

# 集群结构：

## 1. **Master** 节点（控制节点或管理节点）：

RHEL-Server-6.2 --64 位 系统，可以上外网。

运行了 InfuDrQueue Server 程序，用来管理所有的 Slave 节点（计算节点）。同时 Master 节点上运行了一个 InfuDrQueueOnRails 程序，它是一个 Web 管理工具，任意用户可以通过浏览器输入 Master 节点 IP 地址访问 InfuDrQueueOnRails 来提交渲染任务，InfuDrQueueOnRails 会把任务请求转发给后端服务 InfuDrQueue Server，然后 InfuDrQueue Server 在把任务分发给一组 InfuDrQueue Slave 节点来完成渲染任务。

## 2. **Slave**（计算节点或渲染节点）：

Windows 2008 Server R2 --64 位系统

运行了 InfuDrQueue Slave 客户端程序和 MentalRay Standard 渲染器。InfuDrQueue 负责接收 InfuDrQueue Server 发过来的任务交给 MentalRay Standard 渲染器，把渲染模型任务渲染成一组图片后存放在共享存储中。

## 3. **共享存储**：

RHEL-Server-5.5 --64 位系统

集群中只有一个 FC 磁盘阵列，用来最终存放数据。在阵列前端用两个 IO 服务器上的 HBA 卡来直接挂载磁盘阵列设备。同时两个 IO 服务器针对设备做 OCFS2 共享文件系统，可以使两个 IO 同时读、写数据和共享数据(互相能看到对方的数据)，起到数据的一致性。在基于两个 IO 服务器的共享文件系统做 Samba 共享存储，集群中 Master 和 Slave 通过千兆网络共同挂载 Samba 共享存储实现读写数据。同时在两个 IO 服务器上针对 Samba 互相做 HA 热备功能，来保障 Samba 服务的不阶段性。

# 自动化安装 InfuDrQueue Master 节点

1. 安装操作系统，系统为 RHEL-Server-6.2 64 位，最小化安装。
2. 安装完操作系统后，配置网络环境(能上网)、主机名为 FQDN(例如 master.local.com)、关闭 Iptables 和 SeLinux、挂载 RHEL-Server-6.2 ISO 文件并配置 YUM 仓库、挂载 Samba 共享存储.如下操作：

**配置网络：**

```
# vim /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
DEVICE="eth0"
```

```
NM_CONTROLLED="yes"
```

```
ONBOOT="yes"
```

```
BOOTPROTO=static
```

```
IPADDR=192.168.8.123
```

```
NETMASK=255.255.255.0
```

```
GATEWAY=192.168.8.1
```

```
DNS1=8.8.8.8
```

```
# /etc/init.d/network restart
```

```
# ifconfig eth0
```

```
eth0    Link encap:Ethernet  HWaddr 08:00:27:75:FF:FA
```

```
        inet addr:192.168.8.123  Bcast:192.168.8.255  Mask:255.255.255.0
```

```
.....
```

```
# ping www.baidu.com
```

```
PING www.a.shifen.com (115.239.210.26) 56(84) bytes of data.
```

```
64 bytes from 115.239.210.26: icmp_seq=1 ttl=51 time=52.5 ms
```

```
64 bytes from 115.239.210.26: icmp_seq=2 ttl=51 time=48.8 ms
```

```
.....
```

测试表明可以正常上网。

**设置主机名 FQDN:**

```
# hostname master.local.com
```

```
# sysctl -w kernel.hostname=master.local.com
```

```
# vim /etc/sysconfig/network
```

```
NETWORKING=yes
```

```
HOSTNAME=master.local.com
```

```
# vim /etc/hosts
```

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
```

```
:::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
```

```
192.168.8.123 master.local.com master
```

```
# ping master.local.com
```

```
PING master.local.com (192.168.8.123) 56(84) bytes of data.
```

```
64 bytes from master.local.com (192.168.8.123): icmp_seq=1 ttl=64 time=0.050 ms
```

```
....
```

可以正常解析主机名.

**关闭 *Iptables* 和 *SeLinux* :**

```
# iptables -F
```

```
# iptables -t nat -F
```

```
# /etc/init.d/iptables save
```

```
# /etc/init.d/iptables stop
```

```
# chkconfig iptables off
```

```
# setenforce 0
```

```
# sed -i 's/SELINUX=.*$/SELINUX=disabled/g' /etc/selinux/config
```

**挂载 *ISO*、配置 *YUM* 仓库 :**

```
# mount -o loop /opt/rhel-server-6.2-x86_64-dvd.iso /mnt
```

```
# rm -fr /etc/yum.repos.d/*
```

```
# vim /etc/yum.repos.d/dvd.repo
```

```
[dvd]
```

```
name=dvd repo
```

```
baseurl=file:///mnt/
```

```
gpgcheck=0
```

```
# echo "/opt/rhel-server-6.2-x86_64-dvd.iso /mnt iso9660 ro,loop,auto 0
```

0" >> /etc/fstab

**挂载 Samba 共享存储：(192.168.8.200 为两个 IO 服务器的虚拟 IP 地址)**

**# yum -y install samba-client**

**# mkdir /etc/drqueue/logs -p**

**# mkdir /etc/drqueue/tmp -p**

**# smbclient //192.168.8.200/logs /etc/drqueue/logs**

**# smbclient //192.168.8.200/tmp /etc/drqueue/tmp**

OK，基础环境设置完毕！剩下的都是脚本自动化安装，要确保网络畅通！

**3.脚本安装，脚本名字为 install\_DrQueue.sh，需要拷贝到系统中，假设拷贝到了 /opt/ 中：**

**# sh /opt/install\_DrQueue.sh**

....

....

安装速度取决与网络的带宽，正常 10 分钟就能安装好了。

安装好后，打开浏览器访问：

http://192.168.8.123

测试用户：demo 密码：demo

管理用户：admin 密码：password

# 手动安装 InfuDrQueue Master

这里使用 RHEL-6.2 的系统，同时安装了 InFu[DrQueueOnRails](#)。

Master 机器需要上外网，本文档使用 192.168.8.123 来做为本机的 IP 地址，系统内需要有 RHEL 6.2 的 ISO 来做本地 yum 仓库。

配置 yum 仓库：

```
# mount -o loop /opt/rhel-server-6.2-x86_64-dvd.iso /mnt/
# echo '/opt/rhel-server-6.2-x86_64-dvd.iso /mnt iso9660 ro,loop,auto 0 0' >> /etc/fstab
# rm -fr /etc/yum.repos.d/*
# vim /etc/yum.repos.d/dvd.repo
[dvd]
```

```
name=dvd repo
```

```
baseurl=file:///mnt/
```

```
gpgcheck=0
```

```
# yum update
```

## 更改主机名

```
hostname master.local.com
sysctl -w kernel.hostname=master.local.com
sed -i 's/HOSTNAME=.*$/HOSTNAME=master.local.com/g' /etc/sysconfig/network
echo '192.168.8.123 master.local.com' >> /etc/hosts
```

## 配置 yum 仓库

本地的 ISO 仓库那是必须的，还需要 epel 和 rpmforge：

```
rpm -viUh http://apt.sw.be/redhat/el6/en/x86_64/rpmforge/RPMS/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
rpm -viUh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-
```

```
8.noarch.rpm
yum update
```

记得关闭 **iptables** 和 **selinux** :

```
iptables -F
```

```
iptables -t nat -F
```

```
/etc/init.d/iptables save
```

```
/etc/init.d/iptables stop
```

```
chkconfig iptables off
```

```
setenforce 0
```

```
sed -i 's/SELINUX=.*$/SELINUX=disabled/g' /etc/selinux/config
```

## 前期准备

安装依赖软件包 :

```
yum install scons bzip2 unzip zip nmap openssl lynx ncftp gcc gcc-c++ httpd
httpd-devel mysql-server pkgconfig php php-gd php-ldap php-xml php-xmlrpc curl
curl-devel perl-libwww-perl libxml2 libxml2-devel mod_python ntp gtk2-devel
gtk2 python-devel swig scons tcsh pygtk2-devel pkgconfig vnc-server fuse fuse-
ntfs-3g gnome-session ImageMagick mod_xsendfile blender
```

安装 ruby on rails :

```
yum -y install libyaml-devel ruby ruby-devel ruby-libs ruby-irb ruby-rdoc ruby-
mysql rubygems
rpm -vhiU http://download.longgeek.com/Puppet_for_rhel6.2/ruby-devel-1.8.7.352-
3.el6.x86_64.rpm
rpm -vhiU http://download.longgeek.com/Puppet_for_rhel6.2/ruby-rdoc-1.8.7.352-
3.el6.x86_64.rpm
rpm -vhiU http://download.longgeek.com/Puppet_for_rhel6.2/rubygems-1.3.7-
1.el6.noarch.rpm
```

添加 gem 淘宝的源 :

```
gem sources -a http://ruby.taobao.org
http://ruby.taobao.org added to sources
gem sources -l
*** CURRENT SOURCES ***
http://rubygems.org/
```

<http://ruby.taobao.org>

## 安装 DrQueue

下载最新的 DrQueue 源码包：

```
wget http://www.drqueue.org/files/drqueue.0.64.4c3.tgz -P /opt
cd /opt/
tar zxvf drqueue.0.64.4c3.tgz
cd DrQueue-0.64.4c3
scons install PREFIX=/opt
```

scons install 时候没有报错的话会正常安装到 /opt/drqueue 目录下。

## 修改配置文件

修改 master.conf:

```
vim /opt/drqueue/etc/master.conf
logs=/opt/drqueue/logs
tmp=/opt/drqueue/tmp
db=/opt/drqueue/db
bin=/opt/drqueue/bin
etc=/opt/drqueue/etc
```

修改 slave.conf:

```
vim /opt/drqueue/etc/slave.conf
logs=/opt/drqueue/logs
tmp=/opt/drqueue/tmp
pool=test1, test2, test3
```

修改 drqman.conf:

```
vim /opt/drqueue/etc/drqman.conf
logs=/opt/drqueue/logs
tmp=/opt/drqueue/tmp
db=/opt/drqueue/db
```

## 设置环境变量

在 /etc/profile 中加入如下变量：

```
PATH=${PATH}:/opt/drqueue/bin
export PATH
export DRQUEUE_ROOT=/opt/drqueue
export DRQUEUE_TMP=/opt/drqueue/tmp
export DRQUEUE_OS=LINUX
export DRQUEUE_BIN=/opt/drqueue/bin
export DRQUEUE_ETC=/opt/drqueue/etc
export DRQUEUE_DB=/opt/drqueue/db
```

```
export DRQUEUE_LOGS=/opt/drqueue/logs
export DRQUEUE_MASTER=192.168.8.123
export RAILS_ENV=production
```

执行 `source /etc/profile` 使之生效。

接下来可以执行 `master.Linux.x86_64`、`slave.Linux.x86_64` 以及 `drqman.Linux.x86_64` 命令，`drqman` 需要调用图形，可以用 `drqman` 客户端来提交渲染任务。

如下：

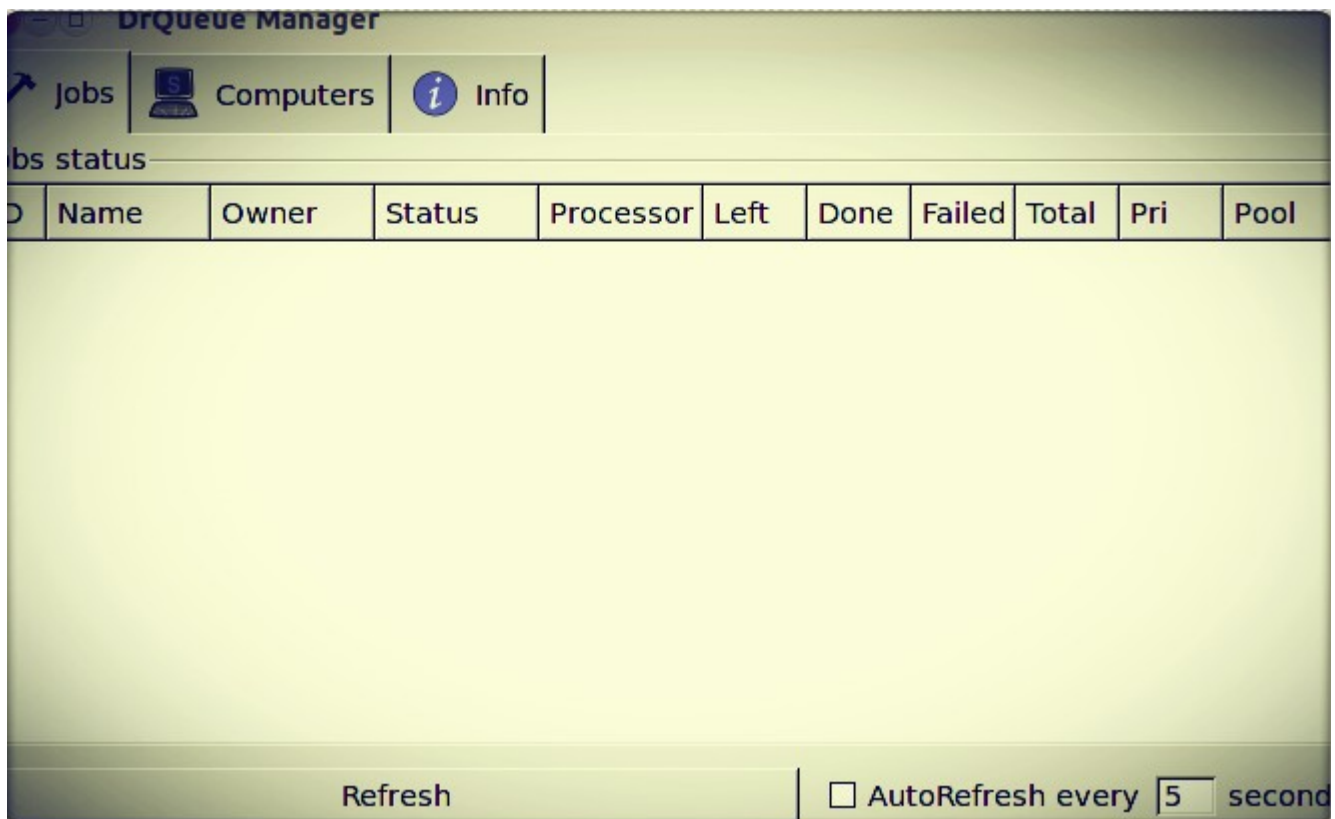
```
master.Linux.x86_64 &
Could not open config file: '/etc/drqueue/master.conf'
Parsing config at: /opt/drqueue/etc/master.conf
Logs on: '/opt/drqueue/logs'
Tmp on: '/opt/drqueue/tmp'
Db on: '/opt/drqueue/db'
Bin on: '/opt/drqueue/bin'
Etc on: '/opt/drqueue/etc'
Waiting for connections...
```

```
slave.Linux.x86_64 &
Could not open config file: '/etc/drqueue/slave.conf'
Parsing config at: /opt/drqueue/etc/slave.conf
Logs on: '/opt/drqueue/logs'
Tmp on: '/opt/drqueue/tmp'
Pools are: 'test1,test2,test3'
HWINFO Report
Name:                master.local.com
Architecture:        Intel
OS:                   Linux
Processor type:       UNKNOWN
64bit-32bit cpu:      64bit
Processor speed:      2679 MHz
Number of processors: 1
Memory:               996 Mbytes
Working silently...
```

使用 `drqman` 需要图形，可以 `drqman` 来提交渲染任务：

```
ssh 192.168.8.123 -X
drqman.Linux.x86_64
```





## DrQueueOnRails 的安装

DrQueueOnRails 是一个 Ruby 写的 web 接口，用来提交渲染任务等功能。

### 依赖环境解决

```
cd /opt/DrQueue-0.64.4c3
ruby DrQueueOnRails_requirements_check.rb #用来检查 DrQueueOnRails 依赖的环境
是否安装
Checking DrQueueOnRails requirements:

RubyGems: installed
DrQueue bindings: not installed
Extra tools for the File class: installed
Net::LDAP library: not installed
RMagick library: not installed
mislav-will_paginate library: not installed
Mongrel web server: not installed (optional)
```

安装没有安装的依赖：

如果直接安装 rails 的话，它会检测 rake 有没有安装，如果没有安装的话会默认安装 10.x.x 以上的版本，而对于 DrQueueOnRails 来说 rake 版本太高了。所以需要先安装 rake，在安装 rails。

```
gem install rake -v 0.8.7 -V
gem install rails -v 2.3.5 -V
```

安装 rmagick 的依赖：

```
yum -y install bzip2-devel libXt-devel libjpeg-devel libtiff-devel
rpm -Uvhi
ftp://ftp.pbone.net/mirror/ftp.scientificlinux.org/linux/scientific/6.2/x86_64/
os/Packages/ghostscript-devel-8.70-11.el6_1.2.x86_64.rpm
rpm -Uvhi http://mirror.centos.org/centos/6/os/x86_64/Packages/lcms-devel-1.19-
1.el6.x86_64.rpm
rpm -Uvhi
ftp://ftp.procmail.org/pub/linux/centos/6.0/os/x86_64/Packages/jasper-devel-
1.900.1-15.el6.x86_64.rpm
rpm -Uvhi
ftp://ftp.mirrorservice.org/sites/ftp.scientificlinux.org/linux/scientific/6.0/
x86_64/os/Packages/ImageMagick-devel-6.5.4.7-5.el6.x86_64.rpm
```

然后安装 rmagick:

```
gem install rmagick -V
```

安装 bindings：

```
yum -y install swig git
gem install DrQueueRubyBindings -V
gem install ruby-net-ldap
```

安装 mislav-will\_paginate 时候，taobao 源里没有，而 rubygems.org 老被墙，只能在添加一个 github 的源：

```
gem sources -a http://gems.github.com
gem install mislav-will_paginate -V
gem install mongrel -V
```

这时候在来检查环境依赖：

```
ruby DrQueueOnRails_requirements_check.rb
```

Checking DrQueueOnRails requirements:

```
RubyGems: installed
DrQueue bindings: installed
Extra tools for the File class: installed
Net::LDAP library: installed
RMagick library: installed
mislav-will_paginate library: installed
Mongrel web server: installed (optional)
```

都 OK 了，接下来就可以安装 DrQueueOnRails 了。

## 安装

```
cd /opt/DrQueue-0.64.4c3/DrQueueOnRails
cd config
cp database.yml.example database.yml
cp environment.rb.example environment.rb
```

修改 database.yml 文件：

```
vim database.yml
production:
  adapter: mysql
  database: DrQueueOnRails
  username: drqueue
  password: drqueue
  host: 192.168.8.123
```

修改 environment.rb 文件中如下选项，其中的 LDAP 选项需要 LDAP 服务器的支持，如果没有配置 LDAP 服务的话可以注释掉，用 demo 用户登录，密码为 demo

```
vim environment.rb
ENV['RAILS_ENV'] ||= 'production'
# DrQueue variables
ENV['DRQUEUE_MASTER'] ||= 'master.local.com'
ENV['DRQUEUE_TMP'] ||= '/opt/drqueue/tmp'
ENV['DRQUEUE_ETC'] ||= '/opt/drqueue/etc'
ENV['DRQUEUE_LOGS'] ||= '/opt/drqueue/logs'

# LDAP variables
ENV['LDAP_TREEBASE'] ||= "dc=local.com"
ENV['LDAP_HOST'] ||= "master.local.com"
ENV['LDAP_PORT'] ||= "389"
ENV['LDAP_FILTER'] ||= "uid"
ENV['LDAP_ATTRS'] ||= "cn"

# protocol for webserver (HTTP, HTTPS)
ENV['WEB_PROTO'] ||= "http"
```

手工用 mysql 命令创建 Dashboard 所需要的数据库并授权：

```
mysql -uroot -p
create database DrQueueOnRails character set utf8;
grant all on DrQueueOnRails.* to 'drqueue'@'master.local.com' identified by
'drqueue';
```

数据库有了，就差表了

```
cd /opt/DrQueue-0.64.4c3/DrQueueOnRails
rake RAILS_ENV=production db:migrate
(in /opt/DrQueue-0.64.4c3/DrQueueOnRails)
fatal: Not a git repository (or any of the parent directories): .git
rake aborted!
end of file reached

(See full trace by running task with --trace)
```

如果出现上面的情况，需要修改 /opt/DrQueue-0.64.4c3/DrQueueOnRails/config/environment.rb 中的第 63 行,改为 APP\_VERSION=""：

```
63 #APP_VERSION = IO.popen("git show --abbrev-commit | head -n
1").readline.split(" ")[1]
64 APP_VERSION = ""
```

在执行 db:migrate , 就可以同步到 DB 中了。

启动内置的 Webrick 服务 :

```
echo "export RAILS_ENV=production" >> /etc/profile  
source /etc/profile  
ruby /opt/DrQueue-0.64.4c3/DrQueueOnRails/script/server
```

浏览器打开 <http://192.168.8.123:3000> 使用演示帐号 : demo 密码 : demo 登录