From: Gerry Houlder, Seagate Technology <gerry.houlder@seagate.com>
Subj: SPC-4: Persistent Reservation fixes
Date: Jan. 8, 2008

Overview

I have noted some spots in description of persistent reservations that I believe are either unclear or wrong. This proposal is an attempt to fix those items.

(a) In clause 5.6.9 there is a description of how to become a reservation holder. Item a) has the rules for All Registrants reservation types (where more than one initiator can simultaneously be a reservation holder). Item b) describes “all other types”, which boil down to types where only one initiator is the reservation holder.

Item b) doesn’t include the RESERVE service action (which is the main way an initiator becomes the reservation holder) and it includes the two register service actions (which would make sense for an All Registrants reservation but not for this type). The Reserve service action needs to be added and the registration service actions should be deleted.

(b) In clause 6.14.3, support for the SPEC_I_PT bit is clearly described for one service action, one other service action is described as not supporting it, and the standard is silent on the other service actions. There should be words added to clarify that the other service actions also do not support the SPEC_I_PT bit.

(c) In clause 5 6.7 the draft standard uses the term “move the reservation”. I think this is unclear, it should say to establish a new reservation holder. I found a number of other clauses that discussed “moving” a reservation and have proposed new wording for each of these cases.

5.6.9 Persistent reservation holder

The persistent reservation holder is determined by the type of the persistent reservation as follows:

a) For a persistent reservation of the type Write Exclusive – All Registrants or Exclusive Access – All Registrants, the persistent reservation holder is any registered I_T nexus; or

b) For all other persistent reservation types, the persistent reservation holder is the I_T nexus:

   A) For which the reservation was established with a PERSISTENT RESERVE OUT command with ~~REGISTER service action, REGISTER AND IGNORE EXISTING KEY~~ RESERVE service action, PREEMPT service action, or PREEMPT AND ABORT service action; or

   B) ~~To which the reservation was moved~~ that became the reservation holder by a PERSISTENT RESERVE OUT command with REGISTER AND MOVE service action.

[Note – no other changes in this clause.]

6.14.3 Basic PERSISTENT RESERVE OUT parameter list

[Note – paragraphs before the SPEC_I_PT bit description are unchanged.]

If the Specify Initiator Ports (SPEC_I_PT) bit is set to zero, the device server shall apply the registration only to the I_T nexus that sent the PERSISTENT RESERVE OUT command. If the SPEC_I_PT bit is set to one for any service action except the REGISTER ~~AND IGNORE EXISTING KEY~~ service action, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the SPEC_I_PT bit is set to one for the REGISTER service action, the additional parameter data (see table 133) shall include a list of transport IDs and the device server shall also apply the registration to the I_T nexus for each initiator port specified by a TransportID. If a registration fails for any initiator port (e.g., if the logical unit does not have enough resources available to hold the registration information), none of the other registrations shall be made.

[Note – No other changes in this clause.]

5.6.2 Third party persistent reservations

Except for all registrants type reservations, a reservation holder (see 5.6.9) may ~~move the persistent reservation to~~ establish a third party as the reservation holder (e.g., a copy manager supporting the EXTENDED COPY command) using the REGISTER AND MOVE service action (see 5.6.7). A copy manager supporting the EXTENDED COPY command may be instructed to ~~move the persistent reservation to~~ establish a specified I_T nexus as the reservation holder using the third party persistent reservations source I_T nexus segment descriptor (see 6.3.7.19).

5.6.7 Registering and moving the reservation

The PERSISTENT RESERVE OUT command REGISTER AND MOVE service action is used to register a specified I_T nexus (see table 36) and ~~move the reservation to~~ establish that I_T nexus as the reservation holder.

[Note – unchanged table and intervening paragraphs are not shown.]

In response to a PERSISTENT RESERVE OUT command with a REGISTER AND MOVE service action the device server shall perform a register and move by doing the following as an uninterrupted series of actions:

a) Process the APTPL bit;
b) Ignore the contents of the SCOPE and TYPE fields;
c) Associate the reservation key specified in the SERVICE ACTION RESERVATION KEY field with the I_T nexus specified as the destination of the register and move, where:
    A) The I_T nexus is specified by the TransportID and the RELATIVE TARGET PORT IDENTIFIER field (see 6.14.4); and
    B) Regardless of the TransportID format used, the association for the initiator port is based on either the initiator port name (see 3.1.53) on SCSI transport protocols where port names are required or the initiator port identifier (see 3.1.52) on SCSI transport protocols where port names are not required;
d) Register the reservation key specified in the SERVICE ACTION RESERVATION KEY field;
e) Retain the reservation key specified in the SERVICE ACTION RESERVATION KEY field and associated information;
f) Release the persistent reservation for the persistent reservation holder (i.e., the I_T nexus on which the command was received);
g) ~~Move~~ establish the specified I_T nexus as the persistent reservation holder of a reservation with ~~to the specified I_T nexus using~~ the same scope and type as the persistent reservation released in item f); and
h) If the UNREG bit is set to one, unregister (see 5.6.10.3) the I_T nexus on which PERSISTENT RESERVE OUT command was received.

It is not an error for a REGISTER AND MOVE service action to register an I_T nexus that is already registered with the same reservation key or a different reservation key.

6.14.2 PERSISTENT RESERVE OUT service actions

When processing the PERSISTENT RESERVE OUT service actions, the device server shall increment the PRgeneration value as specified in 6.13.2.

The PERSISTENT RESERVE OUT command service actions are defined in table 131.

Table 131 — PERSISTENT RESERVE OUT service action codes

Code	Name	Description	PRgeneration field incremented	Parameter list format
00h	REGISTER	Register a reservation key with the device server (see 5.6.6) or unregister a reservation key (see 5.6.10.3).	Yes	Basic (see 6.14.3)
01h	RESERVE	Creates a persistent reservation having a specified SCOPE and TYPE (see 5.6.8). The SCOPE and TYPE of a persistent reservation are defined in 6.13.3.3 and 6.13.3.4.	No	Basic (see 6.14.3)
02h	RELEASE	Releases the selected persistent reservation (see 5.6.10.2).	No	Basic (see 6.14.3)
03h	CLEAR	Clears all reservation keys (i.e., registrations) and all persistent reservations (see 5.6.10.6).	Yes	Basic (see 6.14.3)
04h	PREEMPT	Preempts persistent reservations and/or removes registrations (see 5.6.10.4).	Yes	Basic (see 6.14.3)
05h	PREEMPT AND ABORT	Preempts persistent reservations and/or removes registrations and aborts all tasks for all preempted I_T nexuses (see 5.6.10.4 and 5.6.10.5).	Yes	Basic (see 6.14.3)
06h	REGISTER AND IGNORE EXISTING KEY	Register a reservation key with the device server (see 5.6.6) or unregister a reservation key (see 5.6.10.3).	Yes	Basic (see 6.14.3)
07h	REGISTER AND MOVE	Register a reservation key for another I_T nexus with the device server and establish that I_T nexus as a reservation holder for ~~move~~ a persistent reservation ~~to that I_T nexus~~ (see 5.6.7)	Yes	Register and move (see 6.14.4)
08h – 1Fh	Reserved			

[Note – No other changes in this clause.]

7.5.4.1 Overview of TransportID identifiers

An application client may use a TransportID to specify an initiator port other than the initiator port that is transporting the command and parameter data (e.g., as an Access identifiers (see 8.3.1.3.2) in ACL ACEs)~~,~~ as the initiator port to establish as the reservation holder using a ~~in the I_T nexus to which~~ PERSISTENT RESERVE OUT command with REGISTER AND MOVE service action (see 5.6.7) ~~is moving a persistent reservation)~~.

[Note – No other changes in this clause.]

B.3 Third party reservations

For some uses of the EXTENDED COPY command (see 6.3), the application client performs a locking function to maintain data integrity on the source and may also lock the destination device prior to starting the copy operation. The persistent reservation management method may be used to perform the locking function. Other methods (e.g., access controls, see 8.3) may also perform the locking function.

To accomplish a third party persistent reservation the following steps are recommended:

1) Backup application uses the REGISTER service action to register an I_T nexus with a logical unit (e.g., a tape drive logical unit);
2) Backup application uses the RESERVE service action to establish a persistent reservation with the Exclusive Access type;
3) Backup application prepares the logical unit for access (e.g., medium is loaded and positioned);
4) Backup application uses the REGISTER AND MOVE service action to register the I_T nexus that the copy manager is expected to use and to ~~move the persistent reservation to~~ establish that I_T nexus as the reservation holder;
5) Backup application sends the EXTENDED COPY command to the copy manager that includes a third party persistent reservations source I_T nexus segment descriptor (see 6.3.7.19);
6) Copy manager processes all segment descriptors in the received EXTENDED COPY command except the third party persistent reservations source I_T nexus segment descriptor; and
7) Copy manager issues a REGISTER AND MOVE service action, using the reservation key and I_T nexus specified in the third party persistent reservations source I_T nexus segment descriptor received from the backup application (see step 5), to ~~move the persistent reservation back to~~ reestablish the original I_T nexus as the reservation holder.