Date: May 5, 2003

To: T10 Committee (SCSI)

From: Roger Cummings (VERITAS)

Subject: Two Persistent Reservations problems - updates & decision required

1.0 INTRODUCTION

In November 2002 I presented (on behalf of VERITAS) two problems with the Persistent Reservations definitions that existed in SPC-3 that we felt needed to be addressed in order to be able to migrate applications and operating systems away from supporting (original) Reserve and Release and towards Persistent Reservations. These two problems were:

1) The need to be able to mix usage of reserve/release and persistent reservations in the same system & storage network when dealing with tapes;

2) The need for an analog to the third-party function supported by original reservations in Persistent Reservations.

A reprise of these problems is given below. The descriptions and proposed solutions have been enhanced as a result of input from the November CAP Working Group and input from other people and companies who commented on the first version of this document.

2.0 TWO PROBLEMS

2.1 Mixing Reservations & Persistent Reservations with Tapes

2.1.1 Current Usage

A number of server platforms issue a Reserve command to a serial access device as a matter of course when opening a Volume. The Reserve is issued by the operating system itself, or by the tape class driver, independently of any application. In some, but not all, cases this behavior is configurable, but there is considerable resistance from users to changing the configuration, and it cannot in any event be done on a per-application basis.

Clearly, this behavior makes use of persistent Reservations with tape drives impractical. If an application issues a Persistent Reserve to the Tape LUN before opening a Volume, then the open will fail. If the application attempts to issue a Persistent Reserve after the platform has issued a Reserve, a Reservation Conflict will be returned, but this could be a result of another platform having an Exclusive access Persistent Reserve or this same platform having issued a Reserve.

The VERITAS Data Protection applications are required to operate in a heterogeneous configurations containing mature server platforms which are no longer being developed, but which exhibit the above behavior. Until those platforms are removed from the supported list, therefore, there is little prospect of our applications supporting Persistent Reservations with tape devices, unless the rules for mixing Reservations and Persistent Reservations are changed.

2.1.2 Suggested Usage

Any change to the definition of how Reservations & Persistent Reservations interact will inevitably increase the complexity of those rules from the “a Reservations cause Persistent Reservations to be rejected, even one Registration causes Reservations to be rejected” situation that exists today. Backwards compatibility problems will exist with implementations of Persistent Reservations which conform to the current SPC-2 definition.

To solve the problem outlined above, however, there will have to be support for some mixing of Reservations and Persistent Reservations within the same SCSI Domain. One change which would achieve this effect is suggested below.

2.1.3 New features

The following new responses are proposed as being required of a Logical Unit which is aware of Persistent Reservations but which receives a (original) Reserve or Release.

a) The receipt of a Reserve by a Logical Unit which supports Persistent Reservations shall cause the synthesis in that LU of a Registration for that Initiator Port (if no registration exists) with a either defined key value (suggested FFFFFFFF) or a specific flag set, and an Exclusive Access Reservation if no reservation already exists. If a previous reservation does exist, then if the reservation type is that the Initiator Port is allowed access, a registration is synthesized if necessary and the Reserve succeeds, otherwise a Reservation Conflict is generated. The synthesized persistent reservation will be subject to all of the requirements of a persistent Reservation Exclusive Access reservation as defined in SPC-2.

b) The receipt of a Release by a Logical Unit with an existing synthesized reservation shall be treated as the equivalent of a Register with a zero key value, and both the Registration and Reservation will be removed. The receipt of a Release by a Logical Unit with an existing Persistent Reservation which was not synthesized from a Reserve shall be succeed, but the persistent Reservation will remain, and no registrations will be altered.

Third-party Reservations and Superseding reservations shall be rejected by a Logical Unit which supports Persistent Reservations with a new ASC/ASCQ value of “Persistent Reservations Supported”.

2.2 Third Party Persistent Reservations

2.2.1 Current Usage

The definitions of RESERVE (10) and RELEASE (10) in SPC-2 include the concepts of Third-Party Reserve and Release and Superseding Reservations. These concepts were specifically created for use in the situation where multiple Initiators and a ‘copy engine” which implements the EXTENDED COPY command are present in a system, and they are used extensively in that situation by the VERITAS Data Protection products.

The specific usage of these concepts is as follows. Note that this same sequence will be performed with both Logical Units (source and target) which the copy engine will access, but only one Logical Unit is shown for simplicity

1) Host A sends a Reserve to Logical Unit 1.

2) Host A accesses and verifies the data on Logical Unit 1 (for a serial access Logical Unit some media loading & positioning commands may also be necessary).

3) Host A verifies that Copy Engine C has no outstanding operations.

4) Host A sends a third-party (superseding) Reserve to Logical Unit 1 containing the address of

Copy Engine C.

5) When the Reserve succeeds, Host A sends one or more Extended Copy commands to Copy Engine C which cause it to access Logical Unit 1.

6) When the Extended Copy commands are complete, Host A sends a non third-party (superseding) Reserve to Logical Unit 1.

7) Host A accesses Logical Unit 1 if necessary (for a serial access Logical Unit some media positioning and media unloading commands may also be necessary).

8) Host A sends a Release to Logical Unit 1.

There are three key aspects to the above sequence:

a) At no time is more than one Initiator Port able to access the Logical Unit;

b) Once step 1) is complete, there is no opportunity for another host or copy engine to interject in the process until step 8 is completed;

c) The Copy Engine does not need to be aware of the Reservation status at all.

None of these three aspects are directly supported by the present definition of Persistent Reservations.

a) A group reservation would allow both Host A and Copy Engine C to access the Logical Unit at the same time (and any other Initiator could register and also gain access);

b) Another Initiator could also issue a Preempt and gain access between steps 1 & 8;

c) The Copy Engine is required to Register with the Logical Unit before any type of Persistent Reservation access can be granted.

2.1.2 Suggested Persistent Reservations Usage

The equivalent sequence using Persistent Reservations would need to be something similar to:

1) Host A Sends a Register and Ignore Key command followed by a an Exclusive Access Reserve to Logical Unit 1.

2) Host A accesses and verifies the data on Logical Unit 1 (for a serial access Logical Unit some media loading & positioning commands may also be necessary).

3) Host A verifies that Copy Engine C has no outstanding operations, and sends an Extended Copy command which instructs Copy Engine C to Register with Logical Unit 1. (This needs to be a new type of Registration - see below).

4) Host A sends a (new) Service Action to Logical Unit 1 which Transfers the Exclusive Access reservation to the key value registered earlier by Copy Engine C, but does not remove Host A’s registration.

5) When the Transfer succeeds, Host A sends one or more Extended Copy commands to Copy Engine C which cause it to access Logical Unit 1.

6) When the Extended Copy commands are complete, Host A sends (new) Service Action to Logical Unit 1 which Transfers the Exclusive Access reservation back to its registered key value

(and this service action will only be accepted from the same Initiator port that issued the transfer in step 4). It also sends an Extended Copy command which instructs Copy Engine C to Register with Logical Unit 1 with a key of zero (unless the new type of Register in step 3 was used).

7) Host A accesses Logical Unit 1 if necessary (for a serial access Logical Unit some media positioning and media unloading commands may also be necessary).

8) Host A sends a Release to Logical Unit 1.

Note that the above sequence still requires the Copy Engine to create a Registration with the Logical Unit, and so is still not completely equivalent to the equences used with Third-party Reservations.

2.1.3 New Features Required

The above sequence requires two new features to be defined for Persistent Reservations, and suggested definitions are:

a) A new type of Registration which can be made by the copy engine but which only confers access when specifically enabled (or preempted to) by the Host. This is very important to ensure that the copy engine is prevented from accessing any devices in a situation where its controlling Host is preempted by another Host which then establishes a Registrants Only or All Registrants reservation type. There is an race condition in the way that the Extended Copy command operates which must be protected against at the Logical Unit.

The original VERITAS proposal a new registration type which does not participate in Registrants Only or All Registrants reservations. The November CAP meeting suggested instead a registration which is deleted in the event of a Preempt by another Initiator. VERITAS still prefers the former, on the basis that it believes it may be more useful for future functionality, but can live with the latter.

b) 2) A new Persistent Reserve Out Service Action which can Transfer a Write-Exclusive or Exclusive Access reservation between two registered Initiator Ports without any possibility of another Initiator port being able to influence the reservation.

3.0 Status

Since these proposals were first presented in November 2002 they have been discussed with a number of people in the industry. The problems postulated above have not been disproved, but a number of additional points that merited consideration have been identified and corresponding text included above. No solutions which are easier to implement or markedly different to those outlined above have been suggested, but the author feels that some incremental improvements may still be possible, and some specific cases will require more clarity in the final wording.

4.0 Decisions

While the two proposals presented here have some common threads, their impact on the definitions in SPC could hardly be more different. “Third Party Persistent Reservations“ is a topic which can be addressed as part of SPC-4, but “Mixing Reservations & Persistent Reservations with Tapes” will require both an amendment to SPC-2 and changing the functionality of existing implementations.

There is no sense in proceeding with the former proposal if the latter is unacceptable because in that situation the existing Third-party Reservations will continue to be used for tapes for the forseeable future. The author would therefore like to discuss the latter proposal at the May CAP meeting, and

specifically if he should be preparing a Project Proposal for an SPC-2 amendment for consideration by CAP & T10 in July 2003.