# Project 4: Functional Decomposition ("Grainville")

Behnam Saeedi
(Saeedib@oregonstate.edu)
CS575: Parallel Programming

Spring 2018

**Abstract**

## CONTENTS

## 1 SYSTEM SPECS

The computer used for running these experiments is A lenovo with i7 (Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz) with 8 GB of ram.

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    2
Core(s) per socket:    2
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 58
Model name:            Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz
Stepping:              9
CPU MHz:               2793.686
CPU max MHz:           3100.0000
CPU min MHz:           800.0000
BogoMIPS:              4988.73
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              4096K
NUMA node0 CPU(s):     0-3
```

## 2 APPROACH

This program takes advantage of 5 agents:

1) **Grain:** The grain is the primary food source for the Deer and Noah.
2) **Deer:** The deer is the primary consumer of grain and food for the wolves and Noah
3) **Wolf:** The wolves are primary consumers of the deer flesh, but that does not stop a hungry Noah from eating them.
4) **Noah:** That is right! Noah eats everything and anyone, relentlessly. He is trying to survive the life on Arc ... or fishing boat.
5) **Weather:** Weather dictates the birth and growth rate of the grain, deer and wolves.

Here is the prologue to the story:

Story time!

So God said to Noah, I am going to put an end to all people, for the earth is filled with some wicked stuff because of them! I am surely going to destroy both them and the earth because ... it's more dramatic that way!

Go build an arc and take a pair of each animal (don't forget the Dodo They look like birds, but they can't fly ... I guess chickens too).

- "But god Jesus was good at carpeting, not me " said Noah to God

- Not with the whole "ow your son this and that" again replied God; Just build a canoe or a small fishing boat would do, I'll shrink everything!

So Noah built the boat, but God got busy with planning his next religion and forgot to shrink the animals. Noah decided to teach God a lesson, and only brought some grain, 6 Deers and 2 Wolves to his boat.

Here is what happened next according to Noah's Diary:

Furthermore, we are using barriers in order to synchronize the timing of all agents for the time of the day.

```
#pragma omp barrier
```

Furthermore, we are taking advantage of mutexes in order to make sure multiple threads are not reading and writing at the same time.

```
omp_set_lock(&GrainLock);
omp_set_lock(&DeerLock);
omp_set_lock(&WolfLock);
```

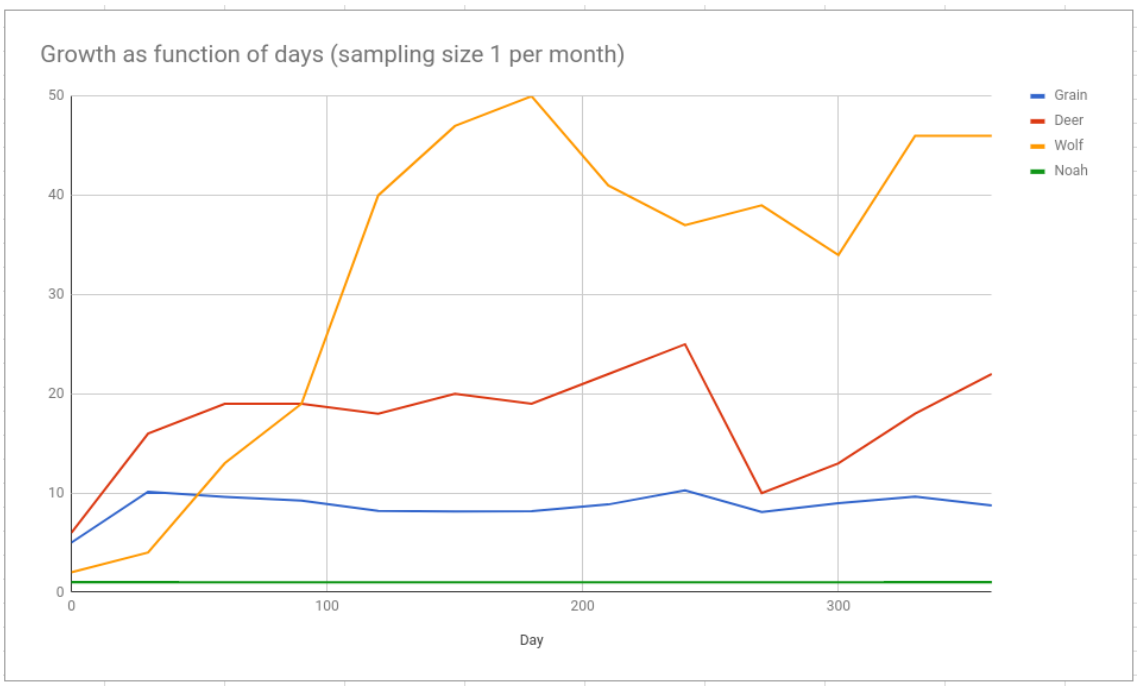
## 3 PERFORMANCE

The program performed well. The 4 threads running took 0.756326 seconds in order to finish the task for the span of time of events. The Grain, dear and wolf population was recorded and analyzed.

## 4 ANALYSIS OF RESULTS

Table 1
Agent vs day sampled 1 per month

| Day vs Type | Grain | Deer | Wolf | Noah |
|---|---|---|---|---|
| 0 | 5 | 6 | 2 | 1 |
| 30 | 10.14 | 16 | 4 | 1 |
| 60 | 9.62 | 19 | 13 | 1 |
| 90 | 9.23 | 19 | 19 | 1 |
| 120 | 8.2 | 18 | 40 | 1 |
| 150 | 8.15 | 20 | 47 | 1 |
| 180 | 8.16 | 19 | 50 | 1 |
| 210 | 8.85 | 22 | 41 | 1 |
| 240 | 10.27 | 25 | 37 | 1 |
| 270 | 8.09 | 10 | 39 | 1 |
| 300 | 8.98 | 13 | 34 | 1 |
| 330 | 9.63 | 18 | 46 | 1 |
| 360 | 8.74 | 22 | 46 | 1 |




Here is the results of the experiment with regards to growth of different agents.

The table indicates how different agents interact with one another. Each agent affects the dynamics of growth rate of other agents.

We can see the wolves were the most successful agent and Noah managed to survive with about 50 hungry wolves. We can see that the drop in population of wolves is immediately after a drop in population of dear and a delayed reaction to a drop to amount of grain at around 10th month (300 days). This information could be found in table 1.

## 5 Conclusion

We can see that It is possible to simultaneously run several threads with different instructions and behaviors. use of mutex, and proper data sharing and timing practices help us to create such a simulation and analyze the produced data.