Justin Bahr
NUID: 001592707

High Performance Computing
Homework 4 - Question 4

I found a paper from 2020 titled “High Performance MPI over the Slingshot Interconnect: Early Experiences” which discusses HPE/Cray’s Slingshot interconnect, which was employed on Frontier, the first exascale supercomputer. The paper also examines current bottlenecks for performance and scalability across nodes and evaluates recent MPI and communication libraries.

MPI on Millions of Cores:
The paper defines any MPI function whose memory utilization or time consumed increases linearly or worse with respect to the number of processes as non scalable. Processes such as MPI_Gatherv, MPI_Scatterv, and MPI_Alltoallw, require arrays of size proportional to the number of processes which results in high memory consumption (i.e. 4 MiB per process for 1 million processes) and requires traversing these large arrays to determine communication needs. Applications with sparse communication patterns often misuse collectives like MPI_Alltoallv, leading to inefficiencies. A possible solution under discussion in MPI-3 is the introduction of sparse collective operations. Additionally, graph topology in MPI requires every process to store the full communication graph, leading to $O(p^2)$ memory usage at best. The new MPI_Dist_graph_create in MPI-2.2 addresses this issue by enabling distributed graph representations, reducing per-process memory requirements. Furthermore, group mapping becomes infeasible at extreme scales due to large memory bandwidth, so improved compact representations are needed to enhance scalability. Current fault detection and error handling methods are not efficient enough for exascale systems where hardware failures are more frequent. Also, non blocking collectives must be added to accommodate for potential load balances during synchronization. Finally, MPI can support multiple threads, but there is room for improvement in hybrid models.

High Performance MPI Over the Slingshot Interconnect: Early Experiences:
The Slingshot interconnect is commonly used in Top500 supercomputers and it offers new adaptive routing and congestion control. The main accomplishments made in MPI of the past 15 years are MPI libraries such as MVAPICH2-GDR, which are optimized for GPU acceleration, integration with hybrid programming such as OpenMP and PGAS for multiple cores, and simulation tools that allow programmers to study and analyze behavior on exascale systems. Shifting from InfiniBand to Slingshot has encouraged Ethernet-based networking to utilize Slingshot’s unique features and reduce latency. Another current barrier is the limited availability of Slingshot systems, limiting large scale benchmarking. Finally, the heavy power consumption of exascale computing and memory latency continue to pose barriers to high performance.

Sources:

[1] Gropp, W., & Lusk, E. (2010). *MPI on millions of cores*. Parallel Processing Letters, 20(4), 333–346. https://doi.org/10.1142/S0129626410000257

[2] Shafie Khorassani, K., Chen, C.-C., Ramesh, B., Shafi, A., Subramoni, H., & Panda, D. K. (2022). *High Performance MPI Over the Slingshot Interconnect: Early Experiences.* In *Practice and Experience in Advanced Research Computing* (PEARC '22). Association for Computing Machinery. https://doi.org/10.1145/3491418.3530773