

FIT CTF

CTF FRAMEWORK FOR ETHICAL HACKING EDUCATION

AUTHOR: BC. HUNG DO

SUPERVISOR: ING. ZBYNĚK LIČKA

ABSTRACT

CTF exercises teach cybersecurity effectively, but VM-based infrastructure consumes excessive resources and requires constant maintenance. We developed a containerized framework that reduces memory consumption by up to 94% at idle and up to 83% under full load. The framework combines fit-ctf (orchestration library), fit-rendezvous (TUI platform), and fit-ctf-virt (deployable stack) — all open-source and self-hosted on bare-metal servers. Key features include per-student unique scenarios to prevent flag sharing and automatic flag validation. This enables institutions to scale CTF assignments effectively while reducing instructor burden.

Old Solution

- VM-based, continuously running login and target nodes
- operated by shell script
- resource intensive (hitting memory ceiling)
- difficult to develop on, scale and maintain
- manual assessment
- only shared network; flag updates mitigates cheating

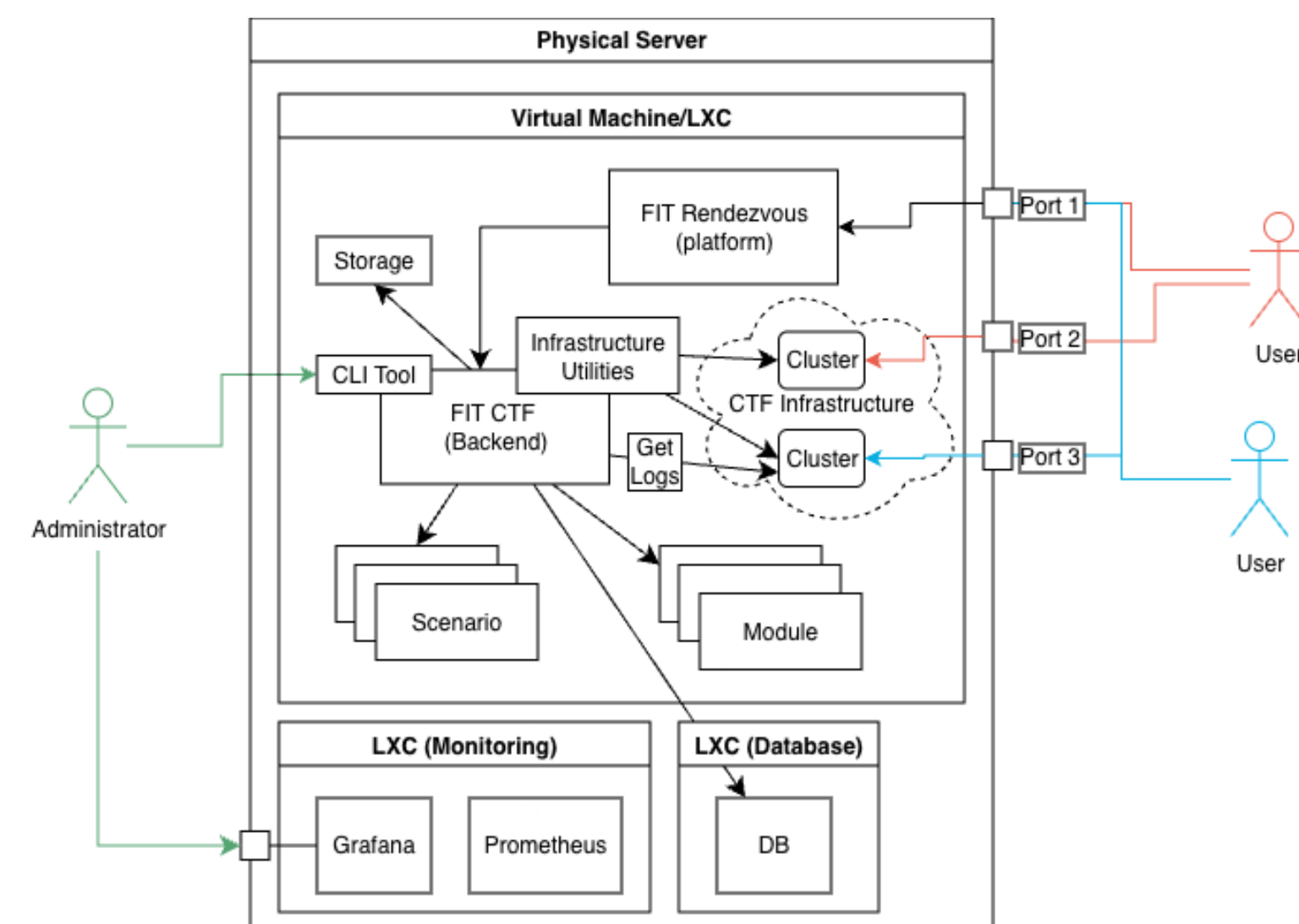
FIT CTF Framework

- **orchestration and resource management library** (fit-ctf)
 - container-based (podman-compose); run on a single server
 - modularity using scenarios
 - reusable environments with multiple containers tied together
 - templating – runtime injection of unique values
 - shared and private network → unique scenario combinations
 - manages users, projects, enrollment, progress instances, etc.
- **TUI platform** (fit-rendezvous)
 - leaderboard, flag submission
- **deployment stack** (fit-ctf-virt)
 - separate database and monitoring (Prometheus + Grafana)

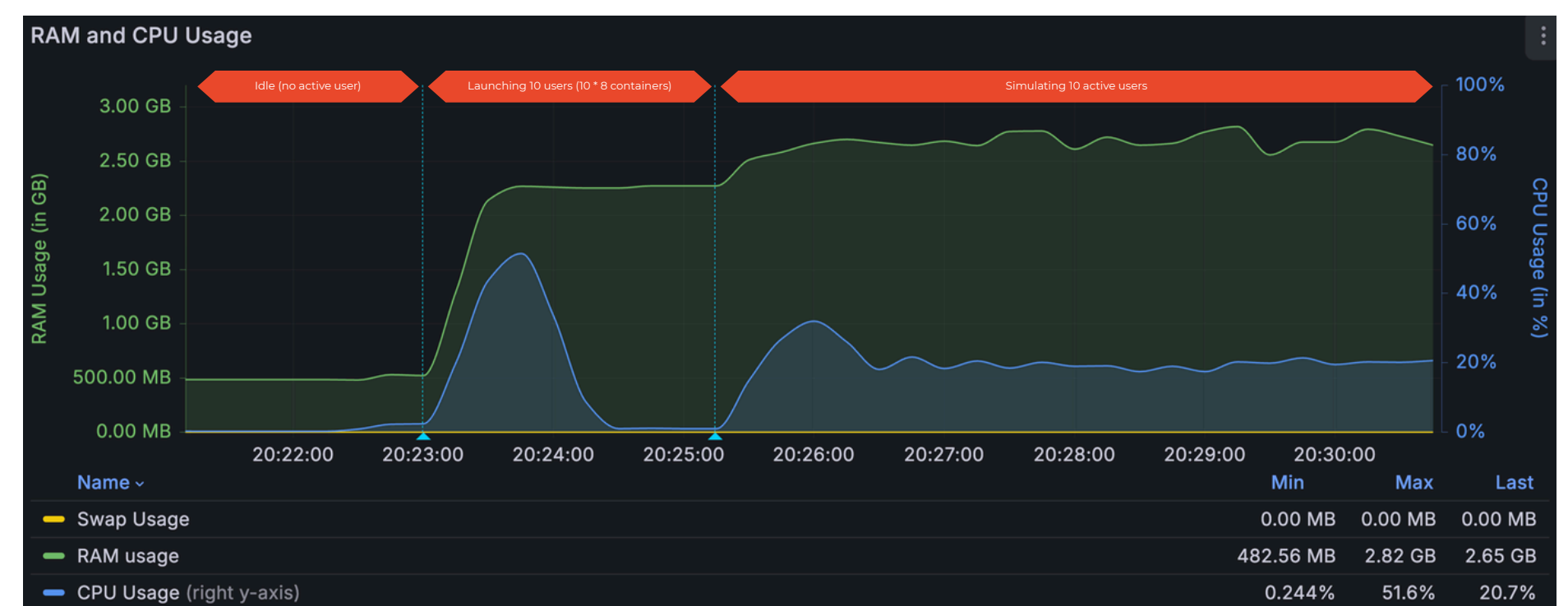
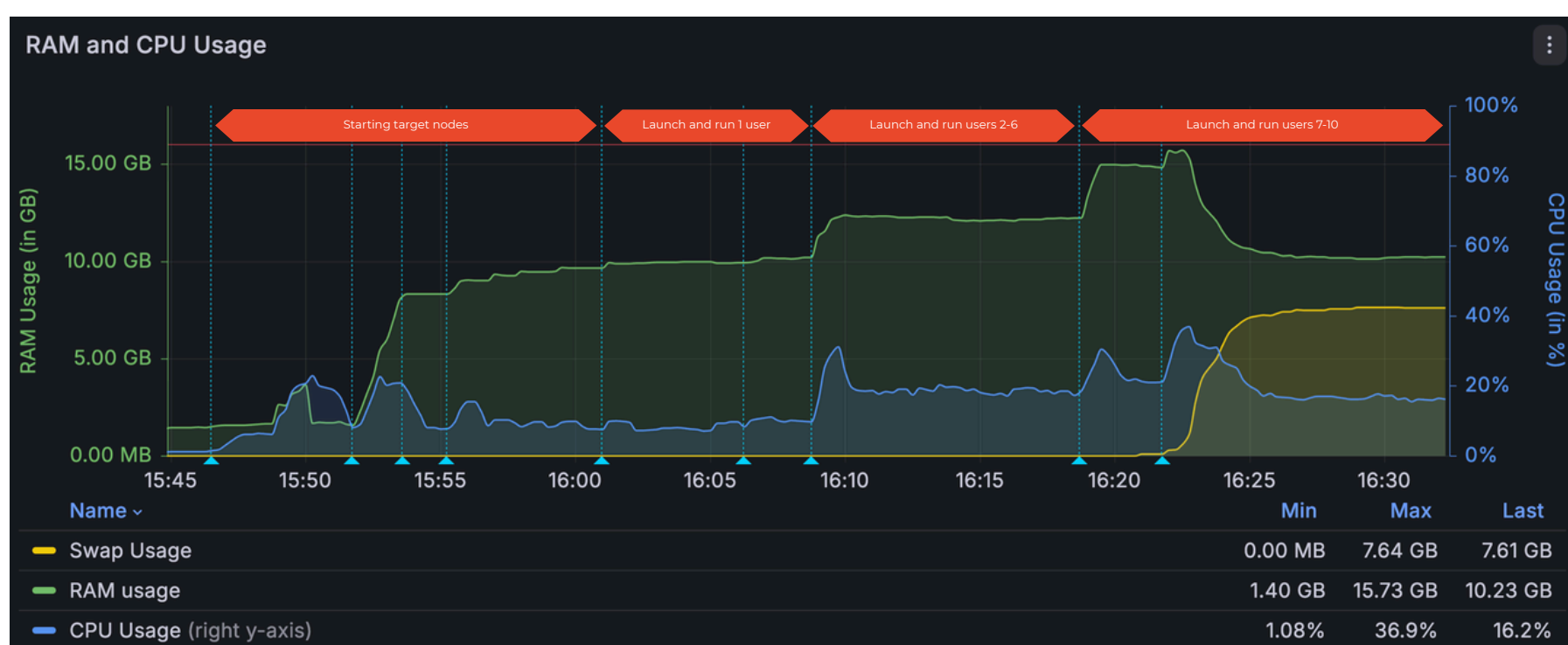
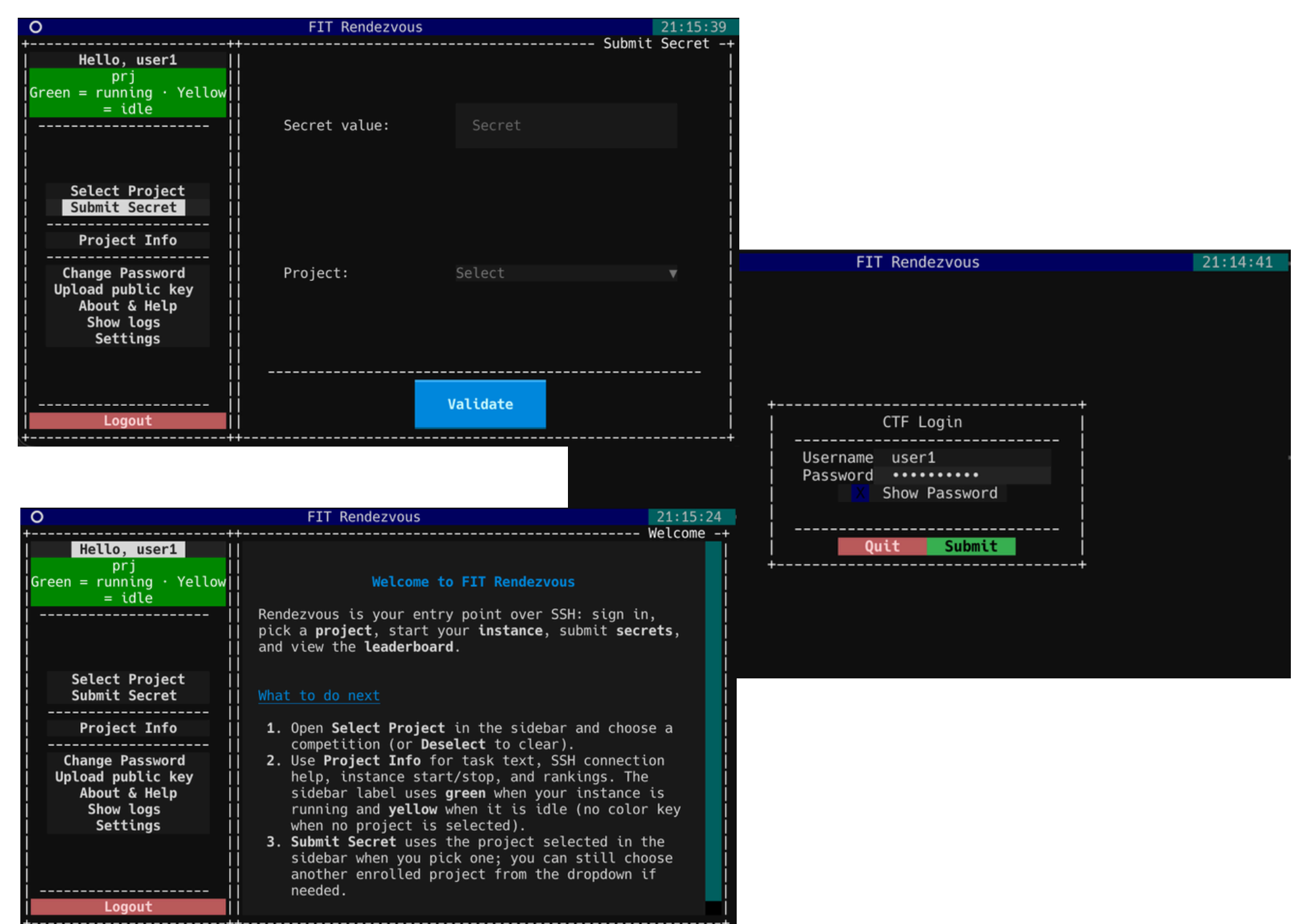
Performance test results

- Lenovo M920q; Intel i5 - 9500T; 16GB RAM + 16GB swap file
- using Ansible to simulate up to **10 active users**
- old solution (left graph)
 - 4 target nodes + 10 login nodes; 1GB RAM limit, 1 CPU core
 - CPU usage: ~7% on idle, ~16% when simulating
 - memory usage:
 - 9GB on idle without login nodes
 - 10/12.2/16+GB when launching and simulating 1/6/10 users
 - swap usage on 10th active login node
- testing new solution (right graph)
 - 10 login nodes; 4 scenarios for each user (in total 80 containers)
 - running inside VM; 10GB RAM; 2 CPU cores
 - CPU usage: ~1% idle, ~20% when simulating
 - memory usage: 500MB on idle, ~2.7GB when simulating
- conclusion: new approach **improves memory utilization by up to 80-90 %**

New Architecture



FIT Rendezvous TUI



Source Code

