CMPUT 481 Assignment 1 Report
Shared Memory Parallel Sorting by Regular Sampling

CCID: jxie2
Student Number: 1372777
Student Name: Jiahui Xie

1. Introduction

Due to the poor time management employed by the author, the implementation was unfinished: he spent a significant amount of time inspecting the exact location of race conditions and unfortunately there was not much time left after the race conditions for phase 2 and 3 were finally seem to be located. The current existing phase 4 of the implementation was plainly wrong because the accumulated length of the final array was much less than the original length specified.

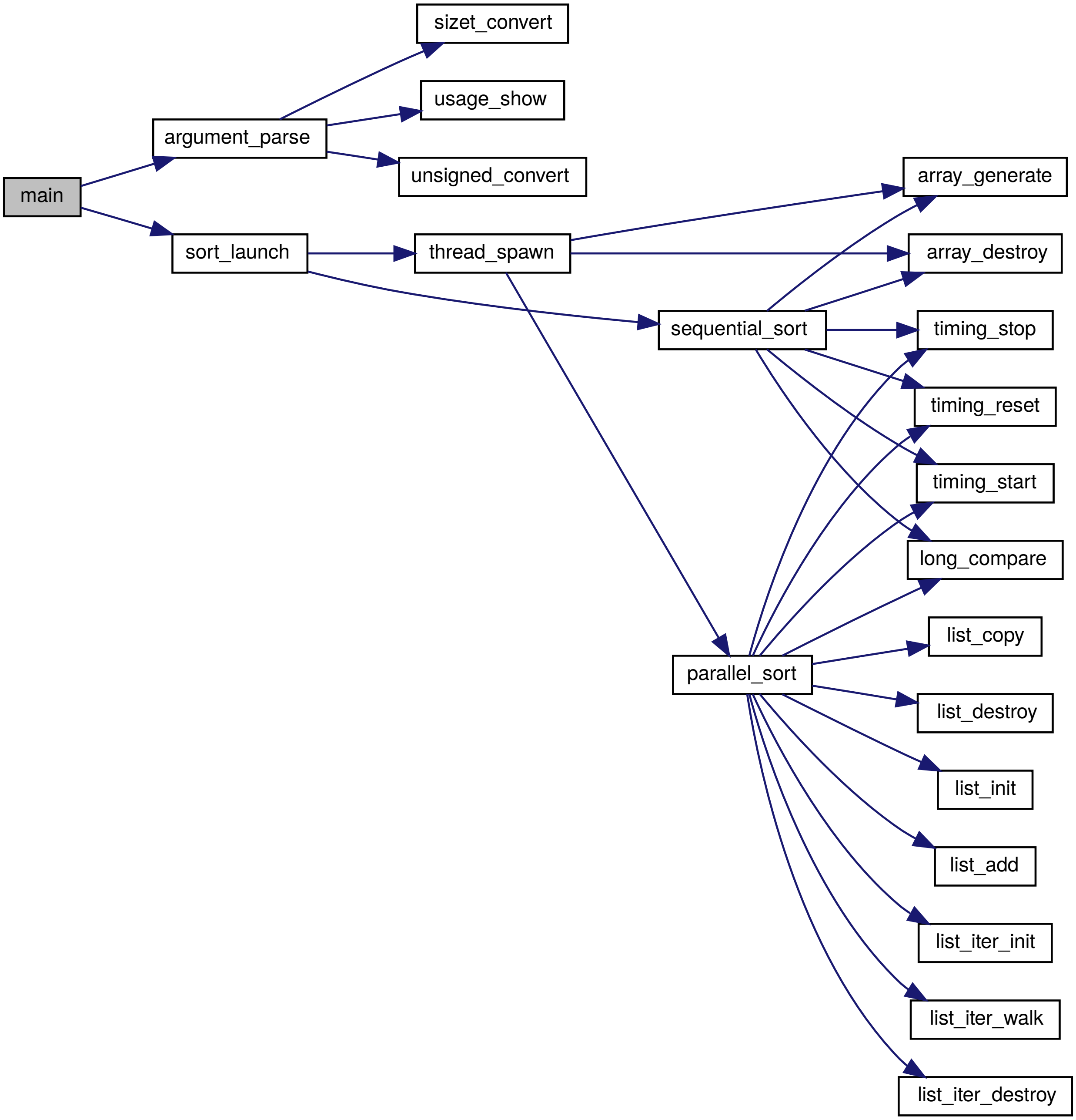

Figure 1: Call Graph of the Main “PSRS” Program

2. Implementation Overview

The usage information of the program can be viewed from the “Getting Started” section of the “README.md” file. Among all of the source files, the most important one is “sort.c” located under the “src” folder: the main parallel sorting by regular sampling algorithm is implemented in the “parallel_sort” function where as the one-thread sequential quick sort is implemented as a function named as (not surprisingly) “sequential_sort” (Figure 1). Another source file that worths special attention is “list.c”: the one-way linked list is used for storing the thread specific local regular samples in phase 1; pivot values obtained from phase 2 is shared through a file scope list variable.

3. Experiment

The experiment can not be performed properly because of the reasons given in section 1. The author deeply apologies for the absence of the most important part of this report.

References

Kerrisk, M. *The Linux Programming Interface.* No Starch Press, San Francisco, 2010.

Li, X., Lu, P., Schaeffer, J., Shillington, J., Wong, P.S., Shi, H. *On the Versatility of Parallel Sorting by Regular Sampling*. Parallel Computing 19, 1079-1103, 1993.