Date: August 31, 2006

To: T10 Committee (SCSI)

From: Roger Cummings (Symantec)

Subject: T10/06-393r0 On-disk bitmap support

Revision History

06-393r0 (August 31, 2006) Original

Background

Distributed applications are making increasing use of bitmap structures to track state and control the operation of various of their features. A number of these bitmap structures are located on disk storage in order that they can survive the failure of one or more servers. While some of these bitmaps are related to the storage on which they reside (e.g. a 'dirty" sector list or a mirror resynchronization scoreboard), many are related to general application features such as:

- Cluster membership
- Cache Status
- Database table manipulation
- Parallelized processing of large data sets

In order to change the status of a single bit in such a bitmap, an application has to read all or a portion of the bitmap structure, change the status of the desired bit, and write the result back to the storage. However if during that process another Initiator begins a similar process to update a different bit, the state of the bitmap can become corrupted.

It is therefore necessary to establish at least "write exclusive" access to the Logical Unit containing the bitmap for the entire duration of the bitmap update process. This represents a significant impediment to the operation of the distributed application, given that by definition it relies on shared access to storage.

Write exclusive access can either be established within the distributed application by the use of global locks and/or a designated bitmap manager, or by the use of SCSI Reservations or Persistent Reservations. The former creates a bottleneck and single point of failure within the application, the latter adds significant overhead to the storage access as follows:

Consider the case where a number of servers running a cluster application are accessing a set of shared storage devices. In normal operation, each server Registers with each device, and an All Registrants SCSI Persistent Reservation is established with each device. However when a server needs to update a bit in a bitmap hosted on one of the shared storage devices it has to:

1) Preempt the existing All Registrants Persistent Reservation, and establish a Write Exclusive Persistent Reservation;
2) Read all or part of the bitmap structure from the storage;
3) Change the state of one or more bits in the bitmap structure;
4) Write all or part of the bitmap structure to the storage;
5) Preempt the existing Write Exclusive Persistent Reservation, and re-establish an All Registrants Persistent Reservation.

During the above process, none of the other servers running the cluster application will be able to access other data on the same storage device as the bitmap structure. After completion of the above process, all of the other servers running the cluster application will also need to reRegister with that storage device.

Proposal

The need to establish write exclusive access during a bitmap update could be avoided if a single SCSI command was capable

of changing the state of one or more bits in the bimap structure regardless of the setting of the rest of the structure. This would allow the “read-update-write” cycle to take place within the Device Server as an uninterrrupted series of actions. This document proposes the definition of a new ORWRITE command for SBC-3 to perform this function.

Note that it is important that this new command only be able to set a bit in the bitmap to a one, to ensure that the final state of the bitmap is independent of the order in which the commands against it are processed. This is the reason that a new command is needed, as none of the existing X?WRITE commands will serve the same purpose.

Suggested Changes

Add the following row to Table 3

Table 3 — SBC-2 commands that are allowed in the presence of various reservations

Command	Addressed logical unit has this type of persistent reservation held by another I_T nexus				
	From any I_T nexus		From registered I_T nexus (RR all types)	From I_T nexus not registered	
	Write Exclusive	Exclusive Access		Write Exclusive - RR	Exclusive Access - RR
ORWRITE	Conflict	Conflict	Allowed	Conflict	Conflict

Key: RR = Registrants Only or All Registrants
Allowed: Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present should complete normally.
Conflict: Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present shall not be performed and the device server shall terminate the command with RESERVATION CONFLICT status.

[a] Logical units claiming compliance with previous versions of this standard (e.g., SBC-2) may return RESERVATION CONFLICT in this case. Logical units may report whether certain commands are allowed in the PERSISTENT RESERVE IN command REPORT CAPABILITIES service action parameter data ALLOW COMMANDS field (see SPC-4).

Add the following row to Table 12

Table 12 — Commands for direct-access block devices

Command name	Operation code [a]	Type [b]	Protection information	Reference
ORWRITE (16)	xxh	O	Yes	5.xx
The following operation codes are vendor-specific: 02h, 05h, 06h, 09h, 0Ch, 0Dh, 0Eh, 0Fh, 10h, 11h, 13h, 14h, 19h, 20h, 21h, 22h, 23h, 24h, 26h, 27h, 29h, 2Ch, 2Dh, and C0h through FFh. All operation codes for direct-access block devices not specified in this table are reserved for future standardization.				

[a] Some commands are defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.
[b] M = command implementation is mandatory. O = command implementation is optional. X = Command implementation requirements detailed in the reference.
[c] Application clients should migrate from READ (6) to READ (10) (see 5.6) and from WRITE (6) to WRITE (10) (see 5.25).
[d] READ CAPACITY (16) is mandatory if protection information is supported and optional otherwise.
[e] If the SCCS bit is set to one in the standard INQUIRY data (see SPC-4), these commands shall be supported as required by SCC-2. If the SCCS bit is set to zero, these commands shall not be supported.
[f] This command shall be supported if the ENCSERV bit is set to one in the standard INQUIRY data (see SPC-4) and may be supported otherwise.

Add the following command definition:

5.XX ORWRITE (16) command

The ORWRITE (16) command (see table yy) requests that the device server perform the following as an uninterrupted series of actions:

1) read the specified logical block(s);
2) transfer logical blocks from the data-out buffer;
3) perform an OR operation with the logical blocks transferred from the data-out buffer and the logical blocks read, storing the resulting OR data in a buffer; and
4) write the logical blocks from the buffer.

Each logical block includes user data and may include protection information, based on the wrprotect field and the medium format.

Table yy — ORWRITE (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (??h)							
1	WRPROTECT			DPO	FUA	Reserved	FUA_NV	Reserved
2	(MSB)							
	LOGICAL BLOCK ADDRESS							
9								(LSB)
10	(MSB)							
	TRANSFER LENGTH							
13								(LSB)
14	Reserved			GROUP NUMBER				
15	CONTROL							

See the WRITE (10) command (see 5.26) for the definitions of the wrprotect field, the fua bit, and the fua_nv bit. See the READ (10) command (see 5.7) for the definition of the dpo bit. See the PRE-FETCH (10) command (see 5.3) for the definition of the logical block address field. See the PRE-FETCH (10) command (see 5.3) and 4.17 for the definition of the group number field.

The transfer length field specifies the number of contiguous logical blocks of data that are read, transferred from the data-out buffer, and ORed into a buffer, starting with the logical block specified by the logical block address field. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The transfer length field is constrained by the maximum transfer length field in the Block Limits VPD page (see 6.4.2).