

TECHNICAL WHITE PAPER

Fast SQL Server Backup and Restore with FlashBlade®

High-performance storage for SQL backups.

Contents

- Introduction 3**
 - How to Use This Guide..... 3
- Microsoft SQL Server 3**
- Pure Storage FlashBlade 3**
 - The FlashBlade Platform 5
- Solution Architecture..... 6**
- Test Environment Overview..... 7**
- System Configuration Details 8**
 - Microsoft SQL Server..... 8
 - FlashArray Configuration 8
 - FlashBlade Configuration..... 9
- Test Cases and Results 10**
 - Test Methodology 10
 - Single Database Tests..... 11
 - Impact of Number of Source Devices 11
 - Impact of Number of Backup Target Files 12
 - Impact of Database Tunables: MAXTRANSFERSIZE 12
 - Impact of Database Tunables: BUFFERCOUNT 13
 - Impact of Backup Compression..... 13
 - Backup Verification Performance..... 14
 - Analysis of Single Server Results 15
- Multi-Server Tests 15**
- Best Practices 16**
- Conclusion 17**



Introduction

Backup and recovery have changed. Traditional backup architectures have been focused on protecting data as fast as possible with recovery being nothing more than an afterthought. As digital transformation accelerates, organizations are laser-focused on eliminating downtime for mission-critical applications to get the most value out of their data. Fast recovery technology is required to drive this additional value. Pure Storage® FlashBlade, with its all-flash, scale-out architecture, is the ideal solution for the Rapid Restore of mission-critical applications.

How to Use This Guide

This document will show how FlashBlade delivers on the promise for fast recovery with mission-critical applications such as SQL Server. This technical white paper is for IT, storage, and DBA specialists using SQL Server native backup and recovery features. It provides guidelines and techniques to optimize backup and restore performance with SQL Server and FlashBlade. Although the focus is on Microsoft SQL Server deployed on physical servers in our testing, the use cases and corresponding best practice recommendations shown here are easily applicable to virtualized SQL Server instances deployed on hypervisors such as VMware vSphere, and Microsoft Hyper-V.

Microsoft SQL Server

Microsoft SQL Server, a relational database management system, is a primary data store for many organizations. As the amount of data increases from a variety of sources, SQL Server database administrators have less time available to perform the essential tasks of backing up data and routine backup testing validation. While traditional backup appliances are designed to efficiently store data, they are notoriously slow at rehydrating and recovering data. The data restore process is utilized more than ever before as DBAs and software developers iterate on their code with up-to-date databases to accelerate the software development process. Pure Storage FlashBlade helps reduce both backup and restore time by providing high bandwidth performance and simple management. Legacy backup solutions place businesses at risk of missing recovery and access service level agreements (SLAs) for their critical production databases. Pure Storage offers a modern rapid recovery architecture, tapping the power of scale-out all-flash storage to restore data rapidly, meet SLAs, and save money.

Pure Storage FlashBlade

Pure Storage FlashBlade is a scale-out, all-flash storage system, powered by a distributed file system that is purpose-built for massive concurrency across all data types. FlashBlade is a unified fast file and object (UFFO) storage platform that can simultaneously host multiple file systems and multi-tenant object stores for thousands of clients. FlashBlade can scale up to multi-petabyte capacity with linear-scale performance, simply by adding a single blade at a time, up to 150 blades. Because of its native scale-out architecture and ability to drive performance for any type of workload, it is considered a data hub that enables enterprises to consolidate a range of workloads, from backup to analytics and AI, on a single platform.



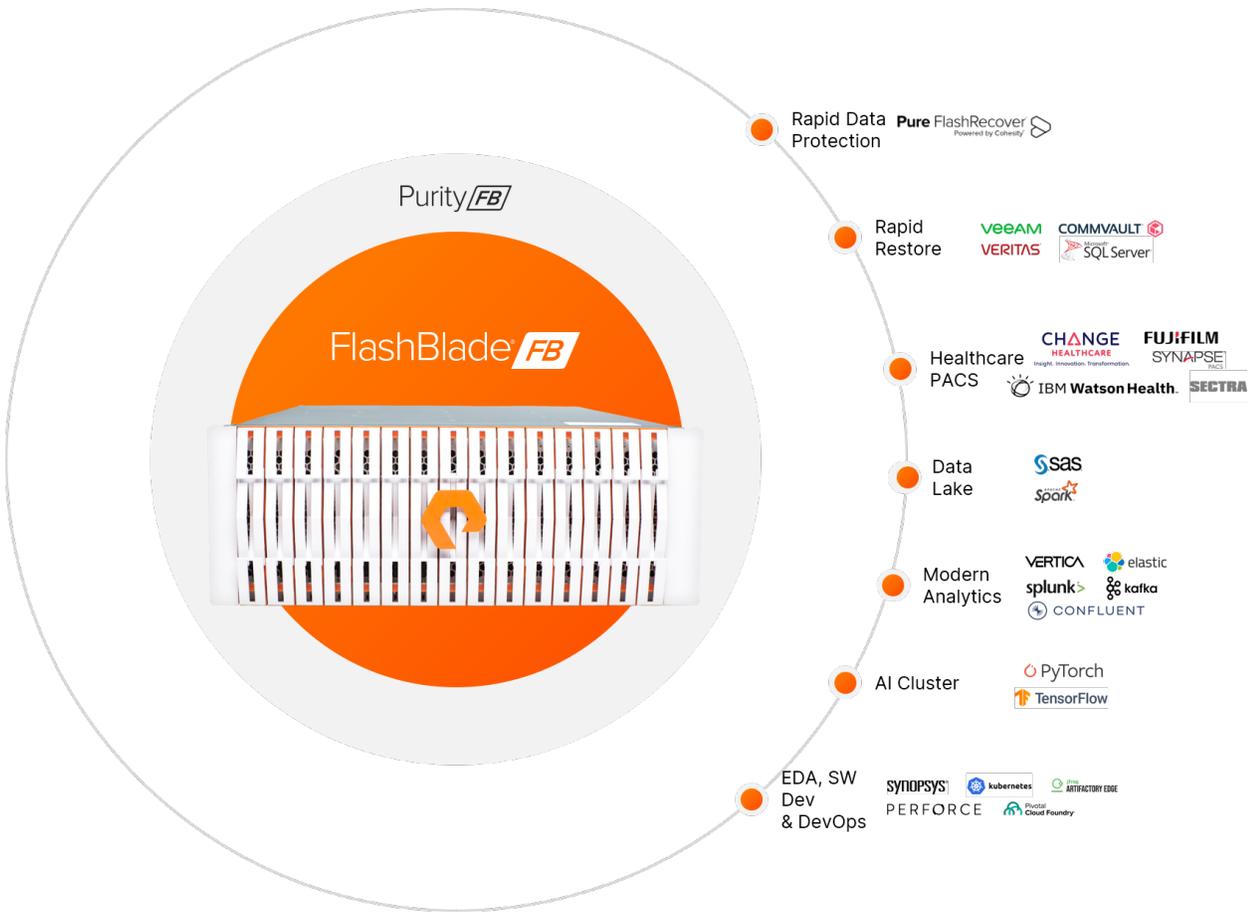


Figure 1: The FlashBlade Unified Fast File and Object storage platform.

Many organizations build their data protection strategy with FlashBlade, enjoying rapid backup and restore performance while investing in a platform that enables them to consolidate data lakes and other data silos. A FlashBlade system’s ability to scale performance and capacity is based on six key innovations:

High-performance Storage: FlashBlade maximizes the advantages of an all-flash architecture by storing data in storage units instead of crippling, high-latency storage media such as traditional spinning disks and conventional solid-state drives. The integration of scalable NVRAM into each storage unit helps scale performance and capacity proportionally when new blades are added to a system.

Unified Network: A FlashBlade system consolidates high communication traffic between clients and internal administrative hosts into a single, reliable high-performing network that supports both IPv4 and IPv6 client access over Ethernet links up to 100GB/s.

Purity//FB Storage Operating System: Purity//FB is a symmetrical operating system running on FlashBlade’s fabric modules. It minimizes workload balancing problems by distributing all client operation requests among the blades on FlashBlade.

Common Media Architectural Design for Files and Objects: FlashBlade’s single underlying media architecture natively supports concurrent access to files via a variety of protocols such as NFSv4.1, NFS over HTTP, and SMB and objects via S3 across the entire FlashBlade configuration.



High performance Replication of Objects and File: The latest version of Purity delivers replication of FlashBlade files and Objects to enable Disaster recovery either to a secondary site or the public cloud.

Simple Usability: Purity//FB on FlashBlade alleviates system management headaches as it simplifies storage operations by performing routine administrative tasks autonomously. With a robust operating system, FlashBlade is capable of self-tuning and providing system alerts when components fail.

The FlashBlade Platform



Figure 2: FlashBlade's ability to scale. A single system can scale from seven to 150 blades and is cloud optimized.

The FlashBlade Platform

A full FlashBlade system configuration consists of up to 10 self-contained rack-mounted chassis interconnected by high-speed links to up to four external fabric modules (XFM). At the rear of each chassis, two on-board fabric modules provide high-speed Ethernet interconnects to the blades, other chassis, and clients using TCP/IP. Fabric modules are interconnected, and each contains a control processor and Ethernet switch ASIC. For reliability, each chassis is equipped with redundant power supplies and cooling fans.

The front of each chassis holds up to 15 blades for processing data operations and storage. Each blade assembly is a self-contained compute module equipped with processors, communication interfaces, and either 17TB or 52TB of flash memory for persistent data storage. The current FlashBlade system can support more than 1.5 million NFSv3 getattr operations per second, or >17 GiB/sec of 512KiB reads or >8 GiB/sec of 512KiB overwrites on a 3:1 compressible dataset in a single 4U chassis with 15 blades. FlashBlade can scale both compute and performance up to a maximum configuration of 10 x 4U chassis with 150 blades.



FlashBlade Blades

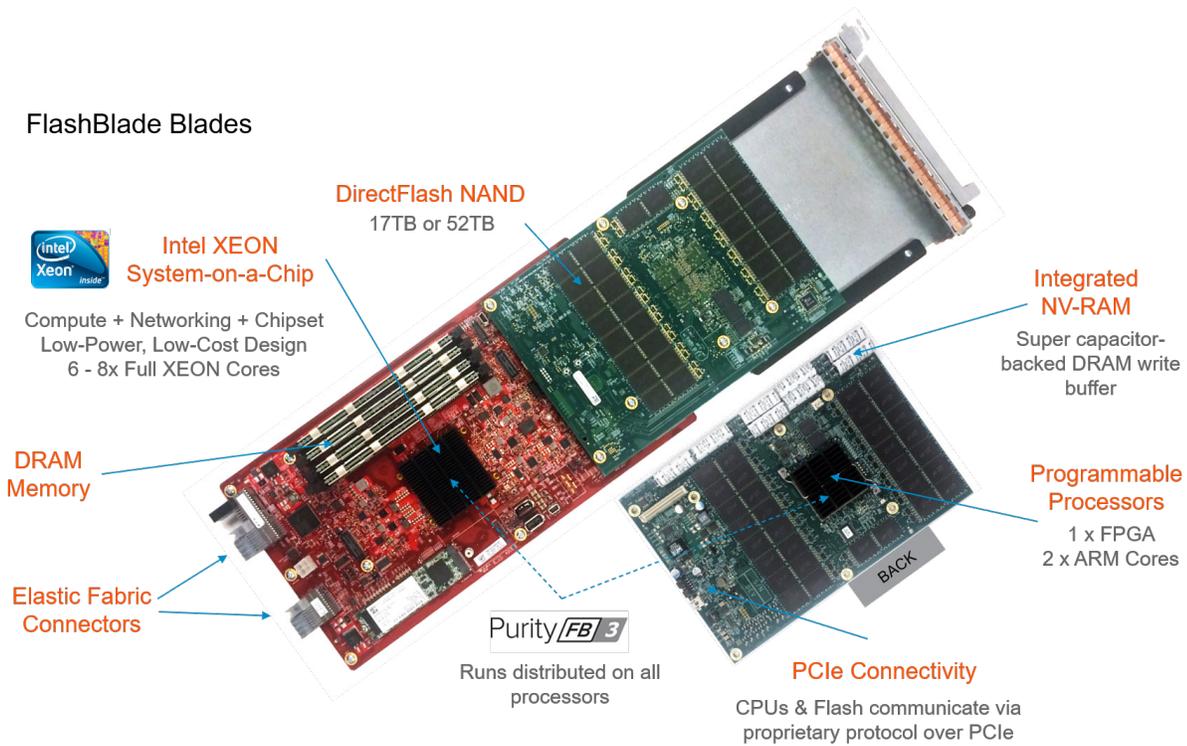


Figure 3: Anatomy of a FlashBlade.

Solution Architecture

Figure 4 represents the high-level architecture with FlashBlade. The primary database is deployed on Pure Storage FlashArray in all our testing. The best practices we outline in this document are also applicable to SQL Server databases protected by FlashBlade and running on any type of storage.

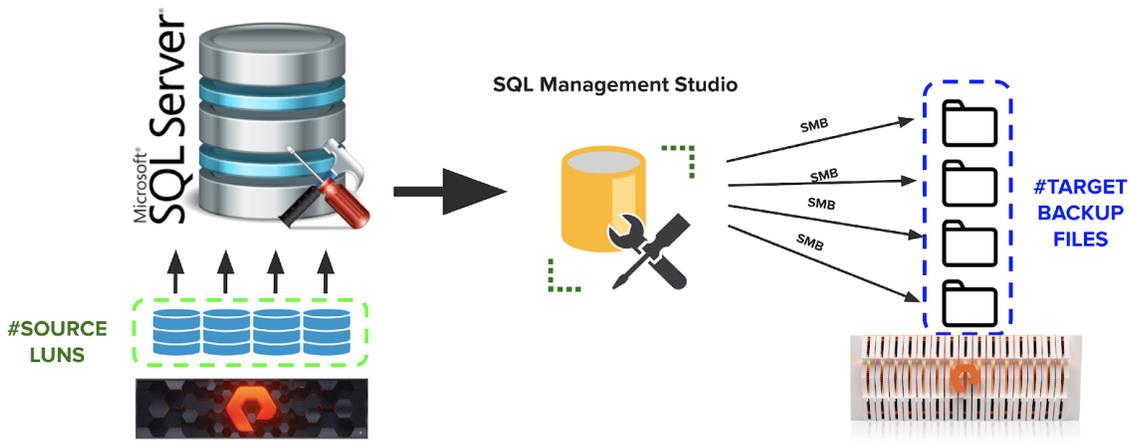


Figure 4: Rapid Restore logical architecture.



Test Environment Overview

Figures 5 and 6 illustrate the FlashBlade test environment used for this white paper. It was designed for testing the performance and scalability of FlashBlade and to determine the best practices for SQL Server backup deployment. In this environment, FlashBlade is configured to store SQL Server backups in SMB volumes. The SQL Management Studio was used to initiate all backup and restore operations. In multi-server testing, a total of eight databases were used, each on its own instance and server.

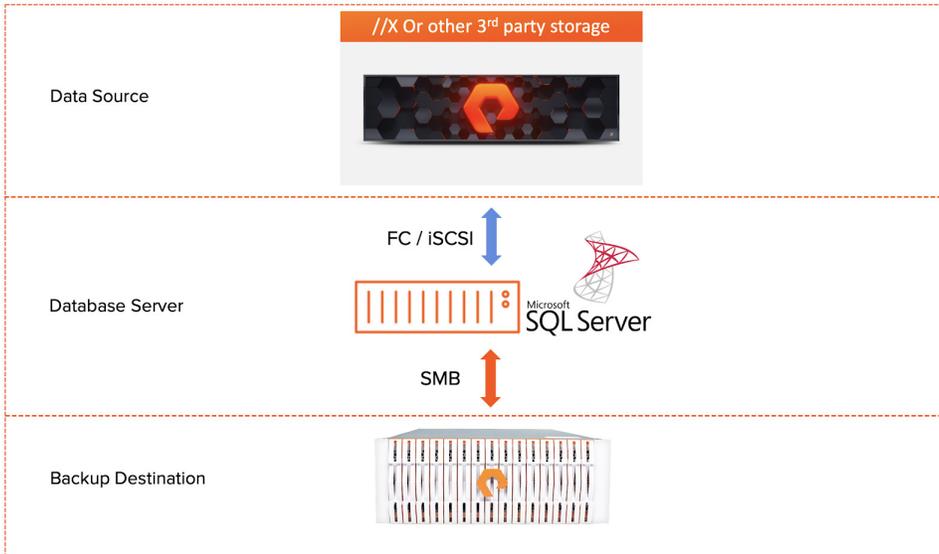


Figure 5: Single server test environment.

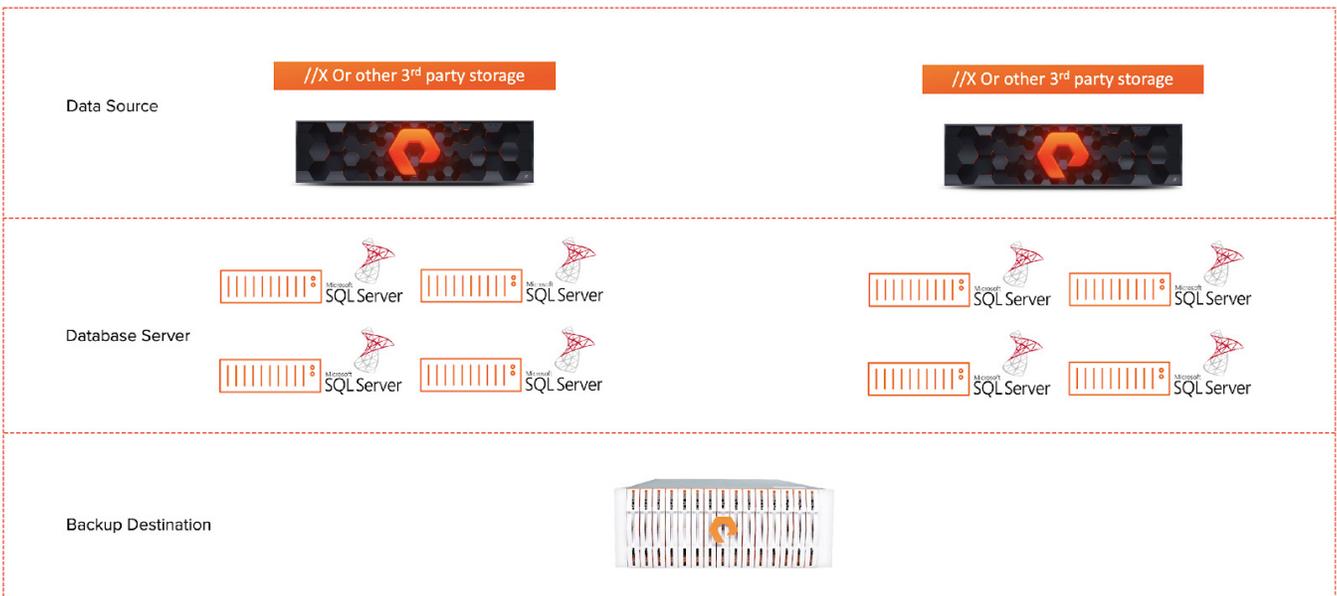


Figure 6: Multiple server test environment.



System Configuration Details

Microsoft SQL Server

Each database server comprises of the following components:

- Compute
 - 2 x Intel Xeon E5-2697 v2 @ 2.70GHz (24 physical cores total), HyperThreading enabled.
 - 768GB RAM
- Network
 - 2 x Mellanox ConnectX-3 family network adapter (40Gbps)
 - 2 x Emulex LPe32002-M2 Fibre Channel Adapter (32Gbps)
- Software
 - Windows Server 2019 Standard
 - Microsoft SQL Server Enterprise (64-bit), Version 15.0.2070.41

FlashArray Configuration

Table 1 shows the configuration of the two Pure FlashArrays used for hosting the SQL Server databases in this solution testing.

Component	FlashArray 1	FlashArray 2
FlashArray	//X70R2	//X70R3
Capacity	174.45TB raw (base chassis)	
Connectivity	4 X 32GB/S redundant Fibre Channel ports 1GB/S redundant Ethernet (management port)	
Software	Purity//FA 6.0.3	

Table 1: FlashArray configuration.

Each FlashArray hosts data volumes corresponding to four SQL Server databases. Each SQL Server database is created on a set of nine LUNs, with one volume for logs and eight volumes for data, as shown in Figure 7. The test database was created on top of a single FileGroup spanning across all eight data LUNs. While having multiple LUNs is not required to achieve good database performance on a FlashArray, more LUNs were used to ensure enough concurrency during database backup and restore operations. The SQL Server launches one backup reader thread for each logical volume that is hosting data files for the source database.



Array	Hosts	Volumes	Pods	File Systems	Policies	
> Hosts > Britannicus-F04-03						
Size	Data Reduction	Unique	Snapshots	Shared	System	Total
17 T	5.8 to 1	189.59 G	0.00	-	-	189.59 G
Connected Volumes ^ 1-9 of 9						
Name	Shared	LUN				
B-Data01	False	4	✕			
B-Data02	False	6	✕			
B-Data03	False	8	✕			
B-Data04	False	5	✕			
B-Data05	False	7	✕			
B-Data06	False	3	✕			
B-Data07	False	2	✕			
B-Data08	False	1	✕			
B-Log	False	9	✕			

Figure 7: Storage Layout of individual database.

FlashBlade Configuration

FlashBlade acts as a backup target for SQL Server backups, using the fast, native SMB protocol to receive the backup data. Unlike traditional SMB implementations, the SMB protocol stack on FlashBlade is designed from the ground up to leverage the power of the Flash storage for very high-performance scalability. The FlashBlade used in testing is configured as shown in Table 2.

Component	FlashArray 1
FlashBlade	30 X 17TB Blades (multi-chassis system)
Capacity	527.77TB Raw
Connectivity	4 X 100Gbps Ethernet (Data) 2 x 1Gb/S Redundant Ethernet (Management Port)
Physical	10U
Software	Purity//FB 3.2.x

Table 2: FlashBlade configuration.



A total of eight file systems were created for storing SQL Server backups as shown in Figure 8. To maximize IO bandwidth, a total of eight virtual interfaces (VIPs) were created on FlashBlade as shown in Figure 9.

Name	Source Location	Source Name	Size	Used	Hard Limit	Created	Protocols	Promotion Status	Writable
SQL_Backup_SMB0					All	All			All
SQL_Backup_SMB01	-	-	10 T	519.72 G	False	2020-12-01 07:29:09	SMB	promoted	True
SQL_Backup_SMB02	-	-	10 T	522.08 G	False	2020-12-01 07:29:22	SMB	promoted	True
SQL_Backup_SMB03	-	-	10 T	518.55 G	False	2020-12-01 07:29:36	SMB	promoted	True
SQL_Backup_SMB04	-	-	10 T	524.55 G	False	2020-12-01 07:29:55	SMB	promoted	True
SQL_Backup_SMB05	-	-	10 T	52013 G	False	2020-12-01 07:30:17	SMB	promoted	True
SQL_Backup_SMB06	-	-	10 T	52700 G	False	2020-12-01 07:30:34	SMB	promoted	True
SQL_Backup_SMB07	-	-	10 T	522.20 G	False	2020-12-01 07:30:47	SMB	promoted	True
SQL_Backup_SMB08	-	-	10 T	541.34 G	False	2020-12-01 07:31:01	SMB	promoted	True

Figure 8: FlashBlade SMB volume configuration.

								data1	10.21.236.235	data	☑ ☒
								data2	10.21.236.236	data	☑ ☒
								data3	10.21.236.237	data	☑ ☒
								data4	10.21.236.238	data	☑ ☒
☑ ☒	VLAN2236	True	10.21.236.0/24	2236	10.21.236.1	9000	uplink	data5	10.21.236.239	data	☑ ☒
								data6	10.21.236.76	data	☑ ☒
								data7	10.21.236.77	data	☑ ☒
								data8	10.21.236.78	data	☑ ☒
								+ Add interface			

Figure 9: FlashBlade Network configuration.

Test Cases and Results

Test Methodology

The SQL Server database provides several options for backing up and restoring databases. The scope of this whitepaper is to characterize the performance behavior of a full set of backup and restore options, which often are the most performance-demanding and crucial for business continuity. Traditional backup appliances that are optimized for space with some form of deduplication technology struggle to meet the SLAs for database restore and verification operations.

The performance of a full database backup and restore operations mainly depend on the following variables:

- **Layout of the Source Database:** SQL Server allocates one backup reader thread for every physical device associated with the database being backed up.
- **Layout of the Backup Target:** SQL Server allocates one backup writer thread for every file on the backup target. Spreading the backup across multiple files and mount points to achieve maximum performance.
- **Tunable Parameters:** Several tunable parameters can impact the performance of backup and restore operations such as MAXTRANSFERSIZE, BUFFERCOUNT, BLOCKSIZE, etc. We strongly encourage customers to calibrate their systems to find optimal settings that meet their requirements. This whitepaper calls out all the parameters used in our testing.



- **Compression:** SQL Server supports compressing the backup files. This saves storage space requirements for backups and puts less load on networks during backup and recovery operations, but at the expense of database host CPU cycles. This whitepaper covers tests conducted with and without compression enabled.

The testing discussed in this paper revolves around characterizing the performance of database backup and recovery by tuning the above parameters. The testing also includes database consolidation use cases. It is common for enterprise customers to have 10s to 100s of SQL Server databases.

Single Database Tests

The main objective of single database tests is to define the best practices to achieve the highest backup and restore performance. We used a SQL Server database of 4TB deployed on a FlashArray //X70R3. The database was created in its own Filegroup with eight database files unless otherwise indicated. The log file was always created on its own LUN.

Impact of Number of Source Devices

The effective backup speed of a SQL Server database can depend upon the number of source devices on which the database is deployed as SQL Server launches one backup reader thread for every physical device. In this test, all other backup parameters are kept constant while varying the number of LUNs of the source database. An identical 4TB sized database was created each time on one, two, four, and eight LUNs to measure database backup performance. The database was backed up to a 'NUL' target to eliminate any backup target performance dependency.

```
BACKUP DATABASE TestDB  
TO DISK = 'NUL'  
WITH MAXTRANSFERSIZE= 524288, BUFFERCOUNT= 512
```

Figure 10 shows the impact of the number of source LUNs for database backup performance. The results are quite contrary to the common belief that more source devices yield better performance. The FlashArray was able to provide almost similar performance irrespective of the number of LUNs used for database deployment. This may not be true with competitive storage arrays and we strongly recommend customers experiment before deciding on a certain layout for their source database. For all FlashArray deployments, having more than two LUNs to deploy SQL Server databases does not seem to yield any sizable performance benefits.

Backup Rate (TB/hr) vs. #Devices

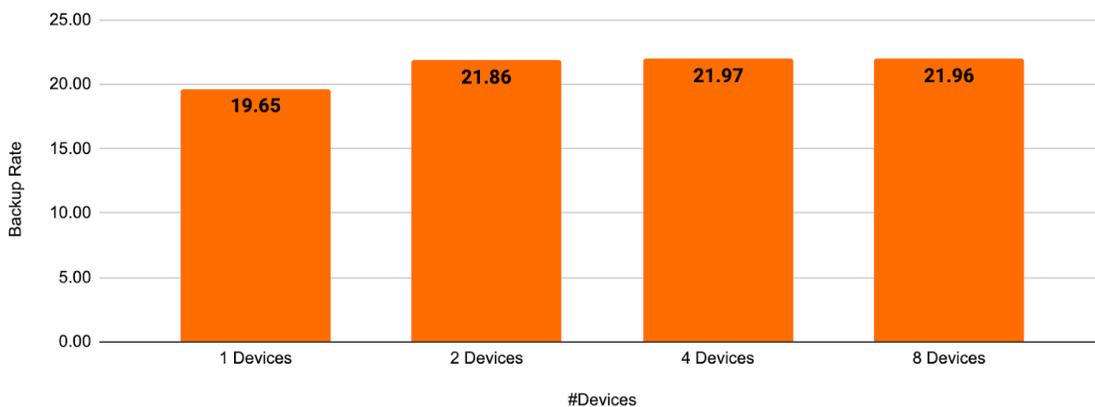


Figure 10: Backup Performance vs. Number of Source database devices.



Impact of Number of Backup Target Files

The effective backup speed of a SQL Server database depends upon the number of target files used for backup as SQL Server launches one backup writer thread for every backup file used. In this test, all other backup parameters are kept constant while varying the number of backup files. The same 4TB database deployed on eight source LUNs was backed up to one, two, four, and eight backup files. The FlashBlade tries to assign each backup file stream to one of its blades using an in-built load balancing algorithm from front-end processing. The load-balancing algorithm uses unique hashes generated from the source IP address, port, destination IP address, and ports. So, creating multiple Virtual IP addresses (VIPs) is recommended on FlashBlade as shown in Figure 9 to achieve maximum performance. There is no requirement to create multiple subnets while using multiple VIPs. All the VIPs can reside in the same subnet.

```
BACKUP DATABASE TestDB
TO DISK = '\\data-vip01\SQL_Backup_SMB01\D\TestDB_4TB_8files_01.Bak',
DISK = '\\data-vip02\SQL_Backup_SMB02\D\TestDB_4TB_8files_02.Bak',
DISK = '\\data-vip03\SQL_Backup_SMB03\D\TestDB_4TB_8files_03.Bak',
DISK = '\\data-vip04\SQL_Backup_SMB04\D\TestDB_4TB_8files_04.Bak',
DISK = '\\data-vip05\SQL_Backup_SMB05\D\TestDB_4TB_8files_05.Bak',
DISK = '\\data-vip06\SQL_Backup_SMB06\D\TestDB_4TB_8files_06.Bak',
DISK = '\\data-vip07\SQL_Backup_SMB07\D\TestDB_4TB_8files_07.Bak',
DISK = '\\data-vip08\SQL_Backup_SMB08\D\TestDB_4TB_8files_08.Bak'
WITH MAXTRANSFERSIZE= 524288, BUFFERCOUNT=512
```

Figure 11 shows the backup performance scalability as we scaled the number of mount points. In this test, one backup file per mount point was used as having more files per single mount point did not yield any additional performance.

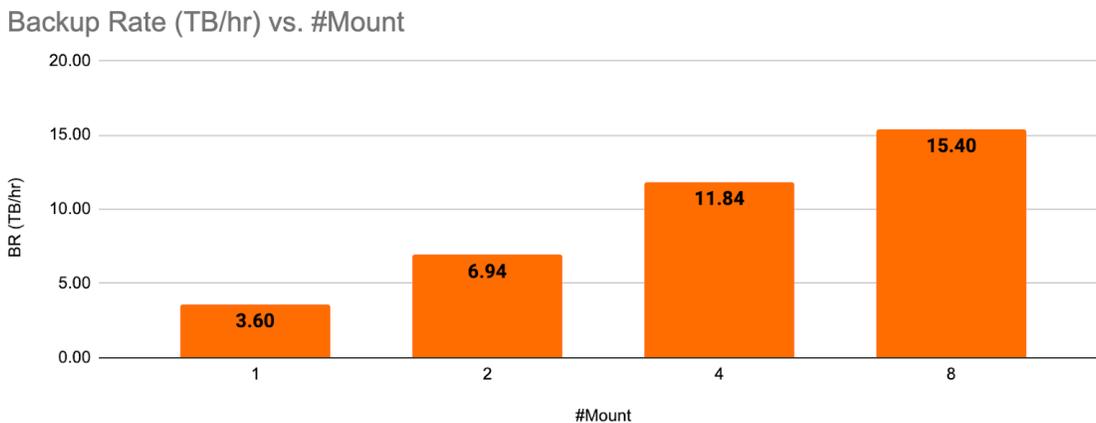


Figure 11: Backup Performance vs. Number of target mount points

Impact of Database Tunables: MAXTRANSFERSIZE

The SQL Server database provides lots of tunable parameters to optimize backup and restore performance. There is no single tunable set of parameters that suits every customer scenario. So, we strongly encourage customers to perform calibration tests to find the optimal setting for their deployment. In this test, the MAXTRANSFERSIZE parameter was varied while keeping every other parameter constant. The same 4TB database deployed on eight source LUNs was backed up to eight backup files with 128 BUFFERCOUNT in all our tests while varying with MAXTRANSFERSIZE to 128KB, 256KB, 512KB, and 1024KB. Figure 12 shows changes in performance as we changed the MAXTRANSFERSIZE value. The optimal value seemed to be around 512KB as per results with a FlashBlade system.



Backup Rate (TB/hr) vs. MAXTRANSFERSIZE

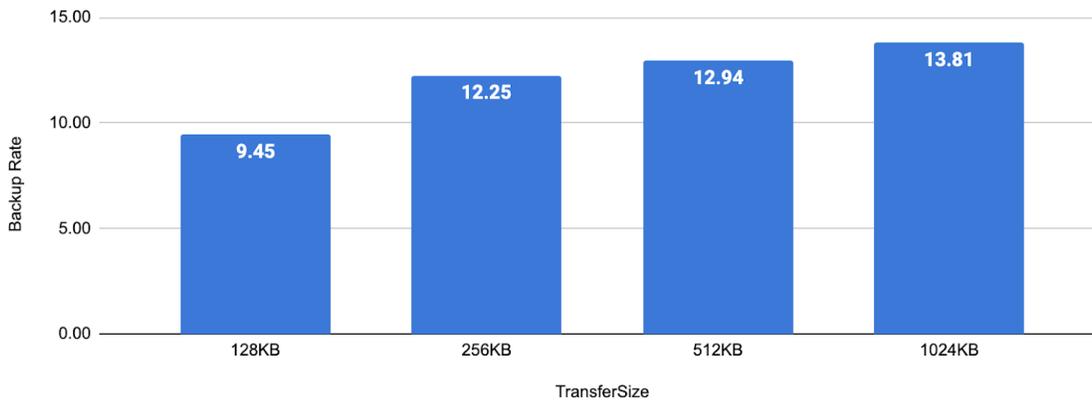


Figure 12: Backup Performance vs. Number of target mount points.

Impact of Database Tunables: BUFFERCOUNT

In this test, the BUFFERCOUNT parameter was varied while keeping every other parameter constant. The same 4TB database deployed on eight source LUNs was backed up to eight backup files with 512K MAXTRANSFERSIZE in all our tests while varying BUFFERCOUNT to eight, 32, 128, and 512. Figure 13 shows performance scalability as the BUFFERCOUNT parameter was scaled. The optimal value seems to be around 512 as per results with a FlashBlade system. Our experiments with values higher than 512 for BUFFERCOUNT did not yield any better results. The BUFFERCOUNT tunable parameter goes together with MAXTRASFERSIZE as SQL Server sets aside the amount of memory equal to the product of these values, i.e. (BUFFERCOUNT * MAXTRASFERSIZE) for buffering purposes. Use caution, as choosing the wrong values for these parameters can cause out-of-memory errors.

Backup Rate (TB/hr) vs. BUFFERCOUNT

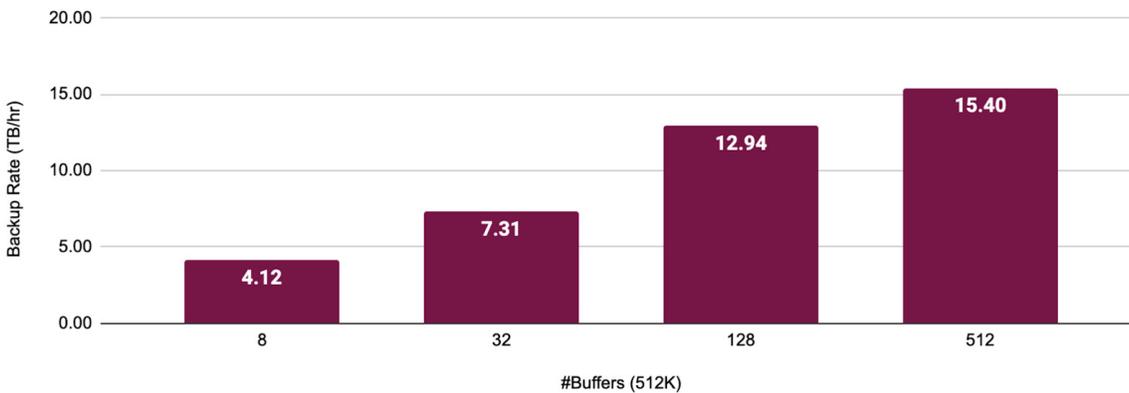


Figure 13: Backup Performance vs. BUFFERCOUNT.

Impact of Backup Compression

It is a very common practice to compress SQL Server backups to minimize storage requirements or to reduce network load or both. While the backup compression can save storage space, it also comes at a cost of host CPU cycles. So, it is always a trade-off for customers and there is no one recommendation that will suit all customers. FlashBlade has built-in native compression and data is always stored in a compressed format to optimize Flash Storage. Our tests indicate that the native compression on FlashBlade achieves almost similar storage savings for SQL Server backups while preserving expensive host CPU cycles for other useful tasks as shown in Figure 14 and summarized here in Table 3.



Metric	Compression Off	Compression On
Host CPU (%) during backup	5%	80%
Host CPU (%) during restore	5%	67%
Space consumed by backup on file system (TB)	4.04	1.04
Space consumed by backup on storage array (TB) (post-array compression)	1.32	1.03

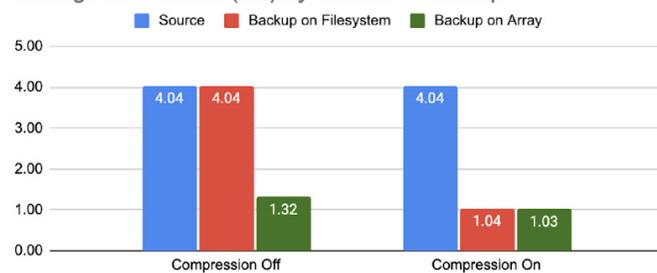
Table 3: Backup Compression metrics.

The actual flash storage consumed on the storage array for a full backup of a 4TB SQL Server database was 1.03TB vs. 1.32TB with and without host-level backup compression, respectively. The corresponding host CPU utilization was 80% vs. 5%. The native compression on FlashBlade was able to provide most capacity savings even in the uncompressed backup case. At the same time, the restores from the compressed backup are twice faster. So, it is purely your choice to enable or disable SQL Server backup compression based on the resources you want to optimize for.

Backup & Restore Rate (TB/hr)



Storage Consumed (TB) by Source & Backup



Host CPU consumed (%)

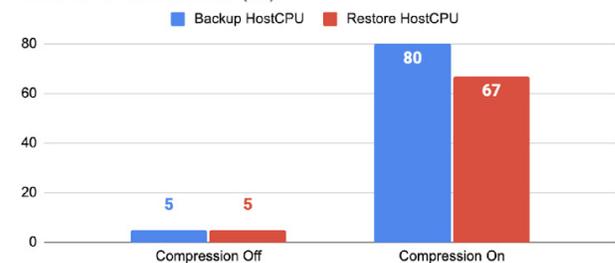


Figure 14: Impact of backup compression.

Backup Verification Performance

The SQL Server administrators routinely perform backup integrity checks, sometimes right after a full database backup is completed. This is one of the most demanding workloads for any backup appliances that optimize for storage with some form of deduplication technology. The sequential read operations from the verification process can trigger a slew of random reads on deduplicated backup appliances resulting in very poor performance. A verification test of a compressed 4TB database backup with eight backup files in our test environment completed in under nine minutes, which translates to more than 27.6TB/hr of verification rate. The limiting factor was the database host CPU, uncompressing the backup. The CPU load was 90% on the host. FlashBlade could have supported even faster verification if the database host had more CPU power.



Analysis of Single Server Results

In summary, all the single database tests demonstrate that the FlashBlade has performance capabilities to meet the demands of enterprise SQL Server database deployments. The most optimal performance can be easily obtained with minimal tuning for many SQL Server databases. Figure 15 compares the results obtained from single database tests with the best possible numbers under ideal circumstances with our system under test. For example, the ideal backup rate is determined by backing the database to a 'NUL' target. Similarly, the ideal restore rate is determined by simply running a 'VERIFYONLY' test against a backup set. These tests isolate read and write performance, whereas with real-world backup and restore operations the data needs to be read and written to and from backup source and targets. Some of the limits shown in our tests are a result of resource bottlenecks elsewhere such as database hosts or storage arrays hosting the database and not from FlashBlade. There is a lot of headroom, with more than 70% left on the FlashBlade to support additional backup workload consolidation.

Backup Rate and Restore Rate (TB/hr) - Single Server

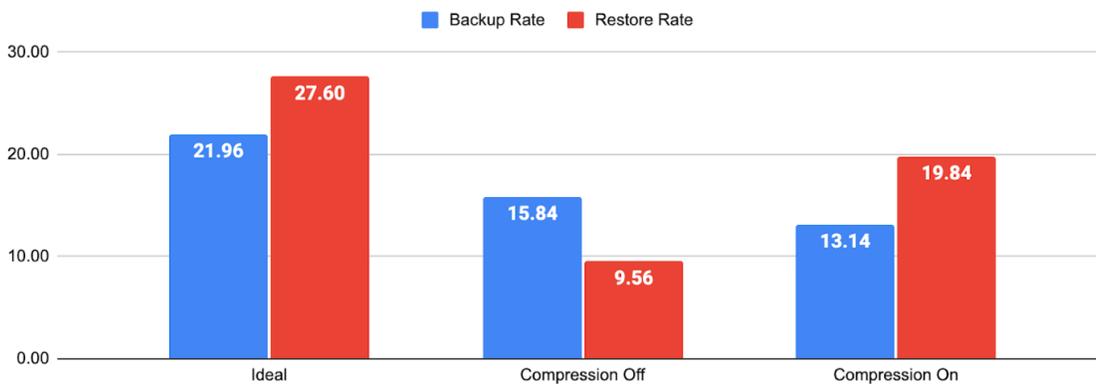


Figure 15: Summary of single server tests.

Multi-Server Tests

It is very common to find customers with 10s to 100s of SQL Server databases. One of Pure Storage’s customers has 1000s of SQL Server databases. The DBAs and Storage Administrators struggle to meet their SLAs for backup and restore/verification operations. The FlashBlade platform can scale linearly and non-disruptively from seven to 150 blades, allowing customers to start small and scale their backup environment without requiring complex configuration changes. Adding more blades provides both the storage and performance required to meet scaling needs. Multi-server testing demonstrates backing up and restoring of eight x 1TB databases deployed on eight physical hosts concurrently, applying all the tuning learned from single server tests. The source databases were split among two FlashArrays, each hosting four databases. Figure 16 summarizes the multi-server tests. Once again, the ideal rates shown are the best possible read and write performance numbers, measured separately, in our system under test. For example, the ideal backup rate was based on the maximum possible read bandwidth both FlashArrays combined provided during concurrent backups from all servers to a 'NUL' target. The restore rate was similarly dictated by the maximum sustained rate of both FlashArrays combined. We were able to achieve almost the best numbers possible with real backup and restore operations in our tests.



Backup Rate and Restore Rate (TB/hr) - Multiple Servers



Figure 16: Summary of multiple server tests.

Best Practices

Achieving the best possible performance on FlashBlade did not require any special tuning. Based on the completed testing, there is some guidance below on how to achieve optimum performance. However, it is strongly recommended that you perform calibration tests to figure out the ideal parameters for your deployment. Most SQL Server databases do not require a special layout to achieve the best possible backup and restore performance when deployed on a FlashArray. You can choose two or more LUNs to deploy their database for optimal backup and restore performance.

1. Create four or more virtual network interfaces (VIPs) on FlashBlade to achieve good backup and restore performance. The creation of multiple subnets is not required, and all the VIPs can be in the same subnet.
2. Set the following backup parameters (please determine the optimal setting for your deployment):
 - a. MAXTRANSFERSIZE = 524288
 - b. BUFFERCOUNT = 512
 - c. BLOCKSIZE – Default value provides the best possible performance in our testing.
3. Enabling backup COMPRESSION provides higher restore performance at the expense of Host CPU cycles. We recommend you make your selections based on the resource you want to optimize.
4. Host Ethernet interface bonding/teaming is recommended to simplify network configuration. It is also a good idea to isolate the backup network from the management network to keep the configuration clean.
5. Statically mounting the volume from FlashBlade over SMB protocol is not required for performing SQL Server backup and recovery. All the performance testing discussed in this whitepaper is performed using UNC paths without statically mounting SMB shares to drive letters. This dramatically simplifies the need for managing network drives.



Conclusion

Organizations today are in an era in which data drives innovation and competitive advantage and data availability are essential to business operations. For these modern enterprises, Rapid Restore with FlashBlade is a compelling alternative to a legacy data protection architecture. The native backup and recovery features built into Microsoft SQL Server, combined with the highly performant native SMB implementation of FlashBlade, provide customers with a very good choice for backup and recovery needs. This combination gives customers the best possible performance while keeping the architecture simple and straightforward. Flashblade's UFFO platform can be used not only for massive backup workload consolidation but also to consolidate many other workloads, including file serving, Analytics, and AI, onto a single scalable platform.

©2021 Pure Storage, the Pure P Logo, and the marks on the Pure Trademark List at <https://www.purestorage.com/legal/productenduserinfo.html> are trademarks of Pure Storage, Inc. Other names are trademarks of their respective owners. Use of Pure Storage Products and Programs are covered by End User Agreements, IP, and other terms, available at: <https://www.purestorage.com/legal/productenduserinfo.html> and <https://www.purestorage.com/patents>

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

purestorage.com

800.379.PURE

