To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 7 March 2007
Subject: 07-080r1 SPC-4 Persistent reservations self-preempting clarification

Revision history

Revision 0 (26 February 2007) First revision
Revision 1 (7 March 2007) Corrected error in overview that was causing confusion - “host B sends a command between 3) and 4)” should be “after 4”

Related documents

spc4r09 - SCSI Primary Commands - 4 (SPC-4) revision 9
02-065r2 - SPC-3 Persistent reservations corrections (Rob Elliott, HP)

Overview

Given this example:

1) Host A registers key AK
2) Host B registers key BK
3) Host A reserves Write Exclusive - Registrants Only
4) Host A preempts with new type Exclusive Access - Registrants Only

If host B sends a command after 4, it should be notified of a unit attention condition with an additional sense code set to RESERVATIONS RELEASED. The summary table of 02-065r2 is the most clear description of this. In the standard, section 5.6.10.1.2 is one place that states the rule.

The only text in 5.6.10.1.1 referring to 5.6.10.1.2 is in table 36, and table 36 is introduced by wording like “if the persistent reservation holder becomes unregistered”, which is not the case in this example; host A remains registered. So, the reader might assume that 5.6.10.1.2 does not apply and assume that the unit attention condition is not created.

This proposal:

a) changes the wording introducing table 36 to cover all release and preempt cases by an application client (but not cover release due to power loss)
b) moves the table earlier, to the end of the release/preempt section list, so it is not confused with removal
c) change the header of 5.6.10.1.2 to include “preempt” not just “release”
d) change the “type or scope changed” sentence to mention it applies to the preempt case, not the release case

Editor’s Note 1: Additional issue that might be addressed in a follow-on proposal or another revision of this proposal: All the rules in 5.6.10.1.2, 5.6.10.1.3, and 5.6.10.1.4(all under the Overview subclause) seem to be duplicates of rules in 5.6.10.2 (RELEASE), 5.6.10.3 (REGISTER), 5.6.10.4 (PREEMPT), 5.6.10.5 (PREEMPT AND ABORT), and 5.6.10.6 (CLEAR). The 5.6.10.x sections might be best removed entirely. Text that seems redundant in those later sections is noted in *italics*.

Suggested changes

5.6.10 Releasing persistent reservations and removing registrations

5.6.10.1 Overview

5.6.10.1.1 Summary of service actions that release persistent reservations and remove registrations

An application client may release or preempt the persistent reservation by issuing one of the following commands through a registered I_T nexus with the RESERVATION KEY field set to the reservation key value that is registered with the logical unit for that I_T nexus:

a) A PERSISTENT RESERVE OUT command with RELEASE service action from a persistent reservation holder (see 5.6.10.2);
b) A PERSISTENT RESERVE OUT command with PREEMPT service action specifying the reservation key of the persistent reservation holder or holders (see 5.6.10.4);
c) A PERSISTENT RESERVE OUT command with PREEMPT AND ABORT service action specifying the reservation key of the persistent reservation holder or holders (see 5.6.10.5);
d) A PERSISTENT RESERVE OUT command with CLEAR service action (see 5.6.10.6); or
e) If the I_T nexus is the persistent reservation holder and the persistent reservation is not an all registrants type, then a PERSISTENT RESERVE OUT command with REGISTER service action or REGISTER AND IGNORE EXISTING KEY service action with the SERVICE ACTION RESERVATION KEY field set to zero (see 5.6.10.3).

Table 36 defines processing for a persistent reservation released or preempted by an application client based on the reservation type.

Table 36 — Processing for a released or preempted persistent reservation~~s~~ [unchanged, but moved from below]

Reservation type	Reference
Write Exclusive - Registrants Only or Exclusive Access - Registrants Only	5.6.10.1.2
Write Exclusive - All Registrants or Exclusive Access - All Registrants	5.6.10.1.3
Write Exclusive or Exclusive Access	5.6.10.1.4

An application client may remove registrations by issuing one of the following commands through a registered I_T nexus with the RESERVATION KEY field set to the reservation key value that is registered with the logical unit for that I_T nexus:

a) A PERSISTENT RESERVE OUT command with PREEMPT service action with the SERVICE ACTION RESERVATION KEY field set to the reservation key (see 5.6.10.4) to be removed;
b) A PERSISTENT RESERVE OUT command with PREEMPT AND ABORT service action with the SERVICE ACTION RESERVATION KEY field set to the reservation key (see 5.6.10.5) to be removed;
c) A PERSISTENT RESERVE OUT command with CLEAR service action (see 5.6.10.6); or
d) A PERSISTENT RESERVE OUT command with REGISTER service action or REGISTER AND IGNORE EXISTING KEY service action with the SERVICE ACTION RESERVATION KEY field set to zero (see 5.6.10.3).

When a reservation key (i.e., registration) has been removed, no information shall be reported for that unregistered I_T nexus in subsequent READ KEYS service actions until the I_T nexus is registered again (see 5.6.6). ~~As shown in table 37, the processing of any persistent reservation whose persistent reservation holder or holders become unregistered depends on the reservation type.~~

~~[Table 36]~~

Loss of power also causes persistent reservations to be released and registrations to be removed ~~Registrations and persistent reservations may also be released by a loss of power,~~ if the persist through power loss capability is not enabled. When the most recent APTPL value received by the device server is zero (see 6.14.3), a power cycle:

a) Releases all persistent reservations; and
b) Removes all registered reservation keys (see 5.6.6).

5.6.10.1.2 Processing for released or preempted Registrants Only persistent reservations

When the persistent reservation holder (see 5.6.9) of a Write Exclusive – Registrants Only or Exclusive Access – Registrants Only type reservation becomes unregistered the persistent reservation shall be released.

For every I_T nexus whose reservation key is removed, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus ~~and~~, with the additional sense code ~~shall be~~ based on the PERSISTENT RESERVE OUT command service action as follows:

a) If the service action was CLEAR, the additional sense code shall be set to RESERVATIONS PREEMPTED; or
b) If the service action was PREEMPT or PREEMPT AND ABORT, the additional sense code shall be set to REGISTRATIONS PREEMPTED.

If the persistent reservation was preempted and the TYPE or SCOPE have changed, then for every I_T nexus whose reservation key was not removed except for the I_T nexus on which the PERSISTENT RESERVE OUT command was received, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus, with the additional sense code set to RESERVATIONS RELEASED. If the TYPE or SCOPE have not changed, then no unit attention condition(s) shall be established for this reason.

If the reservation was released, then for every I_T nexus whose reservation key was not removed except for the I_T nexus on which the PERSISTENT RESERVE OUT command was received, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus, with the additional sense code set to RESERVATIONS RELEASED. If the reservation was not released, then no unit attention condition(s) shall be established for this reason.

5.6.10.1.3 Processing for released All Registrants persistent reservations

A Write Exclusive – All Registrants or Exclusive Access – All Registrants type persistent reservation shall be released when the registration for the last registered I_T nexus is removed or when the TYPE or SCOPE is changed.

The device server shall establish a unit attention condition for the initiator port associated with every registered I_T nexus whose reservation key was removed, with the additional sense code set as follows:

a) If the service action was CLEAR, the additional sense code shall be set to RESERVATIONS PREEMPTED; or
b) If the service action was PREEMPT or PREEMPT AND ABORT, the additional sense code shall be set to REGISTRATIONS PREEMPTED.

If a persistent reservation was released using a RELEASE service action, see 5.6.10.2.

5.6.10.1.4 Processing for other released persistent reservations

When the persistent reservation holder (see 5.6.9) of a Write Exclusive or Exclusive Access type reservation becomes unregistered the persistent reservation shall be released.

Editor’s Note 2: Subsequent sections seem to duplicate all the rules from 5.6.10.1.2, .3, and .4

5.6.10.2 Releasing

Only the persistent reservation holder (see 5.6.9) is allowed to release a persistent reservation.

An application client releases the persistent reservation by issuing a PERSISTENT RESERVE OUT command with RELEASE service action through an I_T nexus that is a persistent reservation holder with the following parameters:

a) RESERVATION KEY field set to the value of the reservation key that is registered with the logical unit for the I_T nexus; and
b) TYPE and SCOPE fields set to match the persistent reservation being released.

In response to a persistent reservation release request from the persistent reservation holder the device server shall perform a release by doing the following as an uninterrupted series of actions:

a) Release the persistent reservation;
b) Not remove any registration(s);
c) If the released persistent reservation is a registrants only type or all registrants type persistent reservation, the device server shall establish a unit attention condition for the initiator port associated with every registered I_T nexus other than I_T nexus on which the PERSISTENT RESERVE OUT command with RELEASE service action was received, with the additional sense code set to RESERVATIONS RELEASED; and
d) If the persistent reservation is of any other type, the device server shall not establish a unit attention condition.

The established persistent reservation shall not be altered and 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 RELEASE OF PERSISTENT RESERVATION, for a PERSISTENT RESERVE OUT command that specifies the release of a persistent reservation if:

a) The requesting I_T nexus is a persistent reservation holder (see 5.6.9); and
b) The SCOPE and TYPE fields do not match the scope and type of the established persistent reservation.

If there is no persistent reservation or in response to a persistent reservation release request from a registered I_T nexus that is not a persistent reservation holder (see 5.6.9), the device server shall do the following:

a) Not release the persistent reservation, if any;
b) Not remove any registrations; and
c) Return GOOD status.

5.6.10.3 Unregistering

An application client may remove a registration for an I_T nexus by issuing a PERSISTENT RESERVE OUT command with REGISTER service action or a REGISTER AND IGNORE EXISTING KEY service action with the SERVICE ACTION RESERVATION KEY field set to zero through that I_T nexus.

If the I_T nexus is a reservation holder, the persistent reservation is of an all registrants type, and the I_T nexus is the last remaining registered I_T nexus, then the device server shall also release the persistent reservation.

If the I_T nexus is the reservation holder and the persistent reservation is of a type other than all registrants, the device server shall also release the persistent reservation. If the persistent reservation is a registrants only type, the device server shall establish a unit attention condition for the initiator port associated with every registered I_T nexus except for the I_T nexus on which the PERSISTENT RESERVE OUT command was received, with the additional sense code set to RESERVATIONS RELEASED.

5.6.10.4 Preempting

...

[figure 3 mentions this in two flowchart boxes: b) Release persistent reservation]

...

5.6.10.4.3 Preempting persistent reservations and registration handling

An application client may preempt the persistent reservation with another persistent reservation by issuing a PERSISTENT RESERVE OUT command with PREEMPT service action or PREEMPT AND ABORT service action through a registered I_T nexus with the following parameters:

a) RESERVATION KEY field set to the value of the reservation key that is registered with the logical unit for the I_T nexus;
b) SERVICE ACTION RESERVATION KEY field set to the value of the reservation key of the persistent reservation to be preempted; and

c) TYPE and SCOPE fields set to define a new persistent reservation. The SCOPE and TYPE of the persistent reservation created by the preempting I_T nexus may be different than those of the persistent reservation being preempted.

If the SERVICE ACTION RESERVATION KEY field identifies a persistent reservation holder (see 5.6.9), the device server shall perform a preempt by doing the following as an uninterrupted series of actions:

a) Release the persistent reservation for the holder identified by the SERVICE ACTION RESERVATION KEY field;
b) Remove the registrations for all I_T nexuses identified by the SERVICE ACTION RESERVATION KEY field, except the I_T nexus that is being used for the PERSISTENT RESERVE OUT command. If an all registrants persistent reservation is present and the SERVICE ACTION RESERVATION KEY field is set to zero, then all registrations shall be removed except for that of the I_T nexus that is being used for the PERSISTENT RESERVE OUT command;
c) Establish a persistent reservation for the preempting I_T nexus using the contents of the SCOPE and TYPE fields;
d) Process tasks as defined in 5.6.1; and
e) Establish a unit attention condition for the initiator port associated with every I_T nexus that lost its persistent reservation and/or registration, with the additional sense code set to REGISTRATIONS PREEMPTED.

...

5.6.10.4.4 Removing registrations

When a registered reservation key does not identify a persistent reservation holder (see 5.6.9), an application client may remove the registration(s) without affecting any persistent reservations by issuing a PERSISTENT RESERVE OUT command with PREEMPT service action through a registered I_T nexus with the following parameters:

a) RESERVATION KEY field set to the value of the reservation key that is registered for the I_T nexus; and
b) SERVICE ACTION RESERVATION KEY field set to match the reservation key of the registration or registrations being removed.

If the SERVICE ACTION RESERVATION KEY field does not identify a persistent reservation holder or there is no persistent reservation holder (i.e., there is no persistent reservation), then the device server shall perform a preempt by doing the following in an uninterrupted series of actions:

a) Remove the registrations for all I_T nexuses specified by the SERVICE ACTION RESERVATION KEY field;
b) Ignore the contents of the SCOPE and TYPE fields;
c) Process tasks as defined in 5.6.1; and
d) Establish a unit attention condition for the initiator port associated with every I_T nexus that lost its registration other than the I_T nexus on which the PERSISTENT RESERVE OUT command was received, with the additional sense code set to REGISTRATIONS PREEMPTED.

...

5.6.10.5 Preempting and aborting

[no additional discussion of a unit attention condition or release in this section]

5.6.10.6 Clearing

Any application client may release the persistent reservation and remove all registrations from a device server by issuing a PERSISTENT RESERVE OUT command with CLEAR service action through a registered I_T nexus with the following parameter:

a) a) RESERVATION KEY field set to the value of the reservation key that is registered with the logical unit for the I_T nexus.

In response to this request the device server shall perform a clear by doing the following as part of an uninterrupted series of actions:

a) Release the persistent reservation, if any;
b) Remove all registration(s) (see 5.6.6);

c) Ignore the contents of the SCOPE and TYPE fields;
d) Continue normal processing of any tasks from any I_T nexus that have been accepted by the device server as allowed (i.e., nonconflicting); and
e) Establish a unit attention condition for the initiator port associated with every registered I_T nexus other than the I_T nexus on which the PERSISTENT RESERVE OUT command with CLEAR service action was received, with the additional sense code set to RESERVATIONS PREEMPTED.

NOTE 10 - Application clients should not use the CLEAR service action except during recovery operations that are associated with a specific initiator port, since the effect of the CLEAR service action defeats the persistent reservations features that protect data integrity.