COMPSCI 4F03 Class Notes

Shawn Li

January 7, 2020

Contents

1 Introduction

1.1 How to handle a large complex task?

1.1.1 Sequential

One person works on different parts of the task.

1.1.2 Parallel

A number of person work on different parts, that are combined to complete the task.

1. Advantages of second approach
   (a) Same job in less time.
   (b) Much bigger jobs can be handled in reasonable time.
2. Presumptions of second approach

(a) There will be contension.

(b) Because of the contension, there have to be communication.

(c) Consequently, in real world, the ideal result would be not be possible, because communication and resolving contension will need time, however, the improvements are significent.

(d) Diminishingr returns: as more workers/threads are added, the benifit of each worker/thread added will be reducing, to a certain point the benifits are almost zero.

1.1.3 Distributed parallelism

Separate memories to reduce contension, but communication will be increasing.

1. Load balancing Work load should be roughly the same for each worker/processors, but communication is still necessary.

1.2 Parallel Programming

Bigger problems can be solved if more than one computers are used in parallel.

1. Sequential programming will not work. Mature tools and compilers are not available.

2. Locations of processors can be different.

   (a) Inside a single enclosure.

   (b) A number of computers connected together.

3. Different interconnection methods can be used.

   (a) Direct connections.

   (b) Ethernet.

1.2.1 Alternatives

Very fast single processor computers with large memory bandwith. It is more familiar with existing technologies. But it is very expensive, gives out a lot of heat. Also, the performance of a single CPU eventually will reach its limit.

1.2.2 Software considerations

1. How to decompose a program into seperate independent parts.
2. Parrellel algorithms.
3. Operating system support.

1.2.3 Different parallel programming models.

1. Shared memory
2. Message passing
3. Instruction level parallelism
4. Multithreading
5. Data parallelism
6. Hybrid systems