



Copyright © 2004-2010 SuperSpeed LLC All Rights Reserved.

07 October 2010

**QUICK REFERENCE FOR
SUPERCACHE AND SUPERVOLUME
SERVER EDITION
v.4.0.273**

Contents:

- 1.0 Features
- 2.0 Functionality
- 3.0 Device Management
- 4.0 Cache Tuning

1.0 Features

1.1 SuperCache (v4.0) features:

- Supports volumes sizes from 257 MB to 256 TB
- Supports cache sizes from 256 MB to 2 TB (63 GB on 32-bit platforms)
- Unmanaged Memory feature supports up to 63 GB on 32-bit platforms
- Supports Basic or Dynamic disks
- Supports caching the operating system's volume (boot partition)
- Supports caching volumes containing page files
- Supports volumes without drive letters
- Supports volumes with proprietary file systems
- Dynamic configuration, including start and stop caching (no reboot)
- Cache page size (granularity) from 4 KB to 256 KB
- Fail-safe cache allocation
- Write-through or write-deferred modes
- Average dirty page latency setting from 1 to 14400 seconds (4 hours)
- Suspended lazy-write option
- Improved support for NUMA on qualified systems

1.2 SuperVolume (v4.0) features:

- Supports volumes sizes from 16 MB to 2 TB (63 GB on 32-bit platforms)
- Unmanaged Memory feature supports up to 63 GB on 32-bit platforms
- Supports Basic or Dynamic disks
- Supports caching the operating system's volume (boot partition)
- Supports caching volumes containing page files
- Supports volumes without drive letters
- Supports volumes with proprietary file systems
- Dynamic configuration, including start and stop caching (no reboot)
- Write-through or write-deferred modes
- Average dirty page latency setting from 1 to 14400 seconds (4 hours)
- Suspended lazy-write option
- Improved support for NUMA on qualified systems

2.0 Functionality

The core functionalities of SuperCache and of SuperVolume are implemented in the driver named "SscVF.sys", which is stored in the "%WinDir%\System32\Drivers" folder. SuperCache and SuperVolume are actually kinds of volume filters. The driver can activate either the SuperCache filter or the SuperVolume filter, but not both concurrently. The filters are activated through a single volume filter device. The ability to activate a filter on a given volume depends on licensing, available memory, and qualifying factors of the volume itself.

SuperCache and SuperVolume are block-level caches, residing below Windows' files-system cache. Both are associated with a volume, while Windows' file-system cache is associated with a file-system. Both cache volumes whether these contain file-systems or not (RAW), and whether they are assigned a drive letter or not. SuperCache is appropriate when the size of the volume requiring acceleration is larger than available physical memory. SuperVolume is appropriate when the volume's size is less than available physical memory.

Almost all the memory resources allocated to SuperCache and SuperVolume are taken from physical memory. This effectively eliminates any burden on the operating system's limited paged and non-paged pools. Nonetheless, care should be exercised to avoid starving the operating system, services and applications of physical memory. Under severe conditions, memory resource starvation may lead to a system crash. Additionally, although physical memory may be available for SuperCache and SuperVolume, over-allocation of the physical memory pool to SuperCache or SuperVolume will adversely affect the performance of other system components reducing overall system performance despite the larger cache sizes.

The x86 versions of Windows XP, Windows Vista and Windows 7 are limited to a 32bit physical memory address space. Because hardware devices require physical memory address space, the actual amount of physical memory address space available to Windows is typically 3.25 GB or less. This means that any RAM above that limit is effectively unusable and wasted. SuperCache and SuperVolume can reclaim that memory by a feature we call Unmanaged Memory. The Unmanaged Memory feature allows SuperCache and SuperVolume to access physical memory that the operating system does not.

For example, on a machine with 6 GB of main memory, Windows 7 x86 will use only about 3.2 GB, leaving about 2.8 GB completely unused. 32bit Windows simply doesn't attempt to manage or use the memory above ~3.2 GB. With their Unmanaged Memory feature, SuperCache and SuperVolume can put the unused 2.8 GB to work to enhance system performance.

The x64 version of Windows XP, and x86 and 64 versions of Windows Vista and Windows 7 provide special support for machines with a nonuniform memory architecture (NUMA). When active on such platforms, the memory manager for SuperCache and SuperVolume examines the processor and memory affinities reported by Windows, and then optimize physical memory allocations across the described NUMA nodes. The NUMA support in SuperCache and SuperVolume greatly improves the speed of memory allocations and equalizes the distribution of cache allocations among the NUMA nodes. Currently, NUMA support does not include the ability to build caches on specific nodes.

Please visit our Web site for more details. www.superspeed.com

3.0 Device Management

Each volume filter device is managed by a command-line utility, Vfcfg.exe, or a Windows shell property page for qualified mounted devices.

3.1 Mounted Device "Performance" Property Page

Mounted devices, including disk partitions or volumes with or without drive letters, appear in the results pane of the MMC Disk Management snap-in. Right-clicking on one of these devices and selecting the Properties item on the resultant shortcut menu brings up that device's property sheet.

When properly installed, and on qualified volumes, SuperCache and SuperVolume add a new property page to this sheet: Performance. Click on the performance tab to access SuperCache, SuperVolume, Memory and Licensing information and configuration. This page is not displayed on devices with removable media, or if the product is improperly installed. The buttons of those products licensed for use will be enabled.

3.2 VFCFG.EXE Command-Line Utility

Each qualified volume's filter device may be controlled via a command-line utility called "Vfcfg.exe". This file is stored in the "%WinDir\System32" folder. This command-line interface (CLI) implements WMI to permit local and remote configuration of the volume filter device.

Its usage is displayed by simply typing its name in a command prompt window. The following text (or a newer version of it) will be displayed:

```
Vfcfg.exe v.4.0 Copyright (c) 2004-2010 SuperSpeed LLC.
```

```
Volume filter device configuration tool. Configures the volume filter device active on a local or remote storage volume (eg. disk partition) to accelerate disk performance. Manages both SuperCache (SC) and SuperVolume (SV) filters.
```

Usage:

```
vfcfg [ [-h hostname] [-n username -p password] ] [volume]
      [-status | -sc | -sv | -stop] [-licensing]
      [-lw latency] [-cs cachesize [-fs]] [-ps pagesize]
```

Where each switch specifies:

```
-h          Host or target computer. Default: local computer.

-n          User name (domain\user or user). Default: current user.

-p          Password for specified user account. For an empty password value use two double quotes (""). For a NULL password value do not specify a password. Default: NULL. (Requires -n)

volume     Drive letter or four-digit identifier specifying target storage volume. Default: all qualified volumes (-status only).
```

- status Display status of filter on one or all storage volumes.

- sc Activate SuperCache filter action. (Requires -cs)

- sv Activate SuperVolume filter action.

- stop Deactivate the current filter action.

- licensing Display licensing information.

- lw Lazywrite latency. 'Off', 'disable', or '0' disables deferred-write mode (no write caching). 'Inf' or 'infinite' enables deferred-write mode with lazywriting suspended (latency is infinite). A time value in seconds, or in hours, minutes and seconds (H:MM:SS), enables deferred-write mode with lazywriting active and dirty pages remaining such, on average, for the time specified. Maximum limit is 14400 seconds, or 4:00:00. Default: off. (SC and SV)

- cs Cache size (in MB). This value must be specified when activating SuperCache. The size may be modified while the cache is active. Cache allocations are made in blocks of 4 MB; the value is rounded up accordingly. No default. (SC only)

- fs Fail-safe enable. Applicable only when increasing the cache size, this switch indicates that if the cache size specified cannot be allocated, the maximum amount available should be used. Default: disabled. (SC only)

- ps Cache page size (in KB). This value may be specified only when activating SuperCache. Once active, the value can be changed only by deactivating the cache and reactivating with a new value. Default: depends on the volume's size and the amount of available memory. (SC only)

All commands may optionally specify the host (target) name, and/or a user name and password.

```
Ex. vfcfg -h server1 -n "Joe Doe" -p mln2o3 c: -status
    vfcfg -h server1 -n "Joe Doe" -p mln2o3 0001 -status
    vfcfg -h server1 -status
```

Page 3

```
vfcfg -n "Joe Doe" -p mln2o3 -status
```

The status of a particular volume may be requested, or, by the absence of a volume specification that of all qualified volumes in the target system.

```
Ex. vfcfg c: -status
    vfcfg 0001 -status
    vfcfg -h server2 -status
```

SuperCache and SuperVolume are made active through the use of the -sc and

-sv switches, respectively. When activating SuperCache, the -ps and -cs values are also required, while the -fs and -lw optionally may be specified. When activating SuperVolume, the -lw value optionally may be specified.

```
Ex. vfcfg c: -sc -ps 32 -cs 256
    vfcfg d: -sv
    vfcfg 0002 -sv
    vfcfg c: -sc -ps 32 -cs 512 -fs -lw 15
    vfcfg 0004 -sc -ps 32 -cs 512 -fs -lw 15
    vfcfg d: -sv -lw 0:01:30
    vfcfg -h server1 -n "Joe Doe" -p mln2o3 c: -sc -ps 32 -cs 256
    vfcfg -h server2 d: -sv -lw inf
```

To modify the current filter action's parameters, simply specify the new parameter(s).

```
Ex. vfcfg c: -lw inf
    vfcfg -h server2 -n "Joe Doe" -p mln2o3 d: -lw 2:30:00
```

To deactivate a filter action, specify -stop.

```
Ex. vfcfg -stop 0001
    vfcfg -stop c:
    vfcfg -h server2 d: -stop
```

To view licensing information, use the -licensing switch.

```
Ex. vfcfg -licensing
    vfcfg -h server2 -licensing
```

4.0 Cache Tuning

The various resources in a computer system – processors, buses, memory, controllers and adapters, storage devices, network interfaces, software, etc. – are all factors which determine the rate at which a given task may be executed. Most system resources are shared by the many different software components that are required to execute a task. For example, processor time, system memory, and storage devices must be shared by the operating system kernel and its drivers, operating system services, and other services and applications.

Balancing the use of system resources so as to maximize system performance is known as system performance tuning. Much of this tuning is realized automatically by the operating system. However, the operating system provides many parameters that can be modified to improve overall system performance, according to the applications or services being executed. (Search 'windows performance' on Microsoft's Web site.)

SuperCache and SuperVolume improve system performance when storage resources are over utilized compared with system memory and processor resources. However, memory and processor resources must be shared among many system components, as described earlier. It is the responsibility of the systems engineer or system administrator to balance the use of these resources in order to optimize system performance.

SuperCache and SuperVolume feature a variety of parameters to assist in system performance tuning. Each parameter is described below, along with recommended settings.

Parameter (Switch)	Product	Range (Default)	Description and use
Lazywrite latency, in seconds (-lw)	SC, SV	(off, 0), 1 - - 14400, infinite	<p>Describes the period, in seconds, during which written data are present in the cache and not on the disk. When off (or 0; write-through mode), write data are written synchronously to the cache and disk.</p> <p>When not 0 (write-deferred mode), write data are written to the cache and asynchronously to the disk. The cache is called 'dirty' while the written data are present in the cache but are not yet written to disk. The amount of time between when the data are written to the cache and when they are written to the disk is the 'lazywrite latency'.</p> <p>When infinite, write data are written to the cache only as required to make new cache mappings. As old pages age, they become eligible for cleaning to be reused for new mappings.</p> <p>Deferred writing trades improved write performance with greater risk of data loss. The longer the period, the lower the rate at which dirty data will be written out to disk by the lazywriter. Cache performance may be improved by lowering this rate (increasing the latency.) However, the longer dirty data are present in the cache, the greater the risk of data loss and corruption in the event of a power failure or system crash.</p> <p>Important: A nonzero lazywrite latency setting is an average: some dirty data will be written out in a shorter period, some will be written out after a longer period. We recommend use of this setting only on systems with uninterruptible power supplies (UPS).</p> <p>SuperCache: Unlike SuperVolume, SuperCache also writes out dirty data in response to internal requests for cache page mappings. If the cache become full of dirty data and read or write requests are made for pages not in the cache, dirty pages will be forcibly written out. This operation is given priority over the lazywriter.</p> <p>SuperVolume: Unlike SuperCache, during initialization, SuperVolume automatically loads the entire volume image into its cache. During this time, the cache write mode is always write-through. Upon completion of the image load, the write mode is changed to the configured setting. While loading, the configured setting will be reported even though it may differ from the actual setting.</p>
Cache size, in MB (cs)	SC	depends on page size (no default)	<p>Describes the size of the cache, in megabytes. This is the single most important parameter to consider for cache tuning. For most server systems, 1 to 2 GB of physical memory (RAM) must be reserved for the operating system, services and applications. The remaining physical memory can be used effectively for SuperCache.</p> <p>On systems with more than 4 GB of RAM, SuperCache can dramatically increase performance with little worry about "starving" memory from other system components. On systems with 4 GB of RAM or less, due consideration must be given to sharing memory resources.</p> <p>Care should always be exercised when attempting to allocate "as much memory as possible" to SuperCache. The system can become memory starved, and its performance greatly hindered. Under extreme circumstance, the system may crash.</p> <p>To tune cache performance, observe the following two metrics: memory utilization in Task Manager, and application performance using an appropriate application performance monitoring tool. Set the cache size to a relatively low value (it's much faster to allocate memory to the cache in a busy system than to free memory from the cache), and, over a reasonable period of time, note the values of the two metrics.</p> <p>Incrementally increase the cache size, watching its effect over time on the two metrics. At some point the performance will peak, after which it will drop again. The performance will begin to drop when the other system components start to 'feel' the lack of physical memory and the operating system begins paging out virtual memory. Reset the cache size to the value where the highest performance was realized. The cache is now rough-tuned. Use the other parameters available to fine-tune the cache's performance.</p>
Page size, in KB (ps)	SC	4, 8, 16, 32, 64, 128, 256	<p>Describes the granularity of the cache and of volume mappings, in kilobytes. It is similar to allocation unit size in the NTFS.</p> <p>Cache fragmentation: The smaller the page size, the lower the average invalid</p>

		<p>data per page (eg. lower cache fragmentation). This increases cache efficiency, improving the probability of cache hits Reduced cache fragmentation is often the single most important factor in improving overall cache performance.</p> <p>Cache overhead: The smaller the page size the greater the system resources required for cache overhead. The physical and virtual memory resources required to manage the cache pages is greater, since there are more pages for a given cache size. Similarly, the physical and virtual memory resources required to manage the volume map is greater, since there are more mappings for a given volume size. And the memory and processor resources required for executing an I/O request are greater, since more pages must be managed per transfer.</p> <p>Other factors: Among the factors to include in determining the cache page size are: the average transfer size, the ratio of reads to writes, and the state of file-system caching (enabled or disabled). Generally, larger page sizes should be used only when transfer sizes are large. The default Windows NT file-system to storage-class driver transfer size is 64 KB. Disk data compression algorithms are optimized to 64 KB.</p> <p>When file-system caching is disabled (files are opened with the FILE_WRITE_THROUGH option), or the storage volume is accessed in RAW mode, the average transfer size is tied directly to the application. PerfMon can report the average size of the transfers, and the number can be used to calculate and set the appropriate cache page size.</p>
--	--	---

~~~~~