To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 23 February 2004
Subject: 04-060r0 SPC-3 Persistent reservations type clarifications

Revision history

Revision 0 (23 February 2004) First revision

Related documents

spc3r17 - SCSI Primary Commands - 3 revision 17

Overview

In the PERSISTENT RESERVE IN command descriptions of the persistent reservation type codes (table 102), the "...associated with..." wording isn't quite right since reservations are I_T based. It doesn't communicate that the T also has to match (or not match, in this context).

Example:

> A task that requests a transfer to the storage medium or cache of the logical unit *from an initiator port that is not associated with a registered I_T nexus* shall be terminated with RESERVATION CONFLICT status.

Imagine a dual ported target where the initiator port (A) has registered through one target port (B) but not the other (C). The initiator port is "associated with a registered I_T nexus" because of the I_T=A_B registration. The intent is to allow commands from I_T=A_B but prohibit commands from I_T=A_C along with all other I_T=*_B and =*_C. The wording allows I_T=A_C as well.

The wording should be more like "tasks not associated with a registered I_T nexus that ... shall be terminated..."

Reservations can be used to block all sorts of commands, not just media access commands. The “transfer to/from the storage medium or cache" wording is too restrictive. Commands like MODE SELECT/MODE SENSE, EXTENDED COPY, etc. don’t fit into that wording. Table 33 (the big reservation table in the model clause) contains the specific requirements for each command and should be referenced instead; the (incomplete) descriptions about reads and writes in table 102 should be simplified.

Suggested changes to SPC-3

5.6.1 Persistent Reservations overview

Reservations may be used to allow a device server to process commands from a selected set of I_T nexuses (i.e., combinations of initiator ports accessing target ports) and reject commands from I_T nexuses outside the selected set. The device server uniquely identifies I_T nexuses using protocol specific mechanisms.

Application clients may add or remove I_T nexuses from the selected set using reservation commands. If the application clients do not cooperate in the reservation protocol, data may be unexpectedly modified and deadlock conditions may occur.

The persistent reservations mechanism allows multiple application clients communicating through multiple I_T nexuses to protect reservation operations across SCSI initiator device failures, which usually involve logical unit resets and involve I_T nexus losses. Persistent reservations persist across recovery actions, to provide application clients with more detailed control over reservations recovery. Persistent reservations are not reset by hard reset, logical unit reset, or I_T nexus loss.

The persistent reservation held by a failing I_T nexus may be preempted by another I_T nexus as part of the recovery process. Persistent reservations shall be retained by the device server until released, preempted, or cleared by mechanisms specified in this standard. Optionally, persistent reservations may be retained when power to the target is removed.

The PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands provide the basic mechanism for dynamic contention resolution in systems with multiple initiator ports using logical units with SCSI 3 multiple target ports.

Before a persistent reservation may be established, an application client shall register each I_T nexus with a device server using a reservation key. Reservation keys are necessary to allow:

a) Authentication of subsequent PERSISTENT RESERVE OUT commands;
b) Identification of other I_T nexuses that are registered;
c) Identification of the reservation key(s) that have an associated persistent reservation;
d) Preemption of a persistent reservation from a failing or uncooperative I_T nexus; and
e) Multiple I_T nexuses to participate in a persistent reservation.

The reservation key provides a method for the application client to associate a protocol-independent identifier with a registered I_T nexus. The reservation key is used in the PERSISTENT RESERVE IN command to identify which I_T nexuses are registered and which I_T nexus, if any, holds the persistent reservation. The reservation key is used in the PERSISTENT RESERVE OUT command to register an I_T nexus, to verify the I_T nexus being used for the PERSISTENT RESERVATION OUT command is registered, and to specify which registrations or persistent reservation to preempt.

Reservation key values may be used by application clients to identify registered I_T nexuses, using application specific methods that are outside the scope of this standard. This standard provides the ability to register no more than one reservation key per I_T nexus. Multiple I_T nexuses may use the same reservation key for a logical unit accessed through the same target port. Multiple initiator ports may use the same reservation key value for a logical unit accessed through the same target ports. An initiator port may use the same reservation key value for a logical unit accessed through different target ports. The logical unit shall maintain a separate reservation key for each I_T nexus, regardless of the reservation key's value.

An application client may register an I_T nexus with multiple logical units in a SCSI target device using any combination of unique or duplicate reservation keys. These rules provide the ability for an application client to preempt multiple I_T nexuses with a single PERSISTENT RESERVE OUT command, but they do not provide the ability for the application client to uniquely identify the I_T nexuses using the PERSISTENT RESERVE commands.

See table 106 in 6.12.2 for a list of PERSISTENT RESERVE OUT service actions. See table 97 in 6.11.2.1 for a list of PERSISTENT RESERVE IN service actions.

The scope of a persistent reservation (see table 101 in 6.11.4.2) shall be the entire logical unit.

~~Persistent reservations may be further qualified by restrictions on types of access (e.g., read, write). However, any restrictions based on the type of reservation are independent of the scope of the reservation.~~

The type of a persistent reservation (see table 102 in 6.11.4.3) defines the selected set of I_T nexuses for which the persistent reservation places restrictions on commands.

~~Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation.~~ The details of which commands are allowed under what types of reservations are described in table 33.

In table 33 and table 34 the following key words are used:

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 a RESERVATION CONFLICT status.

Commands from I_T nexuses holding a reservation should complete normally. The behavior of commands from registered I_T nexuses when a registrants only or all registrants type persistent reservation is present is specified in table 33 and table 34.

An unlinked command shall be checked for reservation conflicts before the task containing that command enters the enabled task state. The reservation state as it exists when the first command in a group of linked commands enters the enabled task state shall be used in checking for reservation conflicts for all the commands in the task.

Once a task has entered the enabled task state, the command or commands comprising that task shall not be terminated with a RESERVATION CONFLICT due to a subsequent reservation. Any command in a group of linked commands that changes the reservation state shall be the last command in the group.

For each command, this standard or other command standard (see 3.1.17) defines the conditions that result in RESERVATION CONFLICT. Command standards define the conditions either in the device model (preferred) or in the descriptions each specific command.

...

6.11.4.3 Persistent Reservations Type

The value in the TYPE field shall specify the characteristics of the persistent reservation being established for all logical blocks within the element or within the logical unit. ~~Table 102 defines the characteristics of the different type values. For each persistent reservation type, table 102 lists code value and describes the required device server support. In table 102, the description of required device server support is divided into three paragraphs:~~

~~1) A definition of the required handling for read operations;~~
~~2) A definition of the required handling for write operations; and~~
~~3) A definition of the persistent reservation holder (see 5.6.7).~~

Table 33 (see 5.6.1) defines the persistent reservation types under which each command defined in this standard is allowed to be processed. Each other command set standard defines the persistent reservation types under which each command defined in that command set standard is allowed to be processed..

Table 33 — Persistent reservation type codes [replaces current table 33]

Code	Name	Description
0h	Obsolete	
1h	Write Exclusive	Some commands (e.g., media-access writes) are only allowed for a single persistent reservation holder (see 5.6.7).
2h	Obsolete	
3h	Exclusive Access	Some commands (e.g., media-access commands) are only allowed for a single persistent reservation holder (see 5.6.7).
4h	Obsolete	
5h	Write Exclusive – Registrants Only	Some commands (e.g., media-access writes) are only allowed for registered I_T nexuses. There is a single persistent reservation holder (see 5.6.7).
6h	Exclusive Access – Registrants Only	Some commands (e.g., media-access commands) are only allowed for registered I_T nexuses. There is a single persistent reservation holder (see 5.6.7).
7h	Write Exclusive – All Registrants	Some commands (e.g., media-access writes) are only allowed for registered I_T nexuses. There may be more than one persistent reservation holder (see 5.6.7).
8h	Exclusive Access – All Registrants	Some commands (e.g., media-access commands) are only allowed for registered I_T nexuses. There may be more than one persistent reservation holder (see 5.6.7).
9h-Fh	Reserved	

~~Table 33 — Persistent reservation type codes~~

Code	Name	Description
~~0h~~		~~Obsolete~~
~~1h~~	~~Write Exclusive~~	~~**Reads Shared**: Any application client on any I_T nexus may initiate tasks that request transfers from the storage medium or cache of the logical unit to the I_T nexus.~~ ~~**Writes Exclusive**: Any task from any I_T nexus other than the I_T nexus holding the persistent reservation that requests a transfer from the I_T nexus to the storage medium or cache of the logical unit shall be terminated with RESERVATION CONFLICT status.~~ ~~**Persistent Reservation Holder**: The I_T nexus that delivered the PERSISTENT RESERVE OUT command with RESERVE, PREEMPT, or PREEMPT AND ABORT service action as identified by its registered reservation key.~~
~~2h~~		~~Obsolete~~
~~3h~~	~~Exclusive Access~~	~~**Reads Exclusive**: Any task from any I_T nexus other than the I_T nexus holding the persistent reservation that requests a transfer from the storage medium or cache of the logical unit to the I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Writes Exclusive**: Any task from any I_T nexus other than the I_T nexus holding the persistent reservation that requests a transfer from the I_T nexus to the storage medium or cache of the logical unit shall be terminated with RESERVATION CONFLICT status.~~ ~~**Persistent Reservation Holder**: The I_T nexus that delivered the PERSISTENT RESERVE OUT command with RESERVE, PREEMPT, or PREEMPT AND ABORT service action as identified by its registered reservation key.~~
~~4h~~		~~Obsolete~~
~~5h~~	~~Write Exclusive – Registrants Only~~	~~**Reads Shared**: Any application client on any I_T nexus may initiate tasks that request transfers from the storage medium or cache of the logical unit to the I_T nexus.~~ ~~**Writes Exclusive**: A task that requests a transfer to the storage medium or cache of the logical unit from an initiator port that is not associated with a registered I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Persistent Reservation Holder**: The I_T nexus that delivered the PERSISTENT RESERVE OUT command with RESERVE, PREEMPT, or PREEMPT AND ABORT service action as identified by its registered reservation key.~~
~~6h~~	~~Exclusive Access – Registrants Only~~	~~**Reads Exclusive**: A task that requests a transfer from the storage medium or cache of the logical unit to an initiator port that is not associated with a registered I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Writes Exclusive**: A task that requests a transfer to the storage medium or cache of the logical unit from an initiator port that is not associated with a registered I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Persistent Reservation Holder**: The I_T nexus that delivered the PERSISTENT RESERVE OUT command with RESERVE, PREEMPT, or PREEMPT AND ABORT service action as identified by its registered reservation key.~~
~~7h~~	~~Write Exclusive – All Registrants~~	~~**Reads Shared**: Any application client on any I_T nexus may initiate tasks that request transfers from the storage medium or cache of the logical unit to the I_Tnexus.~~ ~~**Writes Exclusive**: A task that requests a transfer to the storage medium or cache of the logical unit from an initiator port that is not associated with a registered I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Persistent Reservation Holder**: Any registered I_T nexus as identified by a zero reservation key value.~~
~~8h~~	~~Exclusive Access – All Registrants~~	~~**Reads Exclusive**: A task that requests a transfer from the storage medium or cache of the logical unit to an initiator port that is not associated with a registered I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Writes Exclusive**: A task that requests a transfer to the storage medium or cache of the logical unit from an initiator port that is not associated with a registered I_T nexus shall be terminated with RESERVATION CONFLICT status.~~ ~~**Persistent Reservation Holder**: Any registered I_T nexus as identified by a zero reservation key value.~~
~~9h-Fh~~	~~Reserved~~	