HPSC\_Assignment

Name- Kumar Ronit

Student ID- 18M15304

School- School of Engineering

Department- Department of Mechanical Engineering

**Target:** Write a matrix multiplication code with the highest flop/s

**Run** in the login node

**For Compiling:** make

These codes are implementation of BLISLAB for TSUBAME (Xeon E5-2637 v4- Intel for the **login node**).

The task is to optimize SGEMM considering TSUBAME Broadwell microarchitecture.

1. Following are the sizes of different level caches

L1: 128Kbytes

L2: 1Mbytes

L3: 10Mbytes

TLB: 4Kbytes

Referring to the article published by Kazushige et. al. following calculations are made for choosing mc, kc, nc, mr, nr

* kc double precision floating point should occupy half of the page i.e. TLB

Therefore, for single precision **kc = 256** for the configurations mentioned above.

* mc x kc matrix should occupy half the L2 cache. Therefore

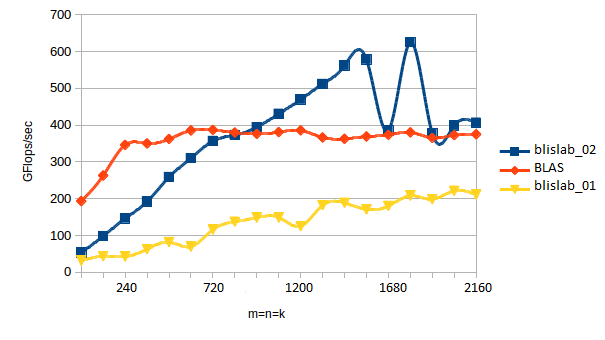
mc x kc x 16 < 0.5 x L2 which gives mc < 128

* kcnr floating point numbers should occupy less than half of the L1 cache and hence kc x nr x 16 < 0.5 x L1 which gives nr < 16
* kc should be chosen relatively large

1. In the first step of blislab, following modifications were made:

* Adding multithreading option around microkernel loop
* Loop unrolling and Register variables while initializing matrix C
* Multithreading using openmp setting OMP\_NUM\_THREADS= bl\_ic\_nt =16 i.e. for 2pkgs x 4 cores/pkgs x 2threads/core

1. In the second step of blislab along with the above modifications assemble code for 16x6 microkernel was incorporated based on the above calculations.



**References:**

1. Jianyu Huang, Robert A. van de Geijin, BLISlab: A Sandbox of Optimizing GEMM, August 31,2016.
2. Kazushige Goto, Robert A. van de Geijin, Anatomy of high-performance matrix multiplication. ACM Transactions on Mathematical Software (TOMS). Volume 34 Issue 3, May 2008, Article No. 12.