NAME

rdma_connect - Initiate an active connection request.

SYNOPSIS

#include <rdma/rdma_cma.h>

int rdma_connect (struct rdma_cm_id *_id_, struct rdma_conn_param *_conn_param_);

ARGUMENTS

id   RDMA identifier.

conn_param   connection parameters. See CONNECTION PROPERTIES below for details.

DESCRIPTION

For an rdma_cm_id of type RDMA_PS_TCP, this call initiates a connection request to a remote destination. For an rdma_cm_id of type RDMA_PS_UDP, it initiates a lookup of the remote QP providing the datagram service.

RETURN VALUE

Returns 0 on success, or -1 on error. If an error occurs, errno will be set to indicate the failure reason.

NOTES

Users must have resolved a route to the destination address by having called rdma_resolve_route or rdma_create_ep before calling this routine.

CONNECTION PROPERTIES

The following properties are used to configure the communication and specified by the conn_param parameter when connecting or establishing datagram communication.

private_data

References a user-controlled data buffer. The contents of the buffer are copied and transparently passed to the remote side as part of the communication request. May be NULL if private_data is not required.

private_data_len

Specifies the size of the user-controlled data buffer. Note that the actual amount of data transferred to the remote side is transport dependent and may be larger than that requested.

responder_resources

The maximum number of outstanding RDMA read and atomic operations that the local side will

accept from the remote side. Applies only to RDMA_PS_TCP. This value must be less than or equal to the local RDMA device attribute max_qp_rd_atom and remote RDMA device attribute max_qp_init_rd_atom. The remote endpoint can adjust this value when accepting the connection.

initiator_depth

The maximum number of outstanding RDMA read and atomic operations that the local side will have to the remote side. Applies only to RDMA_PS_TCP. This value must be less than or equal to the local RDMA device attribute max_qp_init_rd_atom and remote RDMA device attribute max_qp_rd_atom. The remote endpoint can adjust this value when accepting the connection.

flow_control

Specifies if hardware flow control is available. This value is exchanged with the remote peer and is not used to configure the QP. Applies only to RDMA_PS_TCP.

retry_count

The maximum number of times that a data transfer operation should be retried on the connection when an error occurs. This setting controls the number of times to retry send, RDMA, and atomic operations when timeouts occur. Applies only to RDMA_PS_TCP.

rnr_retry_count

The maximum number of times that a send operation from the remote peer should be retried on a connection after receiving a receiver not ready (RNR) error. RNR errors are generated when a send request arrives before a buffer has been posted to receive the incoming data. Applies only to RDMA_PS_TCP.

srq Specifies if the QP associated with the connection is using a shared receive queue. This field is ignored by the library if a QP has been created on the rdma_cm_id. Applies only to RDMA_PS_TCP.

qp_num

Specifies the QP number associated with the connection. This field is ignored by the library if a QP has been created on the rdma_cm_id. Applies only to RDMA_PS_TCP.

INFINIBAND SPECIFIC

In addition to the connection properties defined above, InfiniBand QPs are configured with minimum RNR NAK timer and local ACK timeout values. The minimum RNR NAK timer value is set to 0, for a delay of 655 ms. The local ACK timeout is calculated based on the packet lifetime and local HCA ACK delay. The packet lifetime is determined by the InfiniBand Subnet Administrator and is part of the resolved route (path record) information. The HCA ACK delay is a property of the locally used HCA.

Retry count and RNR retry count values are 3-bit values.

The length of the private data provided by the user is limited to 56 bytes for RDMA_PS_TCP, or 180 bytes for RDMA_PS_UDP.

IWARP SPECIFIC

Connections established over iWarp RDMA devices currently require that the active side of the connection send the first message.

SEE ALSO

rdma_cm(7), rdma_create_id(3), rdma_resolve_route(3), rdma_disconnect(3), rdma_listen(3), rdma_get_cm_event(3)