OPENBACH, OPEN METROLOGY TESTING FRAMEWORK

Global technical presentation
E. Dubois (CNES), D. Pradas (Viveris Technologies)
The benchmark shall be based as much as possible on open-source components.

The benchmark shall be able to be integrated in different types of equipments, servers, clients, hardware and software, with minimal adaptation effort.

The benchmark shall be easily scalable.

OpenBACH shall have a modular structure to facilitate the addition of new:

- software tools
- monitoring parameters
- tasks

All components/entities synchronized
METROLOGY GENERAL USE CASES

Phase 1: Installation/deployment
- Deploy tools/software in entities under test

Phase 2: Preparation of scenarios and launch tests
- Configure OS/Tools/Entities
- Schedule tests/tasks
- Launch tools/software
- Generate traffic

Phase 3: Monitoring
- Check the status of entities/tools/softwares/tasks
- Collect and visualise measures

Phase 4: Results analysis
- Post-processing of results
- Create reports

- Easy and straightforward installation/deployment
- Configuring Jobs/entities and schedule scenarios
- Supervising the benchmark and collecting measures
- Performing post-processing tasks
PRELIMINARY DESIGN

- Controller Component
- Agent Component
- Collector Component
- Auditorium Component
**Job**: A number of individual tasks (one or more) with a common purpose and to be executed in a sole Agent.

**Job instance**: An execution of a job configured with a set of parameters.

**Openbach-function**: Function defined and executed by the Controller allowing to start/stop job/scenario instances, install agents/Jobs, perform information/status requests, etc.

**Scenario**: Set of openbach-functions that allow to perform different tasks might be executed in different Agents to accomplish an action.

**Scenario instance**: An execution of a scenario with a set of parameters.

**Project**: Allows for scenario organization in OpenBACH

**Entity**: Agent representation in a project.
Based on 4 components:

- **Controller**: Centralise and deploy configuration, including jobs/scenarios.
- **Collector**: Centralise the collection of data/status and log messages.
- **Agent**: deployed in the entities under test (WS, ST, proxy, server, etc.)
- **Auditorium**: Centralise the frontends/web interfaces of configuration and monitoring.
DETAILED ARCHITECTURE
Available functions in the Controller:

- **Add/install (delete/remove) Agents and Jobs to/from the benchmark**
- **List** the available Agents and the available jobs per Agent.
- Create/modify/delete a scenario.
- **Configure/launch/stop** scenario instances.
- List the available scenario and scenario instances and their **status**.
- **Send commands** of schedule/start/stop of Jobs instances to the corresponding Agents.
- List the scheduled/started job instances and their **status**.

*How to deploy new Agents, new jobs in the entities (and their dependencies)?*

**Ansible!** Off-the-shelf framework for deployment of configuration, automation of common tasks, etc.
- **Via SSH**
- Idempotence concept
• Administration

**Agents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Collector</th>
<th>Uninstall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>172.20.34.39</td>
<td>172.20.34.39</td>
<td></td>
</tr>
</tbody>
</table>

**Projects topology**

```
Controller

DITG project
```

**Add agent**

- Name
- IP Address
- Username
- Password
- Collector
AUDITORIUM HMI

- Project

Topology

Entity

mptcp proxy server
The MPTCP proxy on the server side

- Description: The MPTCP proxy on the server side
- OS: Linux
- Networks:
  - satellite link
  - 4g link
- Agent:
  - Name: p6_agent3
  - IP: 172.20.34.36
  - Username: exploit
  - Collector: 172.20.34.32
  - Installed jobs: No jobs installed

INSTALL JOBS

UNINSTALL
• Scenario status

ping

ping
(id: 278) 2017-05-19 10:04:09 → 2017-05-19 10:04:28 [Duration of 0:00:18]

fping start_job_instance

fping (172.20.34.39)
(id: 240)

hping start_job_instance

hping (172.20.34.39)
(id: 241)
• Collection and display of log messages
LOGS
• Collection and display of statistics/measures
STATS

![Graph of statistics for endpoints and servers](image)

**Endpoints:** google

**Collectors:** amsterdam

**Multi-value**

**Servers:** backend_01
Main technical points in progress:

- Technical Jobs
- Network integration/configuration
- Postprocessing
- System configuration
- Multi-user benchmark
- System integration
- HMI Evolutions