

Practical Computer Networks and Application

Webserver and Cluster Computing

Summer Term 2020

Prof. Dr. Christian Baun
Henry-Norbert Cocos
Maurizio Petrozziello

`{christianbaun,cocos,petrozziello}@fb2.fra-uas.de`

Computer Science
Faculty of Computer Science and Engineering
Frankfurt University of Applied Sciences

Content

- 1 Introduction
- 2 Webservers
- 3 Cluster Computing
- 4 Configuring the Machines
- 5 References

Introduction

In the last Lab Exercise you did the following things:

- **Learn something about Firewalls**
- **Learn something about Rules, Rule Chains and Policies**
- **Configure a Packet Filter with iptables**

Introduction

In this Lab Exercise we will learn the following things:

- **Set up a Webserver**
- **Set up a Cluster of Webserver**
- **Set up a Load Balancer for the Webserver Cluster**

After this Lab Exercise

You will know how to set up and configure a Cluster of Webservers.
You will know what a Load Balancer does and how they work.

Webservers

Webservers are computers that serve HTML documents to Clients

Common Webserver software on the market are the **Apache Webserver** [?] or **nginx Webserver** [?]

First Webserver was programmed by Tim Berners-Lee in 1989 at CERN (European Organization for Nuclear Research)

Webservers are the central component of the world wide web

Webservers

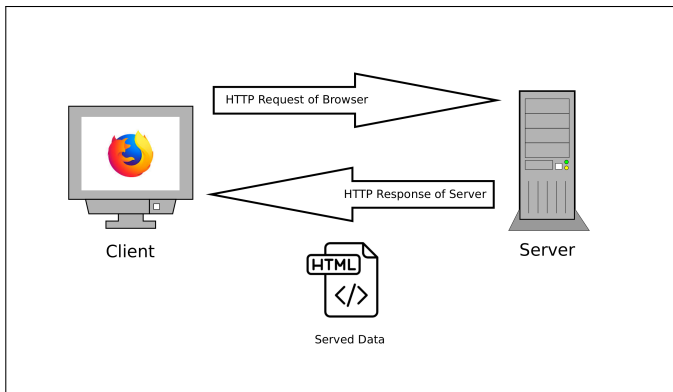


Figure: Browser accessing a Webserver

Load Balancer

Load Balancers are used to distribute requests to multiple Webservers

Common Load Balancer software are **HAProxy** [?], **nginx** [?] or **Apache** [?]

Benefits of using a Load Balancer:

- Increasing reliability of the service
- Increasing availability of the service
- Controlling distribution of requests to Webservers

Load Balancer

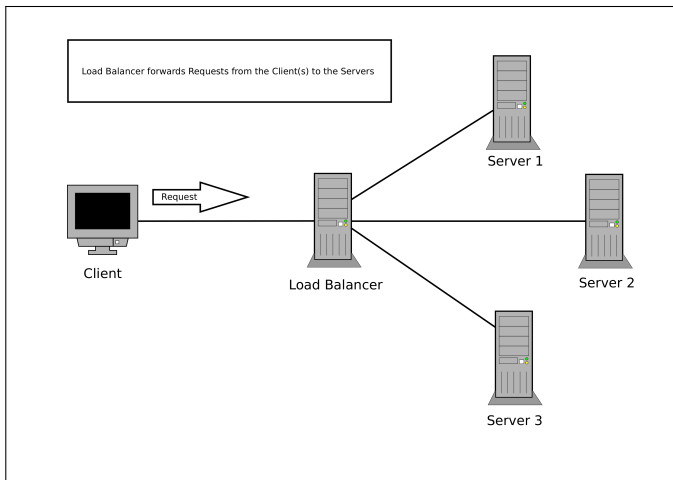


Figure: Load Balancer forwarding Requests

Cluster Computing

Clusters have the following characteristics:

- Computers connected by a network
- Usually consisting of commodity hardware (price efficiency)
- Computers working together on a large scale problem (or many small scale problems)

Categories of Cluster Computing:

- **HA Cluster** – High Availability Cluster
- **HPC Cluster** – High Performance Computing Cluster
- **HTC Cluster** – High Troughput Computing Cluster

HPC and HTC Clusters

HPC Clusters are used for:

- highly complex problems
- highly parallelizable problems
- many fields of application (e.g. weather forecasts, complex calculations in physics, etc.)

HTC Clusters are used for:

- large number of small tasks
- the tasks are just loosely-coupled
- fields of application (e.g. Wikipedia, Web Resources that compute large loads over a long period)

HA Clusters

HA Clusters are used for:

- high available services (e.g. Amazon, Facebook, Twitter)
- small loads, but lot of requests
- reliable service offerings that are not allowed to fail

Benefits of HA Clusters:

- redundancy
- no downtime of service
- failover of hardware if a component fails

Clusters of Webservers

Webserver Clusters are usually HA Clusters. The load on each individual server is low but the number of requests are high. Also a service provider like Amazon does not want its service to be unavailable (or worse fail!), therefore a HA Cluster of Webservers with a Load Balancer is used.

Cluster Computing

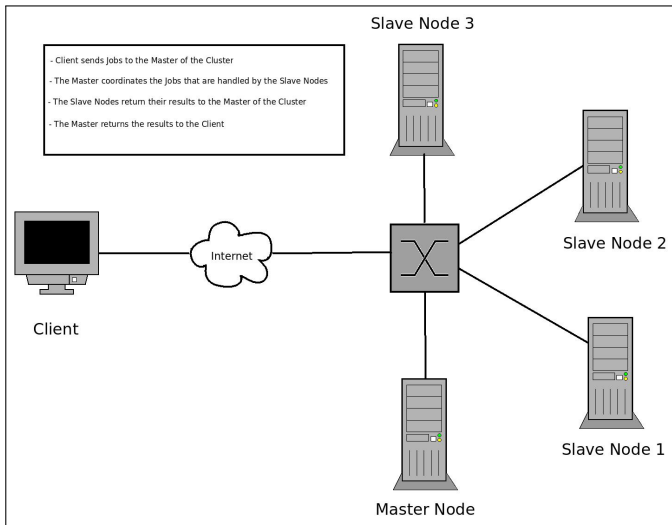


Figure: Cluster Computing

Configuring the Machines (1/6)

In order to build a Webserver Cluster do the following things:

- Install a webserver software on the `mastervm` e.g. Apache [?]
 - `sudo apt-get install apache2`
- Install `dnsmasq` [?] on the `mastervm`
 - `sudo apt-get install dnsmasq`
- Configure the webserver and `dnsmasq`

Configuring the Machines (2/6)

Configuring the `/etc/network/interfaces` file of the `mastervm`:

```
1 auto lo
2 iface lo inet loopback
3
4 # WAN Interface
5 auto enp0s3
6 iface enp0s3 inet dhcp
7
8 # LAN 1
9 auto enp0s8
10 iface enp0s8 inet static
11     address 192.168.1.1
12     netmask 255.255.255.0
13     broadcast 192.168.1.255
```

Listing 1: Interfaces File

Configuring the Machines (3/6)

Configuring the `/etc/apache2/sites-available/000-default.conf` file of the `mastervm`:

```
1 <VirtualHost *:80>
2
3     ServerName master
4     #DocumentRoot /var/www/html
5
6     <Proxy balancer://mycluster/>
7
8     BalancerMember http://master:80
9     BalancerMember http://clone1:80
10    BalancerMember http://clone2:80
11    BalancerMember http://clone3:80
12
13 </Proxy>
14
15 #ProxyPreserveHost On
16
17 ProxyPass / balancer://mycluster/
18 ProxyPassReverse / balancer://mycluster/
19
20 ServerAdmin webmaster@localhost
21 DocumentRoot /var/www/html
22
23 ErrorLog ${APACHE_LOG_DIR}/error.log
24 CustomLog ${APACHE_LOG_DIR}/access.log combined
25
26 </VirtualHost>
```

Listing 2: Configuration of the Load Balancer

Configuring the Machines (4/6)

Configuring `/etc/dnsmasq.conf` of the `mastervm`:

```
# enable DHCP-Server for the following interfaces
interface=enp0s8

# interface enp0s3 has no dhcp because it is the WAN-interface
no-dhcp-interface=enp0s3

# Define IP-Address Ranges for the interfaces
dhcp-range=interface:enp0s8,192.168.1.50,192.168.1.150,infinite

# assign IP-Address according to MAC-Address
dhcp-host=08:00:27:e8:a4:af,clone1,192.168.1.10,infinite

listen-address=127.0.0.1
listen-address=192.168.1.1
```


Configuring the Machines (5/6)

Configuring Port-Forwarding of the mastervm:

```
# Enable NAT-Forwarding for all interfaces
iptables -A FORWARD -o enp0s3 -s 0.0.0.0/0
-m conntrack --ctstate NEW -j ACCEPT
```

```
iptables -A FORWARD -m conntrack
--ctstate ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -t nat -A POSTROUTING
-o enp0s3 -j MASQUERADE
```

```
sysctl -w net.ipv4.ip_forward=1
```

```
# Enable dnsmasq
/etc/init.d/dnsmasq restart
```

Configuring the Machines (6/6)

Define the BalancerMembers clone1 to clone3 which are defined in the /etc/hosts File of the mastervm

Enable Loadbalancing on the mastervm by installing apache2-utils and enable the following modules:

- `sudo a2enmod proxy`
- `sudo a2enmod proxy_http`
- `sudo a2enmod proxy_balancer`
- `sudo a2enmod lbmethod_byrequests`

Then restart the webserver with the following command:

- `sudo systemctl restart apache2.service`

Test the webserver:

- `lynx clone[1-3]`

By enabling promiscuous mode in VirtualBox for the host the webservers can be accessed with:

- `lynx http://<IP-Address of Master>`

Lab Exercise 4

This slide set gives you a brief overview of the tools and technologies discussed in Lab exercise sheet 4.

Hopefully this slide set gives you the ability to solve the tasks of exercise sheet 4!

Lab Exercise 4

Have fun solving the Exercise Sheet and if you have questions, don't be afraid to ask ;-)

Submission Lab Exercise Sheet 4

Please do not forget to submit your results on Moodle until **12th July 2020 !!!**

Conclusion – Lab Exercises

After solving the four Lab Exercise Sheets you have learned the following things:

- **Analyzing Networks and understanding basic Networking technologies**
- **Setting up a Network using Linux Command-Line Tools**
- **Configuring a Firewall in Linux using iptables**
- **Setting up a Cluster of Webservers with a Load Balancer**

Summary

If you have solved all four Lab Exercise Sheets you have a good practical knowledge of Computer Networks, which is a fundamental part of Computer Science. This knowledge can be used to dive deeper into this topic!

Going Further – Linux Professional Institute Certification (LPIC)

Syllabus LPIC-1 (101) [?]:

- System Architecture
- Linux Installation and Package Management
- GNU and Unix Commands
- Devices, Linux Filesystems, Filesystem Hierarchy Standard

Syllabus LPIC-1 (102) [?]:

- Shell Programming & Scripting and Data Management
- User Interfaces and Desktops
- Administrative Tasks
- Essential System Services
- Networking Fundamentals
- Security

References I

- [1] Wiki – apache 2.4. [accessed: May 6, 2020]. [Online]. Available: https://wiki.ubuntuusers.de/Apache_2.4/
- [2] Homepage – nginx. [accessed: May 6, 2020]. [Online]. Available: <https://www.nginx.com/>
- [3] Haproxy – the reliable, high performance tcp/http load balancer. [accessed: May 6, 2020]. [Online]. Available: <https://www.haproxy.org/>
- [4] Apache module mod_proxy_balancer. [accessed: May 6, 2020]. [Online]. Available: https://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html
- [5] Wiki – dnsmasq. [accessed: May 6, 2020]. [Online]. Available: <https://wiki.ubuntuusers.de/Dnsmasq/>
- [6] Linux professional institute. [accessed: May 6, 2020]. [Online]. Available: <http://www.lpi.org/our-certifications/lpic-1-overview>