

# TP3: Ferramenta para monitorização de QoS na Internet

## Entrega intermédia

Simão Cunha<sup>[a93262]</sup>, Gonçalo Pereira<sup>[a93168]</sup>, and Rui Alves<sup>[pg50745]</sup>

Universidade do Minho - Campus de Gualtar, R. da Universidade, 4710-057 Braga Portugal

### Qualidade de Serviço na Internet (2022/2023) - Grupo 1

## 1 Definição do Problema

Neste trabalho pretendemos desenvolver uma ferramenta que automaticamente executa testes de bandwidth, jitter, packet loss, throughput e one-way delay a um servidor remoto da Google Cloud [1] apresentando os resultados ao utilizador. Esta ferramenta tem o objetivo de auxiliar utilizadores que pretendem dar deploy à sua aplicação num servidor Google Cloud fornecendo-o de forma rápida e automática acesso a métricas relevantes do servidor em causa.

## 2 Métricas sob análise e porquê

Na implementação desta ferramenta decidimos implementar unicamente medições ativas sendo estas, como referido anteriormente, bandwidth, jitter, packet loss, throughput e one-way delay. O que nos levou a escolher estas métricas foi o facto de serem as mais relevantes num contexto de comunicação a um servidor remoto. A razão de termos escolhido medir one-way delay ao invés do ping deve-se ao facto do one-way delay transmitir informação mais útil para análise e porque pacotes ICMP do ping não tem prioridade nas queues não refletindo assim, em alguns casos, a realidade.

## 3 Implementação

A implementação divide-se em duas partes distintas *deployment* e *app*. A primeira implementa a automatização de lançamento e de configuração (instalação de software e dependências necessárias) de máquinas virtuais na cloud. O *deployment* não é utilizado pelo utilizador mas sim por quem vai disponibilizar o serviço. A parte da aplicação implementa a lógica de efetuar as medições à máquina escolhida e de efetuar sincronização NTP [5] para o caso da medição do one-way delay. Tanto o *deployment* e *app* são implementados com recurso a Ansible [2], a ferramenta de medição do one-way delay é um programa python desenvolvido por nós que implementa um cliente-servidor com envio de 3 pacotes de proba as outras métricas são recolhidas fazendo uso de iperf [3] e perf3 [4].

A figura abaixo ilustra a arquitetura descrita acima.

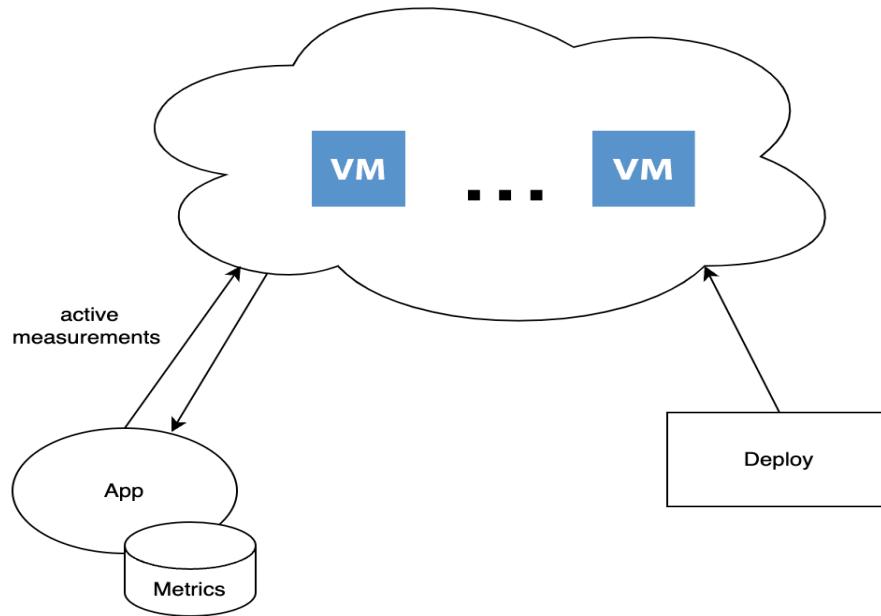


Figura 1: Arquitetura

## 4 Limitações identificadas

Este trabalho prático possui algumas limitações, tais como:

- Resultados são guardados localmente na máquina do utilizador.
- Não implementa uma base de dados remota que agrupa as medições de todos os utilizadores.
- Não implementa um UI e modo de utilização intuitiva a qualquer utilizador
- Só permite testar um servidor de cada vez.

## 5 Testes

Como referido anteriormente, neste trabalho prático estamos a recolher métricas de QoS sobre a ligação a vários servidores da Google Cloud localizados em vários pontos do mundo. Para tal, faremos o *deployment* de uma VM em cada uma dessas zonas, recolhendo os valores da largura de banda, jitter, perda de pacotes, bitrate e one-way delay. De seguida, juntamente com uma *script* em Python, iremos fazer o *parsing* destes valores para um ficheiro CSV guardado localmente onde cada entrada corresponde a um teste. As colunas deste ficheiro CSV são:

- Zone: Zona para onde o teste foi feito;
- Datetime: Timestamp com o formato [Dia da semana por extenso] [Dia da semana numérico] [Mês por extenso] [Hora numérica]:[Minuto numérico] - por exemplo: Thursday 27 April 18:35;
- Bandwidth: Valor da largura de banda obtida por *iperf* no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 87.8 M, que representa 87.8 Mbps;
- Upload Jitter: Valor medido do jitter da nossa máquina até ao servidor medido em milissegundos;
- Download Jitter: Valor medido do jitter do nosso servidor até à nossa máquina medido em milissegundos;

- Upload Packet Loss: Valor medido em percentagem dos pacotes perdidos da nossa máquina até ao servidor;
- Download Packet Loss: Valor medido em percentagem dos pacotes perdidos do servidor até à nossa máquina;
- Upload Bit Rate (With UDP): Valor médio medido do débito com UDP da nossa máquina até ao servidor - no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 1.05 M, que representa 1.05 Mbps;
- Download Bit Rate (With UDP): Valor médio medido do débito com UDP do servidor até à nossa máquina - no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 1.05 M, que representa 1.05 Mbps;
- Max Bit Rate (With UDP): valor máximo medido do débito com UDP durante o teste - no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 1.06 M, que representa 1.06 Mbps;
- Upload Bit Rate (With TCP): Valor médio medido do débito com TCP da nossa máquina até ao servidor - no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 90.0 M, que representa 90.0 Mbps;
- Download Bit Rate (With TCP): Valor médio medido do débito com TCP do servidor até à nossa máquina - no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 88.4 M, que representa 88.4 Mbps;
- Max Bit Rate (With TCP): valor máximo medido do débito com TCP durante o teste - no formato [Valor numérico] [Caractere com a unidade do valor obtido] - por exemplo: 106.0 M, que representa 106.0 Mbps;
- One-Way Delay: valor médio do one-way delay da nossa máquina para o servidor (expresso em ms);

Depois da recolha dos dados, criamos um ficheiro Jupyter Notebook para efetuarmos análise dos dados recolhidos. À data da entrega deste relatório, ainda não recolhemos dados nem efetuamos nenhuma análise, no entanto, na entrega final pretendemos entregar um estudo de desempenho e se tal o permitir apresentar algumas conclusões.

## Referências

1. Website Google Cloud: <https://cloud.google.com/> (consultado em abr. 2023)
2. Website Ansible: <https://www.ansible.com/> (consultado em abr. 2023)
3. Website iperf: <https://iperf.fr/> (consultado em abr. 2023)
4. Documentação iperf3: <https://software.es.net/iperf/> (consultado em abr. 2023)
5. Documentação NTP: <https://www.geeksforgeeks.org/network-time-protocol-ntp/> (consultado em abr. 2023)