University of South Bohemia České Budějovice	
Name	**Tushar Rewatkar**
Personal Number	**B21462**
Course	**UFY/505 – Parallel Programming and Computing**
Assignment	**OpenMP Parallelization of Water Energy Calculation Serial Code**

For the 100k.gro file

Sr No.	Scheduling method	Wall Time/s	Load Imbalance
1	Static	5.792681	1.833338
2	Static, 1	4.233338	0.000220
3	Static, 100	3.202990	0.021996
4	Static, 10000	6.787462	2.000000
5	Dynamic, 100	2.966360	0.020495
6	Guided, 100	2.937857	0.024560
7	Auto	3.042528	0.020977
8	Serial	27.266178	N. A
9	MPI_Parallel (1 threads)	55.341373	0.000000
10	MPI_Parallel (12 threads)	6.895038	0.000220

All Output were taken from meta centrum using qsub commands.

Scheduling Methods

The **static schedule** is characterized by the properties that each thread gets approximately the same number of iterations as any other thread, and each thread can independently determine the iterations assigned to it. Thus, no synchronization is required to distribute the work, and, under the assumption that each iteration requires the same amount of work, all threads should finish at about the same time.

The **dynamic schedule** is characterized by the properties that each thread gets varying, or even unpredictable, amounts of work, and no thread waits at the barrier for longer than it takes another thread to execute its final iteration.

The **guided scheduling** type is similar to the dynamic scheduling type. OpenMP again divides the iterations into chunks. Each thread executes a chunk of iterations and then requests another chunk until there are no more chunks available.

The **auto** scheduling type delegates the decision of the scheduling to the compiler and/or runtime system.

Effect of Scheduling method on Load Imbalance

As you can see from table above, Schedule (static,1) is providing most optimized load balance as compared to schedule(static) and schedule(static,100). When schedule(static,10000), you can see the threads are most unevenly distributed i.e., npairs on some threads are 0, that is why load imbalance is highest. For schedule(dynamic,100), schedule(guided,100) and schedule(auto) all three are having almost same load imbalance. So, to conclude static scheduling with chuck size 1 is having most optimized load balance that means most of tasks are equally distributed on 12 different threads which give us the lowest load imbalance.

OpenMP vs MPI vs Serial

For 100k.gro file, the wall time for the OpenMP version of schedule(static,1) is 4.233338 whereas for MPI version with 12 threads is 6.895038 sec with same load balance of 0.000220. Also, when you compare Serial version with OpenMP the wall times are 27.266178 and 5.792681 respectively. You can see the large difference in Wall time Calculation in Open MP and Serial. Even if you consider any scheduling techniques of OpenMP the wall time is less than MPI version with 12 no. of threads and much lesser than serial version of code. This shows that OpenMP takes a shorter period for completion of task followed by MPI version and serial version takes the longest one.