HPC Spring 2017 – Homework 3

Robert van Engelen

Due date: March 30, 2017

1. Determine the FP:M ratio for the following kernels:

   (a) The 2-norm of a vector

   $$||x||_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

   (b) Scalar vector accumulation (DAXPY)

   $$y_i = y_i + a x_i \qquad \forall i = 1, \ldots, n$$

   (c) Matrix-vector multiply (DGEMV) with inner loop j, non-optimized (no unrolling)

   $$y_i = y_i + \sum_{j=1}^{m} A_{i,j} x_j \qquad \forall i = 1, \ldots, n$$

   (d) DGEMM after hoist/sink of matrix C using unroll and jam of the J and K loops by a factor of 2, assuming $K \to \infty$ and therefore the inner loop dominates the FP:M ratio (i.e. the FP:M ratio may be determined from the inner loop):

```
DO J = 1,N,2
  DO I = 1,M,2
    C11 = C(I,J)
    C12 = C(I+1,J)
    C21 = C(I,J+1)
    C22 = C(I+1,J+1)
    DO L = 1,K
      C11 = C11 + A(L,I) * B(L,J)
      C21 = C21 + A(L,I+1) * B(L,J)
      C12 = C12 + A(L,I) * B(L,J+1)
      C22 = C22 + A(L,I+1) * B(L,J+1)
    ENDDO
    C(I,J)     = C11
    C(I+1,J)   = C12
    C(I,J+1)   = C21
    C(I+1,J+1) = C22
  ENDDO
ENDDO
```

2. Suppose we have an $N \times N$ computational grid that is shaped as a 2D torus. We apply a finite difference operation on this grid with the following stencil:

$$Du_{i,j} = \frac{u_{i+1,j} - u_{i-1,j} + u_{i,j+1} - u_{i,j-1}}{4}$$

Furthermore, suppose we divide the grid into P equal-sized $B \times B$ blocks, where P is the number of processors and each processor executes the finite difference operation on its local $B \times B$ grid block. For the grid points on the edges of its block the processor must communicate with other processors to obtain the values of neighboring grid points. Processors are logically arranged in a $\frac{N}{B} \times \frac{N}{B}$ two-dimensional torus of P processors, i.e. $\sqrt{P} = \frac{N}{B}$, where we assume that N is a multiple of B.

We can implement this as follows:

```
for (i = 0; i < B; i++)
{ for (j = 0; j < B; j++)
  { if (i == 0)   { send(PS, u[i][j]); u1 = recv(PN); } else u1 = u[i-1][j];
    if (i == B-1) { send(PN, u[i][j]); u2 = recv(PS); } else u2 = u[i+1][j];
    if (j == 0)   { send(PE, u[i][j]); u3 = recv(PW); } else u3 = u[i][j-1];
    if (j == B-1) { send(PW, u[i][j]); u4 = recv(PE); } else u4 = u[i][j+1];
    Du[i][j] = (u2-u1+u4-u3)/4.0;
  }
}
```

A processor P communicates with its North (PE), South (PS), East (PE), and West (PW) neighbors on the grid that wraps around in both dimensions.

The communication time of one floating point value versus arithmetic is $\gamma = 100$, i.e. we can perform 100 arithmetic operations in the same time it takes one communication step to complete (a "hop" between two processors).

(a) Assume that the physical processor interconnect network is a 2D torus, so the send/recv operations take only one hop between two processors (one send/recv step for a processor P to communicate with its North, South, East, and West neighbors, and there is no additional latency due to contention on the communication links). Determine t_{comp} and t_{comm}. Express t_{comp} and t_{comm} in terms of B and $\gamma = 100$.

(b) For $P = 4^k$ with $k = 1, \ldots, 6$ and $N = 1024$ plot the log-log relative performance prediction graph of t_{comp} and t_{comm}.

(c) From the plot, determine when $t_{\text{comp}} = t_{\text{comm}}$.

(d) Plot the relative speedup $S_P^1 = \frac{t_1}{t_P}$ for $P = 4^k$ with $k = 1, \ldots, 6$ and $N = 1024$. Without latency hiding we have $t_P = t_{\text{comp}} + t_{\text{comm}}$. Note that for $P = 1$ there is no communication, so you $t_1 = t_{\text{comp}}$.

(e) Plot the relative speedup with perfect latency hiding, that is $t_P = \max(t_{\text{comp}}, t_{\text{comm}})$.

(f) Suppose that the physical processor interconnect network is a linear torus. That is, processors are arranged in a network where each processor has only two links, one to its left and one to its right. We will map the 2D torus on the 1D torus by laying out the 2D torus row-by-row on the 1D torus. That is, a processor P has direct links to West and East neighbors, except for processors on the edges, which communicate to neighbors by $\sqrt{P}$ hops to reach the wrap-around neighbors. Messages from a processor P to North and South neighbors take $\sqrt{P}$ hops. Draw a picture where you map processor $P_{i,j}$ on the logical 2D torus to processor P_{j+ki} with $k = \frac{N}{B}$. Given this physical topology, determine the new t_{comm} with $\gamma = 100$.

(g) Plot the log-log relative performance prediction graph of t_{comp} and t_{comm} for $P = 4^k$ with $k = 1, \ldots, 6$ and $N = 1024$.

(h) From the plot, determine when $t_{\text{comp}} = t_{\text{comm}}$.

(i) Plot the relative speedup for $P = 4^k$ with $k = 1, \ldots, 6$ and $N = 1024$ without latency hiding.

(j) Plot the relative speedup for $P = 4^k$ with $k = 1, \ldots, 6$ and $N = 1024$ with latency hiding.