EXPAND

EXtreme PerformANce testing of custom IoT workloaDs

Stratos Keranidis, PhD

domX

07/02/2022

F4Fp_SME_NGI Experiment Review

EXPAND

- **Experiment Description**
  - Concept and Objectives
  - Background and Motivation
  - Experiment setup
- **Project Results**
  - Groups of Experiments and Results
  - Key findings
- **Business Impact**
  - Impact on business
  - Value perceived
- **Feedback**
  - Used resources and tools
  - Added value of Fed4Fire
  - Testimonial

Experiment Description

domX

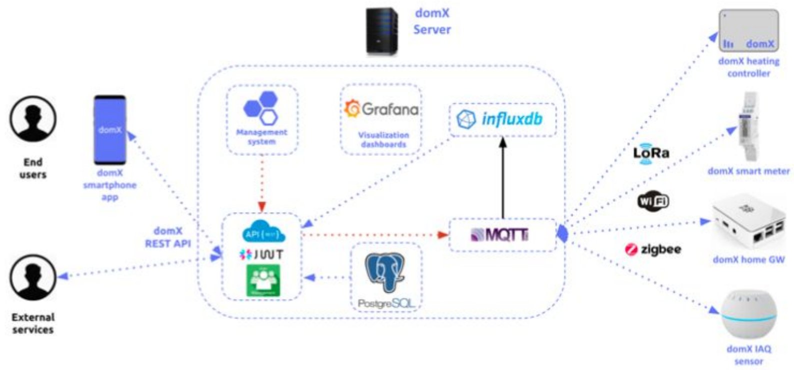
EXPAND

EXtreme PerformANce testing of
custom IoT workloaDs

Concept and objectives

In domX startup, we develop cost-effective and universal IoT products for the home environment, providing for:

- energy metering (electricity and gas)
- appliance control (HVAC controllers)
  improved living (automation gateway)
- climate monitoring (air quality sensors)



domX IoT platform architecture

Continuous need for validating the performance, scalability and reliability of the entire platform.

However, achieving end-to-end testing of a complex IoT platform with production level characteristics is extremely hard, with the core difficulty arising from the fact that most test environments bear little resemblance to the real environment.

Concept and objectives

However, **EXPAND** experiment was ideal for testing and evaluating an emulated version of our production setup.

In EXPAND experiment, we managed to:
- emulate the domX production environment
- emulate the domX IoT device setup
- assess its performance under **realistic workloads and different configuration setups.**

We emulated the traffic load of domX IoT devices for MQTT and HTTP traffic, by using the open source:
- **Locust** load testing tool
- **Paho** MQTT library

We deployed the tailored performance testing implementation in a distributed setup of multiple machines to generate realistic IoT workloads.

Background and Motivation

The experiments have been designed to characterize the impact of three different platform design choices, namely the:

- number of Virtual Machines (VMs)
- number of allocated CPUs
- number of employed platform replicas

Performance evaluation in terms of:

- number of simultaneously serviced clients
- average response time

Motivation:

- define the characteristics of a cost-effective platform setup able to cover our needs

Experiment design:

- **3 distinct groups**

Group 1:

- Multiple replicas in one VM
- Fixed number of CPUs per replica

Group 2:

- One VM - single replica
- Varying number of CPUs per replica

Group 3:

- Multiple replicas across VMs
- Fixed number of CPUs per replica

Experiment setup

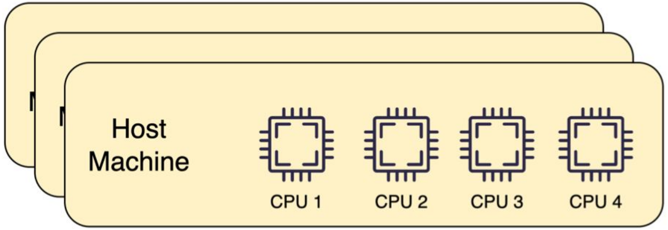
VirtualWall 2 testbed:

- 1-3 Nodes for VerneMQ
- 4-8 Nodes for Locust

Locust settings:

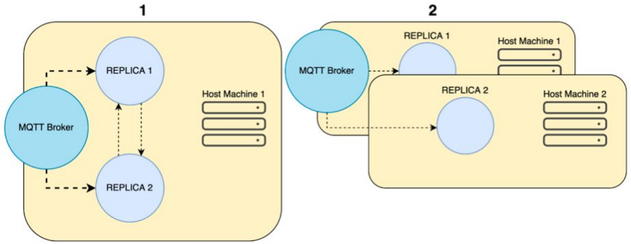
- **Locust Distributed Mode** utilizing 4 CPU cores on each VM and managed to generate 1300 concurrent clients
- Publishing to 21 different topics with varying publish intervals of 2, 10 and 60 seconds with Quality of Service (QoS 0) and varying message payloads.

Locust VMs



VerneMQ settings:

- **Docker Swarm & VerneMQ Clustering** with one master node and multiple node replicas
- client authentication through user credentials and SSL certificates



VerneMQ replicas: on a single VM and on multiple VMs

Project Results

domX

EXPAND

EXtreme PerformANce testing of custom IoT workloaDs

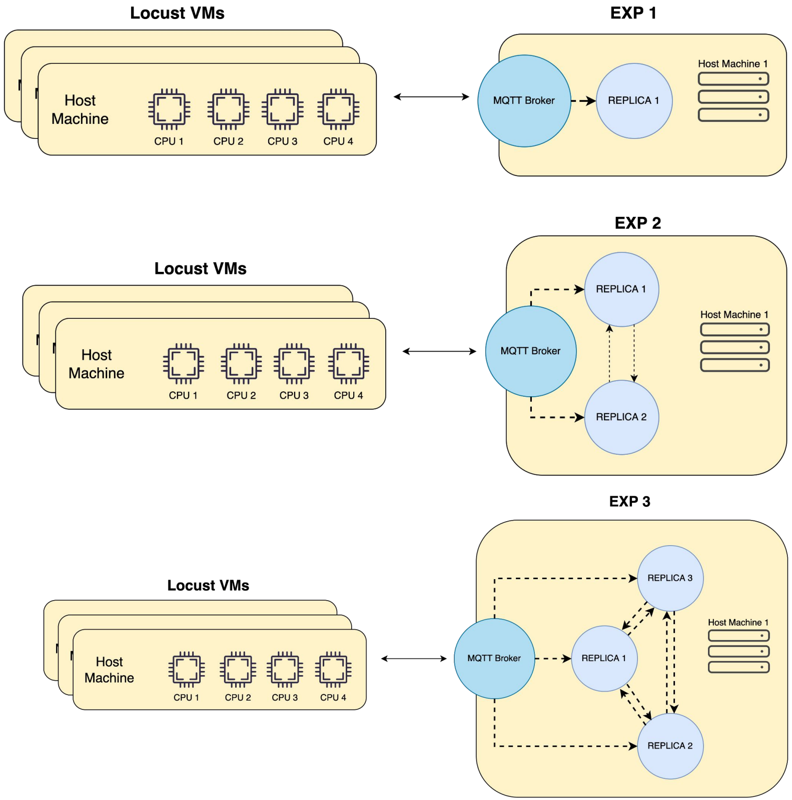
Group 1 Experiment set-up

VerneMQ Docker settings:

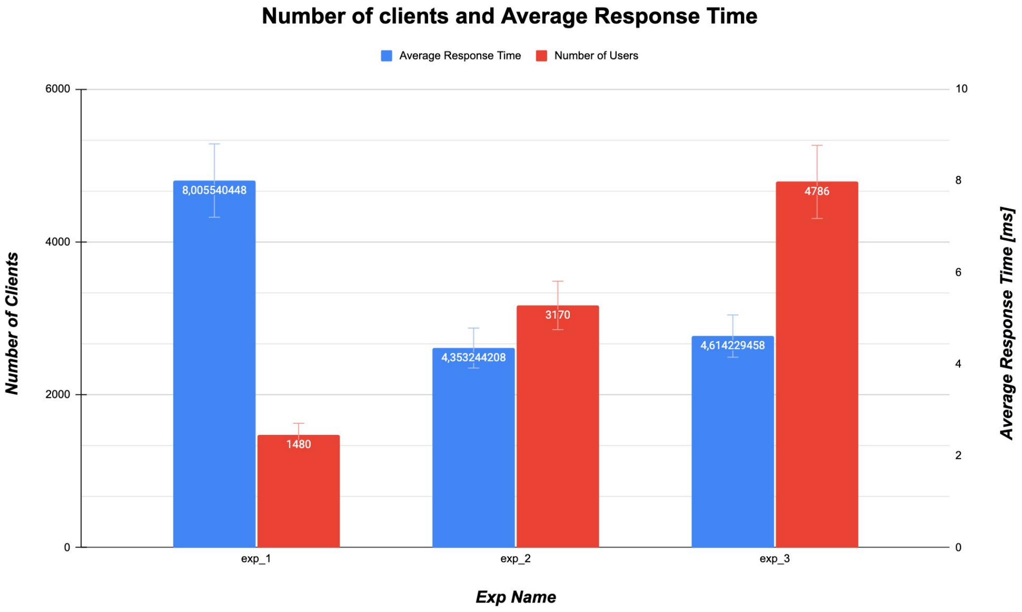
- 0.2 CPUs resource limit on each VerneMQ replica
- 4 GB RAM resource limit on each VerneMQ replica
- ***Increasing number of Replicas***

Locust end-clients settings:

- 200 bytes payload on 2 sec interval
- 500 bytes payload on 10 sec interval
- 700 bytes payload on 60 sec interval



Group 1 Results



Experiment Name	Number of replicas	CPUs per replica	Total CPU utilization
Exp 1	1	0,2	0,2
Exp 2	2	0,2	0,4
Exp 3	3	0,2	0,6

Client capacity:

- Linear increase with # of replicas
- **2.14 X** client increase with **2** replicas
- **3.23 X** client increase with **3** replicas

Response time:

- Significant decrease
- 45% decrease with **2** replicas
- similar decrease with **3** replicas

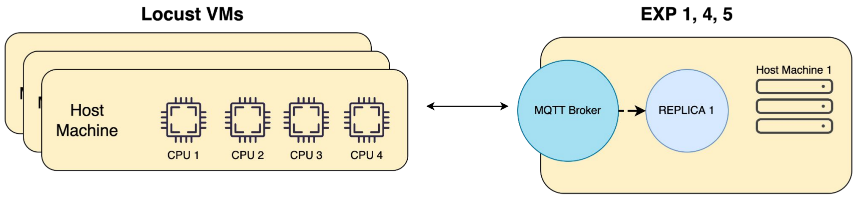
Group 2 Experiment set-up

VirtualWall 2 testbed:

- 1 Node for VerneMQ
- 8 Nodes for Locust

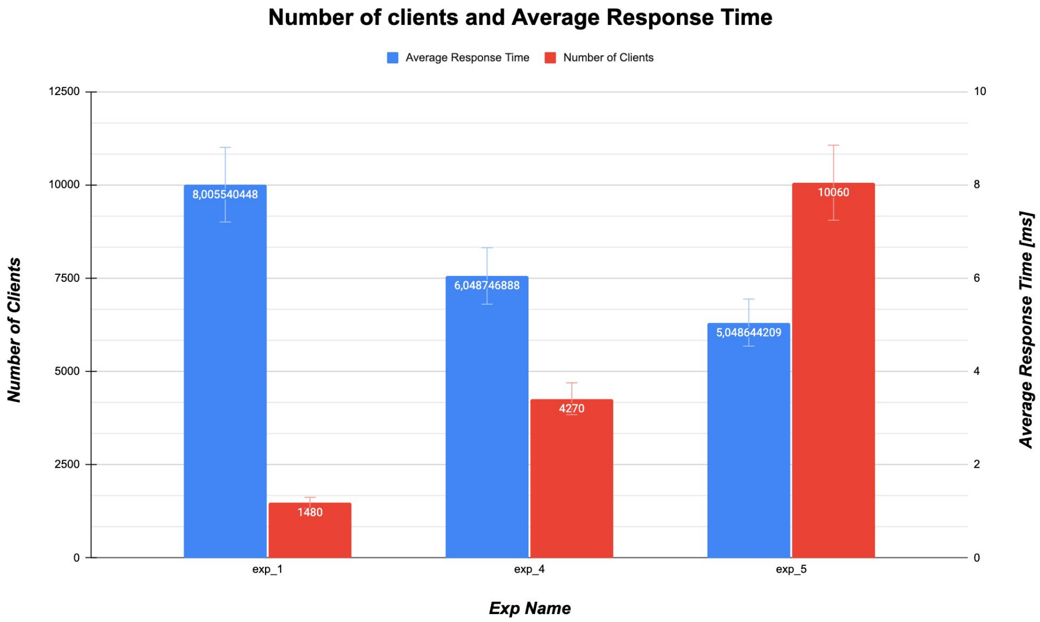
VerneMQ Docker settings:

- 4 GB RAM resource limit on each VerneMQ replica
- ***Increasing number of CPUs per Replica***



Experiment Name	Number of replicas	CPUs per replica	Total CPU utilization
Exp 1	1	0,2	0,2
Exp 4	1	0,4	0,4
Exp 5	1	0,6	0,6

Group 2 Results



Experiment Name	Number of replicas	CPUs per replica	Total CPU utilization
Exp 1	1	0,2	0,2
Exp 4	1	0,4	0,4
Exp 5	1	0,6	0,6

Client capacity:
- Non-linear increase with # of CPUs
- **2.9 x** client increase with 2 CPUs
- **6.8 x** client increase with 3 CPUs

Response time:
- Significant decrease
- 24% decrease with **2** CPUs
- 36% decrease with **3** CPUs

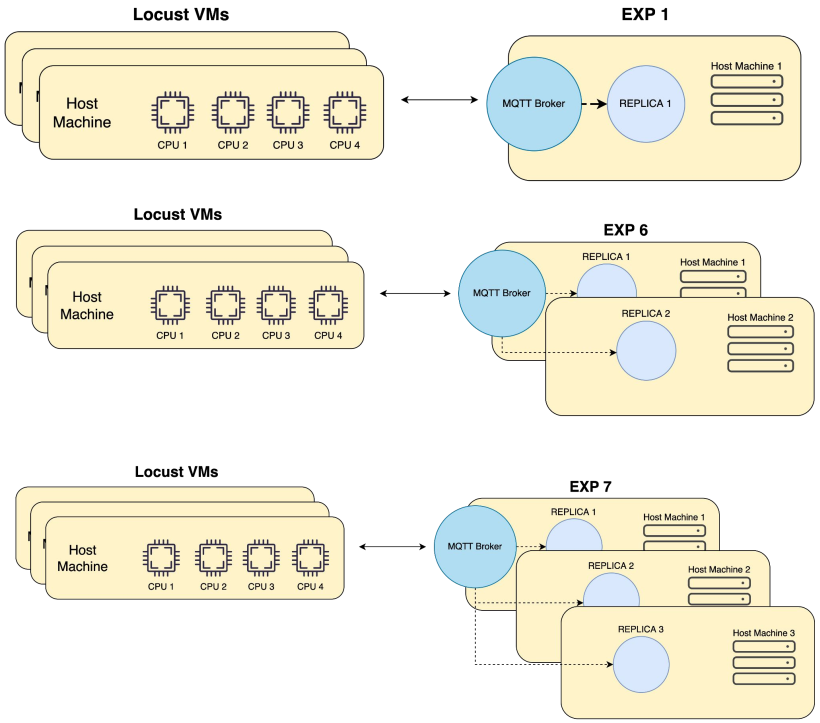
Group 3 Experiment set-up

VirtualWall 2 testbed:

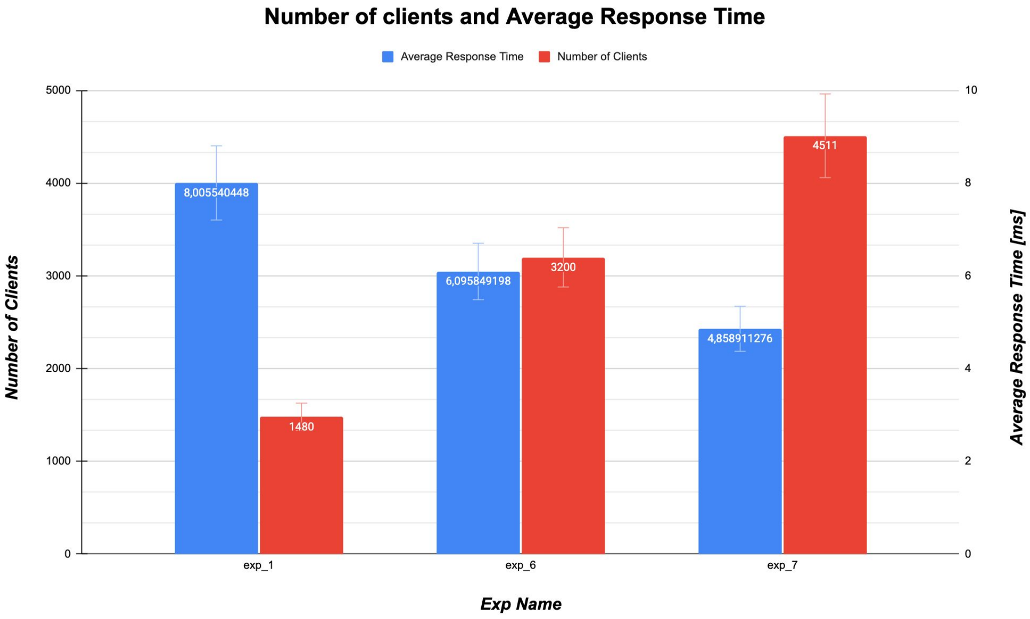
- 3 Nodes for VerneMQ
- 4 Nodes for Locust

VerneMQ Docker settings:

- 0.2 CPUs resource limit on each VerneMQ replica
- 4 GB RAM resource limit on each VerneMQ replica
- ***Increasing number of Replicas across multiple VMs***



Group 3 Results



Experiment Name	Number of replicas	CPUs per replica	Total CPU utilization
Exp 1	1	0,2	0,2
Exp 6	2	0,2	0,4
Exp 7	3	0,2	0,6

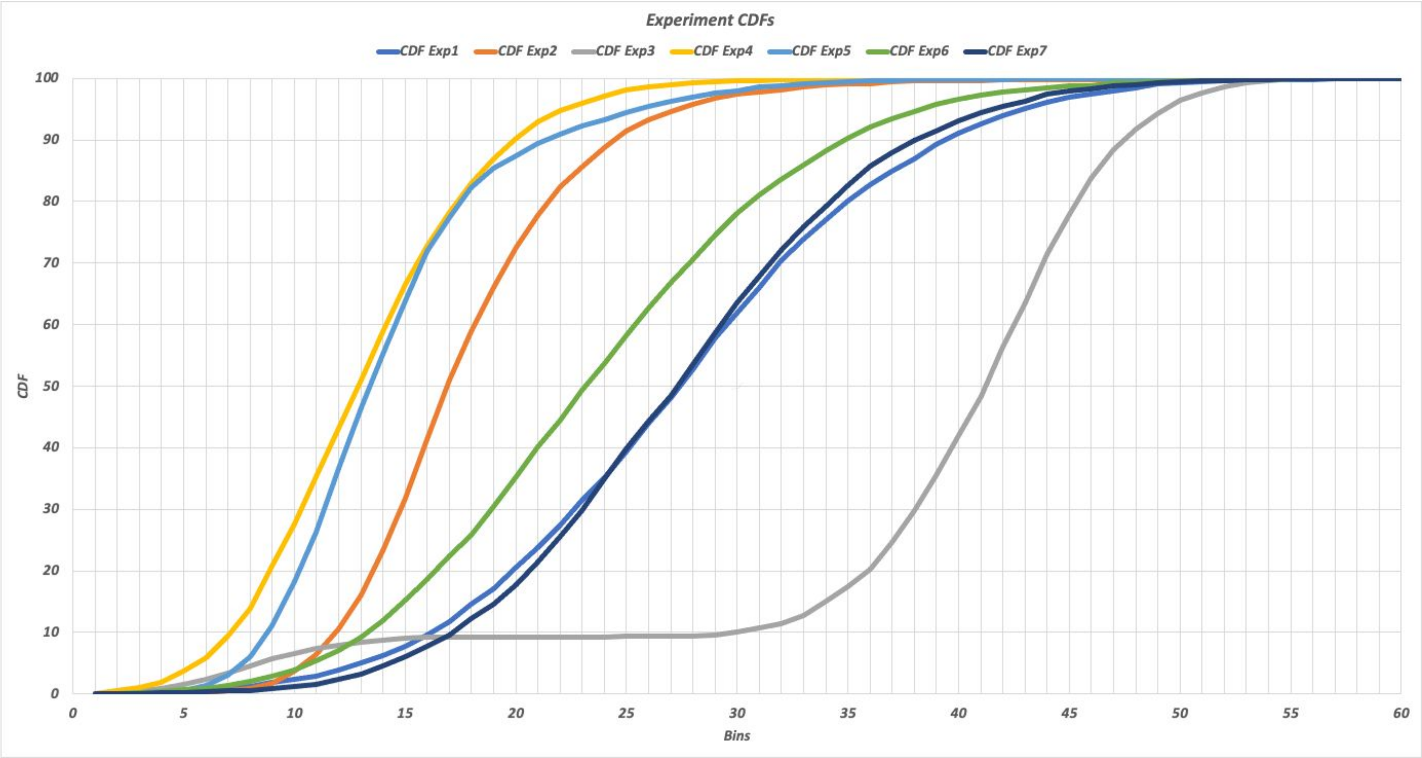
Client capacity:

- Linear increase with # of replicas
- **2.16 x** client increase with **2** replicas
- **3.04 x** client increase with **3** replicas

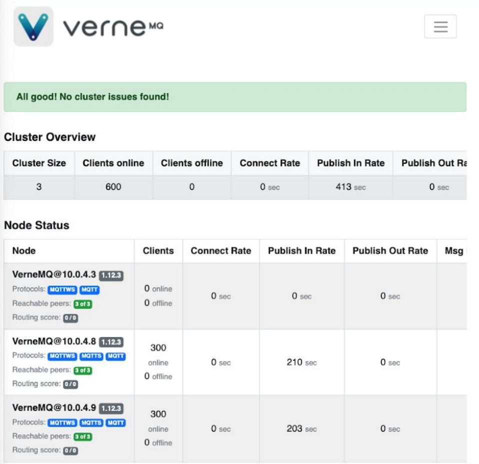
Response time:

- Significant decrease
- 24% decrease with **2** replicas on **2** VMs
- 40% decrease with **3** replicas on **3** VMs

Cumulative Distribution of Avg. Resp. Time



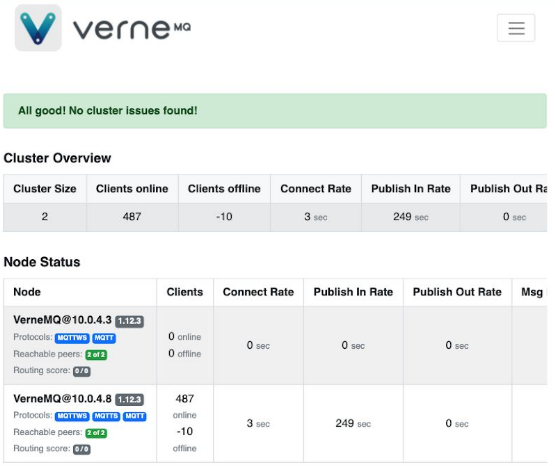
Two Replicas across VMs (1/3)



Benefits:

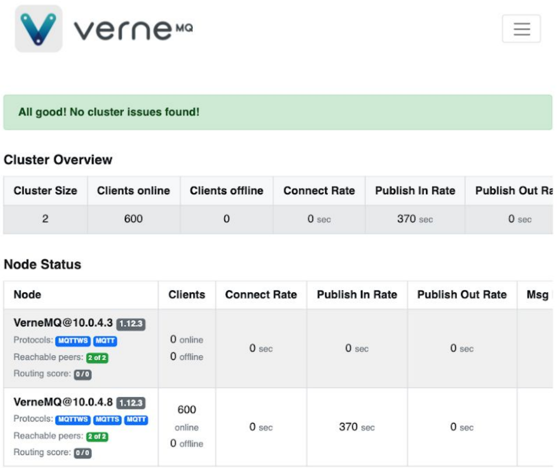
- Fault tolerant setup
- Possible migration of lost connections

Two Replicas across VMs (2/3)



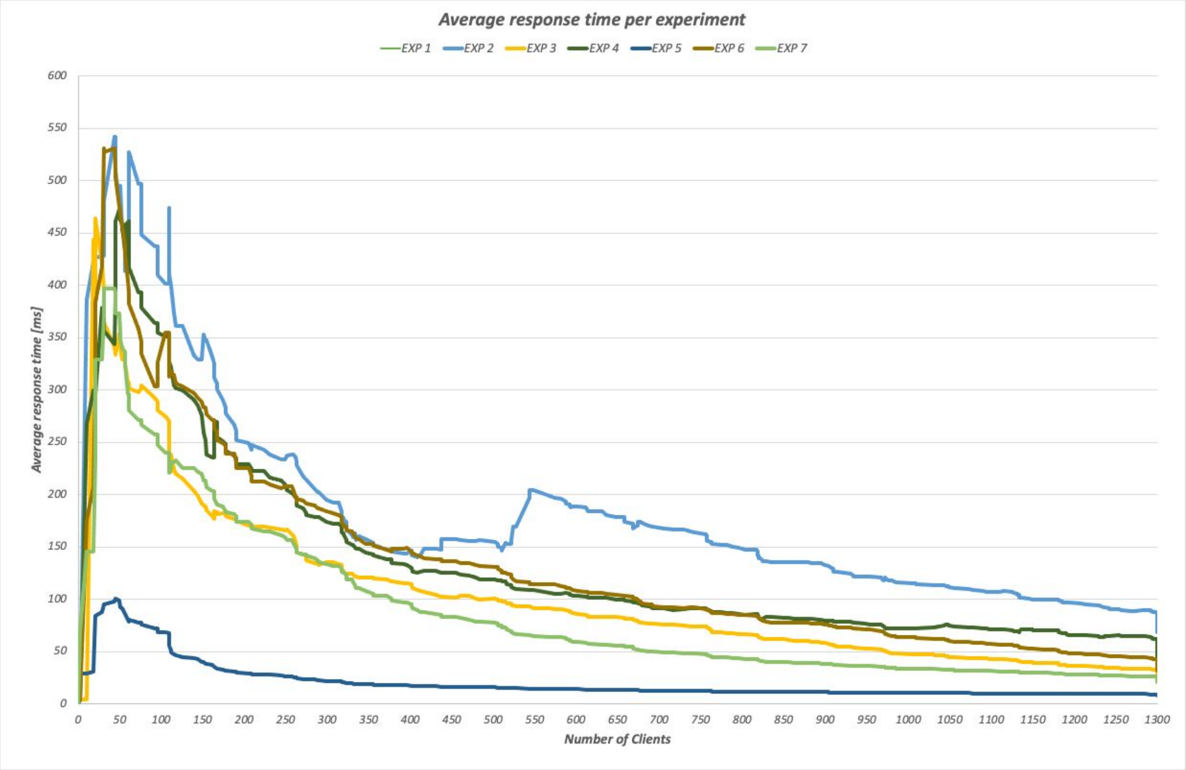
Forced disconnection of first Replica from the MQTT cluster, resulting in a gradual migration of lost connections.

Two Replicas across VMs (3/3)



Successful complete migration of clients to the second Replica

Average Response Time



Average response time when forcing all clients to reconnect to a new Replica.

Key findings

FED4FIRE

Sweet spot:

- Increased number of replicas on a single VM induces coordination overhead
- The use of 2 replicas is enough to mitigate unexpected server downtimes
- No significant RAM utilization is required
- CPU allocation can be dynamically scaled

Planned platform upgrade:

- Reserve 2 different VMs
- Deploy 2 replicas on
- with Docker Swarm & VerneMQ Clustering
- No need to increase RAM beyond 4 GBs
- Current setup can support > 10K devices
- Increase # of CPUs (2->3) only if needed

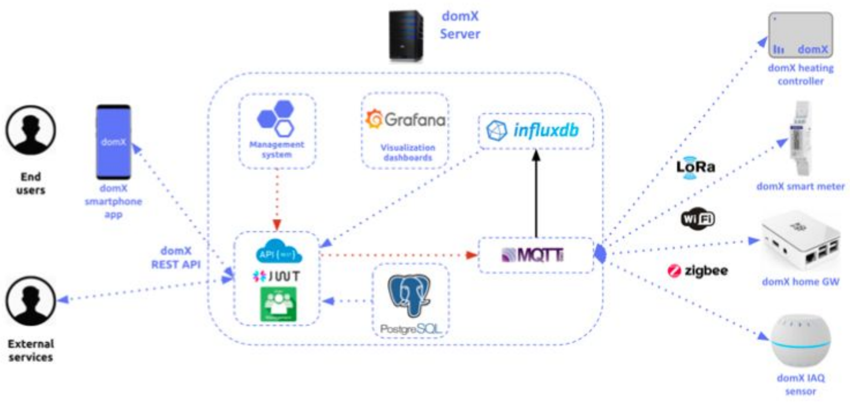
Business Impact

EXPAND

EXtreme PerformANce testing of custom IoT workloaDs

Business Impact

- EXPAND provided domX with pragmatic results for investigating future scalability pathways
- Experimented under realistic stress conditions without disrupting the performance of our connected clients
- Verified the ability of our system to cover current and short-term needs
- Developed a dosckerized load testing framework that emulates our production level environment!



domX IoT platform architecture

Business Impact

EXPAND offered vast business impact:

- Identified the capacity limitations of our existing setup
- Identified the core platform specs that need to be adapted to deliver improved performance
- Developed our platform extension roadmap!
- Saved mission critical money to be be invested in product development.

Further plans

Planned follow-up experimental directions include:

- Stress test our platform under even more dynamic and extreme load conditions.
- Implement auto-scaling as well as auto-healing capabilities testing.

Value perceived

FED4FIRE

The direct value of EXPAND for domX has been vast:

- **gained knowledge on experimenting with clustering technologies** of MQTT brokers, by deploying:
  - replicas on single VMs
  - replicas on separate VMs
- **experimented under realistic** workloads that consider traffic loads of own products
- **tested multiple configurations** on selecting configurations for ideal server setup
- **tested fault tolerant configurations** as mitigation strategies
- **Realized** the performance capabilities of our existing platform setup

Feedback

domX

EXPAND

EXtreme PerformANce testing of custom IoT workloaDs

Feedback

FED4FIRE

Experiment environment

1. We initially planned to use the CityLab testbed, but we faced firewall issues.
2. Migrated everything to the Virtual Wall 2 testbed and deployed the load generation as well VerneMQ configuration entirely.
3. Although the provided resources were more powerful than we anticipated, we integrated resource constraints

Experiments execution

1. The collected results exceeded the initial expectations
   - provided results for future strategic planning
   - outweigh advantages and disadvantages of each configuration
2. Replication of own setup was smooth
3. Online documentation was minimal and can be improved
4. Great collaboration and support by Virtual Wall 2 testbed team

Feedback

User Tools

EXPAND utilized:

- **JFed** for deploying reproducible experiments
- **Docker & Docker-Compose,** for adaptive VM configuration
- **Docker swarm,** for MQTT Broker replication
- **Locust,** for realistic workload generation

Testbed Resources

EXPAND employed Virtual Wall 2 available testbed resources:

- easy setup and configurable approaches
- constantly online and remotely accessible
- minimal time to set up and run the experiments
- minimal repetition on deployment

Added value of Fed4FIRE+

FED4FIRE

Usefulness

1. The Fed4FIRE+ offered experimentation platforms and tools are a great asset and perfectly match the company's experimentation needs.
2. The deployment of relevant resources by domX would not be affordable. In addition, the ability to financially support the execution of experiments is quite important especially for micro-SMEs like domX, which do not have the ability to finance R&D activities with their own funds.
3. Great environment and support required for promoting R&D of startups

Key offerings

1. Availability of testbed resources
2. Heterogeneity of testbed facilities
3. Realistic experimentation conditions
4. Availability of experimentation tools
5. Continuous remote testbed availability
6. Technical support by expert people
7. Combined infrastructure types (IoT, Cloud servers etc)
8. Software interfaces to manage infrastructure (JFed)
9. Resources power (CPU, RAM etc)
10. Freedom of resource customization
11. Ease of experimentation setup
12. Availability of documentation & other resources (highly qualified testbed experts)

Testimonial

"We are now more confident that our product has been stress-tested under realistic conditions. Through the wide set of executed experiments, we are now able to achieve better infrastructure and cost planning in the long term, along with better adaptation of platform specs to the prevailing IoT loads."

EXPAND

EXtreme PerformANce testing of custom IoT workloaDs

WWW.FED4FIRE.EU

Co-funded by the European Union

Co-funded by the Swiss Confederation

This project has received funding from the European Union's Horizon 2020 research and innovation programme, which is co-funded by the European Commission and the Swiss State Secretariat for Education, Research and Innovation, under grant agreement No 732638.