T10/97-246, Revision 1

Date: September 11, 1997

To: T10 Committee

From: Tom Coughlan
Digital Equipment Corporation
Mail Stop ZKO-3-4/U14
110 Spitbrook Road
Nashua, New Hampshire
Telephone: 603-884-0933
E-mail: tom.coughlan@zko.mts.dec.com

Subject: Persistent Reservations with Non-Unique Keys

The Problem

The current specification of Persistent Reservation does not define the device server's behavior when different initiators register with the same key.
The standard does not provide a method for the device server to reject non-unique keys, yet it implies that the Preempt and the Preempt and Clear service actions apply to just one initiator. Which initiator does the device server choose?

The Opportunity

There are cluster applications where it is desirable to manage reservations and preemption on a node basis. In this model, when the node enters the cluster the application registers all of the node's adapters with the same key, and establishes whatever reservations are required. When the node exits the cluster, one or more of the survivors issue a Preempt or a Preempt and Clear with the key that all the exiting adapters registered under. The device server is expected to apply the service action to all initiators registered with the specified key

Summary of the Change

Continue to allow different initiators to register with the same key. Require that the device server apply the Preempt and Preempt and Clear service actions to all initiators registered with the key specified in the command's parameter list.

Extracts from:

SCSI-3 Primary Commands Revision 11a (T10/995D revision 11a)

5.4 Multiple port and multiple initiator behavior

SAM specifies the behavior of logical units being accessed by more than one initiator. Additional 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 target has more than one service delivery port, the arbitration and connection management among the ports is defined by the implementation. Above the interconnect implementation, two contention resolution options exist:

a) If one port on a target is being used by an initiator, accesses attempted through another port may receive a status of BUSY; or
b) If the target has sufficient internal resources, commands may be accepted through other ports while one port is being used.

The device server shall indicate the presence of multiple ports by setting the MultiP bit to 1 in its standard INQUIRY data. Once a device server grants a reservation, all initiators (regardless of port) except the initiator to which the reservation was granted shall be treated as different initiators. 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 removes persistent reservations for ~~an~~other initiator<u>s</u> (see 7.13.1.5);
b) the PERSISTENT RESERVE OUT with Preempt and Clear service action removes persistent reservations and all tasks for ~~an~~other initiator<u>s</u> (see 7.13.1.6);
c) the PERSISTENT RESERVE OUT with Clear service action removes persistent reservations and reservation keys for all initiators (see 7.13.1.4);
d) the TARGET RESET task management function removes reservations established by the Reserve/Release method and removes all tasks for all logical units in the target and for all initiators (see SAM). Persistent reservations remain unmodified;
e) the LOGICAL UNIT RESET task management function removes reservations established by the Reserve/Release method and removes all tasks for all initiators for the addressed logical unit and any logical units depending from it in a hierarchical addressing structure (see SAM). Persistent reservations remain unmodified; and
f) the CLEAR TASK SET task management function removes all tasks for the selected logical unit for all initiators. Most other machine states remain unmodified, including MODE SELECT parameters, reservations, and ACA (see SAM).

7.12.1.1 Read Keys

The Read Keys service action requests that the device server return a parameter list containing a header and a complete list of all reservation keys currently registered with the device server. If multiple initiators have registered with the same key, then that key value shall be listed multiple times, once for each such registration. The keys may have been passed by a PERSISTENT RESERVE OUT command that has performed a Register service action. The relationship between a reservation key and the initiator or port is outside the scope of this standard.

7.12.2 PERSISTENT RESERVE IN parameter data for Read Keys

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the Read Keys service action is shown in table 37.

The Generation value is a 32-bit counter in the device server that shall be incremented every time a PERSISTENT RESERVE OUT command requests a Register, a Clear, a Preempt, or a Preempt and Clear operation. The counter shall not be incremented by a PERSISTENT RESERVE IN command, by a PERSISTENT RESERVE OUT command that performs a Reserve or Release service action, or by a PERSISTENT RESERVE OUT command that is not performed due to an error or reservation conflict. The Generation value shall be set to 0 as part of the power on reset process.

The Generation value allows the application client examining the generation value to verify that the configuration of the initiators attached to a logical unit has not been modified by another application client without the knowledge of the examining application client.

The Additional length field contains a count of the number of bytes in the Reservation key list. If the Allocation length specified by the PERSISTENT RESERVE IN command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed Allocation length shall be sent to the application client. The remaining bytes shall be truncated, although the Additional length field shall still contain the actual number of bytes in the reservation key list without consideration of any truncation resulting from an insufficient Allocation length. This shall not be considered an error.

The Reservation key list contains all the 8-byte reservation keys registered with the device server through PERSISTENT RESERVE OUT Reserve, Preempt, Preempt and Clear, or Register service actions. Each reservation key may be examined by the application client and correlated with a ~~particular~~ set of initiators and SCSI ports by mechanisms outside the scope of this standard.

7.12.3 PERSISTENT RESERVE IN parameter data for Read Reservations

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the Read Reservations service action is shown in table 38.

The Generation field shall be as defined for the PERSISTENT RESERVE IN Read Keys parameter data.

The Additional length field contains a count of the number of bytes in of Reservation descriptors. If the Allocation length specified by the PERSISTENT RESERVE IN command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed Allocation length shall be sent to the application client. The remaining bytes shall be truncated, although the Additional length field shall still contain the actual number of bytes of Reservation descriptors and shall not be affected by the truncation. This shall not be considered an error.

The format of a single read Reservation descriptor is defined in table 39. There shall be one read Reservation descriptor for each persistent reservation held on the logical unit by any initiator.

For each persistent reservation held on the logical unit, there shall be a read Reservation descriptor presented in the list of parameter data returned by the device server in response to the PERSISTENT RESERVE IN command with a Read Reservations action. The descriptor shall contain the Reservation Key under which the persistent reservation is held. The Type and Scope of the persistent reservation as present in the PERSISTENT RESERVE OUT command that created the persistent reservation shall be returned (see 7.12.3.1 and 7.12.3.2).

Reservation key is the registered reservation key under which the reservation is held~~.~~ ~~Using techniques that are outside the scope of this standard, .~~ If initiators use unique keys, then the application should be able to associate the reservation key with the initiator that holds the reservation. This association is done using techniques that are outside the scope of this standard.

If the Scope is an Extent reservation, the Scope-specific address field shall contain the LBA of the first block of the extent and the Extent length field shall contain the number of blocks in the extent. If the Scope is an Element reservation, the Scope-specific address field shall contain the Element address, zero filled in the most significant bytes to fit the field, and the Extent length field shall be set to zero. If the Scope is a Logical Unit reservation, both the Scope-specific address and Extent length fields shall be set to zero.

7.13.1.5 Preempt

The PERSISTENT RESERVE OUT command that successfully performs a Preempt service action shall remove all persistent reservations for ~~the~~ all initiators that are registered with the Service Action Reservation key specified ~~by~~ in the PERSISTENT RESERVE OUT parameter list. ~~The initiator is identified by the reservation key of the initiator to be preempted.~~ -Any commands from any initiator that have been accepted by the device server as nonconflicting shall continue normal execution.

A Unit Attention condition is established for the preempted initiators. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. Subsequent commands are subject to the persistent reservation restrictions established by the preempting initiator.

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PERSISTENT RESERVE OUT command and the corresponding fields in the PERSISTENT RESERVE OUT parameter list.

The registration key for the initiators that has~~s~~ve been preempted shall be reset to default value of zero by the Preempt service action.

A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Preempt service action that conflicts with any active persistent reservations except the preempted reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT is enabled for execution. The PERSISTENT RESERVE OUT command with a Preempt service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.

NOTE 27 For the simplest predictable behavior, the Preempt service action should be performed with the Ordered task attribute.

Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt or Preempt and Clear service action. New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The persistent reservation of a logical unit or the persistent reservation of extents having the same type value shall be permitted if no conflicting persistent reservations other than the reservations being preempted are held by another initiator.

7.13.1.6 Preempt and Clear

The PERSISTENT RESERVE OUT command performing a Preempt and Clear service action removes all persistent reservations for ~~the~~ all initiators that are registered with the Service Action Reservation key ~~-~~specified ~~by~~ in the PERSISTENT RESERVE OUT parameter list. ~~The initiator is identified by the reservation key of the initiator to be preempted.~~ Any commands from the initiators being preempted are each terminated as if an ABORT TASK task management function had been performed by the preempted initiator.

A Unit Attention condition is established for the preempted initiators. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. Subsequent new commands and retries of commands that timed out because they were cleared are subject to the persistent reservation restrictions established by the preempting initiator.

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PERSISTENT RESERVE OUT command and the corresponding fields in the PERSISTENT RESERVE OUT parameter list.

The Preempt and Clear service action shall clear any ACA condition associated with the initiator being preempted and shall clear any tasks with an ACA attribute from that initiator. ACA conditions for other initiators shall prevent the execution of the PERSISTENT RESERVE OUT task, which shall end with status of ACA ACTIVE.

NOTE 28 The Preempt and Clear service action will clear the ACA condition associated with the initiator being preempted eventhough the task is terminated with an ACA

ACTIVE status. Thus, the next command arriving at the device server will not encounter the ACA condition previously active for the initiator being preempted.

Any Asynchronous Event Reporting operations in progress that were initiated by the device server are not affected by the Preempt and Clear service action.

The reservation key registered for the initiators that has~~ve~~ been preempted shall be reset to the default value of zero by the Preempt and Clear service action.

7.13.2 PERSISTENT RESERVE OUT parameter list

The parameter list required to perform the PERSISTENT RESERVE OUT command are defined in table 45. All fields shall be sent on all PERSISTENT RESERVE OUT commands, even if the field is not required for the specified Service action and Scope values.

The Reservation key field contains an 8-byte token provided by the application client to the device server to identify the initiator that is the source of the PERSISTENT RESERVE OUT command. The device server shall verify that the Reservation key field in a PERSISTENT RESERVE OUT command matches the registered reservation key for the initiator from which the command was received. If a PERSISTENT RESERVE OUT command specifies a Reservation key field other than the reservation key registered for the initiator, the device server shall return a RESERVATION CONFLICT status. The reservation key of the initiator shall be valid for all Service action and Scope values.

The Service Action Reservation key field contains information needed for three service actions; the Register, Preempt, and Preempt and Clear service actions. For the Register service action, the Service Action Reservation key field contains the new reservation key to be registered. For the Preempt and Preempt and Clear service actions, the Service Action Reservation key field contains the reservation key of the persistent reservations that ~~is~~ are being preempted. For the Preempt and Preempt and Clear service actions, failure of the Service Action Reservation key to match any registered reservation keys shall result in the device server returning a RESERVATION CONFLICT status. The Service Action Reservation key is ignored for all service actions except those described in this paragraph.

If the Scope is an Extent reservation, the Scope-specific address field shall contain the LBA of the first block of the extent and the Extent length field shall contain the number of blocks in the extent. If the Scope is an Element reservation, the Scope-specific address field shall contain the Element address, zero filled in the most significant bytes to fit the field, and the Extent length field shall be set to zero. If the Service action is Register or Clear or if the Scope is a Logical Unit reservation, both the Scope-specific address and Extent length fields shall be set to zero.

The Activate Persist Through Power Loss (APTPL) bit shall be valid only for the Register service action. In all other cases, the APTPL shall be ignored. Support for an APTPL bit equal to one is optional. If a device server that does not support the APTPL bit value of one receives that value in a Register service action, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL

REQUEST and additional sense data shall be set to INVALID FIELD IN PARAMETER LIST.

If the last valid APTPL bit value received by the device server is zero, the loss of power in the target shall release all persistent reservations and set all reservation keys to their default value of zero. If the last valid APTPL bit value received by the device server is one, the logical unit shall retain all persistent reservations and all reservation keys for all initiators even if power is lost and later returned. The most recently received valid APTPL value from any initiator shall govern logical unit's behavior in the event of power loss.

Table 46 summarizes which fields are set by the application client and interpreted by the device server for each Service action and Scope value. Two PERSISTENT RESERVE OUT parameters are not summarized in table 46; Reservation key and APTPL.