Message-Passing Programming: 2-D Image Recomposition Coursework

Jack Tyler: 27513556

May 24, 2019

In the early days of computing, computational power progressed at an alarming rate; constant field scaling meant that transistors could be fabricated smaller and smaller, allowing them to use less power, and to increase their clock speed. However, transistor manufacture has hit a physical limit for scaling (Mcfarland and Flynn, 1995), and increasing the clock speed any further would require highly capable cooling solutions impractical for use. As a result, processor companies have had to find other methods to speed up computer processor to keep computational power, and Moore's Law, advancing. Perhaps most prominent is the introduction of distributed parallelism, where multiple processor units, which otherwise execute independently and asychronously, may communicate and share tasks.

This type of parallelism is one of the most common implementations currently used in high-performance computing to decrease program execution time. However, care must be taken as to how each processor unit executes the source code; essential data required on multiple processors must be communicated correctly, and in as quick a manner as possible.

This report investigates the run-times for a two-dimensional image re-composition program. In particular, it studies the strong scaling of the algorithm, and analyses the major performance limitations inherent to the program.

1 Introduction to the Problem

Problem is mad fucked, yo.

References

Grant Mcfarland and Michael Flynn. Limits of Scaling MOSFETs. (January), 1995.