THE UNIVERSITY OF MELBOURNE

Research Experience

- **Net Neutrality (MS):** network protocols, protocol blocking, content shaping
- **Cloud and Big Data (PhD):** optimization, performance modelling, resource allocation, job scheduling, reinforcement learning

- **Software Defin** -driven resilient tactical battlefiel

- **Stream Computing (Post Doc)** ocial media data analytics, in-memory caching d

[Muhammed Tawfiqul Islam - Google Scholar]

- **Objectives**
  - Scheduling Big Data Applications in a cloud-deployed cluster, while reducing the cost of VM usages of the whole cluster, prioritize critical/deadline-constrained applications
  - Scheduling Big Data Applications in a hybrid cluster composed of local and Cloud VMs, le e cost, provide deadline guarantee

- **Limitations of Existing approaches**
  - Homogeneous VM assumption leads to resource wastage
  - Performance-aware, but not Cost-efficient
  - No separation between normal and time-critical jobs
  - Multiple executors cannot be placed in the same VM
  - Does not consider pricing model of different VM instance types, and cost efficiency

- **Research Contributions:**
  - Four Job Scheduling Algorithms which prioritize critical jobs and tightly pack jobs in fewer VMs to reduce cost
  - Real implementation of a job scheduling framework on top of Apache Mesos Cluster Manager. Can be extended to add new policies.
  - RM_Simulator: event-based simulator for simulating scheduling policies for big data applications
  - Experiments on Apache Spark Jobs

Problem Formulation (Cloud-based Cluster)

Example scheduling scenarios

Proposed Algorithms

- **Solution Approach (cloud-based cluster):**
  - Best-Fit-Heuristic (BFD): Unifies resource dimensions (CPU, Memory), finds a placement of a job which is cost-effective, and reduces unused resources
  - Integer Linear cking of jobs with cost-minimizati

- **Solution Approach (hybrid**
  - First Fit Heuristic (FF): Use local, then Cloud
  - Greedy Iterative Optimization (GIO): Relaxes the problem from per-job to per-executor basis, uses the pricing model of VMs and job profile information to find the cheapest placement for each executor

System Implementation

RL-based Job Scheduling

- **Limitations of Existing approaches**
  - Cannot learn cluster or application characteristics for efficient optimization of objective
  - Need to be tuned for different scenarios
- **Research Contributions:**

  - RL Model for the j

es)
  - Reward formulatio
  - RL environment implementation for oyed cluster

  - DRL agents (DQN and REINFORC erent characteristics

- **Solution Approach:**
  - Set expected balance between cost-optimized and time-optimized objective
  - DRL agents learn to schedule and optimize objectives entirely by continuous interaction with the cluster simulation environment

RL Model

- Agent observation is made from job requirements and cluster resource details
- Agent takes an action
- Receives a reward and observes another state
- Learns through intera environment
- Agent has no prior knowledge of job arrival, job type, resource constraints, objectives
- Maximizing expected reward = optimizing target objectives
- **Built and trained on TensorFlow Agents framework.**

Performance Evaluation

- Trade-offs between multiple objectives

Questions?

Muhammed Ta olar

For any queries: tawfiqul.isla b.edu.au