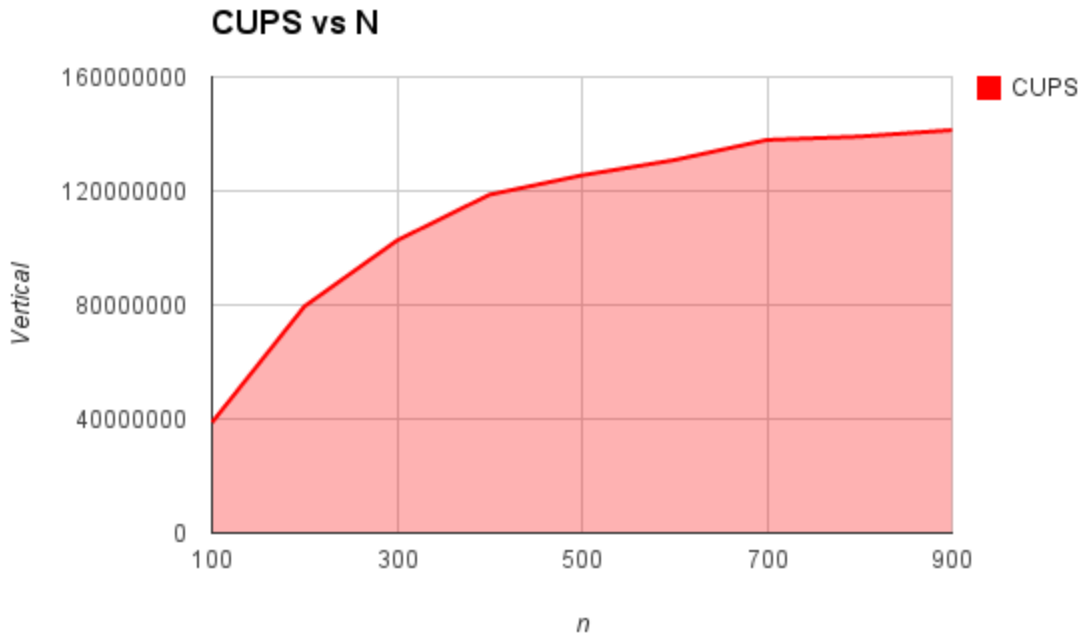
Assignment 5

Liam Cardenas, CSIL: liamcardenas, Perm: 7159965
Steven Cabral, CSIL: scabral, Perm: 6801179

Program instructions: Compile program using normal makefile. Run program by the following command: ./life r <filename> <number of rows/columns> <number of iterations>
Make sure that n is some number divisible by the number of processors or the program will not work.

Time was measured using the real time from the unix/bash "time" command.

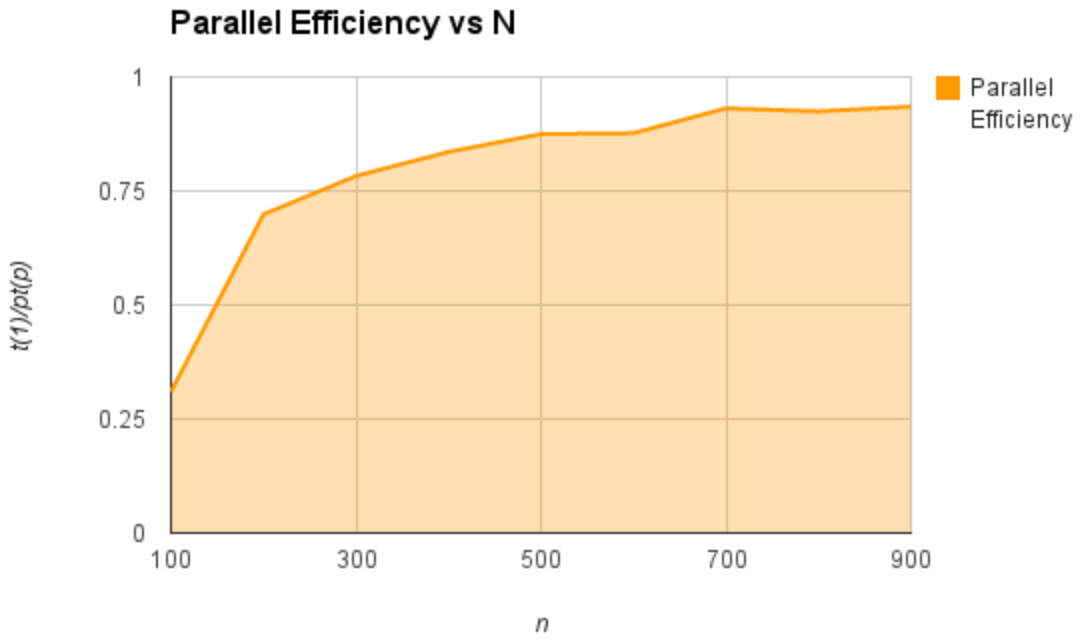
This shows what happens when n is scaled from 100 to 900 on 2 workers, and with 1000 iterations:



n	time	CUPS
100	0.258	38759689.92
200	0.503	79522862.82
300	0.877	102622577
400	1.348	118694362
500	1.992	125502008
600	2.751	130861504.9

700	3.553	137911624
800	4.601	139100195.6
900	5.73	141361256.5

This shows the parallel efficiency, with 2 workers, as n increases from 100 to 900:

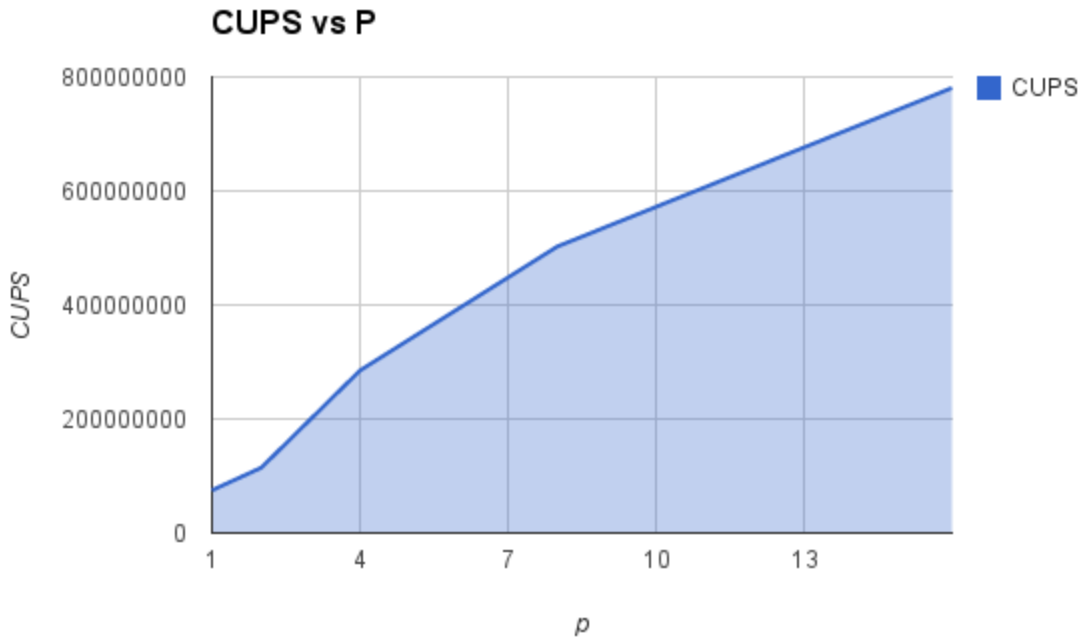


t1 comes from the following table (where p=1)

n	time	Parallel Efficiency
100	0.16	0.3100775194
200	0.703	0.6988071571
300	1.373	0.7827822121
400	2.253	0.8356824926
500	3.485	0.874748996
600	4.824	0.8767720829
700	6.617	0.9311849142
800	8.509	0.9246902847
900	10.715	0.934991274

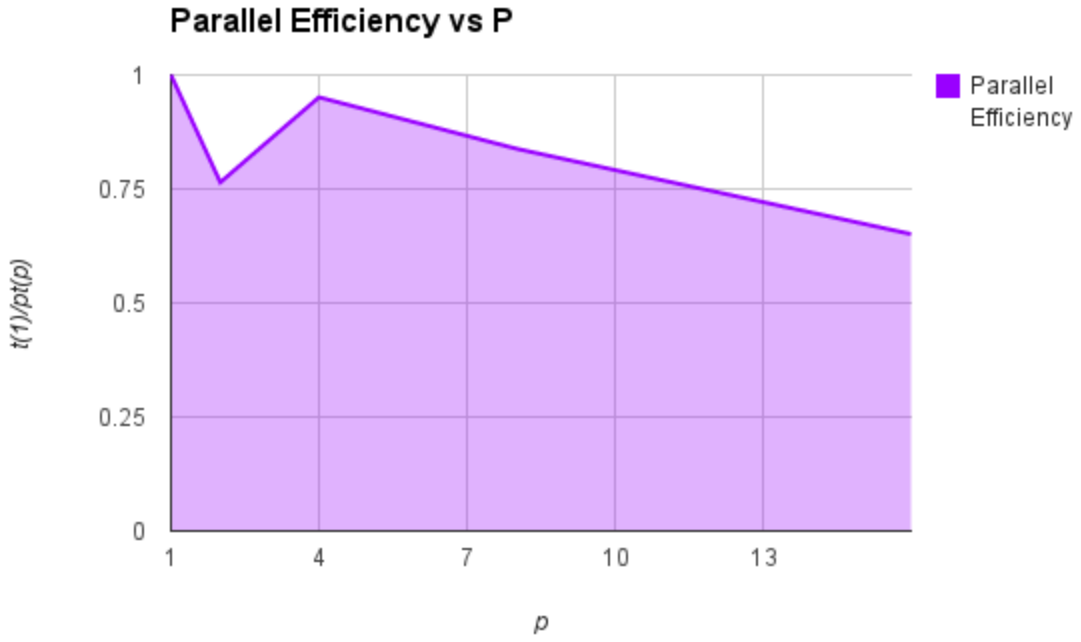
:

This shows what happens when the workers are scaled from 1 to 16 with n = 1600 and 1000 iterations:



p	time	CUPS	Parallel Efficiency
1	34.165	74930484.41	1
2	22.344	114572144.6	0.7645229144
4	8.98	285077951	0.9511414254
8	5.094	502552022	0.8383637613
16	3.279	780725831	0.6512084477

And last, but not least, parallel efficiency vs p, on the previous data set:


Parallel Efficiency vs P
1
0.75
0.5
0.25
0
t(1)/pt(p)
1
4
7
10
13
p
Parallel Efficiency