

Questions and Timing Data

Record just the non-file I/O time in nanoseconds using **-X -F**. All game boards are available in **/space/life** on **beast.cs.pacificu.edu**

Add data from Tables 1, 2, 3, and 5 to the shared spreadsheet.

== Evaluation Thread Speed Up ==

Table 1: Minimal Arch with 2 CPUs

Threads	largeGame.life 10 generations	largeGame.life 100 generations
1		
2		
4		
8		

Table 2: beast.cs.pacificu.edu

Threads	game_10K.life 40 generations	game_20K.life 10 generations	game_20K.life 20 generations
1			
2			
4			
8			
16			

== Evaluate CPU/Core/Thread effects ==

Table 3: beast.cs.pacificu.edu

	game_15K.life 40 generations	Time
One Thread	run normally	
One Thread, one core	taskset -c x ./CS460_Life x is 0-23	
Two Threads, any cores	run normally	
Two Threads, one core	taskset -c x ./CS460_Life ... x is 0-23	
Two Threads, same CPU, different cores <i>Note: this does not pin a particular thread to particular core but ensures that each thread runs on only the listed cores</i>	taskset -c x,y ./CS460_Life ... x is 0-5 y is 0-5, x != y	
Two Threads, different CPUs <i>See Note above</i>	taskset -c x,y ./CS460_Life ... x is 0-5 y is 6-11	
Two Threads, same core (via two HyperThreaded cores) <i>See Note above</i>	taskset -c x,y ./CS460_Life ... x is 0-23 y is (x+12) % 24	

You can run `game_100K.life` as a stress test. One generation with one thread could take 30 minutes (with `-X -F`).

Predictions

Based on your timing data, how long (non-File I/O time), in nanoseconds, do you expect the following scenarios to take? Run each scenario to determine how close your predictions are.

Table 4: Predictions

Generations	Expected Runtime		Observed Runtime	
	Arch largeGame.life 2 threads	Beast game_10K.life 4 threads	Arch largeGame.life 2 threads	Beast game_10K.life 4 threads
50				
200				
1000				

1. Discuss and explain what may cause any differences you see between expected and observed runtime above.
2. How much performance change did you see on Arch with 2 threads?
3. How much performance change did you see on Arch with 8 threads?
4. On beast, how close to a linear speed up do you see by adding more threads?
5. In Table 3, compare Two Threads, one core and Two Threads, same core. Do you see any performance boost from HyperThreading?
6. **We don't teach compilers any more!** Recompile your code using the `-O3` (capital Oh three) compiler option (full optimization for speed) to build the executable **CS460_Life_fast**. Re-run the following games with **CS460_Life_fast**

Table 5: beast.cs.pacificu.edu

Threads	game_10K.life 40 generations	game_20K.life 10 generations	game_20K.life 20 generations
1			
2			
4			
8			
16			

► From the shell the command: `cat /proc/cpuinfo` will display information about the current machine's CPU.