LTB 2021

Load Testing & Benchmarking

Enabling Containerized, Parametric and Distributed Database Deployment and Benchmarking as a Service

George Kousiouris, Harokopio University of Athens

gkousiou@hua.gr

Dimosthenis Kyriazis, University of Piraeus

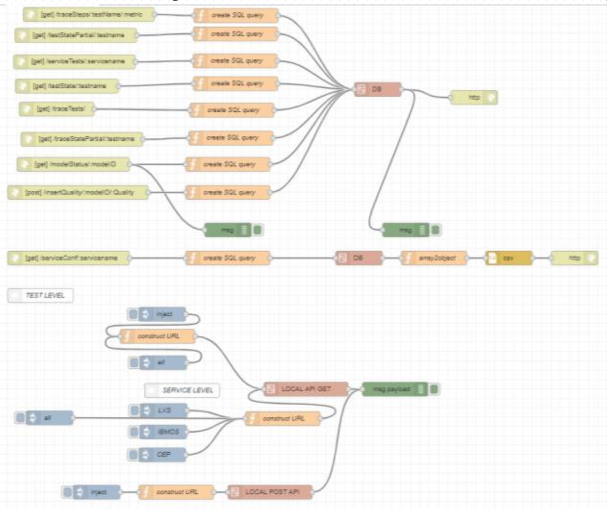
dimos@unipi.gr

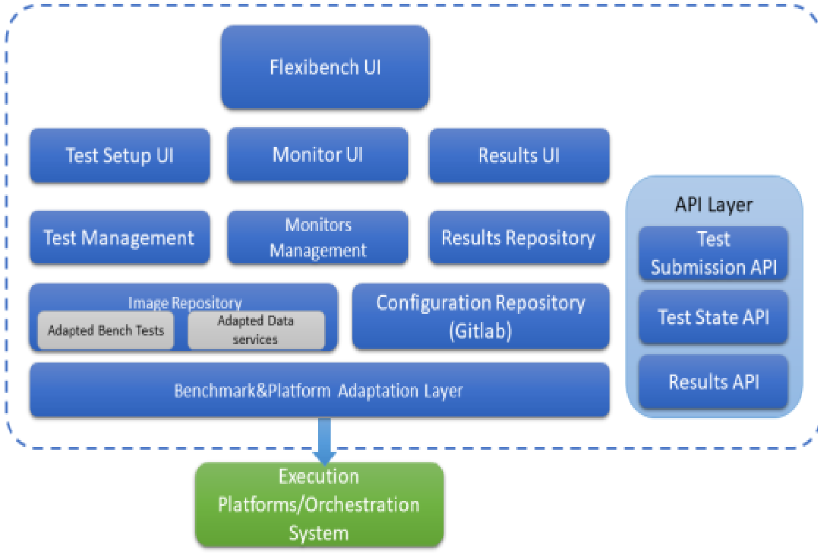
The Ninth International Workshop on
Load Testing and Benchmarking of Software
Systems (LTB 2021)

The research leading to the results has received funding from the European Union's funded Projects BigDataStack (GA No 779747) and PHYSICS (GA No 101017047).

Flexibench Background

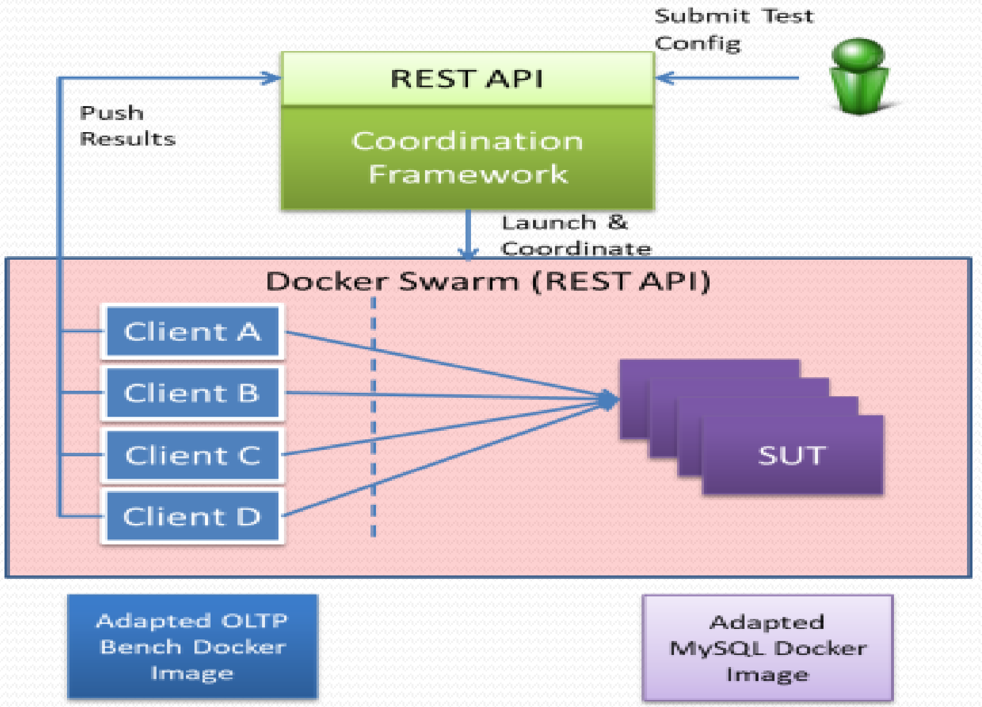
- Stress testing as a service framework
- Creation of dynamic, containerized clusters on demand
  - Based on orchestration of a back-end orchestration platform (Docker Swarm)
- Various modes of operation
  - Sequential, parallel, trace-driven experiments
- Based on adapters
  - for baseline load injectors (Jmeter), platforms, sync logic
- Based on Node-RED
  - Node.js driven event-driven app framework


[get] /traceSteps/:testName/:metric
[get] /testStatePartial/:testname
[get] /serviceTests/:servicename
[get] /testState/:testname
[get] /traceTests/
[get] /traceStatePartial/:testname
[get] /modelStatus/:modelID
[post] /insertQuality/:modelID/:Quality
create SQL query
DB
http
msg
[get] /serviceConf/:servicename
array2object
csv
TEST LEVEL
inject
construct URL
all
SERVICE LEVEL
LXS
IBMOS
OEP
LOCAL API GET
msg.payload
LOCAL POST API

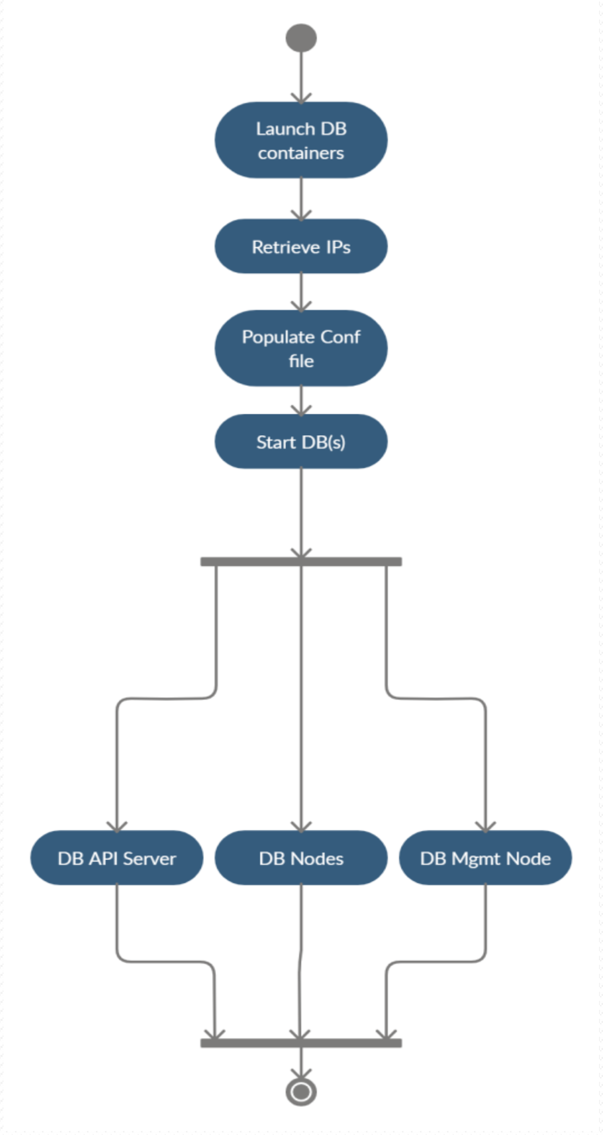


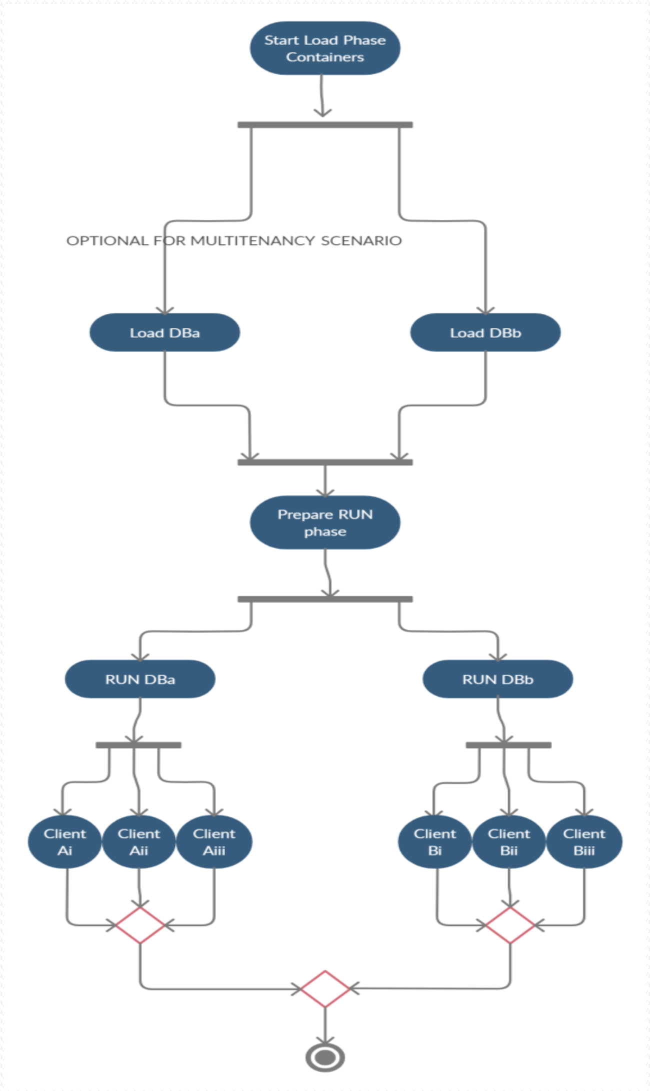
Flexibench DBaaS Tests

- Goal: Performance test configurations of a selected DB instance
  - Launching dynamically parameterized DB instances (MySQL) and then coordinates distributed load injection with OLTPBench/YCSB
  - Can include one instance
    - in parameter sweep kind of testing
    - Detect the effect of various setup parameters (#datanodes, request rate, type of workload etc.) in performance
  - Can include more than one instances
    - For DBaaS co-allocation performance interference investigation



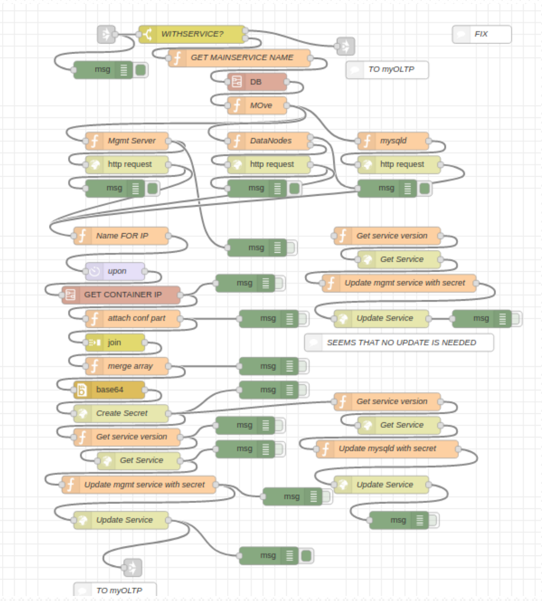
Synchronization needs

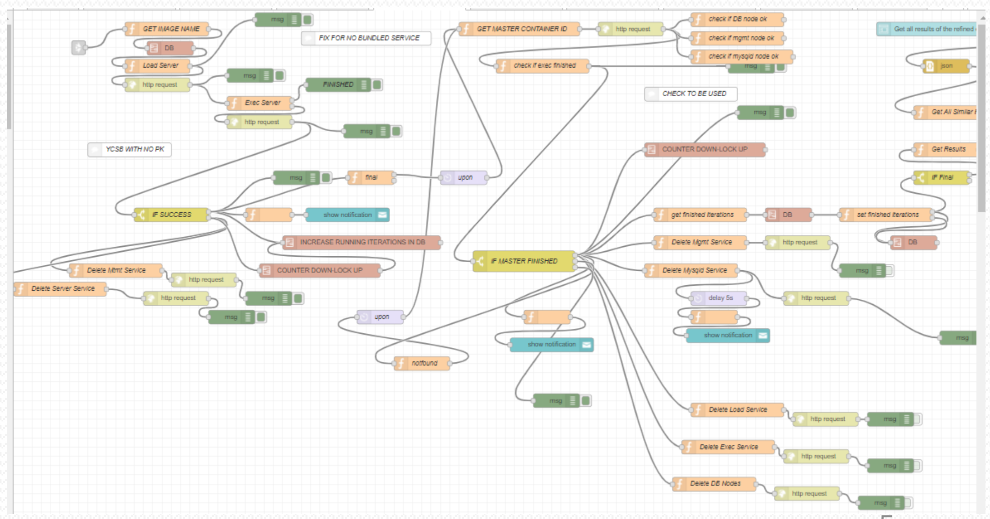




Issues to consider and adapter logic

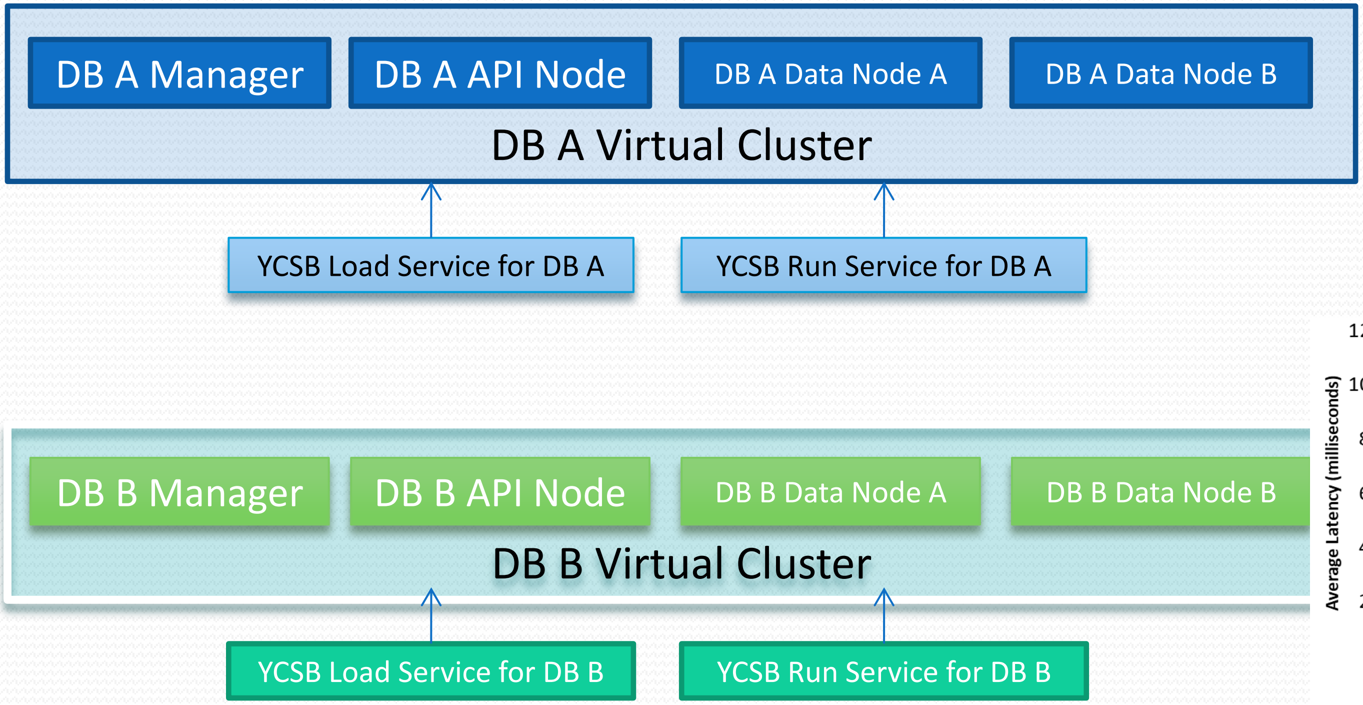
- Change of baseline docker images (MySQL, OLTPBench) to be dynamically configurable and logic to implement experiment synchronization and lifecycle management
    - How information and sync is passed across 4 layers (REST API, server logic, Container configuration and DB scripting)
- YCSB does not have distributed mode
    - Needs to be orchestrated
    - Other bugs: primary key generation conflicts, need to be disabled
- Volumes used for sync files creation not sharable between nodes
    - NFS based mount points
- Trade-off in DB creation for each experiment
- Extensions to other databases need the implementation of the according adapter
    - But adapter logic usually overlaps a lot (exploit existing adapters for reusability)

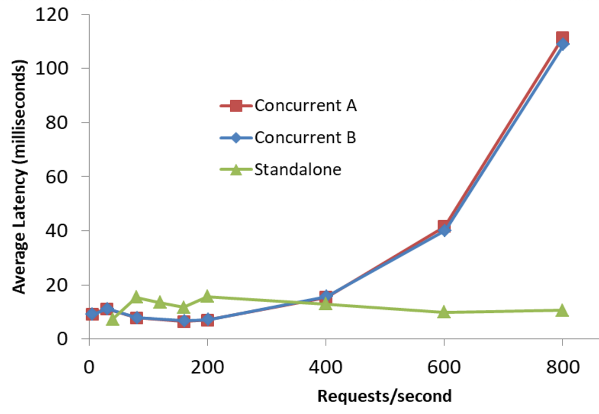




Co-allocation Scenario for performance degradation

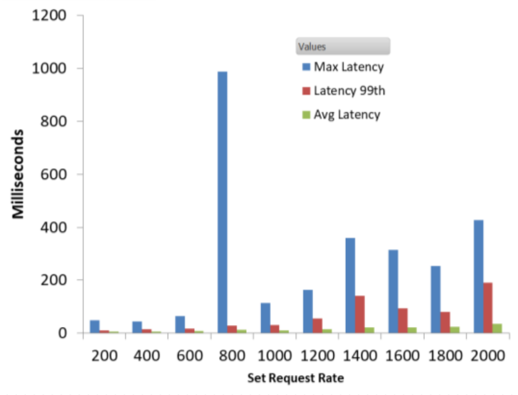
- Two parallel parametric instances of a DB with the databases on the same node
- Suitable for DBaaS co-allocation limits



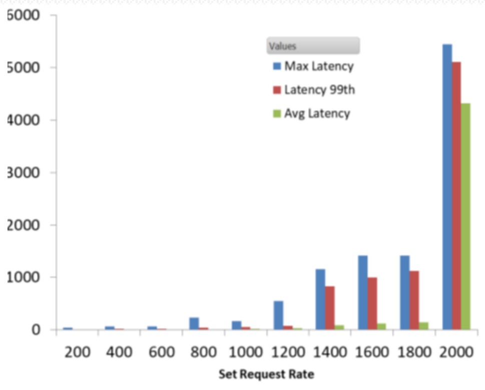


Example Results Analysis

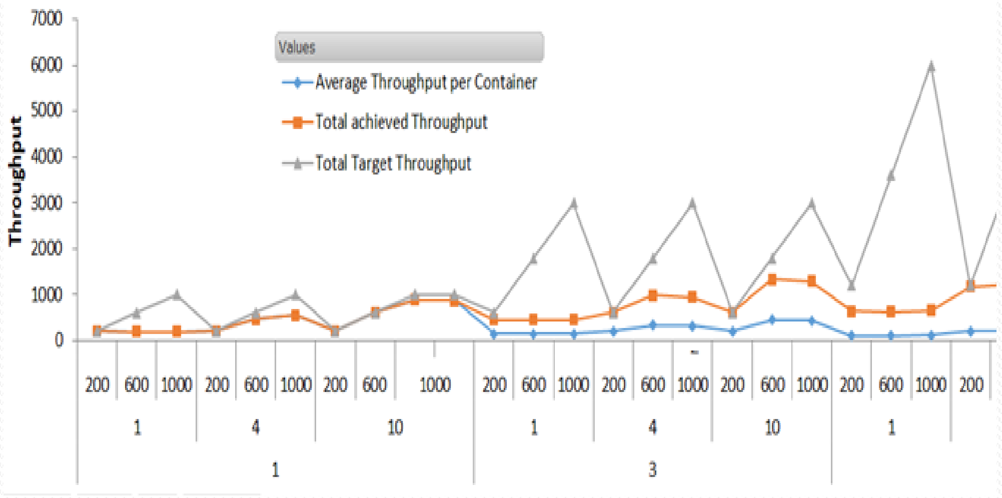
- Investigation of parameters effect on results
  - E.g. #data nodes and request rate in response times
- Investigation of client side bottlenecks
- Results are only indicative and runs were performed for functional testing
  - No optimization was applied on the DB setup, data nodes collocated on the same node



4 datanodes



8 datanodes



Conclusions

- Easy set up and deployment of DB characteristics
  - E.g. size, number of nodes, type of data
- Automated, API based test submission and monitoring
- Ability to include further automation in experiment creation through REST API functionality
- Easier gathering of datasets for further analysis
  - E.g. performance model creation of the SUT
    - Latency=f(#datanodes, param2, param3, request rate)

Future Work

- Improve configuration aspects and packaging of the tool

- Model creation for DB performance

- Extension of Flexibench to FaaS environments

Tool details

- Source code and docs:
  - http://bigdatastack-tasks.ds.unipi.gr/gkousiou/adw
- Docker images
  - https://hub.docker.com/u/gkousiou
- Demo Videos
  - https://www.youtube.com/playlist?list=PLs0yDOuwD6jPdht8A_V9sTO5Et15Z-OOF

Thank you!
Questions?