**Project Proposal**

**Names of team members:** Mohit Palliyil Sathyaseelan, Vishnu V

**Project title:** Parallel Sort of N keys in Integer Sort kernel (NAS parallel Benchmarks)

**Abstract:**

The goal of this project is to sort N keys in parallel using OpenMP and MPI. The N keys are generated anonymously with a sequential key generation algorithm. The algorithm uses a pseudorandom number generator. After the keys are generated by the key generation algorithm, they are sorted using the partial and full verification tests. This benchmark tests a sorting operation that is important in “particle method” codes.

**Proposed approach:**

The project is done using Standard NASA MPI benchmark suite. The IS kernel is being used to do parallel sort over N keys. For the performance of the benchmark, the initial generation of the keys will impact the number; hence we will run multiple tests and take the average value of the results. To compare the performance, both shared and distributed memory models are being used. As a part of this project, we will study the benchmark and compare the performance of the different memory models.

**Team members expertise:**

1. Mohit Palliyil Sathyaseelan: Specialization - Electrical & Computer Engineering (Embedded Systems and IoT, C, Python)
2. Vishnu V:Specialization - Electrical & Computer Engineering (Computer system design, C, Python)

**Demonstration plan:**

1. Initial study of the algorithm: A sequence of keys is said to be sorted if they are arranged in non-descending order. The sorting algorithm used in the IS kernel is to be studied in great detail. The sorting algorithm initially uses a key generation algorithm to generate the keys in a scalar sequential manner. This involves the use of a pseudorandom number generator. Then, using an appropriate memory mapping system, they are stored in memory. The memory mapping should be done in shared or distributed memory. The keys are to be then sorted using partial and full verification tests. This forms the integral part of the algorithm and sorts the keys in a non-descending order.
2. Variation of Memory Models: The keys will be stored in shared global memory or distributed memory. In shared global memory, the keys are first stored in a contiguous address space, every word will have its own respective address space. In the case of distributed memory with say *n* units, each unit will store *k* units where *k* is the total number of keys divided by *n*.
3. Comparison: The performance variations are observed to be compared with theoretical expectations and justification for anomalies to be found.

**Proposed schedule for the remainder of the semester:**

22/3/2022 - 4/6/2022

1. Understanding Kernel IS, make files, Readme files mentioned in the NAS NPB
2. Variation and testing of shared and distributed memory
3. Preparation of Project progress presentation slides.

4/6/2022 - 4/25/2021

Completion of project report by statistical analysis and comparison of data obtained from the project progress.

**References and Links:**

<https://www.nas.nasa.gov/software/npb.html>

<https://www.davidhbailey.com/dhbpapers/npbsc92.pdf>

<http://pages.di.unipi.it/mencagli/publications/preprint-pdp-2018-nas.pdf>