

ICSC ITWG Meeting Minutes, 30. Nov. 2004
Taking minutes: HP (jr)

Meeting attendance diagram (if you have more than 1 minus within the last four ITWG meetings, you are not eligible to vote):

mtg	date	hp	ibm	netapp	sun
---	-----	--	---	-----	---
m-3		+	+	-	-
m-2		+	+	-	-
m-1		+	+	-	-
m-0	nov 30	+	+	-	-

Present:

jim hamrick (jh) [HP]
fredy neeser (fn) [IBM]
jay rosner (jr) [HP]

Minutes to approve: Email from Jay Rosner, subject "ICSC ITWG draft minutes for 11/16/04", sent 11/16/04 4:59PM PT - **Approved**.

Action item review

AI (jr): Add text to diagram to say that TCP ACK could be piggybacked on the first FPDU.

jr: Sent out iWARP Implementer's guide with a footnote on every ACK indicating piggyback possibly on next transaction.

fn: Agree that seems okay.

jr: **Closed**.

AI (jr): Add discussion of how IRD/ORD negotiation is done into app usage section of it_ep_connect man page, and reference to it from app usage section of it_ep_accept man page.

jr: **Pending**.

AI (fn): Adjust current text describing how to unbind a narrow RMR after QP disconnected.

Deallocate bound MW

Dellocate bound MR

fn: **Pending**.

CM issues:

Email from Jay Rosner, subject "Third draft of CM man pages", sent 11/29/04 2:19PM PT

jr: Mentions new draft of pages.

all: Have not had chance to look them.

Errata:

Email from Fredy Neeser, subject "Erratum - Logical OR", sent 11/29/04 8:56AM PT

fn: Logical OR in 'C' language is specifically for TRUE/FALSE. Intent of the fields in the text were flags and make more sense to be treated as bitwise OR.

jr: Agree.

fn: Difficulty is a text change all over the place. It should be in all cases "bitwise OR".

fn: However, `it_rmr_bind()` page allows a few options but the definitions of the bits have a few inconsistencies. For instance, `IT_PRIV_REMOTE (0x0010)` (found on `it_lmr_create` page) really is "IT_PRIV_REMOTE_READ | IT_PRIV_REMOTE_WRITE" in meaning. This could be represented as `0x000c` (i.e. "0x0008 | 0x0004").

jh: Notes that this case has no source code issues since this is a new value.

fn: Has more of a problem with `IT_PRIV_READ_ONLY` since it is mutually exclusive with the remainder of the bits.

jh: Notes that `IT_PRIV_READ_ONLY` prevents remote write (for LMRs - binding windows still possible, however)

fn: Thinks it would be preferable to have an explicit WRITE flag.

jh: Recalls (vaguely) that there may have been an issue that Steve Sistare had exposed with this.

fn: To enable local access only, what needs be set?

jr: Use `IT_PRIV_DEFAULT`.

fn: Notes that the text on line 3671 is ambiguous (and is errata GN3).

jh: The errata GN3 resolution removes the ambiguity (eliminates notion of '0' and specifies local read/write).

fn: Still concerned about `IT_PRIV_READ_ONLY` since it excludes other settings.

jr: Notes that `IT_PRIV_NONE` also mutually excludes the other settings.

jh: Actually `IT_PRIV_NONE` is used for `it_rmr_bind()`.

fn: Text of `it_lmr_create()` indicates that `IT_PRIV_NONE` is not meant to be used for this call.

jr: Text is slightly ambiguous when read out of context (what does "here" refer to?).

fn: Looking at `it_rmr_bind` errata GN19, suggestion was to encourage use of `it_rmr_unbind` instead of `it_rmr_bind` with `IT_PRIV_NONE`.

jr: Doesn't the `IT_PRIV_NONE` allow you to keep the `R_key` valid but prevent access - isn't this a valid programming model?

fn: Yes.

jh: GN19 still allows use of `IT_PRIV_NONE`

fn: Options for MWs - disable remote access privilege, rebind MW with length 0 (only for Wide RMRs),

jh: States that `IT_PRIV_NONE` requires `it_rmr_bind` and thus will generate a new `R_key` - there is no valid programming model and the GN19 errata is accurate.

fn: Suggests removing `IT_PRIV_NONE` from the enum and defining it as zero, suggests removing `IT_PRIV_READ_ONLY` from the enum and defining it as a separate value. Suggests remaining bits be defined as `local_read`, `local_write`, `remote_read`, `remote_write`. And new defines of "local" = `L_R+L_W` and "remote" = `R_R+R_W`.

fn: Wonders why you would define a read-only LMR?
jh: Protect you from yourself?
fn: Thinks then that write-only LMR is also valid since RNIC can then write region but local Consumer has complete liberty to examine region.
fn: Summarizes he would like to clean up definitions of bits and wants a reason why not.
jh: Notes that IB specification mandates local read access on memory regions (IB1.2 page 595 line 1). Cannot create write-only regions in IB. This is why Steve defined the IT_PRIV_READ_ONLY privilege.
fn: Can do so in iWARP.
jh: Is it worth the effort to support iWARP write-only regions as a Transport-dependent programming model?
fn: Thinks, nonetheless, that clearing up confusion with the bits would be an aid to the Consumer.
jh: Concerned about source-code compatibility with IT API 1.0 - should try to avoid busting this.
AI - fn: Generate a proposal for cleaning up access bits. Something like propose a set of bitwise ORable privilege flags that are source code compatible with current IT API and identify invalid combinations.

MM issues:

Email thread started by Fredy Neeser, subject "Memory management: Requirements v0.98 / Voting of updated Narrow RMR binding requirements", sent 11/24/04 10:03AM PT

fn: MM-9.0.D5.[1|2].3 - intent was to give Consumers indication of what can be done with an RMR. For instance, can a peer remotely invalidate it? Similarly, there is a flag for LMRs - local vs remote invalidation.
fn: jr had responded that this seemed like an IA attribute
fn: Notes that IB has a notion of LMRs some of which can be unlinked and others that can be. Thus the LMR attribute.
fn: Instead of LMR attribute, proposes to expose the nature of the invalidation capability via an IA attribute per MW type (i.e. narrow vs wide). However, fn notes that these are constants and thus could simply be specified in text. On the other hand, fn also thinks it may be the case that iWARP may add wide windows, so the IA attribute is future proofing.
fn: Describes how using IA attributes will reduce state maintained by Consumer and implementation. Consumers wishing to use unlink should always create lmr via `it_lmr_create_unlinked()`.

<some discussion not captured in minutes>

fn: MM-10.4.D3.8 and MM-17.1
fn: Proposes an alternative approach here where RI checks the access privileges on posting request.
jr: Concerned that implementations already adhere to the verbs here.
AI - fn: Clarify error handling on locally detected errors on RDMA read requests (say sink buffer access privileges wrong - handle as per verbs or detect locally first?).

fn: MM-12.0.1.D2 cannot use wide RMR as a sink for an RDMA read thus don't need to check for invalidate.

fn: Has sent out a new document summarizing the consensus, subject "Updated Narrow RMR binding requirements for vote", sent 11/30/04 9:43AM PT.
jr: We will call an email vote for 12/3/04 to approve this document.
AI - fn: To generate a ballot based on the above email with the date of vote set to 12/3/04.

Meetings during Xmas:

jr: No meetings the week of the 12/27/04 and 1/4/04.

Rename RMR context to something more meaningful:

fn: Notes that "RMR context" can refer to remote LMR and remote RMR. Suggests that renaming this will reduce the confusion in the specification.
jr: Asks to table this to think about it.
AI - all: Think about the consequences of renaming and suggest terminology.

Question on documenting the required access privileges for data sources/sinks:

fn: Current IT API pages do not specify the privileges required for an RDMA operation. It is particularly a problem for sinks for RDMA reads - for IB local write privilege is required, for iWARP, remote write privilege is required.
jr: Agrees such text is desirable.
AI - fn: Add a sentence per RDMA DTO man page describing access privileges for sinks and sources is sufficient.

Advertising for members within OpenRDMA:

fn: Thinks it might be useful to advertise for new members there.
jr: Suggests tabling for next meeting pending internal discussion on topic.

Meeting adjourned about 15 minutes late.