

Exercise 2

Shruti Shivakumar

August 28, 2017

Code listing

```
#include <stdio.h>
#include <omp.h>
#include <stdlib.h>
#include <math.h>
#include <unistd.h>
#include <sys/time.h>

#define N 5000000
#define Ntimes 10

double get_walltime(void)
{
    struct timeval tp;
    gettimeofday(&tp, NULL);

    return (double) (tp.tv_sec + tp.tv_usec/1.e6);
}

int main(int argc, char **argv)
{
    int i, j, k, n_t;
    double *A = malloc(sizeof(double)*N);
    double *B = malloc(sizeof(double)*N);
    double *C = malloc(sizeof(double)*N);
    for(i=0;i<N;i++)
    {
        A[i] = 1.0;
        B[i] = 2.0;
        C[i] = 3.0;
    }
    double s=4.0, mintime=0.0, maxtime=0.0, avgtime=0.0, best_rate=0.0;
    printf("T Best Rate MB/s Avg time Min time Max time\n");
    for(k=2;k<10;k++)
    {
        for(j=0;j<Ntimes;j++)
        {
            double time;
            time = -get_walltime();
            #pragma omp parallel num_threads(k)
            {
                #pragma omp parallel for
                for(i=0;i<N;i++)
                    A[i] = B[i] + s*C[i];
            }
        }
    }
}
```

```

        time+=get_walltime();
        if(j==1)
            mintime=maxtime=avgtime=time;
        else if(j>1)
        {
            mintime = time < mintime ? time : mintime;
            maxtime = time > maxtime ? time : maxtime;
            avgtime += time;
        }
    }
    avgtime/=(double)(Ntimes-1);
    best_rate = 3.0*sizeof(double)*N/mintime/1024.0/1024.0;
    printf("%d %11.6f %11.6f %11.6f %11.6f\n", k, best_rate, avgtime, m
}
}

```

Compile with (after optimisation):

```
gcc -O12 -fopenmp -o benchmark benchmark.c
```

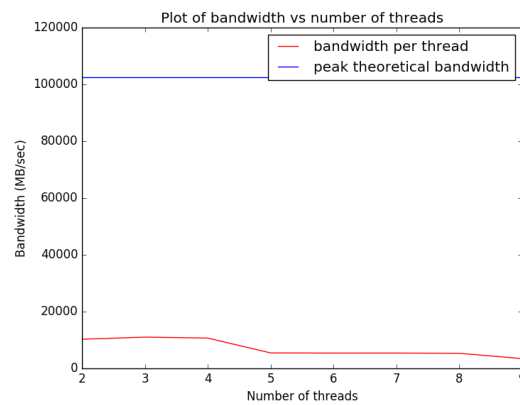


Figure 1: Plot of main memory bandwidth vs number of threads used