

CSE6220 Programming Assignment 2 Report

Boqin Zhang (bzhang376), Siying Cen (scen9), Yi Jiao (yjiao47)

March 10, 2021

1 Algorithm Description

The parallel algorithm for calculating the value of polynomial is composed of four steps.

Firstly, process 0, as the process having the number of terms of the polynomial and the array of constants, computes the number of terms every process needs to handle and scatter the constants to their corresponding processes.

Secondly, let process 0 broadcast the value variable x to every process by hypercubic communication.

Thirdly, all processes perform prefix-sum algorithm with variable x as elements and multiplication as operator. Now every process has its x^i array.

Fourthly, all processes compute the terms $a_i x^i$. Then all processes perform prefix-sum algorithm with terms $a_i x^i$ as elements and addition as operator. Now the last value of the prefix sum on the last process is the value of the polynomial.

At last, let the last process broadcast the value of polynomial to all processes. It will send the value to process 0 at first, then process 0 broadcast it to all other processes by hypercubic communication. Now every process has the final result.

2 Information of Tests

To verify the accuracy of the parallel algorithm, a serial function for computing the value of polynomial is developed.

A polynomial whose length is 1637809 is generated to be the one to compute. 7 values of variable x is prepared for testing the average computation time. The time of computing the first polynomial value is abandoned, so the time to be analyzed is the average of time of computing 6 values.

To let the result of computing time meaningful, the amount of processors used for testing is 2, 4, 6, 8, 12. The submission file is in the folder.

3 Test Results

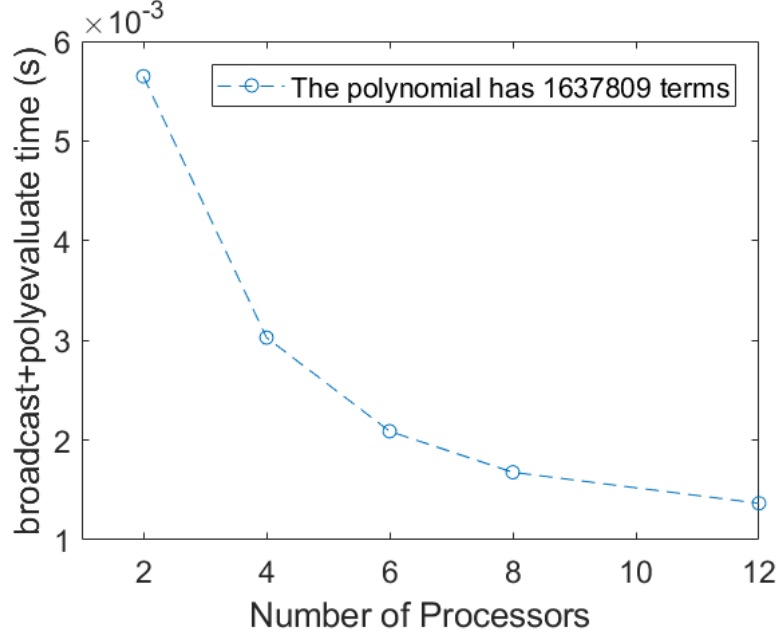


Figure 1: broadcast+PolyEvaluator time

4 Result Analysis

1. The running time of the algorithm is decided by the time of polyEvaluator. The time of broadcast is a small part of total running time.
2. With the increasing of amount of processors, the running time is decreasing like an inverse function. It shows that the algorithm is efficient in the test interval of number of processors.
3. However, it is not a real inverse relationship. If we compute the product of total time and number of processes, we can find that the product becomes larger when more processors are participated in. It means that more resources are used if more processors are used.

Appendix:

The test polynomial file is sample-constants2.txt; the test variable file is sample-values2.txt; the job submission file is pa2.pbs.

The test results are in testNonp_1637809_final.txt, testnp2_1637809_final.txt, testnp4_1637809_final.txt, testnp6_1637809_final.txt, testnp8_1637809_final.txt, testnp12_1637809_final.txt.