



**UNICAMP**

Universidade Estadual de Campinas

## Manual

Autores

Gabriel Bueno de Oliveira 139455  
Joao Guilherme Daros Fidelis 136242  
Lucas Henrique Morais 136640  
Matheus Yokoyama Figueiredo 137036  
Pedro Rodrigues Grijo 139715

# 1 Setup do Cluster

## 1.1 Acesso

O acesso ao cluster pode ser feito de fora e de dentro do IC (Instituto de Computação - UNICAMP).

### 1.1.1 Interno ao IC

Para acesso interno ao ic, basta acessar por ssh o endereço `cbn6.lab.ic.unicamp.br`. No linux, isso pode ser feito através de:

```
$ ssh cbn6.lab.ic.unicamp.br
```

Após a conexão ser feita, insira sua senha.

### 1.1.2 Externo ao IC

Para o acesso externo, é necessário fazer uma conexão de duas fases ou usar um tunnel por uma máquina do ic. Para acesso remoto a uma máquina do ic, no linux, utilize **um** dentre os seguintes comandos:

```
$ ssh -l raXXXXXX ssh.students.ic.unicamp.br
$ ssh -l raXXXXXX ssh2.students.ic.unicamp.br
```

Após a conexão ser feita, insira sua senha. Após feita a credencial, prossiga do mesmo modo como se tivesse acessando de dentro do IC.

# 2 Instalação e Configuração

## 2.1 PostgreSQL

### 2.1.1 Instalação

A versão utilizada do PostgreSQL é a versão 9.5, que pode ser obtida seguindo os seguintes comandos:

```
$ sudo su
# cat /etc/apt/sources.list.d/postgresql.list
# deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg
main
# sudo apt-get install wget ca-certificates
# wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc
| sudo apt-key add -
# sudo apt-get update
# sudo apt-get upgrade
# sudo apt-get install postgresql-9.5
```

### 2.1.2 Configuração

Para configurar o acesso ao banco de dados deve-se alterar o arquivo de configuração principal. Nele são definidos toda a configuração de acesso ao banco de dados e os diretórios de dados que são respectivamente:

```
hba_file = '/etc/postgresql/9.5/main/pg_hba.conf'
data_directory = '/var/lib/postgresql/9.5/main'
```

O arquivo pg\_hba.conf define regras para a conexão ao banco. Por padrão, são só permitidas conexões locais: o usuário administrador do banco (postgres, quando instalado por gerenciador de pacotes) tem acesso via socket e sem senha. os demais usuários, via interface local e com senha:

	local	all	postgres	peer	
	host	all	all	127.0.0.1/32	md5

Para criar o primeiro usuário siga os seguintes comandos:

```
$ sudo su
# su postgres
# psql
=# CREATE ROLE <usr_name> with superuser createdb createrole
login;
=# ALTER ROLE <usr_name> PASSWORD '<password>';
```

### 2.1.3 Carregando o Dump

Para carregar o ddump no banco deve-se criar o banco de dados (tpcw) e um usuário tpcw-user.

```
$ pg_restore -d tpcw tpcw-database-dump.tar
```

## 2.2 Tomcat

### 2.2.1 Instalação

Para instalar o Tomcat (usaremos o tomcat7), precisamos instalar o pacote do tomcat e suas dependências. Para isso, no linux, podemos usar o apt-get:

```
$ sudo apt-get install tomcat7
```

A instalação irá criar um usuário tomcat7 dentro de um grupo, tomcat7. Os arquivos estão em \$CATALINA\_BASE, que na instalação do Ubuntu é /var/lib/tomcat7.

Para teste, acesse em uma máquina do IC o cbn6.lab.ic.unicamp.br:8080.

### 2.2.2 Integração com o PostgreSQL

Para integração com o PostgreSQL é necessário fazer o download do .jar do PostgreSQL JDBC para ser adicionado a pasta \$CATALINA\_BASE/lib. Após fazer o download, modifique os seguintes arquivos com o conteúdo indicado:

- \$CATALINA\_BASE/conf/context.xml: [http://www.ggte.unicamp.br/eam/pluginfile.php/188108/mod\\_wiki/attachments/132/tomcat7-context.xml](http://www.ggte.unicamp.br/eam/pluginfile.php/188108/mod_wiki/attachments/132/tomcat7-context.xml)
- \$CATALINA\_BASE/conf/server.xml: [http://www.ggte.unicamp.br/eam/pluginfile.php/188108/mod\\_wiki/attachments/132/tomcat7-server.xml](http://www.ggte.unicamp.br/eam/pluginfile.php/188108/mod_wiki/attachments/132/tomcat7-server.xml). Para este, modifique os campos de username, password e possivelmente url.

Após essas configurações, reinicie o tomcat:

```
# invoke-rc.d tomcat7 restart
```

## 2.3 TPC-W

As aplicações implantadas no Tomcat são configuradas em \$CATALINA\_BASE/webapps. Para ter permissão de criar aplicações nesta pasta, o usuário tem que ser do grupo tomcat7, para fazer isso, no linux:

```
# adduser [usuário_comum] tomcat7
```

Para instalação default do TPC-W no tomcat7, crie a seguinte árvore:

```
$CATALINA_BASE/webapps
    tpcw
        Images
        WEB-INF
            classes
            web.xml
```

Na pasta tpcw, rode os seguintes comandos:

```
$ sudo cp /nfs/servlet/Images.tar.gz .
$ sudo tar -xzf Images.tar.gz
$ sudo cp /nfs/servlet/tpcw-servlets.tar.gz
$ sudo tar -xzf tpcw-servlets.tar.gz
$ cd tpcw-servlets.tar.gz
$ sudo make
```

No fim, copie todas as \*.class geradas para a pasta WEB-INF/classes e copie o web.xml para pasta WEB-INF/. Reinicie o Tomcat.

## 2.4 RBE

O RBE (Remote Browser Emulator) é o programa empregado para gerar carga para o TPC-W. É um aplicativo que emula um conjunto de clientes que acessam o lado servidor do TPC-W, que implementa uma loja de livros. Os clientes operam como se fossem navegadores web. O código fonte do RBE, assim como arquivo README e o script em Python que realiza a análise das saídas estão disponíveis na pasta do RBE.

### 2.4.1 Automação

Para facilitar a coleta dos dados foi implementado um script em bash que executa o rbe com variando os parâmetros e as saídas, já executa o script analyse.py e gera os gráficos. Para executar o script basta executar:

```
$ ./script.sh
```

## 3 Instalando a Replicação

Era necessário instalar o serviço de Streaming Replication do banco primário para o secundário, para que mudanças feitas no banco primário sejam copiadas para o secundário. Para isto, seguimos tutoriais sugeridos pelo professor. Agora, temos dois bancos de dados em duas máquinas diferentes.

### 3.1 Editar configurações no mestre

Na máquina do banco primário, dbmaster2, vamos criar um usuário "replication" no postgres, que é um usuário com privilégios de replicação. O usuário terá a senha 123456. Para isso, execute os comandos:

```
$ sudo su
# su postgres
# psql
# CREATE ROLE replication WITH REPLICATION PASSWORD '123456'
LOGIN
```

Agora, vamos editar o arquivo /etc/postgresql/9.5/grupo06/postgresql.conf da máquina dbmaster2. Na seção **CONNECTIONS AND AUTHENTICATION**, adicione a linha:

```
listen_addresses = '*'
```

Na seção **WRITE AHEAD LOG**, adicione as linhas:

```
wal_level = hot_standby
```

Em **REPLICATION**, coloque as linhas:

```
max_wal_senders = 5
wal_keep_segments = 32
```

Finalmente, em **ARCHIVING**, adicionamos as linhas:

```

        archive_mode = on
        archive_command = 'test ! -f /var/lib/postgresql/9.5/grupo06/archivedir/\%f
&& cp \%p /var/lib/postgresql/9.5/grupo06/archivedir/\%f'

```

Veja que a pasta /var/lib/postgresql/9.5/grupo06/archivedir/ será criada, (ou crie com mkdir /var/lib/postgresql/9.5/grupo06/archivedir/), e nela ficará contido todos os arquivos log gerados pelos bancos de dados. Agora, temos que editar o arquivo /etc/postgresql/9.5/grupo06/pg\_hba.conf. Adicione a linha no final:

```

                                host      replication      replication      10.1.2.20/32
md5

```

Essa linha permite que o banco em standby consiga se conectar com o primário para fazer a replicação.

### 3.2 Fazer base backup

Devemos fazer um base backup do servidor mestre para o escravo. Utilizaremos o comando pg\_basebackup. Para isso, faça login na máquina do escravo, a db-lave2. Primeiro, a pasta de dados do banco de dados que receberá o base backup deve estar vazia. Logo, apagaremos tudo dentro dela. Depois, utilizaremos o comando de base backup para se conectar no banco de dados mestre e fazer a cópia. Rode o comando:

```

rm /var/lib/postgresql/9.5/grupo06/* -r
pg_basebackup -h 10.1.2.10 -p 5434 -D /var/lib/postgresql/9.5/grupo06
-P -U replication --xlog-method=stream -R

```

A flag -R no final, faz com que o arquivo recovery.conf seja gerado automaticamente no processo.

### 3.3 Configuração do slave

Na máquina escrava, edite o arquivo /etc/postgresql/9.5/grupo06/postgresql.conf, adicionando as mesmas linhas que fez na máquina mestre, como nos passos anteriores. É necessário adicionar mais uma linha, na seção **Standby servers**, adicione a linha:

```

hot_standby = on

```

Isso faz com que o banco escravo seja read-only e aceite apenas comandos de leitura.

## 4 Configuração do haproxy

## 5 Automatização da coleta de dados

A segunda parte do experimento foi dividida em 3 partes. Para executar os testes da primeira parte, foram gerados os scripts `arc2_exp1_script.sh` e `arc2_exp_analise.sh` que devem ser executados da seguinte forma:

```
./arc2_exp1_script.sh  
./arc2_exp_analise.sh 1
```

O argumento passado para o 2o script é simplesmente para geração de arquivos com o nome desejado.

Para executar os testes da segunda parte, foi gerado o script `arc2_exp2_script.sh`. Seguem os comandos para executar:

```
./arc2_exp2_script.sh  
./arc2_exp_analise.sh 2
```

Para executar os testes da terceira parte, foi gerado o script `arc2_exp3_script.sh`. Seguem os comandos para executar

```
./arc2_exp3_script.sh  
./arc2_exp_analise.sh 3
```