

S2MPC: Hands-on Programming Session with ORQ

EuroSys 2026

Eli Baum, 27 April
Boston University

Overview

- Deploy ORQ on a bare-metal CloudLab node
- Walk through an example query
- Write your own query
- Distributed execution over WAN

README

linked on S2MPC website

- Github
 - CASP-Systems-BU/orq
 - Branch s2mpc-2026
 - /bench/queries/s2mpc/README.md

Getting Started

- Create a CloudLab account:

<https://www.cloudlab.us/signup.php?pid=S2MPC-EuroSys26>

- If you already have an account:

- Your username (top-right corner) > Start/Join Project > Join

S2MPC-EuroSys26

- We will manually approve

Queries

- *Distinct Patients*: compute a list of all **distinct patients** who have been diagnosed with **heart disease** and prescribed **aspirin**
- *Secure Yannakakis*: compute the **aggregate cost** incurred by all insurance plans, grouped by **disease class**
- Threat model: multiple hospitals hold patients' medical information; insurance company holds billing and other demographic information

Overview

- ~~Deploy ORQ on a bare-metal CloudLab node~~
- ~~Walk through an example query~~
- Write your own query — `sec_yan` or `sec_yan_hard`
- Distributed execution over WAN
- On P0:

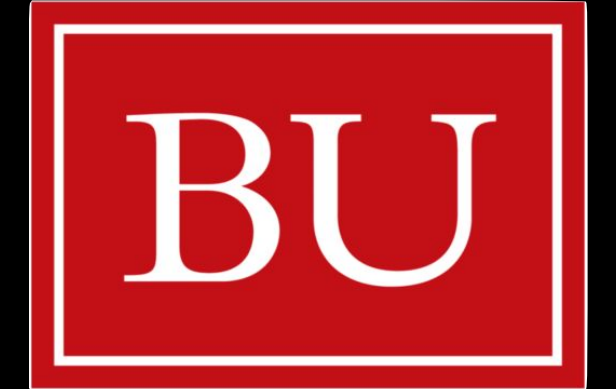
```
mpirun -n 3 -h localhost,P1.cloudlab.us,P2.cloudlab.us sec_yan
```

 - You must all have the same binary compiled — `scp` or use `solution.cpp`
 - If you modify CSVs, don't forget to update `size` variables in the C++

Notes

- Change CMakeLists to `-O2`
- Terminal WD should be in `/s2mpc/orq/build`
- To run the solution: `make sec_yan_solution; mpirun -n 3 sec_yan_solution`
- To run yours: `make sec_yan; mpirun -n 3 sec_yan`
- Fix permissions: `sudo chown -R YOUR_USERNAME /s2mpc/orq`

Thank you!



ORQ: Complex Analytics on Private Data with Strong Security Guarantees (*SOSP 2025, Distinguished Artifact*)



To appear soon in ACM Transactions on Computer Systems

Email me: elibaum@bu.edu



Links to
paper/code/docs