**Data**

Below is the table containing our findings. All measurements are in seconds, taken using the program `time` and recording the real time elapsed. Each test was performed twice, and the faster time was used. In the trials where there was a significant discrepancy between the times, another two tests would be performed, and the outlier result would be discarded, while the lowest of the remaining tests would be inserted into the table. The data was obtained with a blur radius of 10.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Time (in seconds)** | |  |  |
|  | **Image Dimensions** | | | |
| **Processes** | **1920x1080** | **2560x1600** | **2880x2560** | **3840x2160** |
| 1 | 22.05 | 37.727 | 72.766 | 119.575 |
| 2 | 15.132 | 22.117 | 48.571 | 82.787 |
| 3 | 11.386 | 16.935 | 25.652 | 61.476 |
| 4 | 10.087 | 14.236 | 39.178 | 44.997 |
| 8 | 8.036 | 10.115 | 29.294 | 25.074 |
| 16 | 6.759 | 10.359 | 10.333 | 15.974 |
| 32 | 20.91 | 33.27 | 54.467 | 61.469 |
|  |  |  |  |  |
|  |  |  |  |  |
|  | **Speedup** |  |  |  |
|  | **Image Dimensions** | | | |
| **Processes** | **1920x1080** | **2560x1600** | **2880x2560** | **3840x2160** |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1.45717684 | 1.70579192 | 1.49813675 | 1.44436929 |
| 3 | 1.93658879 | 2.22775317 | 2.83665991 | 1.94506799 |
| 4 | 2.18598196 | 2.65011239 | 1.85731788 | 2.65739938 |
| 8 | 2.74390244 | 3.72980722 | 2.4839899 | 4.7688841 |
| 16 | 3.26231691 | 3.64195386 | 7.04209813 | 7.4856016 |
| 32 | 1.05451937 | 1.13396453 | 1.3359649 | 1.9452895 |
|  |  |  |  |  |
|  |  |  |  |  |
|  | **Efficiency** |  |  |  |
|  | **Image Dimensions** | | | |
| **Processes** | **1920x1080** | **2560x1600** | **2880x2560** | **3840x2160** |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 0.72858842 | 0.85289596 | 0.74906837 | 0.72218464 |
| 3 | 0.6455296 | 0.74258439 | 0.9455533 | 0.648356 |
| 4 | 0.54649549 | 0.6625281 | 0.46432947 | 0.66434985 |
| 8 | 0.3429878 | 0.4662259 | 0.31049874 | 0.59611051 |
| 16 | 0.20389481 | 0.22762212 | 0.44013113 | 0.4678501 |
| 32 | 0.03295373 | 0.03543639 | 0.0417489 | 0.0607903 |

**Discussion**

From the data it can be seen that performance improves as the number of processes increases, although, the speedup effect becomes more diminished as each new processor gets added. Efficiency decreases as the number of processors increases for an image. (This means the program is not strongly scalable).

The data obtained with 32 processors is an outlier in every data set. At the time of testing, the server cluster was returning massively slowed down results for any processor count above 17.

The data points for 16 processors and the 2880x2560 and 3840x2160 resolutions were unusually high.

The program is not weakly scaleable because the efficiency doesn’t hold constant when the problem size increases at the same rate as the number of processes. (For example the efficiency between a 1920x1080 image and 2 processors to 2560x1600 and 4 processors.)