Design of Parallel and High Performance Computing
HS 2018
Torsten Hoefler, Markus Püschel **Homework 5**
Department of Computer Science
ETH Zurich

Performance Modeling

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

Assume a NUMA architecture with 2 nodes and a peak memory bandwidth of $B_1 = 74.2$ GB/s. Every node has 4 cores and can carry out up to $P = 332.8$ GFLOPs/sec. Each core runs at 2.6 GHz. If the memory accesses are not well balanced, the peak memory bandwidth becomes $B_2 = 31.32$ GB/s. 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?

Sparse Matrix Vector Multiplication SpMV

The following code compute a Sparse Matrix Vector Multiplication $\overrightarrow{y} = A \cdot \overrightarrow{x}$ between a matrix A (sparse) and a vector $\overrightarrow{x}$ (dense). The matrix is stored in the Compressed Row Storage format.

```
<fill data structures: blockptr, values, col_idx, row_start>

#pragma omp parallel private(i,j,is,ie,j0,y0,thread,bs,be)

thread = omp_get_thread_num()

//Compute the block boundaries
bstart = blockptr[thread]
bend = blockptr[thread+1]

for (i=bstart; i<bend; i++){
  y0=0
  row_start = row_start[i]
  next_row_start = row_start[i+1]

  for (j=row_start; j<next_row_start; j++){
    j0 = col_idx[j]
    y0 += value[j] * x[j0]
  }

  y[i] = y0
}
```

Assume that $\overrightarrow{x}$ and $\overrightarrow{y}$ are kept in cache. The CSR format uses 4byte `integers` to store column indexes. Values are stored using 8byte `doubles`. Compute the operational intensity and check if the code is memory- or compute-bound w.r.t. the previously described architecture (consider only the innermost loop).

You run this code, observing that it reaches a performance up to 5.22 GFLOPS, and you notice that this is mostly due to how the array `value` is stored. Describe an optmization that you can apply to improve the performance.

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.

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 rebalance?

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).