SCSI-3 Exception Recovery

Charles Monia
Digital Equipment Corporation
March 11, 1996
X3T10/95-352R2
In a non-interlocked bus, the potential for out-of-order command execution exists whenever:

- More than one command may be in transit at any time.
- A command completes with an exception status.
- The exception condition is removed before (or upon) the arrival of subsequent in-transit commands.

Examples:

- A Contingent Allegiance conditions occurs (ala SCSI-2 CA),
  - Subsequent command in transit is executed (automatically clearing the CA).
- A command terminates with BUSY status.
  - The BUSY condition may be cleared spontaneously.
  - Subsequent commands in transit may be executed.
Goals
- Prevent a command exception from causing out-of-order command execution,
- Must be backwards compatible with current ACA mechanism,
- Compatible with existing SCSI-2 device drivers,
  - If driver is written to be independent of the interconnect.
- Recovery from all 'non ACA' exceptions is hidden in protocol layer and compatible with SCSI-2 (as seen by the application client).
- Recovery from Contingent Allegiance is compatible with SCSI-2 (as seen by the application client).

Proposal:
- Extend ACA mechanism to handle all exceptions.
What's Different?

■ A command exception exists whenever a command completes with one of the following statuses:
  - ACA or CA conditions
    • CHECK CONDITION
    • COMMAND TERMINATED
  - Command Fault conditions
    • BUSY
    • RESERVATION CONFLICT
    • EXCEPTION ACTIVE (aka "ACA ACTIVE")
    • QUEUE FULL

■ Except for sense data, all exceptions have the same basic behavior as ACA.

■ The faulting initiator must explicitly remove all exception conditions using CLEAR EXCEPTION (was CLEAR ACA).

■ The above functions are mandatory for all non-interlocked protocol implementations.
Exception Handling

When an exception condition occurs the target:
- Blocks further commands from entering the task set from any initiator.
- Blocks the return of further statuses from the logical unit to all initiators.
- The condition must be cleared by the faulted initiator (via a CLEAR EXCEPTION request).

When the initiator's protocol layer detects the exception it:
- Stops sending further commands to the logical unit.
- Flushes all in-transit commands if CA or command fault.
- Marks all flushed commands for resend.
- Passes the exception status to the application client.
- Performs recovery based on exception type.
• Assumptions
  – Initiator’s protocol layer retains lists of pending commands in the order they were sent by the application clients.
    ▶ List of commands waiting to be sent.
    ▶ List of commands that have been sent.
  – A command is removed from the appropriate list:
    ▶ When status is received,
    ▶ When the command is aborted by the application client.
- The initiator’s protocol layer, on receiving the exception status:
  - Passes the exception status to the application client.
  - Stops sending commands to the faulted LUN.
  - Flushes in-transit commands.
    - Procedure is protocol-specific.
  - Marks flushed commands for resend.
  - Sends CLEAR EXCEPTION request
- Resends marked commands when requested by application client.
The initiator's protocol layer, on receiving the exception status:
- Passes the exception status to the application client.
- Stops sending commands to the faulted LUN.
- Flushes in-transit commands.
  - Procedure is protocol-specific.
- Marks flushed commands for resend.
- Next command sent with ACA attribute
- Sends CLEAR EXCEPTION request
- Resends marked commands when requested by application client.
SCSI-2 Compatibility

■ What the Application client sees:
  ▶ Behavior is identical to parallel SCSI.
  ▶ Command Queue frozen as for CAM.
    ▶ No out-of-order command execution.
  ▶ Contingent Allegiance
    ▶ Sense data automatically preserved.
  ▶ Application Client interface for manipulating
    the queue of unsent SCSI commands
    ▶ Implementation-specific but protocol
      independent.

■ What the application client doesn't see:
  ▶ CLEAR EXCEPTION sent by initiator’s
    Protocol layer
  ▶ Management of 'pending command list' by
    protocol layer.
The initiator’s protocol layer, on receiving the exception status:
- Passes the exception status to the application client.
- Stops sending commands to the faulted LUN.
- Does not flush in-transit commands.
- Passes all EXCEPTION ACTIVE statuses to application clients.
- No marked commands to resend.
- Application client sends one or more ACA commands followed by CLEAR ACA request.
- ACA commands do not automatically clear the exception.