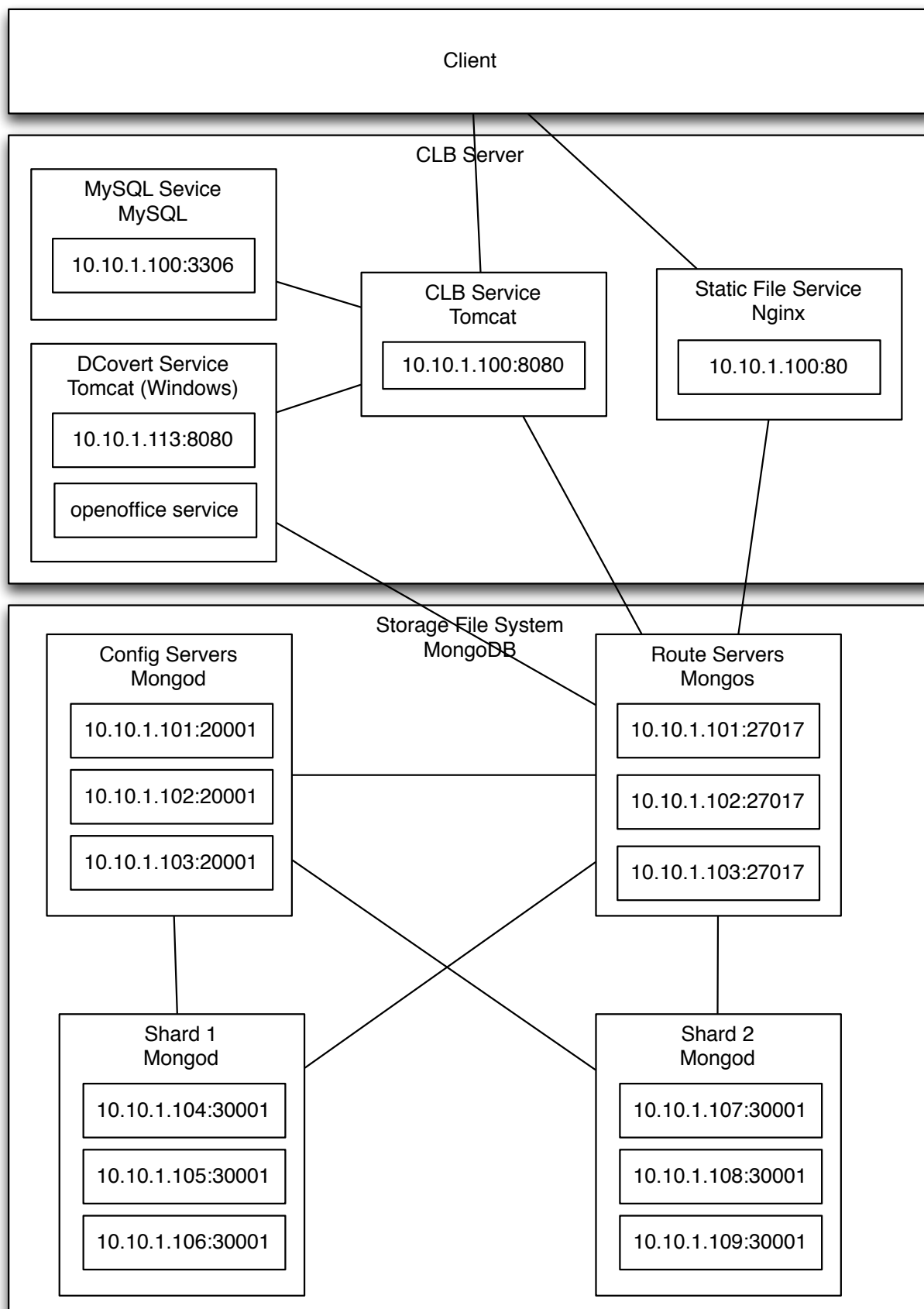


CLB 6.0.1 系统架构与部署说明书

Step 1.准备机器与ip规划



上图里面描述了完整的CLB部署关系图，共需要11台机器，其中CLB+MySQL+Nginx三个服务都部署在ip为10.10.1.100的机器上，负责接受处理用户的上传和下载请求、管理文档数据元信息。这三个服务都是可以扩展的，今后可以视服务压力的具体情况进行扩展。DConvert服务是部署在windows上的文档转换服务，这个服务需要在windows本地安装openoffice的服务。CLB会把文档转换服务转交到DConvert，而为了性能的考虑DConvert会直接从Mongodb中读取和写入文件数据。

剩余的9台机器都是MongoDB服务，因此在每一台机器上都需要安装Mongodb。本次CLB统一使用MongoDB 2.2.2版本，MongoDB的安装目录统一放在/usr/local/mongodb-2.2.2。12台机器中，有3台服务器(101,102,103)放的是MongoDB的配置服务和路由服务。其余9台分为3个组(如104, 105, 106)，在MongoDB术语里把这一组称为Replica Set，中文意思是副本集。在一个副本集中，所有的机器都存储着相同的数据，它们之间的数据同步由MongoDB自行管理。每一个副本集中在任何时刻最多只有一个工作机器，称为Primary机器，其余的叫做Secondary机器，这些机器都是Primary机器的备份节点。一旦Primary机器挂掉了，那么MongoDB会自动的从其余的Secondary机器中选举产生一个新的Primary机器，从而达到高可用性的目标。需要注意的发起选举的条件是必须还有两个及两个以上的Secondary机器存活，当只有一台Secondary机器活着的时候，这台机器永远也无法通过选举成为Primary机器。

在上图中一共有三个副本集，每一个副本集合都存有一部分CLB文件数据。这种关系称为分片，shard。因此在图中shard1、shard2和shard3各存有三分之一的CLB文件数据。每一个shard既可以是一台单独的机器也可以是一个副本集。这些机器的具体安装过程请参考下节。

Step 2.安装基础服务

2.1 CLB Service节点部署

• 部署MySQL

新建clb数据库 如clb6

下载建表语句

<http://ci.cerc.cn:8080/job/clb/ws/target/clb/WEB-INF/conf/clb.sql>

运行建表语句。

• 部署JDK

CLB服务需要使用JDK的版本是1.7.0_08以上的版本，推荐使用jdk1.7.0_11。

去ftp下载jdk1.7.0_11，然后修改JAVA_HOME和CLASSPATH

```
export JAVA_HOME=/usr/lib/jvm/jdk1.7.0_11
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

• 部署Tomcat

下载一个纯Tomcat 7.0.26并解压到/usr/local/tomcat

从ci环境中下载clb的sharedlib，具体命令为：

wget http://ci.cerc.cn:8080/job/ddl/ws/sharedlib/*zip*/sharedlib.zip

解压sharedlib.zip包，jar的存放路径为tomcat/sharedlib/*.jar

修改tomcat/conf下的catalina.properties文件，修改为下面

```
common.loader=${catalina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catalina.home}/lib/*.jar,${catalina.base}/sharedlib,${catalina.base}/sharedlib/*.jar
```

Tomcat默认为8080端口，如果需要修改则更改tomcat/conf/server.xml

- 部署CLB.war

在刚刚部署好的Tomcat的webapps文件夹里面下载clb.war,具体地址为

\$>wget <http://ci.cerc.cn:8080/job/clb/ws/target/clb.war>

启动一次Tomcat将clb.war解压，然后进入clb/WEB-INF/conf目录修改clbconfig.properties

server.name=http\://159.226.11.96:80/clb

clb.db.name = clb3

clb.db.ip = 10.10.1.150:3306

clb.db.user=clive

clb.db.password=clive

#MongoDB url

clb.mongo.url=10.10.1.101

clb.mongo.port=27017

clb.mongo.dbname=docs

#dconvert

clb.dconvert.serverURL=<http://10.10.1.113:8080/dconvert/ServiceServlet>

#nginx

clb.nginx.domain=<http://static.escience.cn>

clb.nginx.context.doc=doc

clb.nginx.context.pdf=pdf

clb.nginx.context.image=image

clb.nginx.context.trivial=trivial

- 部署Nginx(带GridFS模块)

前提需要安装gcc和make，还有zlib-devel

yum -y install gcc automake autoconf libtool make

yum install -y zlib-devel

下载脚本 <http://ftp.cerc.cn/incoming/liji/clb-with-nginx-install.sh>

运行此脚本，该脚本会在当前目录下新建servers和src两个文件夹，并在servers下自动安装nginx1.2.5

2.2 DConvert节点的部署

- 部署Tomcat

- 部署DConvert.war

1) 下载安装OpenOffice，地址：<http://www.openoffice.org/download/>

安装过程直接点下一步，完成后运行

\$>cd C:\Program Files\OpenOffice\OpenOffice.org 3\program

\$>soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;" -nofirststartwizard

2) 下载安装Tomcat，地址：<http://ftp.cerc.cn/incoming/to-pub/%bb%f9%b4%a1%bb%b7%be%b3%c9%fd%bc%b6/>

由于dconvert没有sharelib，如果也不需要修改端口的话，就直接启动就行

3) 部署Dconvert，下载地址为

从SVN中将dconvert检出，地址为：<http://svn.cerc.cn/cn/svn/duckling/trunk/dconvert>
在Eclipse中将其打包成dconvert.war（右键单击项目Export，选择Web->WAR file），然后放到Tomcat的webapps目录下，启动Tomcat即可。

4) 修改dconvert配置文件，tomcat/webapps/dconvert/WEB-INF/conf/dconvertconfig.properties

```
-----
#dconvert server config
server.name=http\://10.10.1.141:8080/dconvert

#MongoDB url
#clb.mongo.url=127.0.0.1
clb.mongo.url=10.10.1.162
#MongoDB port
clb.mongo.port=27017
clb.mongo.db=clb_test
clb.mongo.db.docs=clb_doc
clb.mongo.db.pdf=clb_pdf
duckling.dconvert.localName=dconvert

dconvert.openOfficeHome=C:\Program Files\OpenOffice.org 3
-----
```

2.3 MongoDB集群的部署

1) 安装MongoDB
下载MongoDB-2.2.2到本地，解压到/usr/local/mongodb-2.2.2

2) 文件夹和端口划分

主机名	组名	对应目录	描述
10.10.1.101:27017	mongos	NA	Mongo路由服务器1
10.10.1.102:27017	mongos	NA	Mongo路由服务器2
10.10.1.103:27017	mongos	NA	Mongo路由服务器3
10.10.1.101:20001	config	/mnt/data/clb/config-101	Mongo配置服务器1
10.10.1.102:20001	config	/mnt/data/clb/config-102	Mongo配置服务器2
10.10.1.103:20001	config	/mnt/data/clb/config-103	Mongo配置服务器3
10.10.1.104:30001	shard1	/mnt/data/clb/shard1-104	分片1的第1台机器
10.10.1.105:30001	shard1	/mnt/data/clb/shard1-105	分片1的第2台机器
10.10.1.106:30001	shard1	/mnt/data/clb/shard1-106	分片1的第3台机器
10.10.1.107:30001	shard2	/mnt/data/clb/shard2-107	分片2的第1台机器
10.10.1.108:30001	shard2	/mnt/data/clb/shard2-108	分片2的第2台机器
10.10.1.109:30001	shard2	/mnt/data/clb/shard2-109	分片2的第3台机器

将所有对应的服务器上的目录先建立起来，启动MongoDB的过程放在数据迁移里面完成。

Step 3.数据迁移

1) ddl和dhome挂起维护状态，禁用上传功能；

2) 导出老clb的mysql数据

3) 导出老clb的mongodb文件数据

选104 当shard1的主节点，拷贝shard11的数据到104的文件夹中
\$old-clb/mongodb/data/shard11/* --> /mnt/data/clb/shard1-104/
\$old-clb/mongodb/data/shard11.log --> /mnt/data/clb/logs/
mv shard11.log shard1-104.log

选107当shard2的主节点，拷贝shard21的数据到107的文件夹中
\$old-clb/mongodb/data/shard21/* --> /mnt/data/clb/shard2-107/
\$old-clb/mongodb/data/shard21.log --> /mnt/data/clb/logs/
mv shard21.log shard2-107.log

选101当config的主节点，拷贝config的数据到101的文件夹中
\$old-clb/mongodb/data/config/* --> /mnt/data/clb/config-101/
\$old-clb/mongodb/data/config.log --> /mnt/data/clb/logs/
mv config.log config-101.log

待拷贝完成，在各自的机器上分别执行下列命令

104:(/usr/local/mongodb-2.2.2/bin)
\$>./mongod --port 30001 --dbpath /mnt/data/clb/shard1-104 --fork --rest --shardsvr --replSet shard1 --
logpath /mnt/data/clb/logs/shard1-104.log --logappend --oplogSize 500

107:
\$>./mongod --port 30001 --dbpath /mnt/data/clb/shard2-107 --fork --rest --shardsvr --replSet shard2 --
logpath /mnt/data/clb/logs/shard2-107.log --logappend --oplogSize 500

101:

#启动config服务

\$>./mongod --port 20001 --dbpath /mnt/data/clb/config-101 --fork --configsvr --logpath /mnt/data/clb/
logs/config-101.log --logappend --oplogSize 100

#启动mongos服务

```
$>./mongos --port 27017--configdb 10.10.1.101:20001 --fork --logpath /mnt/data/clb/mongos.log --logappend
```

运行正常的话，都会看到success的提示话语。

4) 更改MongoDB的配置

现在所有MongoDB的服务都已经起来了，但还不能正常工作，必须使用mongo客户端去连接对应的进程修改一些配置才能正常工作。

101:

```
$/mongo --port 20001
```

```
>use config;
```

```
>db.shards.update({"_id":"shard1"},{"$set":{"host":"10.10.1.104:30001"}});
```

```
>db.shards.update({"_id":"shard2"},{"$set":{"host":"10.10.1.107:30001"}});
```

```
>exit
```

104:

```
$/mongo --port 30001
```

```
>use local;
```

```
>db.system.replset.update({"_id":"shard1"},{"$set":{"members":[{"_id": 0, "host" : "10.10.1.104:30001" } ]}});
```

```
>rs.conf();
```

```
>exit
```

107:

```
$/mongo --port 30001
```

```
>use local;
```

```
>db.system.replset.update({"_id":"shard2"},{"$set":{"members":[{"_id": 0, "host" : "10.10.1.104:30001" } ]}});
```

```
>rs.conf();
```

```
>exit
```

做完上述步骤后，再登录101连接mongos进程，端口27017，来检测配置是否生效。

101:

```
$/mongo --port 27017
```

```
>use docs;
```

```
>show collections;
```

如果能正常显示集合列表，则说明已经可以work了。

4) 任意选取一台机器安装clb-migrate.war，这是一台Tomcat webapp。

假定现在在100上把clb-migrate安装起来了

修改clbshowcase.war包中的配置文件，路径为clbshowcase/WEB-INF/conf/clbshowcase.properties

```
-----  
database = aone
```

```
database.ip = localhost:3306
```

```
##### middle clb config #####
```

```
clb.middle.url=http://localhost/clb
```

```
clb.middle.user=aone4cnic
```

```
clb.middle.password=duckling
```

latest clb config

```
clb.latest.mongo.host=127.0.0.1
clb.latest.mongo.port=27017
clb.latest.mongo.dbname=docs
```

5) 访问localhost:8080/clb-migrate/index.html
依次点击第1步，第2步和第3步的按钮，并等待完成。

6) 添加Secondary节点到副本集

```
105:
mkdir /mnt/data/clb/shard1-105
./mongod --port 30001 --dbpath /mnt/data/clb/shard1-105 --fork --rest --shardsvr --replSet shard1 --
logpath /mnt/data/clb/logs/shard1-105.log --logappend --oplogSize 500
```

假定shard1当前主节点为104

```
104:
./mongo --port 30001
>rs.add("10.10.1.105:30001")
>rs.conf()
>exit
```

```
101:
./mongos --port 20001
>use config
>db.shards.update({"_id":"shard2"},{"$set":{"host":"shard1/10.10.1.104:30001,10.10.1.105:30001"}});
>exit
```

把106添加到shard1中的做法和添加105一样。

把107，108，109添加到shard2中，做法与前面添加105的过程一样。