CS 5220 – 2015-09-01 Preclass Questions

Michael Whittaker (mjw297)

August 31, 2015

1. See here.

2. See here.

3. We have t tasks and p stages. Assume each task takes k seconds. In the serial case, we require tk seconds. In the parallel case, we wait k seconds for the first task to finish and then wait $\frac{k}{p}$ seconds for each of the next $(t-1)$ tasks for a total of $k + (t-1)\frac{k}{p}$ seconds. This leads to a total speedup up $\frac{kt}{k+(t-1)\frac{k}{p}}$. If we let the number of tasks tend to infinity, we get

$$\lim_{t\to\infty} \left(\frac{kt}{k + (t-1)\frac{k}{p}} \right) = p$$

4. Serially, we require $1 + 0.5 + 0.25 + 0.5 + 0.5 = 2.75$ seconds. With an arbitrary number of processors, we can compile everything in 2.25 seconds by compiling OpenMPI and OpenBLAS in parallel.

5. Refer to Figure 1 and Figure 2.

6. Refer to Figure 3 and Figure 4.

7. An implementation of the centroid algorithms can be found in `centroid.c`. Algorithm a, b, and c take roughly 70, 130, and 120 milliseconds to process 50,000,000 coordinates. I ran the code on my local computer, not on the cluster.

   To build and run the code, run `make && ./centroid`. It will print the time taken by algorithm a, b, and c in milliseconds followed by the centroids computed by algorithms a, b, and c.

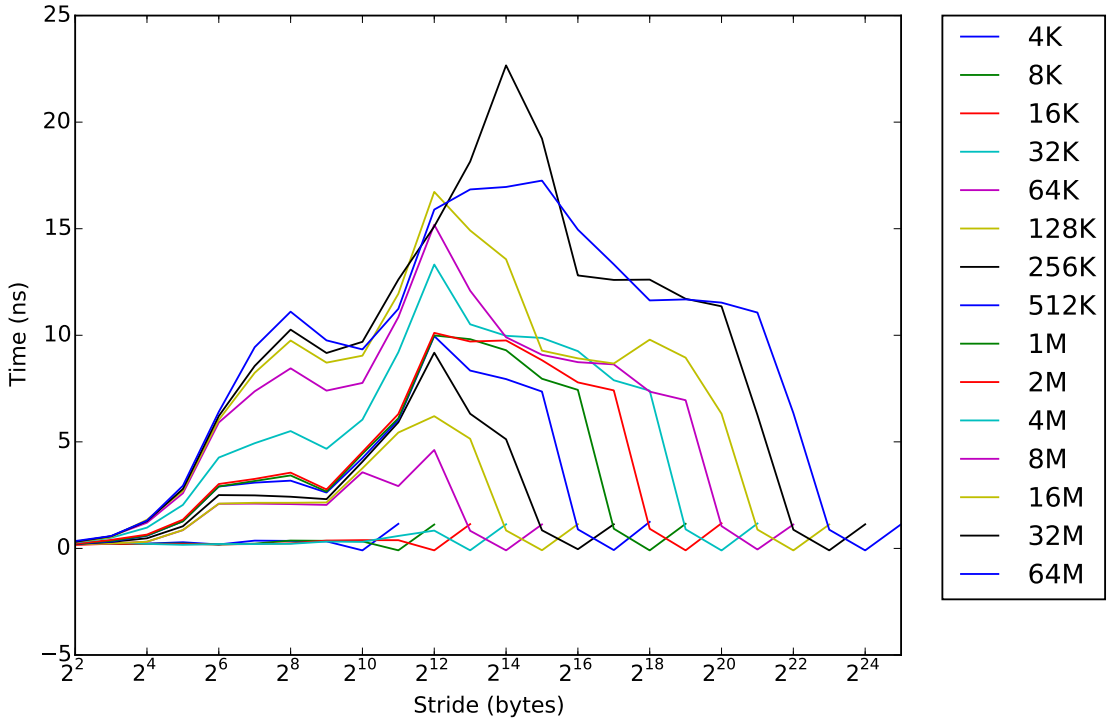

Figure 1: Local membench line plot.

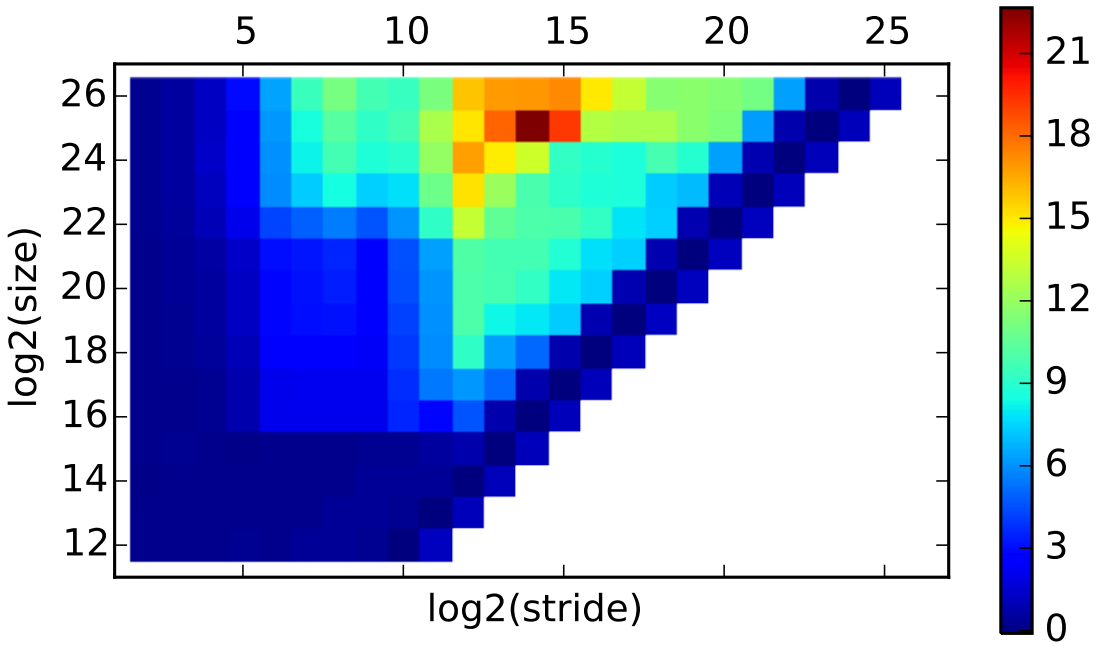

Figure 2: Local membench heat plot.

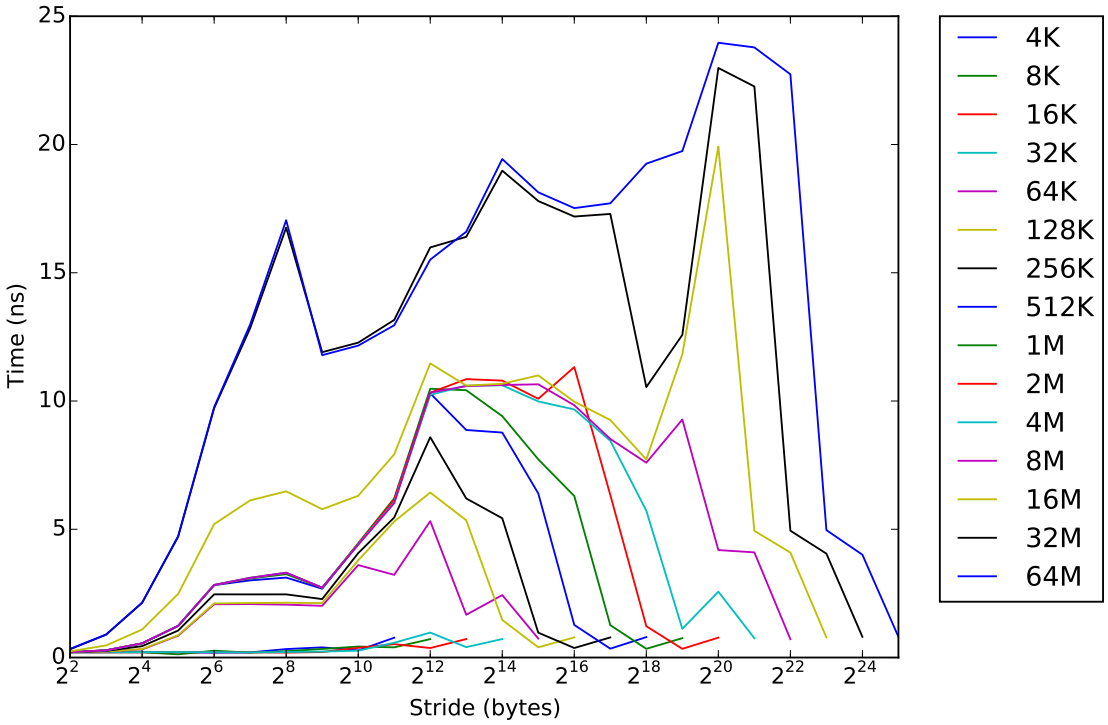

Figure 3: Totient membench line plot.

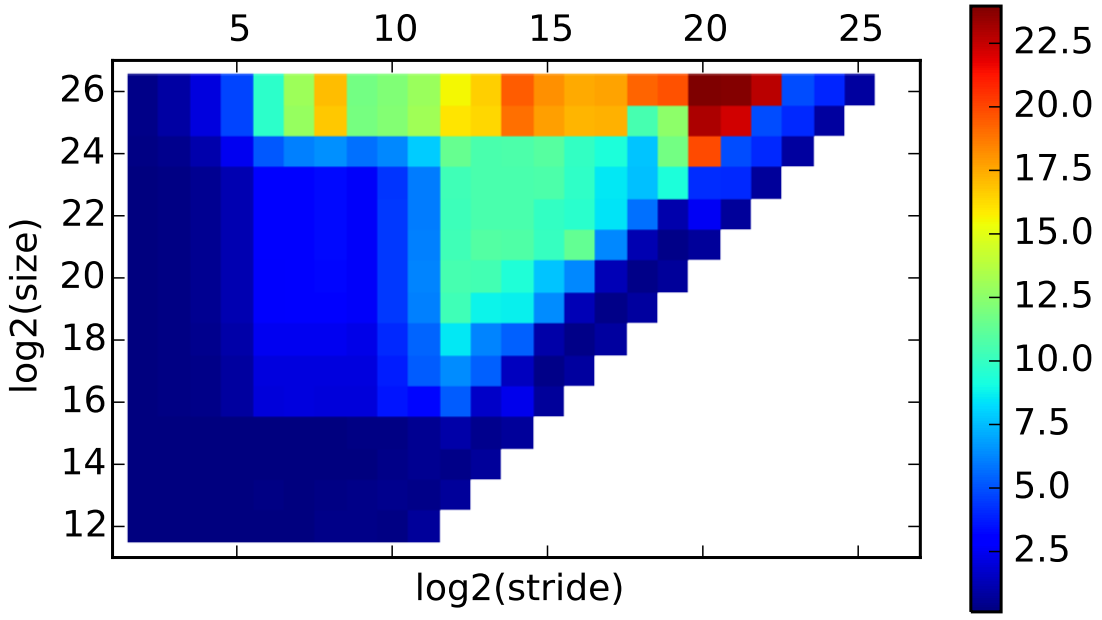

Figure 4: Totient membench heat plot.