

Neste trabalho foi implementado um sistema distribuído utilizando o padrão de pipeline, que realiza um tratamento em uma carga de imagens, produzindo imagens em escala de cinza, verde, vermelho e azul. As imagens usadas são do site [This Person Does Not Exist](#), que usa inteligência artificial para gerar rostos ultra realistas de pessoas fictícias. Foi utilizada a biblioteca [Sucuri](#), na sua versão atualizada para Python 3, para simular os núcleos do processador utilizando grafos, chamados de workers. As funcionalidades de leitura e escrita dos arquivos, a manipulação das imagens, entre outros, é distribuída em um núcleo criado pela biblioteca. Para mensurar o desempenho do sistema, foi utilizada a biblioteca pyDF, enquanto a comunicação cliente-servidor foi realizada através do protocolo xml-rpc. O processamento das imagens foi feito utilizando a biblioteca PIL (Python Imaging Library). Para executar o programa basta iniciar o servidor.py e executar o cliente.py indicando na linha de comando quantos workers serão criados (ex.: python cliente.py 3)

Métricas de desempenho (QoS, latência, speedup, eficiência, throughput, bandwidth)

| Qtd. de núcleos de processador (workers) | Tempo de processamento em segundos |
|--|------------------------------------|
| 1 | 24.840 |
| 2 | 24.141 |
| 3 | 23.918 |
| 4 | 23.853 |
| 5 | 23.818 |
| 6 | 23.804 |

Latência: 24.072 segundos

Speedup: 1.04

Eficiência: 0.173

Throughput: 0.168 Kbps

Bandwidth: 6.487 Kbps

A partir desses dados é possível ver que há uma queda no tempo de execução da tarefa ao adicionar um núcleo de processamento, porém a taxa de queda no tempo decai conforme vamos adicionando mais workers. Logo, existe um ponto onde pode não ser tão vantajoso adicionar mais workers pois o aumento se torna muito pequeno.