Introduction to parallel computing part III

Aslak Tveito

University of Oslo

Geilo, March 2001

Performance Modelling for PDE's

Difficulties with standard performance modelling

- Extremely complicated for real applications
- Puts severe restrictions on the computers
- Almost impossible to do correct
- Difficult to estimate CPU-efforts needed to evaluate elementary functions $(\sin, \cos, \tan, \sqrt{\cdot}, (\cdot)^{1/3}, \ldots)$

We need a method based on the application itself.

Basic operations of methods for solving PDE's

- Inner products
- vector + scalar$\times$vector
- Matrix-vector multiplications
- Simple linear or non-linear updates

Simplified model

Consider a typical 2D problem. Suppose we have

- p processors
- m computational nodes

Organize the processors as a two-dimensional grid of

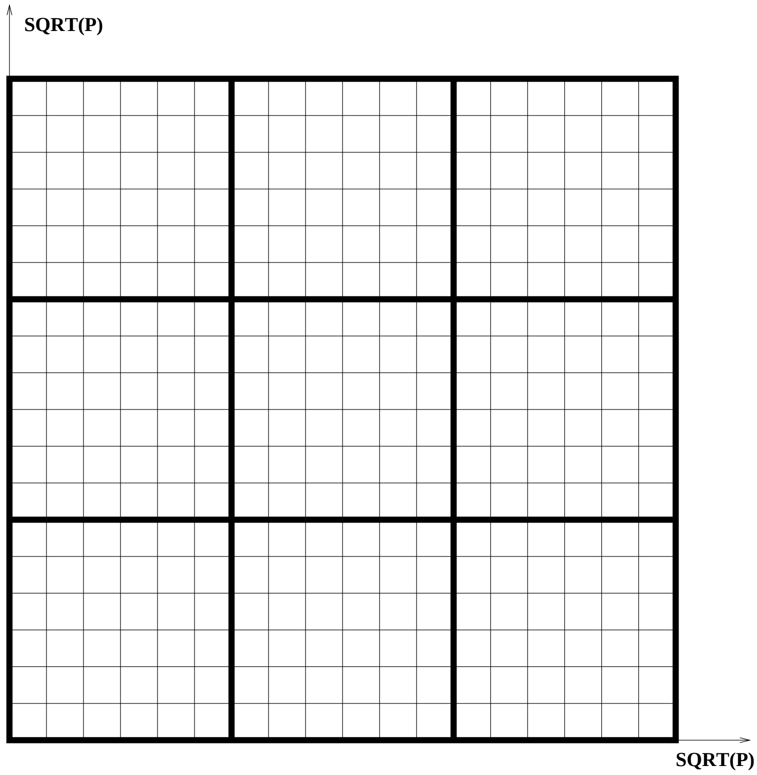
$$p = \sqrt{p} \times \sqrt{p}$$

processors.

Similarly, we assume that the grid is uniform with

$$m = \sqrt{m} \times \sqrt{m}$$

nodes.



On each processor there are

$$\frac{\sqrt{m}}{\sqrt{p}} \times \frac{\sqrt{m}}{\sqrt{p}} = \frac{m}{p}$$

computational nodes. The ”boundary” of each processor consists of

$$\sim \frac{\sqrt{m}}{\sqrt{p}}$$

nodes.

Here '$\sim$' means proportional, i.e.

$$\sim \frac{\sqrt{m}}{\sqrt{p}} \sim 2\frac{\sqrt{m}}{\sqrt{p}} \sim 4\frac{\sqrt{m}}{\sqrt{p}}.$$

One typical step in the PDE-solver involves communications of order

$$T_{cm} \sim \frac{\sqrt{m}}{\sqrt{p}}$$

and the computation time is of order

$$T_{cp} \sim \frac{m}{p}$$

Hence, it is reasonable to assume that the total computing time is

$$T = \alpha\frac{m}{p} + \beta\frac{\sqrt{m}}{p},$$

where α and β must be determined through observations.

Notes

- The model implies that for $\frac{m}{p} = \text{ constant}$, the computing time is constant. This is one way to define an optimal scaling.
- The model is only relevant for

$$p \ll m,$$

which is ok for any reasonable applications. Typically we have

$$m \sim 10^5 p$$

One numerical experiment

We consider a 2D simulator using 400 Conjugate Gradient steps. Based on the observations, we used the least squares method to compute α and β resulting in the following model

$$T = 0.00106\frac{m}{p} + 0.00937\frac{\sqrt{m}}{\sqrt{p}} \approx 10^{-3}\left(\frac{m}{p} + 10\frac{\sqrt{m}}{\sqrt{p}}\right)$$

In the following table, we compare actual timings and estimates. The estimates are given in parentheses.

m	$p=2$	$p=4$	$p=8$	$p=16$
40.401	21.61(22.75)	10.81(11.65)	5.96(6.02)	3.57(3.15)
90.601	49.43(50.02)	24.85(25.43)	12.73(13.01)	7.14(6.71)
160.801	87.67(87.90)	45.07(44.50)	23.06(22.64)	12.21(11.60)
251.001	136.36(136.38)	69.32(68.88)	35.67(34.92)	18.10(17.81)

Finally, we use the model to predict the CPU time for a data set that is not used in the estimation of α and β.

m	$p = 2$	$p = 4$	$p = 8$	$p = 16$
641.601	348.58 (345.44)	175.55 (173.82)	89.16 (87.69)	45.39 (44.39)

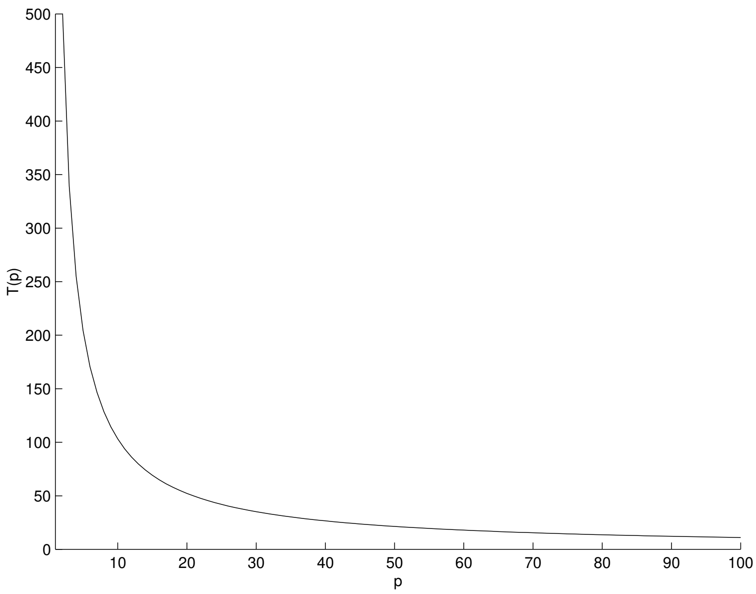
We observe that the estimates given by the performance model is fairly accurate.

Speed-up

Let us put $m = 10^6$, then

$$T(p) \approx 10^3 \left(\frac{1}{p} + \frac{10}{\sqrt{p}}\right),$$

which is graphed below.

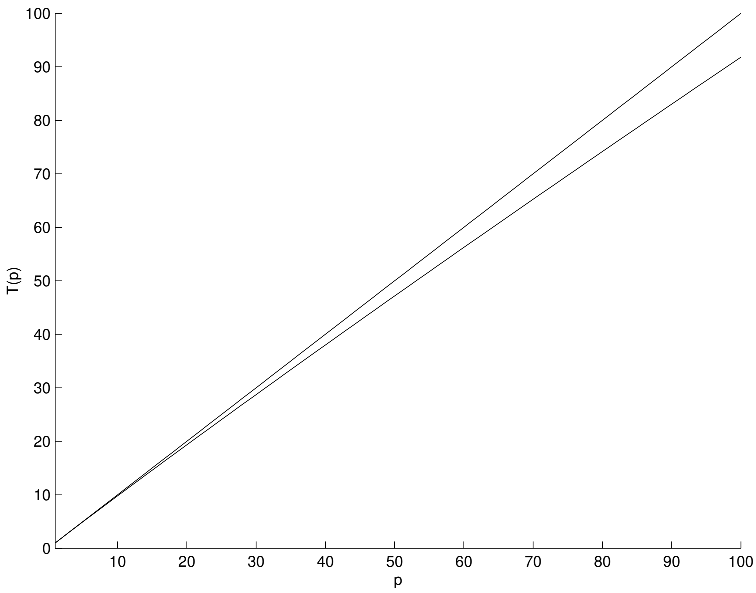


The speed-up, for a fixed value of m, is defined to be

$$S(p) = \frac{T(1)}{T(p)} \approx \frac{m + 10\sqrt{m}}{m + 10\sqrt{mp}}\, p.$$

For $m = 10^6$, we have

$$S(p) = \frac{1010000}{10^6 + 10^4\sqrt{p}}\, p$$



Note that teh speed-up is almost linear for small values of p.

3D

In 3D, we may argue similarly. Suppose we have a computational grid with m nodes and a multicomputer with p processors. We organize the computer in a three-dimensional grid of

$$p = p^{1/3} \times p^{1/3} \times p^{1/3}$$

processors. Similarly, the computational grid consists of

$$m = m^{1/3} \times m^{1/3} \times m^{1/3}$$

nodes.

On each processor there are

$$\frac{m^{1/3}}{p^{1/3}} \times \frac{m^{1/3}}{p^{1/3}} \times \frac{m^{1/3}}{p^{1/3}} = \frac{m}{p}$$

computational nodes. The boundary plane of the nodes located on one processor consist of

$$\sim \frac{m^{1/3}}{p^{1/3}} \times \frac{m^{1/3}}{p^{1/3}} = \frac{m^{2/3}}{p^{2/3}}$$

nodes, so the communication cost of one time step is of order

$$\text{communication} \sim \frac{m^{2/3}}{p^{2/3}}$$

and the computation time is

$$\text{computation} \sim \frac{m}{p}.$$

Hence, the overall computing time is given by

$$T = \alpha\frac{m}{p} + \beta\frac{m^{2/3}}{p^{2/3}}$$

General model

In general, we can argue that the CPU time is

$$T = \alpha \frac{m}{p} + \beta \left(\frac{m}{p} \right)^{\frac{d-1}{d}}$$

where d is the spatial dimension.

Scaling

We observe for the model, that if m/p is kept constant, then also T is constant.