LiteCloud & Skyge HA 8.0

User Guide Manual

BeiJing Skyge Software Company Co,. Ltd

2012-12-27

Catalogue

Brief Introduction

Notic：Before start to install and use LiteCloud software, Please read this manual carefully, Follow the instructions and steps to ensure software can work well.

1、LiteCloud & Skyge HA 8.0 Introduction

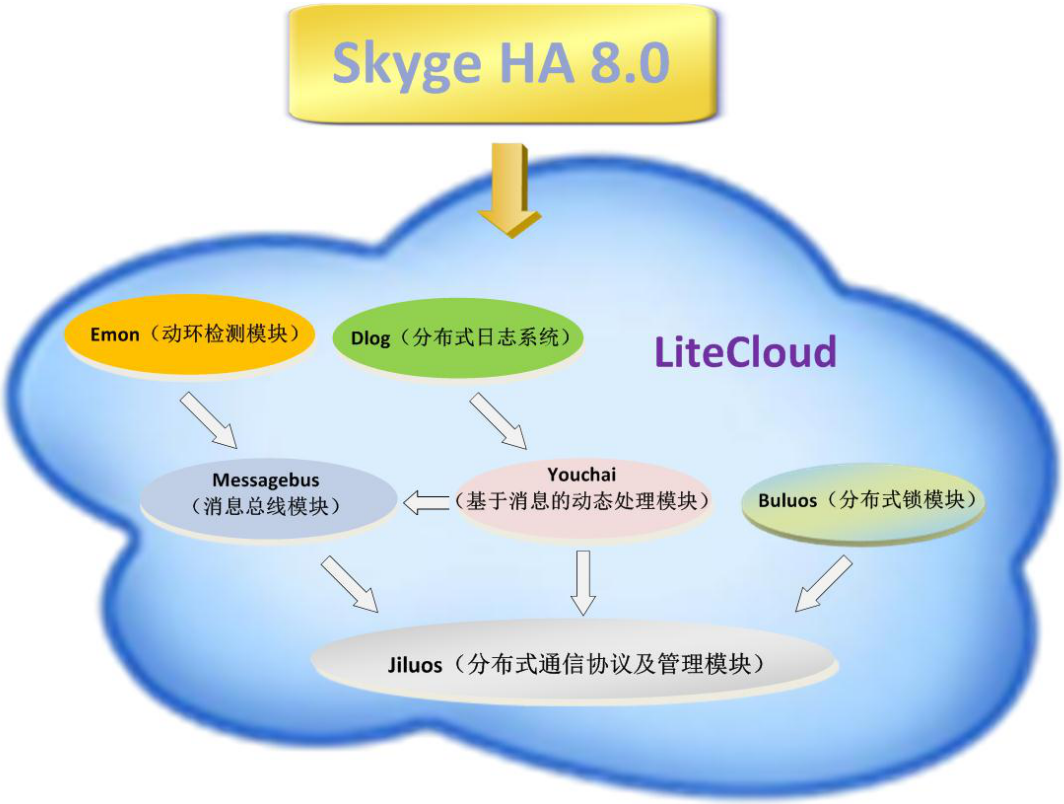
LiteCloud & Skyge HA are developed by Skyge Software Company. LiteCloud is a distributed computing platform which provides lots of interesting features such as cluster communication facility, message bus facility, distributed lock facility and so on. Base on those distributed facilities LiteCloud provides, Skyge HA can provide high availability service for those applications running in LiteCloud. Skyge HA can keep those applications in LiteCloud always running as long as there is still one node alive. Both LiteCloud and Skyge HA run on Linux platform.

LiteCloud distributed platform is a fully distributed computing environment, Which has implemented some fundamental service such as distributed lock system, message bus system, distributed log system and so on. Users don't need to spend any effort to implement those difficult but fundamental facilities. It is very easy to build user's own distributed application. During implementation of LiteCloud, we always keep high available in mind. The whole platform has no explicit master node to avoid Single-Point failure problem and use different ways to ensure high availability. The whole platform provides high performance communication and lock facility which can greatly promote user application's performance. LiteCloud can support up to 64 nodes in one cluster group.

Skyge HA is high availability solution which can integrate applications and their data into one service and provide failover protection for this service.

Skyge HA can protect those critical applications like database, middle wares and important applications from different failure. The services can freely failover among nodes in LiteCloud.

2、LiteCloud & Skyge HA 8.0 Architecture



3、LiteCloud Modules Introduction

LiteCloud is composed by several modules, Which are listed below:

- Jiluos： Base communication module.
- Messagebus ： Provide high performance cluster communication facility.
- Buluos ： Provide distributed lock facility.
- Youchai ： Provide general message handling facility which can be used to implement log system and so on.
- Emon： Provide environment monitoring facility.

Those modules together provide a complete distributed computing platform for end users, Also, Those modules can be used separately because each module is designed as a full function unit.

4、Usage Of Quick Installation Guide

For quick installation, Please refer to those configuration part in different sections. For technical details, Please refer to white paper.

5、LiteCloud & Skyge HA 8.0 Quick Installation Steps

LiteCloud & Skyge HA quick installation has 7 steps.

❶Preparation before installation，refer to 安装前的准备工作

❷Installation of LiteCloud & Skyge HA 8.0 software collections，详见安装 LiteCloud & Skyge HA 8.0

❸Configuration of JILUOS，详见配置 JILUOS

❹Configuration of MESSAGEBUS，详见配置 MESSAGEBUS

❺Configuration of BULUOS，详见配置 BULUOS

❻Configuration of SkygeHA，详见配置 Skyge HA 8.0

❼Configuration of EMON，详见配置 EMON

Section 8 introduce how to uninstall LiteCloud & Skyge HA. There is a FAQ in appendix, which includes some answers to frequent questions.

Preparation Before Installation

Before installation LiteCloud and SkygeHA, need to prepare proper hardware and software environment. There are 6 steps needed.

❶ Understand and analyze your application.
❷ Prepare necessary hardware environments.
❸ Prepare necessary software environments.
❹ Configure hardware environments.
❺ Configure software environments.
❻ Install LiteCloud and SkygeHA。

1.1、Consider From Your Applications

For different applications, different hardware should be considered. Typically, There are two types of applications.

- Pure applications, which doesn't need to store data. For example. LVS under linux, The LVS application only transfer data pass through computer, it doesn't store anything in disks.
- Application with data storage. Most applications belong to this type. For example, Database as Mysql, Oracle. Those applications will create a lot of data and store data in disk files.

For those pure applications, LiteCloud and Skyge HA can support in nature. But for Applications with data storage, things become a little bit complicated. The reason is data should be accessible by another node when failover happens. There are two possible solution for this scenario, one is to use shared storage like disk array, NAS, or IPSAN, The other choice is to use software solution like DRBD or distributed file system. The choice between those two solutions depends on how applications are. IF I/O intensive applications, Disk array, IPSAN are definite good choices, Otherwise, DRBD or distributed file system will be a cost effective solution for

end users.

注意：通常情况下，用户的一个应用对应于 Skyge HA 8.0 的一个服务。如果有多个应用就需要在 Skyge HA 8.0 中配置多个服务，依照每个应用的类型，您可以按照第六章的内容进行相应的配置。

1.2、 Hardware Environment For LiteCloud & Skyge HA 8.0

Required hardware list for LiteCloud & Skyge HA 8.0:

Hardware List			
Hardware type	Number	Usage	Comment
PC SERVER	≥2	Group to be cluster	x86
Network adapter	≥2	2 network adapters are necessary. They will form to be redundant channels to avoid brain-break failure. If there are more than 2 adapters, Use bonding to enhance the availability of network.	The fast network is, the better performance is.
switch	≥1	Typically, one switch is enough, But for high available sake, 2 switches are recommended.	Recommend network adapters link in different switch to avoid single-point failure.
Disk array (optional)	1	Store shared data	Depend on application type

Table 1-1Hardware List

Notice : Generally, two network adapters are necessary in case of brain-break.

1.3、Prepare Software And Hardware Environments

Required software for LiteCloud & Skyge HA 8.0:

Software List	
Software Type	Vendors
Linux	RedHat、Suse、CentOS

表 1-2 软件需求列表

LiteCloud & Skyge HA 8.0 Certified operating system

Certified Operating System	
Red Hat	Red Hat Enterprise Linux Server release 5.6 64 位
	Red Hat Enterprise Linux Server release 6.0 64 位
	Red Hat Enterprise Linux Server release 6.1 64 位
	Red Hat Enterprise Linux Server release 6.2 64 位
	Red Hat Enterprise Linux Server release 6.3 64 位
CentOS	CentOS release 5.6 64 位
	CentOS Linux release 6.0 64 位
	CentOS Linux release 6.1 64 位
	CentOS Linux release 6.2 64 位
	CentOS Linux release 6.3 64 位
SuSE	SUSE Linux Enterprise Server 11.1 64 位
	SUSE Linux Enterprise Server 11.2 64 位

Table 1-3

Notic : When installing operating system, please adopt full installation options to ensure all necessary development lib are installed.

Some packages are needed before installation, For RedHat or CentOS, Please ensure apr-devel, apr-util-devel, libconfig are installed correctly. For SuSE, Please ensure libapr1, libapr1-devel, libapr-util1-devel, libapr-util1 are installed correctly.

1.4、Configure Operating System

Be sure each node has below configuration.

1)、Ensure login as root permission.

2)、Be sure all network adapters are well configured and link with network.

3)、For the reason that Skyge HA uses syslog service on Linux to record all the log, Please be sure syslogd service is enabled and work well.

For example:

```
# /etc/init.d/syslog status
syslogd  (pid 447)  is  runnig ...
klogd  (pid 457)  is  running ...

（或rsyslog服务）
# /etc/init.d/rsyslog status
rsyslogd (pid  1181) is running...
```

4)、Disable SELINUX. For the reason that LiteCloud & SkygeHA need to utilize some basic POSIX semantics, but SELINUX may break this. So, unless users are very skillful in SELINUX to reedit and compile SELINUX rules lib, We recommend to disable it.

If want to disable SELINUX, Please open configure file /etc/selinux/config, and disable SELINUX by setting “SELINUX=disabled”, Save file and reboot computer.

1.5、Prepare User Applications

Install user's applications on each node, Please ensure all the applications are installed correctly and start up well.

If with shared storage device such as disk array, Please be sure those applications on each node can access storage device. Users need to plan how to use those shared storage resources among those applications.

For some applications which have auto start capability, Please disable this function. For example, Some applications may enable the function that lead the operating system to startup themselves when boot up. Skyge HA will take over application's start up, shut down, migration and so on.

1.6、Configure Example

Take a cluster with three nodes as an example. In the following chapters, Installation, configuration, usage will be introduced according to the topology shown in figure 1-1.

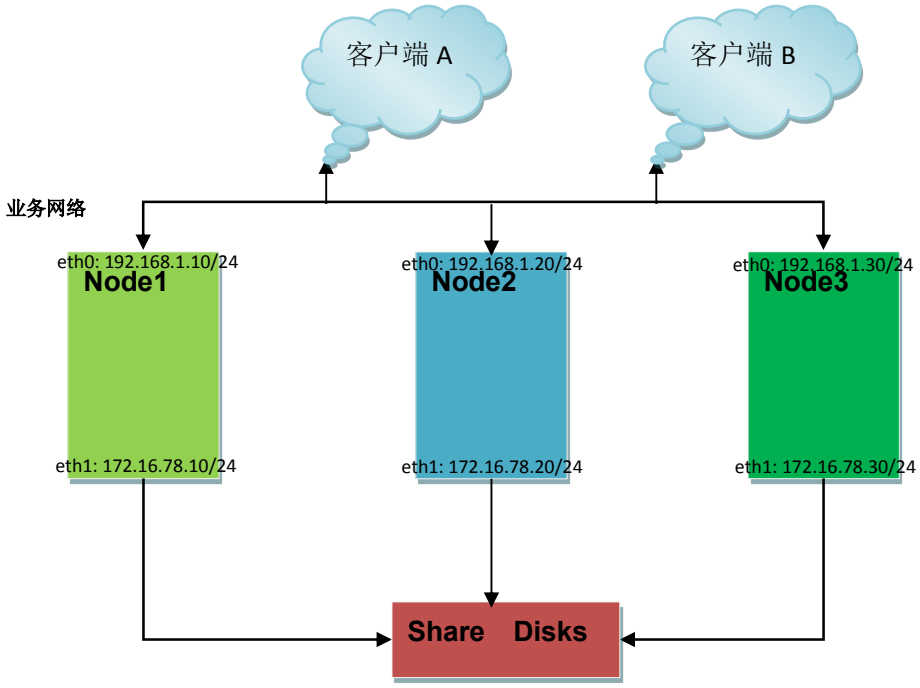


Figure 1-1 Topology Of Three Nodes

As Shown in figure 1-1, There are three nodes which form a cluster, which are indicated as node1, node2 and node3. Each node has two network adapters, eth0 and eth1. Eth0 connects with public network, eth1 connects with internal network. All the three nodes attached a shared disk array.

In the next chapters, all the operation should be applied on each node.

Chapter1. Install LiteCloud & Skyge HA 8.0

Now, start to install software, here to go!

2.1、Install LiteCloud & Skyge HA 8.0

Installation should be applied in root permission.

Insert installation disc into CDROM and mount it. Run install_ha.sh script at the top of dir. Follow the prompt to start installing.

如下图 2-1：

```
#./install_ha.sh
********************************************************************************
                              Skyge Co.,Ltd Copyright

          Lite Cloud & Skyge HA 8.0 is the High Availability
          Software developed by Skyge Co.,Ltd. All right reserved by
          Skyge Co.,Ltd

          Thanks for choosing Skyge cloud software, Any suggestions or
          comments, Please refer to www.skyge.com for help
          or contact our professional service for product support
********************************************************************************
Continue to install the Lite Cloud & Skyge HA 8.0? Please input your choice [y/n]? [y]: y
Detecting your system configuration ......
Your operation system is "RedHat", and the release is "6.0", and the platform is "x86_64".
Do you want to install Lite Cloud & Skyge HA 8.0 for "RedHat 6.0 x86_64" which we
recommend to you?
Please input your choice [y/n]? [y]: y
Installing Lite Cloud & Skyge HA 8.0 ......
```

图 2-1 Install Skyge HA 8.0

Totally, there are six packages will be install on the system. They are jiluos、messagebus、youchai、buluos、skygeha and emon。

```
Please input the third ip address: 192.168.1.1:192.168.1.254
Install Lite Cloud & Skyge HA 8.0 done.

Congratulation !!
Lite Cloud & Skyge HA 8.0 install succed!!

*************************************************************************
        Skyge Soft Technology (Beijing) Co.,Ltd

        Tel:        010-82359411/12
        Fax:        010-82359428
        Mail:       support@skyge.com
        Website:    http://www.skyge.com
*************************************************************************
```

After installation, There is a prompt for inputting ‘Third IP’, User can input over one IPs separated by “:”

Notice :The third IPs should be those IP which “NEVER” fail, such as router, gateway and so on. Why setup third IP, that’s a kind of way for HA software to determine if node has separated from network. If node cannot reach any

如果在安装后，需要改变第三方参考 IP，请修改/etc/skygeha_8/文件中的相应内容。
If need to revise those IPs, Please modify configure file /etc/skygeha_8/third_ips

After installation, Some environment variables are added into /etc/profile.

```
# tail /etc/profile
export PATH=$PATH:/opt/jiluos/bin
export PATH=$PATH:/opt/messagebus/bin
export PATH=$PATH:/opt/youchai/bin
export PATH=$PATH:/opt/buluos/bin
export PATH=$PATH:/opt/skygeha_8/bin
export PATH=$PATH:/opt/emon/bin
```

Run blow command can help loading those variables into user’s current running environment.

```
# source /etc/profile
```

2.2 Configure LiteCloud & Skyge HA 8.0

After installation, configuration should follow some steps. For the reason that LiteCloud is the base of Skyge HA, The configuration step are divided into two parts. Firstly, Configure LiteCloud to form a cluster environment, Secondly, Configure Skyge HA to enable HA functions.

Chapter2. Configure JILUOS

Jiluos is the base of LiteCloud, it provides basic communication functions such as reliable broadcast, some kind of data sync functions, cluster group functions, and so on. All the other packages in LiteCloud and Skyge HA depend on top of this package.

Jiluos package provides command jl_config, Which can help to configure, check, operate Jiluos module. Type jl_config -h can get all help info.

```
# jl_config -h
  -h, --help          Print help and exit
  -c, --config        config data
  -s, --status        Show status of Jiluos
  -n, --netid         Set/Get netid
  -d, --domain        Set domain
  -e, --eth           Set eth card
  -p, --path          Set path number
```

3.1、Configure Jiluos Heartbeat Channels

Jiluos can provide two communication channels, which means when one channel break, the other can take over the communication job. The multiple channel can promote availability greatly.

Kernel module jiluos.ko must be loaded into kernel before configure Jiluos system, jl_config command will load this module automatically if it find module is not in place. Using below command to verify if jiluos module has been loaded successfully.

```
# lsmod |grep jiluos
Module                  Size    Used by
jiluos                  143593    0
```

User can load jiluos module manually:

```
# insmod /opt/jiluos/drivers/jiluos.ko
```

When be sure kernel module has been loaded successfully, the next step is to configure Jiluos to make it join into a subnet and act as a active node.

3.1.1、Configure Jiluos Module

Using below command to configure Jiluos.

```
# jl_config -n netid -d domainid -e ethcard -p pathnumber
```

jl_config parameters are listed in table 3-1:

Param	Scope	Comment
netid	1 --- 1024	A netid represent a virtual sub network, Different sub network doesn't conflict for each other, All nodes in one subnet will work as a cluster group. So, generally, user can consider one subnet represent one LiteCloud cluster.
domainid	1 --- 64	One cluster group own at most 64 nodes, One node own one unique domain ID from 1 to 64, User must ensure the domain ID should be unique in the subnet.
ethcard	Network adapter name	For example : eth0、eth1、bond0
pathnumber	Multipath number 1 or 2	Support multipath to enhance availability.

表 3-1 jl_config 配置参数含义

For example, set up a LiteCloud cluster which contain 3 nodes, The sub network ID is 1000, 3 nodes Domain ID are 1, 2, 3. Every node has 2 network adapters eth0 and eth1. Perform below command:

```
Configure node 1 by using eth0 as multipath channel 1.
 [root@node1 ~]# jl_config -n 1000 -d 1 -e eth0 -p 1
Configure Jiluos system successfully
Configure node 1 by using eth1 as multipath channel 2.
 [root@node1 ~]# jl_config -n 1000 -d 1 -e eth1 -p 2
Configure Jiluos system successfully

Configure node 2 by using eth0 as multipath channel 1.
[root@node2 ~]# jl_config -n 1000 -d 2 -e eth0 -p 1
Configure Jiluos system successfully
Configure node 2 by using eth1 as multipath channel 2.
[root@node2 ~]# jl_config -n 1000 -d 2 -e eth1 -p 2
Configure Jiluos system successfully

Configure node 3 by using eth0 as multipath channel 1.
 [root@node3 ~]# jl_config -n 1000 -d 3 -e eth0 -p 1
Configure Jiluos system successfully
Configure node 3 by using eth1 as multipath channel 2.
[root@node3 ~]# jl_config -n 1000 -d 3 -e eth1 -p 2
Configure Jiluos system successfully
```

3.1.2、Configure Jiluos By Using Configure File

User can write a configuration file to persistent configuration, Name configure file as jiluos.cfg and put it under /etc/jiluos/, For example, below is the contents of node 1 configure file.

```
[root@node1 ~]# cat /etc/jiluos/jiluos.cfg
domain=1;
subnet=1000;
netcard1="eth0";
netcard2="eth1";
```

Use jl_config -c to load configuration file.

```
[root@node1 ~]# jl_config -c
Configure Jiluos system successfully
```

3.2、Check Jiluos Configuration Info

After configuration, use dmesg or jl_config -s to check configuration info.

If configure successfully, user can see blow output by using dmesg command.

```
[root@node3 ~]# dmesg
JILUOS: Enabled bearer <eth:eth0>, discovery domain <220.1000.0>, priority 10,
network plane A1
JILUOS: Established link <220.1000.3:eth0-220.1000.2:eth0> on network plane
A1
JILUOS: Established link <220.1000.3:eth0-220.1000.1:eth0> on network plane
A1
```

After configure all three nodes, run jl_config -s, output is like below

```
[root@node1 ~]# jl_config -s
Type       Lower      Upper      Port Identity              Publication Scope
0          3695083521 3695083521 <220.1000.1:1970733059>    1970733060  cluster
           3695083522 3695083522 <220.1000.2:4005494787>    4005494788  cluster
           3695083523 3695083523 <220.1000.3:996474883>     996474884   cluster
1          1          1          <220.1000.1:1969684481>    1969684482  node
```

The output clearly show that there are 3 nodes in this cluster.

> Notice : Port Identity and Publication are for internal usage.

3.3、Delete Jiluos Configuration Info

If want to remove one node from LiteCloud cluster, remove the jiluos configuration by using below command.

```
# jl_config -r
Remove jiluos system cleanly
```

Chapter3. Configure MessageBus

4.1、Brief Introduction Of MessageBus

In cluster environment, reliable broadcast communication mechanism is very helpful for different applications. LiteCloud provides a new communication module MessageBus which bases on publish and subscribe model.

In MessageBus, there is no IP, port. All the data are wrapped as message, the messages are pushed into one or more buses. One bus is a channel which has a unique name in LiteCloud cluster. Any applications which get interested in the message pass through in the bus can register themselves on the bus, afterwards, all the messages pass through the bus will be routed to this registered application. When applications don't want to receive any messages from one bus, just unregister from the bus, messages routing will stop immediately.

By using message bus to communicate in cluster, the applications only contact with abstract bus name than dedicated IP and ports. And also, message bus can easily implement reliable broadcast in cluster which will reduce the complexity of distributed programming greatly.

4.2、Launch MessageBus

It's extremely simple to launch MessageBus, Just run below command on each node:

```
# mb_message_bus -d
```

When MessageBus daemon launched, users can setup different bus in MessageBus.

If users only want to setup SkygeHA to utilize the high availability, Please skip the following parts in this chapter.

4.3、Create Bus In MessageBus

To create user's own bus in MessageBus, use mb_cli_create_service command. This command can be run on any node in LiteCloud cluster. For example, one any node, run

below command:

```
# mb_cli_create_service -d 1 -n "Mytestmb"
create service Mytestmb 0 1 successfully.
```

This command create a bus named "Mytestmb" on domain node 1.

4.4、Check MessageBus Status

Use command mb_cli_get_service to list all the bus name on one node. For example:

```
# mb_cli_get_service -d 1 -l
There are 1 services
Service0 :    Mytestmb              0             64
```

Each bus will create a subdir in MessageBus configuration dir, User can check them by go through all the subdirs. But generally, we don't propose this way strongly, Wrong operation may break the bus info and result in unpredictable errors.

```
# ls -l /etc/messagebus/slots/
total 0
lrwxrwxrwx 1 root root 33 Dec  27 18:57 64 -> /etc/messagebus/services/Mytestmb

# cat /etc/messagebus/slots/64/service.cfg
type = "collector";
slot = 64;
magic = 1953459315;
```

4.5、Send Message To Bus

Use command mb_general_provider to send message to bus, For example, send "Hello" to "Mytestmb" bus on domain 1.

```
# mb_general_provider -d 1 -n "Mytestmb" -m "Hello"
send successfully
```

4.6、Receive Message From Bus

LiteCloud software collection provide a tool yc_msg_spy which can watch bus and output those standard message passing through in the bus.

yc_msg_spy can run on any nodes in cluster and watch any bus on any node. For example, yc_msg_spy watch bug “Mytestmb” on node 1.

```
# yc_msg_spy -d 1 -n "Mytestmb" -p data
----------receive data msg----------
Data source : (null)
Domain      : 1
DATA Info   :
    mb_i    : 0
    mb_f    : 0.000000
    mb_d    : 0.000000
    mb_s    : Hello
----------receive data msg----------
```

4.7、Delete Bus

mb_cli_delete_service command can delete dedicated bus. For example, remove bus “Mytestmb” on node 1.

```
# # mb_cli_delete_service -d 1 -n "Mytestmb"
Delete service Mytestmb successfully.
```

Chapter4. Configure Buluos System

5.1、Brief introduction Of Buluos

For any cluster system, distributed lock system is the fundamental infrastructure. There is no exception, LiteCloud also provide a distributed lock system which is called Buluos.

Buluos provides 2048 (0 - 2047) slots, each slot contains 8192 (0 - 8191) locks. Totally, one LiteCloud can provide 2048 * 8192 locks.

Buluos lock system is different from other distributed lock system, it provides 4 types of lock.

- Random lock: slot scope is 0 - 511
- Fifo lock: slot scope is 512 - 1023
- Stack lock: slot scope is 1024 - 1535
- User defined lock: slot scope is 1536-2047

Consider this scenario, one lock is hold by an application, there are several applications on different nodes are waiting for the lock. For random lock type, when this lock is released by holder, this lock will be assigned to one waiting application randomly. So, fifo lock type means the application which wait for lock longer than other applications get the lock. Stack lock is the reverse of Fifo lock type. User defined lock type will provide interface to users to define what's the rule of getting the lock for those waiting applications.

5.2、Launch Buluos

For the reason that Buluos depends on MessageBus service, Please be sure MessageBus is running before launch Buluos. And please run bl_init.sh to initialize Buluos running environments if it's the first time to launch Buluos.

Do the same work on each node in cluster.

```
# bl_init.sh
create service log.buluos.skyge.com 0 1 successfully.
log.buluos.skyge.com has been created

# cat /opt/buluos/bin/bl_init.sh
#!/bin/sh
BL_LOG_CHANNEL="log.buluos.skyge.com"
domain=`/opt/jiluos/bin/jl_get_local_domain`
function get_result()
{
       local result=$1
       local name=$2

       if [ $1 -eq 0 ]; then
              echo $name has been created
       else
              echo $name has been existed or created failed
       fi
}
/opt/messagebus/bin/mb_cli_create_service -d $domain -n $BL_LOG_CHANNEL
get_result $? $BL_LOG_CHANNEL
```

buluos -h show all options

```
# buluos -h
  -h, --help            Print help and exit
  -d, --background      Run in background (daemonize)  (default=off)
  -t, --log-type        Set log type :
                           console
                           syslog
                           bus
                           all
  -l, --log-level       Set log Level :
                           LOG_EMERG
                           LOG_ALERT
                           LOG_CRIT
                           LOG_ERR
                           LOG_WARNING
                           LOG_NOTICE
                           LOG_INFO
                           LOG_DEBUG
  -i, --ips             Set IPs, Which is checked to avoid network isolation
```

Buluos option list table :

option	value	comment
-d	none	Run as a background daemon
-t	console	log output to console
	syslog	syslog (/var/log/skygeha.log)
	bus	bus (log.buluos.skygeha.com bus)
	all	all (output to console, syslog and bus)
-l	LOG_EMERG	Log level. Same as linux syslog log level
	LOG_ALERT	
	LOG_CRIT	
	LOG_ERR	
	LOG_WARNING	
	LOG_NOTICE	
	LOG_INFO	
	LOG_DEBUG	
-i	Third Party IP	One or more third party ip can provide a way to avoid split brain.

Table 5-1

For example, launch Buluos in debug level and all the log info output to console, logfile and bus. Using ip 92.168.1.1 and 192.168.1.254 as third party IP, The command like below:

```
# buluos -d -t all -l LOG_DEBUG -i "192.168.1.1:192.168.1.254"
Buluos: Starting up......
```

5.3、Buluos Testing Tools

Buluos provides tools to test distributed lock system which can help users understand buluos better, test_d_lock can acquire lock and release lock

For example, on each node, run below command.

```
# /opt/buluos/tests/test_d_lock -s 1500 -l 100
```

There will be only one node can get the lock and other nodes are in waiting state. Kill the holder application, there will be one application get the lock, and the holder is the latest waiter because 1500 slot reside in stack lock type range.

Chapter5. Configure Skyge HA 8.0

6.1、Brief Introduction Of Skyge HA 8.0

Skyge HA 8.0 is built on top of LiteCloud. Comparing with other HA software, SkygeHA8.0 has a lot of exiting features.

- Support utmost 64 nodes cluster.
- Support add or remove node without stopping cluster.
- Applications running in cluster can switch during all the nodes.
- Switch time is super faster than other HA software, typically, for some simple applications such as MySQL, the switch time only need less 1 second. But other HA software may need over 15 seconds.
- Environment monitoring system can predict some system error and trigger application switch, that feature enhance the availability greatly.
- Provide different level interface for developers and users.
- Easy configure, easy use. Very few configuration options needed.
- Provide plugin mechanism which can help users and developers to extend features.

6.1.1、Limitation Of Skyge HA 8.0

- Skyge HA 8.0 only support Linux platform
- Skyge HA 8.0 cluster cannot go across router, That means all nodes should stay in one network.

6.1.2、Skyge HA 8.0 Terminology

Terminology	Comment
Node	The computer installed SkygeHA and LiteCloud
Master node	The node service is running on.

Terminology	Comment
backup node	Those nodes prepare to take over service running on the master node. When master node has problem, the service will switch from master to one of slave nodes.
Service link	The IP which applications expose there services.
Heartbeat Link	The channels between nodes which used to send heartbeat message to test if node is still active and in good state.
Third party IP	When all the link between nodes are broken, each node should determine its own state, if the node make sure it has isolated with other nodes, this node should isolate itself out of the cluster by reboot or shutdown. That can avoid potential split brain risk. So, typically, the third IP should set to routers, gateways or other machines which 'NEVER' stop.
Service	The service indicate a collection of resources which includes processes, disk resources, network resources, float IPs and so on. One service can contain several resources and each resource can be several.
Float IP	Float IP is also called virtual IP, that's the IP export to outside users. The float IP is handled by SkygeHA, When one service switch from on node to another, The float IP will be shift with service, So, from user's perspective, they don't realize the backend changes because what they face is IPs.
Split Brain	Split brain indicate two parts of the cluster cannot see each other, but both sides are still running. When split brain happen, the two parts of the cluster will consider the other side is down and take over the service, that will lead to the resources in service conflict for each other, for example, the storage resource conflict may crash the file system, and the float IP conflict may block user to access the service. Split brain is very serious fault in HA system, So, HA system use several ways to avoid this fault.
Active-Standby	Active Standby means one node act as master and the other act as backup.
Active-Active	Active Active means both nodes runs different services and act as backup with each other.
Service switch	When node has problem, HA will stop service running on this node and release resources hold by those services. Then those services and resources will be switch to other nodes and start them up.

Table 6-1 Skyge HA 8.0 Terminology

6.2、Skyge HA 8.0 Service Scripts

Before configure service in SkygeHA and LiteCloud, user needs to prepare service scripts. Service script is used to start, stop and check service.

Some services already have script which is used to start stop service. For example, Mysql has mysqld in /etc/init.d/, Either modify the existed service script, or build new service scripts, both are OK if follow Skyge HA rules.

6.2.1、Service Script

One service can only have one service script, it's used to start service, stop service , it's also used to do some service state check.

6.2.2、Edit Service Script

Please follow below example to build user's service script. The installation disc contain some examples of service script, it's service_stat_stop_example.sh

Rules for building service script:

- Script parameters must contain "start" "stop" "status", the script must handle these three parameters.
  - Start : When service start, this parameter is transferred into script. Script should start the service correctly.
  - Stop : When service stop, this parameter is transferred into script, script should stop service correctly.
  - Status: during service's running phase, SkygeHA will invoke service script and pass 'status' parameter in. script need to check service state and report back to HA. For HA itself has some mechanism to check if service process exist or not, status parameters is not a necessary for some services. But for the reason that users know its own service much more than other applications. Users can use this parameter to build user's own service checking process to do accurate service checking.
- When script execute successfully, return 0, otherwise, return 1. For example, when 'start' parameter pass in and script start the service, if service start successfully, script should return 0 back to Skyge HA.
- Ensure service script has execution permission.

➢

6.2.3、Test Service Script

After finish building service script, Please test the script to ensure it works well. Otherwise, a wrong script may break HA system.

For example, there is a service script named myapp.sh

1)、start service

Execute ./myapp.sh start

Check if service has been launched successfully.

Check the return value if it’s value is correct.

2)、stop service

Execute ./myapp.sh stop

Check if service has been stopped successfully.

Check the return value if it’s value is correct.

3)、check service

Execute ./myapp.sh status

Check the return value if it’s correct

When service runs well, service script should return 0, otherwise, it should return 1.

6.3、Launch Skyge HA 8.0

Several steps are needed when launch Skyge HA.

- Initialize Skyge HA 8.0
- Prepare service configuration file on each node
- Set up take over policy.
- Start all Skyge HA processes.

6.3.1、Initialize Skyge HA 8.0

For the reason that Skyge HA is built on top of LiteCloud, So, ensure that message bus and buluos are in running state.

Please run ha_init.sh to initialize Skyge HA running environment if the node has never been configured. The ha_init.sh will setup some necessary bus.

```
# ha_init.sh
create service cmd.launcher.ha.skyge.com 0 1 successfully.
cmd.launcher.ha.skyge.com has been created
create service cmd.service.ha.skyge.com 0 1 successfully.
cmd.service.ha.skyge.com has been created
create service state.service.ha.skyge.com 0 1 successfully.
state.service.ha.skyge.com has been created
create service state.domain.ha.skyge.com 0 1 successfully.
state.domain.ha.skyge.com has been created
create service log.ha.skyge.com 0 1 successfully.
log.ha.skyge.com has been created
create service cmd.safeguard.ha.skyge.com 0 1 successfully.
cmd.safeguard.ha.skyge.com has been created
create service cmd.ecs.ha.skyge.com 0 1 successfully.
cmd.ecs.ha.skyge.com has been created
create service cmd.userheartbeat.ha.skyge.com 0 1 successfully.
cmd.userheartbeat.ha.skyge.com has been created
create service pulse.userheartbeat.ha.skyge.com 0 1 successfully.
pulse.userheartbeat.ha.skyge.com has been created
create service cmd.checker.ha.skyge.com 0 1 successfully.
cmd.checker.ha.skyge.com has been created
```

6.3.2、Prepare Service Configuration File

Each service in Skyge HA needs a configuration file, user can find a sample configuration file in installation disc. For example, below configuration file is for MySQL service.

```
name = "mysql";
onstart = "yes";
script = "/etc/init.d/mysqld";
fip = "eth0-192.168.1.100-255.255.255.0-192.168.1.255:";
store = "192.168.1.253-/dev/sdb1-/var/lib/mysql-755-mysql-mysql:"
start_timeout = 30;
stop_timeout = 30;
check_interval = 5;
```

When building configuration file, Please follow the rules listed in samples. Each line form as 'option = value' and each line should be separated by semicolon.

If value is other than numerical number, use quotation mark to change it to be string.

Option is listed in below table

option	Description	example	comment
name	Service name	mysql apache oracle	Must use character and number, Doesn't support special character such as : *、?、\、# and so on.
script	Service script absolute path	/etc/init.d/mysqld /opt/skygeha_8/myapp.sh	Please refer to section 2 in this chapter.
desc	Service description info	MySQL for Test	Service general description.
fip	Float IP	eth0-192.168.1.100-255.255.255.0-192.168.1.255 :	Each float IP contain network adapter name, IP address, network mask, broadcast name. Those 4 items are separated by dash and different float IP group is separated by colon.
store	Storage info	192.168.1.253-/dev/sdb1-/var/lib/mysql-755-mysql-mysql:	Each storage group contain disk array IP, device name, mount dir, mount permission, mount user and mount user group. Those six items are separated by dash and different storage group is separated by colon.
start_timeout	Service start timeout (second)	100	Maximum service start timeout, if service start time exceed this timeout, SkygeHA will consider the service start failed. Default value is 300 seconds
stop_timeout	Service stop	100	Maximum service stop timeout, if

	timeout (second)		service stop time exceed this timeout, SkygeHA will consider the service stop failed. Default value is 300 seconds
resource_release_timeout	Resource release timeout (second)	30	Maximum service resource release timeout, if service resource release time exceed this timeout, SkygeHA will consider the service resource release failed. Default value is 300 second
onstart	If start when Skyge HA start	yes or no	If set yes, service will start when Skyge HA start. Otherwise, the services need to be started up manually. Default value is no.
check_interval	Service check interval (second)	5	How often Skyge HA will invoke service script to check service status. Default value is 3 seconds.
check_tko	Service check TKO	2	How many times Skyge HA detect service error will lead to service is switched out of this node Default value is 1.
force_load	Force to load	Yes or no	Most of service can bypass this option, but for some applications such as snmpd, smartd cups which close all the file handlers before running, need to set this option to handle this case. Default is no
force_reboot	Force to reboot	yes or no	When service stop failed or resource release failed and this option is set to be yes, system will be reboot to isolate this

			node from cluster for safety sake.

表 6-2 服务配置文件格式说明

6.3.3、Setup Service Take Over Order

When service start on one node and act as master node, Skyge HA can choose backup nodes from the rest nodes in cluster. When master node failed, Those backup node should follow some policies to take over the service.

According to Buluos, LiteCloud provides three lock type, which are random, fifo and stack. The same in Skyge HA, it utilizes this feature to provide three type of take over policy, random, fifo and stack.

In /etc/skygeha_8/slots/, there are random, fifo and stack directory, which correspond to random, fifo and stack take over policy, For example, Those service configuration files put in random directory will follow random take over policy.

Each directory can contain 1024 sub directories, one sub directory correspond to one service,

在服务于起始节点启动后，Skyge HA 可以根据其余备份节点的启动顺序（由节点启动 sk_launcher 进程的先后）设置节点在接管服务时的先后顺序，从而可以更好的控制由哪个节点优先接管指定的用户服务。服务接管顺序分为三种类型：random（随机接管）、fifo（先启动的先接管）和 stack（后启动的先接管）。

在/etc/skygeha_8/slots 下有三个目录 random、fifo 和 stack，分别对应不同的服务接管顺序。每个目录下最多可包含 1024 个子目录，每个子目录对应一个服务，代表每种接管顺序类型均可最大支持 1024 个服务。每个子目录的名称必须以 0 至 1023 之间的整数命名，该子目录只能包含一个 service.cfg 的配置文件且不能为空。请将准备好的服务配置文件，放入您所需要的接管顺序下的子目录中。例如，将上一小节准备的 mysql 的 service.cfg 文件放入/etc/skygeha_8/slots/fifo/0 目录中。

此步操作必须在所有节点上分别执行完成，并且同一服务配置文件必须放置在每个节点相同的目录路径下。

6.3.4、Launch Skyge HA Processes

Skyge HA has four processes.

- sk_manager: HA service management process, it maintain services state and send start or stop command to service launcher.
- sk_launcher: HA service launcher process, it receives command from sk_manager process, it is responsible for starting, stopping service and service allocation and release.
- sk_checker: HA service checking process, it will check service resources and invokes service script to check if all those resources are available for service. Checker will report all the checking state to manager process. Manager will do corresponding actions according to checking results. For example, if checker find service float IP has problem and report to manager, manager process will inform launcher to stop this service and launch service switch process.
- sk_safe_guard: service resource collection process, it will ensure all the resource reside on this node are valid, if some service has been switch to other node but some resources are still kept in this node, safe guard process will remove those resource out of this node.

Launch those four processes on each node like below:

```
# sk_manager -d -t all -l LOG_DEBUG
# sk_launcher -d -t all -l LOG_DEBUG
# sk_checker -d -t all -l LOG_DEBUG -i "192.168.1.1:192.168.1.254"
# sk_safe_guard -d -t all -l LOG_DEBUG
```

For sk_checker, there is an option -i, this indicate third party IP, When all channels between two nodes break, those two nodes cannot see each other, if one side cannot reach any third party IPs, the checker will invoke /opt/skygeha_8/scripts/sk_ha_ck_network_isolation.sh script, typically, there is only one line in this script, that's "reboot". That will help this isolated node detached from cluster to avoid split brain.

由于在服务的配置文件中使用了“onstart=yes”，因此最先启动 sk_launcher 的节点会首先启动用户服务，可以通过检查用户服务进程及 ifconfig 来验证服务启动成功。例如：

```
# ifconfig
eth0:8200 Link encap:Ethernet  HWaddr 00:0C:29:19:D6:48
          inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:19 Base address:0x2000

# ps -A
（节选，表明该节点已成功启动 mysql 服务）
  PID TTY          TIME     CMD
 1645   ?        00:00:00    mb_message_bus
 1650   ?        00:00:00    buluos
 1659   ?        00:00:00    sk_manager
 1663   ?        00:00:00    sk_launcher
 1678   ?        00:00:00    2-0-rr-lqsym
 1933   ?        00:00:00    mysqld_safe
 2030   ?        00:00:00    mysqld
 2063   ?        00:00:00    sk_checker
 2067   ?        00:00:00    ck-nw-watchdog
 2068   ?        00:00:00    ck-jl-watchdog
 2076   ?        00:00:00    2-0-ck-lqsym
 3443   ?        00:00:00    sk_safe_guard
 3721 pts/0      00:00:00    ps
```

在主节点（用户服务正在运行的节点）上会看到相应的释放资源进程（如：2-0-rr-lqsym）其他节点上 mysql 服务没有启动，但会看到等待服务的锁进程（如：2-0-wl-lqsym）。

6.4、Query Skyge HA Service Status

After Skyge HA start and services are launched, use ha_query to get the service state on any node. For example, query all the service state on node 1. The output may like below:

```
# ha_query -d 1
Type : stack : Slot : 2   Service : smb
Type : stack : Slot : 3   Service : httpd
Type : random : Slot : 1   Service : bind
Type : fifo : Slot : 0   Service : mysql
```

Command ha_status can show all the nodes which serves as masters and backup nodes. The output may look like below:

```
# ha_status
Service -- random:  1 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
Service -- fifo:    0 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
Service -- stack:   2 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
Service -- stack:   3 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
^C
```

6.5、Add And Remove Service Dynamically

Skyge HA support to add and remove service dynamically, user doesn't need to stop cluster and HA software.

Use command ha_svc_commander can handle different service actions. Type -h can get all the command info.

```
# ha_svc_commander -h
-h, --help     Print help and exit
-d, --domain   Set domain info
-s, --slot     Set slot info
-t, --type     Service type:
                      random
                      fifo
                      stack
-a, --action   Set action info:
               start: start service, put service in running state or waiting state
               stop : stop service.
               release : release the service which is in waiting state
               add : add one service in skygeha system
               remove  : remove one service out of skygeha system
```

When adding service into Skyge HA, please prepare service configuration file according to section 2 and 3. Put it in one sub directory under /etc/skygeha_8/slots/, and copy it to all the nodes which want to be backup nodes.

For example, on node 1, add a service which has service ID 4. Choose service type

as Stack, Follow below steps to finish the service add and start.

```
# ha_svc_commander -d 1 -s 4 -t stack -a add
# ha_svc_commander -d 1 -s 4 -t stack -a start
```

Stop service before remove service, For example, on node 1, follow below steps to remove service 4.

```
# ha_svc_commander -d 1 -s 4 -t stack -a stop
# ha_svc_commander -d 1 -s 4 -t stack -a remove
```

在其他备份节点（如节点 2）上执行删除操作：

```
# ha_svc_commander -d 2 -s 4 -t stack -a release
# ha_svc_commander -d 2 -s 4 -t stack -a remove
```

删除服务后，请删除所有节点上相应的用户配置文件。

6.6、User Heartbeat

User heartbeat is designed for communication between Skyge HA and user applications. For traditional HA software, HA monitor application by checking if process still exist. If process is running, HA will consider service is in good state. This method is not accurate when process has gone into “dead loop” state. Although processes still exist in system, it cannot provide any service actually.

Skyge HA provide user heartbeat feature which can receive user application’s ‘heartbeat’ messages, Skyge HA will consider user applications died if it doesn’t receive any message in defined period. This feature can help Skyge HA to determine application’s accurate running state and resolve application’s “fake alive” problem.

Chapter6. Configure Emon System

7.1、Brief Introduction Of Eon

Emon is a system monitor which can watch different resources, such as cpu usage, memory usage, storage and so on. It's very easy to use Emon to monitor nodes states in cluster.

Combine with Skyge HA, Emon can provide a way to do switch proactively. Typical HA software switch service when some serious failure happen, but when system is not stable any more, it may affect HA software itself, which means HA software can become not stable too. With Emon, Skyge HA can predict some serious failure before it happens. Consider memory leak as an example, it always steal memory silently and slowly, when all the memory are gone, system is broken and it's very hard for HA to do switch because HA itself is always frozen for memory shortage. For this case, Emon can set several thresholds for memory usage, each threshold links with actions, for example, user can set 2 memory thresholds, 70% and 90%, 70% link with the action of send out warning by email, 90% link with the action of switch all the applications on this node to other nodes. So, when 70% threshold is hit, warning info will be sent out, this can give system administrator an opportunity to handle. Administrator can check system to find out memory leak problem.

To extend flexibility, Emon design a plugin system, user can build different plugin to monitor different resources. When plugin is linked with Emon, Emon can invoke those plugins by indicated intervals.

7.2、Initialize Emon System

If configure Emon for the first time, Please run em_init.sh to initialize Emon first of all. The script is to set up some necessary message bus for Emon.

```
# em_init.sh
create service log.emon.skyge.com 0 1 successfully.
create service cmd.emon.skyge.com 0 1 successfully.
create service cpu_load.emon.skyge.com 0 1 successfully.
create service mem_load.emon.skyge.com 0 1 successfully.
create service disk_usage.emon.skyge.com 0 1 successfully.
```

7.3、Configure ECS

Emon system can be used as an independent software, Before start Emon system, Please configure ECS environment first of all. Use Skyge HA Emon configuration as an example.

7.3.1、Check Emon Plugins

User can check Emon configuration directory to know how may plugins Emon has. For example, There are three default plugins in Emon system. They are cpu_load, disk_usage and mem_load.

```
# ls /etc/emon/services/
cpu_load  disk_usage  mem_load
$匀吧万丧瓜当%

# ls /etc/emon/services/*
/etc/emon/services/cpu_load:
cpu_load.a  cpu_load.la  cpu_load.so  service.cfg

/etc/emon/services/disk_usage:
disk_usage.a  disk_usage.la  disk_usage.so  service.cfg

/etc/emon/services/mem_load:
mem_load.a  mem_load.la  mem_load.so  service.cfg
```

Users can develop their own plugin and link them into Emon system. Details refer to technical manual.

7.3.2、Prepare Threshold Action Script

Users need to prepare action scripts for thresholds, which contain the operation need to perform when threshold is hit.

For example, build a script which will reboot machine when memory usage is higher than 90%. Name the script as testmem90.sh, assign it execution permission and put it in /etc/emon/mem/

```
# ls -l /opt/emon/cpu-mem-disk/testmem90.sh
-rwxr-xr-x 1 root root 74 Dec 26 14:01 /opt/emon/cpu-mem-disk/testcpu90.sh

# cat /opt/emon/cpu-mem-disk/testmem90.sh
#!/bin/sh
reboot
```

7.3.3、Prepare Emon ECS Configuration File

Emon provides three default plugins, which reside in /etc/skygeha_8/ecs/ directory.

```
# ls /etc/skygeha_8/ecs/
cpu_load  disk_usage  mem_load
（如果没有相应目录，请使用 mkdir 命令在/etc/skygeha_8/ecs 中自行创建）
```

Plugin configuration file must be name as service.cfg and put it on related plugin directory. The configuration file is to set up all kinds of thresholds and action scripts. For example, set up a cpu load threshold and assign an action script.

```
# cat /etc/skygeha_8/ecs/cpu_load/service.cfg
watermarks = "wm_90";
wm_90 = "script:/opt/emon/cpu-mem-disk/testcpu90.sh:above";
```

ECS option description, table 7-1:

option	Description
Watermarks	One threshold: watermarks = "wm_*XX*"; Multi threshold: watermarks = "wm_*XX*:wm_*XY*"; In cpu_load and mem_load, XX means the percentage of usage. In Disk_uasage, it means the root device usage (count as KB). For example: set up cpu 90% as a threshold : watermarks = "wm_90";

	Setup disk usage 250MB as a threshold : watermarks = "wm_250000";
wm_XX	wm_*XX* = "script:*pathtoscript*:*direction*"; There can be several lines for wm_XX, wm_XX must match with Watermark option values. Pathtoscript is the absolute path to the actions script, direction has three kind of value, above, below, equal which means the action will be taken when above the threshold, below the threshold and equal to threshold.

表 7-1 ECS 配置文件格式说明

7.3.4、Launch Emon

Run command sk_dynamic_ecs to launch ECS system. -d option can run service in backend.

```
# sk_dynamic_ecs -d -n cpu_load -t all -l LOG_DEBUG
```

7.4、Launch Emon

Use below command to launch Emon system.

```
# emon -d
```

Notic: running in frontend, Emon will continue to output monitor data on screen.

Chapter7. Uninstall LiteCloud & Skyge HA

When ensure to uninstall LiteCloud and Skyge HA, please follow the steps in this chapter.

The uninstall process include below steps.

❶Stop all the service which is managed by LiteCloud and Skyge HA

❷Stop SkygeHA and LiteCloud.

❸Uninstall LiteCloud and Skyge HA.

8.1、Stop All Service In SkygeHA And LiteCloud

Before uninstall LiteCloud and Skyge HA, please ensure all the services on this node has been stoped. Use ha_svc_commander to stop service. For example: if stop stack type service 4 which is running on node 1. Execute below command:

```
# ha_svc_commander -d 1 -s 4 -t stack -a stop
```

8.2、Stop LiteCloud And SkygeHA

Use below command to stop Skyge HA and LiteCloud.

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

8.3、Uninstall LiteCloud And Skyge HA

Insert installation disc and mount to any directory. Run uninstall_lc.sh to uninstall all packages.

```
# ./uninstall_ha.sh
Uninstalling LiteCloud & Skyge HA 8.0 program ......
Do you want to continue [y/n] ? y
LiteCloud & Skyge HA 8.0 uninstallation finished.
```

Appendix LiteCloud And Skyge HA Maintain

1、Skyge HA Running Status

Several ways to check Skyge HA running status.

- Check if all the processes run well
- Check if LiteCloud and service status
- Check Skyge HA service configuration

1.1、Check Skyge HA Daemon Status

Use ps command to check if all the daemon are running.

```
# ps -A
  PID TTY          TIME     CMD
 1645 ?        00:00:00 mb_message_bus
 1650 ?        00:00:00 buluos
 1659 ?        00:00:00 sk_manager
 1663 ?        00:00:00 sk_launcher
 2063 ?        00:00:00 sk_checker
 2067 ?        00:00:00 ck-nw-watchdog
 2068 ?        00:00:00 ck-jl-watchdog
 3443 ?        00:00:00 sk_safe_guard
 3720 ?        00:00:00 emon
 3724 ?        00:00:00 sk-ecs-cpu_load
 3720 ?        00:00:00 sk_safe_guard
```

1.2、Check Skyge HA And Service Status

Use ha_status to check service status in Skyge HA

```
# ha_status
Service -- random:  1 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
Service -- fifo:    0 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
Service -- stack:   2 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
Service -- stack:   3 is running on 1
       domain waitting for lock :[ 2 ] [ 3 ]
^C
```

ha_status can show all the service status in LiteCloud and Skyge HA, list service type, service ID, domain ID and backup node ID.

1.3、Check Skyge HA Service Configuration

Use command ha_query to check all services running on node. For example, use below command to list all service running on node 1.

```
# ha_query -d 1
Type : stack : Slot : 2   Service : smb
Type : stack : Slot : 3   Service : httpd
Type : random : Slot : 1   Service : bind
Type : fifo : Slot : 0   Service : mysql
```

2、Skyge HA Log

All the Skyge HA log are stored in /var/log/skygeha.log file, User can check the log file to get a lot of useful info, or send log file to Skyge Soft to do further analysis.

Notic

- Please don’t remove the log file when Skyge HA is running.
- If there is no log info output to log file, please check if syslogd has been started successfully.

3、FAQ

3.1、Why Installation Fails

If installation fails and output like below:

```
error: Failed dependencies:
libconfig.so.8()(64bit) is needed by buluos-1.0-Header_XX.x86_64
libconfig.so.8()(64bit) is needed by emon-1.0-Header_XX.x86_64
libconfig.so.8()(64bit) is needed by jiluos-1.0-Header_XX.x86_64
libconfig.so.8()(64bit) is needed by messagebus-1.0-Header_XX.x86_64
libconfig.so.8()(64bit) is needed by skygeha-8.0-Header_XX.x86_64
libconfig.so.8()(64bit) is needed by youchai-1.0-Header_XX.x86_64
```

It shows system lack libconfig packages in CentOS and Redhat, in Suse, the packages are libconfig and libconfig-devel. Download them from official website or get them from installation disc.

3.2、Why Fails To Load Jiluos Module

During execution of command jl_config -c, if "err init cfg file" shows on screen, Please check /etc/jiluos/jiluos.cfg configuration file to ensure file exist and all options are well configured according to grammar.

3.4、Why sk_manager Launch Fails

If sk_manager start fails and there are below info shows in log file:

```
ha_create_service : failed to init service on "fifo\XXX\service.cfg"
Can not init service from config path, ret : -1042
```

Please check if /etc/skygeha_8/slots/fifo/XXX/service.cfg file exist and options in file are well configured.

4、Support

LiteCloud and Skyge HA are developed by Skyge Soft, Thanks for choosing our products. Any questions, please refer to our website. If want to get professional and commercial support, Please contact with our sales and support team.

Tel：(010) 82359438

Fax：(010) 82359428

Web： www.skyge.com

E-mail：support@skyge.com