# Master Thesis Topic: Scheduling Map-Reduce like Jobs in Large Compute Clusters

**Background**

Today, big data applications are being processed at a massive scale on large sale computing clusters such as data centres. Highly efficient systems like Map-Reduce, Spark etc., have been successfully deployed in these clusters to manage resources and schedule applications with embarrassingly parallel workloads. At a high level these systems follow a master-slave architecture, where the workload is divided into tasks with each task processing an independent chunk of data. Despite knowing the statistics of these tasks (e.g. size of the data to be read, number of instructions to execute) a priori, straggling tasks arise due to random factors of variation associated with the slave machines [1]. To combat the straggling tasks problem *speculative execution* has been proposed. Two key and independent approaches used in speculative execution are 1) *Restart* - kill a task on a slave machine and restart it on another slave machine 2) *Redundancy* - schedule multiple copies of the same task on different slave machines and when a copy finishes then cancel all other copies.

Restart does not waste resources as it completely kills a task on a machine before that task is scheduled on another machine. However, it requires a “good” estimate for when to kill a task on a machine. On the other hand, Redundacy does not require any estimate, but wastes the machine resources as a single task will be using multiple machines, simultaneously. Both Restart and Redundancy are widely used in practice. However, existing research works study schemes that either use Restart or Redundancy. Also, the analysis of these schemes are limited to idealized models. Very few works study schemes that use Restart in conjunction with Restart, e.g., [1], [2]. In this project we will explore the advantages and disadvantages of schemes that use Restart or Redundancy. We will then aim at proposing schemes that potentially use both Restart and Redundancy. Finally, we will analyse the proposed schemes and prove performance bounds.

**Task**

**Part I (Mandatory):** Understand the problem of stragglers and do a literature review on existing Restart and Redundancy schemes. Propose new algorithms that potentially combines the virtues of both Restart and Redundancy. Verify the performance of the proposed algorithms using trace driven simulation. For the trace driven simulation, statistics of jobs from the Google Cluster traces will be provided.

**Part II:** If the proposed algorithms perform better than the existing alternatives, the student may try and prove performance bounds for the algorithms.

**Required Skills:** The student must have good background in probability and random processes and have taken a basic course in Algorithms (can be a part of programming course). He/She should have strong coding skills in any one of the languages (e.g. C/C++/JAVA/Python). A basic knowledge in Python is an added advantage and helps in manipulating the data from Google Cluster traces.

## Contact

If you are interested in the project please email to Jaya Prakash Champati at jpra@kth.se

## References

[1] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in map-reduce clusters using mantri," in *9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10)*. Vancouver, BC: USENIX Association, 2010.

[2] G. Ananthanarayanan, M. C.-C. Hung, X. Ren, I. Stoica, A. Wierman, and M. Yu, "Grass: Trimming stragglers in approximation analytics," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. Seattle, WA: USENIX Association, 2014, pp. 289–302.