure 3. Type of files reported by Komprise for Android source code

We tested the Android source volume/filesystem in two ways: active and dormant. We wrote a script in the Pure Storage Lab to make changes to the Android source code while the migration was in progress to android source. The script added, modified, and renamed the files in the Android dataset during migration. In the Komprise lab, Android builds were running, creating changes that simulated an active filesystem.

The purpose of testing at both locations was to identify how Komprise handled the delta changes while migrating files to the target FlashBlade. No changes were made to the Android dataset when the dormant migration was performed.



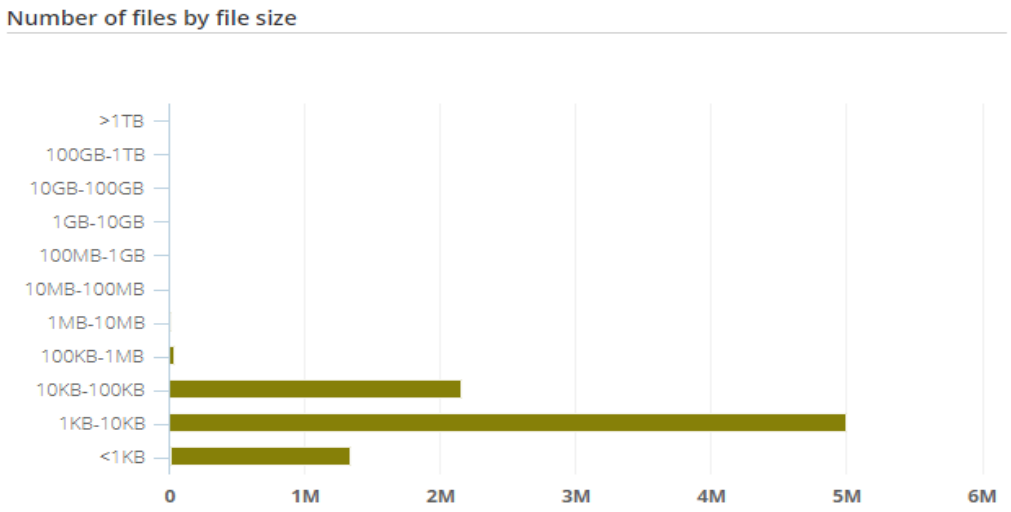


Figure 4. File sizes reported by Komprise for Android source code

The Android source code demonstrates close resemblance of many production file-system structure and file sizes. The Android source code was tested to identify how Komprise performed with small to very-small files with a total of 9M files in this dataset. As reported by Komprise in Figure 4, most of the amount of data consists of files greater than 100MB (top graph), but most of the files are small (1KB–10KB) and almost all are less than 100KB (bottom graph). The dataset included 344,058 directories.

Vdbench Workload Dataset

We downloaded the Vdbench tool from https://www.oracle.com/technetwork/server-storage/vdbench-downloads-1901681.html and used it to configure to generate 0.2TB and 4TB datasets on NetApp, and a 4TB dataset FlashBlade, both in NFS shares. Every file generated was 72KB, so all files are shown in the 10KB–100KB range in Figure 5.

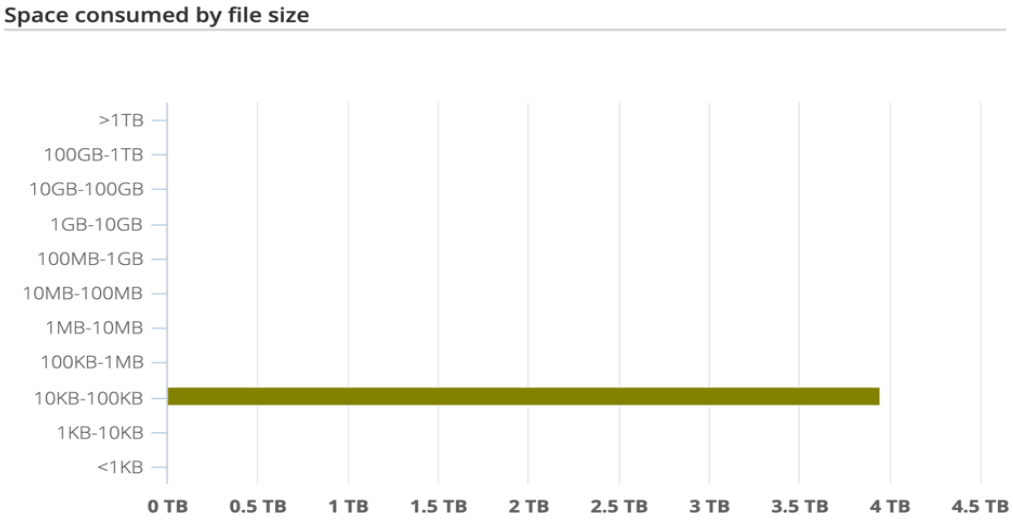


Figure 5. File sizes reported by Komprise for Vdbench dataset

The Vdbench dataset had 4.2M files in the 0.28TB dataset and 59M files in the 4TB dataset on the NetApp source, and 59M files in the 4TB dataset on a seven-blade FlashBlade source. We performed the migration for the dormant dataset without any changes to the filesystem. The directory structure for this Vdbench with 4TB dataset has five levels of directory structure. There are seven sub-directories under each directory level and 500 files in each of the sixth-level subdirectories.

```
.
├── vdb.1_1.dir
│   ├── vdb.2_1.dir
│   │   ├── vdb.3_1.dir
│   │   │   ├── vdb.4_1.dir
│   │   │   │   ├── vdb.5_1.dir
│   │   │   │   │   ├── vdb.6_1.dir
│   │   │   │   │   ├── vdb.6_2.dir
│   │   │   │   │   ├── vdb.6_3.dir
│   │   │   │   │   ├── vdb.6_4.dir
│   │   │   │   │   ├── vdb.6_5.dir
│   │   │   │   │   ├── vdb.6_6.dir
│   │   │   │   │   └── vdb.6_7.dir
│   │   │   │   ├── vdb.5_2.dir
│   │   │   │   │   ├── vdb.6_1.dir
│   │   │   │   │   ├── vdb.6_2.dir
│   │   │   │   │   ├── vdb.6_3.dir
│   │   │   │   │   ├── vdb.6_4.dir
│   │   │   │   │   ├── vdb.6_5.dir
│   │   │   │   │   ├── vdb.6_6.dir
│   │   │   │   │   └── vdb.6_7.dir
│   │   │   │   ├── vdb.5_3.dir
│   │   │   │   │   ├── vdb.6_1.dir
│   │   │   │   │   ├── vdb.6_2.dir
│   │   │   │   │   ├── vdb.6_3.dir
│   │   │   │   │   ├── vdb.6_4.dir
│   │   │   │   │   ├── vdb.6_5.dir
│   │   │   │   │   ├── vdb.6_6.dir
│   │   │   │   │   └── vdb.6_7.dir
```

SpecSFS_2020 Workload Dataset

We downloaded the SpecSFS_2020 from https://pro.spec.org/ and used the EDA modules to generate a front-end and back-end workload custom profile. The custom profile generates 3.39TB filesystem size with 3MB large files.

```
├── CL106_EDA_FRONTEND2
│   └── dir_0
│       └── dir_1
│           └── dir_2
│               └── dir_3
│                   └── dir_4
│                       └── dir_5
│                           ├── Dir0
│                           │   ├── bucket0
│                           │   ├── bucket1
│                           │   ├── bucket10
│                           │   ├── bucket11
│                           │   ├── bucket12
│                           │   ├── bucket13
│                           │   ├── bucket14
│                           │   ├── bucket15
│                           │   ├── bucket16
│                           │   ├── bucket17
│                           │   ├── bucket18
│                           │   ├── bucket19
│                           │   ├── bucket2
│                           │   ├── bucket20
│                           │   ├── bucket21
│                           │   ├── bucket22
│                           │   ├── bucket23
│                           │   ├── bucket24
│                           │   ├── bucket25
│                           │   ├── bucket26
│                           │   ├── bucket3
│                           │   ├── bucket4
│                           │   ├── bucket5
│                           │   ├── bucket6
│                           │   ├── bucket7
│                           │   ├── bucket8
│                           │   └── bucket9
```

There are six levels of subdirectories in the front-end custom profile. The last directory, “dir5,” has seven subdirectories in “Dir0, Dir1, Dir2…Dir6.” Each “DirX” has 26 buckets or subdirectories; each bucket has 50 files each. There is a mix of file sizes that range between <1KB–8KB and 10MB–100MB.

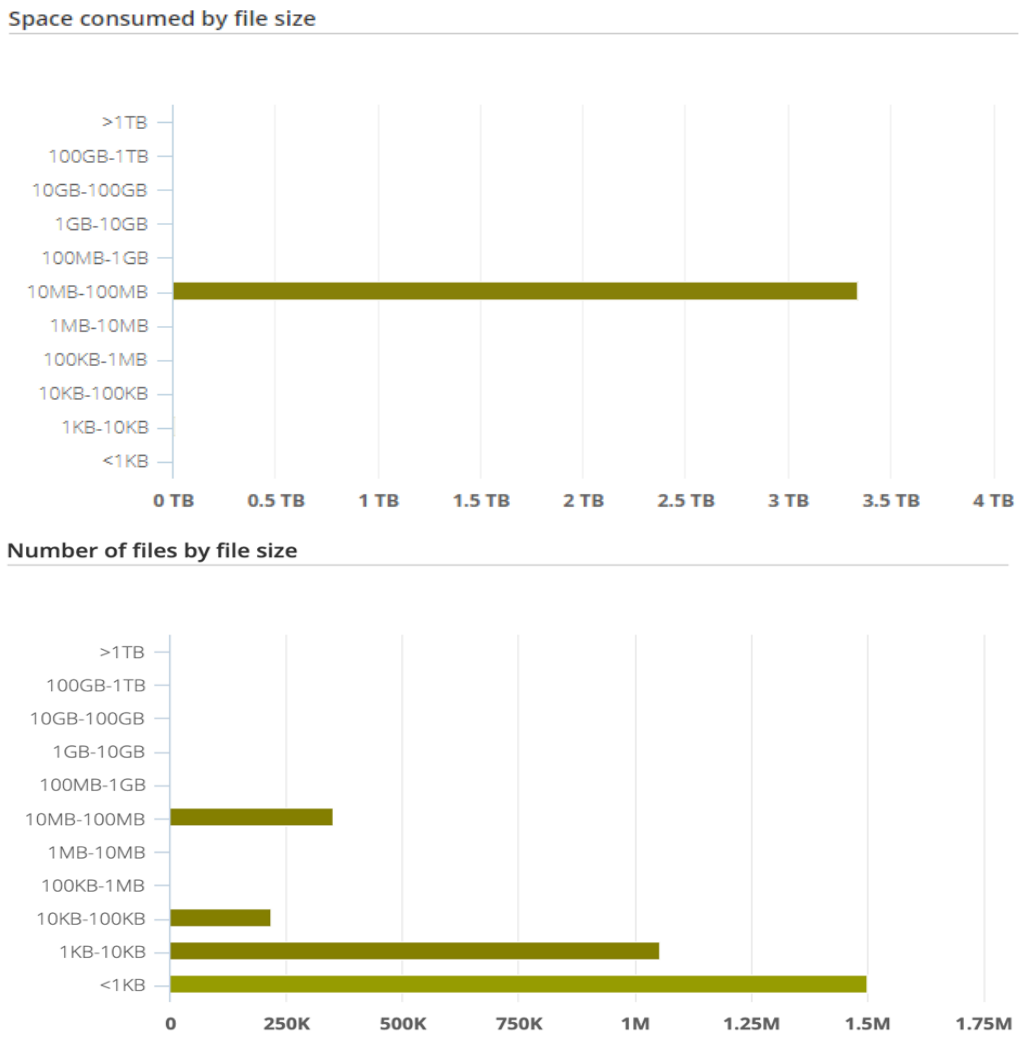


Figure 6. File sizes reported by Komprise for the SpecSFS2020 workload dataset

The back-end custom profile has a similar directory structure as the front-end, except that it has 10 files in each bucket. Most of the amount of data consists of files between 10MB–100MB (top graph, Figure 6); however, most files were smaller than 10KB (bottom graph).

Test Environments

The data migrations for active and dormant datasets were carried out in two lab locations: Pure Storage and Komprise. Pure Storage provided Komprise with the instructions to download and create the Android and Vdbench datasets. Komprise did not test the SpecSFS2020 workload dataset in its lab due to license restrictions. Komprise generated those source datasets on the NetApp storage (both AFF and spinning disks) over NFS. No file migrations were performed across the WAN.

Komprise performed the following migrations in the scenarios listed below.

- NetApp AFF with one share, using one Observer, to 15-blade Pure Storage FlashBlade

- NetApp AFF with two shares, using two Observers, to 15-blade Pure Storage FlashBlade
- NetApp spinning disks one share, using one Observer, to 15-blade Pure Storage FlashBlade

In addition, the 0.28TB Vdbench workload dataset was doubled to 0.56TB across two shares to test with two Komprise Observers on the source NetApp AFF to the target FlashBlade.

Pure Storage performed the migration of the Android, Vdbench, and SpecSFS2020 datasets in its lab.

- Seven-blade FlashBlade with one share, using one Observer, to 15-blade FlashBlade

Pure performed the tests with an active and dormant Android dataset on the NetApp and FlashBlade sources migrating to FlashBlade.

Test Results and Observations

Table 2 summarizes all the test results for the migrations carried out for the datasets in the Pure Storage and Komprise lab environments. All migrations were performed to FlashBlade targets from the sources listed.

Dataset	Source	Size	# files	File sizes	Scan speed	Scan time	Transfer thru-put	Transfer speed	Duration
Android ***active***	Pure	0.222TB	8.4M	Small, avg 28KB	5867 files/sec	24 min	56 MB/sec	2012 files/sec	70 min
Android ***dormant***	Pure	0.222TB	8.4M	Small, avg 28KB	5867 files/sec	24 min	54 MB/sec	1956 files/sec	72 min
SpecSFS ***dormant***	Pure	3.346TB	3.1M	Large, avg 1MB	3472 files/sec	15 min	464 MB/sec	413 files/sec	126 min
Vdbench ***dormant***	Pure	3.945TB	58.8M	Medium, 72KB	5801 files/sec	169 min	158 MB/sec	2249 files/sec	436 min
Vdbench ***1 share, 1 Observer***	NetApp fast	0.282TB	4.2M	Medium, 72KB	11672 files/sec	6 min	90 MB/sec	1273 files/sec	55 min
Vdbench ***1 share, 1 Observer***	NetApp fast	3.945TB	58.8M	Medium, 72KB	11699 files/sec	84 min	90 MB/sec	1280 files/sec	766 min
Vdbench ***2 shares, 2 Observers***	NetApp fast	0.563TB	8.4M	Medium, 72KB	23343 files/sec	6 min	145 MB/sec	2059 files/sec	68 min
Android ***active***	NetApp slow	1.801TB	26.6M	Small, avg 28KB	3928 files/sec	113 min	38 MB/sec	539 files/sec	824 min

Table 2: Test results for the migrations carried out for the listed datasets.

Notes on Table 2:

- Scan speed and scan time measure the performance of Komprise to scan the dataset for changes without transferring any data.
- Transfer throughput and transfer speed measure the rates at which Komprise can transfer data.
- Duration measures the time to complete the migration and depends on how many copy iterations were performed (more iterations for active datasets).

The following are the observations from the results listed above.

- Migration performance for the common datasets (Android and Vdbench) is much higher for FlashBlade-to-FlashBlade migrations compared to NetApp-to-FlashBlade, due to the comparatively slower performance of the NetApp.
- In the course of developing the results, Komprise implemented optimizations to work with FlashBlade that enabled a significant improvement in scan speed and transfer rate of small files within the data center. Those optimizations also reduced migration time by 50% on FlashBlade.
- The optimized scan time helps to improve the overall migration time after the first copy iteration until final cut-over.
- Komprise can parallelize migrations by handling multiple shares with multiple Observers. Performance increases by 60% when adding a second share and Observer.

Conclusion

Organizations planning to migrate data from traditional scale-up and scale-out storage platforms to FlashBlade can now use Komprise as the migration tool of choice. Migrating data using Komprise from NAS shares on any Komprise-qualified storage to FlashBlade provides adaptive scaling and speed during the data mobility process. Komprise has optimized the migration capability to integrate FlashBlade features more natively in order to handle the migration without complex configurations. The variety of test datasets used for data migration using Komprise reduced the migration time to move the data to FlashBlade while preserving both data integrity and all file metadata and permissions.

Next Steps

Learn more about Pure FlashBlade: https://www.purestorage.com/products/flashblade

© 2019 Pure Storage, Inc. All rights reserved. Pure Storage, Pure1, Pure1 Meta, the P Logo, DirectFlash, Evergreen, FlashBlade and FlashStack, and ObjectEngine are trademarks or registered trademarks of Pure Storage, Inc. in the U.S. and other countries. All other trademarks are registered marks of their respective owners.

The Pure Storage products and programs described in this documentation are distributed under a license agreement restricting the use, copying, distribution, and decompilation/reverse engineering of the products. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors, if any. Pure Storage may make improvements and/or changes in the Pure Storage products and/or the programs described in this documentation at any time without notice.

THIS DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041

October 2019