

To: T10 Technical Committee
 From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
 Date: 26 June 2001
 Subject: SPC-3 Letting persistent reservations ignore target ports

Revision History

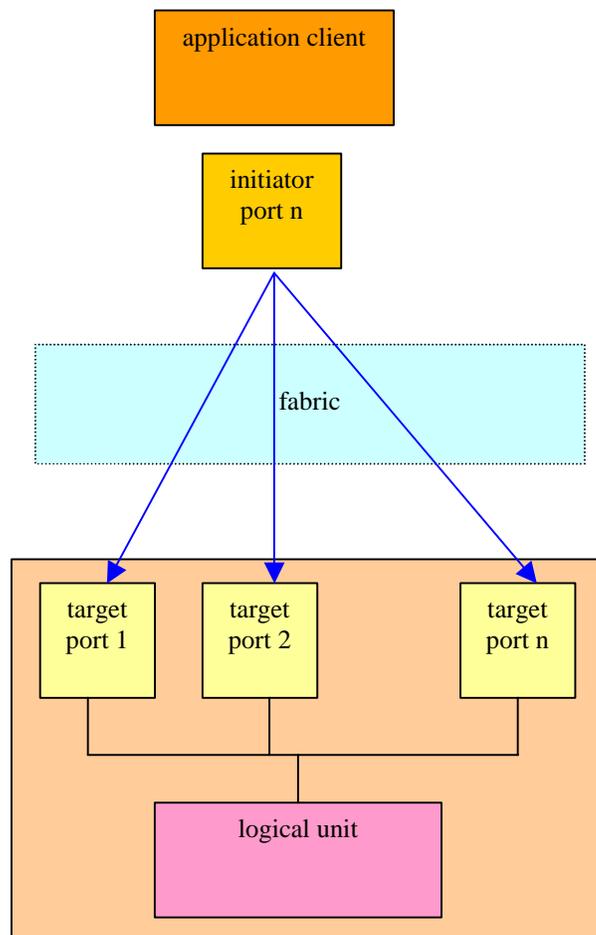
Revision 0 (5 March 2001) first revision. Not complete, but as presented in CAP working group.
 Revision 1 (17 April 2001). Removed mode page option; use a bit in the CDB. Removed target port group relationship. Add PR IN report capability.
 Revision 2 (26 June 2001). New approach: let target port be ignored if the initiator port name is world wide unique.

Related Documents

T10/spc2r19 – SCSI Primary Commands revision 19 (by Ralph Weber)
 T10/00-182r0 - SAM-2 device and port names (by Jim Hafner)

Overview

SPC-2 requires that logical units remember the target port through which a reservation was made in addition to the initiator port that originated the reservation. An application client wishing to make a reservation needs to run the PERSISTENT RESERVE command through each target port that can route commands to the logical unit.



In many cases, the logical unit doesn't care which target port the reservation came through. It just wants to distinguish between initiators. Requiring reservation commands for each port burdens the application client with issuing extra commands and burdens the logical unit with extra non-volatile storage.

The reason for this rule is that the target ports may be in different SCSI domains, so they cannot rely on the initiator port identifier alone to truly identify the initiator. If both domains were on parallel SCSI busses, for example, both initiators would likely appear as ID 7.

On many fabrics, however, the initiator port is identified not only by an initiator port identifier but also an initiator port name. The initiator port name may be required to be world wide unique. When this is true, the worry about different SCSI domains disappears.

This proposal clarifies when the logical unit uses the initiator port identifier and initiator port name in reservations and when the target port must be included. It adds a PR IN service action to report compliance with this new standard or behavior compatible with previous standards.

Suggested Changes

Text is from SPC-2 revision 19.

5.5.1 Reservations overview

Reservations may be used to allow a device server to execute commands from a selected set of initiators. The device server shall reject commands from initiators outside the selected set of initiators by uniquely identifying initiators using protocol specific mechanisms.

'''

5.5.3.1 Overview of the Persistent Reservations management method

...

The PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands provide the basic mechanism for dynamic contention resolution in multiple initiator systems using multiple port targets. Before a persistent reservation may be established, an initiator shall register 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 initiators that are registered;
- c) identification of the reservation key(s) that have an associated reservation;
- d) preemption of a persistent reservation from a failing or uncooperative initiator; and
- e) multiple initiators to participate in a reservation.

The reservation key provides a method for the application client to associate a protocol-independent identifier with an initiator [on a specific port of a device server](#). The reservation key is used in the PERSISTENT RESERVE IN command to identify which initiators are registered and which initiator, if any, holds the reservation. The reservation key is used in the PERSISTENT RESERVE OUT command; to register an initiator, to verify the initiator issuing the PERSISTENT RESERVATION OUT command is registered, and to specify which initiator's registration or persistent reservation to preempt.

Reservation key values may be used by application clients to identify initiators, using application specific methods that are outside the scope of this standard. This standard provides the ability to register no more than one key per initiator/logical unit pair. Multiple initiators may use the same key for a logical unit. An initiator may establish registrations for multiple logical units in a SCSI device using any combination of unique or duplicate keys. These rules provide the ability for an application client to preempt multiple initiators with a single PERSISTENT RESERVE OUT command, but they do not provide the ability for the application client to uniquely the initiators using the PERSISTENT RESERVE commands.

5.5.3.x Initiator identifying information

On protocols where the initiator port name is required and is world wide unique, the initiator identifying information consists of the initiator port name. Device servers on target ports connected to such protocols shall include the REPORT CAPABILITIES service action in PERSISTENT RESERVE IN with the TARGET PORT IGNORED bit set to one (see 7.2.10.x).

On protocols where the initiator port name is required but is not world wide unique, the initiator identifying information consists of the initiator port name and the relative port identifier of the target port.

On protocols where the initiator port name is not required, the initiator identifying information consists of the initiator port identifier and the relative port identifier of the target port.

5.5.3.2 Preserving persistent reservations

The application client may request activation of the persist through power loss device server capability to preserve the persistent reservation and registration keys across power cycles by setting the APTPL bit to one in PERSISTENT RESERVE OUT parameter data sent with a REGISTER, or a REGISTER AND IGNORE EXISTING KEY service action.

After the application client enables the persist through power loss capability the device server shall preserve all current and future registrations and persistent reservations associated with the logical unit to which the REGISTER or the REGISTER AND IGNORE EXISTING KEY service action was addressed until an application client disables the persist through power loss capability. The APTPL value from the most recent successfully completed REGISTER or REGISTER AND IGNORE EXISTING KEY service action from any application client shall determine the logical unit's behavior in the event of a power loss.

The device server shall preserve the following information for each registration across any reset, and if the persist through power loss capability is enabled, across any power cycle:

- a) ~~i~~Initiator ~~identifier~~identifying information (see 5.5.3.x); and
- b) reservation key; ~~and~~
- ~~e) when supported by the protocol, the initiator port's world wide identification.~~

The device server shall preserve the following reservation information across any reset, and if the persist through power loss capability is enabled, across any power cycle:

- a) ~~i~~Initiator ~~identifier~~identifying information (see 5.5.3.x);
- b) reservation key;
- c) scope; ~~and~~
- d) type; ~~and~~
- ~~e) when supported by the protocol, the initiator port's world wide identification.~~

~~For those protocols for which the initiator port's world wide identification is available to the device server the initiator port's world wide identification shall be used to determine if the initiator identifier has changed. This determination shall be made at any time the target detects that the configuration of the system may have changed. If the initiator identifier changed, the device server shall assign the new initiator identifier to the existing registration and reservation of the initiator port having the same world wide identification.~~

...

5.5.3.4 Registering

...

In response to a PERSISTENT RESERVE OUT with a REGISTER or a REGISTER AND IGNORE EXISTING KEY service action the device server shall perform a registration by doing the following as an uninterrupted series of actions:

- a) Process the registration request regardless of any persistent reservations;
- b) process the APTPL bit;
- c) ignore the contents of the SCOPE and TYPE fields;

- d) map the reservation key to the registering initiator using the initiator ~~identification and, if available, the initiator port's world-wide identification identifying information (see 5.5.3.x)~~;
- e) register the reservation key without changing any a persistent reservation that may exist; and
- f) retain the reservation key and associated information.

5.6 Multiple port and multiple initiator behavior

SAM-2 specifies the behavior of logical units being accessed by more than one initiator. Additional ~~service-deliverytarget~~ ports provide alternate ~~service-delivery~~ paths through which the device server may be reached and may also provide connectivity for additional initiators. An alternate path may be used to improve the availability of devices in the presence of certain types of failures and to improve the performance of devices whose other paths may be busy.

If a SCSI ~~target~~ device has more than one ~~service-deliverytarget~~ port, the arbitration and connection management among the ~~service-deliverytarget~~ ports is vendor specific. If one ~~service-deliverytarget~~ port is being used by an initiator, accesses attempted through other ~~service-deliverytarget~~ port(s) may:

- a) receive a status of BUSY; or
- b) be accepted as if the other ~~service-deliverytarget~~ port(s) were not in use.

The device server shall indicate the presence of multiple ~~target~~ ports by setting the MULTIP bit to 1 in its standard INQUIRY data.

For the purposes of handling reservations, other initiators are defined as all initiators ~~on the same service-delivery port except the initiator holding the reservation and all initiators on all other service-delivery ports that do not share the same initiator identifying information.~~

Only the following operations allow an initiator to interact with the tasks of another initiator, regardless of the service delivery port:

- a) the PERSISTENT RESERVE OUT with PREEMPT service action preempts persistent reservations for other initiators (see 5.5.3.6.3);
- b) the PERSISTENT RESERVE OUT with PREEMPT AND ABORT service action preempts persistent reservations and all tasks for other initiators (see 5.5.3.6.4);
- c) the PERSISTENT RESERVE OUT with CLEAR service action releases persistent reservations and removes reservation keys for all initiators (see 5.5.3.6.5);
- d) the TARGET RESET task management function releases reservations established by the reserve/release method and removes all tasks ~~for all initiators~~ for all logical units in the target ~~and for all initiators~~ (see SAM-2). Persistent reservations remain unmodified;
- e) the LOGICAL UNIT RESET task management function releases reservations established by the reserve/release method and removes all tasks for all initiators for the addressed logical unit and any logical units issuing from it in a hierarchical addressing structure (see SAM-2). Persistent reservations remain unmodified; and
- f) the CLEAR TASK SET task management function removes all tasks ~~for all initiators~~ for the selected logical unit ~~for all initiators~~. Most other logical unit states remain unmodified, including MODE SELECT parameters, reservations, and ACA (see SAM-2).

~~[Editor's note: just making d) e) and f) use the same order above]~~

7.10.2 PERSISTENT RESERVE IN service actions

7.10.2.1 Summary of PERSISTENT RESERVE IN service actions

The service action codes for the PERSISTENT RESERVE IN command are defined in table 67.

Table 67 — PERSISTENT RESERVE IN service action codes

Code	Name	Description
00h	READ KEYS	Reads all registered Reservation Keys
01h	READ RESERVATION	Reads the current persistent reservations
02h	REPORT CAPABILITIES	Returns capability information
02h-03h – 1Fh	Reserved	Reserved

7.10.2.x Report Capabilities

The REPORT CAPABILITIES service action requests that the device server return a parameter page indicating support for various persistent reservation features.

Table 73. REPORT CAPABILITIES parameter data

<u>Byte\Bit</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	<u>0</u>
<u>0</u>	<u>Rsvd</u>	<u>Rsvd</u>	<u>Rsvd</u>	<u>Rsvd</u>	<u>Rsvd</u>	<u>ELEMENT</u> <u>SCOPE</u>	<u>APTPL</u>	<u>TARGET</u> <u>PORT</u> <u>IGNORED</u>
<u>1</u>	<u>RESERVED</u>							
<u>2</u>	<u>RESERVED</u>							
<u>3</u>	<u>RESERVED</u>							
<u>4</u>	<u>RESERVED</u>							
<u>5</u>	<u>RESERVED</u>							
<u>6</u>	<u>RESERVED</u>							
<u>7</u>	<u>RESERVED</u>							

A TARGET PORT IGNORED bit of one indicates that the target port does not include the relative target port identifier in the initiator identifying information. A TARGET PORT IGNORED bit of zero indicates that the target port includes the relative target port identifier in the initiator identifying information.

[Editor's note: bits for the two existing optional features are also proposed. They will make it easier for software to confirm that these features are enabled. They don't alleviate the need for software to have other means to determine this, however, since they didn't exist before although the optional features did. They could easily be removed from this proposal.]

The APTPL bit indicates that the device server supports the APTPL in the PERSISTENT RESERVE OUT command.

The ELEMENT SCOPE bit indicates that the device server supports a SCOPE value of ELEMENT_SCOPE in the persistent reservation commands.