QUESTION 4 (EXTRA):

MPI on Millions of Cores

Primary motive of this paper:

The primary motive of this paper is to examine the issue of scalability of MPI to very large systems. It discusses MPI specification and the areas with scalability issues. The paper also provides examples of non-scalable parts of the MPI standard include irregular collectives and some other functions that take array arguments of size proportional to the total number of processes. MPI features include support for building complex libraries, clear semantics for interoperation with thread and other techniques when dealt with memory and data constraints.

Positive and negative effects on exploiting parallelism that occurred after 2010:

- There are a lot of changes happened in the system architecture that occurred over recent years and there are lot of processing elements added to improve the computing performance. These system architecture are designed in a way to exploit parallelism on millions of cores.
- The conventional process mapping focused only on locality improvisation. In 2019, NUMA systems were developed to avoid congestion problem and it increases the overall performance.
- Millions of cores cause heavy congestion problems and it also needs MPI to map every process. The automatic mapping for adapting to communication was developed that automatically maps the locality and memory congestion.
- The Hardware infrastructure for scalability on MPI_INIT is well efficiently designed but the software infrastructure is vague and also. it requires optimized runtime environments when run in large machines. A scalable Launch/ Resource Reducing algorithm was developed to enable the launch of large-scale MPI applications.
- Lots of new algorithms were designed after 2010, that can use lots of cores parallelly to work on the scientific calculations.The performance of large scale scientific codes on many core processors is a tiresome process to optimize and a model called Lattice Boltzmann was developed to resolve this problem.
- Scaling problem is difficult to fix and also it takes a lot of time. It usually results in bug reproduction and is difficult to fix. Methods were developed to avoid this scalability problem that works fine on point to point communication. But the downside is that it is not suitable for collective communication.

Exascale computing:

The things I considered for contrasting between these two papers are

- Scientific Computing
- Scaling
- Hardware and Architecture challenges

Scientific Computing:

Exa-scale computing: In 1980s, vector supercomputing is used in High Performance computing and later in 1990s Parallel Processing and shared memory multiprocessors were taken into consideration. Today, clusters are used with computational accelerators in the form of coprocessors which also includes storage area networks. MPI and OpenMP are used for inter node and intra node parallelism.

MPI on MOC: It is carried out on a two-three dimensional mesh with nearest neighbour communication, but the application is parallelized with 1-D mesh results in contiguous buffer and it sends and receives messages.It may also result in non contiguous buffers when the number of processors and mesh cells scales up.

Scaling:

Exa-scale computing: Few years ago, some teraflops and terabytes were considered as the state of art technology for advanced computing.Now it is going up to some metafops and many petabytes of secondary storage.

MPI on MOC: Non Scalable parts of the MPI standard consists of irregular collectives and some array arguments of size which is proportional to the total number of processes.In order to obtain scalable performance the MPI should implementations mayneed topology aware or depend on global collective acceleration support.

Hardware and Architecture challenges:

Exa-scale computing: Post Dennard Scaling gives us the idea for shrinking transistors and yielding smaller circuits while having the same power density. Decreasing a transistor’s linear size by a factor of two thus reduced power by a factor of four, or with both

voltage and current halving.Superpipelining,Scoreboarding,Vectorization,and Parallelization must be balanced against their energy consumption.

MPI on MOC: One-sided operations implemented on architectures that offer remote direct memory access (RDMA). The hardware architecture is developed in a way that can make the processors talk with each other such that it can support mpi to transfer messages faster between millions of cores.

Cited: Reed, Daniel & Dongarra, Jack. (2015). Exascale Computing and Big Data. Communications of the ACM. 58. 56-68. 10.1145/2699414.