



Clusters de RaspberryPi utilizando PP (parallel Python)

RODRIGO CARVALHO
GUILHERME TABACOW
GUSTAVO VILLELA
LUCAS COELHO
RAPHAEL HUNGRIA

São Caetano do Sul, Outubro de 2019

Objetivos

- ▶ Criação de um cluster.
- ▶ Verificar desempenhos.
- ▶ Analisar métodos para o funcionamento.
- ▶ Montagem e instalação.

Equipamento

Switch



Equipamento

Raspberry Pi 3b



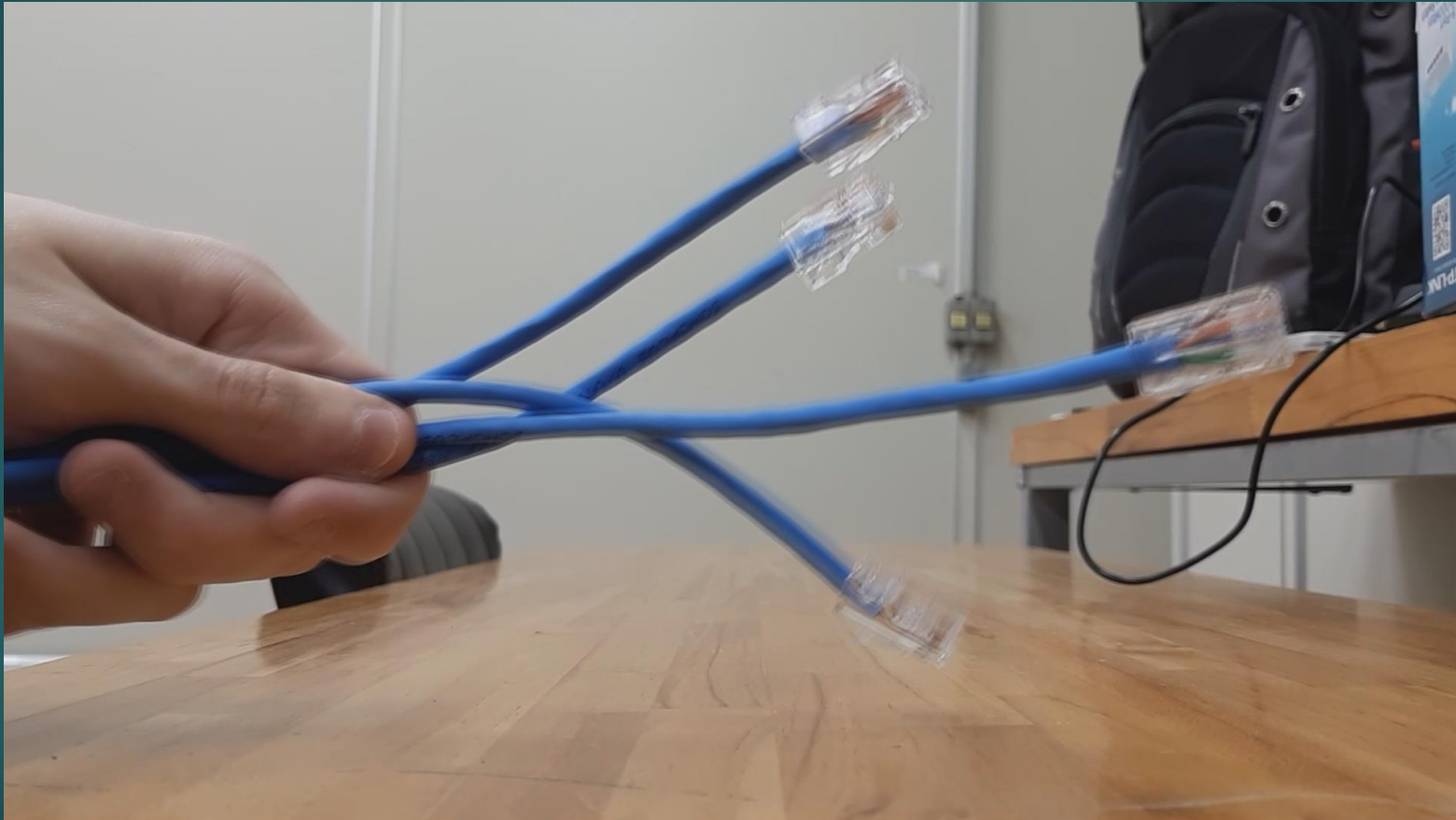
Equipamento

Adaptador de cartão sd para usb



Equipamento

Cabo de rede



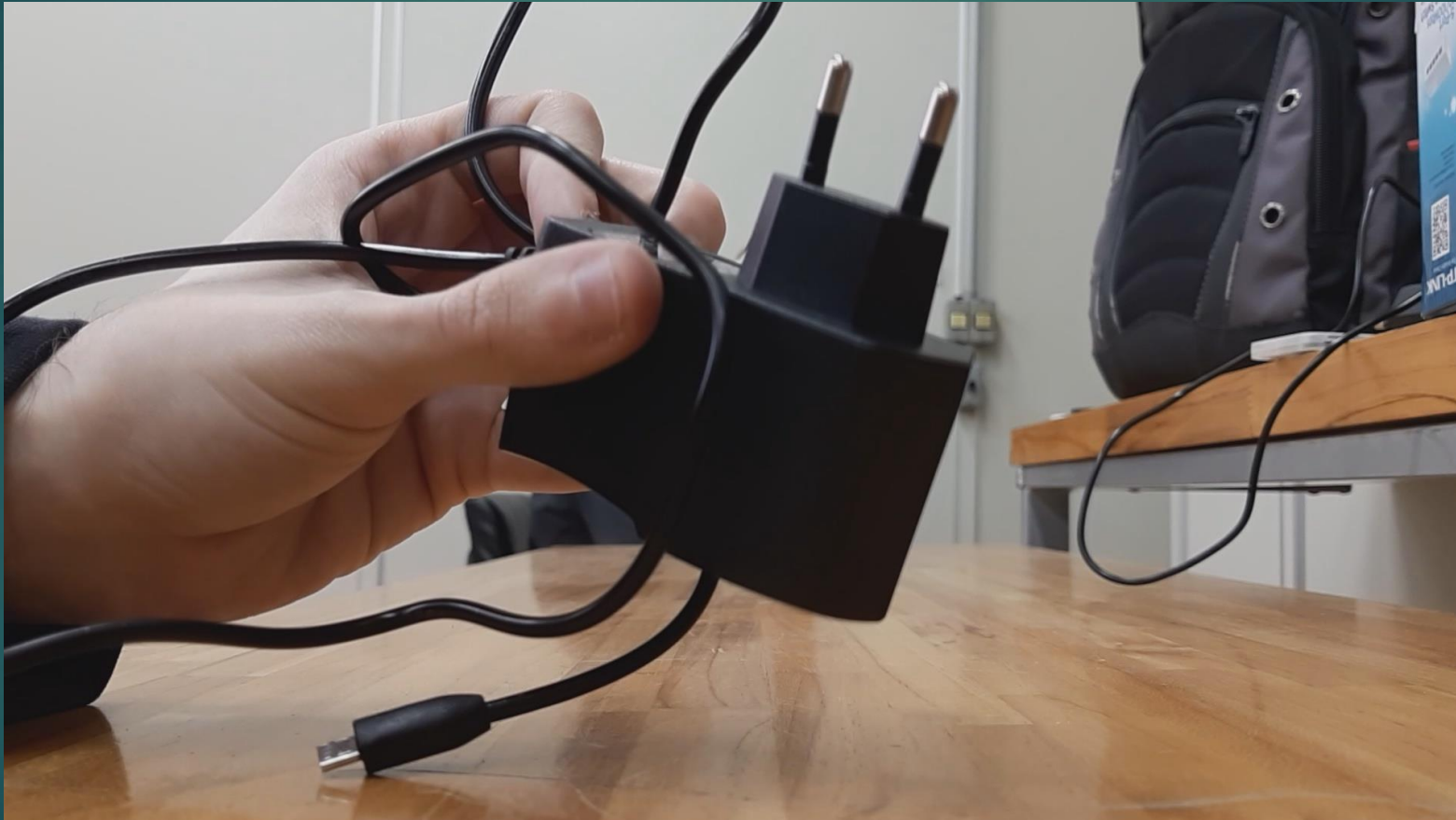
Equipamento

Cartão SD 16 gb class 10



Equipamento

Fonte com entrada micro-usb



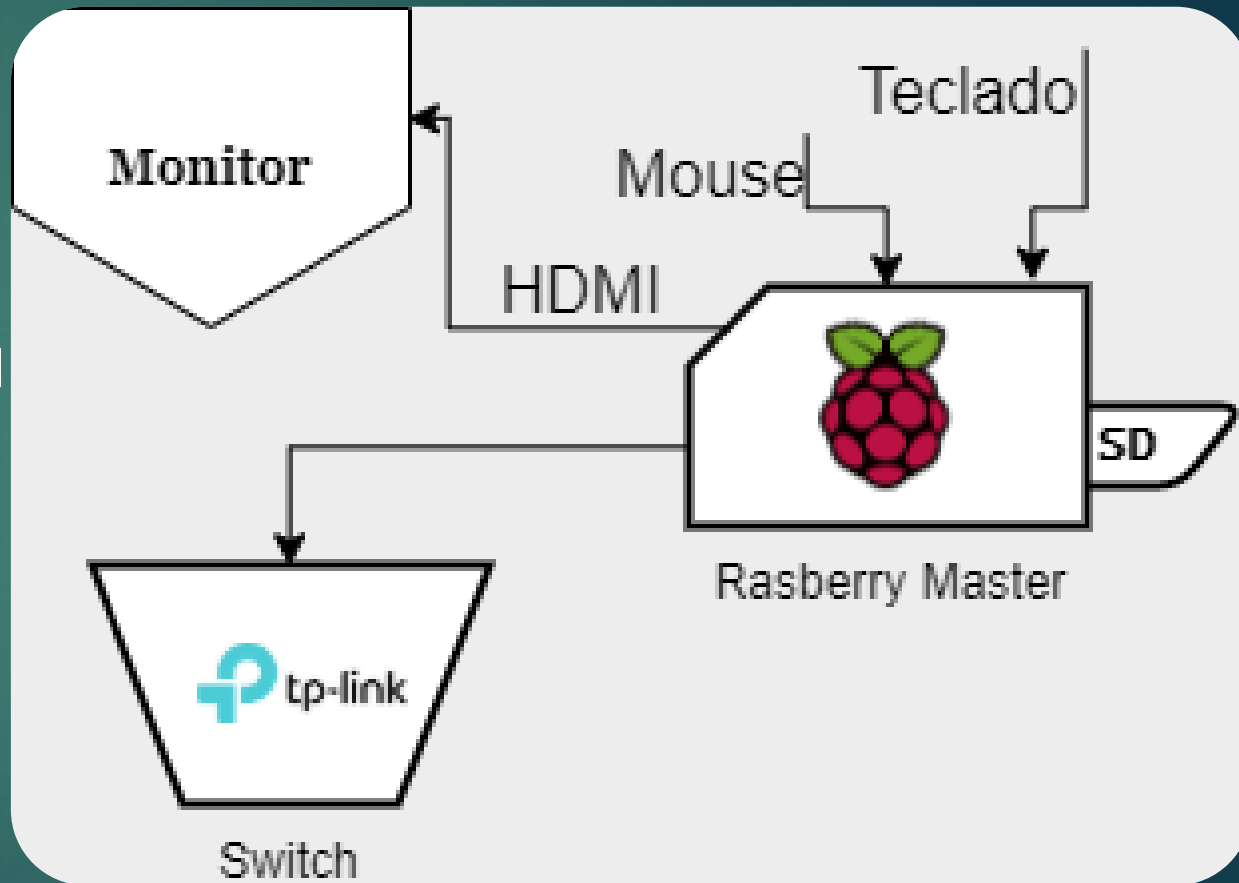
Equipamento

Fonte do Switch

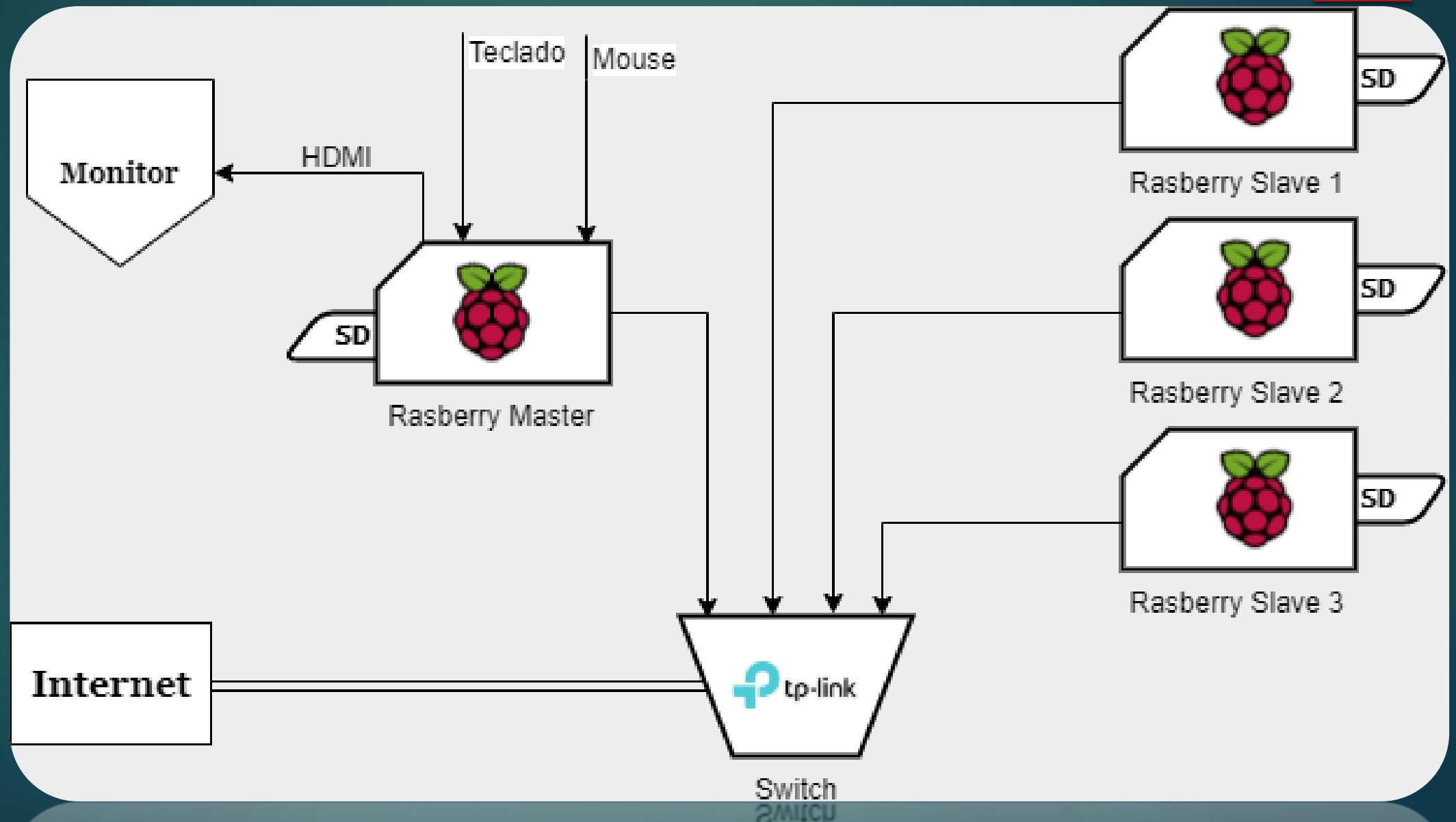


Montagem

- ▶ Flash nos SDs
- ▶ Colocar os SDs em cada raspberry
- ▶ Conectar os Raspberrys ao Switch
- ▶ No master, conectar teclado, mouse e HDMI
- ▶ Configurar cada Raspberry individualmente
- ▶ Rodar o programa no master



Montagem

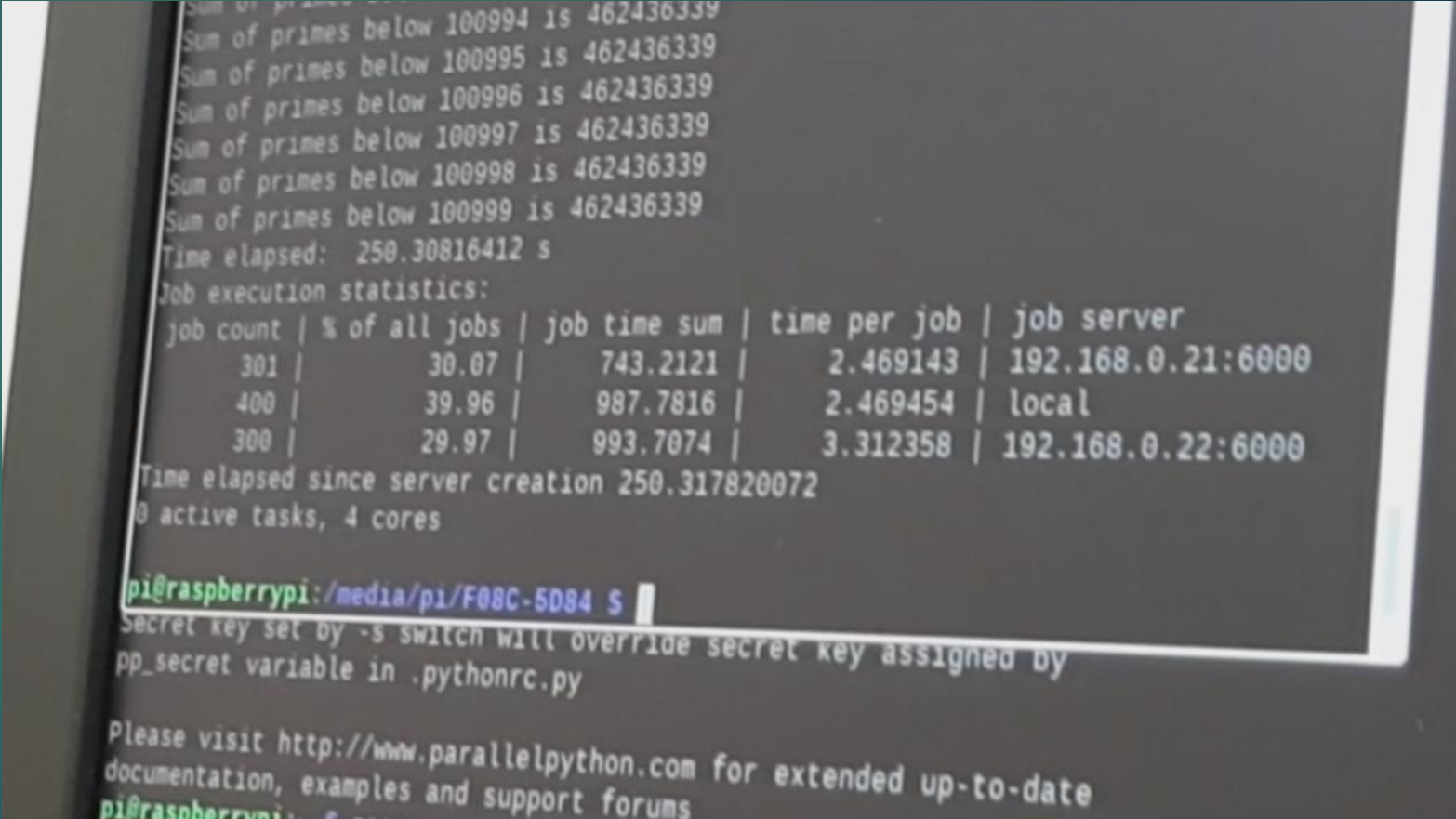


Configuração dos Raspberrys

- ▶ No raspberry escolhido como master foi usada uma distribuição do Raspbian com interface gráfica, e no restante com CLI.
- ▶ Os raspberrys foram conectados na internet e atualizados.
- ▶ O Python e o pacote Parallel Python foram instalados em todos.
- ▶ As configurações de IP foram alteradas para ter o IP estático em todas as máquinas.
- ▶ Nos raspberrys que servirão de slaves, o servidor SSH foi configurado para permitir o acesso remoto.
- ▶ Os slaves foram acessados remotamente a partir do master, e neles rodamos uma instancia de servidor do Parallel Python. Isso permite que o programa rodado no master seja distribuído por todos.

Análise de desempenho do cluster

- Geralmente: Mais nós no cluster, melhor desempenho!



```
Sum of primes below 100994 is 462436339
Sum of primes below 100995 is 462436339
Sum of primes below 100996 is 462436339
Sum of primes below 100997 is 462436339
Sum of primes below 100998 is 462436339
Sum of primes below 100999 is 462436339
Time elapsed: 250.30816412 s
Job execution statistics:
  job count | % of all jobs | job time sum | time per job | job server
          301 |      30.07 |    743.2121 |    2.469143 | 192.168.0.21:6000
          400 |      39.96 |    987.7816 |    2.469454 | local
          300 |      29.97 |    993.7074 |    3.312358 | 192.168.0.22:6000
Time elapsed since server creation 250.317820072
0 active tasks, 4 cores

pi@raspberrypi:/media/pi/F08C-5D84 $
Secret key set by -s switch will override secret key assigned by
pp_secret variable in .pythonrc.py

Please visit http://www.parallelpython.com for extended up-to-date
documentation, examples and support forums
pi@raspberrypi:~$
```


Conclusão

- ▶ Na aplicação realizada, tivemos apenas 30% de melhora.
- ▶ Eficientes para aplicações maiores
- ▶ Preço relativamente baixo
- ▶ Montagem simples
- ▶ Boa "alternativa" para supercomputadores

Informações adicionais

- ▶ Clusters funcionam melhor com máquinas "simétricas"
- ▶ Fácil manutenção sem a necessidade de atrapalhar o funcionamento
- ▶ "Não é necessário depender de um único fornecedor ou prestador de serviço para reposição de componentes" - [Emerson Alecrim](#)
- ▶ Não é "absoluto"
 - ▶ Espaço.
 - ▶ Rede local, portanto, não permite acesso distante.
 - ▶ Bottleneck de comunicação.

Bibliografia

- ▶ <https://jack.eti.br/a-velha-curiosidade-sobre-clusters/>
- ▶ <https://www.infowester.com/cluster.php>
- ▶ <https://www.parallelpython.com/>
- ▶ <https://medium.com/@glmdev/building-a-raspberry-pi-cluster-784f0df9afbd>
- ▶ <https://www.machinelearningplus.com/python/parallel-processing-python/>



Clusters de RaspberryPi utilizando PP (parallel Python)

RODRIGO CARVALHO
GUILHERME TABACOW
GUSTAVO VILLELA
LUCAS COELHO
RAPHAEL HUNGRIA

São Caetano do Sul, Outubro de 2019