My research interest broadly lies in the domains of Distributed Systems, Clouds, and Cluster Job Scheduling. My work spans scheduling distributed jobs, scheduling data center network flows, and learning runtime properties of entities in the cloud online. More recently, I have been exploring transfer learning and its application for online learning.

1. Contributions from Ph.D. Research

For my Ph.D. thesis [6], I studied the scheduling and learning problems in distributed systems. I have six (5 + 1) first-authored papers published in prestigious peer-reviewed conferences and journals from my thesis. They are in USENIX NSDI [1], IEEE ToN [2], and USENIX ATC [3], ACM CoNEXT [4], USENIX Hot Cloud [5]. Another of my papers is under review at IEEE ToCC. The professors I worked with have received an NSF grant that derives its motivation from my Ph.D. thesis work.

1.1 Problem and Challenges

Use of shared clusters for various purposes is growing rapidly. One of the primary uses is to schedule distributed data analytics jobs. Such jobs typically arrive online and compete for shared resources. With the growing business impact of distributed big data analytics jobs and shared clusters, optimizing their execution and resource consumption has become crucial. To best exploit the cluster and to ensure that jobs also meet their service level objectives, efficient job scheduling is essential. Since jobs arrive online, their runtime characteristics are not known a priori. This lack of information makes it challenging for the scheduler to determine the right order for running the jobs that maximize resource utilization and meet the application service level objective (SLOs). Workload with mixed SLOs further increases the challenge.

A key feature in which a typical distributed data analytics job differs from standard job scheduling is their massive parallel structure. In a typical distributed data analytics job, there are multiple phases e.g., a MapReduce job has three phases - map, shuffle (communication) and, reduce. Each of these phases has multiple entities, e.g., multiple tasks in the map and reduce phase, and multiple flows in the shuffle phase. However, all of these components are working towards the same goal, i.e., job completion. As a result, for job completion, all components of all of its phases should finish. Also, the phases may have a dependency on the previous ones, e.g., in principle, in map-reduce job’s shuffle cannot begin till the map finishes, and reduce cannot begin till shuffle finishes. This makes synchronization across phases as well as within phases important for efficient execution.

The above two challenges give rise to an interesting cluster scheduling problem, which can be seen in two steps:

1. We need to learn runtime properties of jobs online, and use them for more efficient scheduling
2. Cluster scheduler should ensure synchronization across execution of different sub-entities.

1.2 Our Solution

Our solution for the above mentioned cluster scheduling problem can be categorized into following two contributions:

1. A New Class of Online Learning Algorithms based on Sampling the Spatial Dimension.
2. New Multi-Task Scheduling Algorithms by Synchronizing the Spatial Dimension.
1.2.1 Contribution 1

An effective way to tackle the challenges of cluster scheduling is to learn the runtime characteristics of pending jobs, as accurately estimating job runtime characteristics allows the scheduler to exploit offline scheduling algorithms that are known to be optimal, e.g., Shortest Job First for minimizing the average completion time.

Prior work in this area has focused primarily on offline learning techniques and has relied on learning from historical execution. These approaches have well-established drawbacks like data not keeping up with new changes. Keeping up with the rapidly evolving cluster environment with old data is challenging. Further, these works cannot capture a real-time dynamic cluster atmosphere. Finally, most of the existing works have directly translated the algorithms designed for single CPU systems to the domain of distributed systems. My work addresses these open questions with a novel observation in my thesis. That is, distributed jobs have an important spatial dimension, an inherent property of distributed jobs. My thesis presents the first study highlighting it and underscoring its importance in efficient cluster job scheduling. I developed two new classes of spatial dimension-based algorithms to address the two primary challenges of cluster scheduling.

First, in my thesis, we proposed, validated, and designed two complete systems that exploit spatial dimension learning algorithms. The work demonstrates high similarity in runtime properties between sub-entities of the same job by detailed trace analysis on four different industrial cluster traces. The thesis identifies design challenges and proposes principles for a sampling-based learning system for two examples, first for a coflow scheduler and second for a cluster job scheduler. Along with theoretical proofs to validate the idea, it also includes extensive simulation and real-testbed evaluation of this learning scheme against two major learning approaches: History-based offline learning and Least-Attained-Service (LAS) based online learning. The new approach gained 1.56x and 1.51x improvement in average completion time respectively. In comparison to LAS based we can reduce the master-slave control message by upto 10 times. As compared to history based learning we reduce the prediction error by 2x in the median case and upto 9x in 90th percentile.

1.2.2 Contribution 2

My thesis also proposes, provides design for, and demonstrates the effectiveness of new multi-task scheduling algorithms based on effective synchronization across the spatial dimension. The work underlines and validates, by experimental analysis, the importance of synchronization between sub-entities (flows, tasks) of a distributed entity (coflow, data analytics jobs) for its efficient execution. The work also highlights that not considering sibling sub-entities when scheduling something may also lead to suboptimal overall cluster performance. Second, when taking into account the spatial dimension of distributed entities, we observe that SJF (based on the total size) is not optimal in the first place. Intuitively, in a single-CPU job scheduling of N jobs, scheduling any job to run first will block the same number of other jobs, N − 1. In scheduling distributed entities across multiple machines, however, since different entities have different numbers of sub-entities distributed at the different machines, scheduling a different entity (its sub-entities) first can block a different number of other entities (at the machines where its sub-entities lie). We denote this degree of competition as “contention”. In other words, the waiting time of other entities will depend on the duration as well as the contention of the entity across its machines. SJF only considers total duration, and ignores the contention which can result in poor completion time. Hence, we propose use of Lease-Contention-First policy. The thesis proposes and provides the design and implementation of a full coflow scheduler based on these assertions.
2. Contributions from Research at Nokia & Bell Labs
At Nokia Bell Labs, I have worked on applying transfer learning for CSI configuration and updates online. This work at Nokia is proprietary, and I cannot reveal further details. We have one paper at IEEE WCNC’22. I am also working on patent applications from the project.

3. Future Research Plan
For future research, I am interested in the general area of distributed systems. More specifically, I am interested in exploring the systematic use of spatial dimension for real-time learning and other performance improvement aspects of distributed systems. I am summarizing some of the key areas where the use of “Spatial Dimension” can be further explored.

3.1 Extending the idea of Spatial Dimension

3.1.1 Combining Sampling-based and History-based learning
The sampling-based approach has its own limitations. For example, for a job with a very high skew across tasks, sampling may not perform the best in all cases. Similarly, for some thin jobs (jobs with very few tasks), delay due to sampling might be unaffordable. The same may apply to multi-phase jobs. An approach to solve such issues could be to merge sampling-based learning approaches, defined in [1] and [2], with history-based learning. The merged approach could be helpful in several ways: (1) Cases where skew across the spatial dimension is high. For example, we can use historical data to apply error correction or predict the expected skew – In some cases, a history-based prediction could be better for high skew jobs. (2) Cases where there are multiple phases in one distributed work, e.g., multi-phase data analytics jobs like map-reduce jobs, as the delay due to sampling in each phase could add up significantly. We sample only in one phase and extrapolate it using historical data. I plan to explore the space of merging the two approaches further and design systems that could practically apply them.

3.1.2 Transfer Learning
The use of Neural Networks is snowballing and is solving many problems. However, keeping the model up to date is challenging with fast-evolving user behavior, environment, hardware, and software configuration. Additionally, the unavailability of data, specially labeled data, is also a challenge. The “Transfer Learning” method helps to resolve this issue. In this method, we update an existing pre-trained model using a relatively very small (~1%), compared to the initial dataset, sample of training data to update. We can exploit spatial dimension and transfer learning to collect small training samples from each job and update a pre-trained neural network model for each newly arriving job. A similar approach can be extended to any generic distributed entity to update a pre-trained neural network model using transfer learning methods.

3.2 Making Sampling-based approach more practical

3.2.1 Generalizing the approach for other SLOs
The benefit of using sampling-based learning techniques for scheduling to optimize average completion times [1, 2] is that we need to predict the average of task runtimes and accuracy only matters to get the correct ordering of jobs in the systems. However, these benefits will not apply to the cases where the optimization goals depend on maximum task length like deadline. For such cases, we need to take additional steps like error correction, different statistical methods for predicting, etc. I plan to pursue further in this direction to extend spatial dimension based sampling-based learning techniques for other SLOs such as deadlines, both soft and hard, optimizing makespan, etc.

3.4 Theoretical Analysis
In my thesis, we have performed theoretical analysis on simplified jobs and distributed systems models. I want to extend those analyses in collaboration with experts in theory on more generic and realistic models for jobs and computing clusters (like job arrival rate, workload
distribution, etc.). It will be helpful to get tighter bounds on sampling-based predictions and runtime with more realistic assumptions.

4. Professional Service Activities

I have always been inclined to volunteer for the community. That has translated to professional volunteering as well. I served on the TPC of IEEE ICNP 2021, IEEE ICNP 2022. I am also a reviewer for the journals IEEE ToCC and IEEE ToN and IFIP Networking '22. I have been a session chair and shepherd for IEEE ICNP 2021. I also volunteer to give talks on research and academic topics to students at various colleges. So far, I have spoken at 12 different colleges; this also includes invites by ACM chapters; I am also serving on Ph.D. committees. During my Ph.D. time, I was a Computer Science Graduate Student Board member and a Senator in Purdue Graduate Student Government for two terms. I also volunteer for an organization that aims to facilitate professional networking among Indian-origin scholars, professionals, and students to enable professional progress.

References


[6]: A. Jajoo, “Exploiting the spatial dimension of big data jobs for efficient cluster job scheduling,” Purdue Univ. Graduate School, 2020


