Quantum

EXTERNAL MEMO

DATE: October 14, 1991

TO: X3T9.2 Members

FROM: Jim McGrath (408-894-4504)

SUBJ.: Command Queuing Model

Attached is the presentation covering the command queuing model Quantum is proposing for SCSI 3. The concept is to restrict our definition of queuing to only encompass elements that are visible across the SCSI interface (e.g., SCSI bus phase changes). Different levels of functionality, distinguished by their ease of host implementation, can then be defined.

The concept of data invariance is also considered in this proposal. It is not enough to specify that the state of the disk data always end up being the same as that arrived at by a strict FIFO execution of commands in a non-queued system. The state of the disk data must also be examined with respect to events preventing the execution of all commands (e.g. power failures). It must also be true that any sequence of command executions result in the user receiving the same set of data - even in the event of a power failure.

When invariance is sacrificed, it is up to the system to prevent integrity problems. Note that this is a perfectly reasonable imposition of the system in many cases. Our goal should be to identify, not to outlaw, those cases where an integrity problem could occur.

Although a lot of thought has gone into this model, I actively encourage comments. I would particularly invite people responsible for driver design to comment on how this model reflects the complexity of their work.

Tagged Command Queuing

James McGrath
Systems Engineer
408-894-4504

October 14, 1991

Quantum

Levels of Implementation

1 FIFO

2 Streamed

3 Reordered

Level Imposed Limits on Command Reordering

- **Reads**
  - **Amount of Buffered User Data**
- **Writes**
  - **Amount of Buffered User Data**
  - **Data Integrity**
  - **Ability to Prestage User Data into Buffer**
- **Ability to Prestage User Data for Writing is the Only Limit that is a Function of the Level of Queuing Implementation**

FIFO Example

Command	LBA
Read	100
Write	10
Write	11

FIFO

- Can Overlap Most Drive and Host Overhead with Command Execution
- Sequential Throughput
  - No Increase for Reads or Writes
- Random Throughput
  - Small Increase for Writes (same as for write caching)
  - Small Increase for Reads
  - Reads can be Reordered, Although Data Must be Transferred in Order

Streamed Example

Command	LBA
Read	100
Write	10
Write	11

Streamed

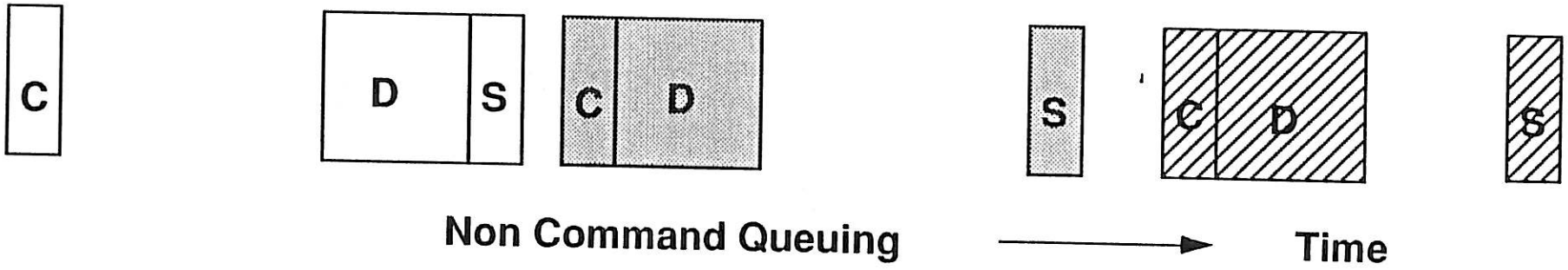
- Can Stream Data to the Drive
- Read Streaming (ala Prefetching)
- Write Streaming (ala Write Caching)
- Unlike Write Caching, Status is not Returned until Data has been Written to the Disk

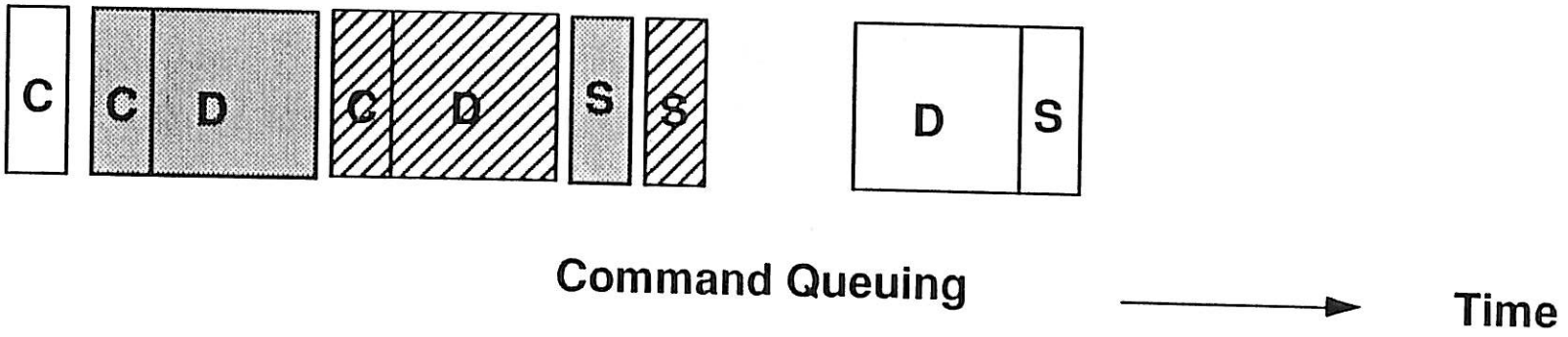
Reordered

- Command Can Complete Execution Before a Previous Command is Finished
- Command Status Returned in Any Order
- Host Manages Multiple I/O Contexts
- Queue Depth > 8

LC77

Reordered Example





Quantum

Open Issues

- **Sending Back QUEUE FULL status without violating the restrictions of Level 1 and Level 2.**
  - o **Treat QUEUE FULL as a special status.**
  - o **Take in any number of commands, and delay returning status until the proper time.**

- **Handling early CHECK CONDITION status generated by SCSI bus errors (e.g. parity errors).**
  - o **Treat this as a special situation.**
  - o **Continue execution, and delay returning status until the proper time.**

Degree Distinctions

- State of the Data Stored on the Disk
- State of Data Retrieved from the Host
- Ability to Tolerate Interruptions in Command Execution (e.g., Power Failures)

FIFO

- Writes Cannot be Reordered with Respect to other Writes
- Reads Can be Reordered with Respect to other Reads
- Reads and Writes Can be Reordered with Respect to Each Other as Long as their LBA ranges do not Intersect

File

- When the drive has become idle, the state of the data on the disk is exactly the same as if the commands were executed in FIFO order without command queuing.
- All of the data returned to the user is exactly the same data that would have been returned if the commands were executed in FIFO order without command queuing.
- The first condition is met even if command execution is interrupted at any time.

File

- No Current Distinction Between the File and Transaction Degrees
- Data Integrity Problems Can be Resolved by Insuring that Command Execution is Not Interrupted (e.g., a UPS).
- Equivalent to Write Reordering Implementations of Write Caching

Transaction

- Writes Can be Reordered with Respect to other Writes
- Reads Can be Reordered with Respect to other Reads
- Reads and Writes Can be Reordered with Respect to Each Other as Long as their LBA ranges do not Intersect
- If a Write is Followed by a Read of the Same LBA, then the Read May Be Executed Before the Write, Provided it Returns the Data Associated with the Write Command