Analysis of Mandelbrot Set Computation Speed Up:

With Static and Dynamic Load Partitioning

Bandith Phommounivong

Dr. Harris

CS 415

PA 02

3/9/2014

Introduction

In this project, we explore one example of parallelism, the calculation of the Mandelbrot set. This example is often called “Embarrassingly Parallel” because each data set can be computed independent of the rest of the data sets. “Embarrassingly Parallel” is more correctly described as naturally parallel. Ideally, problems like the Mandelbrot Set can be computed quicker with more processes, however, there are many issues such as communication between processes and the difference between computational speed that prevent this from happening to full potential.

Theory

Mandelbrot Set computation has a sequential time complexity of O(n). This is because the sequential computation time ts is a function of the number of pixels n and the number of iteration per pixel Max\_Iter. Given that Max\_Iter is some constant k, ts is therefore O(n).

To compare, the parallel algorithm we have to calculate the Speed Up factor.

Speed up factor = ts /tp

This relationship will show the performance of the parallel algorithm is relative to the sequential algorithm. However, this alone is is not enough.

To show that parallelism is worth the effort, the efficient has to also be calculated. The efficiency is:

Efficiency = Speed Up / number of processes

This will show how much speed up there was for each processor. This metric is important when judging whether additional processors will be beneficial.

Algorithm

Sequential

For (row = 0; row < number of rows; row++) {

For (col =0; col < number of columns; col++) {

Calculate real seed based on cols

Calculate imaginary seed on rows

Calculate the number of iteration using seeds

}

}

Static

Master

Remainder = number of rows mod number of slaves

For (row = 0; row < number of rows – remainder; row++) {

Recv(Row Number, Any)

Recv(Temporary Buffer, Row Number)

For (col = 0; col < number of columns; col++) {

Set column elements of Image Buffer to Temporary Buffer;

}

}

For (row = number of row - remainder; row < number of row; row++) {

For (col =0; col < number of columns; col++) {

Calculate real seed based on cols

Calculate imaginary seed on rows

Calculate the number of iteration using seeds

}

}

Slave

Row Size = number of rows to compute / number of slaves

Start = slave id \* Area

For (row = start; row < start + Row Size; row++){

Calculate real and imaginary seeds

Set temporary buffer of finished rows

}

Send the row number

Send row data

Parallel

Master

Initialize count and row to 0

For (k = 0; k < number of slaves; k++){

Send row number to slave k

Increment count and row

}

Do {

Recv row from any slave

If (row is less than display height) {

Send another row to the slave

Increment count and row

}

else

Send termination tag

}

} while count is greater than 0

Slave

Recv first row to compute

While (source tag is not termination tag){

Compute imaginary seed

For all elements in a row

Compute real seed

Compute pixel

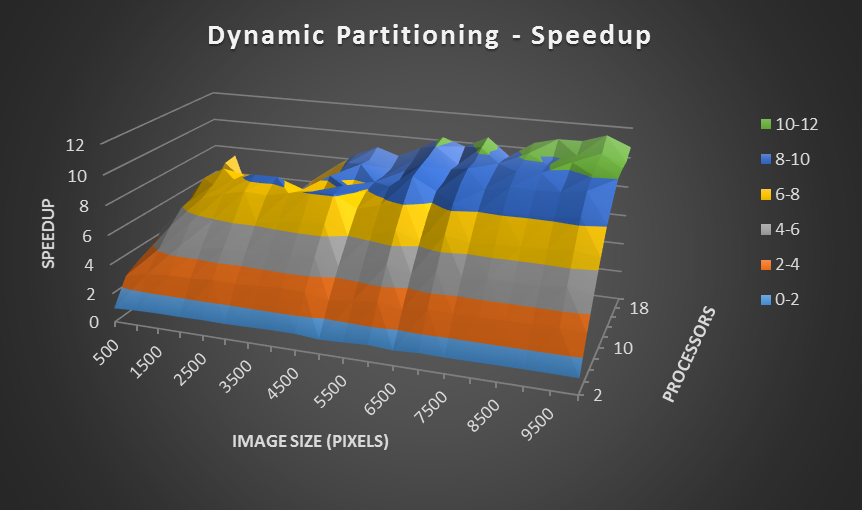
Send row back to master

Recv next row

}

Graphs and Data

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dynamic | | | | | | | | | | |
|  | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 500 | 0.9405554 | 2.5997859 | 3.0879872 | 3.4192594 | 4.240801 | 4.325858 | 4.696548 | 4.057707 | 4.246671 | 4.726947 |
| 1000 | 1.0185961 | 2.9131678 | 4.0794337 | 5.1311242 | 5.955883 | 6.207182 | 7.078497 | 6.298571 | 7.278476 | 7.397764 |
| 1500 | 1.0491848 | 3.0611741 | 4.5052305 | 5.6504946 | 6.790845 | 7.249863 | 7.940378 | 7.23699 | 2.954117 | 3.012781 |
| 2000 | 1.0408691 | 3.0429929 | 4.5745418 | 5.9334981 | 7.256914 | 7.79287 | 7.649179 | 5.148905 | 4.432512 | 3.981846 |
| 2500 | 1.0687746 | 3.1714129 | 4.7748349 | 6.1902935 | 7.611622 | 8.291047 | 5.230204 | 4.877526 | 4.951315 | 4.975792 |
| 3000 | 1.0456804 | 3.1073462 | 4.697663 | 6.1860287 | 7.510874 | 8.316377 | 5.945098 | 5.856164 | 5.888553 | 6.267181 |
| 3500 | 1.0727014 | 3.1973027 | 4.8358781 | 6.4035186 | 7.83238 | 7.462799 | 6.923669 | 6.837961 | 7.035667 | 7.380012 |
| 4000 | 1.1067271 | 3.2725095 | 4.980706 | 6.590531 | 8.108342 | 7.31229 | 8.227801 | 7.987607 | 8.213835 | 8.401342 |
| 4500 | 1.1052197 | 3.2282568 | 4.9759373 | 6.601191 | 8.109818 | 8.198058 | 7.77019 | 8.752165 | 8.583386 | 9.078462 |
| 5000 | 0.9665718 | 2.8986639 | 4.5318725 | 5.979367 | 7.313854 | 7.947613 | 8.139599 | 8.035584 | 8.101277 | 8.594473 |
| 5500 | 1.026295 | 3.0832308 | 4.7134415 | 6.3219463 | 7.717461 | 8.474787 | 8.198632 | 8.618064 | 8.881435 | 9.330932 |
| 6000 | 1.1055906 | 3.2645593 | 5.102973 | 6.7727743 | 8.199619 | 9.328625 | 9.568926 | 9.744236 | 10.01933 | 10.24426 |
| 6500 | 1.0018829 | 2.9790251 | 4.6795975 | 6.2289304 | 7.563593 | 8.49744 | 8.64872 | 9.011235 | 8.895141 | 9.728142 |
| 7000 | 1.0904978 | 3.2604035 | 5.0846324 | 6.8475101 | 8.347185 | 9.214717 | 9.714973 | 9.72608 | 10.22076 | 10.52612 |
| 7500 | 1.0592255 | 3.1428571 | 4.9068463 | 6.5447812 | 7.988131 | 8.963127 | 9.348914 | 9.734286 | 9.606497 | 9.283162 |
| 8000 | 1.0693815 | 3.2018362 | 5.0453248 | 6.6720554 | 8.179929 | 9.328886 | 9.828277 | 9.93474 | 10.13868 | 10.43043 |
| 8500 | 1.0876397 | 3.2083621 | 5.0205975 | 6.6667162 | 8.123893 | 9.261022 | 9.897913 | 9.960026 | 9.860878 | 10.82895 |
| 9000 | 1.0886183 | 3.2086527 | 5.0671176 | 6.7034469 | 8.192746 | 9.302311 | 9.648599 | 10.29604 | 10.27992 | 10.85025 |
| 9500 | 1.0833778 | 3.255925 | 5.1524313 | 6.8505803 | 8.413365 | 9.546188 | 10.07759 | 10.31812 | 10.58941 | 11.37656 |
| 10000 | 1.0645959 | 3.1308048 | 4.9134665 | 6.5943485 | 8.061955 | 9.075197 | 10.01342 | 10.1708 | 10.17239 | 10.72471 |



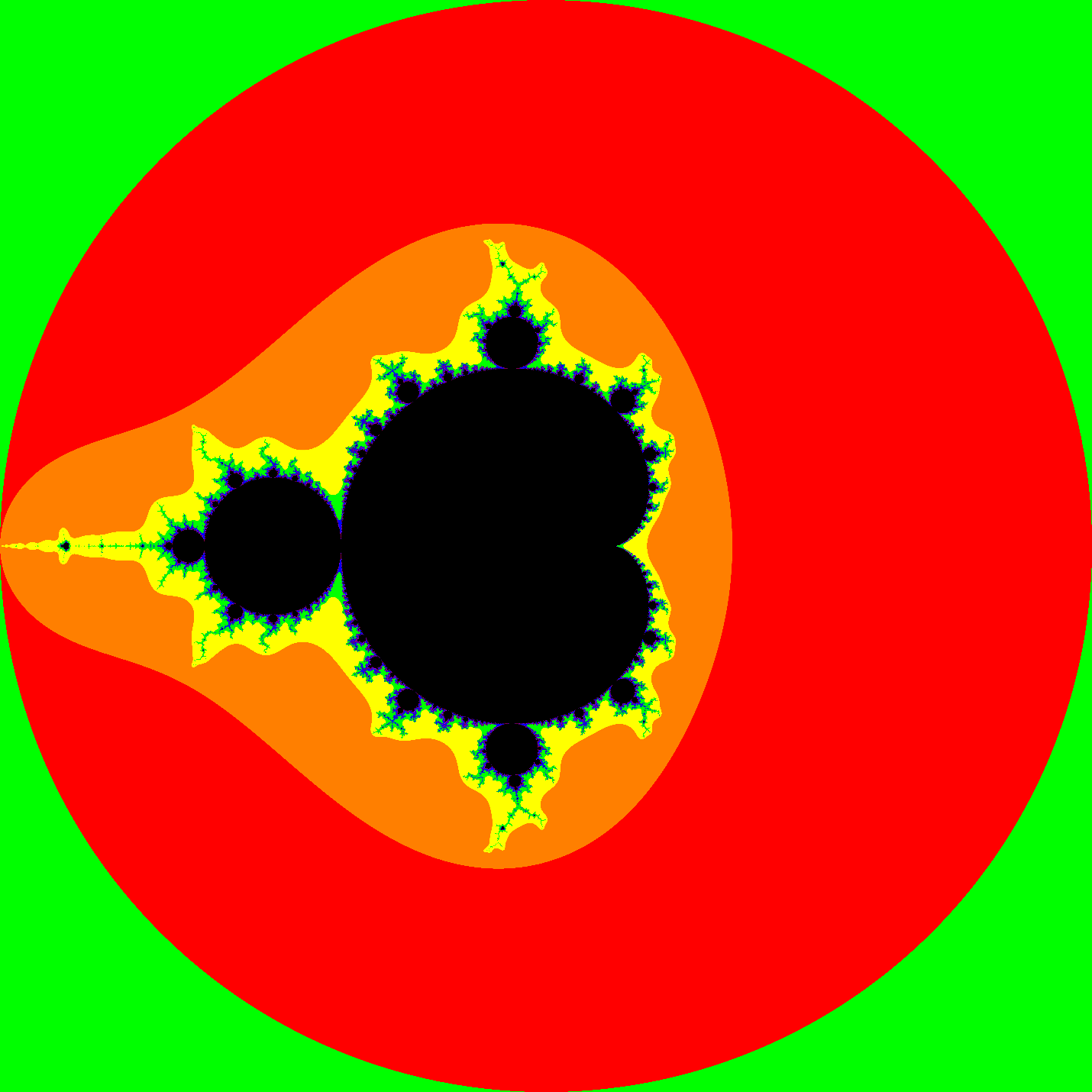
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Static Partitioning | | | | | | | | | | |
|  | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 500 | 0.8820693 | 1.0078708 | 1.434807 | 1.7560169 | 1.919402 | 2.383997 | 3.029063 | 2.615374 | 0.718276 | 0.274206 |
| 1000 | 0.9517663 | 1.1190007 | 1.5436521 | 1.8656351 | 2.242425 | 2.831674 | 3.333474 | 2.444287 | 1.9461 | 0.586243 |
| 1500 | 0.95176 | 1.1934198 | 1.5823271 | 1.9957586 | 2.361645 | 2.8719 | 3.399267 | 3.650197 | 3.411969 | 1.792857 |
| 2000 | 0.9566014 | 1.0897937 | 1.519096 | 1.9804798 | 2.370156 | 2.900547 | 3.385117 | 4.079227 | 2.437499 | 2.678949 |
| 2500 | 0.9882256 | 1.1401165 | 1.6530883 | 2.0044045 | 2.48397 | 2.993744 | 3.359077 | 4.053541 | 3.463306 | 2.280644 |
| 3000 | 0.9340097 | 1.0797894 | 1.4790202 | 2.0322286 | 2.414024 | 2.913085 | 3.482991 | 3.922858 | 4.165193 | 3.754955 |
| 3500 | 0.986683 | 1.1524049 | 1.5552635 | 2.0213683 | 2.43609 | 2.989694 | 3.470206 | 4.154522 | 4.384126 | 4.395317 |
| 4000 | 1.0017479 | 1.1596893 | 1.585628 | 2.0632351 | 2.54681 | 2.970289 | 3.571479 | 4.015357 | 4.637406 | 4.505139 |
| 4500 | 1.0306042 | 1.1337902 | 1.6187681 | 2.0226983 | 2.558841 | 3.081637 | 3.456601 | 4.050281 | 4.513214 | 4.703242 |
| 5000 | 0.9472912 | 1.0719235 | 1.5052214 | 1.8527143 | 2.223101 | 2.753658 | 3.190794 | 3.603685 | 4.173028 | 3.627963 |
| 5500 | 0.9658863 | 1.0605243 | 1.4899421 | 1.9915174 | 2.388706 | 2.955671 | 3.316448 | 3.757667 | 4.23751 | 4.417489 |
| 6000 | 1.0318023 | 1.1631465 | 1.6327276 | 2.0268909 | 2.556338 | 3.064401 | 3.649392 | 4.123652 | 4.623842 | 4.828285 |
| 6500 | 0.9176383 | 1.0345758 | 1.4504252 | 1.8898301 | 2.275149 | 2.751053 | 3.157298 | 3.86598 | 4.151693 | 4.054583 |
| 7000 | 1.0113125 | 1.1600849 | 1.5674982 | 2.128497 | 2.549591 | 3.06516 | 3.483365 | 4.027711 | 4.54408 | 4.539817 |
| 7500 | 0.9631659 | 1.1403258 | 1.5661203 | 1.9305334 | 2.37051 | 3.01713 | 3.314297 | 3.943605 | 4.46597 | 4.320037 |
| 8000 | 0.9676265 | 1.1467551 | 1.5543219 | 2.0287972 | 2.4916 | 3.019106 | 3.47144 | 4.045847 | 4.798002 | 4.442673 |
| 8500 | 0.9849447 | 1.1367903 | 1.555254 | 2.0305936 | 2.441351 | 3.022989 | 3.393088 | 3.934545 | 4.698115 | 4.697509 |
| 9000 | 0.9887106 | 1.1435862 | 1.5284503 | 2.0310924 | 2.492656 | 3.036865 | 3.678845 | 4.162773 | 4.449722 | 4.713082 |
| 9500 | 1.0521276 | 1.1968788 | 1.6409259 | 2.061149 | 2.627171 | 3.255678 | 3.563526 | 4.21231 | 4.646204 | 4.733136 |
| 10000 | 0.9697281 | 1.1163514 | 1.54472 | 1.9826556 | 2.386608 | 2.855013 | 3.489184 | 4.090417 | 4.457901 | 4.448595 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Efficiency - Dynamic | | | | | | | | | | |
|  | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 500 | 47.02777 | 64.994648 | 51.466453 | 42.740742 | 42.40801 | 36.04881 | 33.54677 | 25.36067 | 23.59262 | 23.63473 |
| 1000 | 50.929805 | 72.829194 | 67.990561 | 64.139052 | 59.55883 | 51.72652 | 50.5607 | 39.36607 | 40.43598 | 36.98882 |
| 1500 | 52.459242 | 76.529352 | 75.087175 | 70.631183 | 67.90845 | 60.41552 | 56.71698 | 45.23118 | 16.41176 | 15.0639 |
| 2000 | 52.043453 | 76.074822 | 76.242363 | 74.168727 | 72.56914 | 64.94058 | 54.63699 | 32.18066 | 24.62507 | 19.90923 |
| 2500 | 53.438728 | 79.285323 | 79.580582 | 77.378668 | 76.11622 | 69.09206 | 37.3586 | 30.48454 | 27.50731 | 24.87896 |
| 3000 | 52.284022 | 77.683656 | 78.294383 | 77.325358 | 75.10874 | 69.30315 | 42.46498 | 36.60102 | 32.71418 | 31.3359 |
| 3500 | 53.635071 | 79.932569 | 80.597969 | 80.043982 | 78.3238 | 62.18999 | 49.45478 | 42.73725 | 39.08704 | 36.90006 |
| 4000 | 55.336353 | 81.812737 | 83.011767 | 82.381637 | 81.08342 | 60.93575 | 58.77001 | 49.92255 | 45.63242 | 42.00671 |
| 4500 | 55.260987 | 80.706419 | 82.932288 | 82.514887 | 81.09818 | 68.31715 | 55.50136 | 54.70103 | 47.68548 | 45.39231 |
| 5000 | 48.328592 | 72.466596 | 75.531208 | 74.742087 | 73.13854 | 66.23011 | 58.13999 | 50.2224 | 45.00709 | 42.97236 |
| 5500 | 51.314749 | 77.08077 | 78.557359 | 79.024329 | 77.17461 | 70.62322 | 58.56166 | 53.8629 | 49.3413 | 46.65466 |
| 6000 | 55.279528 | 81.613983 | 85.049551 | 84.659679 | 81.99619 | 77.73854 | 68.34947 | 60.90148 | 55.66294 | 51.2213 |
| 6500 | 50.094146 | 74.475627 | 77.993291 | 77.86163 | 75.63593 | 70.812 | 61.77657 | 56.32022 | 49.41745 | 48.64071 |
| 7000 | 54.524888 | 81.510088 | 84.743874 | 85.593876 | 83.47185 | 76.78931 | 69.39266 | 60.788 | 56.78199 | 52.63061 |
| 7500 | 52.961274 | 78.571429 | 81.780772 | 81.809764 | 79.88131 | 74.69273 | 66.77795 | 60.83929 | 53.36943 | 46.41581 |
| 8000 | 53.469075 | 80.045904 | 84.088746 | 83.400693 | 81.79929 | 77.74071 | 70.20198 | 62.09212 | 56.32602 | 52.15213 |
| 8500 | 54.381983 | 80.209053 | 83.676624 | 83.333952 | 81.23893 | 77.17518 | 70.69938 | 62.25016 | 54.78266 | 54.14475 |
| 9000 | 54.430914 | 80.216318 | 84.45196 | 83.793087 | 81.92746 | 77.51926 | 68.91856 | 64.35023 | 57.11066 | 54.25123 |
| 9500 | 54.168891 | 81.398126 | 85.873855 | 85.632253 | 84.13365 | 79.55157 | 71.98277 | 64.48824 | 58.83006 | 56.88282 |
| 10000 | 53.229795 | 78.27012 | 81.891109 | 82.429357 | 80.61955 | 75.62664 | 71.52441 | 63.5675 | 56.51329 | 53.62355 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Efficiency - Static | | | | | | | | | | |
|  | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 500 | 44.103463 | 25.19677 | 23.913449 | 21.950211 | 19.19402 | 19.86664 | 21.63616 | 16.34609 | 3.990425 | 1.371031 |
| 1000 | 47.588317 | 27.975016 | 25.727535 | 23.320439 | 22.42425 | 23.59729 | 23.81053 | 15.27679 | 10.81167 | 2.931213 |
| 1500 | 47.588001 | 29.835496 | 26.372119 | 24.946982 | 23.61645 | 23.9325 | 24.28048 | 22.81373 | 18.95538 | 8.964287 |
| 2000 | 47.830072 | 27.244843 | 25.318267 | 24.755997 | 23.70156 | 24.17122 | 24.17941 | 25.49517 | 13.54166 | 13.39474 |
| 2500 | 49.411278 | 28.502912 | 27.551472 | 25.055056 | 24.8397 | 24.94787 | 23.99341 | 25.33463 | 19.24059 | 11.40322 |
| 3000 | 46.700483 | 26.994736 | 24.650337 | 25.402858 | 24.14024 | 24.27571 | 24.87851 | 24.51787 | 23.13996 | 18.77477 |
| 3500 | 49.334149 | 28.810122 | 25.921059 | 25.267104 | 24.3609 | 24.91411 | 24.78719 | 25.96576 | 24.35626 | 21.97659 |
| 4000 | 50.087394 | 28.992232 | 26.427134 | 25.790439 | 25.4681 | 24.75241 | 25.51056 | 25.09598 | 25.76337 | 22.5257 |
| 4500 | 51.530209 | 28.344754 | 26.979468 | 25.283729 | 25.58841 | 25.68031 | 24.69 | 25.31426 | 25.07341 | 23.51621 |
| 5000 | 47.364559 | 26.798088 | 25.087023 | 23.158929 | 22.23101 | 22.94715 | 22.79138 | 22.52303 | 23.18349 | 18.13981 |
| 5500 | 48.294314 | 26.513109 | 24.832368 | 24.893967 | 23.88706 | 24.63059 | 23.68891 | 23.48542 | 23.54172 | 22.08745 |
| 6000 | 51.590114 | 29.078662 | 27.212126 | 25.336137 | 25.56338 | 25.53667 | 26.06708 | 25.77283 | 25.68801 | 24.14142 |
| 6500 | 45.881916 | 25.864395 | 24.173753 | 23.622877 | 22.75149 | 22.92544 | 22.55213 | 24.16238 | 23.06496 | 20.27292 |
| 7000 | 50.565625 | 29.002123 | 26.12497 | 26.606212 | 25.49591 | 25.543 | 24.88118 | 25.17319 | 25.24489 | 22.69908 |
| 7500 | 48.158294 | 28.508146 | 26.102004 | 24.131667 | 23.7051 | 25.14275 | 23.67355 | 24.64753 | 24.81094 | 21.60019 |
| 8000 | 48.381326 | 28.668878 | 25.905365 | 25.359965 | 24.916 | 25.15921 | 24.796 | 25.28654 | 26.65556 | 22.21337 |
| 8500 | 49.247236 | 28.419758 | 25.9209 | 25.38242 | 24.41351 | 25.19158 | 24.23635 | 24.5909 | 26.10064 | 23.48754 |
| 9000 | 49.435532 | 28.589655 | 25.474172 | 25.388655 | 24.92656 | 25.30721 | 26.27746 | 26.01733 | 24.72068 | 23.56541 |
| 9500 | 52.606378 | 29.921971 | 27.348765 | 25.764363 | 26.27171 | 27.13065 | 25.45376 | 26.32694 | 25.81224 | 23.66568 |
| 10000 | 48.486406 | 27.908784 | 25.745333 | 24.783195 | 23.86608 | 23.79177 | 24.92274 | 25.56511 | 24.76612 | 22.24297 |

Picture

Picture is color with 9 colors. Each selected based on what power of two, the number of iterations is less than.



Issues

The issues encountered in this projects can be divided into two types: Bash and Grid related or algorithm.

The particular algorithm issue encountered was that if the message sent from slave to master was too large it can cause memory issue. Static partition was rewritten to avoid this issue.

The Grid would sometimes not take in anymore task if there was too many runs of the code. This happened usually if the Bash file was written to run ten times for each image size. However, it was random and anything less than five runs for each file size would often succeed.