IBM Spectrum Protect

# Benchmarking the IBM Spectrum Protect Cloud Cache and Object Storage Throughput

## Version 1.3

*James Damgar*

*IBM Spectrum Protect Performance Evaluation*

# CONTENTS

# Introduction

## 1.1 Cloud cache disk benchmarking

As a part of sizing and vetting a cloud cache for a cloud-container storage pool, you should run disk benchmark tests to measure and validate the capability of the disk volumes underlying the IBM Spectrum Protect™ cloud accelerator cache. In this way, you help to ensure that overlapped 128 - 256 KiB write-and-read throughput can achieve a rate high enough such that the server's bottleneck for input/output (IO) would be at the instance-to-object storage network level and not the disk level for the cloud cache. The goal is to ensure that disk storage can perform at a rate such that the IBM Spectrum Protect server could utilize it during overlapped data ingestion and cloud transfer operations and be able to stress the network link layer simultaneously. By running the tests and making any required adjustments in your storage environment, you can help to optimize the throughput of data ingestion.

To run a cloud cache disk benchmarking test, use the **tsmdiskperf.pl** Perl script, which is provided as part of the `Benchmarking.zip/tar.gz` package that is attached to this web page. To run the script, issue the following commands:

```
perl tsmdiskperf.pl workload=stgpool fslist=directory_list
```

```
perl tsmdiskperf.pl workload=db fslist=directory_list
```

where *directory_list* specifies a comma-separated list of directory paths. The preferred method is to begin testing with one directory path per storage pool directory and ensure that each storage pool directory is a separate file system. The storage pool directory represents the storage that you plan to use for your cloud cache. Then, gradually add subdirectories to the list, cycling across the storage pool directories in a round-robin distribution, as shown in the following example:

First thread-pair test:

```
/sp/stg1/dir
```

Second thread-pair test:

```
/sp/stg1/dir,/sp/stg2/dir
```

Third thread-pair test:

```
/sp/stg1/dir,/sp/stg2/dir,/sp/stg1/dir2
```

Fourth thread-pair test:

`/sp/stg1/dir,/sp/stg2/dir,/sp/stg1/dir2,/sp/stg2/dir2`

Eighth thread-pair test:

`/sp/stg1/dir,/sp/stg2/dir,/sp/stg1/dir2,/sp/stg2/dir2,/sp/stg1/dir3,/`

The goal is to continue the thread-pair tests until the throughput stabilizes.

With a `stgpool` workload specification, the script drives a 256 KiB I/O pattern, whereas with a `db` workload specification, the script drives 8 KiB operations. For each directory path that is provided as a value to the comma-separated file space list (`fslist`), two I/O processes are created to write and read to test files that are generated in that directory.

Typical script output for a `stgpool` workload run resembles the following example:

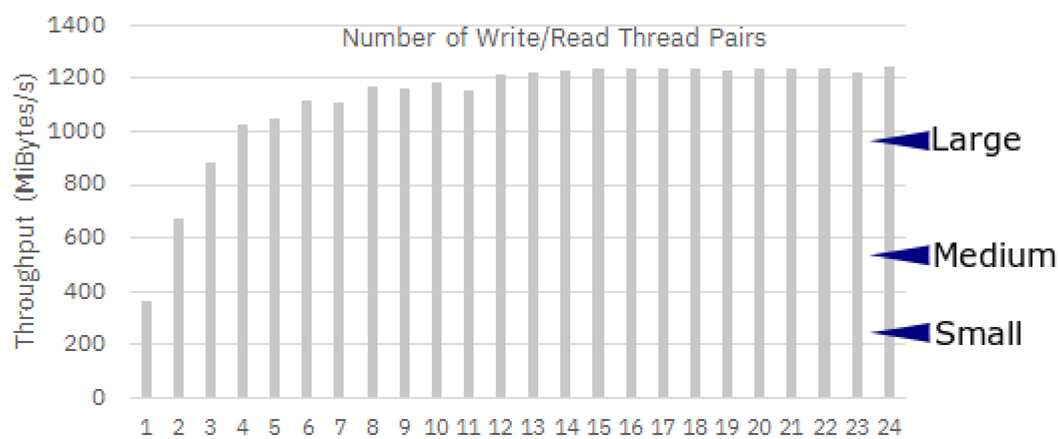
```
======================================================================
: IBM Spectrum Protect disk performance test    (Program version 3.1b)
:
: Workload type:                stgpool
: Number of filesystems:        1
: Mode:                         readwrite
: Files to write per fs:        5
: File size:                    2 GB
:
======================================================================
:
: Beginning I/O test.
: The test can take upwards of ten minutes, please be patient ...
: Starting write thread ID: 1 on filesystem /sp/sp_cc/1
: Starting read thread ID: 2 on filesystem /sp/sp_cc/1
: All threads are finished.  Stopping iostat process with id 111519
===================================================================
:  RESULTS:
:  Devices reported on from  output:
:  dm-2
:
:  Average R Throughput (KB/sec):       19473.92
:  Average W Throughput (KB/sec):       19377.45
:  Avg Combined Throughput (MB/sec):    37.94
:  Max Combined Throughput (MB/sec):    160.57
:
:  Average IOPS:                        464.63
```

```
:  Peak IOPS:                         2154.57 at 11/10/2017 04:22:32
:
:  Total elapsed time (seconds):      443
===================================================================
```

For cloud cache `stgpool` workloads, monitor the `Avg Combined Throughput (MB/sec)` value. The goal is to determine the largest aggregate average throughput for writes and reads to the cloud cache disk such that overlapped backup ingestion and data transfer to object storage will not be constrained by disk capability.

The script also should be run in `db` workload mode to determine the `Average IOPS` value to evaluate database disk capability. Here, the small random IOPS capability of the underlying disk that is used for the IBM Db2® database is of interest.

To obtain measurements of your own, conduct individual tests. You can increase the number of write/read thread pairs (and directories) until the average throughput, the average IOPS, or both stabilize. Increase the number of write/read pairs up to a maximum of 100 thread pairs (100 directories) to model a workload that is comparable to an IBM Spectrum Protect environment. Consider exporting the `Avg Combined Throughput (MC/sec)` values to a spreadsheet and then presenting the throughput and thread count values in a chart. After several tests, when the throughput stabilizes, you can determine the maximum rate. In the following example, the maximum rate of throughput is about 1200 MiB (mebibytes) per second:




The **Large**, **Medium**, and **Small** arrows indicate the minimum throughput level that the cloud cache disk should be capable of with the overlapped write/read IO workload in order to support the following levels of throughput for daily data ingestion:

| | |
|---|---|
| **Large Blueprint:** | **20 - 100 TB per day** |
| **Medium Blueprint:** | **10 - 20 TB per day** |
| **Small Blueprint:** | **Up to 10 TB per day** |

When the average throughput stabilizes, the throughput level is comparable to the throughput capability of the cloud cache. You can use disk monitoring tools such as Linux iostat or Microsoft Windows perfmon to measure and monitor disk utilization, average request wait times, and other factors during testing to determine the characteristics and limitations of the disk. If throughput does not reach required levels, consider modifying the disk configuration. To achieve higher rates of throughput, you might have to use flash or solid-state drive (SSD) disk technology.

## 1.2 Object storage benchmarking

Another important step in validating the capability of an IBM Spectrum Protect solution using a cloud-container storage pool and object storage is to benchmark the throughput of the server instance to the object storage system with a workload that is typical of IBM Spectrum Protect. Ideally, any in-the-cloud IBM Spectrum Protect solution should be network bound in terms of its connection to object storage. After inline data deduplication, compression, and encryption, the back-end ingestion rate over an HTTPS connection should dictate an upper bound for daily ingestion performance of a system.

The `Benchmarking.zip/tar.gz` package includes an executable Java JAR program (`SPObjBench.jar`) that emulates the behavior of the cloud storage application programming interfaces (APIs) used by the server. The tool can be used to drive backup and restore-type operations to and from object storage, including direct HTTP PUT, GET, and multipart file upload operations, and range-read restore behavior with a variable number of threads.

Also included in the Benchmarking package is a Perl script, `tsmobjperf.pl`, which can be used to automate execution of the `SPObjBench.jar` file with several thread counts to measure ingest (PUT) and restore (GET) scalability.

When backing up data to a cloud-container storage pool, the IBM Spectrum Protect server attempts to upload up to 100 1 GB disk container files from the accelerator cache in parallel by using a multipart upload operation. Within a production environment, this operation would occur in conjunction with overlapped ingestion to another set of container files within the same storage pool directory locations on cloud cache disk storage.

If possible, use the same Java Runtime Environment (JRE) as the IBM Spectrum Protect server in order to accurately measure indicative results. By default, the IBM Java runtime is installed within the “jre” sub-directory of the IBM Spectrum Protect installation path. For example, on AIX and Linux platforms, the default location of the Java executable is:

```
/opt/tivoli/tsm/jre/bin/java
```

A Java version of 8 or higher is required.

The `tsmobjperf.pl` wrapper script includes a command-line parameter for choosing a version of Java to use during testing.

To attempt to isolate and gauge the instance-to-object storage ingestion capability of a system, you can complete the following steps:

1. Populate a set of 10 1 GB files in a memory-mapped file system location (if possible) to use as source data for ingestion. The use of memory-mapped locations (such as `tmpfs` on Linux) is preferred to eliminate bottlenecks at the source disk. For a Linux system with at least 11 GB of free RAM, run the following commands:

   ```
   mkdir /mnt/ramdisk

   mount -t tmpfs -o size=11g tmpfs /mnt/ramdisk

   for I in `seq 10`; do dd if=/dev/urandom
   of=/mnt/ramdisk/file.$I bs=1048576 count=1024; done
   ```

2. To run a set of automated tests scaling from 1 to 100 threads, run the `tsmobjperf.pl` tool by using the recently created RAM disk files as source files to upload. If more threads are specified than files are present in the source list, the tool completes a round-robin action over these source files. Because all activity is read-only, using separate file handles from memory-mapped sources, multiple threads sharing the same file is not a concern. To test with 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 threads, run the tool as follows, specifying the arguments as needed:

   **For Amazon Simple Storage Service (S3) protocol object storage systems including IBM Cloud Object Storage, Amazon S3, and supported S3 protocol vendors:**

   ```
   perl tsmobjperf.pl type=s3 endpoints=endpoint user="user"
   pass="pass" bucket=bucket min=1 max=100 step=10 mode=both
   java=path_to_java_exe flist=comma_delimited_source_files_list
   ```

   **For Microsoft Azure Blob storage:**

   ```
   perl tsmobjperf.pl type=azure endpoints=endpoint pass="pass"
   bucket=bucket min=1 max=100 step=10 mode=both
   java=path_to_java_exe flist=comma_delimited_source_files_list
   ```

   where:

   - *type* is `s3` for IBM Cloud Object Storage, Amazon Simple Storage Service (Amazon S3), or approved S3 protocol object storage systems. Use `azure` for Microsoft Azure Blob storage.
   - *endpoint* specifies a comma-delimited list of one or more IP addresses or URLs for the object storage endpoints. For security reasons, these endpoints should be accessed over an HTTPS connection. For S3-based object storage systems, specify an endpoint URL. For Microsoft Azure based systems, specify the URL of a user's Blob storage account.
   - For S3, *user* specifies a public key ID. Enclose the parameter value in double quotation marks (").
   - For S3, *pass* specifies the secret key for a user who has valid S3 credentials to create buckets and PUT and GET objects in the region indicated by the *endpoint* URL. For Azure, the *pass* value should be a Shared Access

Signature (SAS) Token with sufficient write/read access rights to the Blob storage account over an HTTPS or HTTP connection. (An HTTPS connection is preferred for security reasons.) These values align with those that are used to define an IBM Spectrum Protect cloud-container storage pool, either via the Operations Center or the command line. Enclose the parameter value in double quotation marks (“).

- The *bucket* value should be an S3 bucket or vault name or an Azure container name for which a credentialed user has assigned PUT and GET access and that exists in the object storage system.
- The *min* and *max* values specify the minimum and maximum thread counts to test.
- The *step* value specifies the increase in thread count from test to test.
- The *java* parameter specifies the Java runtime executable to use. If possible, use the “java” provided with the IBM Spectrum Protect server installation.
- The `flist` parameter specifies a comma-delimited list of source files to be used for multipart upload. These files should be the same as those created earlier in the memory-mapped file system.

The following example is for data transfer to an IBM Cloud Object Storage Regional (South) endpoint, using up to 100 upload threads with an existing test bucket. Restore tests are conducted afterwards:

```
perl tsmobjperf.pl type=s3 endpoints=https:// s3.us-
south.objectstorage.service.networklayer.com user=”PUBLICKEYID”
pass=”SECRETKEY” bucket=testbucket min=1 max=100 step=10
mode=both java=/opt/tivoli/tsm/jre/bin/java
flist=/mnt/ramdisk/file.1,/mnt/ramdisk/file.2,/mnt/ramdisk/file.
3,/mnt/ramdisk/file.4,/mnt/ramdisk/file.5,/mnt/ramdisk/file.6,/m
nt/ramdisk/file.7,/mnt/ramdisk/file.8,/mnt/ramdisk/file.9
,/mnt/ramdisk/file.10
```

The following example is for data transfer to a Microsoft Azure Blob object storage endpoint in the West US 2 (Oregon) Region, using up to 100 upload threads with an existing test container. Restore tests are conducted afterwards:

```
perl tsmobjperf.pl type=azure endpoints=
https://spobjpvthot.blob.core.windows.net/
user=”STORAGEACCOUNTNAME” pass=”SASTOKENSTRING”
bucket=testcontainer min=1 max=100 step=10 mode=both
/opt/tivoli/tsm/jre/bin/java
flist=/mnt/ramdisk/file.1,/mnt/ramdisk/file.2,/mnt/ramdisk/file.
3,/mnt/ramdisk/file.4,/mnt/ramdisk/file.5,/mnt/ramdisk/file.6,/m
nt/ramdisk/file.7,/mnt/ramdisk/file.8,/mnt/ramdisk/file.9
,/mnt/ramdisk/file.10
```

Each thread count test (for 1, 10, 20, or more threads) will upload 10 1 GB objects per thread. The previous examples would result in a total of 5510 GB of data being stored to the test bucket or container after all thread tests are completed. The tool does not remove objects that are created. You must remove the objects manually after test completion.

Upon test completion, the tool generates aggregate throughput metrics that can be used to estimate instance-to-object storage performance rates in an IBM Spectrum Protect production environment. Data is provided in comma-separated-value (CSV) format and the output of the `SPObjBench.jar` tool can be inspected upon completion as well. For example:

```
===================================================================
: IBM Spectrum Protect object storage test
:
: Test Mode:       both
: Type:            s3
: Total Endpoints: 1
: Min Threads:     1
: Max Threads:     100
: Thread Step:     10
: File List:
/mnt/ramdisk/file.1,/mnt/ramdisk/file.2,/mnt/ramdisk/file.3,/mnt/ramdisk/file.4,
/mnt/ramdisk/file.5,/mnt/ramdisk/file.6,/mnt/ramdisk/file.7,/mnt/ramdisk/file.8,
/mnt/ramdisk/file.9 ,/mnt/ramdisk/file.10
: Using java:      /opt/tivoli/tsm/jre/bin/java
===================================================================


SPObjBench.jar output being captured to file:   tsmobjperf.1540336631.out


===================================================================
: Test Results
Thread Count, Write Throughput (MB/s), Read Throughput (MB/s)
1, XXX, YYY
10, XXX, YYY
20, XXX, YYY
30, XXX, YYY
40, XXX, YYY
50, XXX, YYY
60, XXX, YYY
70, XXX, YYY
80, XXX, YYY
```

```
90, XXX, YYY

100, XXX, YYY

====================================================================
```

You might find it useful to monitor network transmission rates externally from the tool, as well, to validate the absolute throughput rate that is experienced to object storage over the network (which should be an Ethernet network). The tool reports an aggregate rate that can include build-up and tear-down overhead associated with the tool. Calculating an actual transmission rate from the instance-to-object storage while the test is running can give an indication of the throughput limits of the environment. On Linux, for example, the `dstat` utility can be used to monitor several system metrics at once, including network interface send and receive statistics, by using the basic command:

```
% dstat
```

The output is similar to the following example:

```
You did not select any stats, using -cdngy by default.
----total-cpu-usage---- -dsk/total- -net/total- ---paging-- ---system--
usr sys idl wai hiq siq| read  writ| recv  send|  in   out | int   csw
  0   0 100   0   0   0|  60B 2511B|   0     0 |   0     0 |  76    71
 15   1  84   0   0   1|   0    24k|1674k  58M|   0     0 |  42k 2785
 15   1  83   0   0   1|   0     0 |1838k  62M|   0     0 |  46k 2969
 16   1  82   0   0   1|   0     0 |1832k  61M|   0     0 |  45k 3127
 15   1  84   0   0   1|   0     0 |1753k  61M|   0     0 |  44k 2822
 16   1  83   0   0   1|   0     0 |1811k  62M|   0     0 |  45k 3001
 15   1  83   0   0   1|   0     0 |1778k  62M|   0     0 |  45k 3068
 16   1  82   0   0   1|   0     0 |1870k  63M|   0     0 |  46k 3068
 16   1  82   0   0   1|   0     0 |1933k  64M|   0     0 |  46k 3011
 15   1  83   0   0   1|   0     0 |1834k  63M|   0     0 |  46k 2974
```

The `dstat` tool generates a new line of metrics at a configured interval, much like the standard `iostat` and `netstat` utilities. In the previous output, the `net/total send` column is of greatest interest, here reported in mebibytes, as an indication of how quickly data could be sent to the object storage endpoint from the server.

The following average write throughput thresholds should be met or exceeded, in general, for the object storage system and network to support the following Blueprint levels of ingestion throughput:

Small Blueprint (up to 10 TB per day): Approximately 100-200 MiB/second, or more

Medium Blueprint (10 - 20 TB per day): Approximately 400-600 MiB/second, or more

Large Blueprint (20 - 100 TB per day): Approximately 800-1000 MiB/second, or more

# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive, MD-NC119*
*Armonk, NY 10504-1785*
*US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive, MD-NC119*
*Armonk, NY 10504-1785*
*US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.



## Trademarks

