

ICSC ITWG Meeting Minutes
5/17/2005

5/17/05 ICSC ITWG Meeting Minutes

Taking minutes: HP (FW)

Y HP Jim Hamrick (JH)
Y HP Jay Rosser (JR)
Y HP Fred Worley (FW)
Y IBM Fredy Neeser (FN)
N NetApp
N Sun

cascading ascii art attendance diagram. (if you have more than 1 minus visible, you are not eligible to vote.)

	hp	ibm	netapp	sun
	----	-----	-----	----
m-3	+	+	-	-
m-2	+	+	-	-
m-1	+	+	-	-
m-0	+	+	-	-

Next Meeting:

Tuesday **5/24/05**, 8am-10am PDT

ACTION summary:

PENDING AIs:

1. AI (FN) – Ask Martin/Cathy whether possible to make hyperlinks blue in PDF documents.
2. AI (FN): Report back to ITWG on semantics of IOVA output in IB's Register Physical Memory verb.
3. AI (All) – consider whether we wish to rename `it_lmr_sync_rdma_read` and `it_lmr_sync_rdma_write` to more intuitive names and then deprecate the existing calls.

NEW AIs:

4. AI (JH): Add text to the usage section of all posting calls to clarify appropriate usage to avoid WQ and CQ overflow
5. AI (JR): Determine how DTO 6 high level requirement vote was resolved
6. AI (All): Consider if the IT-API documentation should elaborate on async errors pertaining to the data source in all work request man pages

- Agenda bashing, approve minutes
 - o No minutes for 3. May (meeting was cancelled).

ICSC ITWG Meeting Minutes
5/17/2005

- Ballot on subject "IB Atomics detailed requirements", email sent by Jay Rosser on 05/12/2005.
 - o HP, IBM voted to approve on reflector
 - o **Ballot APPROVED**

- Posting a list of Work Requests
 - o Motivation for FN proposal:
 - § See email from FN, Subject: Re: Detailed requirements for posting a list of Work Requests, 4/27/05
 - § Error semantics: if there is an error it is difficult to describe what happened with the current interface
 - § Would be preferable to have a method to see how many work requests are in the cache
 - Prefer not to blur the difference between cached and processed work requests
 - § Could provide a parameter for "number of work requests cached"
 - Work requests posted using `it_post` command but have not been converted to WQEs yet
 - § Currently provides # of work requests *processed*
 - § Suggestion to provide # of work requests *cached*
 - § Note that either number can be derived from the other given that the Consumer tracks the total number of work requests posted
 - § Note that # of work requests processed grows without bound and can wrap
 - Number of work requests cached is a small number bound by the size of the DTO
 - Managing the number of work requests posted could be difficult for the consumer, particularly to handle the wrap case
 - o How does the consumer size its work queue?
 - § There is both a cache and an internal work queue
 - § How does consumer avoid Too Many Posts error?
 - Imp has freedom to move items out of the cache and into the work queue at any given time
 - Consumer should be limited by the actual size of the work queue (can not assume any given number of requests to be left in the cache at any given time)
 - o Immediate error that is returned by a post call can refer to a different call (that of a previously cached request) – this could be confusing to a consumer
 - § Suggest that if error due to a cached work request, return a specific immediate error – `IT_CACHED_WORK_REQUEST_ERROR`
 - § Additional OUT parameter to identify the precise cause of the error
 - Struct with 2 fields: specific error value, # of cached requests
 - Goal to remain backward compatible with `it_status_t`
 - Consumer could pre-allocate
 - Consumers that are 2.0 compliant don't need this value

ICSC ITWG Meeting Minutes

5/17/2005

- Struct would always be filled in by the implementation (provided that a valid structure pointer is passed in to the call)
- § JH: would force implementation to do additional bookkeeping that is required only to provide more consumer-friendly error information
 - Proposal above is sufficiently low overhead
- § Remaining work requests would be flushed on error
- § In the case of an error, what should number of work requests refer to?
 - Number of work requests cached is # of work requests posted but not yet converted to WQEs
 - This value would include the failing work request
- o Clarification of purpose for proposal
 - § Can't you derive this information without the special code?
 - § Yes, possible, but somewhat confusing within the documentation
 - § Could have it_post_send call return an error code that only pertains to an it_lrm_link call
 - § Proposal provides more clarity for the consumer that the error provided is for a different call and that extra work needs to be done by the consumer (using the # of cached work requests and the consumer's list of outstanding requests (consumers responsibility to create and maintain))
- o What does it mean to overflow a queue?
 - § Queue consists of a cache and an implementation internal work queue
 - § How does consumer know how "overflow" is defined?
- o Immediate errors that can be non-catastrophic
 - § Too Many Posts error – why would this need to be catastrophic?
 - Implementation of recovery would be tricky
 - Would need to have a way to hold back later work items posted but not processed to maintain ordering until resources are available
 - § Natural programming model to post until you get "too many posts" error – some consumers may want to do this
 - § Agree it is possible to construct an implementation that works properly and handles the too many posts error
 - If you care about post order, you have to serialize part of the pat in your consumer code that does the posting
 - If you have do that serialization, same amount of work for the consumer to do this themselves
 - There may be ULPs that don't care about the order...
- o What to do on error
 - § Return too many posts error
 - § Leave it in the cache
 - To do this, must worry about the size of the cache vs the size of the endpoint work queue
 - § What are implications of the coalesce flag?
 - Return Too Many Posts when you post some amount more than the WQ size without having reaped a completion

ICSC ITWG Meeting Minutes

5/17/2005

- Would that correspond where you post with the new model and the coalesce flag being clear?
 - § Yes, equiv to current posting semantics
 - § Would like to maintain that model with the coal. Flag on
- “Too Many Posts” error should depend only on # of WQs posted and # of completions reaped
 - Some fuzziness here – can guarantee with programming model that you will not get this error, but can not programmatically guarantee that you will get this error
- Cache size
 - § Sizing cache to be same size as the internal work queue
 - If you have previously posted requests with the coalesce flag set, still could have work requests in the cache
 - Would you then get a Too Many Posts error (if you keep posting with the coalesce flag begin set)?
- Review of current semantics:
 - § Suppose WQ with send queue depth of 2
 - § CQ with depth of 2 also implied
 - Consumer could create CQ with size larger than WQ to enable WQ overflow without CQ overflow
 - § Can post 4 WQs and not necessarily get an error
 - § Would get an error on the 5th post for sure
 - § Some implementations could get an error on the 3rd post
 - § Note that posting more than the WQ size could result in overflow of the CQ
 - CQ overflow can be a catastrophic failure on iWARP
 - Consumer should ensure that they do not overflow the CQ (either by not overflowing the WQ or by sizing the CQ larger than the WQ)
 - § **AI (JH): Add text to the usage section of all posting calls to clarify appropriate usage to avoid WQ and CQ overflow**
 - § If Consumer does not have more outstanding work requests than the size of the internal Send Queue (as returned by it_ep_query) then the Consumer will never overflow a queue
- Why does the consumer need to know that the cache exists?
 - § Because of error semantics for immediate errors on cached work requests
- Further discussion of outstanding request semantic
 - § First post could go to the cache
 - § Consumer could count elements in the cache (not yet converted to Work Requests) as outstanding
 - § Post n work requests with the coalesce flag set
 - n “outstanding”
 - Could move work requests to work queue at any time

ICSC ITWG Meeting Minutes

5/17/2005

- Next time you invoke the post call, possible that none of the requests have moved to the work queue
 - Next post call will fail
- Number of outstanding events will not decrement until completion is reaped
- Implementation can convert posted requests to WQEs whenever it chooses to do so, asynchronously
 - Should not frustrate the conservative programmer
 - May frustrate the aggressive programmer who is trying to use the cache as additional WQE storage
- What does “number of work requests posted” mean?
 - § Does “post” refer to calling `it_post_sent` call or some internal operation corresponding to the post verb?
 - § JH: Number of times the post call (a posting call) has been invoked
- Proposal Summary: Provide additional detail on immediate error for cached work requests
 - § Posting calls may return a new error value:
`IT_CACHED_WORK_REQUEST_ERROR`
 - § Posting calls extended to take an additional OUT parameter (name TBD)
 - struct containing
 - specific error value for previously posted WR that has failed
 - current number of cached work requests
 - § Allows consumer to determine which specific request has failed, assuming that consumer tracks outstanding work requests
 - Consumer to allocate the struct
 - § Proposal discussed; issue still open
- DTO
 - § Size of the work queue determines the lower bound on the # of work requests that may be posted to the work queue
 - § See 6.1.1 and 6.1.2
 - § AI (JR): Determine how DTO 6 high level requirement vote was resolved
- Reason
 - § Suppose consumer posts WQs in groups
 - § Bind, send, unbind
 - § Now work of 3 WRs fails
 - Was it the bind, send or unbind that failed?
- How should the consumer track the outstanding work requests?
 - § Need to provide guidance in the programmers guide
 - § JR: Query the endpoint for `MAX_REQUEST_DTO` or `MAX_RECEIVE_DTO`

ICSC ITWG Meeting Minutes
5/17/2005

- Available space in the send queue = MAX_REQUEST_DTO minus the number of send queue post calls plus the number of events reaped for that send queue
- Email thread on subject "Erratum GN213 (IBM201): RDMA Read Failures on iWARP" started by Fredy Neeser on May 11, 2005.
 - o Do we have consensus on RDMA Read error handling?
 - § FN: Difficult for IHVs to provide support for proposed RDMA Read error semantics
 - Expect that not all RNIC implementations will have sufficient support
 - Not appropriate to impose use of particular hardware implementation within the API definition
 - Desire to support advanced RDMA Read error handling without requiring it
 - § If a DDP Write results in an access violation, what should DDP do?
 - DDP would not know if a failing RDMA Write is an RDMA Write or an RDMA Read Response
 - Spec says should (not MUST) tell RDMAP precisely what kind of request it was
 - § Consensus that this summary looks correct
 - o Access violations
 - § Only talk about errors related to the data sync; not about errors related to the data source
 - Some notes, but not at the same level of detail
 - Some text now in dto_status man page
 - § Should there be a more detailed description of local errors (than exists on the dto_status page) to provide consistent style for errors pertaining to data source and data sync
 - § Note that care must be taken to keep dto_status and post man pages in sync
 - § **AI (All): Consider if the IT-API documentation should elaborate on async errors pertaining to the data source in all work request man pages**
 - o Advanced RDMA Read error handling support: Decide if we want to request that RNIC-PI tells consumers if such support is available.
 - § If IHV provides advanced features, how does the Consumer get to know that the feature is available?
 - § Agreed that advanced error reporting is useful
 - § Not clear that the knowledge of IHV ability to support / not support advanced error handling would require any change to the programming model used by the Consumer
 - § If not, no need to provide attribute at IT-API level
 - § May not need to provide attribute at RNIC-PI level either
 - § Agreed that IT-API should request of RNIC-PI optional support for advanced error handling

ICSC ITWG Meeting Minutes
5/17/2005

- Handling of errata against v2.0: Which version of IT-API will resolve them?
 - o Classes of changes to the document:
 - § Updating hyperlinks
 - Editorial change; shouldn't be any issue with fixing these now
 - § Fixing errata
 - Semantic change; could impact current consumers
 - o Should we use v2.1 just to resolve errata against v2.0 and use a later version for new functionality?
 - § Could release new version (with new minor version number) to fix errata (including semantic changes) and update the hyper links
 - § Could add new functionality in a later document with a different new version number
 - § Potential alternative: release corrigenda
 - Release separate document correcting semantic changes (without release of full document)
 - § Potential alternative: release updated

- Action item review
 - AI (JR): Put IB Atomics requirements to vote.
 - o Closed

 - AI (All) – research whether Terminate message can be correlated to pending RDMA Read.
 - o Closed

 - AI (FN) – Ask Martin/Cathy whether possible to make hyperlinks blue in PDF documents.
 - o Determined that this can be done
 - o Pending to determine how to address

 - AI (FN): Revisit it_lmr_create to determine if an address space qualifier is necessary.
 - o No sufficiently compelling argument in favor of space qualifier on it_lmr_create determined to date
 - o Closed, pending new arguments

 - AI (FW) – pursue discussion of RNIC failover and host memory movement w.r.t. memory registration.
 - o No sufficiently compelling argument from this source in favor of space qualifier on it_lmr_create determined to date
 - o Closed, pending new arguments

ICSC ITWG Meeting Minutes
5/17/2005

- AI (FN): Report back to ITWG on semantics of IOVA output in IB's Register Physical Memory verb.
 - o Pending, being discussed in RNIC-PI WG

- AI (ITWG) – deprecate “out of resources” error code from the Bind calls in IT-API 2.1 and add Bind to list of routines valid to call from a callback.
 - o Added requirement to MM requirements to capture this issue
 - o Closed

- AI (JR,FW) – determine if option Two (supporting non-aligned relative addressing for Atomics) is truly necessary and return a proposal.
 - o Added warning to Atomics requirement
 - o Closed

- AI (JR) – add requirement for Atomics stating that if the Consumer wants to use non-coherent memory at the Atomic target and wishes to access the Atomic target operand via CPU rather than HCA, then they must use the sync operations.
 - o Closed

- AI (All) – consider whether we wish to rename `it_lmr_sync_rdma_read` and `it_lmr_sync_rdma_write` to more intuitive names and then deprecate the existing calls.
 - o Pending

- AI (JR) – find the original note on the subject of posting API to ITWG page and follow up with Martin.
 - o API posted to ITWG page
 - o Closed

- Any other business

Meeting adjourned 10:24am PDT