

# Assignment 2: Parallel Programming

## 1 Introduction

Traditionally, computers have only one processor and algorithms were designed to be sequential. But, today's computers have more than one processing element (either multiple CPU cores or GPU). Sequential programs are slow as they need to process the instructions one by one. Performance of some sequential programs (not all) can be improved by exploiting the parallel architectures. Let us consider a vector addition example. Two vectors of length  $n$  can be added in parallel using at most  $n$  processors, as shown in Figure 1<sup>1</sup>. Here,  $p_i$  denotes  $i^{th}$  processor. If we have  $n$  processors, then asymptotic run time of a vector addition program can be reduced from  $O(n)$  to  $O(1)$  upon parallelizing.

In general, if you have  $p$  processors and execution time is  $t_s$  seconds for a sequential version of an application, then you should see a speed-up (theoretically) of  $O(\frac{t_s}{p})$  upon parallelizing the application. Please note that all sequential algorithms can't be converted to parallel algorithms and you have to be cautious while taking these design decisions.

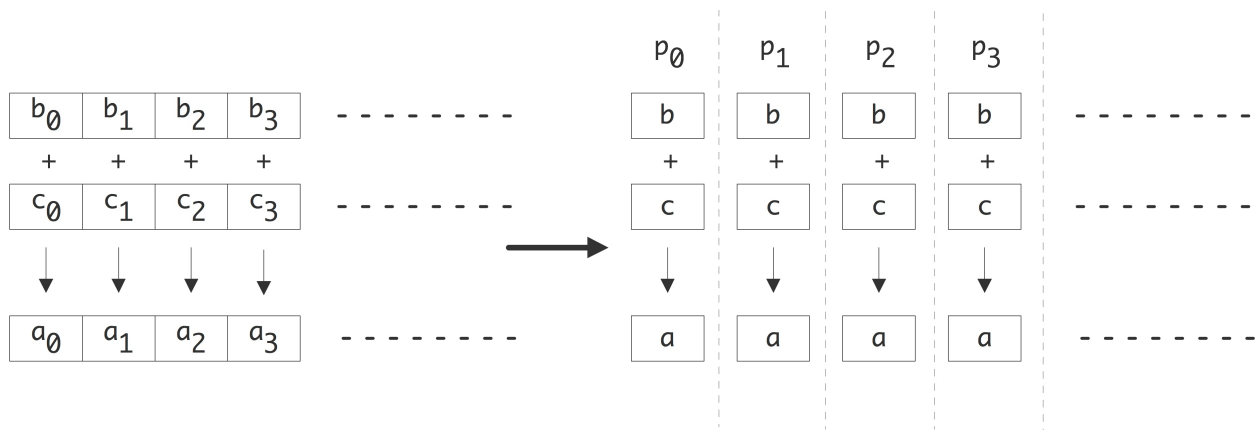


Figure 1: Vector addition: Sequential to Parallel example

## 2 Assignment

In this assignment, you are asked to select an application of your choice and parallelize it. You can use any programming language (C/C++, Java, Python, etc.) with which you are familiar. Also, you are free to use any hardware (CPU, GPU or APU) and Operating system (Windows, Linux, OS X). You are allowed to use external libraries such as OpenMP, OpenCL, CUDA, etc., but you are not allowed to use others work (such as sample programs and applications/programs written by other people).

You are required to submit a report of about 3-4 pages along with the code (both sequential version and parallel version) in a zip file. You should name your zip file as: **uwnetid\_a2.zip**. If you are submitting as a group of two, then please mention the group member names in report as well as in header of source code file.

Your report should contain at least following information:

<sup>1</sup>source: <http://pages.tacc.utexas.edu/~eijkhout/istc/html/parallel.html>

- *Programming language and hardware details:* In this part, you should include details about programming language and hardware. This is a place to mention about external libraries also.
- *Application details and design decision:* In this part, you should try to answer questions such as Which application are you parallelizing? Why you choose this particular application? What is interesting about this application?, etc.
- *Discussion:* In this part, you should discuss the performance of the application in terms of different parameters which you feel are relevant for analysis. Speed-up in execution time could be one of the parameters.

**Few important points:**

- Although you are flexible to choose any application, you have to be particular about your parameter selection. For example, if you chose  $\mathbf{n}=\mathbf{p}=4$  for vector addition example, then speed-up in performance is negligible. To see speed-up for such simple applications, you need to use a very large value of  $\mathbf{n}$ , say 100,000. Poor selection of parameters will lead to marks deduction.
- Some applications which might be of interest are: matrix multiplication, image filtering, discrete cosine transform, and discrete wavelet transform.

We will reward entries that go above and beyond the assignment requirements.