Contents

1 Data Parallelism

A data parallel computation performs the **same operation(s) to different items** of data at the same time

2 Task Parallelism

- A task parallel computation performs **distinct computations** (tasks) at the same time
- Set of tasks is **fixed** and so parallelism is not scalable
- One common example of task parallelism is pipelining, where a series of tasks **are solved in sequence**

3 Speedup

Speedup is the execution time of a sequential program (TS) divided by the execution time of a parallel program (TP) that computes the same result

3.1 Superlinear Speedup

- Sometimes (rarely) a parallel program on P processors may have a speedup greater than P. The speedup graph goes above P, hence **superlinear**
- The parallel version may be able to use cache memory more effectively (one per processor), reducing main memory access

3.2 Efficiency

Efficiency is normalised speedup, and indicates how efficiently each processor is used

$$Efficiency = Speedup/P$$

3.3 Concerns with Speedup

- New generations of hardware have generally increased processor performance more than communication latency
- Specific choice of T_S (sequential execution time) for comparison may be unfair
- Comparison may be with T_P for 1 processor (generally slower than T_S), termed relative speedup

3.4 Scaled Speedup vs Fixed-Size Speedup

- Speedup is most intuitive when comparing the same problem on different machines, i.e. **fixed size speedup**
- **A small problem** may not scale well to a **large machine**, but a **large problem** may not be executable on **a small machine**
- Scaled speedup tries to increase the size of the problem to **match the size of the machine** But in practice this can be hard or impossible, for example parts of the algorithm or memory requirements may be *non-linear*

4 Sources of Performance Loss

4.1 Overhead

Overhead is any cost incurred in the parallel solution but not in the serial solution:

- Time required to set up and tear down threads and processes
- Communication among threads and processes
- Synchronization, i.e. coordinating work with other threads or processes (acquiring, releasing locks etc.)
- Additional Computation, i.e. extra work such as calculating **what part of the problem** this thread should tackle
- Additional memory may be required by the parallel version, which may limit its use

4.2 Contention for resources

- Contention is degradation of system performance caused by **competition for a shared resource**
- Can cause parallel programs to become slower as more processors are added

4.3 Idle processors

Ideally all processors will be working all of the time, but a thread or process may not be able to proceed because of

- A lack of work to do (*load imbalance*, some threads have more work to do than others)
- Waiting for some external event
  - typically the arrival of new data to work on
  - Time to **load data from disk storage** may be a limiting factor, especially when working with large data sets
  - For a memory-bound computation the speed at which data can be fetched from main memory is the key limiting factor

4.4 Non-parallelisable computation

If a computation is inherently sequential then more processors will not help. The sequential operation is essential and fundamental to the algorithm

5 Amdahls Law

If $1/S$ of a computation is inherently sequential the maximum possible speedup is S

$$T_P = 1/S * T_S + (1 - 1/S) * T_S/P$$

Where $1/S$ is the percentage of sequential code and P is the number of processors

5.1 Notes on Amdahls Law

- Bad news: the parallelisable part probably **wont be perfectly parallelisable**. So maximum speedup wont be achieved
- Good news: Amdahls law applies to scaling a specific instance of a problem
  - If instead we try a problem that is (say) twice as big, then the **inherently sequential component may not be twice as big**
  - So S will often be **bigger for bigger problems**
  - So a larger parallel machine may allow us to **tackle larger problems in the same time**, even if it doesnt perform much faster for smaller problems than a smaller machine

Reference section

placeholder