VM Selection for Financial Exchanges in the Cloud

Daniel Duclos-Cavalcanti (Student),
Muhammad Haseeb (NYU Supervisor), Navidreza Asadi (TUM Supervisor)
Chair of Communication Networks, School of CIT
Technical University of Munich (TUM)

1 Overview

Financial exchanges have shown interest in migrating their current infrastructure to the public cloud, potentially allowing for greater scalability, robustness, and lower costs. In contrast to the exchange's current on-premise data centers, the public cloud does not offer native mechanisms for fair and performant multicast delivery. Disseminating market data to market participants (MPs) both fast and almost simultaneously is a needed requirement to ensure a fair market.

Recent work, namely Jasper [1], addressed this by presenting a scalable and fair multicast solution for financial exchanges in the cloud. It does so via the employment of an overlay multicast tree, clock synchronization, kernel by-passing, and more. Jasper achieves considerable performance and fairness, and attempts to address the high network-latency variance in the cloud through custom techniques such as hedging. However, there is room for improvement by optimizing in-host latency via confronting the issue of inconsistent VM performance in the cloud.

LemonDrop [2] tackles this very problem of discrepant VM behavior within identical instance configurations in the cloud. LemonDrop selects and schedules a subset of VMs optimized for an application's latency needs, by quickly detecting under-performing ones (*Lemons or Stragglers*). It frames the selection and scheduling of VMs as a Quadratic Assignment Problem (QAP), minimizing requests among VMs by reconfiguring their relative placement and dropping individual ones completely. To accomplish this, LemonDrop uses knowledge of an application's request patterns and real-time intra-VM latency to obtain its proposed configuration.

Stragglers can likely affect Jasper's overall system performance drastically. This proposed work aims to develop a simpler heuristic than LemonDrop's, that achieves reasonably good results adapted to the smaller problem set of a multicast tree. We believe that due to the known high latency variance in the cloud, LemonDrop's approach of using all-to-all latency measurements is an overly-complex solution that may capture an unrealistic snapshot of the cluster's behavior.

2 Objectives

Develop a VM Selection Heuristic for tree-like networks in the cloud.

1. Develop Server-Client Manager application.
   (a) Server: Connects to all client nodes and applies requests onto them:
      i. Launch a process and store it's PID.
      ii. Report data on a previously ran process.
      iii. Kill a previously ran process.
   (b) Client: Waits on Server's connection and reacts to requests.
2. Apply Testbench through Server-Client Manager:
   (a) Server allocate a pool of 'N' VMs in the cloud and stores initial configuration.
   (b) Server is able to command client nodes to run, terminate and report on:
      i. Jasper
      ii. Intra-VM OWD Latency Measurements
      iii. LemonDrop
   (c) Server can change change cluter configuration via killing and relaunching processes.
3. Develop Heuristic:
   (a) Recreate tree by iterativally:
      i. Benchmark intra-VM latency and choose best node among the pool.
      ii. Place/Assign node to the next available slot in the tree.

3 Experimental Setup

Cloud stack deployment, node benchmarking and heuristic development/formulation would be done via Google Cloud Platform (GCP) credits provided by Dr.Sivaraman's and his team at Systems@NYU.

References

[1] M. Haseeb, J. Geng, U. Butler, X. Hao, D. Duclos-Cavalcanti, and A. Sivaraman, "Jasper: Scalable and fair multicast for financial exchanges in the cloud," 2024.

[2] V. Sachidananda, *Scheduling and Autoscaling Methods for Low Latency Applications*. Stanford University, 2022.