

快速安装手册

指导您快速安装及使用LiteCloud & Skyge HA 8.0 的手册,以节省您的宝贵时间。

北京神州天勤软件技术有限公司

2012-12-27





目录

简介		4
1、 l	LiteCloud & Skyge HA 8.0 简介	4
2、 l	LiteCloud & Skyge HA 8.0 的架构图	5
3、I	LiteCloud 的组成模块	5
	快速安装手册的使用方法	
5、 l	LiteCloud & Skyge HA 8.0 快速安装的主体过程	
第一章	安装前的准备工作	
	了解和分析您的应用	
	准备 LiteCloud & Skyge HA 8.0 需要的硬件环境	
	准备 LiteCloud & Skyge HA 8.0 需要的软件环境	
	安装前需要准备的操作系统的配置	
	安装前需要准备用户的应用程序	
	安装的配置示例	
第二章	安装 LiteCloud & Skyge HA 8.0	
	安装 LiteCloud & Skyge HA 8.0	
	注册 LiteCloud & Skyge HA 8.0	
第三章	配置 JILUOS	
3.1、	配置 jiluos 心跳通道	
	3.1.1 、手工配置 jiluos 心跳通道	
	3.1.2 、使用配置文件配置 jiluos	
	· 查看节点 jiluos 配置信息	
	删除 jiluos 配置信息	
第四章	配置 MESSAGEBUS	
	MESSAGEBUS 简介	
	启动 MESSAGEBUS	
	创建消息总线	
	查看消息总线	
	向消息总线发消息	
	接收指定的消息	
	删除消息总线	
第五章	配置 BULUOS	
	BULUOS 简介	
	// // // // // // // // // // // // //	
3.3、 第六章		
	配置 Skyge HA 8.0	
0.1	6.1.1、Skyge HA 8.0 的使用限制	
	6.1.2、Skyge HA 8.0 相关术语	
6.2	6.1.2、Skyge HA 8.0 相关不语	
0.25	6.2.1、用户脚本	
	6.2.2、编写用户脚本	
	U.C.C 1 7四 ¬J / I / ルデイ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	



6.2.3、测试用户脚本	29
6.3、启动 Skyge HA 8.0	30
6.3.1、初始化 Skyge HA 8.0	30
6.3.2、准备服务配置文件	31
6.3.3、设置服务接管顺序	32
6.3.4、启动 Skyge HA 的所有进程	33
6.4、查看 Skyge HA 服务及状态	34
6.5、动态添加及删除服务	35
6.6、用户心跳功能	36
第七章 配置 EMON	37
7.1、EMON 简介	37
7.2、初始化 EMON	37
7.3、配置 ECS	37
7.3.1、查看 ECS 支持模块	38
7.3.2、准备 ECS 执行脚本	38
7.3.3、准备 Skyge HA 的 ECS 配置文件	38
7.3.4、启动动环监控进程	39
7.4、启动 EMON	39
第八章 卸载 LiteCloud & Skyge HA 8.0	
8.1、停止所有用户程序	40
8.2、停止 LiteCloud & Skyge HA 8.0	
8.3、卸载LiteCloud & Skyge HA 8.0	40
附录 LiteCloud & Skyge HA 8.0 的维护	41
1、Skyge HA 8.0 的运行状态	41
1.1、检查各守护进程的运行状态	41
1.2、检查集群与服务的运行状态	42
1.3、检查配置的 Skyge HA 服务	43
2、Skyge HA 8.0 的日志信息	44
	45
3.1、为什么会安装失败	
3.2、为什么无法执行命令	
3.3、为什么加载 jiluos 配置会失败	
3.4、为什么 sk_manager 启动会失败	46
4、技术支持	46



简介

! 注意:如果您是第一次使用 LiteCloud & Skyge HA 8.0,在您开始安装和

使用之前,请全面仔细地阅读本手册!否则,可能会因为您的疏忽而使得安装过程

出现不必要的问题!

1、LiteCloud & Skyge HA 8.0 简介

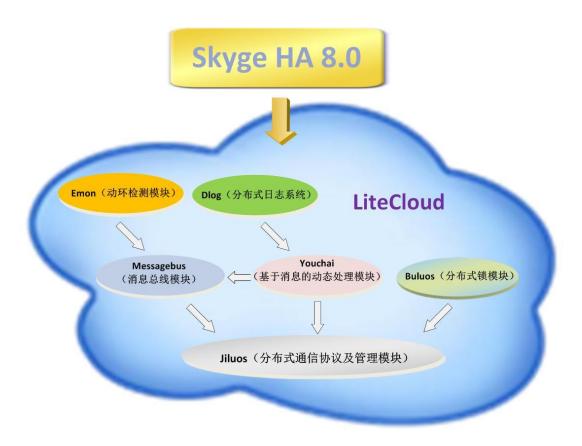
LiteCloud 是北京神州天勤软件技术有限公司自主研发并拥有自主知识产权的云计算平台产品。云计算本质上就是集群计算(CLUSTER COMPUTING)。LiteCloud 是全新一代的云计算平台、分布式系统平台、高可用平台,它可以完全满足企业级的云计算应用需求。同时它也是一款第三方的云计算平台,既可以允许开发者将写好的程序放在"云"里运行,也可以提供基于"云"的服务,供开发者创建应用时采用。开发者不必构建自己的基础,完全可以依靠 LiteCloud 来创建新的应用。LiteCloud 的直接用户是开发者,而不是最终用户。

Skyge HA 8.0 是北京神州天勤软件技术有限公司推出的为满足电信级和企业级应用的高可用集群软件产品。Skyge HA 8.0 基于最新的 LiteCloud 云计算平台,不仅全面实现了多对多的高可用集群功能,而且还更灵敏、更可靠、更稳定,大大减少了用户服务的故障切换时间,使得能够更好的满足用户业务的连续性和不同应用对高可用的要求。

需要连续高可用运行的用户业务计算机在安装 LiteCloud & Skyge HA 8.0 后,每个节点均具备了随时准备运行业务的能力。当正在运行业务的节点由于故障(如:掉电、关机、重启、业务网络断掉、用户进程被杀掉等)或其他原因终止用户业务后,立刻会有备份节点接管运行。只要网络中还有一台备份节点存在,用户业务就不会终止。



2、LiteCloud & Skyge HA 8.0 的架构图



3、LiteCloud 的组成模块

LiteCloud 由多个模块组成,模块间既相互合作又相互独立。每一个模块都可以独立的运行并提供相应的服务。任何一个模块都可调,使用灵活,配置简单,且执行速度非常快。

LiteCloud 是一个平台,用户可以基于 LiteCloud 轻松构建自己的"云"计算平台。LiteCloud 的多语言支持涵盖主流的开发语言: C、C++、Python、Perl、PHP、Java、Ruby。

LiteCloud 的五个组成模块:

- ▶ Jiluos: 基础通信模块。
- ➤ Messagebus: 消息总线模块。
- ▶ Buluos : 分布式锁模块。
- ➤ Youchai: 通用消息处理模块。
- ▶ Emon: 环境监测模块。

4、快速安装手册的使用方法

由于本手册是快速安装文档,所以各个章节的部分基本上都需要仔细阅读。如果您只是



为了很快完成配置,也可直接阅读每个章节中的配置小节部分。

5、LiteCloud & Skyge HA 8.0 快速安装的主体过程

LiteCloud & Skyge HA 8.0 快速安装的主体过程有如下七个步骤:

- ●进行安装前的准备工作,详见安装前的准备工作
- ❷安装 LiteCloud & Skyge HA 8.0 软件包,详见安装 LiteCloud & Skyge HA 8.0
- ❸配置 JILUOS,详见配置 JILUOS
- ●启动 MESSAGEBUS,详见配置 MESSAGEBUS
- ⑤启动 BULUOS, 详见配置 BULUOS
- ⑥配置并启动 SkygeHA, 详见配置 Skyge HA 8.0
- ②配置监控系统 EMON, 详见配置 EMON

另外,本手册还在第八章介绍了如何卸载 LiteCloud & Skyge HA 8.0, 并且在附录中还列出了 LiteCloud & Skyge HA 8.0 安装使用过程中常见的问题以及如何进行一些后期的维护。



第一章 安装前的准备工作

安装 LiteCloud & Skyge HA 8.0 之前,需要您做好相关的软硬件环境的准备工作。否则,可能会影响您随后的安装过程,使您遇到不必要的麻烦。安装前的准备工作可以分为如下六个步骤:

- ●了解和分析您的应用;
- ❷准备 LiteCloud & Skyge HA 8.0 需要的硬件环境;
- **3**准备 LiteCloud & Skyge HA 8.0 需要的软件环境;
- 母安装前需要准备的操作系统的配置;
- ⑤安装前需要准备的用户的应用程序;
- 6安装的配置示例。

1.1、了解和分析您的应用

LiteCloud & Skyge HA 8.0 可以支持绝大多数的高可用应用。根据用户应用是否需要使用共享的数据资源,Skyge HA 8.0 将用户的应用分为两类:

▶ 第一类应用:需要存取共享数据的应用。

这类应用的特点是,应用的数据非常重要且 I/O 负载较高,必须使用共享磁盘阵列来保证应用数据的一致性、可靠性和高性能。此类应用的典型例子是各类数据库应用,如 Oracle、Sybase、MySQL等。也就是说,使用 Skyge HA 8.0 的系统本身作为后台数据系统。

▶ 第二类应用:不需要存取共享数据的应用。

这类应用的特点是,使用 Skyge HA 8.0 的系统只是用于进程和浮动 IP 等资源的切换。一般情况下,这种类型的应用只是用来作为前端接入、前置机来使用。典型的应用有 WEB/MAIL 等网络应用或者是用户自己的业务应用。这类应用通常会有另外的后台数据库系统来保存数据。

上注意: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、准备 LiteCloud & Skyge HA 8.0 需要的硬件环境

安装 LiteCloud & Skyge HA 8.0 的硬件需求列表,见下表 1-1::

硬件需求列表			
硬件类型	数量	用途	备注
PC SERVER	≥2	安装节点所需的 Linux 操作系统	x86 架构
硬盘	1	用于安装 Linux 操作系统,其中/opt 目录可用空间	建议做 RAID-1 或
	1	应不小于 1GB	RAID-5
		至少要 2 块网卡,其中一块用于业务网线,另一块	业务链路使用千兆网
网卡	≥2	用于心跳线。如果有更多的网卡,可以为网卡做	卡,心跳链路使用百兆
		bonding,实现业务链路和心跳链路的冗余	网卡即可
交换机	≥1	通常情况下,一个交换机就够了; 如果要做业务网卡的 bonding,则必须要 2 个交换机	为保证高可用性,建议 使用两台交换机,机器 上的两块网卡分别连 接到不同的交换机上, 以保证网络连接的冗 余性
磁盘阵列 (可选)	1	用于存放共享数据	视实际应用而定

表 1-1 硬件需求列表

说明:我们建议在集群中使用两块网卡建立心跳通道,以保证我们应用服务的高可用。



1.3、准备 LiteCloud & Skyge HA 8.0 需要的软件环境

安装 LiteCloud & Skyge HA 8.0 的软件需求列表,见下表 1-2:

软件需求列表		
软件类型	用途	
Linux 操作系统	用于节点操作系统(可以是 RedHat、Suse、CentOS 等主流 Linux 发行版)	

表 1-2 软件需求列表

LiteCloud & Skyge HA 8.0 默认支持的操作系统列表,见下表 1-3:

	默认支持的操作系统列表		
	Red Hat Enterprise Linux Server release 5.6 64 位		
	Red Hat Enterprise Linux Server release 6.0 64位		
Red Hat	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 release 5.6 64位		
	CentOS Linux release 6.0 64位		
CentOS	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	SUSE Linux Enterprise Server 11.2 64位		

表 1-3 默认支持操作系统列表

!注意:用户在安装操作系统时,建议用户采用完全安装操作系统的模式,并且将操作系统安装在本地具有 RAID-10 (即 RAID 0+1)功能的磁盘设备上。这样可以保证操作系统功能的完备性和操作系统数据的可靠性。

在安装 LiteCloud & Skyge HA 8.0 前,需要安装相应的软件包。在 RedHat 和 CentOS 中,可能需要先行安装 apr-devel、apr-util-devel、libconfig 软件包。在 SuSE 系统中可能需要先行安装 libapr1、libapr1-devel、libapr-util1-devel、libapr-util1 软件包。



1.4、安装前需要准备的操作系统的配置

每个节点的操作系统配置需要确认如下环节:

- 1)、请确认您登陆操作系统的身份是 root 权限, 因为随后的安装配置都需要是 root 权限。
- 2)、所有网卡是否配置好。(包括网线是否连好、网卡 bonding 是否配置正确等)。
- 3)、由于 Skyge HA 8.0 使用了 Linux 操作系统的 syslog 服务(或 rsyslog 服务)来打印 Skyge HA 8.0 的 LOG 日志, 所以请确认 syslog 服务已经配置并且正在运行。例如:

```
# /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)、禁用 SELINUX。请设置系统中的/etc/selinux/config 文件中的 "SELINUX=disabled", 并在修改成功后重新启动系统。
- 5)、Skyge HA 8.0 的日志系统要求每台节点的系统时钟一致,以方便后期维护。如果每台系统时钟不一致,请使用 date 以及 hwclock 命令来调整时钟,使得每台节点的时钟大致相同,误差小于 60 秒。



1.5、安装前需要准备用户的应用程序

接下来,请在每个节点上安装用户的应用程序,每个节点的安装配置都要求是对等的,也就是说,在两个节点上安装的用户的 UID、GID、应用程序所在的目录、使用的共享磁盘设备名称、网卡名称等都应该一致。

通常,都会将应用程序安装在每个节点的本地硬盘上。

如果您的应用类别是第一类应用时,可能还需要在共享磁盘设备上准备共享的数据。例如:对于 Oracle 应用,您需要将表空间文件存放在共享磁盘设备上。

如果使用 Skyge HA 来保证用户服务的高可用集群,请将用户服务本身的开机自行启动功能关闭,以免出现多个节点同时运行造成资源冲突(如:共享磁盘资源)。



1.6、安装的配置示例

现在我们引入一个三个节点的例子来说明具体的过程。在随后几章的安装、配置、使用等章节中的具体内容将参照如下图 1-1 拓扑结构的情况进行说明。

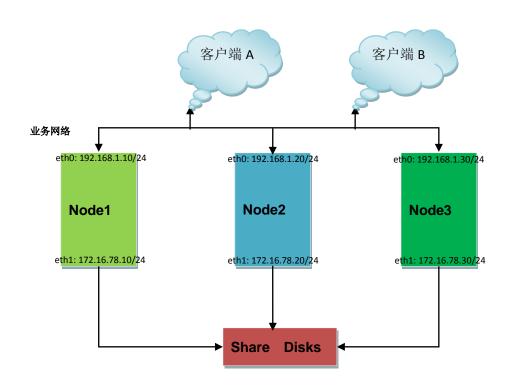


图 1-1 多节点集群架构

如上图 1-1 所示,有 3 台服务器,分别为 node1、node2 和 node3。假设每台机器有两个网卡 eth0、eth1,每台服务器上的 eth0(业务网络)连接至公网交换机,eth1 连接至内网交换机。此外,每台服务器通过光纤与共享的磁盘阵列相连。

说明:该图中的地址及结构可根据实际网络自行调整。

后续章节在配置时,除特别指出外,所有操作均需在每个节点上分别执行。



第二章 安装 LiteCloud & Skyge HA 8.0

在做好了第一章中所列的各项准备工作之后,现在就可以安装软件了。 安装过程如下:

- ❶安装 LiteCloud & Skyge HA 8.0
- ②注册 LiteCloud & Skyge HA 8.0

2.1、安装 LiteCloud & Skyge HA 8.0

我们需要以 root 身份在每个节点上安装 LiteCloud & Skyge HA 8.0。

插入安装光盘到节点的光驱并 mount 到任意目录后,光盘根目录中有一个安装文件 install_ha.sh,请运行如下命令进行安装。根据您的操作系统版本并参照其提示完成 LiteCloud & Skyge HA 8.0 的安装。如下图 2-1:

图 2-1 安装 Skyge HA 8.0

当输入 y 后,程序会显示检测到的 Linux 系统以及版本,如果输入 y 确认检测到的系统以及版本,则程序会安装相应的软件包。LiteCloud & Skyge HA 8.0 共包含六个软件包: jiluos、messagebus、youchai、buluos、skygeha 和 emon。



软件包安装后,会出现提示配置 LiteCloud & Skyge HA 8.0 的第三方参考 IP (可以输入一个或多个,多个之间请用:分隔。并且保证每个第三方 IP 当前皆可 ping 通)。

♪ 注意:第三方参考 IP 地址一般为网关地址。是当节点之间的所有心跳链路都断开后,每个节点用以判断自己的状态。如果 ping 不通第三方参考 IP 地址,则表示自己已经成为了"孤岛";如果可以 ping 的通第三方参考 IP 地址,那么认为自己的网络环境是正常的。

这里输入"192.168.1.1:192.168.1.254"后,完成整个安装。

如果在安装后,需要改变第三方参考 IP,请修改/etc/init.d/skyge 文件中的相应内容。 安装后会在系统的/etc/profile 文件中添加相应的 PATH 环境变量。主要用于输入命令 时无需再指定完整命令路径。

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

如果需要立即加载环境变量可重新登录控制台或在当前控制台中执行:

source /etc/profile



2.2、注册 LiteCloud & Skyge HA 8.0

安装完软件之后,您需要注册产品,也就是说需要安装软件的 license,以保证 LiteCloud & Skyge HA 8.0 正常运行。注册产品的步骤如下:

1)、获得每个节点的硬件号。

分别在每个节点上运行"/opt/jiluos/bin/jl_get_hw_code", 其输出信息如下:

/opt/jiluos/bin/jl_get_hw_code

eth0 : 000c2919d648 eth1 : 000c2919d652

仔细记录每台节点的硬件号,以便在随后的注册中使用。

- 2)、目前只能将上一步获得的输出信息发送到神州天勤的技术支持邮箱: <u>support@skyge.com</u>来获得 LiteCloud & Skyge HA 8.0 的软件 license。
- 3)、在获得每个节点的对应 license 文件后(文件名以. lic 为文件后缀),请将其复制到所有安装包目录中,license 文件名称须与软件包的名字相同,后缀为. lic。文件名如下:

/etc/jiluos/lic/jiluos.lic
/etc/messagebus/lic/messagebus.lic
/etc/youchai/lic/youchai.lic
/etc/buluos/lic/buluos.lic
/etc/skygeha_8/lic/skygeha.lic
/etc/emon/lic/emon.lic

至此,就可以开始以下的配置工作了。



第三章 配置 JILUOS

Jiluos 是 LiteCloud 的基础,是其他所有模块的依赖。它提供了集群内程序间的依赖 关系处理并可以提供依赖关系的接口。用户只需按照提供的编程接口编写程序,即可以完全 解决程序之间的依赖关系。Jiluos 可以保证只有程序的单实例在运行。

因此在所有配置进行之前,必须首先配置 JILUOS。配置主要使用命令 jl_config,该命令可以用于配置、查看、删除 jiluos 的信息。通过使用 - h 可获得 jl_config 命令的帮助信息:

3.1、配置 jiluos 心跳通道

配置 JILUOS 即配置 jiluos 的心跳通道,通过 jiluos 的心跳通道各节点可以进行相互及时的通讯,同时为其他模块的使用提供了一个统一的通讯平台。

每个节点配置时必须首先加载 jiluos. ko 驱动模块,在首次执行 jl_config 命令时系统会自动加载该模块。验证 jiluos. ko 模块加载成功的方法为:

```
# lsmod |grep jiluos

Module Size Used by
jiluos 143593 0
```

该模块也可以手工加载,手工加载方式为:

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

配置 jiluos 心跳通道既可以使用手工配置,也可以使用预先写好的配置文件进行配置。

3.1.1、手工配置 jiluos 心跳通道

手工配置 jiluos 心跳通道,使用如下命令(请用相应参数替换命令中的红色斜体字部分):



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

jl config 配置参数含义如下表 3-1:

参数	取值范围	含义
netid	1至 1024 的正整数	一个 netid 代表一个虚拟网络。所有 netid 相
		同的节点将作为一个LiteCloud整体进行工作。
domainid	1 至 64 的正整数	一个 domainid 对应一个计算机节点,为节点的
		唯一 id。因此在一个 netid 中的多个节点间,
		该值不能相同。
ethcard	为本机网卡名称	如: eth0、eth1、bond0 等
pathnumber	1或2(只支持两个通道)	每个通道可以对应使用 bonding 的 ethcard

表 3-1 jl config 配置参数含义

按照第一章的配置示例,分别进行三个节点的 jiluos 配置。例如它们的 netid 都为 1000, domainid 分别为 1、2、3,每个节点均有两条心跳通道,分别为 eth0 和 eth1。配置如下:

```
配置计算机节点 node1 的第一条 jiluos 心跳通道为:
[root@node1 ~]# jl config -n 1000 -d 1 -e eth0 -p 1
Configure Jiluos system successfully
配置计算机节点 node1 的第二条 jiluos 心跳通道为:
[root@node1 ~]# jl_config -n 1000 -d 1 -e eth1 -p 2
Configure Jiluos system successfully
配置计算机节点 node2 的第一条 jiluos 心跳通道为:
[root@node2 ~]# jl config -n 1000 -d 2 -e eth0 -p 1
Configure Jiluos system successfully
配置计算机节点 node2 的第二条 jiluos 心跳通道为:
[root@node2 ~]# jl config -n 1000 -d 2 -e eth1 -p 2
Configure Jiluos system successfully
配置计算机节点 node3 的第一条 jiluos 心跳通道为:
[root@node3 ~]# jl config -n 1000 -d 3 -e eth0 -p 1
Configure Jiluos system successfully
配置计算机节点 node3 的第二条 jiluos 心跳通道为:
[root@node3 ~]# jl config -n 1000 -d 3 -e eth1 -p 2
Configure Jiluos system successfully
```



3.1.2、使用配置文件配置 jiluos

可以使用预先写好的配置文件进行 jiluos 的心跳通道配置。该配置文件为/etc/jiluos/jiluos.cfg,内容如下(以node1节点的配置文件为例):

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

您可以从光盘中的 examples 目录下复制范例文件,请根据您的实际环境,对应修改该文件中等号后面的相应内容。准备好了 jiluos. cfg 配置文件,即可使用如下命令进行配置:

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

3.2、查看节点 jiluos 配置信息

配置完 jiluos 的心跳通道后,可通过 dmesg 或 ji_config -s 来验证或查看节点的配置信息。

在每个节点通过心跳通道建立连接后,使用 dmesg 命令可以看到如下类似信息:

```
配置节点 node3 的第一条 jiluos 心跳通道后,执行 dmesg 命令:
[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
```

配置完三台节点后,在 node1 上运行 jl_config -s 命令,可以看到如下类似信息:

```
[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
```

通过输出的 Scope 列,可以看到三个节点已经加入到该集群(cluster)中,且该运行命令的节点(node)为 domainid为 1的计算机。



上注意:上面输出的 Port Identity 及 Publication 列因各自的配置不同会生成不同的编号序列,该信息仅做为样例参考,不可用于完全比对。下文中若输出的相似信息,也同为样例,不再赘述。

3.3、删除 jiluos 配置信息

如果需要重新配置 jiluos 信息,请务必先删除之前配置的相应信息。删除命令为:

jl_config -r
Remove jiluos system cleanly



第四章 配置 MESSAGEBUS

4.1、MESSAGEBUS 简介

MESSAGEBUS (消息总线)提供了一种方便快捷的程序间通信的机制。它屏蔽掉了集群通信中复杂的地址机制,数据提供方和数据接收方都不需要复杂的地址、socket 等接口,使用起来更加高效,快捷,方便。每个节点将自己注册在消息总线服务上,无需知晓其他注册在总线上的节点对象。数据的提供方只需要向相关的服务上发消息,数据接收方订阅自己感兴趣的服务。消息总线服务进行消息的调度和转发,将消息发送给指定的对象,数据发送方发到相关服务的数据会被发送到任何订阅这个服务的数据接收方。

由于这种消息(Message),具有统一的格式,便于传递,可以更好的保证数据传递的完整性,可靠性,顺序性,一致性。消息可以是广播的形式,也可以是点对点的。同时消息服务可以将一大堆分布在不同物理机上的应用整合起来,进行通信,可以将一些小的应用整合为一个大的,可用的应用系统。

4.2、启动 MESSAGEBUS

若想创建新的消息总线或使用 Skyge HA 8.0,必须先要在每个集群节点上启动消息总线服务。后台运行消息总线服务的方法如下:

```
# mb message bus -d
```

消息总线服务启动后, 便可根据自己的需要创建相应的消息总线。

如果您只需配置 Skyge HA 8.0 服务,无需自己开发基于 LiteCloud 的应用,可跳过本章后续小节。

4.3、创建消息总线

如果您需要创建自己的消息总线,可使用 mb_cli_create_service 命令,该命令可以在任一台加入该同一 netid 中的节点上运行。例如,可以在任一节点上,为 domain 为 1 的节点创建一个新的总线 "Mytestmb",命令如下:

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

4.4、查看消息总线

使用 mb_cli_get_service 命令可以查看到指定节点上的所有总线,例如查看 domain 为 1 的节点上的总线:



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

每个新创建的总线均会在 messagebus 的配置目录中生成相应的目录和文件,因此也可通过查看生成的总线配置文件来验证,如:

```
# 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、向消息总线发消息

使用 mb_general_provider 命令可以向指定的消息总线发送消息。例如,向前面小节中 创建的 Mytestmb 总线发送消息 Hello:

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

4.6、接收指定的消息

LiteCloud 中的消息处理由 youchai 模块完成,它内置标准日志处理模块,可以进行日志的动态处理,也可以使用自定义脚本,通过总线来获取日志或程序信息。

使用 youchai 软件包中的 yc_msg_spy 命令可以在任一节点上接收(或侦听)指定节点的总线所发送的消息。例如,接收上一小节中发送的总线消息中的 data 部分:



4.7、删除消息总线

使用 mb_cli_delete_service 命令可以删除指定节点上的相应总线,例如删除 domain 为 1 的节点上的 Mytestmb 总线:

mb_cli_delete_service -d 1 -n "Mytestmb"

Delete service Mytestmb successfully.



第五章 配置 BULUOS

5.1、BULUOS 简介

BULUOS 在 LiteCloud 内提供了方便快捷的跨机器、跨进程的锁机制,便于应用程序之间进行协调工作。只有获得锁的节点才可以使用服务资源,其他节点要想获得该服务资源只能等待获得锁的节点释放锁。

可提供 2048 个 (0-2047) 锁槽位 (slot),每个槽位支持 8192 个 (0-8191) 锁,共提供 2048*8192 个分布式锁。它提供的不同种类锁模式为用户提供最大可能的灵活性。多种锁模式如下:

- ➤ random(随机锁): slot 范围为 0 511
- ▶ fifo (先进先出锁): slot 范围为 512 1023
- ▶ stack (后进先出锁): slot 范围为 1024 1535
- ▶ 用户自定义锁: slot 范围为 1536-2047

BULUOS 提供的可靠、高效的分布式锁机制,使得锁的相应时间压缩到 10ms 以内,大大提高了分布式应用程序的响应速度。同时提供集群计算中的选举机制,使用户不必再花费心思完成此类工作。

5.2、启动 BULUOS

如果该节点是首次启动 BULUOS,请首先进行 BULUOS 的初始化,以便生成相应的总线用于 BULUOS 的日志消息。初始化命令如下(此命令需在所有节点上运行):



```
# bl_init.sh
create service log.buluos.skyge.com 0 1 successfully.
log.buluos.skyge.com has been created
可以使用 cat 命令查看 bl init.sh 脚本的内容(实际为创建 log.buluos.skyge.com 总线):
# 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
```

启动 BULUOS 使用命令 buluos,通过-h选项可以获得该命令的帮助信息:

```
# 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
 -1, --log-level
                      Set log Level :
                         LOG EMERG
                         LOG_ALERT
                         LOG_CRIT
                         LOG ERR
                         LOG WARNING
                         LOG NOTICE
                         LOG_INFO
                         LOG_DEBUG
                      Set IPs, Which is checked to avoid network isolation
 -i, --ips
```



buluos 配置选项含义如下表 5-1:

选项	参数值	含义
-d	无	后台运行该服务
-t	console	日志类型可以选择将 buluos 日志输出至 console (控制台)
	syslog	syslog (/var/log/skygeha.log 日志)
	bus	bus (log.buluos.skygeha.com总线)
	all	all(包含前面所有位置)
-1	LOG_EMERG	记录的日志级别可以为 EMERG (紧急)
	LOG_ALERT	ALERT (报警)
	LOG_CRIT	CRIT (严重)
	LOG_ERR	ERR (错误)
	LOG_WARNING	WARNING (警示)
	LOG_NOTICE	NOTICE (通知)
	LOG_INFO	INFO (信息)
	LOG_DEBUG	DEBUG (调试)。其中 DEBUG 为最详细的日志级别。
-i	第三方参考 IP	一个或多个第三方参考 IP 地址。所有 IP 地址需要放入成对的双引号中,多个
		第三方 IP 地址间用冒号分隔。

表 5-1 buluos 配置选项含义

例如,在每个节点上后台启动 buluos,并且都需要将 DEBUG 的日志级别信息输出到所有 (all) 日志类型中,使用的第三方参考 ip 为 192. 168. 1. 1 和 192. 168. 1. 254。运行命令如下:

```
# buluos -d -t all -l LOG_DEBUG -i "192.168.1.1:192.168.1.254"

Buluos: Starting up.....
```

5.3、测试锁工具

为了更好的理解锁的工作方式,可以使用/opt/buluos/tests/test_d_lock 测试锁工具来模拟锁的获取与释放操作。执行该命令前,请务必先执行上一小节的启动 BULUOS 服务。

在任一节点上的多个控制台或多个节点(同一 netid)上先后输入如下命令进行锁测试:

```
# /opt/buluos/tests/test_d_lock -s 600 -1 100
其中-s 为槽位号 , -1 为锁编号。
```

除第一个启动 test_d_lock 命令的控制台会显示"XXX process get lock"表明 XXX 进程获得锁之外,其他控制台或节点均会处于等待锁状态。只有在获得锁的控制台释放锁(使用Ctrl+C)后,就会有另一个控制台或节点显示"XXX process get lock"。



第六章 配置 Skyge HA 8.0

6.1、Skyge HA 8.0 简介

Skyge HA 8.0 完全架构在 LiteCloud 所提供的基础设施上,因此与以往的 Skyge HA 产品相比,可以说是全新一代的高可用集群产品。它具有如下特点:

- ▶ 支持 1-64 台机器组成集群。只要有一台机器还在运行,集群中的服务就不会中断。
- ▶ 支持集群节点动态加载、卸载,随时扩充和缩减,按需组群。
- ▶ 服务可以在集群内部自由切换、转移、动态添加并启动、按需停止及动态删除。
- ➤ 系统感知时间从传统的 10-20 秒大幅缩减到 20ms 左右,切换性能缩减到 1 秒内。显著提高了系统的可用性。
- ➤ 环境监测机制能够提前预警系统潜在的故障。
- ▶ 提供丰富的开放接口和应用命令,用户可以灵活定制自己需要的服务和开发。
- 配置简单,使用灵活,提供处理接口便于对不同数据进行灵活处理。灵活的插件机制,容易扩展检测模块。
- > 对于现阶段的云计算中的核心部分提供强大的保护机制。

6.1.1、Skyge HA 8.0 的使用限制

- ▶ Skyge HA 8.0 目前只支持主流 Linux 平台的高可用。
- ➤ Skyge HA 8.0 暂不支持并行处理的应用。也就是说,不支持同一个应用在多个节点同时并发运行的应用,例如 Oracle RAC。
- ➤ 配置 Skyge HA 8.0 的计算机节点的心跳方式时,必须保证至少一条心跳通道正常工作。如果两台节点之间的所有心跳通道都发生故障而不能正常连通,有可能会导致 Skyge HA 8.0 发生裂脑(split-brain)。发生裂脑后, Skyge HA 8.0 有可能会导致用户的资源不一致。

6.1.2、Skyge HA 8.0 相关术语

相关术语见下表 6-1:

术语	描述
节点	安装运行 Skyge HA 8.0 软件的计算机
主节点	指服务运行所在的节点或服务器。



术语	描述
备节点	指还没有运行服务的节点或服务器。如果主节点发生任何故障,服务就会从主 节点切换到其中一个备节点上。这个备节点也就转变成主节点。
工作链路	也称为业务链路,指集群向外提供服务的链路,从服务器到对外交换机的链路
心跳链路	节点之间相互通信的通道,用以传递心跳信息,获得对方信息。也包含工作链路
第三方参考 IP 地址	当节点之间的所有心跳链路都断开后,每个节点用以判断自己的状态。如果 ping 不通第三方参考 IP 地址,则表示自己已经成为了"孤岛";如果可以 ping 的通第三方参考 IP 地址,那么认为自己的网络环境是正常的。第三方参考 IP 地址一般为网关地址
服务	指用户应用相关的一组资源的集合,包括用户应用的进程资源、磁盘资源、网 卡资源以及浮动 IP 资源等。服务可以是其中几种资源或者全部资源的组合。
浮动 IP 地址	也称为漂移 IP 地址、虚地址,即为客户端直接访问业务的 IP 地址。两个节点的本地 IP 对于客户端来说是透明的,浮动 IP 地址是由 HA 在启动业务的时候拉起的,之后它随着业务的切换而漂移到对方节点
裂脑	指 HA 的节点之间彼此失去了联系,但是单个节点的 HA 仍然运行正常发生裂脑的充要条件是: • HA 的节点之间的所有心跳通道都发生了故障,导致 HA 的节点间失去了任何联系 • 每个节点的 HA 软件正在正常运行 裂脑带来的直接后果是导致每个节点都会各自启动并运行同一个服务,竞争同一个服务的资源,这样有可能会导致资源(尤其是共享数据)被损坏
主备方式 (Active-Standby)	即一个节点做为主节点,对外提供服务;其它节点做为备节点,监控主节点,随时准备接管服务。
互备方式 (Active-Active)	指每个节点上都有服务(每个节点上运行的服务不同)在运行,此时每个节点 既是本节点上服务的主节点,又是其它节点上服务的备节点
服务的切换	是指当服务在一个节点上出现故障,首先需要在故障节点上停止该服务以释放 服务所占用的所有资源,然后在另外一个节点上启动该服务以保证服务的可用 的过程
服务切换时间	是指服务不用的时间,就是服务正在恢复之中但尚不可用的时间段。对于两个节点 node1 和 node2 来说,如果服务在 node1 上运行(node1 为主节点),当前出现故障的时刻为 T1,当 node1 出现故障时,开始在 node2 上启动该服务直到该服务可用的时刻为 T2,那么服务切换的时间为(T2-T1)。一般而言,服务切换的时间≈(约等于)检测到服务在 node1 上出现故障的时间+在 node1 上停止服务的时间+在 node2 上启动服务的时间。服务停止和启动时间视具体的应用而定,而检测到服务出现错误的时间可以通过对 Skyge HA 的配置来调整

表 6-1 Skyge HA 8.0 相关术语

6.2、Skyge HA 8.0 用户脚本

在配置 Skyge HA 8.0 的服务时,需要配置服务的用户脚本。用户脚本也就是启动、停止和检测服务状态的脚本,用来启动和停止服务相关的应用程序及检测服务的运行情况。

用户脚本可以使用服务自身提供的启停脚本(如:/etc/init.d/XXX),也可以根据应



用的实际情况来重新编写。如果需要用户脚本,那么在配置服务之前首先需要准备好用户脚本,并且测试用户脚本正确无误后方可开始配置服务。

6.2.1、用户脚本

用户脚本是一个 shell 的脚本,用来启动和停止服务相关的应用程序及检测服务的运行情况。一个 Skyge HA 8.0 服务只能配置一个用户脚本。

6.2.2、编写用户脚本

请参照下面的例子来编写用户脚本,安装光盘的 examples 目录下也有用户脚本的例子, 名字为 service_stat_stop_example. sh。

在编写用户脚本时,需要注意以下问题:

- ▶ 脚本的参数必须包含对输入参数 start、stop 及 status 的相应处理,在启动部分加入用户所有程序的启动命令,在停止部分加入用户所有程序的停止命令,在状态检测部分加入对用户所有程序的检测命令。也就是说脚本的格式必须是规定的格式。
- ▶ 请在脚本中加入 log 信息,以便发生故障时准确定位故障原因。
- ▶ 脚本执行成功时(包含启动成功、停止成功及检测所有程序状态为运行时)要明确返回 0,失败时(包含启动失败、停止失败及检测任一程序状态为停止时)要明确返回 1。 特别是当应用程序有标准的启动/停止脚本时,请确认该脚本在成功时是否返回 0,失 败时是否返回 1,如果不是,请不要直接使用应用程序提供的脚本,请手动编写脚本。
- ➤ 对于操作系统默认安装的一些应用的脚本,有些脚本在启动/停止时会判断是否已经启动/停止该服务了,在已经启动/停止时,该脚本返回值是 1,这不满足 Skyge HA 的接口,需要将其改成 0。当然,用户自己编写的脚本在判断服务是否已经启动/停止时,也应该满足返回值为 0 的条件。
- ▶ 如果启动和停止分别有多步操作,停止操作的执行顺序一定要和启动操作的执行顺序相反。
- ▶ 用户脚本要有可执行权限。



6.2.3、测试用户脚本

用户脚本写好后,请务必在每个节点上通过测试检查用户脚本是否正确。 例如,用户脚本的名字为 myapp. sh。

1)、启动

首先执行./myapp.sh start

然后 echo \$?, 判断脚本的返回值是否为 0, 0 表示脚本执行成功。

检查应用程序是否正常启动。

如果脚本的返回值为0并且应用程序正常启动了,表示用户脚本在启动时是正确的。

2)、停止

首先执行./myapp.sh stop

然后 echo \$?, 判断脚本的返回值是否为 0, 0 表示脚本执行成功。

检查应用程序是否正常停止。

如果脚本的返回值为0并且应用程序正常停止了,表示用户脚本在停止时是正确的。

3)、应用程序正常运行时

执行检测脚本 ./myapp.sh status

然后 echo \$?, 判断脚本的返回值是否为 0。

如果脚本的返回值为 0, 表示检测脚本是正确的。

4)、应用程序运行不正常时

执行检测脚本 ./myapp. sh status

然后 echo \$?, 判断脚本的返回值是否为 1。

如果脚本的返回值为1,表示检测脚本是正确的。



6.3、启动 Skyge HA 8.0

启动 Skyge HA 8.0 主要需要以下几个步骤:

- ➤ 初始化 Skyge HA 8.0
- ▶ 准备服务配置文件
- ▶ 设置服务接管顺序
- ▶ 启动 Skyge HA 的所有进程。

6.3.1、初始化 Skyge HA 8.0

如果该节点是首次启动 Skyge HA 8.0,请首先进行 Skyge HA 的初始化,以便生成相应的总线用于 Skyge HA 8.0 各类消息的传递。初始化命令如下(此命令需在所有节点上运行):

```
# 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
可以使用 cat 命令查看 ha init.sh 脚本的内容 (实际为创建多条 SkygeHA 总线):
# cat /opt/skygeha_8/bin/ha_init.sh
```

为了验证 Skyge HA 8.0 初始化成功,可以使用 messagebus 软件包中的 mb_cli_get_service 命令或检查生成的相应总线目录文件来查看指定节点上的所有总线,例如使用 mb_cli_get_service 命令查看 domain 为 1 的节点上的总线:



```
# mb_cli_get_service -d 1 -1
```

或通过查看生成的总线配置文件,如:

```
# ls -l /etc/messagebus/slots/
# cat /etc/messagebus/slots/X/service.cfg
其中 x 为相应的总线编号
```

6.3.2、准备服务配置文件

每一个 Skyge HA 8.0 服务需要有一个服务配置文件 (service.cfg)。您可以从光盘中的 examples 目录下复制范例文件。例如,以下为 mysql 服务使用的样例配置文件:

```
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;
```

该配置文件必须按照规定的格式编写。每行代表一个服务配置选项内容,由"选项=参数;"组成。如果该参数为数值型数据,可以不用使用双引号。允许在配置文件中使用的选项及含义如下表 6-2 (无需使用的选项行可以不填写或使用"#"开头将该行注释掉):

选项	参数含义	内容举例	说明
name	Skyge HA 的服务	mysql	必备行 。必须使用字母或数字,不支
	名称	apache	持中文及特殊含义字符(如: *、?、
		oracle	\、#等)
script	用户脚本的绝对	/etc/init.d/mysqld	必备行。请保证该用户脚本已符合本
	路径	/opt/skygeha_8/myapp.sh	章第二小节的要求
desc	描述信息	MySQL for Test	用于辅助说明
fip	浮动 IP 信息	eth0-192. 168. 1. 100-255.	每组信息包含网卡名、IP、掩码、本
		255. 255. 0-192. 168. 1. 255	网广播地址。此四项间使用"-"分
		:	隔,":"结尾。多组信息可连续填写。
store	挂载共享磁盘的	192. 168. 1. 253-/dev/sdb1	每组信息包含磁盘阵列 IP、挂载设
	相关信息	-/var/lib/mysql-755-mys	备、挂载目录、挂载权限、挂载用户、
		ql-mysql:	挂载组。此六项间使用"-"分隔,



50 III 0.0 D(XE)	N 1 /1/1	
		":"结尾。多组信息可连续填写。
服务启动超时时	100	能够接受的服务启动最大时间,在此
间(秒)		时间范围内没有成功启动即认为服
		务启动失败。默认值 300 秒
服务停止超时时	100	能够接受的服务启动或者停止的最
间(秒)		大时间,在此时间范围内没有成功停
		止,即认为服务停止失败。默认值
		300 秒
释放资源超时时	30	释放资源需要等待的最大时间。
间(秒)		
是否自动启动	yes 或 no	设置成 yes, 该服务将在启动
		sk_launcher 后立即启动。否则需要
		使用命令 ha_svc_commander 手工启
		动。默认值为 no
检查服务时间间	5	建议根据用户应用要求的切换时间
隔(秒)		设定合适的值。默认值3秒
检查服务失败次	2	是允许服务连续错误的次数。默认值
数		1次
强制加载	yes	多数服务无需使用该选项。有些服务
		必须使用(如:crond、snmpd、smartd、
		cups 等),否则将无法管理该服务。
强制重启	yes	当服务停止或释放资源失败后,可以
		选择将该节点重启系统,已保证释放
	服务启动超时时间(秒) 服务启动超时时间(秒) 解放(秒) 是否自动 检查(秒) 检查(秒) 检查(数) 强制加载	间(秒) 100 服务停止超时时间(秒) 30 释放资源超时时间(秒) yes或no 检查服务时间间隔(秒) 5 检查服务失败次数 2 强制加载 yes

表 6-2 服务配置文件格式说明

6.3.3、设置服务接管顺序

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、启动 Skyge HA 的所有进程

Skyge HA 8.0包含四个主要的服务进程:

- ▶ sk_manager 进程: HA 服务的管理进程。主要用于检查 HA 服务的当前启停状态,并向 sk launcher 进程发送启动或停止服务的命令。
- ▶ sk_launcher 进程:接收 sk_manager 的命令,负责服务的启停操作。启动后会生成相 应的子进程来监听每个用户服务,该子进程负责用户服务及资源锁的获取与释放。
- ➤ sk_checker 进程: 检查服务进程。根据设置的检查间隔(check_interval)定期检查服务状态(包括浮动 IP、用户服务进程等),并将检查结果及时通知 sk_manager。sk_manager 根据服务的状态再决定是否需要通知 sk_launcher 进行服务的启停操作。
- ➤ sk_safe_guard 进程: 监测监控用户服务的释放资源进程(sk_launcher 在启动用户服务后的相应子进程)。在释放资源进程被意外杀掉后,按照预先设置进行处理,防止发生用户服务无人接管的现象。

在每个节点上启动所有进程的命令如下(参数含义详见表 5-1):

```
# 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
```

其中的-i 选项为设置的第三方参考 IP。当节点之间的所有心跳链路都断开后,每个节点会使用它来判断自己的状态。如果 ping 不通所有的第三方参考 IP 地址,则表示自己已经成为了"孤岛"。若此时该节点正在运行用户服务,则该节点会根据脚本"/opt/skygeha_8/scripts/sk_ha_ck_network_isolation.sh"中的设置执行相应的操作(如: reboot。该脚本内容默认为空)。如果可以 ping 的通任一个第三方参考 IP 地址,那么认为自己的网络环境是正常的。

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
(节选,表明该节点已加载该浮动 IP)
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)
                         CMD
 PID TTY
                 TIME
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
               00:00:00
3721 pts/0
                         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、查看 Skyge HA 服务及状态

在服务进程启动后,可以使用 ha_query 命令查看指定节点配置的所有用户服务。例如, 在任一节点上查看 domainid 为 1 的节点上的所有用户服务(根据实际的配置的用户服务不同,输出可能不同,此处仅为举例):

```
# 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
```

使用ha status命令查看用户服务当前的运行节点(node1)及备份节点(node2和node3):



```
# 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、动态添加及删除服务

Skyge HA 支持动态的添加和删除服务的操作,使得添加和删除操作更加灵活方便。即在现有用户服务正在运行时,可以根据实际情况的需要随时增加新的用户服务,不会对当前正在运行的服务有任何影响。同时当用户服务不再需要时,也可在不影响其他正在运行的用户服务时停止并删除该用户服务。

动态添加及删除服务的命令为 ha_svc_commander,通过-h 选项可以看到该命令的帮助信息:

添加用户服务时,首先请参照本章第二、三小节的内容,将准备好的用户配置文件 (service.cfg)放入所有节点中/etc/skygeha_8/slots 的服务类型子目录(random 或 fifo 或 stack)下的编号目录中。

例如,在 domainid 为 1 的节点上,添加了一个编号为 4 的用户服务,该用户服务的服务类型选择为 stack,使用如下命令完成动态添加及手工启动服务操作:



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

如果该服务的配置文件为 onstart=yes,则该用户服务会立即在节点 1 启动。其他节点也可照此方法添加并启动服务,但备份节点会处于等待锁状态。

删除用户服务前,请先停止或释放该服务。例如,在 domainid 为 1 的节点(当前正在运行服务的节点)上,删除编号为 4 的 stack 类型的用户服务:

```
# 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、用户心跳功能

用户心跳是 Skyge HA 与用户应用服务软件之间通信所需的信息。它是指用户服务软件可以通过定时发送心跳信号给 Skyge HA,以便 Skyge HA 及时监测该服务的工作状态。当用户心跳停止时,证明用户服务已宕掉,Skyge HA 会切换服务到备用节点上。此种功能可以避免用户服务的假工作状态(表面在运行,内部服务已僵死)。该功能需要用户服务软件中增加向 Skyge HA 的发送心跳信号的程序代码段。

使用 sk_user_heartbeat 命令可以启动用户心跳检测功能,当 sk_user_heartbeat 进程在指定的时间内未接收到用户服务程序的心跳信号,即按照预设的操作(如停止本机服务以便切换)执行。用户心跳功能可以更好的保证用户程序的连续高可用。



第七章 配置 EMON

7.1、EMON 简介

EMON 主要是用来监控系统性能(如: CPU 占用率、内存占用率、硬盘使用量等)的软件。通过性能工具很容易及时观测到每个节点的设备工作状态,对合理调整、分配系统资源,提高系统整体性能起到至关重要的作用。

与 SkygeHA 相结合,EMON 还提供了主动切换技术。当用户设置了所要监控的设备达到指定阈值时执行的脚本后,在 EMON 检测到该设备达到了该预设值后,会根据对应脚本中的预设操作主动将当前节点上运行的服务停止并切换至其他节点。

同时 EMON 也给用户提供了程序接口,当 CPU、内存、硬盘等占用达到设定的数值后允许执行用户自己提供的程序来缓解系统的负载。如:设定了节点的硬盘分区占用量达到预设值后,会主动调用用户上传的程序,该程序可以对分区的数据进行备份、删除等等一系列操作。

7.2、初始化 EMON

如果该节点是首次启动 EMON,请首先进行 EMON 的初始化,以便生成相应的总线用于 EMON 监控消息的传递。初始化命令如下(此命令需在使用 EMON 的节点上运行);

```
# 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.

可以使用 cat 命令查看 em_init.sh 脚本的内容(实际为创建多条 emon 总线):

# cat /opt/emon/bin/em_init.sh
```

7.3、配置 ECS

EMON 的监控功能可以独立使用,也可以由其他软件调用。在启动 EMON 之前,需要首先配置 ECS(环境监控系统)。这里我们通过配置 Skyge HA 8.0 的 ECS 来使用 EMON 的监控功能。



7.3.1、查看 ECS 支持模块

通过如下方法,可以看出 EMON 系统目前仅支持三类性能监控: CPU 占用率、内存占用率和硬盘使用量。

```
# ls /etc/emon/services/
cpu_load disk_usage mem_load
(包含三个目录)

# 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
```

用户可以根据实际环境的需要, 按照规范开发自己的性能检测程序模块。

7.3.2、准备 ECS 执行脚本

ECS 执行脚本是指当所要监控的设备达到指定阈值时需要执行的操作脚本。根据实际需要,我们可在脚本中放入所需的操作命令(如:重启系统、停止某服务、删除临时文件等)。例如,准备一个在 cpu 达到 90%时需要执行 reboot 的脚本 testcpu90. sh,我们将它放入某个目录中(如:/etc/emon/cpu-mem-disk中),且该脚本必须有可执行权限。

```
# ls -l /opt/emon/cpu-mem-disk/testcpu90.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/testcpu90.sh
#!/bin/sh
reboot
```

7.3.3、准备 Skyge HA 的 ECS 配置文件

首先准备 Skyge HA 8.0 的 ecs 监控目录,目前仅支持 cpu_load、mem_load 和 disk_usage 三种监控信息。可以使用如下命令查看:



```
# 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)
```

性能监控服务配置文件 service. cfg 必须放于需要的监控目录中。该配置文件的主要作用是设置指定阈值及监控设备符合指定阈值时可以执行的脚本。例如,设置在 CPU 占用率达到 90%以上时,执行相应的 testcpu90. sh 脚本的样例如下:

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

ECS 配置文件格式说明, 见下表 7-1:

	7 111 12 27 2 2 1 1 1
选项行	格式说明(请使用实际值代替相应的红色斜体字部分)
watermarks 行	单个阈值为: watermarks = "wm_ <mark>XX</mark> ";
	或多个阈值为: watermarks = "wm_XX:wm_XY";
	只能有一行。在 cpu_load 和 mem_load 中,该 XX 为以整数表示的百分比
	占用率;在 disk_usage 中为已使用的根文件系统的磁盘使用量(以 KB
	计算)
	例如:设置 cpu 的占用率达到 90%时的阈值为: watermarks = "wm_90";
	或设置根系统的磁盘占用量达到约 250MB 时的阈值:
	watermarks = "wm_250000";
wm_XX 行	<pre>wm_XX = "script:pathtoscript:direction";</pre>
	可以有多行,每行指定一个阈值操作,必须与 watermarks 行中的参数对
	应。其中的 XX 为 watermarks 中设置的阈值,pathtoscript 为执行脚本
	的绝对路径, direction 的取值可以为 above(超过)、below(不足)、equal
	(等于)。

表 7-1 ECS 配置文件格式说明

7.3.4、启动动环监控进程

启动动环监控进程使用命令 sk dynamic ecs, 通过-d 选项可以使其后台运行:

```
# sk_dynamic_ecs -d -n cpu_load -t all -1 LOG_DEBUG
```

7.4、启动 EMON

启动 EMON 使用命令 emon, 通过-d 选项可以使其后台运行:

```
# emon -d
```

注:前台运行时,执行 emon 命令的控制台中会连续显示当前检测到的系统性能监控值。



第八章 卸载 LiteCloud & Skyge HA 8.0

当您请确实不再需要使用 LiteCloud & Skyge HA 8.0 软件时,请按照本章节内容完成 LiteCloud & Skyge HA 8.0 的卸载工作。

卸载 LiteCloud & Skyge HA 8.0 的过程如下:

- ●停止所有基于 LiteCloud & Skyge HA 8.0 的用户程序
- ❷停止 LiteCloud & Skyge HA 8.0
- ❸卸载 LiteCloud & Skyge HA 8.0

8.1、停止所有用户程序

在您卸载 LiteCloud & Skyge HA 8.0 之前,请确保在该节点上的用户服务已停止。如果用户服务正在该节点上运行,请先及时归档或备份相应的用户服务数据,然后可以使用ha_svc_commander 命令手工停止用户服务。例如:在 domainid 为 1 的节点上停止服务编号为 4 的 stack 类型的用户服务:

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

8.2、停止 LiteCloud & Skyge HA 8.0

停止 LiteCloud & Skyge HA 8.0 使用如下命令:

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

8.3、卸载 LiteCloud & Skyge HA 8.0

插入安装光盘到节点的光驱并 mount 到任意目录后,光盘根目录中有一个用于卸载的脚本文件 uninstall_lc. sh。请运行如下命令进行卸载:

```
# ./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.
```

注意:我们需要在所有节点上卸载LiteCloud & Skyge HA 8.0软件。



附录 LiteCloud & Skyge HA 8.0 的维护

1、Skyge HA 8.0 的运行状态

Skyge HA 8.0 安装配置完成后,可以通过以下的方式来检测 Skyge HA 8.0 的运行状态。

- ❶检查各守护进程的运行状态。
- ❷检查集群与服务的运行状态。
- ❸检查配置的 Skyge HA 服务。

1.1、检查各守护进程的运行状态

可以使用下列命令检查各守护进程的运行状态:

# ps -	A		
PID 5	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、检查集群与服务的运行状态

可以利用以下的命令检查 Skyge HA 8.0 的集群与服务的运行状态:

```
# 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
```

该命令可以显示出集群内所有用户服务的各服务的类型(random 或 fifo 或 stack)、服务编号、正在运行的节点 domainid 和所有等待接管的备份节点 domainid 信息。



1.3、检查配置的 Skyge HA 服务

可以用 ha_query 命令查看指定节点 (如: -d 1 代表 domainid 为 1) 上配置的 Skyge HA 服务。该命令允许在相同 netid 的任意节点上运行。

```
# 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 8.0 的日志信息

Skyge HA 8.0 的日志信息存放在每个节点的/var/log/skygeha.log 文件中,如果您的系统在使用过程中出现问题,您可以首先阅读两个节点的/var/log/skygeha.log 日志文件,通过日志可以初步定位一些基本的故障和原因。

注意:

- ➤ 在 Skyge HA 8.0 运行期间,请不要手动修改和删除两个节点上的/var/log/skygeha.log 文件,以免 log 信息丢失。
- ▶ 如果您发现 /var/log/skygeha.log 日志文件没有正常记录日志信息,可以执行 "/etc/init.d/syslog restart"(或 rsyslog 服务) 重新激活/var/log/skygeha.log 文件的日志功能。



3、FAQ

3.1、为什么会安装失败

在手工安装 LiteCloud & Skyge HA 8.0 软件包时,出现类似如下依赖提示导致安装失败:

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

此时请先行安装相应的 libconfig 安装包 (CentOS 或 RedHat 系统中),或 libconfig 及 libconfig—devel 安装包 (SUSE 系统中)。可以从安装光盘中找到相应操作系统版本的以 libconfig 开头所有的 RPM 包,然后通过"rpm—ivh libconfig*.rpm"来手工安装。

3.2、为什么无法执行命令

在执行命令时,出现如下提示。此时请检查 LiteCloud & Skyge HA 8.0 各软件的相应目录下 license 文件是否存在,并且测试用 license 文件没有过期。

-----License Expired------

Welcome to use products of Skyge Soft Co,.Ltd

Your license has expired or not valid.

Please visit our web site or contact with our company to get

commercial license.

www.skyge.com

3.3、为什么加载 jiluos 配置会失败

在执行 jiluos -c 时,出现 "err init cfg file"错误提示。此时请检查/etc/jiluos/jiluos.cfg 文件是否存在,并且该文件的格式和语法是否满足要求。



3.4、为什么 sk_manager 启动会失败

如果 sk_manager 进程启动失败,通过/var/log/skygeha 的日志中发现类似如下错误信息:

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

此时请检查/etc/skygeha_8/slots/fifo/XXX/service.cfg 文件是否存在,并且该文件的格式和语法是否满足要求。(注意: name 和 script 行为必须包含行)

4、技术支持

北京神州天勤软件技术有限公司衷心感谢您使用本公司集群软件产品。您的选择是我们 前进的动力,您的支持是我们成功的基石。您在使用此产品时,如发现问题或有什么建议, 请与我们联系,我们将不胜感谢。

电话: (010) 82359438

传真: (010) 82359428

网站: www.skyge.com

E-mail: support@skyge.com