**PhD Dissertation Title:** Coded approaches for straggler-resilient distributed matrix computations: Numerical stability, sparsity and leveraging partial computations

**Abstract:** Distributed matrix computations over large clusters often suffer from the problem of slow or failed worker nodes (called stragglers) which can dominate the overall job execution time. Recently, different coding techniques have been utilized to mitigate the effect of stragglers where worker nodes are assigned the job of processing encoded submatrices of the original matrices. These works utilize concepts from erasure coding to mitigate the effect of stragglers by running “coded” copies of tasks comprising a job; stragglers are typically treated as erasures. However much of prior work in this area is (i) either sub-optimal in terms of its straggler resilience, (ii) suffers from numerical problems, i.e., there is a blow-up of round-off errors in the decoded result owing to the high condition numbers of the corresponding decoding matrices, (iii) requires dense linear combinations of submatrices of the original matrices which destroy their inherent sparsity of the `input' matrices (when applicable), or (iv) treats slow nodes as erasures, so ignores the potentially useful partial computations performed by them. This is problematic as it results in significantly higher worker computation times while suffering from significant numerical stability issues.

In this work we present different schemes that remove these limitations. First we propose a convolutional coding approach which is optimal in terms of its straggler resilience, and can be decoded using a fast peeling decoder that only involves add/subtract operations. The numerical robustness of the proposed random convolutional code approach can be theoretically quantified by deriving a computable upper bound on the worst case condition number over all possible decoding matrices by drawing connections with the properties of large block Toeplitz matrices. Our next approach based on cyclic placement allows us to leverage partial computation by stragglers while imposing constraints on the level of coding that is required in generating the encoded submatrices. Our scheme can continue to have the optimal threshold of prior work and also allows us to trade off the straggler resilience with the worker computation speed for sparse `input' matrices. Extensive numerical experiments done in AWS (Amazon Web Services) cluster support our findings and confirm the superiority of proposed approaches to other available methods.