Intel® RAID Controller SRCS16

Performance-Tuning White Paper

Revision 1.0

March 18, 2005

Enterprise Platforms and Services Division - Marketing

Revision History

Date	Revision Number	Modifications
March 18, 2005	1.0	Initial release.

Disclaimers

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Intel® RAID Controller SRCS16 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel, Pentium, Itanium, and Xeon are trademarks or registered trademarks of Intel Corporation.

*Other brands and names may be claimed as the property of others.

Table of Contents

1. Overview

Getting the best performance from a RAID subsystem is both an art and a science. Specific rules can be used in known configurations to enhance performance, but in most cases you will need to analyze the system I/O patterns to determine the appropriate changes that need to be made. Faster processors, faster and higher density system RAM, and faster front-side bus architectures will increase the performance of most applications.

Many factors affect overall system performance. This white paper discusses only the practical application of changing the configuration settings for the Intel® RAID controller SRCS16, and the settings for the relative hard disk and logical array to adjust the RAID I/O performance.

2. Performance Factors

To calculate the read or write performance of an array, the number of simultaneous I/O operations that can be performed on the array is divided by the time taken to perform an I/O operation. The result is the number of I/O operations the array can perform per second.

Several factors affect the performance of the RAID subsystem. The following RAID subsystem factors directly contribute to performance:

- Available PCI bus bandwidth
- RAID Logical Drive Cache setting
- Stripe size
- Disk cache
- RAID level
- Ratio of read versus write operations
- Ratio of sequential versus random operations
- Number of disks in an array

3. Available PCI Bus Bandwidth

All I/O from a PC-based Host Bus Adapter (HBA) to the RAID array must pass through the PCI bus. Essentially, this bus is the main data conduit through which all data passes to and from all PCI devices. Since the HBA directly controls I/O to the RAID array, the throughput of the PCI bus is extremely important. Consider the following industry standard PCI formula:

```
PCI bus throughput (MBps) = PCI bus width (32-bit, 64-bit)/8*Bus
Speed (33.3MHz, 66.6MHz, 133MHz)
```

Given the formula, the following are the maximum data throughputs for associated standard PCI bus types:

- PCI 32-bit 33MHz = 133MBps
- PCI 32-bit 66MHz = 266MBps
- PCI 64-bit 33MHz = 266MBps
- PCI 64-bit 66MHz = 533MBps
- PCI-X* 64-bit 133MHz = 1066MBps

Intel RAID Controller SRCS16 storage adapters provide a high-performance intelligent peripheral component interconnect to high-speed serialized AT attachment (PCI-to-Serial ATA) interface with redundant array of independent disks (RAID) control capabilities. The RAID Controller SRCS16 is a PCI 64-bit 66MHz adapter that conforms to the *PCI Local Bus Specification*, Revision 2.2, and is backward compatible with previous versions of the specification.

The total throughput is basically the maximum amount of data that is able to pass through the entire PCI bus at any one time. These numbers are theoretical limits. It is not possible to transfer data up to these speeds because the PCI bus has an inherent operational overhead that usually consumes about 15-20% of the theoretical bandwidth. Because a PCI bus is a shared bus (except PCI Express*), to get best performance for the RAID controller, the RAID Controller SRCS16 should be installed into a PCI slot on a bus with no other PCI device.

4. Available SATA Bus Bandwidth

The implementation of the SATA specification allows for 150MB/s point-to-point transactions for devices meeting the SATA I specification or 300MB/s for devices meeting the SATA II extension specification. A 5-10% bus management overhead must be considered. SATA drives may burst at the maximum data transfer rate, but sustained transfer rates are much lower for hard disks currently available.

5. RAID Logical Drive Cache Policy

Cache-to-cache I/O is much faster than any other type of I/O operation occurring on the SATA bus. Therefore, an increase in HBA cache memory allows more data to be queued for cache-only operation. The type of cache operation that is configured on the HBA must qualify this supposition, because it will affect how the cache is used during typical I/O operations. There is a point of diminishing returns in adding HBA cache memory, and the performance benefit of adding cache memory is affected by the type of application IO transactions. Real-world application testing indicates that for most applications, little benefit is gained by adding cache memory above 256MB.

The RAID Controller SRCS16 cache policies consist of read and write functions. Read policies are comprised of one of the following: read-ahead, adaptive, and normal (no read-ahead policy). Write policies can be divided into two types: write-back and write through. The cache options and settings of Intel RAID controllers can be unique for each logical drive. This section describes the methods used by the SRCS16 RAID controller to cache data for host I/O.

Note: Generally, when write back caching is used and the system I/O involves a large percentage of write operations, adding more cache to a controller can get better write performance. However, additional cache usually has a little benefit on read performance because most operating systems have a built-in data caching feature.

5.1.1 Cached IO Policy

The cache policy applies to I/O on a specific logical drive. It does not affect the read ahead cache. The options are Cached I/O or Direct I/O. Cached I/O buffers all reads in cache memory. Direct I/O does not buffer reads in cache memory. When possible, Direct I/O does not override the cache policy settings. Direct I/O transfers data to cache and the host concurrently. If the same data block is read again, the host reads it from cache memory.

- **Cached I/O:** All reads will first look at cache. If a cache hit occurs, the data will be read from cache; if not, the data will be read from disk and the read data will be buffered into cache. All writes to disk are also written to cache.
- **Direct I/O:** When possible, no cache is involved for both reads and writes. The data transfers will be directly from host to disk and from disk to host.

5.1.2 Write Policy

You can set the write policy to Write-back or Write-through. In Write back caching, the controller sends a data transfer completion signal to the host when the controller cache receives all the data in a transaction. In Write through caching, the controller sends a data transfer completion signal to the host after the disk subsystem receives all the data in a transaction. Write-through caching has a data security advantage over write-back caching. Write-back caching has a performance advantage over write-through caching, but it should only be enabled when the optional battery backup module is installed.

- **Write Back:** I/O completion is signaled when data is transferred to cache.
- **Write Through:** I/O completion is signaled only after the data is written to hard disk.

5.1.3 Read Policy

You can set Read Policy to Normal, Read-ahead, or Adaptive. Normal specifies that the controller does not use read-ahead for the current logical drive. Read-ahead specifies that additional consecutive stripes are read and buffered into cache. This option will improve performance for sequential reads. Adaptive specifies that the controller begins using read ahead if the two most recent disk accesses occurred in sequential sectors.

- **Normal (No Read Ahead):** Provides no read ahead for the logical drive.
- **Read-ahead:** Additional consecutive stripes/lines are read and buffered into cache.
- **Adaptive:** The read-ahead will be automatically turned on and off depending upon whether the disk is accessed for sequential reads or random reads.

6. Stripe Size

For I/O intensive or small block random access database accesses, striping the hard disks in the array with stripes larger than a single record, so that a record falls entirely within one or two stripes, will optimize performance. For data intensive environments or large block sequential access systems that access large records, small stripes (512-byte) cause each record to span across all the hard disks in the array. With each disk storing a portion of the data from the record, accesses are faster because the data transfer interleaves onto multiple disks. However, small stripes rule out multiple overlapped data operations because each access will typically involve all disks. Applications that utilize long record accesses, such as on-demand video, document management, or data acquisition, work best with small stripe arrays.

Small stripes require synchronized spindle disks to prevent degraded performance when accessing short records. Without synchronized spindles, each disk in the array may be at a different rotational position from when their data was written. Completing a disk access requires waiting until each disk has accessed its portion of the record, which can take an extra rotation of the disk platter on one or more disks. The more disks in the array, the longer the average access time for the array. Synchronized spindles ensure that every disk in the array reaches its data during the same rotation of their respective platters. The access time of the array becomes equal to the average access time of a single disk instead of approaching the product of access time and the number of disks in the array.

7. Hard Disk Cache

Note: Disk caching algorithms (how the hard disk caches data) vary by manufacturer.

The disk cache provides high performance for a sequential read access. Reading data that the host computer has not yet requested into the data buffer is done in advance which allows the data to be directly transferred when requested resulting in lower latency.

On the other hand, the disk cache enhances write performance. As soon as the device receives all of the data into its buffer, the device reports to the host that it completed the write command. The device assumes responsibility to write the data to the disk.

During a data write that follows the acknowledgment of a write command, a soft or hard reset does not affect the write operation. However, a power-off immediately terminates the write operation. A power-off while the write cache is enabled may cause the loss of the data that remains in the cache. There is a possibility that a power-off, even after write command completion, may cause a loss of data from the write cache.

There is no way to provide a battery backup of the data that is temporarily stored in the hard disk cache but has not been written to disk. The addition of an uninterruptible power supply (UPS) may add additional security but should not be considered to be a guarantee that data cannot be lost.

No more than one sector can be lost by power down during write operation while the write cache is disabled. A power-off during a write operation may create an incomplete sector. This would be reported as a data error when read. The sector can be recovered by a re-write operation. A hard reset does not cause any data loss.

Command queuing allows the host to issue concurrent commands to the same device so that the hard disk can deal with more I/O at the same time. SATA I drives do not support native command queuing. Because of this, a disk cache has added benefit. SCSI technology and SATA II technology allow multiple commands on the bus at the same time. This enhances the performance and the benefits of a disk cache are not as pronounced.

With disk cache on, the RAID Controller SRCS16 subsystem performance increases. However, data could be lost in the event of a system failure or power loss because the disk has no mechanism to back-up the disk cache. Because of this risk, the RAID Controller SRCS16 RAID automatically disables the disk cache. The user can enable the disk cache by using a command-line tool or other utility. The disk cache setting remains persistent during reboot when the disk cache policy is changed.

8. Guidelines for Optimizing Performance

Note: Because of the performance factors mentioned earlier in this document, it is difficult to see the actual data transfer rates for an individual device. The goal of this document is to help you achieve the best acceptable performance for your RAID subsystem, not the maximum possible performance.

The optimization of the overall performance of your RAID subsystem requires careful pre-analysis and integrative consideration of several factors and their interaction in the actual application. This section discusses those factors and how they can used to maximize performance.

8.1 System Cache Memory

When configuring a server, you should always consider the effect of the operating system cache on the RAID subsystem. In effect, the operating system cache acts as a read filter for the RAID subsystem. The operating system cache can eliminate many disk read operations, but all write operations must go through to the disk.

Note: Most operating systems have caching capability to automatically manage the size of their disk cache, but this dynamic resizing can dramatically reduce availability of memory for application processing. Some operating systems have the capability to manually set a size limit for the operating system's disk cache that can provide better performance without consuming excessive amounts of system memory.

8.2 RAID Cache Memory

The Intel RAID controller cache is used for predictive reads, write-backs, and as temporary storage during RAID 5 parity calculations. Because the operating system cache size is increased, more read hits are serviced from the operating system cache, resulting in fewer reads issued to the RAID subsystem. However, the number of disk writes stays relatively constant. At some point, increasing the cache for the operating system will not result in significantly reduced disk I/O.

8.3 RAID Logical Drive Cache Policy

Environments with a large number of sequential reads and predictive caching should generate a high number of Read-Ahead Hits relative to Total Sectors. These cache hits will reduce the number of seek operations and increase overall performance.

With the write-back policy enabled on an HBA, the I/O performance can be improved because the HBA can move to the next task more quickly when writing to cache memory. It does not need to wait for the data to be written to disk. The downside to a write-back policy is that data will be lost or can become corrupted if a system failure occurs before the data in the cache is written to the disk. Intel RAID controllers have battery backup options to ensure cache viability when operating under a write-back cache policy.

Note: Enabling the write-back cache has a risk of data corruption in the event of a power outage. To avoid this risk, Intel provides a battery backup unit (BBU) for the cache memory. Installing the BBU ensures the data will be restored correctly when power is restored.

8.3.1 Hard Disk Cache Setting

To enhance performance, the cache of the hard disk can be enabled. The risk of enabling the disk cache is that the data can be lost or become corrupted if an unexpected system power off occurs before the data in the cache is written to disk. An UPS should be used if the disk cache is enabled. A command-line utility is used to enable the disk cache.

8.3.2 Stripe Size

Choose the stripe size relative to both the I/O segment size and the number of hard disks in the array, so that most I/O operations either:

- Cross many stripes and involve all hard disks in the array

Or

- Do not cross stripes and involve only one hard disk

9. Summary

Intel is committed to providing customers with a stable, high-performance and high-reliability product. In this document we have tried to advise you of potential risks and trade-offs. Hopefully, this will enable you to configure your RAID subsystem to best meet your needs.

Intel® RAID controllers are configured with a default setting of hard disk cache disabled. Disabling the hard disk cache provides increased data protection in the event of system failure. The Intel RAID Controller SRCS16 for SATA I will experience a decrease in performance with the hard disk cache disabled.