CS336 : Parallel & Distributed processing

Project 1 Report

Summary of tasks

For this project the main aim was to understand how circadian clocks work and then actually the infrastructure for our simulations working. The eventual goal is simulating mammalian circadian clock as a network of oscillators that signal each other, but for this project we are simulating the oscillators without any signaling. My work initially started with getting to understand the operations of the codes that support the simulation. My coding was updating the methods for phase_support.c which is the heart of the simulation and it runs statistical analyses through methods within it. After completing the tasks, I the tested the executables and then used valgrind to check for memory leaks.

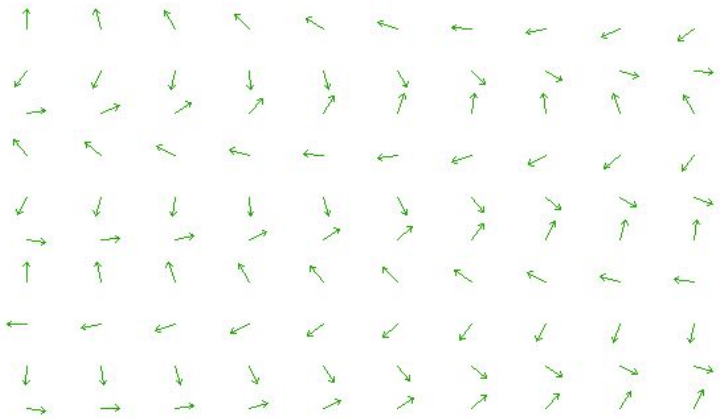
Tasks

Adding the following methods to phase_support.c

runPhaseSimulation - uses the Forward Euler method from Stephanie's guide to solve the set of differential equations that model Nx oscillators without any signals. Having written my method, i used disp_phase to visualize these oscillators.

5 oscillators

100 oscillators



runPhaseSimulationAndDetectEvents- Using the pseudocode guidelines, I made this function which returns an array of simulator events after processing it from event counts of the oscillators and the time series.
After compiling the test program, I got the following output:

```
[mggamedz@n23 ~/proj02]$ ./sim_events 2 5
Oscillator 0: 61 300
Oscillator 1: 61 302
Oscillator 2: 61 303
Oscillator 3: 61 304
Oscillator 4: 62 305
Ran in 0.001054 seconds
[mggamedz@n23 ~/proj02]$
```

Lastly I wrote **runPhaseSimulationFindEventStats**, which computes the standard deviation of the event times for each cycle. Testing this with sim_stats, i found this output for sim_stats:

```
[mggamedz@n23 ~/proj02]$ ./sim_stats 2 5
 0.126491 0.328634
[mggamedz@n23 ~/proj02]$
```

This was slightly different from Stephanie's example, but still close.

Extensions

1. Used valgrind to demonstrate no memory leaks . for sim_stas , no memory leaks

```
[mggamedz@n23 ~/proj02]$ valgrind ./sim_stats 2 5
==117241== Memcheck, a memory error detector
==117241== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et
 al.
==117241== Using Valgrind-3.12.0 and LibVEX; rerun with -h for copyrig
ht info
==117241== Command: ./sim_stats 2 5
==117241==
==117241== Conditional jump or move depends on uninitialised value(s)
==117241==    at 0x400F25: runPhaseSimulationAndDetectEvents (in /mnt/
export/home/mggamedz/proj02/sim_stats)
==117241==    by 0x40101F: runPhaseSimulationFindEventStats (in /mnt/e
xport/home/mggamedz/proj02/sim_stats)
==117241==    by 0x401CB9: main (in /mnt/export/home/mggamedz/proj02/s
im_stats)
==117241==
==117241== Use of uninitialised value of size 8
==117241==    at 0x400F5C: runPhaseSimulationAndDetectEvents (in /mnt/
export/home/mggamedz/proj02/sim_stats)
==117241==    by 0x40101F: runPhaseSimulationFindEventStats (in /mnt/e
xport/home/mggamedz/proj02/sim_stats)
==117241==    by 0x401CB9: main (in /mnt/export/home/mggamedz/proj02/s
im_stats)
==117241==
 0.126491 0.328634
==117241==
==117241== HEAP SUMMARY:
==117241==     in use at exit: 0 bytes in 0 blocks
==117241==   total heap usage: 7 allocs, 7 frees, 48,180 bytes allocat
ed
==117241==
==117241== All heap blocks were freed -- no leaks are possible
==117241==
==117241== For counts of detected and suppressed errors, rerun with: -
v
==117241== Use --track-origins=yes to see where uninitialised values c
ome from
==117241== ERROR SUMMARY: 60 errors from 2 contexts (suppressed: 0 fro
m 0)
[mggamedz@n23 ~/proj02]$
```

Sim_events no memory leaks as well

```
[mggamedz@n23 ~/proj02]$ valgrind ./sim_events 2 5
==117314== Memcheck, a memory error detector
==117314== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et
 al.
==117314== Using Valgrind-3.12.0 and LibVEX; rerun with -h for copyrig
ht info
==117314== Command: ./sim_events 2 5
==117314==
a
b
==117314== Conditional jump or move depends on uninitialised value(s)
==117314==    at 0x400F39: runPhaseSimulationAndDetectEvents (in /mnt/
export/home/mggamedz/proj02/sim_events)
==117314==    by 0x401BC3: main (in /mnt/export/home/mggamedz/proj02/s
im_events)
==117314==
==117314== Use of uninitialised value of size 8
==117314==    at 0x400F70: runPhaseSimulationAndDetectEvents (in /mnt/
export/home/mggamedz/proj02/sim_events)
==117314==    by 0x401BC3: main (in /mnt/export/home/mggamedz/proj02/s
im_events)
==117314==
Oscillator 0: 61 300
Oscillator 1: 61 302
Oscillator 2: 61 303
Oscillator 3: 61 304
Oscillator 4: 62 305
Ran in 0.011456 seconds
==117314==
==117314== HEAP SUMMARY:
==117314==     in use at exit: 0 bytes in 0 blocks
==117314==   total heap usage: 5 allocs, 5 frees, 48,140 bytes allocat
ed
==117314==
==117314== All heap blocks were freed -- no leaks are possible
==117314==
==117314== For counts of detected and suppressed errors, rerun with: -
v
==117314== Use --track-origins=yes to see where uninitialised values c
ome from
==117314== ERROR SUMMARY: 60 errors from 2 contexts (suppressed: 0 fro
m 0)
[mggamedz@n23 ~/proj02]$
```

Sim_slow no memory leaks even here

```
[mggamedz@n23 ~/proj02]$ valgrind ./sim_slow uncoupled_5.phs 5
==117321== Memcheck, a memory error detector
==117321== Copyright (C) 2002-2015, and GNU GPL'd, by Julian Seward et
 al.
==117321== Using Valgrind-3.12.0 and LibVEX; rerun with -h for copyrig
ht info
==117321== Command: ./sim_slow uncoupled_5.phs 5
==117321==
gets here
here too
passes this
here as well
ret 0.000000 0.000000
==117321==
==117321== HEAP SUMMARY:
==117321==     in use at exit: 0 bytes in 0 blocks
==117321==   total heap usage: 4 allocs, 4 frees, 48,648 bytes allocat
ed
==117321==
==117321== All heap blocks were freed -- no leaks are possible
==117321==
==117321== For counts of detected and suppressed errors, rerun with: -
v
==117321== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from
 0)
[mggamedz@n23 ~/proj02]$
```

Collaborators

I got help from Brandon and Zhuofan on this project