Sammendrag Superdatat

torholmslettebak

07.01.2015

Innhold

1 Introduction **1**

2 Supercomputing **1**
2.1 Solution time on supercomputer 1
2.1.1 Single-processor performance 2

3 Floating point representation - the IEEE standard **2**

1 Introduction

2 Supercomputing

Using computers to solve problems which are compute intensive, i.e., problems which require large computing resources in terms of memory or floating point operations or both. Good examples are weather forecast, oil exploration, seismic analysis and computational mechanics.

Due to the resource demanding aspects of supercomputing, it is important to make the whole solution process as efficient as possible. Several fields are important to study: numerical/computational algorithms, as fast, robust and accurate as possible, the software development, treatment of large data sets, visualization, and validation of the simulation results.

Supercomputing implies the use of parallel processing, which means underlying algorithms need to be designed and tuned for this environment, to efficiently share information between processor units.

2.1 Solution time on supercomputer

Let T_1 be the solution time of some problem on a single-processor machine. Assume that we port this simulation to a multi-processor system with P processors. Ideally the solution time would be reduced by a factor of P. We would like a speedup:

$$S_p = T_1/T_p = P \tag{1}$$

Where T_p is the solution time on P processors. Typically the speedup, $S_p < P$. If we increase the number of processors to solve the problem, initially we will have a decent speedup, followed by a degradation as we add more processors. In order to achieve a perfect speedup, we cannot have overhead[1] when moving from single-processor system to a multi-processor system. In reality, we have communication between the processors, i.e. overhead. As the number of processors is increased, the work per processors goes down, while the overall communication (overhead) goes up. One critical issue is therefore to minimize the communication overhead so that we can take advantage of larger systems (at least for fixed problems, more communication needed for unfixed problems).

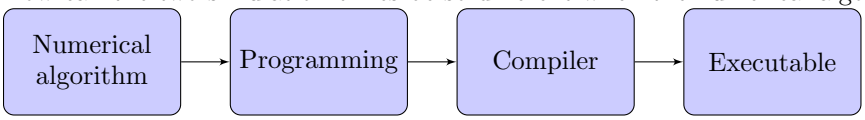
[1] overhead - combination of excess or indirect computation time, memory, bandwidth, or other resources that are required to attain a particular goal

2.1.1 Single-processor performance

We have to different versions, A and B, of an application. The underlying numerical algorithms are identical. The simulation times denoted as T_1^A and T_1^B. We observe that:

$$T_1^A = 3T_1^B \tag{2}$$

How can the two simulation times be so different when the numerical algorithms are the same.



A few aspects that can cause this:

- Different choice of data structure / memory layout in programming
- different use of optimized numerical libraries
- different choice of compiler optimization

Other ways to change solution time for this simulation could be to change the computer, or change the computational/numerical algorithms.

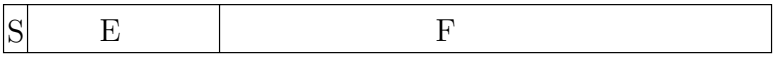
3 Floating point representation - the IEEE standard

All real numbers are represented with a finite number of bits. The actual decimal value V of the floating point is:

$$V = (-1)^S \cdot 2^{E-B} \cdot M \tag{3}$$

A value $S = 0$ means that the number is positive, while S = 1 means that the number is negative. E is an exponent that can be adjusted so that the Mantissa can be expressed as

$$M = 1 \cdot b_1 b_2 ... \tag{4}$$



Figur 1: Each floating point has a binary representation with three fields: S denotes the sign of the number, E is an exponent, and F is the fraction part of the mantissa