

ICSC ITWG Meeting Minutes
5/24/2005

5/24/05 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 5/31/05, 8am-10am PDT

ACTION summary:

PENDING AIs:

1. AI (FN): Report back to ITWG on semantics of IOVA output in IB's Register Physical Memory verb.
2. 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.
3. AI (JH): Add text to the usage section of all posting calls to clarify appropriate usage to avoid WQ and CQ overflow
4. AI (JR): Determine how DTO 6 high level requirement vote was resolved
5. AI (All): Consider if the IT-API documentation should elaborate on async errors pertaining to the data source in all work request man pages

NEW AIs:

6. AI (JH): Provide additional description of what is meant in list of work requests proposal by # of work requests processed; include usage example
7. AI (JH): Research waiting on an fd attached to an EVD in current ITAPI
8. AI (JH): Propose requirements such that when a callback routine is attached to an EVD, that is the only notification pathway that functions
9. AI (FN): Propose resolution for ambiguity of shared state of LMRs for LMRs that are a) created in a shared state and b) transition into a shared state

- Agenda bashing, approve minutes

ICSC ITWG Meeting Minutes

5/24/2005

- o Email from Jay Rosser, subject "ICSC ITWG draft minutes for May 10, 2005" sent 05/10/2005
- o [Minutes approved](#)
- Handling of errata against v2.0
 - o Decide on one of the alternatives:
 - § Increase minor version number to fix errata only
 - § Release corrigenda for v2.0 (separate document correcting semantic errors, without release of a full document)
 - § Wait for next version and provide both errata fixes and new functionality in one step
 - o Discussion:
 - § Releasing errata as part of 2.1 would still provide fixes to users faster than errata was addressed from version 1.0
 - § If there is a critical mass of addressed errata that is mostly or completely addressed, then it may make sense to release a corrigenda update now
 - § Issues: how severe current errata is, how long it will take to release 2.1 (still substantial work to do on memory management)
 - § [Agreed that interim release should not be done – would do either corrigenda or wait for release 2.1](#)
 - § Discussion tabled to allow further consideration
 - § To be revisited next meeting
- First draft of detailed requirements for callbacks
 - o See email from JH, Subject: First draft of detailed requirements for callbacks, Date: 19 May 2005 08:46:00 -0700
 - o Question of what address space would be used to invoke the callback
 - o Easiest solution: Callback would be performed in the same address space in which it was first attached
 - § If solution wants to go to the effort in the interrupt handler to load the context used in the initial call, implementation may do so
 - o When will the callback be re-invoked after first invocation?
 - § Invoke first time when notification takes place for EVD
 - § Invoke again after it has been drained dry
 - What verbs support today
 - Startup semantics tailored for request for notification to function
 - o Question on CB7
 - § “after callback routine has first been attached to an EVD it will not be invoked until consumer has dequeued all events from the EVD”
 - Necessary to dequeue all events from callback handler but not sufficient
 - To get first callback, must:
 - o Attach callback routine
 - o Must dequeue all events from EVD

ICSC ITWG Meeting Minutes

5/24/2005

- Notification must subsequently take place for EVD
- Question on CB8
 - § “Once invoked, must not be invoked again until consumer has consumed all events from the EVD”
 - Should this be “*callback routine has returned and consumer has consumed all events...?*”
 - Interpretation is correct – there can only be one instance of the callback routine active at any time, so it must return
- Methods of notification (fd and callbacks)
 - § If you are using a callback, should that be the only notification mechanism?
 - Question raised on reflector
 - General agreement that there should only be one notification mechanism per EVD
 - AI (JH): Propose requirements such that when a callback routine is attached to an EVD, that is the only notification pathway that functions
- FD semantics
 - § If an fd is associated with an EVD, can you still wait on an EVD?
 - § Does it make sense to wait on an EVD if an fd is attached to it?
 - § AI (JH): Research waiting on an fd attached to an EVD in current ITAPI
- Question on CB4
 - § Discussion of constraints on freeing an EVD from within a callback
 - § If you are in a callback routine and you attempt to free the callback routine your EVD is associated with that is a deadlock scenario
 - § CB4 is a case where you are trying to free a different EVD, but a callback for that EVD is still in progress
 - If you are trying to free a different EVD routine, that is not a deadlock condition
 - § Example
 - EVD A, Callback A, EVD B, Callback B
 - Callback A can free EVD B
 - Callback A can not free EVD A
 - If Callback A wants to free EVD B, then Callback A must wait until any active invocation of Callback B completes before freeing EVD B
 - § Why does CB4 specify waiting for the callback routine to complete, rather than returning an error?
 - Goal is to allow the free to complete once the potential deadlock condition is cleared
 - Don't want EVD free from within unassociated callback to be prohibited
 - Don't want EVD free from within unassociated callback to be unreliable

ICSC ITWG Meeting Minutes

5/24/2005

- Note that you can still deadlock if you allow multiple callbacks from different EVDs to occur concurrently
- § Could return a new return code, e.g. IT_ERR_EVD_BUSY
 - What would consumer do about it?
- § Could provide guidance to consumers on programming
 - Detach callback before freeing the EVD
- § Would implementations support multiple callbacks (for different EVDs) simultaneously
 - Not unreasonable for an MP configuration
- § If multiple callbacks can be active simultaneously, waiting for completion as specified CB4 could create a new deadlock condition (e.g. having Callback A wait for completion of Callback B to free EVD B, Callback B simultaneously waiting for completion of Callback A to free EVD A)
 - To avoid deadlock scenario, could return error value instead of waiting
- § Recommend that an attempt to free an EVD that has a callback routine attached and that callback routine is actively executing will generate an error
 - JH recommends a new error – IT_ERR_EVD_BUSY has a specific semantic that may not fit this model
- o Question on CB11
 - § CB 11 states that the list of routines that may not be called is limited to ITAPI routines
 - “Specification of which routines outside of the IT-API a callback routine may invoke is outside the scope of the IT-API”
 - § Is there guidance to consumers, e.g. resource allocating calls or potentially blocking calls should not be used?
 - § JH: Implementations may choose to allow this behavior – not clear what guidance we can provide that still allows implementations this flexibility
- Posting a List of Work Requests
 - o See email from FN, Subject: Re: Detailed requirements for posting a list of Work Requests, 4/27/05
 - o Continued discussion on number of work requests processed parameter
 - § From previous meeting, interpretation is that this parameter grows without bound and will eventually wrap
 - § JH: Should grow only to max work requests parameter
 - o FN: Proposed changing sense of parameter to # of work requests cached
 - o Discussion of the sense of the terms “processed” and “cached”
 - o Example for clarity in terminology (FN):
 - § Some work requests have been converted to WQEs but have not completed yet – call this *nl*

ICSC ITWG Meeting Minutes

5/24/2005

- § Some work requests have been posted, but have not been converted to WQEs yet – call this $n2$ (cached)
- § Number of “outstanding work requests” = $n1 + n2$
- § Number of “outstanding work requests” = number posted – number completed
 - Same as $n1 + n2$
- § What are we trying to convey to the consumer?
- § JH: Want to return “incremental $n1$ values”
- § Determined that further clarity in email would be helpful
- § **AI (JH): Provide additional description of what is meant in list of work requests proposal by # of work requests processed; include usage example**

Action item review:

- AI (FN): Ask Martin/Cathy whether it is possible to make hyperlinks blue in PDF documents.
 - o Resolved
- AI (FN): Report back to ITWG on semantics of IOVA output in IB’s Register Physical Memory verb.
 - o Several places in IB where this is also used as an output
 - o Semantics are not clearly described
 - o Intention seems to be that IOVA output should be the address that the consumer is going to use in the work request
 - o With zero based addressing, could be zero
 - o Purely a convenience for the consumer – can derive all this information for himself
 - o For zero based, consumer knows that he should use an offset of zero for the first byte
 - o Not clear what the benefit of this output will be
 - o ITAPI specifies that this should be an output, but does not specify the semantics
 - o Would like to see RNICPI WG either:
 - § Provide greater clarity on the semantics, or
 - § Eliminate IOVA as an output parameter
 - o AI (All HP): Follow up with RNICPI WG on semantics of IOVA output in IB’s Register Physical Memory Verb
 - o Pending
- 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

ICSC ITWG Meeting Minutes
5/24/2005

- AI (JH): Add text to the usage section of all posting calls to clarify appropriate usage to avoid WQ and CQ overflow
 - Pending
- AI (JR): Determine how DTO 6 high level requirement vote was resolved
 - Pending
- AI (All): Consider if the IT-API documentation should elaborate on async errors pertaining to the data source in all work request man pages
 - Pending
- Email subject "Shared state of LMRs", sent by FN on 05/23/2005:
 - Is there a reliable method to determine if an LMR is in the shared state?
 - How does one determine if an LRM is in the shared state?
 - § State diagram for iWARP shows that once you enter the shared state, there is now way back
 - § For IB, there is a paragraph that says you can convert an existing memory region to a shared memory region
 - Is there a semantic in the ITAPI (prior to 2.1) for Shared LMRs?
 - § Fast registration and invalidate are prohibited on Shared LMRs in 2.1
 - Would like shared state associated with an LRM to be meaningful of the current shared state of the LMR rather than the creation state of the LMR
 - § Would this inconvenience any previous consumer of the LMR?
 - If there is a single LRM, turn the shared flag on and don't share it with anyone, is it still shared?
 - § Note: First memory region you create is not shared
 - Recommendation:
 - § Support two different shared attributes associated with an LMR
 - Created with a shared bit
 - Currently in the shared state
 - § Trying to force them both into the creation flags mechanism could potentially break existing ITAPI applications
 - § What is the benefit of having a "created with a shared bit" flag?
 - Discussion of consumer benefit of "created with shared bit" flag
 - Use case not clear
 - AI (FN): Propose resolution for ambiguity of shared state of LMRs for LMRs that are a) created in a shared state and b) transition into a shared state
- Any other business

Meeting adjourned, 9:05am PDT