CSc8220: Project Assignment

Due on April. 29, 2019, 12:30pm

- The goal of the project is to explore the the emerging research problems related to advanced computer networks. All the submitted materials such as the abstract and report must use word or latex following the IEEE template (single space, 11 pt font size). No copy and paste are allowed in the report. Students will group into teams of up to three to carry out the course project. You will be responsible for completing the work if some members drop the class.

- Each team (up to three students) will select one specific problem from the list of ***suggestive project topics*** below. The team then proposes new approaches/algorithms for the problem or provide a new/better/extended approach to the problem. The project can solve the selected research problem through mathematical/analytical or experimental or a solid combination of the two. *The team must implement the proposed approach/algorithm.* You are free to choose any programming platform but must ensure that your code is readable with proper indentation and good comments. You should also give a demo to the TA to ensure that your code is running and the results are up to the marks. 10% bonus would be given if you implement and compare your proposed approach/algorithm with one existing literature approach/algorithm. The final paper must include the problem definition, various solutions, implementations and comparison of the results. Hence, a graph that compares the results of these approaches/algorithms may also be included in the final paper. The final report should be at least 5 pages (not including citations, 11pt font) following the template.

- On Mar. 11, 12:30pm, teams must submit a 3-page project progress report. Feel free to drop by during instructor office hours to discuss and develop your project ideas further. The progress report should include the following items. Name your file something like NFV-progress.doc.
  - Introduction
  - Detailed problem statement
  - Motivation: Why is the survey/problem interesting? Why is the survey/problem important?
  - Related work: brief literature review (does not have to be complete, but it has to show how much you've read)
  - Your plan of attack: milestones, division of work, etc.
  - Research progress, algorithms to be implemented and compared, papers selected and the coding
  - Submission: hard copy to be handed in class and the soft copy should be submitted to the CSc8220 dropbox on https://gastate.view.usg.edu/.

- Starting from Mar. 25, each team will be scheduled to make a ~25 min presentation on the project progress. report. The ***progress report schedule*** is listed below.
  - 3/25: Team xxx and Team xxx
  - 3/26: Team xxx and Team xxx
  - 4/1: Team xxx and Team xxx
  - 4/3: Team xxx and Team xxx
  - 4/8: Team xxx and Team xxx
- Apr. 17- 29, teams will make a ~30min final project presentation.
- The soft copy of project presentation should be submitted to the CSc8220 dropbox on https://gastate.view.usg.edu/.

- On April 29, 12:30pm, a hard copy of the final report must be handed in and a soft copy of the whole project package including source codes, figures, latex files must be submitted to the CSc8220 dropbox on https://gastate.view.usg.edu/. The final report should be at least 5 pages (not including citations, 11pt font) following the template.

Suggestive project topics

Datacenter Networks (DCN)

Scheduling jobs across geo-distributed datacenters with max-min fairness

Abstract: It has become routine for large volumes of data to be generated, stored, and processed across geographically distributed datacenters. To run a single data analytic job on such geo-distributed data, recent research proposed to distribute its tasks across datacenters, considering both data locality and network bandwidth across datacenters. Yet, it remains an open problem in the more general case, where multiple analytic jobs need to fairly share the resources at these geo-distributed data-centers. In this paper, we focus on the problem of assigning tasks belonging to multiple jobs across datacenters, with the specific objective of achieving max-min fairness across jobs sharing these datacenters, in terms of their job completion times. We formulate this problem as a lexicographical minimization problem, which is challenging to solve in practice due to its inherent multi-objective and discrete nature. To address these challenges, we iteratively solve its single-objective subproblems, which can be transformed to equivalent linear programming (LP) problems to be efficiently solved, thanks to their favorable properties. As a highlight of this paper, we have designed and implemented our proposed solution as a fair job scheduler based on Apache Spark, a modern data processing framework. With extensive evaluations of our real-world implementation on Amazon EC2, we have shown convincing evidence that max-min fairness has been achieved using our new job scheduler.

Multi-tenant multi-objective bandwidth allocation in datacenters using stacked congestion control

Abstract: In datacenter networks, flows can have different performance objectives. We use a tenant-objective division to denote all flows of a tenant that share the same objective. Bandwidth allocation in datacenters should support not only performance isolation among divisions but also objective-oriented scheduling among flows within the same division. This paper studies the Multi-Tenant Multi-Objective (MT-MO) bandwidth allocation problem. To our best knowledge, no existing practical work support performance isolation and objective scheduling simultaneously. We propose Stacked Congestion Control (SCC), a distributed host-based bandwidth allocation design, where an underlay congestion control (UCC) layer handles contention among divisions, and a private congestion control (PCC) layer for each division optimizes its performance objective. Via the tenant-objective tunnel abstraction, SCC achieves weighted bandwidth sharing for each division in a distributed and transparent way. By adding a rate-limiting send queue in the ingress of each tunnel, mechanisms between performance isolation and objective scheduling are completely decoupled. We evaluate SCC both on a small-scale testbed and with large-scale NS-2 simulations. Compared to the direct coexistence cases, SCC reduces latency by up to 40% for Latency-Sensitive flows, deadline miss ratio by up to 3.2× for Deadline-Sensitive flows, and average flow-completion-time by up to 53% for Completion-Sensitive flows.