

## Project 2

### Diego Loya

### Fractal

		Runtimes			
		Number of Processes			
width x frames		1	5	10	15
	250x30	1.0333	0.2081	0.1041	0.0694
	250x60	2.0296	0.4159	0.2079	0.1388
	500x30	4.1255	0.8297	0.4208	0.2767
	500x60	8.1073	1.6582	0.8307	0.5532

		Speedup			
		Number of Processes			
width x frames		1	5	10	15
	250x30	-	4.97	9.93	14.89
	250x60	-	4.88	9.76	14.62
	500x30	-	4.97	9.80	14.91
	500x60	-	4.89	9.76	14.66

		Efficiency ratio			
		Number of Processes			
width x frames		1	5	10	15
	250x30	-	0.99	0.99	0.99
	250x60	-	0.98	0.98	0.97
	500x30	-	0.99	0.98	0.99
	500x60	-	0.98	0.98	0.98

The times decreased as we increased the number of processes because the tasks were split into each one. There is not a linear decrease in runtime because not the whole code was parallelized, though it was very close because most of the heavy-load part of the code was successfully parallelized.

There was no barrier needed because we called MPI\_Gather and all collective communication calls are blocking. This means that there is no need to implement a barrier under these circumstances because there is an implicit synchronization when MPI\_Gather is used.