

Mahesh Wosti

CSIS 492 Senior Seminar

Dr. Andrew Chen

4/20/2021

## Parallel Computing- Pros and Cons explained

### **Abstract:**

Parallel computing has always fascinated me since I became aware of it. I think it is the only path forward if we are to build and make a super-fast computer in days to come. For my senior seminar, I want to build a simple framework of the parallel system and dig deeper into potential benefits and drawbacks of using a parallel system compared to the serial computing that is still pervasive in today's modern technology

### **Background:**

Parallel computing can be simply defined as running more than two programs or instructions at a time. In order to understand the parallel system, it is very crucial for us to understand what serial computing is. In serial computing, a problem is divided into instructions and each instruction is executed one after other sequentially in the processor. To understand parallel computation, it is critical to understand why the multi-tasking we perform in our modern computers is not a parallel rather sequential processing. In the modern computers we use, the speed is really fast and processor switch between processes so rapidly that we do not realize that things are happening sequentially. With such a fast speed of processor to switch back and forth between the processes, we feel like our computer is multitasking, equally running both program at same time. But, in reality, only one process gets executed at a time.

Now, as we understand how sequential computation work and how multi-processing exists in sequential computation, it's time to understand how parallel computation work. In parallel computation, a problem is broken down into different tasks. Each task is then broken down into small instructions and those instructions from different tasks can be executed simultaneously at the same time. As several things are running at the same time, we also need certain control and coordination mechanism to combine the result of each processes into a final result and we implement particular mechanism called MPI which stands for Message Passing Interface.

### **Requirements:**

Here are some of the requirements that need to be fulfilled to make sure parallel computation indeed is the solution to our problem.

1. We should be easily able to split the problem into small task that can be executed independently.
2. We should be able to perform two or more tasks simultaneously in the solution.
3. We should be able to execute multiple instructions at any moment in time.
4. The solution should take less time than traditional computation paradigm.

The physical resource that are required for implementing parallel computation technique is either multiple cores processors or group of computer connected by network in such a way that compute resources can be shared.

### **Importance of Parallel Computing:**

Now that we have got the basic insights about what parallel computation is, let's discuss why parallel computing is important to us in this modern world.

- Higher Speed
- To understand the complex and dynamic world that we live in
- To understand the world where things are simultaneous rather than sequential

Parallel computation is proven to be faster. Example: Super computers extensively use parallel computation technique by making a networks of thousands and thousands of compute devices and sharing all their resources in the network. The fastest supercomputer that exists in today's world has the maximum speed of  $10^{18}$  calculations per second.

Talking about understanding the complex world, we can implement and test several variables at the same time using parallel system. In real world, several factors play a role for certain phenomenon to occur both natural and artificial. For example: weather, change of season, temperature changes simultaneously than sequentially. So, if we are to study these natural phenomenon in greater precision, we need suitable computation tool to do so and parallel computation plays a role to meet this greater precision role. Parallel computation not only increases speed but it also helps to correlate several variables that are affecting the phenomenon at the same.

### **Pros of Parallel Computation:**

Now, let's talk about some of the advantages of parallel computation.

- Greater Precision
- Less Processing time – Higher Speed
- Less Processing Cost
- Solve complex problem of parallel world
- Full Utilization of compute resources
- Provides concurrency

As already discussed earlier in the paper, the precision is one of the most important benefits of parallel computation. Parallel computation makes the compute resource more scalable and we can break the boundary of memory and the processors to solve a particular problem. So, better the compute resource, enough the compute resource, better the precision will be.

Higher speed certainly reduces the time to perform long calculations which otherwise would have been super long if sequential processes were to be followed. Further, splitting and doing two things at the same time certainly makes things faster.

When we talk about big calculations, certainly the cost of computation would reduce if we are to deploy parallel computation than sequential computation. Big calculation requires higher

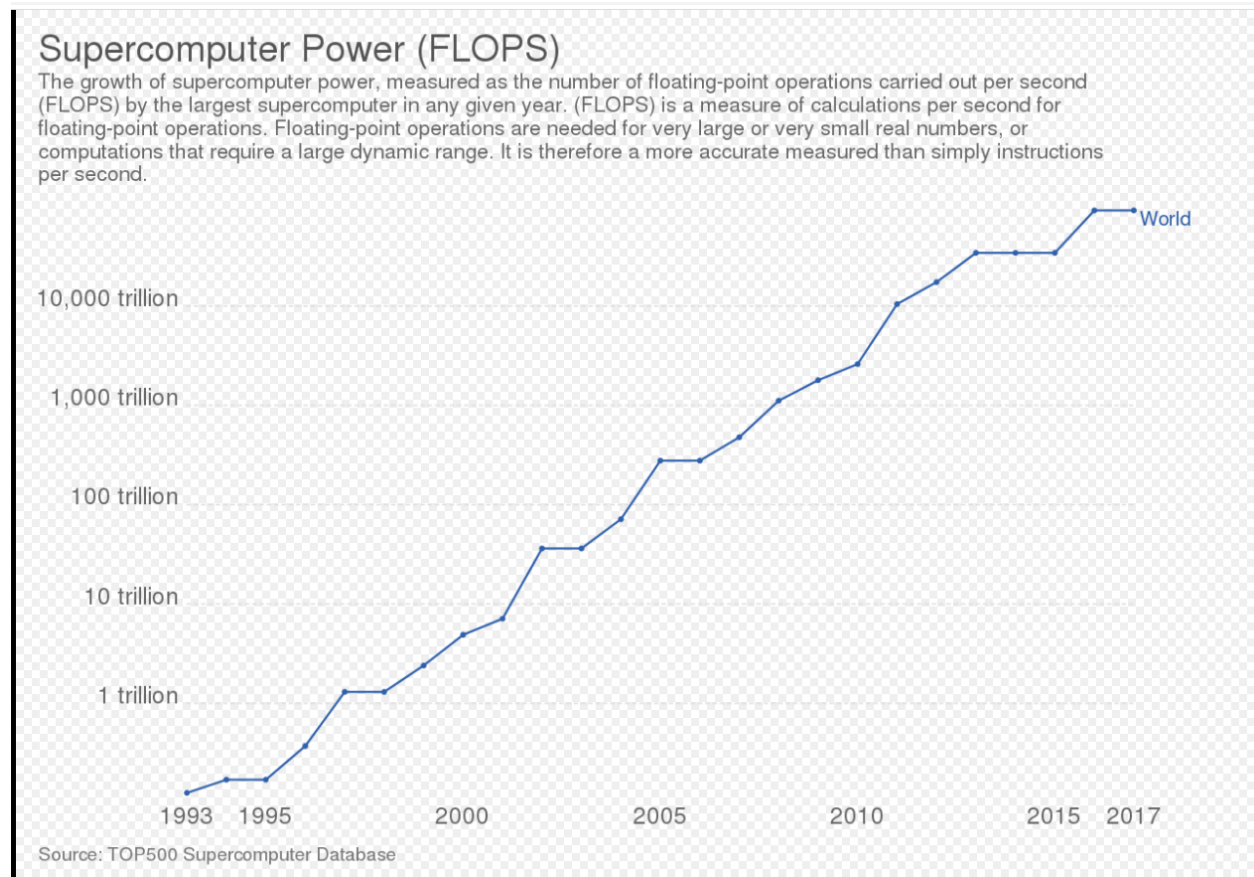
processing time, higher labor hours if we are to use sequential computation and we cannot guarantee precision with limitation in compute resources. However, parallel computation can solve all these problem.

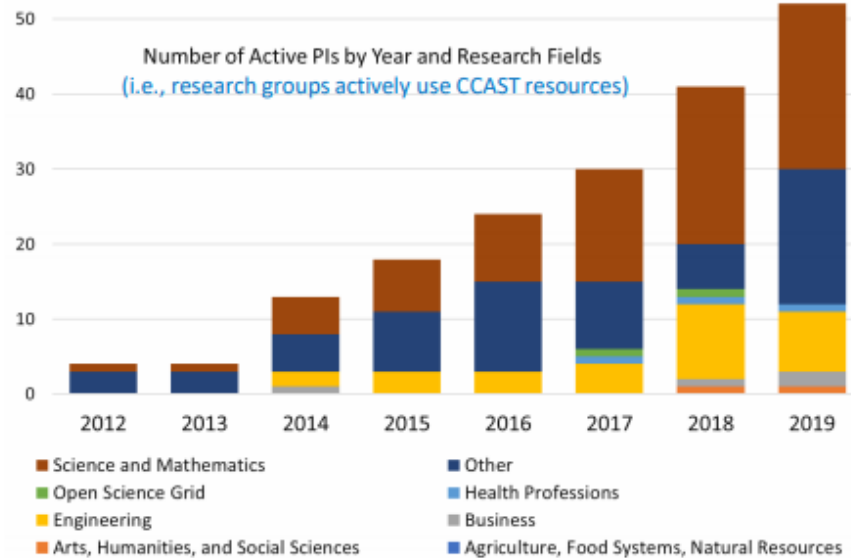
Talking about solving complex problem, we already discussed it in the importance section on how parallel computation is best fit for understanding and researching the parallel world that we live in.

Talking about full utilization of compute resources, it is a matter of fact that the hardware we have are designed parallel and they are being wasted with the sequential form of computation that still predominates the computing industry. If we can implement certain form of parallel computation, it will certainly help reduce the wastage of those resources and when things are shared, they are proper utilized which happens in the parallel computation.

Concurrency is all about ability to do more than two things at the same time which is certainly possible with parallel computation. This is useful to understand the correlation between several variables that are all affecting the particular phenomenon and concurrency is what comes to play.

The graph below shows the growth of speed of super computer aka parallel computation over the period of time.





As of 1/2021, CCAST has >120 PIs (=research groups) and >400 users in total

Source: K. Hoang CCAST NDSU

### Cons of Parallel Computation:

Now that we have discussed about the advantages of parallel system, it is time to discuss what some of the potential disadvantages of parallel computation are. Let's discuss the limitation of parallel computations from following components.

- Design
- Coding
- Debugging
- Tuning
- Maintenance

Talking about the Design, it is very difficult to implement and design a parallel computation sometimes depending on the nature of problems. Some problems are very difficult to parallelize. Sometimes even impossible. We should really be considering what percent of problems can be parallelized. If we cannot parallelize the entire problem, it might sometimes be worthless solving the problem using parallel computation as we might not be able to achieve the greater speed we aimed compared to sequential programming.

There is another design process problem on how we will make the processes to communicate with one other. Communication between the processes is one of the two extra things we need to implement if we are to implement parallel computation instead of sequential computation. In parallel computation, more than two processes are happening at the same time and sometimes these processes need to communicate with one other, exchange some variables values with one

other to complete the task and bring it to end results. This thing might get complicated and sometimes can prevent the programmer from achieving the accuracy and speed.

Similarly, synchronization between the processes is another important design problem. How would we set up different processes and synchronize them so that we reach the final results without any errors and with aimed speed?

Talking about coding, as mentioned earlier that we need to implement extra things such as communication and synchronization in the program, the coding we do will be completely different in the parallel computing. We need extra work which requires extra time and the labor cost on things like implementing the communication and synchronization pattern in the code.

Since we implement all these extra stuff in our traditional way of coding, it might be difficult for us to debug and find the errors in the code. We know a lot of things are happening in the background and we are using a lot of processors. We still do not have very effective debugging tools and technology that meets the need of debugging parallel programs.

Similarly, there might be tuning complexity on trying to balance the perfect synchronization between the processes to achieve the highest possible speed. There are certain parallel overheads involved in the parallel computation like time for communication and synchronization. If these internal processes are taking more time and is affecting the total process time of the problem, we might not achieve our goal of speed compared to sequential computation. So, we need to find a perfect balance on trying to limit the time and resources on internal things required to implement parallel computation to achieve the greater speed.

Likewise, it is very clear to us now that it would certainly be difficult for us to maintain these complex systems if we are facing the difficulty from the first step.

Here is an example of how programs look like in two computation systems for simple hello world.

### Parallel computation

```
1 #include <stdio.h>
2 #include <mpi.h>
3
4 main (int argc, char *argv[])
5 {
6
7     int myId;
8
9     MPI_Init(&argc,&argv);
10
11     MPI_Comm_rank(MPI_COMM_WORLD,&myId);
12     printf("Hello World %i \n",myId);
13     MPI_Finalize();
14
15     return 0;
16 }
```

### Serial Computation

```
1 #include <stdio.h>
2 int main(){
3
4
5 printf("Hello World !");
6
7 return 0;
8
9
10 }
11 |
```

## **MPI- Message Passing Interface:**

Now, that we have discussed about communication and synchronization in detail in previous part, it's time to understand how these things are done in parallel computation. So, we have a library called MPI which stands for Message Passing Interface which helps to achieve the communication between the processes. There are popular libraries of MPI in several programming languages and MPICH, MPI4PY, OPEN MPI are some examples.

Here are some of the basic methods of MPI

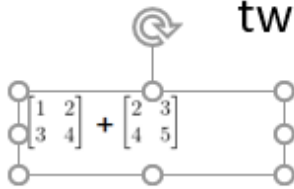
- MPI\_Init – Initialize MPI
- MPI\_Comm\_size – How many Processors are there?
- MPI\_Comm\_rank – My process number
- MPI\_Send – Send a message
- MPI\_Recv – Receive a message
- MPI\_Finalize – Close MPI universe

## **Concept of Master and Worker:**

In mpi world, there is the concept of Master and Worker. Master is the node that takes the user instruction, splits it into different small task and send it to the workers and also receives the end results from each of its worker. So, we can basically say master as a supervisor node.

On the other hand, Workers are individual nodes that receive the task from the Master, complete the task assign and send the result back to Master.

## Simple Parallel program to calculate sum of two matrix(2\*2) matrix



```
element in A's position 0, 0 is 1.00
element in B's position 0, 0 is 2.00
*****
element in A's position 0, 1 is 2.00
element in B's position 0, 1 is 3.00
*****
element in A's position 1, 0 is 3.00
element in B's position 1, 0 is 4.00
*****
element in A's position 1, 1 is 4.00
element in B's position 1, 1 is 5.00
```

```
maahesh@maahesh-virtual-machine:~/Desktop$ mpicc sum.c -o sum
maahesh@maahesh-virtual-machine:~/Desktop$ mpirun -np 2 ./sum
```

```
Done .
Recieved results from task 1
Result of the sum
3.00 5.00
7.00 9.00 *****done*****
maahesh@maahesh-virtual-machine:~/Desktop$
```

Parallel computation using mpicc library to print sum of two matrices

### Reflection on the project:

After a semester of research on parallel computation and its related components, I have realized that parallel computers are really good when it comes to speed and solving problems of parallel world. I also become aware that it has already helped in several modern scientific inventions of today's modern world and it has been implemented in lots of industries and field. The coolest thing about parallel computer or super computer is its speed. The fastest computer we have has the speed of  $10^{18}$  calculations per sec which is a lot.

Talking about some cons, I think its super complicated to design the parallel program and implement that in code to solve a particular problem. I think it requires a lot of home work on how to design the solution of problem in such a way that we can combine the speed of parallel computation and achieve the best possible result. Also, I think it is still quite costly to implement parallel system to be used in personal homes and to solve normal problems. Parallel computer is better for complex problems. Also, I realized the workforce in parallel computation industries is really scarce. The report from ACM states that 93% of HPC centers still have the difficulty hiring the staff with requisite skills.

I also had some personal thoughts on some ethical issues revolving around the use of parallel computation. The higher speed of parallel computation can replace the labor force and also be used in some destructive jobs like military. We do not know if the beauty of parallel computation will only be used in solving the real-world problems. There are potentials that it can be used to create further problems in the world. So, I think ethical part of the usage should also be equally considered by the engineers and people involved in this industry to get the maximum benefit from this wonderful technology of Parallel computation.

#### Works Cited:

1. Mpich.org. 2021. MPICH | High-Performance Portable MPI. [online] Available at: [Accessed 3 February 2021].
2. Mcs.anl.gov. 2021. Message Passing Interface. [online] Available at: [Accessed 3 February 2021]
3. Hpc.llnl.gov. 2021. Introduction to Parallel Computing Tutorial | High Performance Computing. [online] Available at: [Accessed 3 February 2021]
4. Vishkin, U., 1996. Can parallel algorithms enhance serial implementation? *Communications of the ACM*, 39(9), pp.88-91.
5. Charcranoon, S., Robertazzi, T. and Luryi, S., 2000. Parallel processor configuration design with processing/transmission costs. *IEEE Transactions on Computers*, 49(9), pp.987-991.
6. Yang, X. et al.(2021) 'Parallel Computing for Efficient and Intelligent Industrial Internet of Health Things: An Overview', *Complexity*,pp. 1-11. doi: 10.1155/2021/6636898
7. Sharma, G., Martin, J. MATLAB®: A Language for Parallel Computing. *Int J Parallel Prog* 37, 3–36 (2009). <https://doi.org/10.1007/s10766-008-0082-5>
8. Qing Wu, Maksym Spiryagin, Colin Cole & Tim McSweeney (2020) Parallel computing in railway research, *International Journal of Rail Transportation*, 8:2, 111-134, DOI: 10.1080/23248378.2018.1553115.
9. F. Cicirelli, A. Giordano and C. Mastroianni, "Analysis of Global and Local Synchronization in Parallel Computing," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 988-1000, 1 May 2021, doi: 10.1109/TPDS.2020.3037469.
10. Cui, Y., Chen, Z., Li, L. et al. An efficient parallel computing strategy for the processing of large GNSS network datasets. *GPS Solut* 25, 36 (2021).
11. L. S. Nyland, J. F. Prins, A. Goldberg and P. H. Mills, "A design methodology for data-parallel applications," in *IEEE Transactions on Software Engineering*, vol. 26, no. 4, pp. 293-314, April 2000, doi: 10.1109/32.844491.
12. Marowka, A. On parallel software engineering education using python. *Educ Inf Technol* 23, 357–372 (2018). <https://doi-org.ezproxy.lib.ndsu.nodak.edu/10.1007/s10639-017-9607-0>
13. K. A. Frenkel, "Preparing a Parallel Programming Workforce," *ACM*, 22-Aug-2013. [Online]. Available: <https://cacm.acm.org/news/167104-preparing-a-parallel-programming-workforce/fulltext>. [Accessed: 18-Apr-2021].