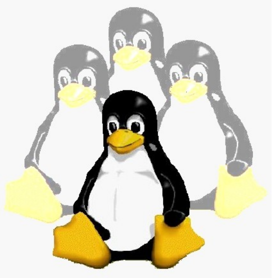
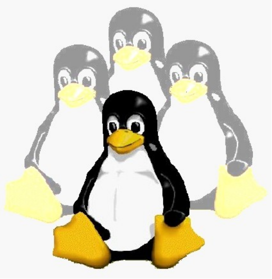


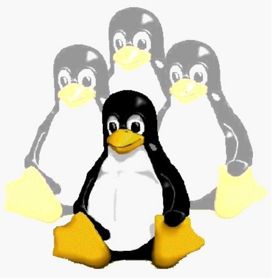
High Availability Clusters in Linux

Sulamita Garcia
EDS Unix Specialist
sulamita@linuxchix.org.br



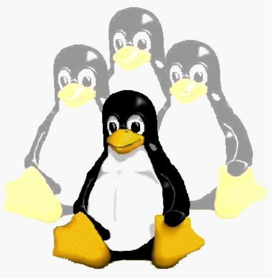
What are clusters

- A set of computers working as one
- High Performance Computing
  - Super computers
- Load Balance
  - Easy way to improve responsiveness: less delay
- High Availability
  - Always responding



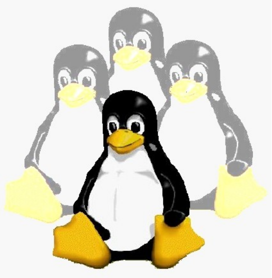
Why High Availability

- We all depend on digital systems
  - From light to banks, any stop is a nightmare
  - Even e-mail communication can cost: clients, time lines, productivity
- It is a plus in services
  - Security also means availability and failure recover



Failures

- There is always a possibility of error
- Failure: physical - electrical or mechanical
- Error: a failure which affects the data, changing a value
- Fault: a failure causes a crash or freezes the system



High Availability

- Systems always online
- Classified by number of 9s
  - 99,99%
  - 99,999% <- majority
  - 99,9999%...
  - Suppliers always try improve this number
- Availability == 1 is hypothetical – there is always a chance of failure

Ways to HA

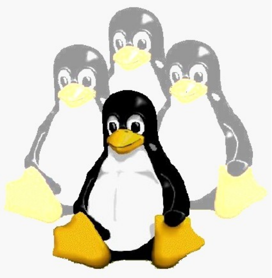
- Fault tolerance software
  - Avoiding failures to become errors and errors become defects
  - Complex and heavy
- Hardware
  - Can perform many tasks
  - Very expansive

High Availability

- Work with fault possibility
- Redundancy by hardware and control by software
- Usual hardware
- Machines recover themselves automatically

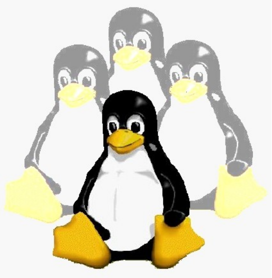
Identifying the environment

- A set of resources to take care of
- Tests to be run frequently
- Actions to run if these tests to fail
- Tools to check and manage the environment



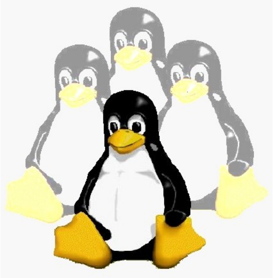
Resources

- A web server
- A link
- Network card state
- A storage unit



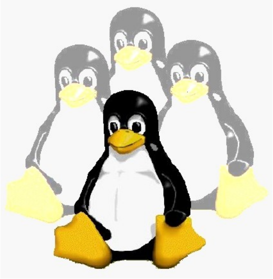
Actions or tests

- Reload the service
- Reboot the machine
- fsck the filesystem
- Configuration of alternative routes
- Notification to admin by pager, mail



High Availability - Contingency

- Raid
- Redundant energy font
- Two Internet link
- Two network cards
- Data Replication
- Configuration replicating
- Replication of user information...



Replication data

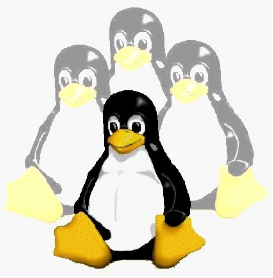
- Depends on data
- Does it change to much?
- Does it have much access?
- Can you loose some data?
- How much load the machine can have?

DRBD

- Data replication block device
  - Block replication: don't understand data
- Replicates partitions, but not files nor directories
- Mirroring: two nodes at time
- Data immediately replicated: highly reliable

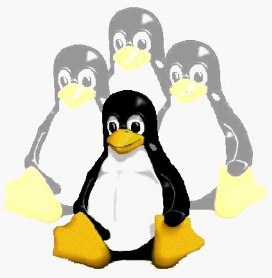
DRBD

- Metadata: 128Mb for mirroring
  - Separate partition: indexing
  - Or shrink your data to grarantee 128Mb to drbd
- Separate link connection
- Any filesystem: ext3, reiserfs, etc
- Load: must be a well designed project
- Great project, few people: if you can contribute



Databases

- Drbd replicates blocks: don't know about registers
- A wrong register can crash entire database
- Most databases already has a way to do it
  - Oracle, LDAP(directory service - slurpd)
- Master -> slaves
  - Share the load



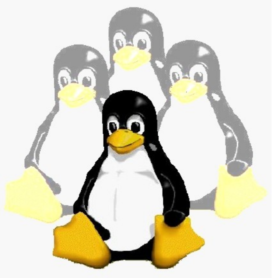
Databases

- Mysql:
  - Master: my.cnf

```
[mysqld]
log-bin
server-id=1
```

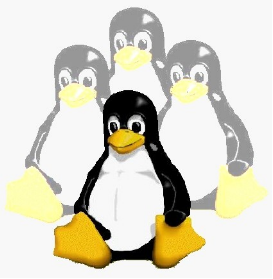
  - Slaves:

```
mysql>CHANGE MASTER TO MASTER_HOST='SERVER',
MASTER_USER='REPLICATION',
   MASTER_PASSWORD='MYPASS',
   MASTER_LOG_FILE='SERVER-BIN.00001',
   MASTER_LOG=211;
mysql> START SLAVE;
```



Another way

- Rsync
  - Replicates files and directories
  - Updates
  - Scheduling
  - Low load
  - Permission: users, machines
  - Can loose some data

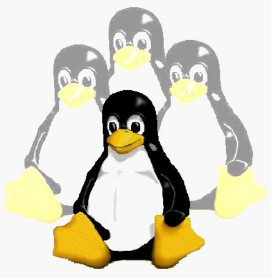


Monitoring

- Heartbeat
  - Check other machine's availability
  - Define primary/secondary
  - When a primary does not answer, the secondary takeovers the services and resources
  - Take care of to small times: machines can fight over services, that's not good

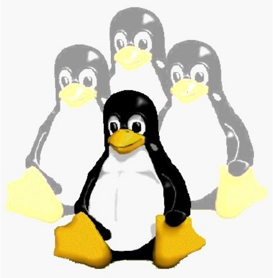
Mon

- Small
- Many monitors and alerts
  - Monitor check a service
  - Alert takes an action: mail, pager, command



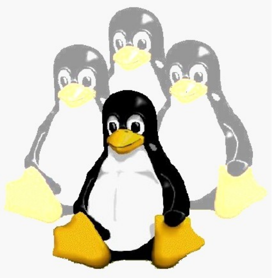
Entire enviroment

- Data replication:
  - drbd/rsync/database schema
- Heartbeat:
  - Primary does not answer, secondary takes the ip service and starts the services -> services available
- Mon:
  - Some services are not responding: mail the admin, stop heartbeat



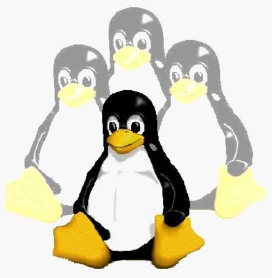
Load balance

- A controller divides the requests among a set of machines
- Share the load
- Easily recover if a box failed



Linux Virtual Server

- Load balance
  - Priority, last used, least used, round robin, or combined
- Controller: can respond or not
- Take care of data: services as http



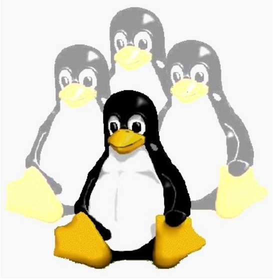
Some simple load balance

- DNS: same address

```
www   IN  A   192.168.0.7
www   IN  A   192.168.0.8
```

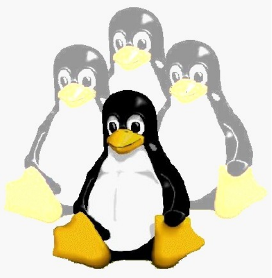
- Iptables: several or a range –to:

```
iptables -t nat -A OUTPUT -p tcp -d xxx -j DNAT --to x1 --to x2
```



Remember

- What you need to replicate
- How exactly should be it
- How much the load does the box support
- The network
- If schedule, how repeatedly will those periods occur



Links and Questions?

- http://www.linux-ha.org/
- http://www.drbd.org/
- http://www.linuxvirtualserver.org/

sulamita@linuxchix.org.br