Challenge Question: Write a short essay (at least 500 words) comparing sockets-based communication with MPI communication.

While both socket-based communication as well as message passing interface (MPI) communication are used in distributed systems, they still have a few differences. Both can be used to provide a standard interface for communicating between processes, though sockets are more standardized, with an official governing organization controlling their protocols. The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011 and is commonly used as the foundation for other applications to be built on top of. Message passing interface libraries, on the other hand, do not have any specific controlling body. Instead, it is common to use one of a handful of open-source libraries such as MPICH or Open MPI. Both are available for use and together cover multiple variations of MPI. MPI-1 has no shared memory model whereas MPI-2 has only a limited distributed shared memory model. Designing programs to use the MPI model can provide some advantages when using non-uniform memory access architectures, as the MPI model encourages memory locality. Sockets are mainly used for sending information between a client and server, where a server is sitting with a socket open and ready to establish a (most likely TCP) connection. A client then connects to this open socket, initializes the connection, and then sends data. It is overall intended to prioritize one side over the other. Which side this is often depends on perspective, though it is often the client. With MPI, data is sent using a message passing model, where nodes simply convey information back and forth. There isn't necessarily one node doing most of the sending or receiving.

Sockets can send any type of data in any format and the client can usually receive it correctly and make sense of it. With MPI, there has to be intentional thought put into small details such as the endian-ness of any participating system. If two machines try to communicate over MPI with one using big-endian and the other using little-endian, there will be problems. This is not a factor with socket-based communications.

Sockets are also built to run at a higher level on the OSI model. Because they often use TCP/IP, the data is guaranteed to be sent reliably and correctly. Additionally, socket based connections connect a server and client via a port. On the other hand, MPI allows for a lot of flexibility and gives the user freedom to choose if data guarantees are important for the specific use case. This allows it to be more efficient in some cases and to potentially be used in a greater variety of places. MPI also can be implemented or configured to be run at various different levels of the OSI model.

Sockets are intended to be used more over the internet, where connections can vary and be more or less reliable day by day. MPI work best with reliable, local, high speed connections such as the connection typically found in data centers.

The primary place MPI excels over regular sockets is in large scale parallel computing systems. MPI is designed to work well in a highly-scaled environment and handle many processes and nodes.

Feedback Questions:

1) Was this homework too difficult, or too easy?

   This homework was not too bad, though it was very difficult trying to understand the best way to implement a pub-sub architecture.

2) Was the assignment fun or challenging?

   It was both fun and challenging, though the graduate question was difficult to answer to the required length.

3) Was there something that was unclear?

   The way that a pub-sub model was supposed to be implemented into the existing architecture was confusing.

4) Was the homework too long for the given amount of time?

   Extending the deadline to midnight today was great. I wasn't able to spend as much time on it over spring break as I would've liked but I was still able to work on it some when I had internet (which I used for references).

5) What did you learn from this homework?

   I learned about using redis' pub-sub features and other quality of life features I can use when programming in Python.