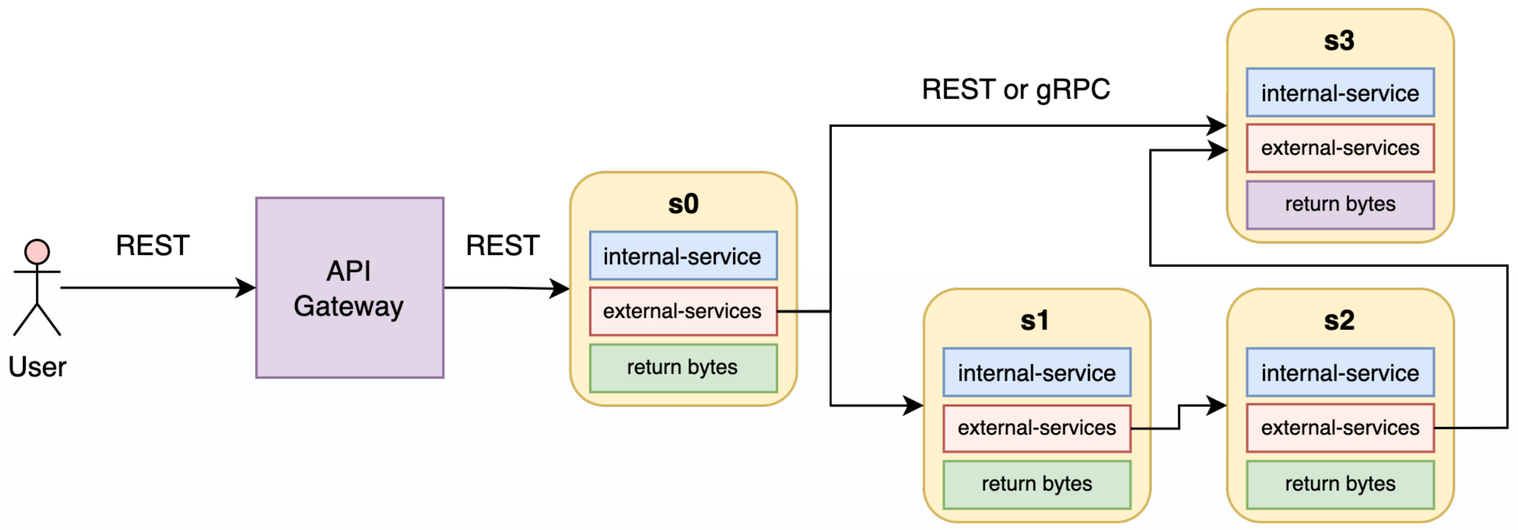
µBench - A factory of benchmarking microservices applications

Andrea Detti, Ludovico Funari, Luca Petrucci
CNIT and University of Rome “Tor Vergata”



µBench generated microservice application

What is it ?

µBench creates ***dummy*** **microservice applications** that can be customized by the user and run on Kubernetes.
µBench is a tool for benchmarking cloud/edge computing platforms that run microservice applications.

Who cares ?

Researchers and cloud platform developers who lack real microservice applications to benchmark their findings.
Can also be used for **educational purposes** to show advantages and problems of microservice applications.

Which level of customization ?

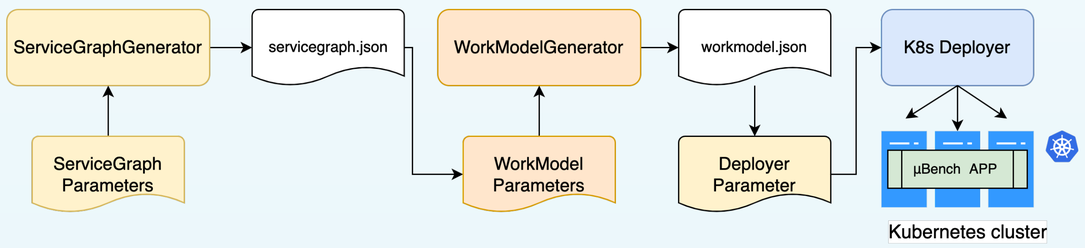
- **dependency graph** among microservices (service graph)
- **stress functions** (e.g. for CPU, memory, I/O, network) run by microservices
- Kubernetes **CPU and Memory request, scheduler,** etc.

What I need ?

Written in **Python**. You need a **Kubernetes** cluster with **kubectl** access from the machine running µBench. With **Microsoft Visual Studio Code**, you have some Run and Debug actions already there.

How to create a µBench App ?

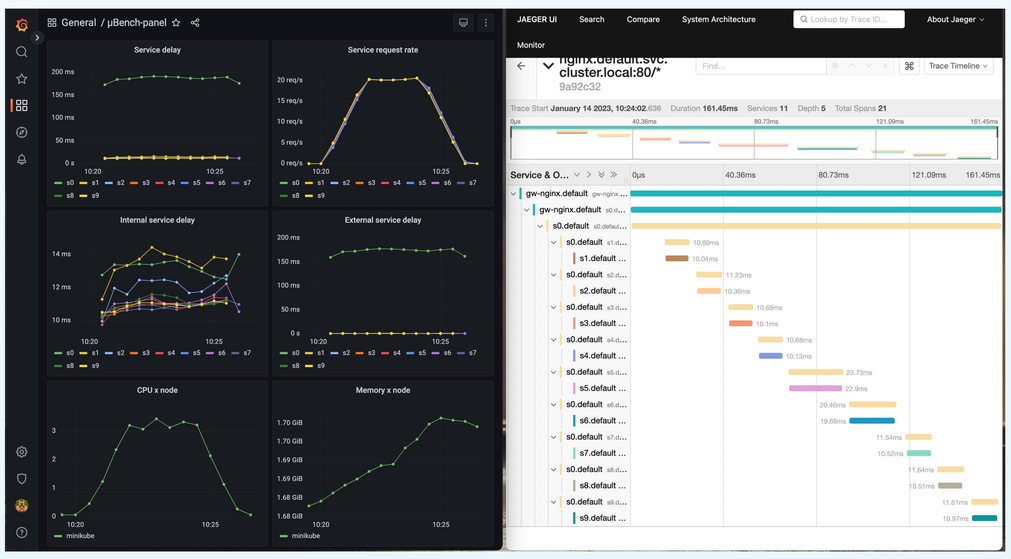
Manually, by writing a **workmodel.json** file that describes the workload of each microservice and then running the **K8s Deployer** tool. For large applications better to use the **µBench toolchain** that creates **random apps** with configurable statistical properties.



µBench toolchain

What can I observe ?

µBench is thoroughly integrated with **Prometheus and Istio service mesh**. You can use **Grafana** for real time metric plots, **Jaeger** for traces and **Kiali** to observe the service graph.



Some research areas

Benchmark for :
- Resource **schedulers** for cloud applications
- Placement policies for **cloud/edge cloud continuum** application
- **Load balancing** policies for service mesh
- **ML models** for digital twin of microservice apps

µBench is partially funded by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, Partnership on “Telecommunications of the Future,” Program “RESTART” under Grant PE00000001, “Netwin” Project (CUP E83C22004640001)

https://github.com/mSvcBench/muBench