



Instituto Superior de Engenharia de Coimbra
Departamento de Engenharia Informática e Sistemas
Licenciatura em Engenharia Informática

Disponibilidade e Desempenho 2017/2018

**Implementação de Serviços WEB/BD/Armazenamento com
suporte a balanceamento de carga e alta disponibilidade**



?
?
Nuno Fidalgo: 21140369
?

Coimbra, 8 de março 2018

Índice

1. Introdução	1
2. Raspberry Pi	2
3. Cluster	2
4. Serviços Utilizados	3
4.1. Keepalived	3
4.2. HAProxy	4
4.3. PGPool	4
4.4. GlusterFs	6
4.5. Apache	6
4.6. PostgreSQL	7
5. Configurações	8
5.1. Apache	8
5.2. HAProxy	9
5.3. Keepalived	10
5.4. PostgreSQL	11
5.5. Estrutura	12
6. Bibliografia	13
7. Conclusão	15

1. Introdução

Apesar de hoje em dia o processamento que temos disponível, ser muito superior aquele que tínhamos há alguns anos atrás, seja nos nossos computadores pessoais ou em casa, para muitos developers de software criar e manter grandes aplicações era um grande desafio, especialmente aplicações desenvolvidas em C/C++.

Ficamos então com a questão de como podemos melhorar, acelerar os trabalhos pesados que os grandes projetos nos apresentam?

A ideia que apresentamos e pretendemos demonstrar é de que como podemos usar várias workstations ou servers para podermos compilar grandes projetos existentes, tais como o Linux Kernel, ou o LibreOffice?

Existe uma solução para este desafio, a aplicação distcc, que consiste em usar os recursos de processamento de múltiplos computadores existentes numa rede local, na qual consiste em dividir o processo de compilação por vários blocos e partilhar estes mesmo blocos por vários nós disponíveis na rede de forma a acelerar o processo de compilação.

Em complemento à aplicação distcc apresentamos também a aplicação ccache, que tem como principal funcionalidade de fazer um repositório de dados usando os ficheiros object próprios da linguagem C/C++.

Para efeitos de demonstração, pretendamos criar e usar um cluster de quatro Raspberry Pi 2 onde iremos mostrar os passos que usamos para configurar os sistemas de forma a quando se manda compilar uma qualquer aplicação podermos usar o poder de processamento partilhado disponível no cluster.

2. Raspberry Pi

O Raspberry Pi é um computador de tamanho e performance reduzido com a principal tarefa de ser uma plataforma de desenvolvimento. Apesar de neste momento já existirem diversas versões do mesmo, existe um padrão em termos de hardware.

Conecta-se a um monitor de computador ou televisão através de uma porta HDMI, possuindo também varias portas usb e uma porta ethernet. É compatível também com sistemas operativos Linux/Windows.



Figura 1 - Raspberry Pi

3. Cluster

Um cluster consiste em computadores que trabalham em conjunto de modo a serem um único sistema. É também um aglomerado de computadores que vão executar um trabalho dividindo o processamento em esses mesmos computadores.

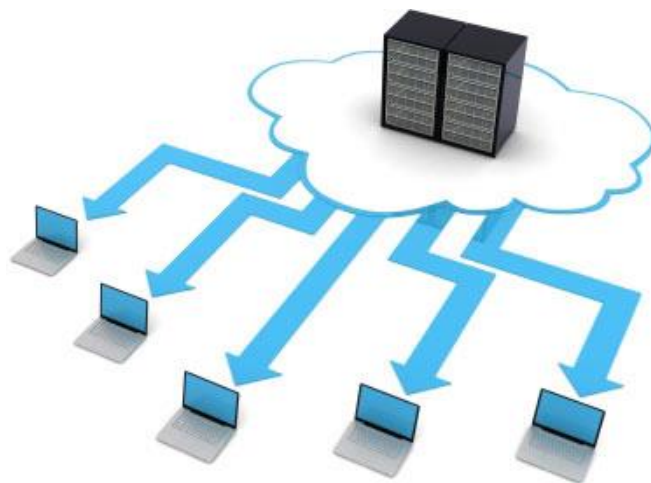


Figura 2 - Cluster

4. Serviços Utilizados

4.1. Keepalived

Keepalived é um software de encaminhamento que funciona com base no protocolo VRRP (Virtual Router Redundancy Protocol) e fornece instalações simples e robustas para balanceamento de carga e alta disponibilidade.

Uma ligação keepalived é uma ligação persistente entre um cliente e um servidor, impedindo assim que a ligação quebre.

Uma ligação HTTP padrão geralmente é desligada quando termina um pedido o que significa que o servidor feche a ligação TCP. De forma a que a ligação esteja sempre ligada para responder a vários pedidos é utilizado o keepalived.

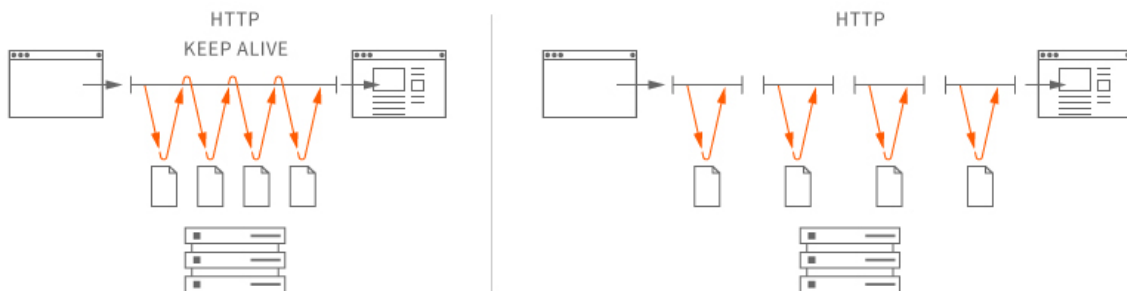


Figura 3 - Keepalived

Tem como vantagens:

- Aumenta a rapidez do website;
- Reduz o uso do CPU;

Desvantagens:

- Aumento do uso da memória: os processos do apache têm que manter as ligações à espera de novos pedidos.

4.2. HAProxy

O serviço HAProxy é um serviço Linux que garante um balanceamento de carga e alta disponibilidade num aglomerado de servidores. Este recebe os pedidos e atua como um proxy, criando um canal entre cliente-servidor. Possui mecanismos para escolher o servidor que responde ao pedido como o:

- Round-Robin: o servidor é escolhido de forma circular, independentemente da carga dos servidores;
- Leastconn: seleciona o servidor com o menor número de ligações. É recomendado para sessões maiores;
- Source: este modo seleciona qual o servidor que vai ser usado baseado num hash do IP de origem. Garante que um utilizador conectar-se-á sempre ao mesmo servidor.

4.3. PGPool

O Pgbpool-II funciona entre servidores PostgreSQL e um cliente de base de dados PostgreSQL. Fornecendo as seguintes funcionalidades:

- Pool de Ligação: guarda as ligações aos servidores PostgreSQL e reutiliza sempre que uma nova ligação surja com as mesmas propriedades, faz com reduza a sobrecarga de ligações.
- Replicação: pode gerir vários servidores PostgreSQL. Utiliza a função de replicação que cria um backup em tempo real em discos físicos.
- Balanceamento de carga: se uma base de dados for replicada uma consulta ao servidor irá retornar o mesmo resultado. Aproveita o recurso de replicação para reduzir a carga em cada servidor.

- Excesso de ligações: existe um limite no número máximo de ligações em simultâneo com o servidor PostgreSQL e as ligações são rejeitadas após o limite ser ultrapassado.

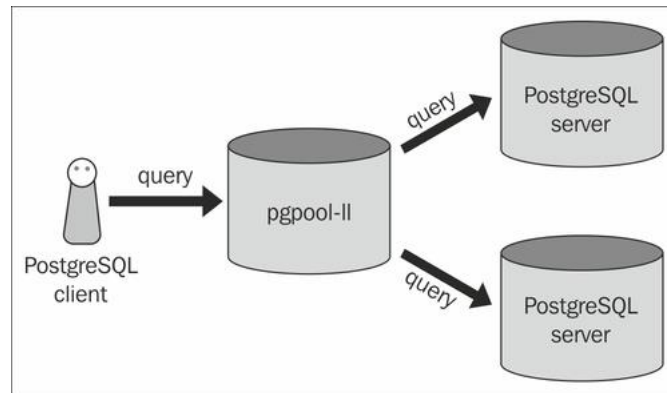


Figura 4 - PGPool

4.4. GlusterFs

É um sistema de armazenamento distribuído que agrega múltiplas unidades de armazenamento remotas num único volume. As unidades bricks são distribuídas pela rede num único sistema de ficheiros paralelo, permitindo escalabilidade de milhares de bricks.

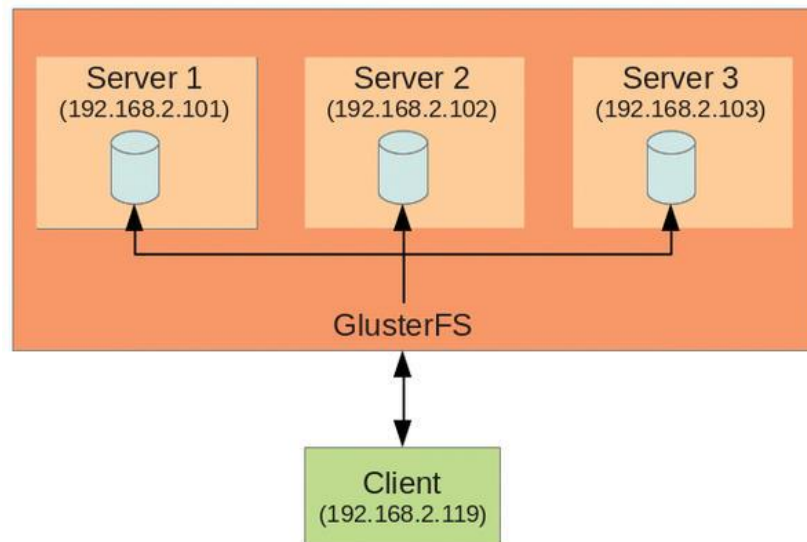


Figura 5 - Exemplo Arquitetura GlusterFS

4.5. Apache

O Apache é uma aplicação que se pode instalar no Raspberry Pi que permite servir paginas web. Por si só, o Apache pode servir paginas HTML por HTTP e com módulos adicionais podendo também servir paginas web dinâmicas que usem linguagens de scripting como o PHP.

Isto permite construir um servidor web Apache numa intranet local, sem necessidade de sistemas complexos, com custos reduzidos de instalação e manutenção muito simples. Torna-se um ambiente ideal para desenvolvimento web, em que simplesmente se tem de arrastar os ficheiros HTML para a diretoria do Apache.

Podemos ainda permitir FTP e SSH. No nosso caso para facilidade de acesso e utilização, o SSH já esta a ser usado.

4.6. PostgreSQL

O PostgreSQL é um sistema de base de dados open source muito poderoso. Possui mais de 15 anos de desenvolvimento e com uma arquitetura comprovada que lhe vale a atual reputação de confiabilidade, integridade de dados e correção.

O PostgreSQL foi desenhado para correr em ambientes Linux.



Figura 6 - PostgreSQL

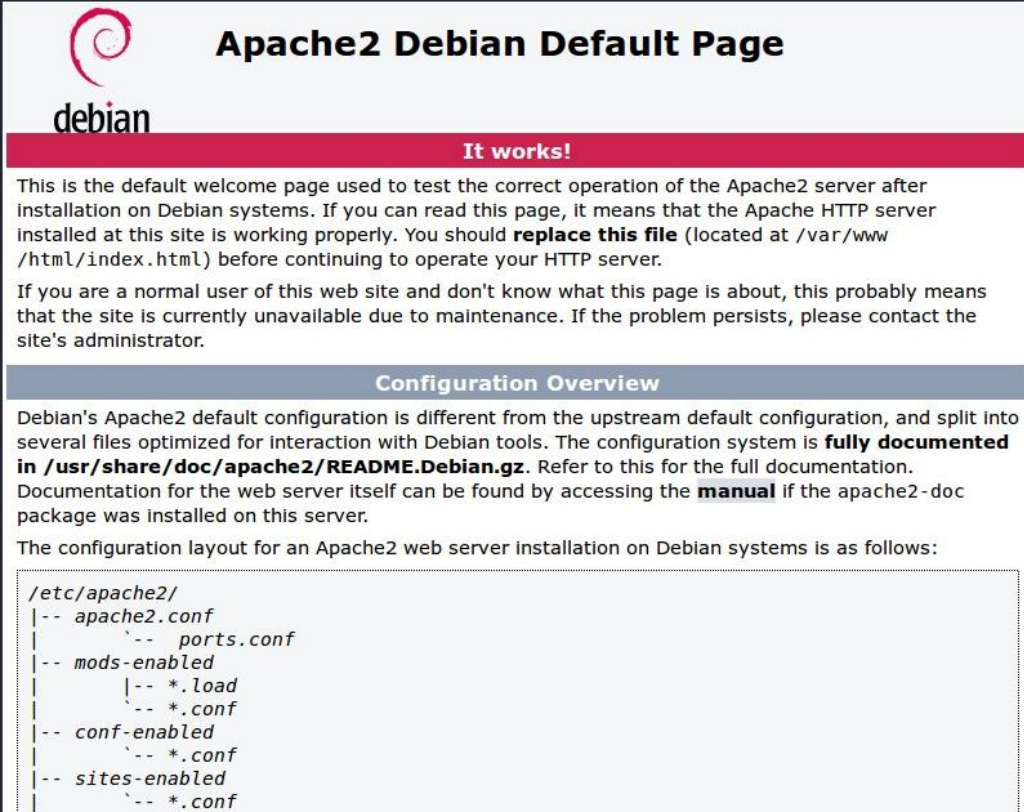
5. Configurações

5.1. Apache

Uma vez que o Raspberry corre principalmente em sistemas operativos Linux, temos que o instalar via linha de comandos:

- `sudo apt-get install apache2 -y`
- `sudo apt-get install -y apache2 php php7.0-fpm libapache2-mod-php phpmyadmin`
- `sudo systemctl enable apache2`
- `sudo systemctl restart apache2`
- create file: `sudo vim /var/www/html/info.php<?php phpinfo(); ?>`

Por omissão, o Apache cria um ficheiro HTML de teste na pasta '/var/www/html/index.html', o qual podemos alterar/substituir para usar as nossas próprias paginas. Esta está disponível quando se utiliza o browser para o localhost (`http://localhost/`) ou o endereço do Raspberry Pi usado.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

Figura 7 - Página Default Apache

PHP Version 7.0.27-0+deb9u1	
System	Linux nuster1 4.9.59-v7+ #1047 SMP Sun Oct 29 12:19:23 GMT 2017 armv7l
Build Date	Jan 5 2018 13:51:52
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.0/apache2
Loaded Configuration File	/etc/php/7.0/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.0/apache2/conf.d
Additional .ini files parsed	/etc/php/7.0/apache2/conf.d/10-opcache.ini, /etc/php/7.0/apache2/conf.d/10-pdo.ini, /etc/php/7.0/apache2/conf.d/20-calendar.ini, /etc/php/7.0/apache2/conf.d/20-ctype.ini, /etc/php/7.0/apache2/conf.d/20-exif.ini, /etc/php/7.0/apache2/conf.d/20-fileinfo.ini, /etc/php/7.0/apache2/conf.d/20-ftp.ini, /etc/php/7.0/apache2/conf.d/20-gettext.ini, /etc/php/7.0/apache2/conf.d/20-iconv.ini, /etc/php/7.0/apache2/conf.d/20-json.ini, /etc/php/7.0/apache2/conf.d/20-pdo_pgsql.ini, /etc/php/7.0/apache2/conf.d/20-pgsql.ini, /etc/php/7.0/apache2/conf.d/20-phar.ini, /etc/php/7.0/apache2/conf.d/20-posix.ini, /etc/php/7.0/apache2/conf.d/20-readline.ini, /etc/php/7.0/apache2/conf.d/20-shmop.ini, /etc/php/7.0/apache2/conf.d/20-sockets.ini, /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.0/apache2/conf.d/20-sysvsem.ini, /etc/php/7.0/apache2/conf.d/20-sysvshm.ini, /etc/php/7.0/apache2/conf.d/20-tokenizer.ini
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012,NTS
PHP Extension Build	API20151012,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled

Figura 8 - Resultado da aplicação do Apache

5.2. HAProxy

Uma vez que temos apenas um balanceador de carga o tornaria num Single-Point-of-Failure, optamos por ter dois. Para isto utilizamos uma configuração Ativo/Passivo (Master/Slave) nos HAProxy com keepalived. Tem como funcionalidades:

- Balanceamento de carga e alta disponibilidade num aglomerado de servidores;
- Recebe os pedidos e atua como um proxy, criando um canal entre cliente-servidor.



Figura 9 - Resultado do Balanceamento de Carga

A imagem abaixo mostra-nos as estatísticas dos nossos servidores. É usado um código de cores. Neste caso temos quatro servidores a verde, o que significa que está tudo no estado de ativo.

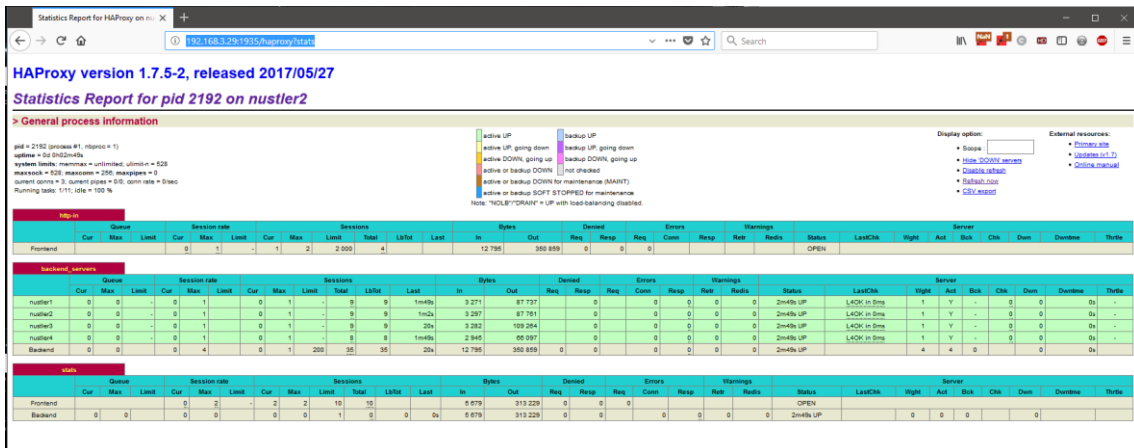


Figura 10 - Estatísticas do HAProxy

5.3. Keepalived

O keepalived é um software de encaminhamento usado para monitorização de serviços ou sistemas e que fornece failover automático para um standby caso haja algum problema. Isto traz-nos uma disponibilidade elevada, retirando o SPOF que existia quando apenas tínhamos um HAProxy a funcionar.

Para pôr em prática estas funcionalidades, o keepalived utiliza VRRP (Virtual Router Redundancy Protocol). Este protocolo é usado de forma comum, pois além de trazer alta disponibilidade para uma rede ou sub-rede, oferece também balanceamento de carga, apesar de neste trabalho termos usado apenas para redundância/ disponibilidade (Balanceamento de carga - HAProxy).

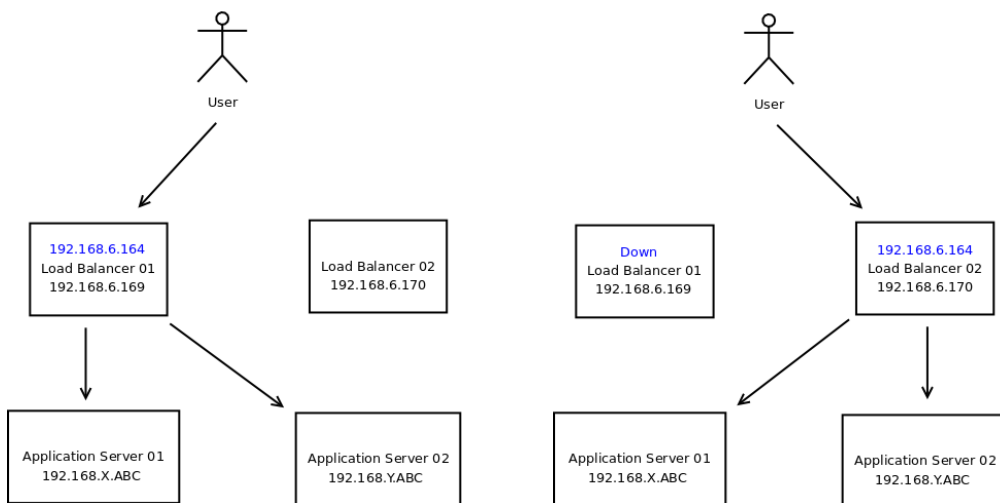


Figura 11 - Direcção do tráfego numa situação normal (esquerda), e em caso de falha do LB1 (direita)

5.4. PostgreSQL

O PostgreSQL foi instalado nos quatro raspberry pi como base de dados onde foi posteriormente instalado o Pgpool-II, para depois ser possível fazer replicação, que será neste caso a Hot Standby.

Este tipo de replicação, resume-se a copiar a transacções de logs da base de dados Master para a Slave onde esta tem apenas o papel de read only.

Este log de transacção é gerado pelo PostgreSQL e consiste em todos os snapshots de binários com todas as modificações feitas na base de dados.

Configuração:

```
PostgreSQL with pgPool & Replication

# install on nustler 2/3/4
apt-cache search postgresql
apt-cache show postgresql: 9.6+181+deb9u1
sudo apt-get install postgresql php-pgsql postgresql-9.6-pgpool2 -y *
option postgis2
sudo systemctl status postgresql

# set postgres password
sudo -u postgres psqlm\password postgres \q

# add new user
sudo su postgres
createuser pi -P -interactive password set: dd@isec

# replication
sudo -u postgres psql
CREATE ROLE replication WITH REPLICATION PASSWORD 'reppassword' LOGIN;
\q
sudo vim /var/lib/postgresql/.pgpass
*:*:*:replication:reppassword
sudo chown postgres:postgres /var/lib/postgresql/.pgpass
sudo chmod 0600 /var/lib/postgresql/.pgpass
sudo vim /etc/postgresql/9.6/main/postgresql.conf
listen_addresses = '*'
port = 5432
sudo vim /etc/postgresql/9.6/main/pg_hba.conf
host    all             all             all             trust
host    replication    replication    all             trust
```

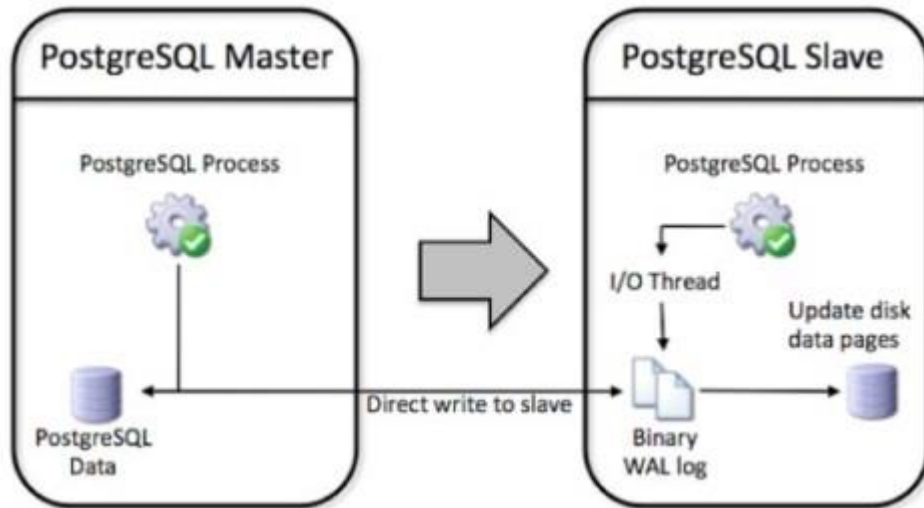
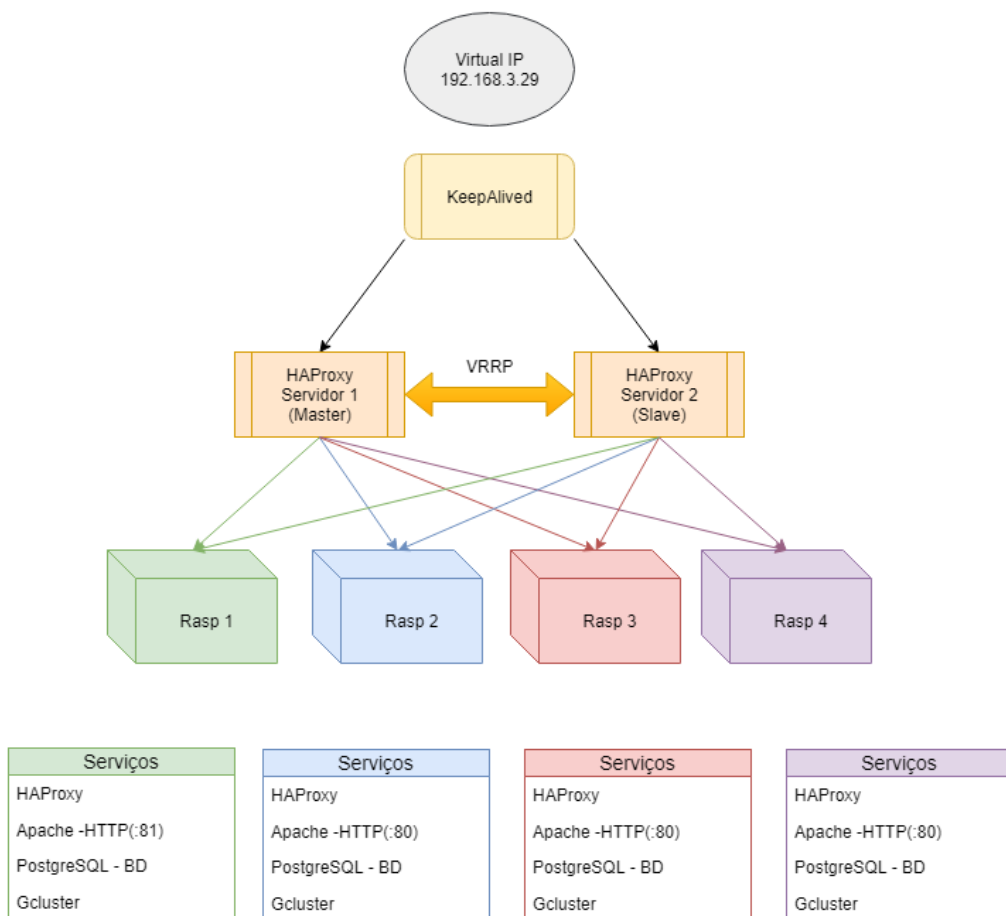


Figura 12 – Processo - PostgreSQL

5.5. Estrutura

VRRP



6. Bibliografia

<https://pt.stackoverflow.com/questions/78317/o-que-%C3%A9-connection-keep-alive>

<https://pt.wikipedia.org/wiki/Cluster>

https://pt.wikipedia.org/wiki/Raspberry_Pi

<https://pplware.sapo.pt/tutoriais/tutorial-balanceamento-de-carga-em-servidores-com-haproxy/>

<https://phcco.com/alta-disponibilidade-e-balanceamento-de-carga-http-com-haproxy>

http://www.pgpool.net/mediawiki/index.php/Main_Page

https://www.server-world.info/en/note?os=CentOS_7&p=haproxy&f=1

<https://tecadmin.net/how-to-configure-haproxy-statics/>

<https://linode.com/docs/databases/postgresql/create-a-highly-available-postgresql-cluster-using-patroni-and-haproxy/>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-highly-available-haproxy-servers-with-keepalived-and-floating-ips-on-ubuntu-14-04>

<https://www.linux.com/news/redundant-routers-linux-and-keepalived>

http://www.pgpool.net/pgpool-web/contrib_docs/watchdog_master_slave_3.3/en.html

<http://pjkh.com/articles/postgresql-replication-and-load-balancing-with-pgpool2/>

<https://www.keyup.eu/en/blog/89-replication-and-load-balancing-with-postgresql-and-pgpool2>

http://www.pgpool.net/pgpool-web/contrib_docs/watchdog_master_slave/en.html#checking_vip

<https://www.fatdragon.me/blog/2016/05/postgresql-ha-pgpool-ii-part-6>

<http://banoffeepiserver.com/glusterfs/set-up-glusterfs-on-two-nodes.html>

<http://banoffeepiserver.com/banana-pi-server-cluster/>

<https://nickhowell.co.uk/2016/07/23/raspberry-pi-nas-with-gluster/>

https://www.server-world.info/en/note?os=CentOS_7&p=glusterfs&f=6

<http://docs.gluster.org/en/latest/>

<https://howtoraspberrypi.com/how-to-install-web-server-raspberry-pi-lamp/>

<https://www.raspberrypi.org/magpi/apache-web-server/>

<https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>

<https://www.howtoforge.com/setting-up-a-high-availability-load-balancer-with-haproxy-keepalived-on-debian-lenny-p2>

<http://dasunhegoda.com/how-to-setup-haproxy-with-keepalived/833/>

<http://keepalived.org/>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-master-slave-replication-on-postgresql-on-an-ubuntu-12-04-vps>

7. Conclusão

Realizamos este trabalho prático da unidade curricular de Disponibilidade e Desempenho, no que respeita à Implementação de Serviços WEB/BD/Armazenamento com suporte a balanceamento de carga e alta disponibilidade.

Abordamos diversos protocolos e diversas estruturas o que nos permitiu além do conteúdo lecionado na unidade curricular de Disponibilidade e Desempenho, uma maior aprendizagem ao nível do balanceamento de carga, mas também ao nível da disponibilidade dos diversos serviços utilizados.

Consideramos que cumprimos os requisitos que eram propostos para este trabalho prático.