

ICSC ITWG Meeting Minutes
6/14/2005

6/14/05 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 **6/21/05**, 8am-10am PDT

ACTION summary:

PENDING AIs:

1. AI (MM subgroup): Add text to implementer's guide clarifying proper behavior if VA on input and output differ
2. AI (MM subgroup): Replace wording on line 4251 that currently reads "refers to the same physical memory pages as the new LMR" with "the identical set of physical memory pages as the new LMR" and add to the end of the sentence "and has the same length".

NEW AIs:

3. AI (FN): Summarize it_lmr_create issue of different VAs on input and output and provide text to resolve; post to bugzilla
4. AI (FN): Clarify definition of "same physical memory pages" and provide text to resolve; post to bugzilla
5. AI (JR): Post summary of physical to bus address translation issue to reflector for discussion
6. AI (JR): Update proposal summarizing today's discussion; discuss applicability of a flag parameter for both regions and(/or) windows
7. AI (MM subteam): Issue a revised set of detailed requirements with summary updates to prepare for votes
8. AI (MM subteam): Review MM-4.0.D3 to make sure list of completion errors is complete
9. AI (FN): Propose new mechanism for handling completion errors

ICSC ITWG Meeting Minutes 6/14/2005

- Agenda bashing, approve minutes
 - o Email from Fredy Neeser, subject "ICSC ITWG draft minutes for 31. May 2005", sent 6/1/05 1:31AM PT
 - o [Minutes approved](#)
- Action item review
 - o AI (MM subgroup): Add text to implementer's guide clarifying proper behavior if VA on input and output differ
 - § Proposal (as prev. discussed): After LMR create creates a memory region using register_physical verb that it go and check the actual IOVA that was allocated; make sure that the only adjustment is that it be adjusted to a page boundary – otherwise reject the registration
 - § **Pending**
 - § **AI (FN): Summarize it_lmr_create issue of different VAs on input and output and provide text to resolve; post to bugzilla**
 - o AI (JH): Create new errata from email from Jim Hamrick, subject "AI: Proposed text for avoiding WQ and CQ overflow", sent 5/31/05 2:34PM PT
 - § [Errata posted to bugzilla](#)
 - § [Closed](#)
 - o AI (JH): Create formal proposal for preferred method of DTO list requirements and present to RNICPI
 - § RNICPI reps driving this issue in RNCPI WG
 - § Vote scheduled for Thursday, 6/16/05
 - Some confusion on the current status of the vote
 - § Note that the terminology may not be clear – “posting to the RNIC” may be subject to different interpretation by members
 - § [Closed](#)
 - o AI (MM subgroup): Replace wording on line 4251 that currently reads “refers to the same physical memory pages as the new LMR” with “the identical set of physical memory pages as the new LMR” and add to the end of the sentence “and has the same length”.
 - § **Pending**
 - § **AI (FN): Clarify definition of “same physical memory pages” and provide text to resolve; post to bugzilla**
 - o AI (JR): Research previous discussions of physical to bus address translation responsibility and report to WG
 - § Suggestion from previous ITWG to add a flag to the mem_map call
 - § Important to consider efficiency of the process
 - § Is it sufficiently efficient to translate PBLs to bus addresses one address at a time?
 - § [Closed](#)
 - § **AI (JR): Post summary of physical to bus address translation issue to reflector for discussion**
- Memory Mgmt
 - o Email thread started by Jay Rosser, subject "Two additional MM issues", sent 6/6/05 4:03PM PT
 - o A)

ICSC ITWG Meeting Minutes
6/14/2005

- § User mode consumer not using ITAPI interacting with a kernel subsystem that does use ITAPI
- § Enhancement that would make it easier for kernel-mode consumer to perform registrations of the user mode address space such that zero copy from user mode buffers could be achieved by the kernel application
- § Had discussed previously as an enhancement to the map_mem interface
- § Issues with adding a generic capability to lmr_create
 - How to do allocation in an interrupt context, etc
- § May be sufficient to give the ITAPI implementation information that this is occurring
 - Propose to add an address space identifier flag to it_lrm_create call
- § Discussion:
 - Kernel mode consumer creates an LMR
 - How would this LMR actually be used? Would user library convert user's command to an it_post_send or it_rdma_write?
 - Yes
 - Example: kernel mode implementation of SDP
 - User mode consumer interface is SOCKETS
 - SDP implementation is translating user calls to ITAPI equivalents
 - Would this be in the user's context?
 - Yes, but on the kernel side of the system call
 - If you are in the user's context, would not the virtual address space of the user be visible to the system call?
 - Is it not the case that some OSs would allow the user's address space to be used directly?
 - FN: Believe Linux on x86 allows kernel to use user-space addresses
 - Some OSs do require some additional mechanism to access user space addresses from kernel space
 - E.g. creating both user and kernel aliases for a given region of memory
- o B)
 - § Pinning and mapping and registering with the card
 - § What if the consumer goes off and frees that buffer (just calls free() – poorly behaved consumer from ITAPI perspective)
 - § Proposed models:
 - Don't unpin the memory if free() is called
 - Don't deregister the memory if free() is called
 - § Both proposals have pros and cons
 - § Could abstract the housekeeping activities for it_map_pin; thus not require the ITAPI consumer to have to deal with any of these issues

ICSC ITWG Meeting Minutes
6/14/2005

- § FN: desirable to have pinning/unpinning on system to be done by a single entity to ensure that reference counting is done correctly
- § RNICPI assumes existence of OS service for pinning/unpinning
 - ITAPI could / should use the same service
 - Propose defining an interface to use this service
- § If you register a memory region in RNICPI there is an up call from RNICPI to the OS to do the pinning
 - Interface is specified by RNICPI
 - Propose that a similar (or, if appropriate, identical) interface be adopted by ITAPI
- § If pinning is required, verbs layer would do an up call into the access layer
 - risvc_pin call
 - Access layer (= OS) provides the pinning service
 - If consumer on top of ITAPI wants to pin, it would call down into the access layer (it_mem_pin)
- § Discussion:
 - For error case semantic, both up call and down call interfaces may need to be informed (depending on what state is tracked in the implementation)
 - Implementation dependent
 - Suggest raising the issue in the implementer's guide rather than mandating a particular solution
- § Related mail from Caitlin, 6/14/05
 - Issue: Using address space id in a context that is not the current execution context
- C)
 - § For lmr_link you don't need this flag because there is no reason to interpret the address
 - § For rmr_link, seems like it is the natural thing to allow a window binding programming model (although it may not be necessary)
 - Had to create the LMR in order to create the window
 - Can create an RMR in that LMR call
 - Seems natural that you should be allowed to create windows
 - Have to ensure that address space is consistent between memory Region and memory Window
 - Ex: Starting with an address in the user virt. addr space
 - Now bind MW to MR
 - Makes sense if the starting address of the memory window is in the same address space
 - § How else would you determine the starting address?
 - This would imply need for flag in rmr_link call
 - Alt Ex: 2 processes – one that creates LMR, another that binds RMR

ICSC ITWG Meeting Minutes

6/14/2005

- Kernel consumer creates LRM, possibly also RMRs
- User process binds LRMs to RMR
- How would you convey RMR handle across user/kernel boundary?
- For suggested use model, all ITAPI would be handled in the kernel
 - Rmr_link would be done in the kernel
 - Do you need a flag to RMR link to indicate that this is a user mode address that you are trying to bind?
 - § Current inputs to RMR link includes LRM handle that had been created using that user/kernel mode flag
 - § Is that sufficient information to pass to RMR link
 - § Desire to specify, but conflicting desire to not change the call signature
 - Fundamental problem of 2 address spaces for a single process
 - § Single bit can specify which address space to use
 - § Would be more explicit / potentially more clear to provide the same flag to both calls
 - Does require change to call interface
 - No convenient flag parameter to overload
 - If we don't have the address space in lmr_link, the interpretation of the address now depends on the mode in which the other object was created (namely the LMR)
 - Would this need to be an attribute of the LRM? How else would you determine the starting address returned by lmr_query?
 - § **AGREED: If flag is added, would need to be an attribute of the LMR**
- Is it preferable to support or not support RMR binding?
 - Could require at creation time that the MW be bound to a user-mode address
 - Could add the flag for both lmr_create and rmr_create
 - Would provide consistent interface for LRMs and RMRs
 - § Would still require a change to the signature of rmr_create – it doesn't have a flags parameter
- § **AI (JR): Update proposal summarizing today's discussion; discuss applicability of a flag parameter for both regions and(/or) windows**
- [JH departs]

ICSC ITWG Meeting Minutes
6/14/2005

- o Email from Fredy Neeser, subject "MM Detailed Requirements - June 10", sent 6/10/05 8:34AM PT
- o **AI (MM subteam): Issue a revised set of detailed requirements with summary updates to prepare for votes**
- o A) Issue: What happens if you mix/match priv. and non-priv. endpoints?
 - § Mixing could occur due to use of a common shared receive queue
 - § Through the common SRQ, could post receives which use a direct LRM handle
 - Not an issue as long as the receives are processed on the priv. endpoint
 - If you also have non-priv. endpoints, you would get an error when the receive is actually processed
 - § Originally wanted to avoid this by:
 - Distinguishing priv. and non-priv. SRQs
 - o Can only post priv. WQs to priv SRQ
 - If EP is not priv., could not associate with a non-priv SRQ
 - § Problems with Original proposal
 - Required processing in the fast path
 - o Look at the LRM handles in any work request
 - o Alternative to avoid that processing overhead
 - § Strictly constrain association of priv EPs and priv SRQs to endpoint creation time
 - § Achieves the same result with greater efficiency
 - Fast path check may not be needed because there is an equiv. check when the WRQ is actually processed
 - o Receive work request comes to SRQ
 - o Error will occur if STag 0 work request is dequeued for a non-privileged endpoint
 - § Question of whether this is all necessary
 - Can not fully protect the consumer from all hazardous use models
 - Providing good documentation on use cases and risks may be preferable to doing the work to prevent a dangerous use model
 - § Proposed fix: Document that hazard that exists in the application usage section
 - Proposed text is in the revised detailed requirements (6/13/05 version, not yet posted to reflector)
- o B) Review of MM Detailed Requirements status
 - § Direct LRM handle section is being revised – not ready for vote
 - Completion error issue
 - o IB verbs define this type of error as local completion error
 - o iWARP verbs have a more explicit error code
 - o Should there be 2 different error codes for this?
 - o Options
 - § Invalid STAG

ICSC ITWG Meeting Minutes

6/14/2005

- § Local QP non privileged
 - Proposal to simply return “Local Protection Error” to consumer; implementation could provide more detail if required
 - Proposal for new completion error – could be used in other contexts
 - § E.g. fast register on non-priv. endpoint
 - § Both IB and iWARP treat the capability of doing this through a single flag (attribute of a QP)
 - § Single error if the user of this attribute is violated
 - § Could you tell on IB that this happened?
- Proposal: Leave it up to implementer to return “local QP non-priv” or “local protection error”
 - Invalid STAG clearly maps to local protection error for ITAPI
 - QP not in priv mode maps to local EP non-privileged
 - Consumer would have to write code that would handle both of those errors
- Alternate proposal:
 - Use local protection error only
 - § IB also has a separate, distinct error code for fast register errors – “memory management operation error”
 - § New error code in IB v1.2?
- Agreed that it is preferable to provide a single error for a given failure type across all transports
- Further investigation required
- **AI (FN): Propose new mechanism for handling completion errors**
 - Email from Fredy Neeser, subject "Atomics and access control settings for MRs and MWs", sent 6/13/05 6:21AM PT
 - § Meeting adjourned prior to discussion
- General IT-API 2.1 discussion
 - Email from Jim Hamrick, subject "Please enter future bugs/errata into bugzilla", sent 6/9/05 4:14PM PT
 - § No discussion
- Any other business
 - Non-privileged endpoint processing a receive that comes from a shared receive queue
 - § Discussion captured above
 - Fast memory registration on a device that supports variable page sizes in it's physical buffer list – is there a completion error that should be surfaced
 - § If you allocate using variable sized pages and fast register, but pass in a PBL with an incompatible set of pages on it

ICSC ITWG Meeting Minutes
6/14/2005

- § Is this possible?
- § Are there error codes to cover this case?
- § If you use a page size in a PBL that is not supported by the RNIC, what does it do?
 - Same error code would be appropriate
 - iWARP has “Invalid page size” as a completion error for fast register (p212, quite a few errors that can be generated)
 - IB 1.2 spec (p635, line 16/17) specifies “PBL exceeds size of allocated PBL”
- § Proposal to define error value and allow implementers to map transport specific errors to ITAPI error values
- § **AI (MM subteam): Review MM-4.0.D3 to make sure list of completion errors is complete**

Meeting adjourned, 10:07am PDT