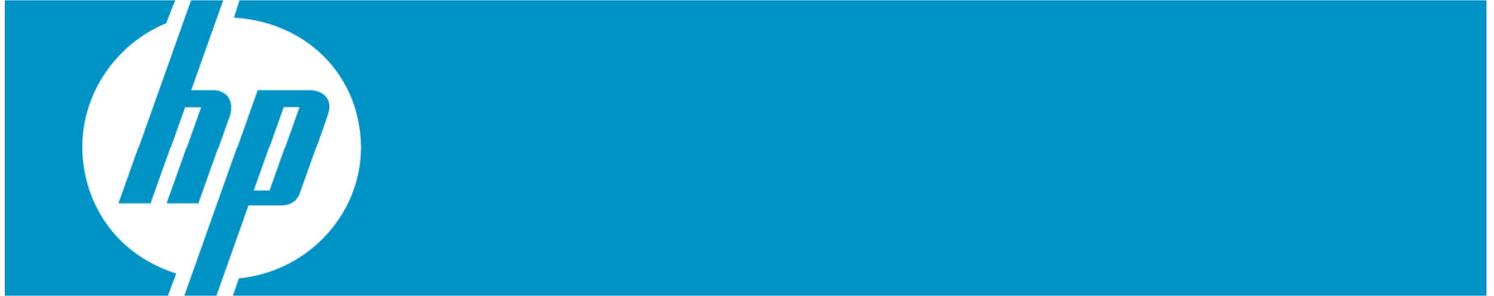


P Muralidhar Kini



Using XFC to get System/Volume/File/IO Statistics	1
XFC introduction	2
Statistics provided by XFC.....	2
HotFile statistics	2
Hotfile – TOPQIO.....	2
Hotfile – TOPHITRATE	3
Memory statistics.....	4
System statistics	5
Periodic system activity statistics.....	6
Systemwide IO size distribution statistic	6
Volume statistics.....	8
File statistics	9
IO statistics.....	11
Summary.....	12

XFC introduction

The OpenVMS operating system has two types of caches – the data cache and the metadata cache. The data cache is referred to as eXtended File Cache (XFC) and the metadata cache is called the eXtended Qio Processor (XQP). XFC caches contents of all files present on the disk, excluding directory files. XQP caches file headers and directory files.

Statistics provided by XFC

XFC provides various statistics of the system, which can be used for the overall system performance tuning. XFC provides the statistics at the system, volume, file and I/O level. The DCL commands or the XFC SDA extension can be used to get the XFC statistics. In most scenarios, DCL as well as XFC SDA extensions provide a similar type of statistics. The XFC SDA extension generally provides little more information than the corresponding DCL commands.

Some of the important statistics include Read hit rate, XFC cache memory, I/O size distribution, and so on. This paper explores about the various statistics that are provided by XFC.

HotFile statistics

Hotfile is a file in the system that records many activities in terms of read and write operations. In many environments it is observed that 80% of the I/O's that are issued are on 20% of the files present in the system. This being the case, it is very important to know which files that are present in the system fall into this 20% category.

Hotfile - TOPQIO

On a given volume, the TOPQIO keyword instructs XFC to display the first 'N' entries in descending order of the number of IOs done to the file.

DCL command

```
$ SHOW MEM/CACHE=(VOLUME=<VOL-NAME>,OPEN,CLOSED, TOPQIO:N)
<VOL-NAME>: Volume name
N: Number of entries to be displayed
```

Sample output

```
$$SHOW MEMORY/CACHE=(VOLUME=BRAMHA$DKB200:,OPEN,CLOSED, TOPQIO:2)
System Memory Resources on 30-MAY-2010 03:13:44.48
```

Extended File Cache Top QIO File Statistics:

_BRAMHA\$DKB200: (DISK\$BR_V83R), Caching mode is VIOC Compatible

_BRAMHA\$DKB200:[MURALI]PERFORM_TEST.COM;68 (389451, 1057, 0) (closed)

Caching is enabled, active caching mode is Write Through

Allocated pages	1	Total QIOs	1177
Read hits	1176	Virtual reads	1176
Virtual writes	1	Hit rate	100%
Read aheads	0	Read throughs	1176
Write throughs	1	Read arounds	0
		Write arounds	0

_BRAMHA\$DKB200:[MURALI]TEST_CONFIG.COM;47 (389469, 13875, 0) (closed)

Caching is enabled, active caching mode is Write Through

Allocated pages	1	Total QIOs	393
Read hits	392	Virtual reads	392

Virtual writes	1	Hit rate	100%
Read aheads	0	Read throughs	392
Write throughs	1	Read arounds	0
		Write arounds	0

Total of 2 files for this volume

\$

Interpreting the output

On the disk BRAMHA\$DKB200, the maximum number of IOs are done to the file PERFORM_TEST.COM (389451, 1057, 0).

- The total number of IOs done to the file is 1177. Among these IOs, 1176 are read IOs and 1 write IO.
- The read hits of the file are 100%. This is excellent because all attempts to read the file were satisfied from the cache itself.
- Allocated pages of the file is 1 page. This is a small temporary file and hence consumes a very small portion of XFC cache memory.

Hotfile - TOPHITRATE

On a given volume, the TOPHITRATE keyword instructs XFC to display the first 'N' entries in descending order of read hit rates of the file.

DCL command

```
$ SHOW MEMORY/CACHE=(VOLUME=<VOL-NAME>,OPEN,CLOSED,TOPHITRATE:N)
<VOL-NAME>: Volume name
N: Number of entries to be displayed
```

Sample output

```
$SHOW MEMORY/CACHE=(VOLUME=BRAMHA$DKB200:,OPEN,CLOSED,TOPHITRATE:2)
System Memory Resources on 30-MAY-2010 03:27:08.67
```

Extended File Cache Top Hitrate File Statistics:

```
_BRAMHA$DKB200: (DISK$BR_V83R), Caching mode is VIOC Compatible

_BRAMHA$DKB200:[MURALI]TRACE_SETUP.COM;5 (389474, 1, 0) (closed)
Caching is enabled, active caching mode is Write Through
Allocated pages          1      Total QIOs          5
Read hits                 4      Virtual reads     4
Virtual writes           1      Hit rate          100%
Read aheads              0      Read throughs    4
Write throughs           1      Read arounds     0
                               Write arounds    0
```

```
_BRAMHA$DKB200:[MURALI]PERFORM_TEST.COM;68 (389451, 1057, 0) (closed)
Caching is enabled, active caching mode is Write Through
Allocated pages          1      Total QIOs       1177
Read hits                1176    Virtual reads    1176
Virtual writes           1      Hit rate         100%
Read aheads              0      Read throughs   1176
Write throughs           1      Read arounds    0
                               Write arounds    0
```

Total of 2 files for this volume

\$

Interpreting the output

On the disk BRAMHA\$DKB200, the file with maximum read hit rate is TRACE_SETUP.COM (389474, 1, 0).

- The total number of IOs done to the file is 5. Among these IOs, 4 are read IOs and 1 write IO.
- The read hits of the file are 100%. This is excellent because all attempts to read the file were satisfied from the cache itself.
- Allocated page of the file is one page. This is a small temporary file and hence consumes a very small portion of the XFC cache memory.

Memory statistics

The "SHOW MEM/CACHE" DCL command provides the XFC memory utilization statistics at the system level. It also provides other important statistics, such as read hit rate, and number of read/write IOs at the system level.

DCL command

```
$SHOW MEMORY/CACHE
```

Sample output

```
$SHOW MEMORY/CACHE
System Memory Resources on 30-MAY-2010 04:37:45.74

Extended File Cache (Time of last reset: 29-MAY-2010 00:25:38.06)
  Allocated (MBytes)  80.90   Maximum size (MBytes)  2047.81
  Free (MBytes)      0.00   Minimum size (MBytes)  3.12
  In use (MBytes)    80.90   Percentage Read I/Os   77%
  Read hit rate      90%    Write hit rate          0%
  Read I/O count     70914  Write I/O count         20452
  Read hit count     63875  Write hit count         0
  Reads bypassing cache 204    Writes bypassing cache 24
  Files cached op    398    Files cached closed    312
  Vols in Full XFC mode 0      Vols in VIOC Compatible mode 1
  Vols in No Caching mode 0    Vols in Perm. No Caching mode 0
$
```

Interpreting the output

- XFC memory can grow to a maximum of 2GB, that is 2047.81 MB
- Currently XFC has allocated itself a memory of 80.9 MB
- The free memory with XFC is 0MB. Any subsequent request to cache a file would cause XFC to expand in size.
- The overall read hit rate of the system is 90%. This is a good hit rate.
- Out of the total IOs issued in the system, 77% of the IOs are read IOs.

SDA command

```
SDA> SHOW MEM/CACHE
```

Sample output

```
SDA> SHOW MEM/CACHE
System Memory Resources from Running System on 30-MAY-2010 04:44:48.61
-----
Extended File Cache (Time of last reset: 29-MAY-2010 00:25:38.06)
  Allocated (MBytes)  80.90   Maximum size (MBytes)  2047.81
  Free (MBytes)      0.00   Minimum size (MBytes)  3.12
  In use (MBytes)    80.90   Percentage Read I/Os   77%
  Read hit rate      90%    Write hit rate          0%
```

Read I/O count	71049	Write I/O count	20453
Read hit count	64006	Write hit count	0
Reads bypassing cache	204	Writes bypassing cache	24
Files cached open	399	Files cached closed	311
Vols in Full XFC mode	0	Vols in VIOC Compatible mode	1
Vols in No Caching Mode	0	Vols in Perm. No Caching mode	0

SDA>

Interpreting the output

The output is interpreted similar to that of the corresponding "SHOW MEM/CACHE" DCL command.

System statistics

The "SHOW MEM/CACHE" DCL command or the "XFC SHOW SUMMARY" SDA command provides the XFC statistic at the system level. Some of the important statistics that can be obtained are read hit rate and the number of read/write IOs.

DCL command

\$ SHOW MEM/CACHE

Sample output

```
$SHOW MEMORY/CACHE
      System Memory Resources on 30-MAY-2010 04:37:45.74

Extended File Cache (Time of last reset: 29-MAY-2010 00:25:38.06)
  Allocated (MBytes)      80.90   Maximum size (MBytes)      2047.81
  Free (MBytes)          0.00   Minimum size (MBytes)      3.12
  In use (MBytes)        80.90   Percentage Read I/Os       77%
  Read hit rate           90%    Write hit rate              0%
  Read I/O count          70914  Write I/O count             20452
  Read hit co             63875  Write hit count             0
  Reads bypassing cache   204   Writes bypassing cache     24
  Files cached open       398   Files cached closed        312
  Vols in Full XFC mode   0     Vols in VIOC Compatible mode 1
  Vols in No Caching mode 0     Vols in Perm. No Caching mode 0
$
```

Interpreting the output

In the system,

- The overall read hit rate of the system is 90%. This is a very good hit rate.
- Out of the total IOs issued in the system 77% of the IOs are read IOs.
- In this system, the number of open files cached by XFC is 398 and the number of closed files cached by XFC is 312.

SDA command

SDA> XFC SHOW SUMMARY

Sample output

```
SDA> XFC SHOW SUMMARY
XFC Summary (current time 30-MAY-2010 04:59:24.36)
-----
...
Virtual Reads:           71090
Read Hits:               64019
Read Cache Hit Percentage: 90.05 %
Virtual Writes:          20770
Open Files:              400
Closed Files in the Cache: 311
Total Files in Cache:    711
```

...
SDA>

Interpreting the system output

- 71090 virtual read IOs and 20770 virtual write IOs are issued
- Read hit rate is 90.05%. This is a very good hit rate.
- The total number of files in the cache is 711. Out of these, 400 files are open files and 311 files are closed files.

Periodic system activity statistics

The "SHOW MEM/CACHE" DCL command provides the XFC memory utilization statistics at the system level. It also provides other important statistics such as read hit rate and the number of read/write IOs at the system level. XFC records the activity of the system on a periodic basis, which is at a regular interval every 10 minutes. XFC records various information about the system, which would be useful to track the behavior of the system. Some of the statistics captured are:

- Number of read/write IOs
- Read hit rate
- Open/closed files in cache
- XFC cache memory

SDA command

SDA>XFC SHOW HISTORY

Sample output

SDA>XFC SHOW HISTORY

XFC Performance History (current time 30-MAY-2010 05:20:43.64)

```
-----  
      Time          Peak.....Cache   I/O   Counts.....      Allocated  
                IO      Reads Hits   Rate  Writes      Memory  
...  
29-MAY 01:05      1         10     7   70%     316 ...  73.5 MB  
29-MAY 00:55      3    49364 46923 95%    2231 ...  71.0 MB  
29-MAY 00:45      1         315    265 84%      25 ...  66.9 MB  
29-MAY 00:35      8    17234 13279 77%     167 ...  65.9 MB  
...  
SDA>
```

Interpreting the output

Between the time intervals of 29-MAY 00:35 and 29-MAY 00:45, the following data was gathered:

- Number of read IOs were 315, Number of Write IOs were 25
- Read hit rate was 84%
- The XFC cache memory increased from 65.9 MB to 66.9MB

Systemwide IO size distribution statistic

XFC also keeps track of all the IOs issued in the system based on the IO size i.e., 1 block or 2 blocks and so on.

For IOs of particular size, XFC keeps track of statistics such as:

- Number of read operations
- Number of write operations
- Number of read hits

This statistic provides information on the read hit of IOs based upon the IO size.

DCL command

```
$ SHOW MEM/CACHE/FULL
```

Sample output

```
$SHOW MEMORY/CACHE/FULL
      System Memory Resources on 31-MAY-2010 02:36:47.05

Extended File Cache (Time of last reset: 29-MAY-2010 00:25:38.06)
...
I/O Statistics - Distributions (MAX_IO_SIZE: 127)
-----
Transfer Size:      Reads  Read Hits  Writes
  1 Block IO:      28160  27428    2384
  2 Block IO:       6948   6345     24
  3 Block IO:       402    307      34
...
126 Block IO           2      0        0
>127<256 Block IO:    27            3
>255      Block IO:   53            0
-----
Totals:                73403  65848    34574
$
```

Interpreting the system output

- 238 4 write IOs were of size 1 block. 28160 IOs were of size 1 block and among these 27428 IOs were a read hit.
- 27 read IOs and 3 write IOs that were issued were of size in the range of 127-256 blocks. None of them was a read hit.
- 53 read IOs and no write IOs issued were of size greater than 255 blocks. None of them was a read hit.

SDA command

```
SDA> XFC SHOW SUMMARY/STATS
```

Sample output

```
SDA> XFC SHOW SUMMARY/STATS
XFC Summary (current time 31-MAY-2010 02:51:16.69)
-----
...
I/O Statistics - Distributions (MAX_IO_SIZE: 127)
-----
Transfer Size      Reads      Hit Rate  Writes ...
  (blocks)
  1                28196      97%      2384 ...
  2                6962       91%       24 ...
  3                403        76%       34 ...
...
Reads between 128 and 255 blocks since last reset:    27
Read hits between 128 and 255 blocks since last reset: 0 (0%)
Reads larger than 256 blocks since last reset:        53
Read hits larger than 256 blocks since last reset:    0 (0%)
Writes between 128 and 255 blocks since last reset:    3
Writes larger than 256 blocks since last reset:        0
...
SDA>
```

Interpreting the system output

- 28196 read IOs and 2384 write IOs were of size 1 block. The read hit rate for 1 block IOs is 97%.
- 27 read IOs and 3 write IOs issued were of size in the range of 127-256 block. None of them was a read hit.
- 53 read IOs and no write IOs that were issued were of size greater than 255 blocks. None of them was a read hit.

Volume statistics

The "SHOW MEM/CACHE=VOL=<VOL-NAME>" DCL command or the "SDA>XFC SHOW VOLUME/NAME=<VOL-NAME>" command provides the XFC statistics at the volume level. Some of the important statistics are read hit rate and the number of read/write IOs.

DCL command

```
$ SHOW MEMORY/CACHE=VOLUME=<VOL-NAME>
<VOL-NAME>: Name of the volume
```

Sample output

```
$SHOW MEMORY/CACHE=VOLUME=BRAMHA$DKA100:
System Memory Resources on 30-MAY-2010 05:38:56.87
```

Extended File Cache Volume Statistics:

```
_BRAMHA$DKA100: (DISK$BR_V84R), Caching mode is VIOC Compatible
Open files          398      Closed files        314
Files ever opened  1519      Files ever deposited 807
Allocated pages    9715      Locks acquired      0
Total QIOs         85299     Read hits           61370
Virtual reads      66304     Virtual writes      18995
Read hit rate      92 %      Read aheads         344
Read throughs     66132     Write throughs     18992
Read arounds      172      Write arounds       3
Ave Disk I/O Resp Time incl cache hits (microseconds) 1126
```

```
Total of 1 volume in cache
$
```

Interpreting the output

On the volume BRAMHA\$DKA100:

- The number of open files in cache is 398 and the number of closed files in the cache is 314.
- This volume has consumed 9715 pages of the total XFC cache memory
- The read hit rate of the volume is 92%. This is a very good read hit rate.
- Total IOs done on this volume is 85299. Out of this amount, 66304 IOs are virtual reads and 18995 IOs are virtual writes.

SDA command

```
SDA> XFC SHOW VOLUME/NAME=<VOL-NAME>
<VOL-NAME> = DISK$<LBL>
<LBL> is the label of the volume.
```

Sample output

```
SDA> XFC SHOW VOLUME/NAME=DISK$BR_V84R
```

Full List of XFC Cached Volumes (CVBs)

```
-----
Cache Volume Block (CVB)
```

```

-----
Statistics Valid From: 29-MAY-2010 00:25:40.06
...
Cached Open Files: 400
Cached Closed Files: 312
...
Name: DISK$BR_V84R
QIO count: 85447
Virtual Read Count: 66452
Virtual Write Count: 18995
Read Percentage: 77 %
Hit Rate: 71 %
...
I/O Response Times (This Volume)
  Overall Average: 1.1242 milliseconds
  Cache Hit: 0.0070 milliseconds
  Disk: 3.9962 milliseconds
  Accuracy: 99 %
...
SDA>

```

Interpreting the output

On the DISK\$BR_V84R volume:

- The number of open files in cache is 400 and the number of closed files in the cache is 312.
- The hit rate of the volume is 71%.
Note: This is its hit rate and NOT the read hit rate, where
Hit rate = read hits / (virtual reads + virtual writes)
- Total IOs done on this volume is 85447. Out of this amount, 66452 IOs are virtual reads and 18995 IOs are virtual writes.
- XFC computes the IO response time for various IOs that are issued to this volume. I.e., on any given volume, for each IOs, XFC keeps track of the following:
 - o IOs that start and end on the same CPU and are cache hits
 - o IOs that start and end on the same CPU and are cache misses

File statistics

The "SHOW MEM/CACHE=FILE=<FILE-NAME>" DCL command or the "SDA>XFC SHOW FILE/ID=<FILE-ID-HEX>" command provides the XFC statistics at the file level. Some of the important statistics are read hit rate, number of read/write IOs, and XFC cache memory consumed by the file.

DCL command

```

$SHOW MEMORY/CACHE=FILE=<FILE-NAME>
<FILE-NAME>: Full path of the file

```

Sample output

```

$SHOW MEMORY/CACHE=FILE=BRAMHA$DKB200:[MURALI]PERFORM_TEST.COM
System Memory Resources on 30-MAY-2010 06:01:04.46
Extended File Cache File Statistics:

```

```

_BRAMHA$DKB200:[MURALI]PERFORM_TEST.COM;68 (389451, 1057, 0) (closed)
Caching is enabled, active caching mode is Write Through
  Allocated pages      1      Total QIOs          1177
  Read hits            1176    Virtual reads       1176
  Virtual writes       1      Hit rate            100 %
  Read aheads          0      Read throughs      1176
  Write throughs       1      Read arounds        0
                                   Write arounds        0
$

```

Interpreting the output

For the file PERFORM_TEST.COM (389451, 1057, 0):

- This file has consumed one page of the total XFC cache memory.
- The read hit rate of the file is 100%. This means that all attempts to access the file were satisfied from the cache itself.
- Total IOs done on this file is 1177. Out of these, 1176 IOs are virtual reads and 1 IO is virtual write.

SDA command

```
SDA> XFC SHOW FILE/ID=<FID-IN-HEX>/STATS
```

<FID-IN-HEX> = File number component of the file's FID in Hex

Sample output

```
SDA> XFC SHOW FILE/ID=2368/STATS
```

Full XFC Cache File Block (CFB) Details

```
-----  
File: DISK$TEST:[SYS0.SYSERR]ERRLOG.SYS;1
```

```
...
```

```
External FID:          (9064,1,0)
```

```
...
```

```
File IO Statistics (current time 29-MAY-2010 09:19:13.85)
```

```
-----  
Statistics Valid From: 24-MAY-2010 03:13:24.80
```

```
Total QIOs to this file:      2297  
Read IOs to this file:       1530  
Read Hits:                    1220  
Hit Rate:                     53.11 %  
Write IOs to this file:      767  
Truncates:                    1  
Accesses:                     765  
Deaccesses:                   765
```

```
I/O Response Times (This File)
```

```
Overall Average:              1.9058 milliseconds  
Cache Hit:                    0.0266 milliseconds  
Disk:                         4.0346 milliseconds  
Accuracy:                     100 %
```

```
...
```

```
SDA>
```

Interpreting the output

For the file ERRLOG.SYS (9064, 1, 0):

- Total IOs done on this file is 2297. Out of these, 1530 IOs are virtual reads and 767 IOs are virtual writes.
- The hit rate of the file is 53.11%.
(Note: This is the hit rate and NOT the read hit rate).
Hit rate = read hits / (virtual reads + virtual writes)
- There has been 1 truncate, 765 access and 765 deaccess operations on the file
- XFC computes the IO response time for various IOs that are issued to this file
I.e. on the file, for each IOs, XFC keeps track of:
 - o IOs that are cache hits which start and end on the same CPU
 - o IOs that are cache miss which start and end on the same CPU

IO statistics

XFC creates a context block for every IO that it handles. The context block stores various information about the IO request such as IRP, operation start time, current IO phase and so on. At any given point of time, the context block indicates the current state of the IO.

IO statistics are generally used for debugging purpose. While analyzing the crash dump, you may be interested to know the current state of an IO request. In such cases, if the IO request has gone through XFC, then XFC would have created a context block for that IO request. The context block can then be examined to know the current state of the IO request.

If XFC is handling multiple IO requests, then it would have created multiple context blocks, one for each IO request. In such cases, you must examine the context block that corresponds to the IO request that you are looking for. This is done by examining the list of all the context blocks and then selecting a particular context block based on whether its IRP field corresponds to the IRP we are looking for.

SDA command

```
SDA> XFC SHOW CONTEXT
```

Sample output

```
SDA> XFC SHOW CONTEXT
```

```
List of All XFC Active Contexts (CTX) (current time 31-MAY-2010  
03:04:47.33)
```

```
-----  
Context (CTX) Address: FFFFFFFF89379910  
I/O Phase:             eiopDiskIO  
I/O Type:              eiotWriteThrough  
Operation started:     31-MAY-2010 03:04:47.32  
Stall Reason:          estrDiskIO  
...  
Stall Op (IRP):        FFFFFFFF8916DBC0  
...  
  Cache Hit:           False  
...  
Volume (CVB):          0000000000000000  
Volume Id:             FFFFFFFFEE0189120  
File Id:               0000000000000398F  
...  
Cache File Block:      FFFFFFFF7D715940  
Process (PCB):         FFFFFFFF896B0CC0  
...  
SDA>
```

Interpreting the output

For the context block FFFFFFFF89379910:

- XFC received the IO request on "31-MAY-2010 03:04:47.32".
- The IO request is for a write operation.
- The file id of the file to which this IO is done is "398F"
- This IO request is not a cache hit and hence the data has to be fetched from the disk
- Other structures associated with this IO are:
 - IRP - FFFFFFFF8916DBC0 -> IO request packet
 - PCB - FFFFFFFF896B0CC0 -> process control block

Summary

The techniques described here allow a user to get the XFC statistics at the system/volume/file/IO level. These statistics should then be analyzed to determine the hot files in the system at the volume level and also if the overall system performance is acceptable or not.

One of the important parameters that impacts the system performance is the read hit rate of the system /volume/file, which depends on a number of factors such as:

- Number of local read/write IOs to the file
- Number of clusterwide read/write IOs to the file

Once we collect the XFC statistics, for example,

- Read hit rate of the system is 50%

This means the overall cache read hit rates on the system is 50%. This is very low and hence there is a need to tune the system to increase the overall cache read hit rates. This would involve analyzing the IOs issued on the system, i.e. on different volumes by various applications.

- Read hit rate of the volume V1 is 50%

This means that the cache hit rate on the volume V1 is 50%. This is very low and hence there is a need to tune the system with respect to the IO operations on the volume to increase the read hit rate of the volume. This would involve analyzing the IOs issued on the volume V1 by various applications.