

Trabalho Prático de

# Sistemas Críticos

## 2024 / 2025

Trabalho elaborado por:  
8220302 - Leandro Afonso

Curso de:  
**Licenciatura em Segurança Informática em Redes de Computadores**

Docentes:  
António Alberto dos Santos Pinto ([aap@estg.ipp.pt](mailto:aap@estg.ipp.pt))  
Hélder Renato Xisto Leão Gonçalves ([hrlg@estg.ipp.pt](mailto:hrlg@estg.ipp.pt))

Felgueiras, 4 de fevereiro de 2025

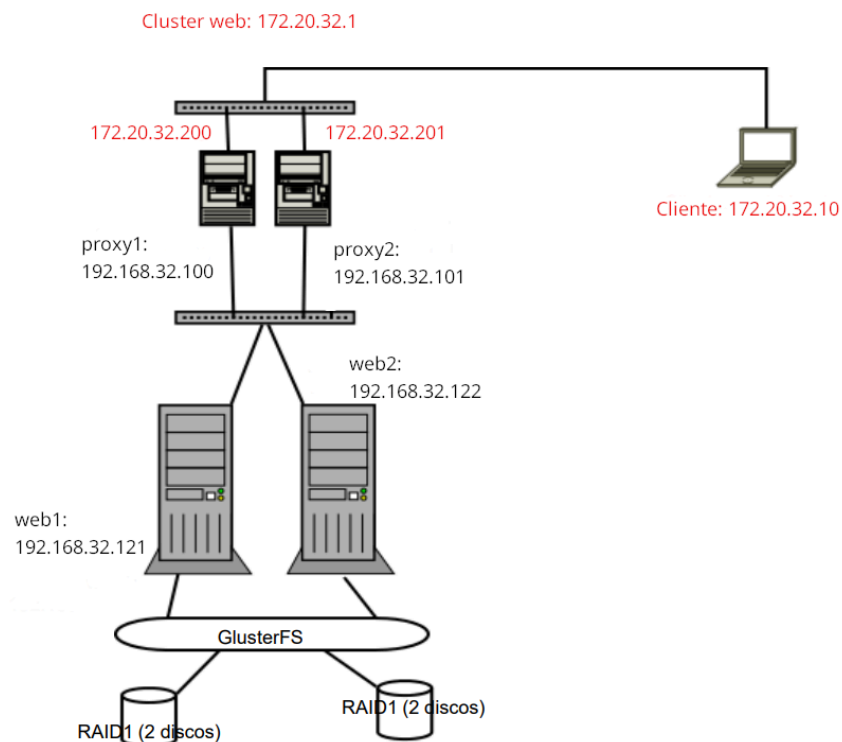
# Índice

Introdução.....	3
Desenho da Infraestrutura.....	3
Configuração das máquinas.....	4
Configuração Web.....	4
Configuração dos HAProxy.....	8
Avaliação da Infraestrutura.....	9
Ferramenta de teste.....	10
Ambas máquinas web ativas:.....	10
Apenas uma máquina web:.....	12
Conclusão.....	15

## Introdução

Este relatório documenta o desenvolvimento e a implementação de um *mini-cluster* altamente disponível, tolerante a falhas e com balanceamento de carga, utilizando o sistema operacional Ubuntu Server como base, com a exceção do cliente em que foi usado o Fedora com o GNOME. O projeto foi realizado no âmbito da unidade curricular de Sistemas Críticos, com o objetivo de criar uma infraestrutura robusta para suportar as operações de *e-commerce* de uma empresa fictícia. Durante a realização deste trabalho, foram explorados conceitos fundamentais de alta disponibilidade e sistemas distribuídos, aplicados por meio de serviços como *NGINX* e *GlusterFS*. O relatório detalha as decisões tomadas, as configurações implementadas e os testes realizados para avaliar o desempenho e a resiliência do sistema, sem o uso de ferramentas opcionais como o *Vagrant*.

## Desenho da Infraestrutura



# Configuração das máquinas

## Configuração Web

Havendo configurações//comandos que foram feitas//executados em todas as máquinas, estas foram todas feitas numa unica máquina base e depois clonadas, com a exceção do cliente. Inicialmente foi executado o comando ***apt update && apt upgrade -y*** como *root*. Sendo que a configuração de todas as máquinas foi realizada como *root*.

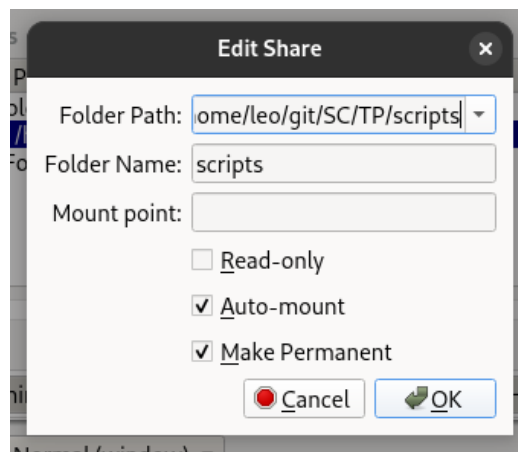
De seguida executei ***apt install netplan.io*** para instalar o **netplan**, facilitando a atribuição de IP's nas máquinas por interface.

```
GNU nano 7.2 /etc/netplan/00-installer-config.yaml
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      addresses: [192.168.32.121/24]
```

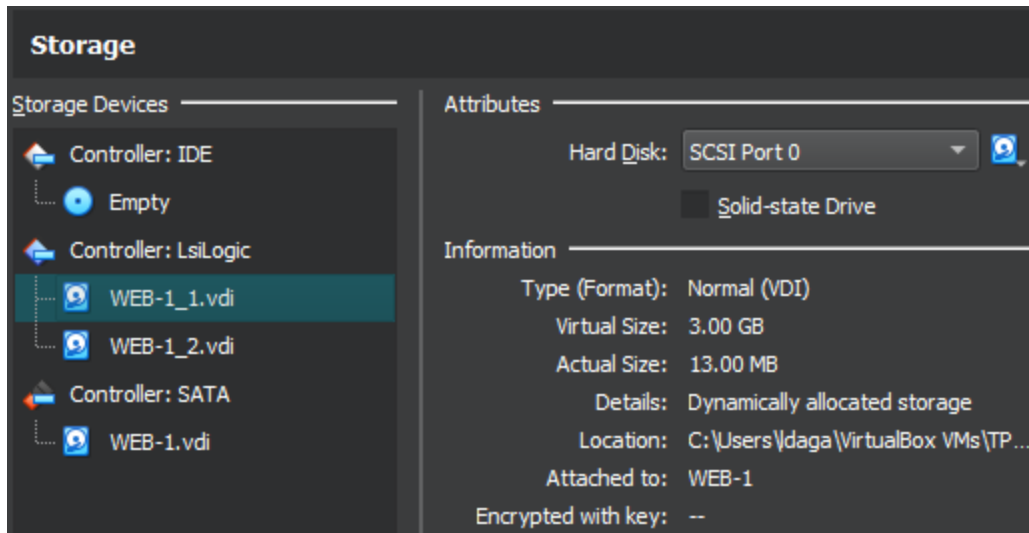
Sendo este IP alterado por máquina e até adicionando uma interface extra nas máquinas de *proxy*.

Esta configuração pode ser aplicada executando ***netplan apply***.


Após isto é partilhada uma pasta do sistema *host*, com o ponto de montagem em */scripts*, em que se encontram não só *scripts*, mas também ficheiros pré-configurados para certos serviços.



É de mencionar que é adicionado ao */etc/fstab* de cada máquina uma linha que monta automaticamente este *path* cada vez que esta é reiniciada, com a exceção do cliente.



Aqui adicionamos 2 discos de 3GB cada, com tamanho dinamicamente alocado.

```
scripts >  raid-setup.sh
1  #!/bin/bash
2
3  # instalação mdadm
4  apt install mdadm -y
5
6  # criação do raid1
7  mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc -R
8
9  # criar o sistema de arquivos
10 mkfs.ext4 /dev/md0
11 mkdir /raid1
12 mount /dev/md0 /raid1
13 echo '/dev/md127 /raid1 ext4 defaults 0 0' | tee -a /etc/fstab
```

Este script é responsável por instalar o *mdadm* e criar os sistema de RAID-1, formatá-lo como ext4 e montá-lo em */raid1*.

```

1  #!/bin/bash
2
3  # instalação glusterfs
4  apt install glusterfs-server -y
5
6  # ativar e iniciar o serviço
7  systemctl enable --now glusterd
8
9  # criar o diretorio onde o glusterfs vai ser montado
10 mkdir -p /raid1

```

Aqui instalamos o GlusterFS e iniciamos o serviço.

```

scripts > monitor-setup.sh
1  #!/bin/bash
2
3  apt install snmpd snmp libsnmp-dev -y
4
5  cp /scripts/snmpd.conf /etc/snmp/snmpd.conf
6
7  # Install Ganglia
8  apt install ganglia-monitor -y
9
10 # Create the /etc/ganglia directory if it doesn't exist
11 mkdir -p /etc/ganglia
12
13 # Copy Ganglia configuration file
14 cp /scripts/gmond.conf /etc/ganglia/gmond.conf
15
16 # Restart and Enable the service
17 systemctl restart ganglia-monitor
18 systemctl enable ganglia-monitor
19
20 systemctl enable snmpd
21
22 reboot

```

Aqui é instalado o monitor do *Ganglia*, um serviço de monitorização.

```
$ glusterfs.sh
1  #!/bin/bash
2
3  STORAGE_DIR=/raid1
4  VOLUME_NAME=storage
5
6  SERVER1=192.168.32.121
7  SERVER2=192.168.32.122
8
9  # https://docs.gluster.org/en/latest/Install-Guide/Configure/
10 gluster peer probe $SERVER1
11 gluster peer probe $SERVER2
12
13 gluster volume create $VOLUME_NAME replica 2 $SERVER1:$STORAGE_DIR $SERVER2:$STORAGE_DIR force
14 gluster volume start $VOLUME_NAME
```

Já este script pode ser executado em qualquer máquina *web*, tanto é que este automaticamente é executado ao correr o *web.sh*. Este é responsável por montar os vários ficheiros partilháveis entre as várias máquinas.

```
scripts > mount-glusterfs.sh
1  #!/bin/bash
2
3  mkdir -p /cluster
4  echo 'localhost:/storage /cluster glusterfs defaults,_netdev 0 0' | tee -a /etc/fstab
5  mount -a
```

Este *script* adiciona o ponto de montagem do *GlusterFS* ao *fstab* e monta de seguida este sistema.

```
scripts > web-setup.sh
1  #!/bin/bash
2
3  # Install Docker Compose and MySQL Client for testing purposes
4  apt install docker.io docker-compose -y
5
6  # Enable and start Docker
7  systemctl enable --now docker
8
9  # Pull the images for a faster deployment
10 docker-compose --project-directory /scripts/ pull
11 docker pull ubuntu:latest
```

Nestas máquinas é executado este script de instalação do *Docker* e *docker-compose* visto que em termos de *gestão* de portas achei mais simples.

```

$ web.sh
1  #!/bin/bash
2
3  # Mount GlusterFS on the /cluster folder
4  /scripts/mount-glusterfs.sh
5
6  # Create the folders for Ganglia, Apache and Cacti
7  mkdir -p /cluster/www/ganglia/gmetad /cluster/www/ganglia/apache
8  cp /scripts/gmetad.conf /cluster/www/ganglia/gmetad/gmetad.conf
9  cp /scripts/ganglia-vhost.conf /cluster/www/ganglia/apache/ganglia-vhost.conf
10
11 # Start the containers
12 docker-compose --project-directory /scripts/ up --build -d

```

Aqui é montado o `/cluster` e criado/copiados vários ficheiros que vão ser usados pelo *container* do *Docker*.

## Configuração dos HAProxy

```

scripts > haproxy-setup.sh
1  #!/bin/bash
2
3  # Install HAProxy, Pacemaker, Corosync and Pcs
4  apt install haproxy corosync pacemaker pcs -y
5
6  # Copy HAProxy and Corosync configuration files
7  cp /scripts/corosync-haproxy.conf /etc/corosync/corosync.conf
8  cp /scripts/haproxy.cfg /etc/haproxy/haproxy.cfg
9
10 # Start the services
11 systemctl enable --now corosync pacemaker pcsd
12
13 # Create the HAProxy resource
14 pcs resource create haproxy systemd:haproxy op monitor interval=1s
15
16 # Disable STONITH since we won't be using it
17 pcs property set stonith-enabled=false
18
19 # Create the VirtualIP resource
20 pcs resource create virtualip ocf:heartbeat:IPaddr2 ip=172.20.32.1 cidr_netmask=24 op monitor interval=1s
21
22 # Set the HAProxy resource to start with the VirtualIP resource
23 pcs constraint colocation add haproxy with virtualip INFINITY --force
24
25 # Set the migration threshold
26 pcs resource update haproxy meta migration-threshold=0
27 pcs resource update virtualip meta migration-threshold=0
28
29 # Restart the services
30 systemctl restart corosync pacemaker pcsd

```

Aqui copiamos os arquivos de configuração do Corosync e do HAProxy para seus diretórios correspondentes, garantindo que o sistema utilizará os parâmetros previamente definidos. Com os arquivos configurados, habilitamos e iniciamos os serviços essenciais para a



comunicação e gerenciamento do cluster, permitindo que sejam executados automaticamente após reinicializações do sistema. Após a inicialização dos serviços, definimos o HAProxy como um recurso gerenciado pelo Pacemaker e configuramos um monitoramento periódico para verificar sua disponibilidade. Também ajustamos a configuração do cluster para desativar o mecanismo de STONITH, uma vez que não será utilizado neste ambiente. Para disponibilizar o serviço de forma transparente aos clientes, criamos um recurso de IP virtual que permitirá o acesso ao serviço independentemente do nó que estiver ativo no momento. Além disso, estabelecemos uma regra no cluster para garantir que o HAProxy só será iniciado caso o IP virtual esteja ativo, garantindo que os serviços sejam disponibilizados corretamente.

## Avaliação da Infraestrutura

A avaliação da infraestrutura foi conduzida para testar sua escalabilidade, desempenho e robustez. O principal objetivo foi analisar a resposta do sistema diante do aumento da carga de utilizadores, com foco no serviço web e na base de dados, assegurando que o balanceamento de carga e os mecanismos de alta disponibilidade funcione conforme esperado.

Os principais pontos avaliados foram:

- **Escalabilidade do Sistema Web:** Verificar o comportamento da infraestrutura diante do aumento do número de requisições simultâneas aos servidores web (web1 e web2), analisando o impacto no uso da CPU, memória e interfaces de rede.
- **Desempenho do Balanceador de Carga:** Avaliar a eficiência do HAProxy na distribuição equilibrada do tráfego entre os servidores, além de testar sua resiliência e a capacidade de manter a disponibilidade dos serviços em caso de falha de um dos proxies ou servidores.
- **Monitoramento com Ganglia:** Utilizar a ferramenta Ganglia para acompanhar o consumo de recursos do sistema, como CPU, memória e tráfego de rede, durante os testes de carga.

## Ferramenta de teste

Para a execução destes testes usei o Apache Benchmark (ab), ferramenta este que permite fazer pedidos simultâneos a um servidor *web*. Com isto em mente, os comandos usados foram os seguintes:

```
ab -n 32000 -c 30 http://172.20.32.1/ganglia
ab -n 32000 -c 60 http://172.20.32.1/ganglia
ab -n 32000 -c 120 http://172.20.32.1/ganglia
ab -n 32000 -c 240 http://172.20.32.1/ganglia
```

Que, como resultado, determinaram o seguinte:

### Ambas máquinas web ativas:

```
leapvbox:~$ ab -n 32000 -c 30 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests

Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:      169 bytes

Concurrency Level:    30
Time taken for tests: 26.503 seconds
Complete requests:    32000
Failed requests:      0
Non-2xx responses:    32000
Total transferred:    11712000 bytes
HTML transferred:    5408000 bytes
Requests per second:  1207.41 [#/sec] (mean)
Time per request:     24.847 [ms] (mean)
Time per request:     0.828 [ms] (mean, across all concurrent requests)
Transfer rate:        431.55 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:  0    0  0.2    0    5
Processing: 7  24  4.2   23   48
Waiting:  2  24  4.2   23   48
Total:    7  25  4.2   24   48

Percentage of the requests served within a certain time (ms)
 50%    24
 66%    25
 75%    26
 80%    27
 90%    30
 95%    35
 98%    38
 99%    39
100%    48 (longest request)

leapvbox:~$ ab -n 32000 -c 60 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests

Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:      169 bytes

Concurrency Level:    60
Time taken for tests: 31.659 seconds
Complete requests:    32000
Failed requests:      0
Non-2xx responses:    32000
Total transferred:    11712000 bytes
HTML transferred:    5408000 bytes
Requests per second:  1010.76 [#/sec] (mean)
Time per request:     59.361 [ms] (mean)
Time per request:     0.989 [ms] (mean, across all concurrent requests)
Transfer rate:        361.27 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:  0    0  0.4    0    9
Processing: 9  59 12.4   56  116
Waiting:  2  59 12.4   56  116
Total:   12  59 12.4   56  116

Percentage of the requests served within a certain time (ms)
 50%    56
 66%    60
 75%    63
 80%    66
 90%    78
 95%    87
 98%    95
 99%   100
100%   116 (longest request)
```

```

leapvbox:~$ ab -n 32000 -c 120 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests


Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:      169 bytes

Concurrency Level:    120
Time taken for tests:  29.512 seconds
Complete requests:    32000
Failed requests:       0
Non-2xx responses:    32000
Total transferred:    11712000 bytes
HTML transferred:     5408000 bytes
Requests per second:  1084.31 [#/sec] (mean)
Time per request:      110.669 [ms] (mean)
Time per request:      0.922 [ms] (mean, across all concurrent requests)
Transfer rate:         387.56 [Kbytes/sec] received


Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0     0   1.5      0     41
Processing: 43    110 19.6    105    211
Waiting:    3     110 19.6    105    210
Total:      46    110 19.6    105    211


Percentage of the requests served within a certain time (ms)
 50%    105
 66%    111
 75%    116
 80%    120
 90%    138
 95%    152
 98%    169
 99%    180
100%    211 (longest request)

```

```

leapvbox:~$ ab -n 32000 -c 240 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests


Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:      169 bytes

Concurrency Level:    240
Time taken for tests:  32.152 seconds
Complete requests:    32000
Failed requests:       0
Non-2xx responses:    32000
Total transferred:    11712000 bytes
HTML transferred:     5408000 bytes
Requests per second:  995.26 [#/sec] (mean)
Time per request:      241.143 [ms] (mean)
Time per request:      1.005 [ms] (mean, across all concurrent requests)
Transfer rate:         355.73 [Kbytes/sec] received


Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0     1   1.9      0     31
Processing: 37    240 40.7    231    370
Waiting:    6     240 40.7    231    370
Total:      38    240 40.4    231    370


Percentage of the requests served within a certain time (ms)
 50%    231
 66%    250
 75%    264
 80%    274
 90%    299
 95%    320
 98%    344
 99%    353
100%    370 (longest request)

```

## Apenas uma máquina web:

```
leodvbox:~$ ab -n 32000 -c 30 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests


Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:       169 bytes

Concurrency Level:     30
Time taken for tests:  26.303 seconds
Complete requests:     32000
Failed requests:        0
Non-2xx responses:     32000
Total transferred:     11712000 bytes
HTML transferred:      5408000 bytes
Requests per second:   1216.59 [#/sec] (mean)
Time per request:      24.659 [ms] (mean)
Time per request:      0.822 [ms] (mean, across all concurrent requests)
Transfer rate:          434.84 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        0   0.2      0      5
Processing:      8       24   5.0     23     66
Waiting:         3       24   5.0     23     66
Total:           9       25   5.0     23     68

Percentage of the requests served within a certain time (ms)
 50%    23
 66%    25
 75%    26
 80%    27
 90%    31
 95%    35
 98%    39
 99%    42
100%    68 (longest request)
```

```
leodvbox:~$ ab -n 32000 -c 60 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests


Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:       169 bytes

Concurrency Level:     60
Time taken for tests:  45.146 seconds
Complete requests:     32000
Failed requests:        0
Non-2xx responses:     32000
Total transferred:     11712000 bytes
HTML transferred:      5408000 bytes
Requests per second:   708.80 [#/sec] (mean)
Time per request:      84.650 [ms] (mean)
Time per request:      1.411 [ms] (mean, across all concurrent requests)
Transfer rate:          253.34 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        0   0.5      0     13
Processing:      6      84 661.4     52 15372
Waiting:         3      84 661.4     52 15372
Total:           8     85 661.4     53 15372

Percentage of the requests served within a certain time (ms)
 50%    53
 66%    58
 75%    62
 80%    64
 90%    73
 95%    81
 98%    89
 99%    99
100% 15372 (longest request)
```

```
leodvbox:~$ ab -n 32000 -c 120 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests


Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:       169 bytes

Concurrency Level:     120
Time taken for tests:  29.611 seconds
Complete requests:     32000
Failed requests:        0
Non-2xx responses:     32000
Total transferred:     11712000 bytes
HTML transferred:      5408000 bytes
Requests per second:   1080.69 [#/sec] (mean)
Time per request:      111.040 [ms] (mean)
Time per request:      0.925 [ms] (mean, across all concurrent requests)
Transfer rate:          386.26 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        0   1.0      0     31
Processing:     24     110  20.1    107    187
Waiting:         3     110  20.1    107    187
Total:          24     111  20.1    107    188

Percentage of the requests served within a certain time (ms)
 50%    107
 66%   114
 75%   121
 80%   125
 90%   139
 95%   152
 98%   164
 99%   170
100%   188 (longest request)
```

```
leodvbox:~$ ab -n 32000 -c 240 http://172.20.32.1/ganglia
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 172.20.32.1 (be patient)
Completed 3200 requests
Completed 6400 requests
Completed 9600 requests
Completed 12800 requests
Completed 16000 requests
Completed 19200 requests
Completed 22400 requests
Completed 25600 requests
Completed 28800 requests
Completed 32000 requests
Finished 32000 requests


Server Software:      nginx/1.27.3
Server Hostname:      172.20.32.1
Server Port:          80

Document Path:        /ganglia
Document Length:       169 bytes

Concurrency Level:     240
Time taken for tests:  30.132 seconds
Complete requests:     32000
Failed requests:        0
Non-2xx responses:     32000
Total transferred:     11712000 bytes
HTML transferred:      5408000 bytes
Requests per second:   1062.01 [#/sec] (mean)
Time per request:      225.987 [ms] (mean)
Time per request:      0.942 [ms] (mean, across all concurrent requests)
Transfer rate:          379.58 [Kbytes/sec] received

Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        0   1.2      0     28
Processing:     13     225  38.1    218   367
Waiting:         4     225  38.1    218   367
Total:          13     225  37.8    218   368

Percentage of the requests served within a certain time (ms)
 50%    218
 66%    234
 75%    245
 80%    253
 90%    277
 95%    296
 98%    319
 99%    341
100%    368 (longest request)
```

## Avaliação do desempenho

### Com balanceamento de carga (HAProxy):

- **Pedidos por segundo:** Entre 995 e 1207 pedidos.
- **Tempo total dos pedidos:** Entre 26.5 e 32.2 segundos.
- **Tempo por pedido:** Entre 0.82 e 1 milissegundos.

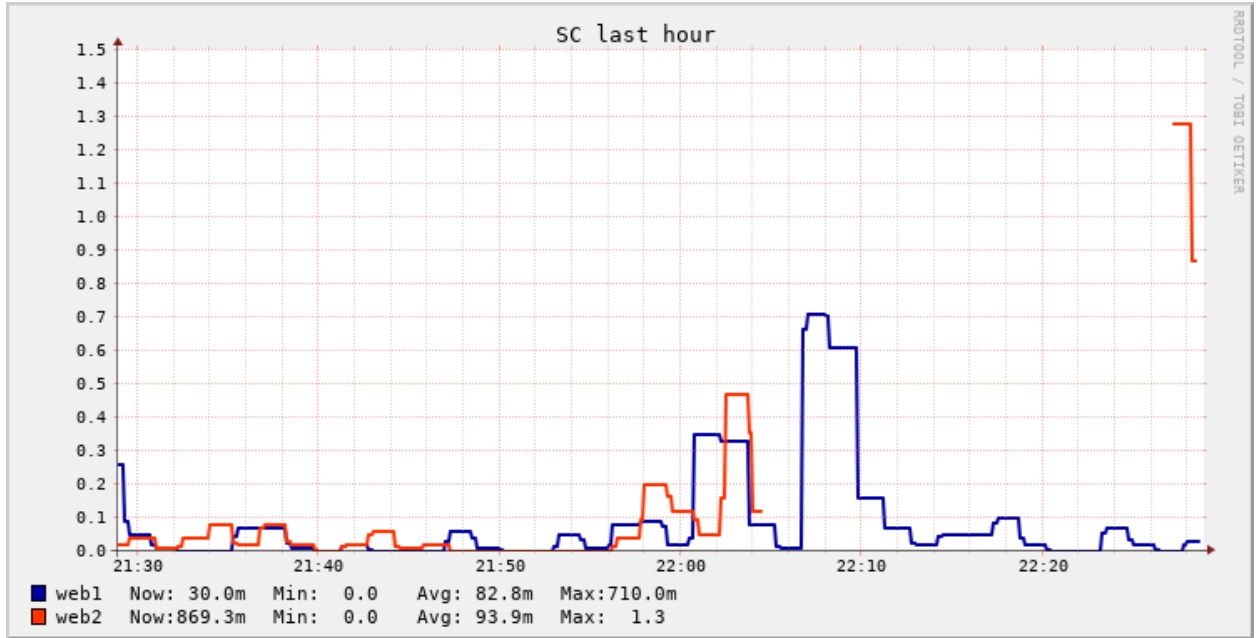
### Sem balanceamento de carga:

- **Pedidos por segundo:** Entre 708 e 1216.
- **Tempo total dos pedidos:** Entre 26.3 e 45.15.
- **Tempo por pedido:** Entre 0.82 e 1.41 milissegundos.

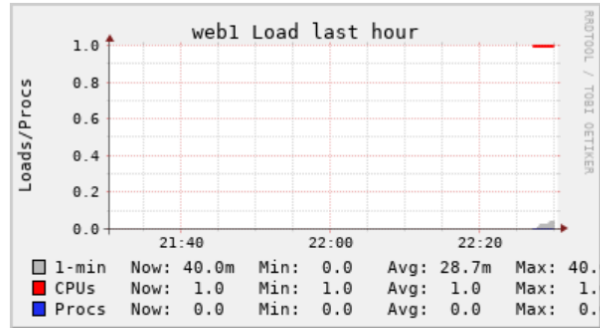
## Análise da diferença de desempenho

Os testes de desempenho foram realizados para comparar a eficiência da infraestrutura com e sem o balanceamento de carga utilizando o *HAProxy*. Os resultados demonstram o impacto positivo da distribuição de tráfego na estabilidade e no tempo de resposta do sistema. Com o balanceamento de carga ativo, a taxa de requisições por segundo variou entre **995 e 1207**, apresentando um comportamento mais consistente. O tempo total necessário para processar todas as requisições ficou entre **26,5 e 32,2 segundos**, e o tempo médio por requisição variou entre **0,82 e 1 milissegundo**. Esses valores indicam que o *HAProxy* conseguiu distribuir eficientemente as requisições entre os servidores, garantindo tempos de resposta estáveis. Sem o balanceamento de carga, a taxa de requisições por segundo teve uma variação maior, oscilando entre **708 e 1216**. O tempo total de processamento das requisições aumentou significativamente em alguns casos, chegando a **45,15 segundos**, o que demonstra um impacto negativo na performance sob carga mais elevada. O tempo médio por requisição também teve uma piora perceptível, variando entre **0,82 e 1,41 milissegundos**, evidenciando uma degradação no desempenho conforme a carga aumentava.

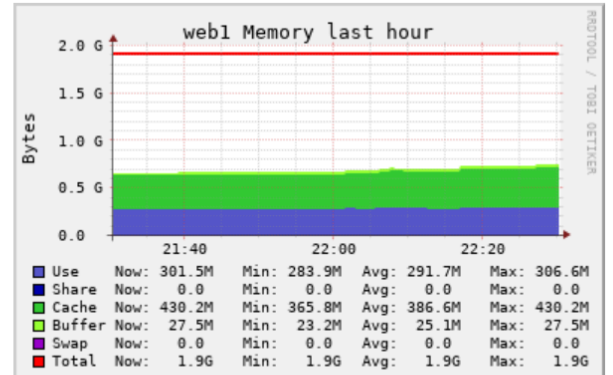
A análise dos resultados indica que o *HAProxy* contribuiu para uma distribuição mais uniforme do tráfego, evitando sobrecarga em um único servidor e reduzindo o tempo total de resposta. Além disso, a menor variação nos tempos médios sugere uma operação mais previsível e eficiente, essencial para a escalabilidade e a manutenção da qualidade do serviço em cenários de alta demanda.



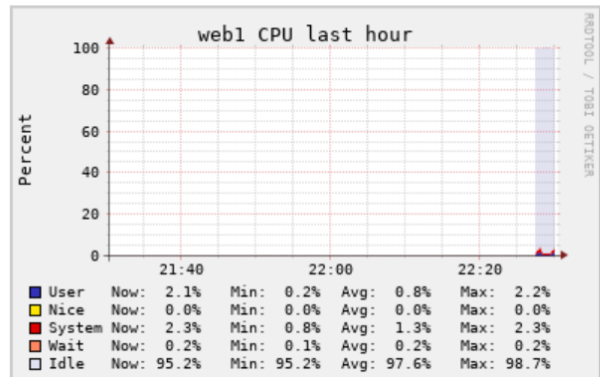
CSV JSON Inspect Hide/Show Events



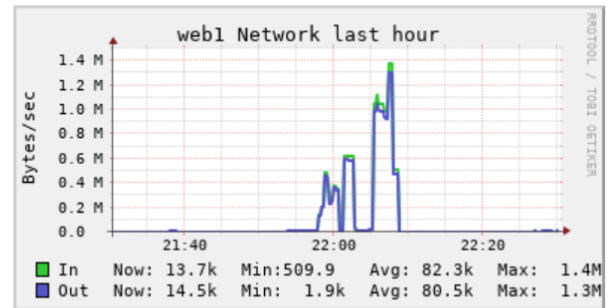
CSV JSON Inspect Hide/Show Events

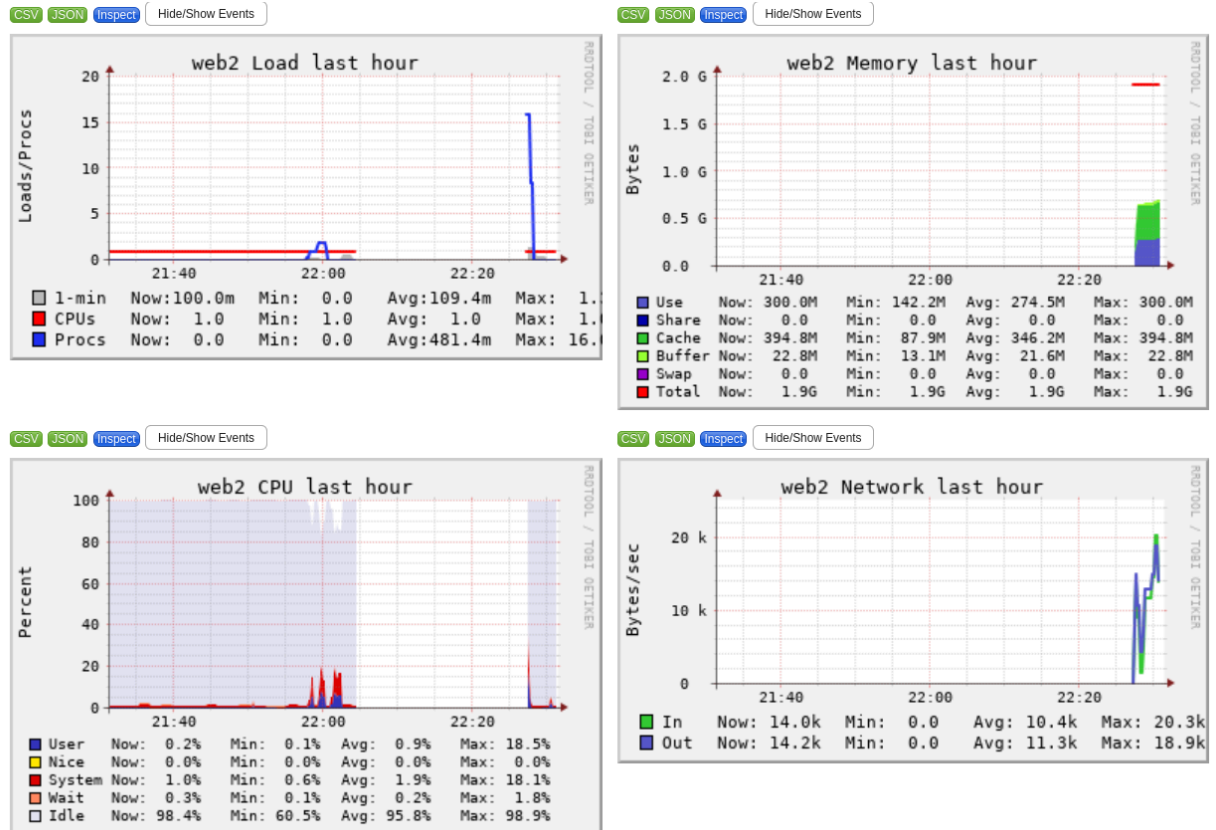


CSV JSON Inspect Hide/Show Events



CSV JSON Inspect Hide/Show Events





## Conclusão

O objetivo deste trabalho foi implementar uma infraestrutura de alta disponibilidade, focando no balanceamento de carga, redundância de servidores e armazenamento compartilhado. A solução utilizou HAProxy e GlusterFS para garantir escalabilidade, resiliência e desempenho.

Todos os principais objetivos foram alcançados: a infraestrutura foi configurada corretamente, os servidores operam com redundância e o balanceamento de carga foi implementado com sucesso. Testes confirmaram a capacidade do sistema de lidar com falhas, garantindo recuperação automática e distribuição eficiente do tráfego. O armazenamento compartilhado via GlusterFS assegurou a consistência e redundância dos dados. A infraestrutura demonstrou estabilidade, garantindo a continuidade dos serviços mesmo diante de falhas simuladas. O sistema está pronto para uso em produção e pode ser aprimorado com melhorias no monitoramento e na integração de testes de carga.