Accelerating Worm Segmentation through Inter-node Parallelism

Vineeth Remanan Pillai[1], Daniela Stan Raicu[2], Ioan Raicu[1]
[1]Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA
[2]School of Computing, DePaul University, Chicago, IL, USA
vpillai3@hawk.iit.edu, draicu@cs.depaul.edu, iraicu@cs.iit.edu

***Abstract* - C.Elegans is a type of roundworm and one of the researcher's choice of small worms when it comes to neuronal development studies and molecular and developmental biology studies. A large amount of data is generated in C.Elegans tracking, which then is processed to determine the path, speed, and trajectory of the worm, in order to identify higher level functions and behaviors. This work aimed to accelerate the processing of the C.Elegans tracking data through inter-node parallelism. We achieved two to three orders of magnitude performance improvement when comparing to the original Java-based processing system.**

I. Overview

One of the main goals of neuroscience is to understand how complex behaviors arise from the networks of neurons. Specifically, given that numerous psychiatric and neurological diseases in humans are associated with aberrant neural wirings, unraveling the complete neural connection map would help us understand brain diseases and identify therapeutic targets. The nematode Caenorhabditis (C.) elegans (see Figure 1) is widely used for unraveling the principle underlying functional neural circuits. C.Elegans was the first multicellular organism to have its whole genome sequenced. The worms have a simple neural network with exactly 302 neurons and approximately 7000 synaptic connections.

Figure 1: C. elegans

To study the locomotory behavior of C.Elegans, a setup consisting of both hardware and software was designed [1] (see Figure 2). The setup was designed specifically to track one worm at a time. The goal was to understand and study the locomotory behavior of the worm at various environments like a food deprived situation for example, for a long duration of time. Knowledge gained from these studies could be applied to complex nervous systems like those of mammals.

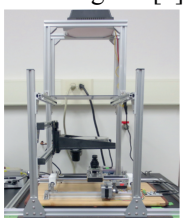

Figure 2: C.Elegans tracker

II. Proposed Work

This work aims to accelerate the modeling and analysis of the C. elegans locomotory behavior to establish functional neural maps using high-resolution, high speed video data. The original processing systems was implemented in Java which performed worm segmentation by taking the raw video of worm's movements and processes the video to track and record the movement of the worm. Individual video frames are extracted and then converted to gray scale. Then a smoothing (blur) filter is applied to reduce the noise and homogenize pixel densities on the worm body. These filtered images are then converted to a binary image which has only black or white pixels. Black pixels represent the background and white represents the worm.

We developed a processing system implemented in C++ (porting all the functionality of the Java implementation), and used multi-threading [2], MPI [3], and Swift [4] to accelerate the processing through inter-node parallelism. The design goal was to make the application scalable in a high performance environment with minimal dependency to external libraries to aid the portability across various super computing environments.

III. Performance Evaluation

We evaluated the performance characteristics of the application across several implementations including Java, C++, Swift, and MPI. Figure 2 compares the cost of processing 47,000 images at various resolutions (such as those found in Figure 1) across our various implementations on a single system with 24-cores and 128GB RAM.

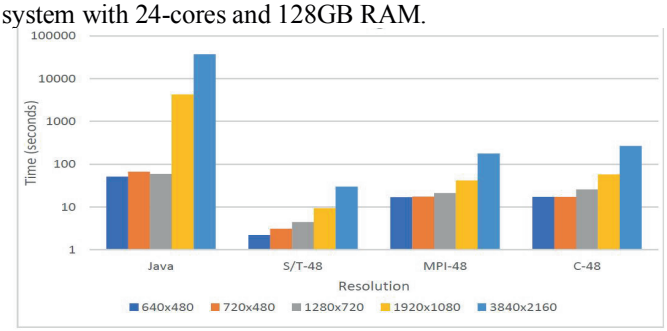

Figure 2: Performance comparison of C.Elegans processing

These results show that the Java code has significant overheads compared to the other existing parallel solutions that are based on C++. At low resolution, we are able to achieve more than an order of magnitude performance improvement, while at high resolution, the improvements are even more dramatic (2 to 3 orders of magnitude), reducing run time from hours to minutes. The Swift and MPI approaches are possible to carry out parallelism across nodes as well, which will be explored in future work.

ACKNOWLEDGMENT

This work was supported in part by the NSF awards OCI-1054974 and NSF-1461260, as well as the Chameleon Testbed (NSF award 1743358 and 1419141).

References

[1] K. Moy, et al. "Computational Methods for Tracking, Quantitative Assessment, and Visualization of C. elegans Locomotory Behavior", PloS one 10, e0145870. 2015
[2] David R. Butenhof, "Programming with POSIX Threads", 1997
[3] William Groppa, et al. "A high-performance, portable implementation of the MPI message passing interface standard", 1996
[4] Michael Wilde, et al. "Extreme-scale scripting: Opportunities for large task-parallel applications on petascale computers", SciDAC 2009