

3/8/05 ICSC ITWG Meeting Minutes

Taking minutes: HP (FW)

Y HP Jim Hamrick (JH) (joined at 10:23 PST)
Y HP Jay Rosser (JR)
Y HP Fred Worley (FW)
Y IBM Fredy Neeser (FN)
N NetApp Arkady Kanevksy (AK)
N Sun Matt Pearson (MP)

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 3/15/05, 10am-12pm PST

ACTION summary:

PENDING AIs:

1. AI (JR): Lead further discussion to complete: Produce requirements for RDDP WG informational draft to support IOH functionality.
2. AI (JR) – do some experimentation to fix how it_post_rdma_read_to_rmr appears in the “legal blackline” comparison document.

NEW AIs:

3. AI (JR): Add new requirement: It is mapper’s responsibility to ensure that the mapping is w.r.t. the appropriate IA
4. AI (RJ): Propose errata on 2.0 that having successfully completed socket conversion calling close on the SD has no effect on the RDMA connection
5. AI (FN, All): Provide feedback on RJ’s proposal for compatibility header file text and backward compatibility section
6. AI (JR): Fix DTO defect (Errata global numbers 200, 201)
7. AI (JR by Friday, 3/11/05): Investigate and respond to propose to use source-incompatible solution to address type issue (Errata global number 202)
8. AI (JR): Request status update from Martin with expected delivery date of the document back to the WG
9. AI(JR): Make ORD 3.0 a High Priority HLR for v2.1
10. AI (JR): Mark MM-9.11 as complete and remove from the priority list\

Follow-up agenda items:

- § Reach agreement on handling type abuse in `it_lmr_triplet_t` – is it acceptable to resolve in a way that breaks source code compatibility?

Minutes:

- o Agenda bashing, approve minutes
 - o Minutes to approve:
 - o Minutes for 8. Feb. 2005,
email "ICSC ITWG draft minutes for 2/8/05" sent on Feb 8 by Fred Worley.
 - o Minutes for 15. Feb. 2005,
email "ICSC ITWG draft minutes for 15 Feb 2005" sent on Feb 15 by Jay

Rosser.

[All minutes approved](#)

o Action item review

- AI (FN): Get rid of MM-16.4.D2.1 and MM-16.4.D2.2
- AI (FN): Document that an unlink operation that fails remotely will not directly manifest as an error locally. Also add text to Implementers guide that they need to absorb the Async Error and not manifest it via the IT API.
 - o [Discussion of both of above AIs]
 - o Different behavior for iWARP and IB
 - o Adjusted detailed requirements to allow for common behavior in the API for IB and iWARP
 - o [Changes complete](#)
- AI (JR): Generate a proposal for a mapping service to generate a PBL with bus addresses) given a virtual address range and a virtual address space ID(virtual addresses)
 - o See mail "AI - mapping service" from JR, sent 3/7/2005 4:12 PM
 - o Provides requirements for mapping service that generates PBL
 - o Such service helps 2 use models
 - § Case where IO addresses and processor addresses are not identical
 - § Privileged consumer attempting to map user mode addresses when not in process context
 - Potentially the interrupt context
 - o There are other mechanisms that could solve this problem
 - § E.g. `it_lrm_create`
 - § However, in some cases (e.g. STag of 0, fast memory registration) where this new interface would be helpful
 - o Discussion of `lrm_create`:
 - § `Lrm_create` must therefore both pin and map memory
 - § It will generate bus addresses (or equiv)
 - § This may not be visible to the consumer

- Discussion of new map_mem use model
 - § What would consumer do to use this?
 - Pin buffer
 - Call it_map_mem
 - Possibly several times
 - Construct a PBL
 - Use BPL for fast registration
 - PBL could also be passed to it_lrm_create (in v2.1)
 - § Additional use models:
 - With STag of 0, there is only a single segment you can use
 - If it_map_mem results in a single triplet that covers the entire memory range then you can use an LMR triplet
 - DTO calls allow you to pass an array of LRM triplets
 - Could create an array from the results of multiple calls to it_map_mem
 - Must not exceed the maximum length of a segment
- Use model summary: 3 use models
 - § Fast memory registration
 - § LMR create with a PBL (v2.1)
 - § Direct LMR handle (STag of 0)
- Requirements summary:
 - § New mapping call it_map_mem
 - § It_map_mem call will construct contiguous bus addresses (accessible by the RNIC) and returns the mapping
 - § Consumers responsibility to construct the PBL
 - Major simplifying assumption
 - Difficult to pass PBLs around
 - § Address identifier in requirement 22.3 provides ability to allow kernel consumer to map user memory
 - § Note on type of address space identifier (22.4.2.2)
 - 64 bits may not be sufficient
 - 128 bits expected to be sufficient
 - Void* or uint64_t may not be appropriate
 - Need to come up with an appropriate value for this type
- Will it_lrm_create attempt to validate a PBL?
 - § No; if you pass in the PBL it_lrm_create will not validate
 - § If instead you pass in an address range, it_lrm_create will create the PBL itself
 - § Users of PBLs will be privileged applications; they are assumed be trusted to be well crafted
 - § The consumer is also responsible for ensuring that memory is pinned when a PBL is used
 - API is not responsible for validating
 - § Note that text should be added to the man page to warn consumer of this responsibility

- § If you are using a PBL then it is assumed by the API that the memory has already been pinned and mapped (for the IA of interest) and the API has no responsibility to validate that the memory has been pinned or mapped.
 - PBL by itself does not have an association with an IA
 - It is mapper's responsibility to ensure that the mapping is w.r.t. the appropriate IA
 - AI (JR): Add new requirement: It is mapper's responsibility to ensure that the mapping is w.r.t. the appropriate IA
 - Additional discussion:
 - § What should the type of pointers in PBLs be?
 - Void* should not be used
 - Would uint64_t * be preferable?
 - Is uint64_t enough?
 - Fundamentally, IO address size is platform dependent
 - Appropriate to make the size of the data structure OS dependent
 - Can use existing it_busaddr_t type and make size OS dependent
 - How can you efficiently map for small PBL elements?
 - For page lists, would have larger pages
 - For block lists, would have to call map mem many times and this could be inefficient
 - Would pointer arithmetic for it_busaddr_t types be required?
 - Working assumption based on discussion that math ops are not required
 - Option exists for creating math ops in the API if they should be needed for the it_busaddr_t
 - Agreed to consider efficiency and review in later meeting
 - § Is the name "Physical Bus List" appropriate given that the IO addresses will not always be physical?
- AI (JR/FN): Determine if keeping around socket descriptors for lifetime of converted connection is too resource-consuming
 - Problem: OSs typically assign SDs from file descriptor space
 - § Fd space is a scarce resource for the system
 - § Goal to reduce demands on fd resources
 - § Desired to eliminate the fd resource after conversion of the socket to RMDA mode
 - § However, requirement to be able to change socket options (e.g. keepalive)
 - Question:
 - § Is the fd space sufficiently scarce that we need to be concerned about preserving it?

- RNICPI WG direction
 - § Believe RNICPI wants to give themselves the liberty to free up a kernel TCP/IP stack resource and replace it with a card TCP/IP stack resource
- After discussion, believe:
 - § Fd, SD do not need to change
 - § LLP may change
 - § Does not constitute errata for ITAPI v2
- What can consumer do with socket after conversion?
 - § Can pass to setsockopt or to close
 - § Why do we support close?
 - Close would give the fd and socket resources back to the OS
 - JR: believes that closing the SD after conversion of connection to RDMA mode would not close the connection; it would simply release the resources of the SD
 - FN: believes that closing the SD after conversion of connection to RDMA mode would close the connection; believes that close support should be removed and that it_ep_disconnect should clean up the SD
 - Man page does not specify what will happen if the SD is closed
 - Proposed New Errata:
 - Agreement: After successful conversion of a socket to RDMA mode, closing the SD should not effect the RDMA connection
 - AI (RJ): Propose errata on 2.0 that having successfully completed socket conversion calling close on the SD has no effect on the RDMA connection
- LLP handle can change when you call modify_qp_to_rpf
- First time you provide SD to modify QP call
- Want to add a requirement that a consumer who wants to fiddle with the LLP handle later on must first query the QP to see if the SD has changed or the LLP handle has changed
- Allows the implementation not to keep around the SD after creation
 - § Can keep around an alternate descriptor that has lower resource consumption
 - § Could also keep the same SD if required
 - § Allows the implementation to either change or not change the SD
- How would this be presented to ITAPI consumers?
 - § If you want to manipulate a socket option, e.g., keepalive
 - Could no longer use the original SD
- Proposal (JR): fd doesn't change, SD doesn't change, but LLP handle may change
 - § There is a close association between the SD and the fd, but LLP handle may actually change during the modify_qp call

- Completed
 - AI (JR): Lead further discussion to complete: Produce requirements for RDDP WG informational draft to support IOH functionality
 - Pending
 - AI (JH) – Supply a definition for the Service Request Handle object
 - Completed
 - AI (FN) – Fix language of Sections 1.5 and 4.1
 - Completed
 - AI (JR) – do some experimentation to fix how `it_post_rdma_read_to_rmr` appears in the “legal blackline” comparison document.
 - Deferred; need to get spec back from the OpenGroup
 - Pending
 - AI (FN) – Draft a sentence about local errors being described in `it_dto_status_t` as a cross reference in all DTO pages (should appear as a first sentence in ASYNCHRONOUS ERRORS section).
 - Completed
 - AI (JR) – Replace `ird_ord_support` flags in `it_ia_info_t`. Where `ird_ord_support` is `IT_TRUE`, add text stating that if `IRD_ORD` suppression is attempted on a transport that cannot support it (say `IB`), then an immediate error will be emitted (`IT_ERR_INVALID_FLAGS`). If the `ird_ord_support` is `IT_FALSE`, add text stating that `IRD_ORD_SUPPRESS` flag is ignored.
 - Completed
 - AI (JR) – Compile all the needed compatibility header file text and add to the BW Compat section (as well as to the header file).
 - Completed
 - JR would appreciate review
 - See section at line 10926 (beginning (2nd page) of Appendix C)
 - AI (all) – Review Tentatively Done TODOs (TODO list) as per FN’s email.
 - Completed
- o Errata against v2.0 Draft
- See email thread "Latest IT-API defect spreadsheet" started by Jim Hamrick on March 8
 - Type abuse in `it_lmr_triplet_t` - Email thread started by Jim Hamrick on Feb. 16
 - When we have a direct LRM handle, are addresses used with that something that comes from LRMs or can consumers use a BPL directly?
 - When you use a 0 STag you use a bus address
 - § This is another type of address that would be used

- Several options to address the problem
 - § Union – size_t, it_busaddr_t, other
 - Not backwards compatible
 - How important is backwards compatibility?
 - Changing from it_lmr_triplet_t to another type may create more work for the consumer than changing it_lrm_triplet_t itself
 - Creates a single place where the change (for consumers need to be made)
 - Would not require changes for the man pages
 - In the future, we can simply add another field to the union with a new name – having broken source code compatibility once you don't need to do it again
 - Do you need a type identifier for the union discriminator?
 - You can query the LMR to determine that it uses absolute or relative addressing
 - Can you take an arbitrary LMR handle and always discriminate how an address needs to be interpreted?
 - If you want to do on output could expose another attribute in 2.1 for LMRs
 - If you have a conventional LMR you can query it
 - Can determine relative or absolute addressing
 - Can't query a direct LMR handle
 - § It is just a constant
 - § Therefore address would have to be a bus address
 - Therefore, it is possible in all cases to determine what the address type will be based on the LMR handle
 - § Which is more important, breaking source code compatibility or adding work to the man pages?
 - FN: using a union appears to be a cleaner solution from a technical perspective
 - **Agreed to defer decision to next meeting**

○ Publication Process for IT-API v2.0

- ICSC Status Update
 - No update – contact at ISCS is on holiday
- Errata against v2.0 draft that should be fixed before publication
 - Things that are severity 2 and below do not need to pause the publication for
 - Severity 3 and above items should be addressed before publishing v2.0 of the API
 - Agree that top Errata should be resolved before publication

- AI (JR): Fix DTO defect (Errata global numbers 200, 201)
- AI (JR by Friday, 3/11/05): Investigate and respond to propose to use source-incompatible solution to address type issue (Errata global number 202)
- Other TODOs?
 - AI (JR): Request status update from Martin with expected delivery date of the document back to the WG
 - Note that we do not necessarily need to wait until the document comes back to implement our fixes; updating the returned document with the fixes should be straightforward

o Prioritization for IT-API v2.1

Proposal from email thread started by Jay Rosser on 03/07/2005:

- High priority:
 - All the remaining new memory management requirements (MM-X)
 - Callbacks for DTO, CM, and Async events for privileged consumers (P1P2-T9.3.1, P1P2-T10.3.1, P1P2-T11.3.1)
 - All of the Affiliated/Unaffiliated Events/Errors (AU-X) (though I am not totally clear whether or not they are already done).
 - IA exposing whether or not overflow detection occurs on DTO SEVDs (DTO-3.0) (arguably an errata on IT-API 2.0)
 - Support for posting a list of DTOs (DTO-1.0)
 - InfiniBand Atomics (P1P2-T3.5)
- Medium priority:
 - Explicit alternate path support (P1P2.P1.X and P1P2-P2.X and P1P2-T3.3)
- Low priority:
 - Event completion coalescing (P1P2-T9.6)
 - Overflowing work queues (I think the current vote just correlates to IT-API 1.0 concepts, so nothing may need be done) (DTO-6.X)
- Discussion of appropriate priority for IB Atomics
 - IB supports Fetch and Add and Compare and Swap
 - JR believes they could provide benefit to a number of applications
 - Could implement as a single DTO operation with flags to indicate each of the 2 possible atomic operations to perform
 - § Everything else needed to make an atomic operation work is already in the API
- Discussion of DTO-ORD3
 - Agreement that this should be addressed
 - Performance enhancement, not correctness issue
 - OK to address in v2.1 rather than as errata to v2.0
 - See lines 8754
 - AI(JR): Make ORD 3.0 a High Priority HLR for v2.1

- § Discussion of MM-9.11
 - Have addressed without noticing by giving more flexibility to applications – can use wide RMRs even on iWARP
 - AI (JR): Mark MM-9.11 as complete and remove from the priority list

- MM issues for v2.1
 - Continue revision of Detailed MM Requirements
 - Update document to reflect v2.0 (e.g. Absolute vs. Relative Addressing)
 - Continue building consensus on MM features and detailed requirements for v2.1

 - it_lmr_link - rmr_context should be an OUT (or IN/OUT?)

- Next steps
 - § Continue MM requirements completion
 - § Discuss callback function issues
 - Questions about callback functions:
 - Are callback functions associated with an endpoint, send queue or receive queue?
 - Are callback functions a replacement for the EVD?
 - When does a callback occur?
 - § Is a callback function called once per generated completion queue element?
 - Raises questions about completion suppression
 - Why are callbacks only important about privileged consumers

- Any other business

Meeting adjourned, 12:17pm PST