Design of Parallel and High Performance Computing
HS 2013
Markus Püschel, Torsten Hoefler
Department of Computer Science
ETH Zurich

Homework 8
Out: 2013-29-11
Revision: 1

Performance Modelling

Little's Law

Imagine you want to board a train which leaves in 20 minutes. But before you have to buy the train ticket at a counter. You see that there are about 50 people in line before you. Serving a customer takes 40 seconds on average.

What property has to hold for this system to be stable? Will you miss your train?

Roofline Model

The Intel Core i7 2600 CPU has a peak memory bandwidth of 21 GB/s. It is running at 3.4 GHz. It has 4 cores and can carry out up to 16 SP FLOPs/cycle. Draw a roofline plot for this processor. If a program and input combination land on the lower left of the plot, what does this tell you about the program?

Will all program executions yield points which lie either on the diagonal or on the "roof" of the roofline plot?

Balance Principles

Matrix Multiplication

Show that the operational intensity of a tiled matrix multiplication is in the order of $\sqrt{m}$, where m is the cache-size of the processor.

If we assume the matrix multiplication requires exactly n^3 flops and $12 * n^2$ memory operations of byte granularity, is matrix multiplication memory bound or CPU bound on a the machine described above? Does this change (if yes, when) if we assume:

- computational bandwidth of processors doubles every 18 months
- memory bandwidth doubles every three years
- cache size doubles every three years

Stencil Computation

For the following code executed on a single core

```
for (i=0..n)
    for (j=0..n)
        a[i,j] = (a[i+1,j]+a[i-1,j]+a[i,j+1]+a[i,j-1]+a[i,j]) / 5
```

if we increase the floating-point performance by a factor of 2, how much does the cache size m have to be increased to rebalence?

How does this change if we assume many iterations of the above code are carried out, parallelized across multiple cores?

Balance Principles and Multicore

Imagine a processor X_p as a collection of processing elements, connected by a shared bus with bandwidth β. The main memory is also connected to the bus. Each processing element has a local memory (cache) of size m.

We used the processor X_1 to perform matrix multiplication, and m was tuned in such a way that the computation is balanced. Now we increase the number of processing elements, so instead of X_1 we use a parallel version, X_{16}. How should we increase m, so that the computation remains balanced (if β remains unchanged).