DAT 2.0 immediate data proposal

January 2006

Contents

1. Design Overview

Some RDMA capable transports provide a mechanism to send a 32-bit immediate data buffer along with message sends and RDMA write operations. The immediate data is guaranteed to arrive, in order, after all message or RDMA write data is placed at the remote endpoint. This feature allows the receiving end of the RDMA write operation to be immediately notified of the RDMA write completion independently of sending a separate send message after the operation.

2. Structures and Types

2.1.1 DAT_DTO_COMPLETION_EVENT_DATA (modified)

To accommodate immediate data delivery via events, DAT_DTO_COMPLETION_EVENT_DATA is augmented with an additional entry to include 32-bit immediate data.

```
#include <dat/dat_extensions.h>
typedef struct dat_dto_completion_event_data
{
      DAT_EP_HANDLE                 ep_handle;
      DAT_DTO_COOKIE                user_cookie;
      DAT_DTO_COMPLETION_STATUS     status;
      DAT_SEG_LENGTH                transfered_length;
      DAT_DTOS                      operation;
      DAT_RMR_CONTEXT               rmr_context;
      DAT_UINT32                    immed_data;
      DAT_DTO_EXTENSION_DATA        extension;

} DAT_DTO_COMPLETION_EVENT_DATA;
```

2.1.2 DAT_DTOS (modified)

The DAT_DTOS specifies data transfer operation types.

```
typedef enum dat_dtos
{
      DAT_SEND,
      DAT_SEND_IMMED,
      DAT_RDMA_WRITE,
      DAT_RDMA_WRITE_IMMED,
      DAT_RDMA_READ,
      DAT_RECEIVE,
      DAT_RECEIVE_WITH_INVALIDATE,
      DAT_RECEIVE_IMMED,
      DAT_RECEIVE_RDMA_IMMED,
      DAT_BIND_MW,
      DAT_EXTENSION,
      DAT_INVALID
} DAT_DTOS;
```

2.1.3 DAT_DTO_FLAGS (new)

The DAT_DTO_FLAGS specifies special methods for posted operations.

```
typedef enum dat_dto_flags
{
    DAT_DTO_IMMED_FLAG         = 0x1,/* idata sent after buffer in order */
    DAT_DTO_IMMED_CONFIRM_FLAG = 0x2 /* confirm receipt before completion */
} DAT_DTO_FLAGS;
```

3. API's

Each API below details input/output arguments and completion semantics. Explicit return codes are not given but they can be assumed to be logical uses of existing DAT return codes.

3.1.1 dat_ep_post_send_immed() (new)

```
DAT_RETURN
dat_ep_post_send_immed(
      IN DAT_EP_HANDLE        ep_handle,
      IN DAT_COUNT            num_segments
      IN DAT_LMR_TRIPLET      *local_iov,
      IN DAT_DTO_COOKIE       user_cookie,
      IN DAT_UINT32           immediate_data,
      IN DAT_DTO_FLAGS        dto_flags,
      IN DAT_COMPLETION_FLAGS completion_flags);
```

This asynchronous call performs a normal message send to the remote endpoint followed by a post of an extended immediate data value to the receive EVD on the remote endpoint. The message and immediate data will consume one posted receive buffer at the remote endpoint.

DTO Flags:

DAT_IMMED_FLAG requests that the supplied 'immediate' value be sent as the payload of a four byte send following the RDMA Write, or any transport-dependent equivalent thereof.

DAT_CONFIRM_FLAG requests that this DTO not complete until receipt by the far end is confirmed. Event completions are as follow:

Endpoint	**EVD**	**DTO operation type**	**Event Data Type**
Initiator	Request DAT_DTO_COMPLETION_EVENT	DAT_SEND_IMMED	DAT_DTO_COMPLETION_EVENT_DATA
Remote	Receive DAT_DTO_COMPLETION_EVENT	DAT_RECEIVE_IMMED	DAT_DTO_COMPLETION_EVENT_DATA

3.1.2 dat_ep_post_write_immed() (new)

```
DAT_RETURN
dat_ep_post_write_immed(
      IN DAT_EP_HANDLE        ep_handle,
      IN DAT_COUNT            num_segments
      IN DAT_LMR_TRIPLET      *local_iov,
      IN DAT_DTO_COOKIE       user_cookie,
      IN DAT_RMR_TRIPLE       *remote_iov,
      IN DAT_UINT32           immediate_data,
      IN DAT_DTO_FLAGS        dto_flags,
      IN DAT_COMPLETION_FLAGS completion_flags);
```

This asynchronous call performs a normal RDMA write to the remote endpoint followed by a post of an extended immediate data value to the receive EVD on the remote endpoint. The immediate data will consume a posted receive immediate resource (could be zero byte posting) at the remote endpoint.

DTO Flags:

DAT_IMMED_FLAG requests that the supplied 'immediate' value be sent as the payload of a four byte send following the RDMA Write, or any transport-dependent equivalent thereof.

DAT_CONFIRM_FLAG requests that this DTO not complete until receipt by the far end is confirmed. Event completions are as follow:

Endpoint	EVD	DTO operation type	Event Data Type
Initiator	Request DAT_DTO_COMPLETION_EVENT	DAT_RDMA_WRITE_IMMED	DAT_DTO_COMPLETION_EVENT_DATA
Remote	Receive DAT_DTO_COMPLETION_EVENT	DAT_RECEIVE_RDMA_IMMED	DAT_DTO_COMPLETION_EVENT_DATA

3.1.3 dat_ep_post_recv_immed() (new)

```
DAT_RETURN
dat_ep_post_recv_immed(
      IN DAT_EP_HANDLE        ep_handle,
      IN DAT_COUNT            size,
      IN DAT_LMR_TRIPLET      *local_iov,
      IN DAT_DTO_COOKIE       user_cookie,
      IN DAT_COMPLETION_FLAGS completion_flags);
```

This call performs a normal post receive message to the local endpoint that includes additional 32-bit buffer space to receive immediate data. Operation will accept a zero byte message for receiving immediate data only. Event completion for the request completes as follow:

Endpoint	EVD	DTO operation Type	Extension Event Data Type
Initiator	Receive DAT_DTO_COMPLETION_EVENT	 DAT_RECEIVE DAT_RECEIVE_IMMED DAT_RECEIVE_RDMA_IMMED	DAT_DTO_COMPLETION_EVENT_DATA No Immediate data received with message Immediate data received with message send Immediate data received with RDMA write