Secure Multiparty Computation
Sprint 1

Developer | Hasnain Abdur Rehman| hasnain@bu.edu
Developer | Pierre-François Wolfe | pwolfe@bu.edu
Developer | Samyak Jain | samyakj@bu.edu
Developer | Suli Hu | sulihu@bu.edu
Developer | Yufeng Lin | yflin@bu.edu
Mentor/Client | John Liagouris | liagos@bu.edu
Mentor/Client | Vasiliki Kalavri | vkalavri@bu.edu
Subject-Matter Expert | Mayank Varia | varia@bu.edu

BOSTON UNIVERSITY

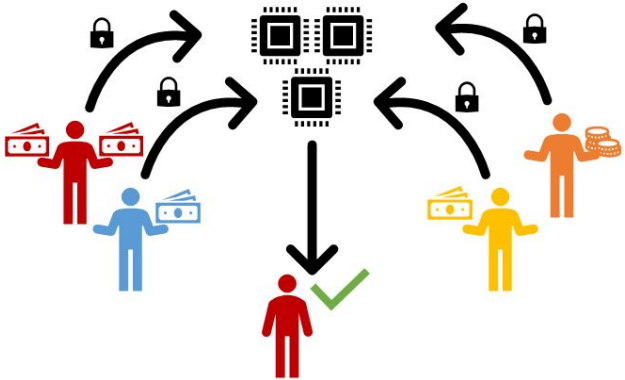
Presentation Outline

- Multi-Party Computation (MPC) Overview
- Mentor/Client Motivation
- Project Goals & Sprint 1 Stories/Tasks
- Work Accomplished & Information Learned
  - VMs (OpenStack)
  - Containers (OpenShift)
  - Bare Metal (CloudLab)
- Project Organization Assessment (Burndown)
- Sprint 2 Focus

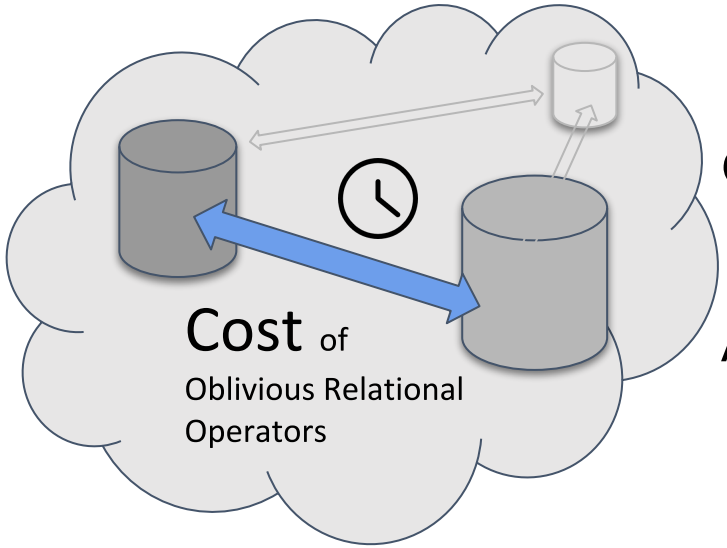
What is Multi-Party Computation (MPC)?

- MPC handles
  - Multiple (private) data owners
  - Mutually agreed-upon calculation
  - … and provides insights **without revealing private data**

- Main types of MPC
  - Garbled Circuits
  - Secret Sharing
    - Used in our project



Mentor/Client Motivation



Current work:

Relational query plans performed under MPC

- Previously: separate secure and insecure operations
- New approach: perform everything securely and optimize “end-to-end oblivious queries”

Optimization (of query plans)

- Reduce space (memory) usage
- Reduce inter-party communication

Approach

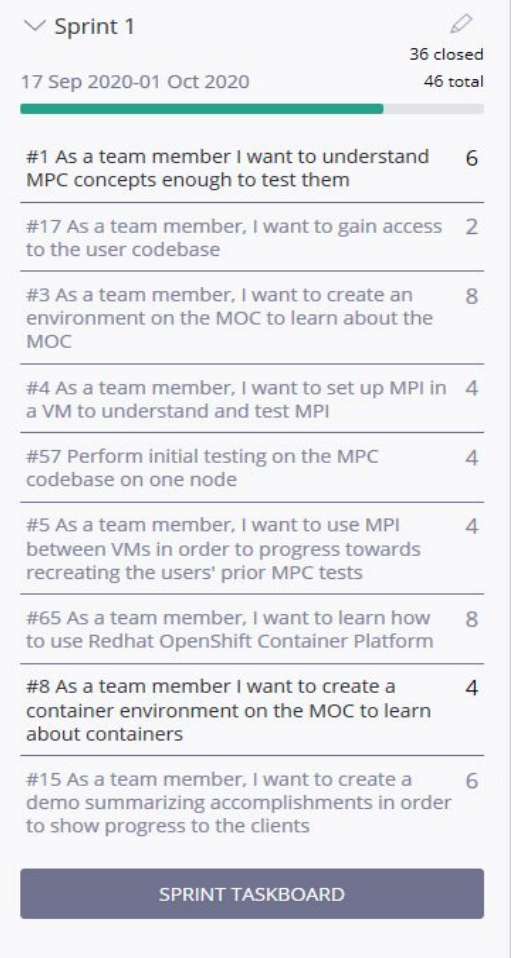
- C implementation with minimal dependencies
  - libsodium is used
- MPC “oblivious” primitives used to compose relational operators
- Create efficient plans by reordering relational operators

BOSTON UNIVERSITY

Project Goals

High level Objectives

- Configure and document different deployments for MPC
  - Profile resource use and latency: where is the performance going?
    - Comm. vs. Compute (roofline?)
    - OS vs. library vs. network stack
  - Examine hardware/software stack trade offs
    - Performance
    - Ease of configuration

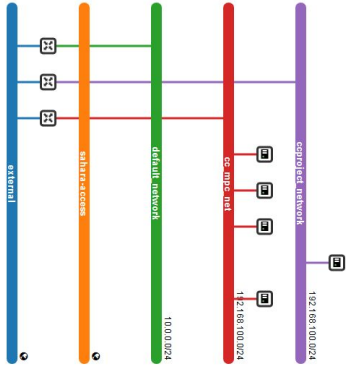


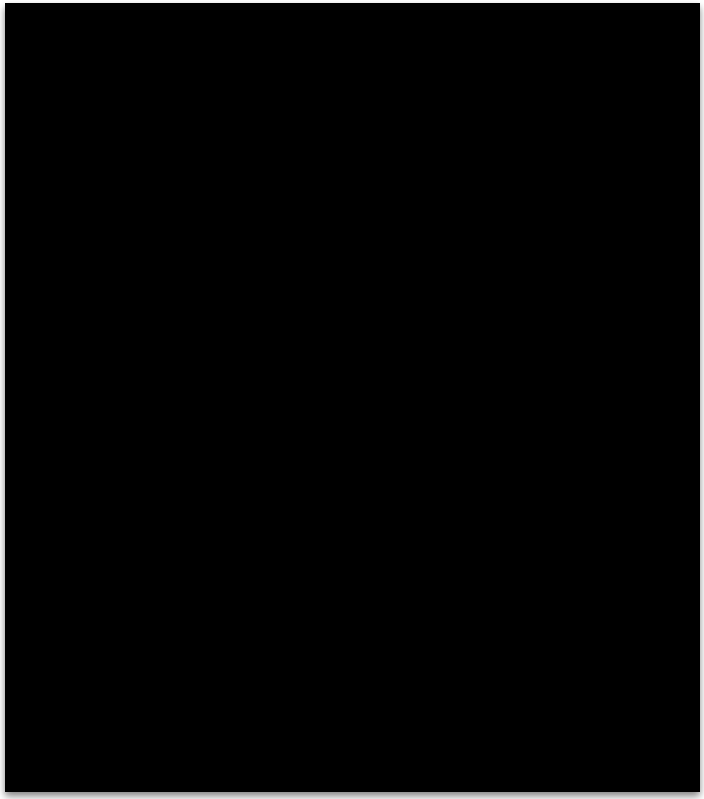
BOSTON UNIVERSITY

VMs (OpenStack)

- MOC OpenStack & outside guides
  - To stand up a VM
- Installed the Libsodium and MPI
  - Dependencies for MPC codebase
  - Tested MPC codebase on multiple cores and with “oversubscribe”
- Cloned VM with all needed configurations
  - 3 matching VMs on the same subnet
  - Tested MPC codebase on multi-VM configuration
- Small code changes and steps needed documented in repository

Network architecture of our system (red subnet)





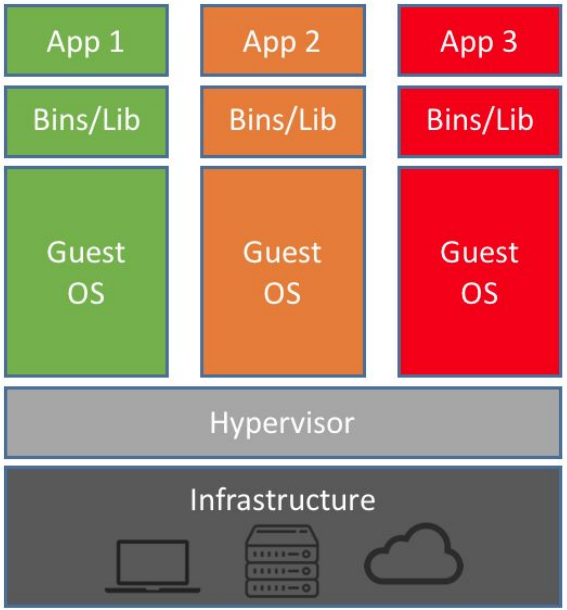
BOSTON UNIVERSITY

Containers

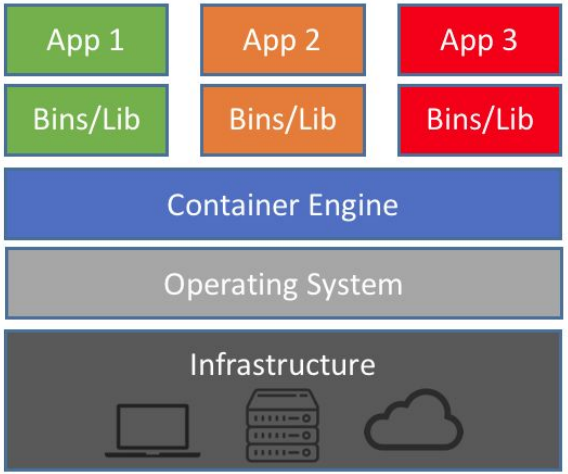
Motivation:

- VM's are resource hungry
    - Idle resources wasted if not used by the application
- Launching a new VM for a each new computing party is impractical as number of computing parties in the scenario is scaled.
- Need a way to 'orchestrate' the creation and management of these VMs.

Containers



Machine Virtualization



Containers

BOSTON UNIVERSITY

Enter: OpenShift Container platform

- PaaS that combines Docker containers orchestrated and managed by Kubernetes on a foundation of RHEL
- Choose target application platform, provide SCM (github ,gitlab) link and the build starts !
- Allows CI/CD: Any change made in source code automatically triggers integration (build) and deployment.
- Abstracts away docker and kubernetes through its GUI.

OpenShift Container platform

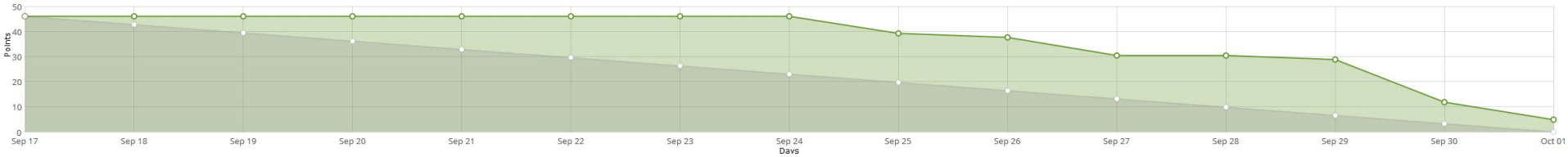
- These are the things I've done to get started with Openshift and deploying applications on it:
  - Installed minishift on local machine and set up a local version of openshift for testing off of the cloud.
  - Set up containers for sample applications on Python(Flask), Ruby,etc.
  - Tested inter-container communication using provided Openshift infrastructure.
  - Tested CI/CD on these applications.

- **Currently, looking into using 'Conan', a C/C++ application manager to take off code from VM deployments, package them as containers.**

Bare Metal (CloudLab)

- Not just Bare Metal, also has software options
  - Potentially use preconfigured community configurations
- Opportunity for more comparable tests
- Focus on Bare Metal environment initially
  - Apply MOC OpenStack and OpenShift experience to CloudLab later

Sprint 1 Burndown Chart



- Observations:
  - Late Sprint 1 start - backlog creation Sept 23/24
  - Scheduling across 3 time zones
    - No formal planning poker - did not help time estimates
    - Self-selection somewhat uneven
  - Still managed to accomplish most tasks/stories

BOSTON UNIVERSITY

Sprint 2 - Some Known Stories

- Backlog grooming & Planning Poker → 8/1 or 8/2
  - As a team member, I want to establish more clarity early in the sprint to increase productivity.
- VMs (OpenStack) → Initial MPC profiling
  - As a researcher, I want to profile the MPC codebase to verify replicability.
- Containers (OpenShift) → Use Conan to build C application and deploy on Openstack.
  - As a team member, I want to configure a custom image to support the MPC codebase.
- Bare Metal (CloudLab) → Gain access & run Hello World
  - As a team member, I want to create a test program on CloudLab to learn more about the tools available.
- MPC Familiarity → Meeting scheduled with Mayank
  - As a team member, I want to hear about the relevant algorithms in detail in order to better understand the bottlenecks in MPC.

Thank you

...any questions?

BOSTON UNIVERSITY