Workshop in Communication Networks
Exercise 3

Keren Sfez - 342768264
Oz Matlaw - 329603674

Assignment Goal:

The goal of this exercise is to compare the Rendezvous protocol to a trivial “eager” case. The throughput was measured by sending messages from the client to the server using RDMA WRITE and receiving control messages from the server to the client using IBV_WR_SEND.

Warm-Up Phase:

In our warm-up phase, we conducted 50 iterations to allow the system to reach a stable state before collecting actual performance measurements. The warm-up phase played a crucial role in ensuring that the system had sufficient time to stabilize, minimizing any transient effects that could impact the accuracy of the measurements. By performing a substantial number of iterations, we provided ample opportunity for the system's resources to be properly allocated, caches to be populated, and adaptive mechanisms to settle into an optimal state. This approach allowed us to obtain more reliable and representative performance measurements during the subsequent data collection phase.

Chosen Parameters:

For the measurements, the following parameters were chosen:

- rx_depth: 6000
- tx_depth: 6000
- Number of iterations in Warm-up phase: 50
- Number of total iterations: 60000

Results:

SET THROUGHPUT MEASURE

Eager Protocol

1	0.000092	Gigabytes/Second
2	0.000183	Gigabytes/Second
4	0.000366	Gigabytes/Second
8	0.000728	Gigabytes/Second
16	0.001467	Gigabytes/Second
32	0.002922	Gigabytes/Second
64	0.005821	Gigabytes/Second
128	0.011676	Gigabytes/Second
256	0.023329	Gigabytes/Second
512	0.046841	Gigabytes/Second
1024	0.093497	Gigabytes/Second
2048	0.186629	Gigabytes/Second

Rendezvous Protocol

4096	0.067817	Gigabytes/Second
8192	0.119940	Gigabytes/Second
16384	0.203804	Gigabytes/Second
32768	0.357349	Gigabytes/Second
65536	0.563887	Gigabytes/Second
131072	0.884952	Gigabytes/Second
262144	0.599692	Gigabytes/Second
524288	1.402583	Gigabytes/Second
1048576	1.554689	Gigabytes/Second

GET THROUGHPUT MEASURE

Eager Protocol

1	0.000050	Gigabytes/Second
2	0.000099	Gigabytes/Second
4	0.000053	Gigabytes/Second
8	0.000396	Gigabytes/Second
16	0.000794	Gigabytes/Second
32	0.001591	Gigabytes/Second
64	0.003167	Gigabytes/Second
128	0.006317	Gigabytes/Second
256	0.012457	Gigabytes/Second
512	0.024887	Gigabytes/Second
1024	0.048707	Gigabytes/Second
2048	0.042823	Gigabytes/Second

Rendezvous Protocol

4096	0.172299	Gigabytes/Second
8192	0.117828	Gigabytes/Second
16384	0.205533	Gigabytes/Second
32768	0.362269	Gigabytes/Second
65536	0.550561	Gigabytes/Second
131072	0.751572	Gigabytes/Second
262144	0.950703	Gigabytes/Second
524288	1.111347	Gigabytes/Second
1048576	1.246729	Gigabytes/Second

The Rendezvous protocol is designed to handle large messages more efficiently than the Eager protocol. In the Eager protocol, the entire message is sent immediately, which works well for small messages but can result in high overhead for large messages. On the other hand, the Rendezvous protocol performs a two-step process. First, it sends a lightweight control message to establish the communication and the message metadata. Then, it transfers the actual data separately. This approach reduces the communication overhead for large messages and allows the protocol to scale better with increasing message sizes.

As a result, the Rendezvous protocol achieves higher throughput for both "SET" and "GET" operations, making it a more suitable choice for applications involving large message transfers.

Impact of Other Parameters:

Choosing different parameters, such as varying the depth, warm-up duration, or iterations, could lead to different results. For example, reducing the depth may result in lower throughput due to increased contention and overhead. Shortening the warm-up phase might introduce measurement artifacts and inaccuracies. Adjusting the number of iterations can affect the stability and statistical significance of the results. Careful consideration of these parameters is crucial to ensure accurate measurements.

Improvement from eager to rendezvous:

The results clearly indicate that the Rendezvous protocol achieves higher throughput for both "SET" and "GET" operations across various message sizes. As the message size grows, the Rendezvous protocol continues to scale effectively and maintains a consistently high throughput.

The improvement from the Eager to Rendezvous protocol is evident in the provided throughput measurements. For both "SET" and "GET" operations, the Rendezvous protocol demonstrates substantially higher throughput values compared to the Eager protocol, especially for larger message sizes. This improvement in throughput is critical for applications that involve transferring substantial amounts of data.

In summary, the Rendezvous protocol's ability to handle large messages efficiently and its scalability make it a superior choice over the Eager protocol for high-performance data transfers. Its optimized approach minimizes communication overhead and maximizes data transfer rates, enabling faster and more effective communication between the client and the server.