

# COVID-19 Modeling with the Game of Life

Using CUDA and MPI

Daniel Ackermans

Computer Science

Rensselaer Polytechnic Institute

Troy, New York, USA

ackerd2@rpi.edu

## ABSTRACT

The focus of the current scientific community is finding a solution to the Covid-19 pandemic. With current estimates for a vaccine predicting a release date of anywhere from late March, 2021 to the following September, it is clear that we have a long time to go before we are able to truly protect ourselves from this virus. The key focus for the present is using every available resource to limit and predict the spread of the COVID-19 virus so that when the time comes for vaccination, there are no surprises or obstacles remaining to healing those who are afflicted, and protecting the rest. This paper presents a multi-faceted model, based on the Game of Life Cellular Automata Model by John Conway, to predict the spread of Covid-19 within the United States of America. Included are the ability to generate a world in differing patterns including improper and proper social distancing, single and distributed clustering, and random distribution, as well as to model spread by differing methods, including an algorithm based around a given chance of infection. Model efficiency is tested using both strong and weak scaling methods, along with an up to date dataset from the New York Times. Finally, a best fit model is proposed based on comparisons with actual data and reliability of statistics from trial to trial.

## I. INTRODUCTION

COVID-19, or the Coronavirus, was first seen in Wuhan, China in December, 2019 [2]. Since then, it has spread rapidly around the world, causing large scale lock downs and a disruption to the normal operations of everyday life. On March 11, 2020 the virus was officially declared a pandemic by the World Health Organization. [8] According to the CDC COVID-19 response team, in a report published on March 18, 2020, while around 81% of the cases seen were mild [4][8], the majority of deaths, around 80%, occurred in the elderly population aged 60 years of age or older. [8] This means that unlike previous pandemic pathogens we have seen, our older populations are at a much higher risk than the younger populations. One of the most crucial ways that people are currently working to combat spread is through limited travel and the practice of social distancing. Research on the best methods for implementations of such restrictions

is ongoing. While some research shows that a case reduction of up to 10% could be seen with the implementation of travel restrictions, the same research also notes that with testing rates just shy of 25%, it is hard to be certain of any concrete number and it is best to assume that travel restrictions are having a negligible effect on case spread. [11] Additionally, travel restrictions, regardless of strength or severity, rely on a similar reduction in transmissibility in order to be effective, as even a 90% travel reduction only delays epidemic spread for 2 weeks, with no apparent mitigation to case spread. [11]. The same principle applies through to social distancing, with an infection rate 2-3 time higher than is needed for natural flattening of the curve from any environmental factors, the effect of human imposed social distancing needs to more than halve transmission [4] in order to see a relief in cases anywhere in the near or hopeful future before a vaccine is found. The best hope that has been found is that of stringent lockdown procedures, beyond that of travel restrictions and social distancing. Such measures when imposed in Wuhan, China not only showed a 100% increase in disease doubling time, but such lockdown conditions have also been shown to help reduce the connection between air travel and spread of the virus. [9]

The research done within this study aims to create a model for COVID-19 spread and death rates which matches the current situation as it is unfolding within the United States of America. The goal of these findings is to be able to adapt such a model to not only better predict future case spread, but to be able to make suggestions of the ways case spread can be further reduced without an increase to loss of life. It is expected that such a model will follow a statistic based approach to spread and be similar to the social distancing currently being recommended by officials around the country and the world. This paper builds on an already existing model of life and death prediction, and expands on its basic principles for use with a human life based simulation and prolonged effects of cell change.

## II. ALGORITHMIC IMPLEMENTATION

The codebase for this study is written in C using both the MPI message passing & I/O functionality, as well

as CUDA libraries which utilize the available GPU nodes at the RPI Center for Computational Innovations in order to run multiple world simulations in parallel. All experiments were completed using up to three of the 252 nodes on the dcs AiMOS supercomputer. Each node contains an available 6 NVIDIA Tesla v100 GPUs with 32 GiB each of memory, 2 IBM Power 9 processors clocked at 3.15 GHz, and 512 GiB of RAM. [1] Following each simulation day, a comparison is run between these worlds to determine each worlds' increase in cases from the previous day. At the conclusion of the experiment, these numbers were compared to the expected results seen in the New York Times dataset [3] to determine accuracy for each of the relevant data points: initialization and spread pattern, spread rate and death rate. Strong and weak scaling studies were also conducted using a variance in world size and node rank count to determine the effective scaling results. The outcome of all of these tests is presented within the Results sections of this paper.

### A. Game of life (GOL)

The parallel algorithm used for this study is rooted in the Game of Life (GOL) Cellular Automata model by John Conway, in which there are three main rules: Survival, where a living cell with 2 or 3 neighboring (alive) cells survives; Death, where a similar cell with greater than 4 or fewer than 2 neighbors dies from over-population or isolation respectively, and Birth, where a non-living cell, with 3 living neighbors, is brought to life. [6] GOL is built around a grid of cells, each with a starting value of 0 or 1, representing death or life respectively. In order to accurately represent the presence and spread of the coronavirus, this model expands that idea to a world where each cell contains a value in the range of 0 to 21, depending on the expected remaining length of infection for the patient at the given cell. Cells are populated based on a set of statistics published on GitHub by the New York Times [3]. This data lists the number of total COVID-19 cases and total COVID-19 related deaths in the United States through that day. (Figure 1). The data for the last  $n+1$  days, where  $n$  is equal to the number of simulation days, is used for comparison with result data, and the data from day 0 of the confined data set is used to seed the number of cases and deaths at the start of day 0. Each simulation begins with the input of 2 world patterns (initialization and spread), 2 disease rates (spread and death), the size of the simulation world, number of simulation days, number of threads allocated to each program rank, and whether the final state of the world should be output to file. Figure 2 shows the output from a run with configuration `2 0 3 3 4096 7 128 1`, and Figure 3 shows a breakdown of the input arguments and their ranges. Upon each iteration (day) of the simulation, the grid is transformed by the algorithm specified by the second position of the input configuration. Pattern 0 resembles the basic GOL algorithm, pattern 1 uses the given infection rate (position 3 argument), to determine spread to uninfected cells, pattern 2 simulates a

worst case scenario where anyone with at least 2 infected persons around them contracts the disease, and pattern 3 simulates a best case scenario, where at least  $\frac{3}{4}$  of the people surrounding the individual must be infected for the person to contract the disease.

2020-04-18	728094	34726
2020-04-19	753314	36109
2020-04-20	780449	37795
2020-04-21	805771	40316
2020-04-22	834338	42495
2020-04-23	868019	44516
2020-04-24	904604	46503
2020-04-25	938750	48325
2020-04-26	965214	49454
2020-04-27	988143	50819
2020-04-28	1012572	53034
2020-04-29	1039166	55399
2020-04-30	1069559	57570
2020-05-01	1103544	59392
2020-05-02	1133083	61077

**Figure 1: Statistic from the us.csv showing the total cases and total deaths up through the shown date. Data in image ranges from 4/19 – 5/02.**

### B. MPI

Message Passing Interface (MPI) was used in this study to pass the results of each iteration to each of the other ranks in the study, along with outputting them for checkpointing along the way. Following each iteration, the results of the previous iteration were read in, and compared to those of the current generation of ranks. This allows for a comparison and averaging of results for a final average comparison with the statistical numbers from the New York Times. Message passing between ranks took place using `MPI_IRecv` and `MPI_Isend`, where each rank sends its data, an array of unsigned char objects to each other rank, and receives the data sent from them. Parallel I/O using MPI was achieved with the functionality in `MPI_File_write_at` and `MPI_File_read_at`, with each rank writing out its current grid to file, and reading in its previous grid for comparison. Timings of these actions were tested and are reported later. Additionally, interaction between the Graphics Processing Units (GPUs) available at the CCI, and CUDA code base, was facilitated through the use of MPI ranks, where each rank was run by an available GPU on the DCS partition of the AiMOS supercomputer. Testing ranged from the use of 1 to 12 ranks, each run in groups of up to 6 per compute node and each on a dedicated GPU.



**Figure 2: The non-zero output from an example run of the implemented algorithm, numbers shown are days left of infection for persons with COVID-19.**

### C. CUDA

The parallel multithreading of this study was done using NVIDIA's CUDA libraries, specifically the curand random number generator library, cuda runtime library, and general GPU interaction functionalities. Each MPI rank was assigned its own cuda device, and individual thread blocks, each containing 128 threads throughout the study. Memory on each device/GPU was separately allocated using cudaMallocManaged so that it was able to also be read at any time by the CPU, and each was assigned a distinct curand random number generator for initializing its simulation world. The CUDA kernel used consists of 2 parts, an initializer, which ran the generalized Game of Life simulation and swapped the current data with the resultant data following its return, along with facilitating all MPI message passing and parallel I/O, and the main kernel. The main kernel split the task of iterating through the world amongst its assigned 128 threads, with each thread computing the next state of a singular cell in the world based on its neighbors, and assigned spread pattern algorithm, before moving to the next cell. Within both the kernel, and initialization sections of the cuda codebase, curand random number generators were used to determine the next status of a cell or beginning days left of infection for a cell respectively. Implementation style and methods for both of these generators were drawn directly from the NVIDIA CUDA Toolkit Documentation. [7]

#### D. Infection spread

Across all platforms used for the implementation of the algorithms and codebases used for these experiments, the overarching goal was to achieve an accurate model for predicting the spread and death rates of the coronavirus. The studies done and results presented assume knowledge of the status and health condition of every individual. Additional simplifications made in these experiments include an assumed static population size and by extension a 1-to-1 replacement rate, meaning for every death, another person is

simultaneously born to take that person's place in the world. As previously presented in Figure 3, the world is initialized to a given disease state  $i$  days before the current date, with a seed number of diseases coming from the New York Times data set as well as a starting number of deaths. These cases are spread around the grid based on the pattern 1 input, before starting the simulation at day 0. Case spread is modeled using the method specified by input pattern 2, these include an algorithmic control based around the spread model of the Game of Life model used to build this experiment, a statistic based model which uses the user supplied case spread rate to spread the disease, and two polar opposite models, one for a worst case spread scenario and one for a best case spread scenario. Each of these algorithms bases its determination on two factors, the current state of the given person (infected or healthy) and the number of infected people in the 8 cells immediately surrounding the infected person. This number of infected people is defined as the number of people who predicted to be at least 1, but no more than 16 days away from being healthy. While the maximum seed value for a given cell is 21, the virus has been predicted to have an incubation period of about 5–6 days, [4] and therefore, in this model, no infection can occur from a given person until at least 5 days after they have become infected. These implementations were run in varying configurations to determine the best fit scenario for modeling the spread of COVID-19.

### III. EXPERIMENTAL RESULTS

In order to measure the performance and reliability of the kernel and Parallel MPI I/O being implemented, scaling studies were run across the system's available resources. Experiments were run on configurations from a size of 1 rank, up to a size of 18 ranks taking up the capacity of 3 complete nodes. It should be noted that due to algorithm design, Parallel file I/O was disabled for the strong and weak scaling studies, but enabled for the I/O studies. Additionally, due to issues within the code, experiments were not able to be run on 3 or 5 rank configurations, as noted in the data for the Parallel I/O studies. The results of these experiments are as follows.

```
##Compile line arguments
> slurm*.sh:
2 1 3 3 300000 7 128 0
^ ^ ^ ^ ^      ^   ^   ^
| | | | |       | |   |
| | | | |       | |   output(on/off) (0 or 1)
| | | | |       | |   number of threads (1-128)
| | | | |       | |   iterations (> 0)
| | | | |       | |   sim. world size ( < 2,000,000)
| | | | |       | |   death rate (0-100)
| | | | |       | |   infection rate (0-100)
| | | | |       | |   spread pattern (0-3)
| | | | |       | |   initialization pattern (0-4)
```

**Figure 3: Command line arguments to the main algorithm with description of the range and purpose of each.**

### A. Strong scaling

The strong scaling study was conducted across 4 node/rank arrangements ([Chart 1](#)), each one with half the number of people per rank as the previous, but the same number of total people/cells in the simulation. Due to the nature of the simulation to work solely on square worlds, the scaling rates are adjusted to be accurate at scale, even if the resultant world size is slightly reduced after the implicit rounding caused taking a square root of an integer. The results of this strong scaling study ([Chart 2](#)), plotted on exponential x-axis ([Graph 1](#)), show an exponential increase in time as the individual world size grows, with a single rank run time of 0.226 seconds, while producing an 8 rank run time of 1.365 seconds, both with 262,144 total people in the world.

Strong scaling arrangement		
Overall world size	People/rank	Node arrangement
262,144 people	32,768 people	2 nodes, 8 ranks
262,144 people	65,536 people	1 node, 4 ranks
262,144 people	131,072 people	1 node, 2 ranks
262,144 people	262,144 people	1 node, 1 rank

Strong Scaling	
Ranks	Time (s)
1 rank	0.226
2 ranks	0.337
4 ranks	0.609
8 ranks	1.365

**Chart 1 & 2: The test structure and final results of the strong scaling experiment (Graph located on back page)**

### B. Weak scaling

The weak scaling study, also conducted across 4 node/rank arrangements ([Chart 3](#)), consisted of the same number of people per rank across all runs, regardless of node or rank count. Each rank of this study contained 299,209 people, with an eventual complete world size of 2,393,672 people across all ranks in the 2 node, 8 total rank set up. The results of this study ([Chart 4](#)) shows a consistent time of around 5.271 seconds across all tests ([Graph 2](#)) regardless of rank. This result has been attributed to the fact that each world is run in parallel with little interaction between ranks, because of this, an increase in rank count should not affect the overall performance of the given test.

Weak scaling arrangement		
Overall world size	People/rank	Node arrangement
299,209 people	299,209 people	1 nodes, 1 rank
598,418 people	299,209 people	2 nodes, 1 rank
1,196,836 people	299,209 people	4 nodes, 1 rank
2,393,672 people	299,209 people	8 nodes, 2 ranks

Weak Scaling	
Ranks	Time(s)
1 rank	5.245
2 ranks	5.256
4 ranks	5.286
8 ranks	5.295

**Chart 3 & 4: The test structure and final results of the weak scaling experiment (Graph located on back page)**

### C. Parallel I/O scaling

In addition to general strong and weak scaling studies, a study of the MPI parallel I/O scaling was also conducted to measure the performance impact of a growing world size. Data for overall program runtime ([Chart 5](#)) was collected and plotted ([Graph 3](#)) to show the results of this study. The study was conducted with the same basic input arguments, 2 1 3 3 300000 7 128, for all experimental runs. This resulted in a trend of linear increase, that is as the number of nodes increased, so did total run time. This trend is as expected as a large grid size would take more time to write to, and read back from the file. This conclusion is further supported by a previous finding, in the weak scaling study, that a doubling of the number of nodes, given the same input arguments, does not increase run time by any significant amount (increase is around a 100<sup>th</sup> of a second). Therefore, all increase seen within this study is due to an increase in parallel file I/O time. Additional data was collected for the individual read and write times for each rank configuration. These can be seen in [Graph 4](#) and [Chart 6](#).



Parallel IO scaling	
Ranks	Time (s)
1 rank	17.921
2 ranks	51.151
3 ranks	N/A
4 ranks	120.832
5 ranks	N/A
6 ranks	181.563
12 ranks	379.133
18 ranks	583.924

Parallel I/O tick time		
Configuration	Read time (s)	Write time (s)
1 rank	1	1
2 ranks	2.5	4
3 ranks	N/A	N/A
4 ranks	6	9
5 ranks	N/A	N/A
6 ranks	10.5	13.5
12 ranks	22.625	28.1
18 ranks	33.11	43.36

**Chart 5 & 6: The results of the Parallel I/O scaling test for both overall time and read/write tick time (Graph located on back page)**

#### IV. SCIENTIFIC RESULTS

While calculating final experiment results pertaining to infectious spread, it was discovered that experimental initialization of world cases does not account specifically for the number of recovered cases, nor that of already deceased patients. Because of this it was decided that experiments showing a slight overestimate in the number of predicted cases would be ranked higher than those showing an underestimate. Experimental results were grouped into three subcategories, initialization pattern, spread pattern, and death/cases infection rate. While the most impactful and scientifically relevant of these categories is death/case infection rate, each category contributes a piece to the overall solution.

##### A. Initialization pattern

Initialization testing consisted of 5 distinct options. Proper social distancing consisted of infected individuals staying as far away from the next individual as possible, while improper social distancing consisted of them keeping

exactly one cell between each other centered at the middle of the world. Familial, or proper clustering assumed a family group of 4 infected individuals keeping the maximum distance from the next infected familial group, and improper clustering created one large square of infected individuals at the center of the grid. The final pattern used was random distribution, where infected individuals were dispersed randomly throughout the grid. Initialization tests were run with the statistics spread pattern, and a 3% rate for both spread and death. The results of this experiment do not give a lot of insight into their usefulness, but as the tests came out, the one large cluster model showed the least deaths, and the proper distancing model showed the closest to expected results for the number of resultant cases.

##### B. Spread pattern

Spread pattern algorithms consisted of 4 separate choices. The control pattern, modeled after the Game of Life spread algorithm, depends on the number of infectious cases around the given person: fewer than two cases and they are safe, fewer than four cases and they get a mild, 14 day illness, but greater than 4 cases and they get a full-blown 21 day case. The statistics based algorithm determines, for each infectious person surrounding the current healthy cell, if the randomly generated number is within the given spread chance, if so, the disease is passed on for the full 21 days. The final two algorithms are a mirror image of each other. For the worst case model, if greater than 1 person in the surrounding area is infectious, then the disease is passed on for the full 21 days, while for the best case model, if at least 2 of the surrounding people are healthy, then the disease is not passed on, otherwise it is passed on for a mild case of 14 days. Results of the spread, once adjusted for the offset between initialized cases and subtracting for deaths show a clear leader in the stats spread pattern. While this pattern was seeded with a 3% infection rate, the results produced are both within range of the expected results, and are not too low or extremely too high. The GOL based spread model performed the worst with final case count, at scale, of greater than 2,000 cases, while the expected outcome of a proper model would have been 970 cases.

##### C. Infection/Death rate

For the purposes of these experiments, and following a trial period of general ranges, a range of 1-5 was picked for both death and infection rate. Rates were tested with their matching rate being set at a control of 3 being that it was the middle value of both ranges. Prior to testing, these results were hypothesized to be the most revealing and helpful of all results collected by this study. Results from infection rate testing reveal that the tests run produced numbers below the expected results. Based on the initialization model (single clustering) used and its results in previous tests, the best case scenario is expected to be much higher than the given rate, possibly closer to 10-15%. The outcome of death rate testing showed inverse and very hopeful results. Death rate results,

along with results from all trials throughout, showed death results much higher than the expected number. For the majority of experiments, the expected final death count was 52, while the actual result was usually around 95 total deaths. From these, it is expected that the actual death rate lies somewhere below 1% of overall infections.

## V. RELATED WORK

There seems to be a certain universality in the time evolution of COVID-19 [2] among data collected from the spread in China and Italy. On the positive side of this fact is the finding that between quarantine, social distancing, and isolation of those who are infected, what has happened in China shows that the epidemic can be contained [4] when all rules and recommendations are properly followed. On the other side however, is the fact that an estimated 80% of people with COVID-19 have mild or asymptomatic disease and the disease can last for 10 days or more after the incubation/infectiousness period [4]. This makes it very hard to predict exactly who is carrying the disease and when, and could lead to a resurgence of cases if lockdowns aren't maintained for a long enough period of time.

Locational temperature and humidity were reported to be major factors in the spread of this disease. Not only are temperature and humidity known factors in SARS-CoV, MERS-CoV and influenza survival, but the areas most hard hit by this disease show a trend of having an average temperature between 5-11°C during the time of study (January - February) and/or the immediate future (March - April) [5]. Even with this data, the effect of season on COVID-19 transmission and spread is still an unknown, and with its current reproduction number being around a 2-3, and needing to be at unity ( $R_0 = 1$ ) for the epidemic to begin its decline, a 60% reduction would have to be seen in disease spread for this to occur. Because of this, it is possible that even the warm summer months won't be able to reduce transmission low enough to reach unity [4].

## VI. SUMMARY

After running numerous tests for modeling of disease spread many of the results found require more testing to be verified and strengthened before they can produce a solid model on their own. However, current social convention of quarantining in small, properly social distanced groups is the leader in methods for limiting the spread of this virus while keeping the death rate low. Additionally, the predicted death rate is lower than any currently being seen in the world. On the other side of that however, the predicted current spread rate of the virus is higher than any rate hypothesized by this study.

The algorithms used to implement this adaptation of the Game of Life show a proper trend upon both strong and weak scaling, and show a positive linear trend in I/O scaling with an increase in world number.

## VII. CONCLUSION

Further research is needed to conclude a proper model for the spread of COVID-19 within the United States of America. With the current model, a predicted best case model of an average 12 percent infection rate with a 1% death rate in a world with proper distancing produces results with cases in an acceptable range, but deaths 6 times higher than those predicted. Since a greater number of cases leads to a greater number of deaths, future work first needs to focus on correcting the case results to a rate commensurate with the expected cases before venturing into finding an appropriate death rate. Additionally, further work needs to be done into finding correct spread prediction algorithms either for public knowledge or accurate result representation. The initialization algorithms presented in this paper, while basic, are a good ground work to base further experiments off of with refining to exact placement of cases and allocation of cases to certain regions and chunks. Due to scaling issues with the used resources, all testing was run at a massively scaled down level from actual full scale testing. For full result conclusion and verification, testing with a world containing around 300 million persons/cells [10] is needed, along with a testing period of at least 4 weeks or 28 days, to allow for multiple full generations of the virus to run their course and produce more accurate results with less chance of outliers or unbalanced percentages. Takeaways from this research include general models for population configuration, a suggestion for more accurate infection/death rates, and a small scale testing platform for further modeling before a larger scale experiment.

Additional research on the topic should also take into consideration the age and health condition of a person, a dynamic infection rate based on preexisting conditions and the surrounding environment, the effect of multiple population make ups in different sectors of a world, the effects of testing on spread, and the effect that leaving dead cells on the board instead of replacing them with a new healthy cell has. The use of a class or struct object for the representation of persons in the grid is recommended as this would allow for multiple factors to influence how the virus spreads and how an individual practices social distancing.

## VIII. ACKNOWLEDGMENTS

Thank you to Dr. Christopher Carothers and Angelo Luna for your contributions to making this research successful and fruitful, and for your education and guidance throughout the past semester.

## IX. REFERENCES

- [1] Anonymous, 2020. DCS Supercomputer, [https://secure.cci.rpi.edu/wiki/index.php?title=DCS\\_Supercomputer](https://secure.cci.rpi.edu/wiki/index.php?title=DCS_Supercomputer).
- [2] Duccio Fanelli, Francesco Piazza. 2020. Predicate path expressions. In Proceedings of the 6th. Analysis and forecast of

COVID-19 spreading in China, Italy and France. Elsevier. DOI: <https://doi.org/10.1016/j.chaos.2020.109761>

[3] The New York Time. 2020. Coronavirus (Covid-19) Data in the United States. <https://github.com/nytimes/covid-19-data>.

[4] Roy Anderson Et. Al. 2020. How will country-based mitigation measures influence the course of the COVID-19 epidemic?. Elsevier. DOI: [https://doi.org/10.1016/S0140-6736\(20\)30567-5](https://doi.org/10.1016/S0140-6736(20)30567-5)

[5] Mohammad Sajadi Et. Al. March, 2020. Temperature, Humidity and Latitude Analysis to Predict Potential Spread and Seasonality for COVID-19. DOI: <https://dx.doi.org/10.2139/ssrn.3550308>

[6] Lorena Caballero, Bob Hodge, Sergio Hernandez. 2016. Conway's "Game of Life" and the Epigenetic Principle. Front. Cell. Infect. Microbiol. 6:57. DOI: <http://dx.doi.org/10.3389/fcimb.2016.00057>

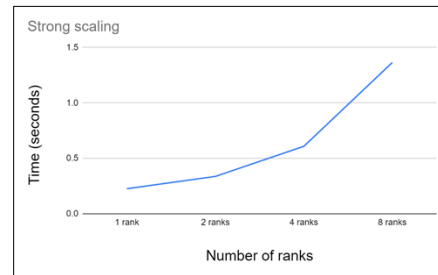
[7] NVIDIA Corporation. 2019. cuRAND, the CUDA random number generation library. <https://docs.nvidia.com/cuda/curand/index.html>

[8] Tamara Pilishvili, Et. Al. Severe Outcomes Among Patients with Coronavirus Disease 2019 (COVID-19) — United States, February 12–March 16, 2020: CDC COVID-19 Response Team. In Proceedings of MMWR Vol. 69 No.12 CDC, Atlanta, GA, USA, 4 pages. <https://www.cdc.gov/mmwr/volumes/69/wr/pdfs/mm6912e2-H.pdf>

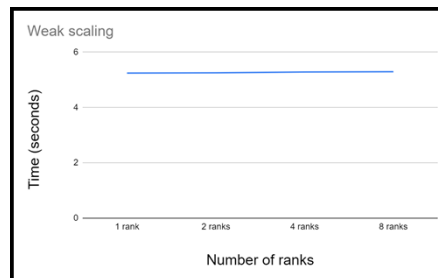
[9] Hien Lau Et. Al. 2020. The positive impact of lockdown in Wuhan on containing the COVID-19 outbreak in China. In Journal of Travel Medicine. DOI: <https://doi.org/10.1093/jtm/taaa037>

[10] United States Census Bureau. 2020. U.S. and World Population Clock. <https://www.census.gov/popclock/>

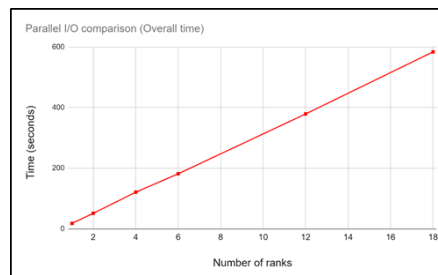
[11] M. Chinazzi Et Al. 2020. The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. the American Association for the Advancement of Science, New York, NY, USA, DOI: <https://doi.org/10.1126/science.aba9757>



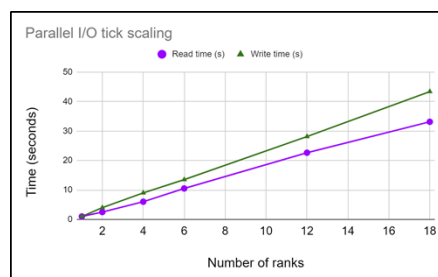
**Graph 1: The results graph from the Strong scaling experiment (Graph data from Chart 2)**



**Graph 2: The results graph from the Weak scaling experiment (Graph data from Chart 4)**



**Graph 3: The results graph from the Parallel I/O scaling experiment (Graph data from Chart 5)**



**Graph 4: The results graph from the Parallel I/O ticks scaling experiment (Graph data from Chart 6)**