

**GBASE<sup>®</sup>**  
让世界用上中国的数据库

# GBase 8s 连接管理器(CM)

# 前言：GBase 8s 的连接

- GBase 8s 的连接是指客户端(应用服务器) 到服务器端的连接方式。
  - 无集群：客户端(应用服务器) -> 服务器端
  - 有集群：客户端(应用服务器) -> CM -> 服务器端
- GBase 8s 连接的前提
  - 网络连接
  - 信任的网络主机
  - SQLHOSTS 原理
- 连接的手段
  - GBASEDBTSQLHOSTS 是控制连接的手段。
  - CM 主导集群连接的对象，优先权，故障转移

# 目录

一	网络连接
二	SQLHOSTS
三	连接管理器详解
四	连接管理器示例

# 网络连接(Linux)

## > /etc/hosts

- sqlhosts里面的主机名(hostname) 必须和/etc/hosts 里面的相一致。比如下列 node01, node02, node03

```
#/etc/hosts
node01      192.168.1.101
node02      192.168.1.102
node03      192.168.1.103
```

## > /etc/services

- GBASEDBTSQLHOSTS 里面的servicename或服务端口号,比如下列的9088 , 9089 必须和 /etc/services 里面的相一致.

```
# GBASEDBTSQLHOSTS
#dbservername nettype hostname servicename options
g_db           group      -           -           i=10,e=gbase03
gbase01        onsoctcp  node01      sqlexec     g=g_db
gbase02        onsoctcp  node02      sqlexec-ssl g=g_db
gbase03        onsoctcp  node03      9088        g=g_db
```

# 网络连接(Linux)

- › sqlhosts里面的servicename或服务端口号必须和 /etc/services 里面的相一致。比如下列
  - servicename: sqlexec, sqlexec-ssl
  - service端口号: 9088 , 9089

```
# /etc/services
sqlexec          9088/tcp
sqlexec          9088/udp
sqlexec-ssl     9089/tcp
sqlexec-ssl     9089/udp
```

```
# GBASEDBTSQLHOSTS
#dbservername  nettype  hostname  servicename  options
g_db           group    -         -            i=10,e=gbase03
gbase01       onsoctcp node01    sqlexec      g=g_db
gbase02       onsoctcp node02    sqlexec-ssl  g=g_db
gbase03       onsoctcp node03    9088         g=g_db
```

# 网络连接(Linux) – 防火墙设置

## › 方法一：关掉防火墙

- 使用root用户执行命令：`/sbin/service iptables stop`

## › 方法二：打开数据库实例连接的端口

以root用户登录

- 1) 修改`/etc/sysconfig/iptables` 文件，添加以下内容：

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9089 -j ACCEPT
```

注：该行应位于最后一行的前面(如下所示)，9089为数据库实例监听的端口

- 2) 重启防火墙

```
service iptables restart
```

```
# /etc/sysconfig/iptables (RHEL6)
```

```
.....
```

```
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9089 -j ACCEPT
```

```
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
```

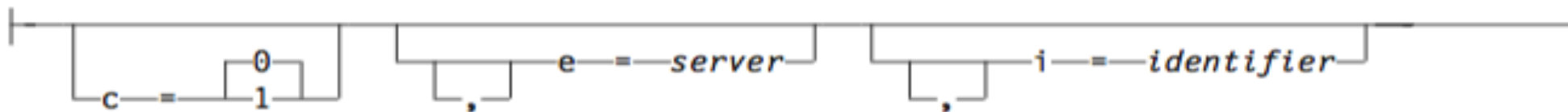
# 目录

一	网络连接
二	SQLHOSTS
三	连接管理器详解
四	连接管理器示例

# SQLHOSTS组

- › **GBASEDBTSQLHOSTS 环境变量决定 每一个客户端工具的连接。**
- › 改变 GBASEDBTSQLHOSTS 环境变量决定连接的方向
  - c=重新连接的是顺序连接，还是随机连接
  - e=数据库组中的最后一个数据库节点；比如最后一个组是 gbase03
  - i=数据库组的独立编号；比如 g\_db 的编号是 10.

## Group options:

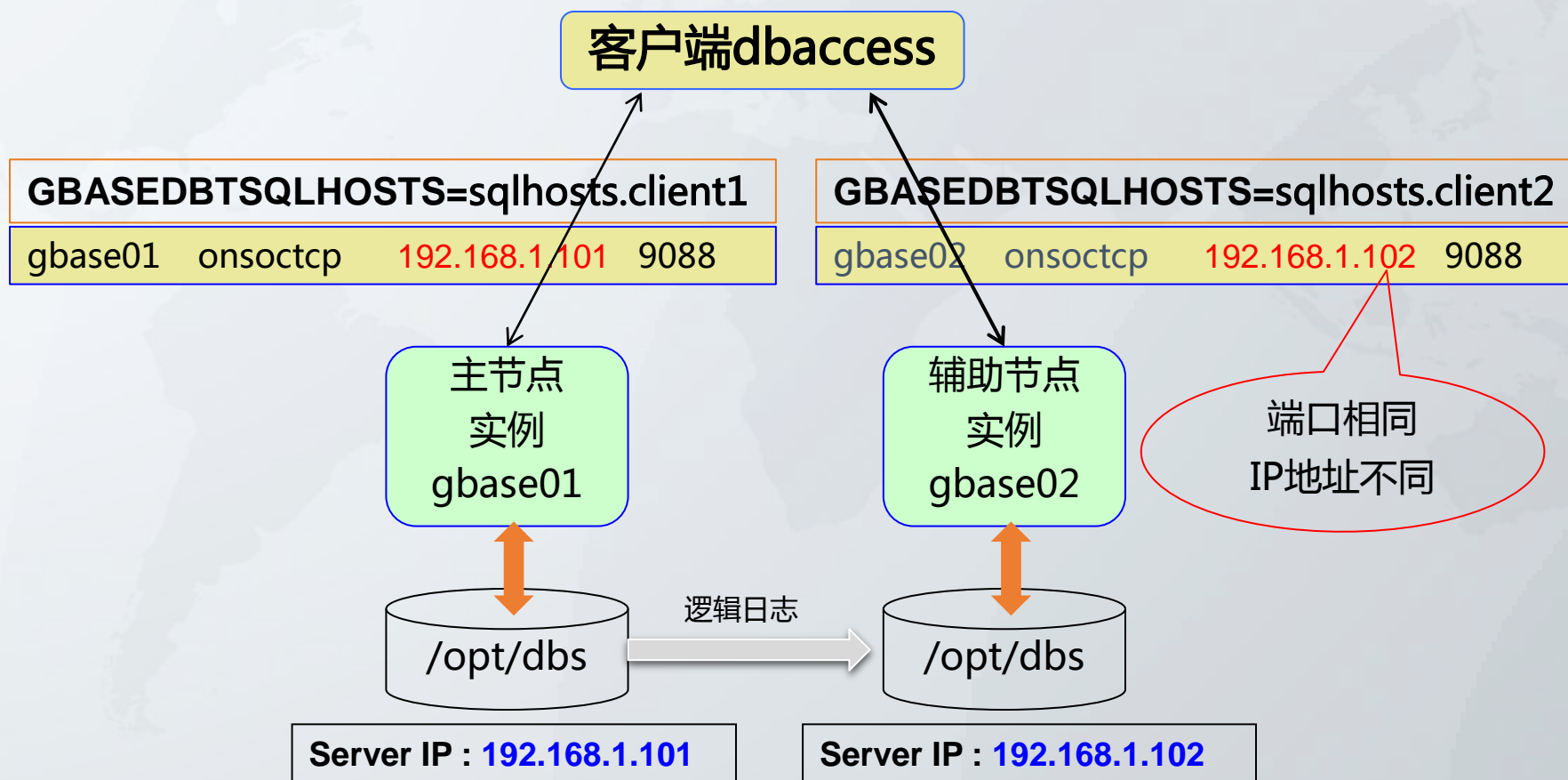


```
# GBASEDBTSQLHOSTS
#dbservername nettype hostname servicename options
g_db           group      -          -          i=10,e=gbase03
gbase01        onsoctcp node01     sqlexec    g=g_db
gbase02        onsoctcp node02     sqlexec-ssl g=g_db
gbase03        onsoctcp node03     9088       g=g_db
gbasedbt       onsoctcp node03t    9088
```



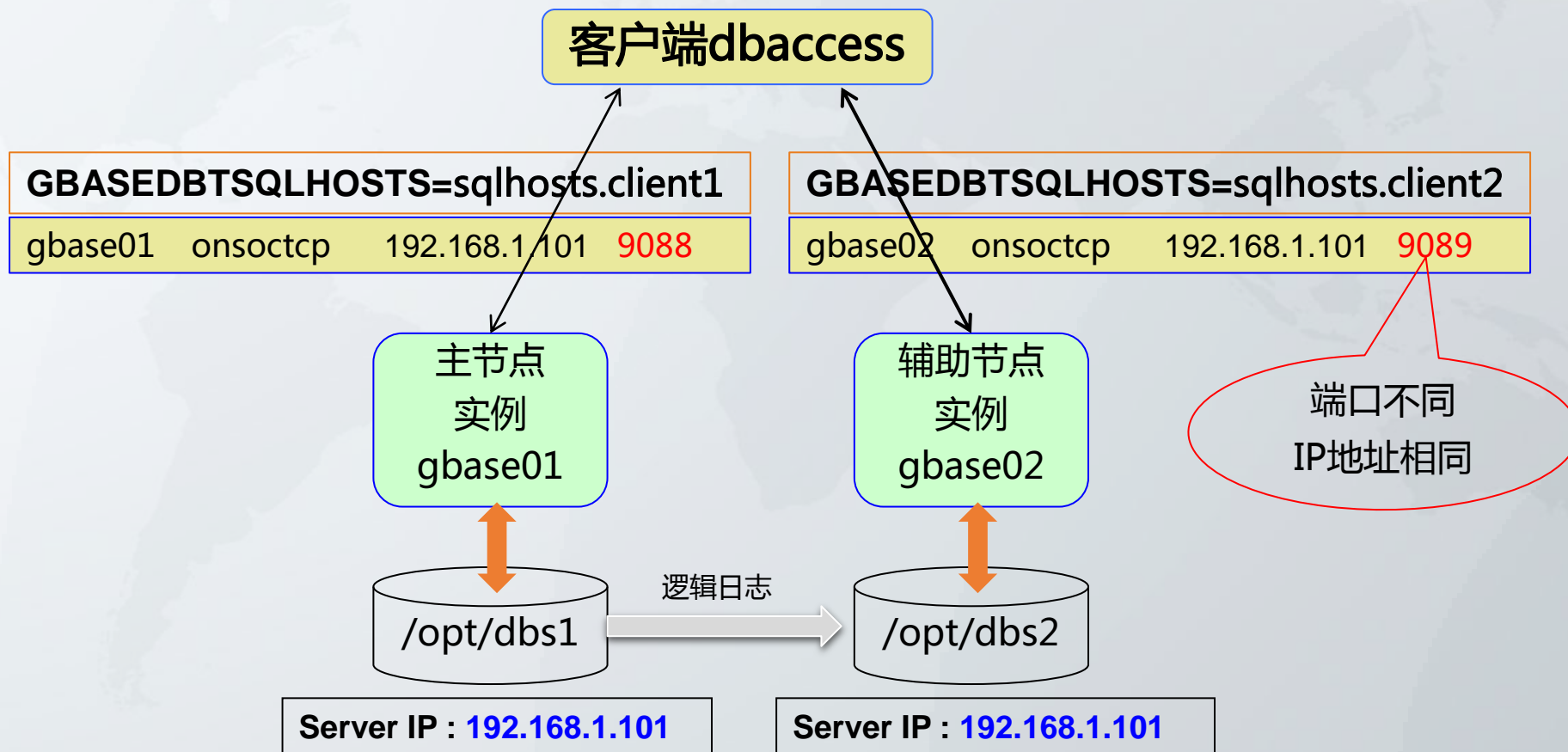
# SQLHOSTS - 连接 Client/Server

- › GBASEDBTSQLHOSTS环境变数 决定客户端连接不同的实例或集群
- › 不同的 sqlhosts 让客户端如 dbaccess 作不同的连接



# SQLHOSTS - 连接 Client/Server

- › 客户端连接同服务器的两个的实例
- › Port 端口号，9088 & 9089 是区分两实例的方法



# SQLHOSTS – 连接示例

```
# GBASEDBTSQLHOSTS
#dbservername nettype hostname servicename options
g_db group - - i=10,e=gbase02
gbase01 onsoctcp 192.168.1.101 9088 g=g_db
gbase02 onsoctcp 192.168.1.102 9088 g=g_db
```

## 连接示例

- dbaccess sysmaster@gbase01 /\* 主节点实例名 \*/
- dbaccess sysmaster@gbase02 /\* 辅助节点实例名 \*/
- dbaccess sysmaster@g\_db /\* 实例的群组名 \*/

```
CONNECTION: Connect Disconnect Exit
Connect to a database environment.
```

```
----- sysmaster@gbase01----- Press CTRL-W for Help -----
```

连接后以取得  
数据库服务器  
数据库@实例名

# 目录

一	网络连接
二	SQLHOSTS
三	连接管理器详解
四	连接管理器示例

# 连接管理器 ( CM ) 概述

## › 什么是连接管理器 ( Connection Manager )

- 一个用于管理客户端对GBase 8s 进行连接访问的组件
- CM的设置 决定客户端的连接顺序与方向
- CM的设置决定GBase 8s高可用集群中
  - 主节点到HAC节点的 故障切换(Failover)
  - 主节点角色的 转移到辅节点服务器
  - SSC高可用集群中 巨量客户端的负载均衡的功能
- CM 配置的流向契约(SLA) , 决定客户端在 GBase 8s高可用集群中的连接顺序
- 在安装包中或CSDK中安装使用
- 可以单独运行或者在一个服务器组中运行

# 连接管理器的五种角色

- ▶ GBase 8s 连接管理器用来监视和维护客户机连接，以及将客户机连接请求重定向或者以代理的方式到连接单元中的适当服务器。
- ▶ 连接管理器执行五种角色：
  1. 流向协议(SLA)服务等级协议
    - ▶ Server-Level Agreement，数据库服务器的
  2. 自动故障转移
    - ▶ Automatic failover for database servers
  3. 基于规则的连接重定向和**负载均衡**
    - ▶ Rule-based connection redirection and load balancing
  4. **代理服务器**连接管理
    - ▶ Proxy-server connection management
  5. 应用程序服务器的故障转移优先级
    - ▶ Failover prioritization for application servers

# 连接管理器的配置文件

- ▶ 连接管理器程序配置为使用 GBASEDBTSQLHOSTS文件，配置方法与 GBase 8s 数据库服务器相同
- ▶ 相关的配置文件：
  - GBASEDBTSQLHOSTS配置文件
    - 配置相关数据库服务器组信息，SLA，以及监听。
  - CMCONFIG配置文件
    - \$ GBASEDBTDIR/etc/cmsm.cfg.sample示例配置文件，有多种版本
  - CMCONFIG配置文件中指定的\$ CMALARMPROGRAM（可选）
    - 设置在CM出现故障时，触发的事件

# 连接管理器工作原理

## › 连接管理器CM

- 根据CMCONFIG 的配置参数及GBASEDBTSQLHOSTS中的SLA定义来发现需要连接的集群。

## › CMCONFIG 配置参数

- 来源：oncmsm 的配置文件或命令行参数
- 关键内容
  - CLUSTER：定义一个HA集群
  - GBASEDBTSERVER：从属于该集群的Server或Servers Group
  - SLA：该集群中定义的路由信息
  - FOC：故障转移的规则
  - SQLHOSTS：指定使用GBASEDBTSQLHOSTS配置文件的范围

## › GBASEDBTSQLHOSTS

- 包含HA集群中的所有Server信息、Servers Group和SLA信息
- 通过环境变量定义



# 连接管理器配置文件CMCONFIG

```
# CMCONFIG
NAME CM1 /* CM 名称*/
LOG 1 /* CM 是否有LOG, 1=有*/
LOGFILE $GBASEBTDIR/tmp/cm1.log /* CM LOG 所在*/
CM_TIMEOUT 30 /* 多个 CM 之间的等待时间, 缺省=60s*/
EVENT_TIMEOUT 30 /* CM 故障转移等待, 缺省=60s */
/* 适用在断电, 断网时的等待时间 */
SECONDARY_EVENT_TIMEOUT 30 /* 辅助节点不相应的等待, 缺省=60s */
SQLHOSTS LOCAL /* 本地或原地 local + remote */
LOCAL_IP 192.0.2.0,192.0.2.1 /* CM 连接的IP, 适用双网卡*/
CLUSTER g_db /* 数据库的群组名 */
{
    GBASEBTSERVER g_db
    SLA oltp_1 DBSERVERS=primary
    SLA payroll_1 DBSERVERS=(PRI,HDR) POLICY=WORKLOAD
    SLA report_1 DBSERVERS=(SDS,HDR) POLICY=WORKLOAD
    FOC ORDER=ENABLED PRIORITY=1 /*故障切换(失效转移)控制 */
    CMALARMPROGRAM $GBASEBTDIR/etc/cmalarmprogram.sh
}
```

# CMCONFIG配置文件中参数解释

- › NAME：当前CM的名称
- › LOG
  - 指定CM的Log模式
  - 取值：0-不记录日志；1-记录所有日志；2-只记录CM与节点间数据传递的活动；3-只记录CM与节点间传递的数据。
  - 日志记录到 LOGFILE 中
- › LOGFILE：当前CM的日志文件
- › SQLHOSTS
  - 指定CM在什么位置搜索GBASEDBTSERVER指定的服务
  - 取值包括LOCAL、REMOTE和LOCAL+REMOTE
- › LOCAL\_IP
  - CM所在机器的IP地址，该IP所在网络为HA集群心跳网络
  - 当CM在应用管理器上并具备双网卡
  - LOCAL\_IP 192.0.2.2, 192.0.2.3 /\* 代表两个网卡 \*/

# CMCONFIG配置文件中参数解释 ( cont. )

## › DEBUG

- 指定CM是否记录SQL或ESQL/C的错误信息，0~9
- 错误信息记录到 LOGFILE 中

## › CM\_TIMEOUT

- 指定高可用集群等待仲裁者身份的CM事件的超时时间，单位为秒
- 如果超过这个时间，则当前高可用集群的Primary节点会把当前高可用集群中PRIORITY最小的CM设置为仲裁者

## › CLUSTER

- 指定当前高可用集群的名称

# CMCONFIG配置文件中参数解释 ( cont. )

当主节点失联时CM或是等待超时，CM 以此参数响应。

适用于 1)主节点失联, 2)网络断线, 3)拔电源

## › EVENT\_TIMEOUT

- 指定CM等待高可用集群中主节点的超时时间，单位为秒
- 如果CM在该参数指定的超时时间内未收到与主节点相关的事件，则CM认为主节点OFFLINE并开始进行故障切换。
- 该参数值设置为-1时，表示CM无限期等待；0~30之间则等待30秒；大于30时等待该参数值指定的秒数。

## › SECONDARY\_EVENT\_TIMEOUT

- 指定CM等待高可用集群中辅节点的等待超时，单位为秒
- 如果CM在该参数指定的超时时间内未收到来自于辅节点的事件，则CM认为该辅节点从当前高可用集群中断开
- 该参数值设置为-1时，表示CM无限期等待；0~30之间则等待30秒；大于30时等待该参数值指定的秒数。

# CMCONFIG配置文件中参数解释 ( cont. )

CLUSTER 数据库的群组内的定义

## ➤ GBASEDBTSERVER

- 指 sqlhosts 中某个 group\_name

## ➤ FOC

- Fail-Over Control , 故障切换 ( 失效转移 ) 控制

- 定义出现故障时的应对策略

- PRIORITY

- 指定故障转移规则, 并指定当前CM与主节点连接的优先级, PRIORITY用于指定当前CM与主节点连接的优先级, 越小越优先。

- TIMEOUT

- 用于指定EVENT\_TIMEOUT超时后还需要多少秒CM才开始进行故障转移, 如果不指定, 则EVENT\_TIMEOUT超时后CM立即开始故障转移。

# CMCONFIG配置文件中参数解释 ( cont. )

CLUSTER 数据库的群组内的定义

## › SLA

- 基于规则的连接定向
- Server-Level Agreement , 服务等级协议
- 本意是关于网络服务供应商和客户间的一份协议，其中定义了服务类型、服务质量和客户付款等术语。

## › POLICY

- 负载均衡的参数 – POUNDROBIN , WORKLOAD
- 本意是把某类任务分摊到多个操作单元上执行
- 目的：增加吞吐量、改进系统性能、提升客户体验

## › MODE

- REDIRECT : 重定向模式 ( 默认 )
- PROXY : 代理模式

# 流向协议SLA

## › 基于SLA的连接管理

- 在CM中定义多个SLA
- 不同的SLA请求可以被路由到不同的节点——连接重定向
- SLA定义包括：名称、SERVER类型、POLICY、USEALIAS等

例如：

SLA oltp	DBSERVERS=PRI
SLA report	DBSERVERS=HDR,PRI
SLA query	DBSERVERS=(SDS,HDR) POLICY=WORKLOAD
SLA query2	DBSERVERS=(SDS+HDR) POLICY=ROUNDROBIN

1. oltp 使用 oltp的所有连接请求都会被路由给 PRI节点
2. report 所有连接请求会优先路由给HAC节点，之后是PRI节点
3. query 连接请求由 WORKLOAD策略决定 路由给SSC节点或HAC节点
4. query2 连接请求ROUNDROBIN策略决定 路由给SSC节点和HAC节点。

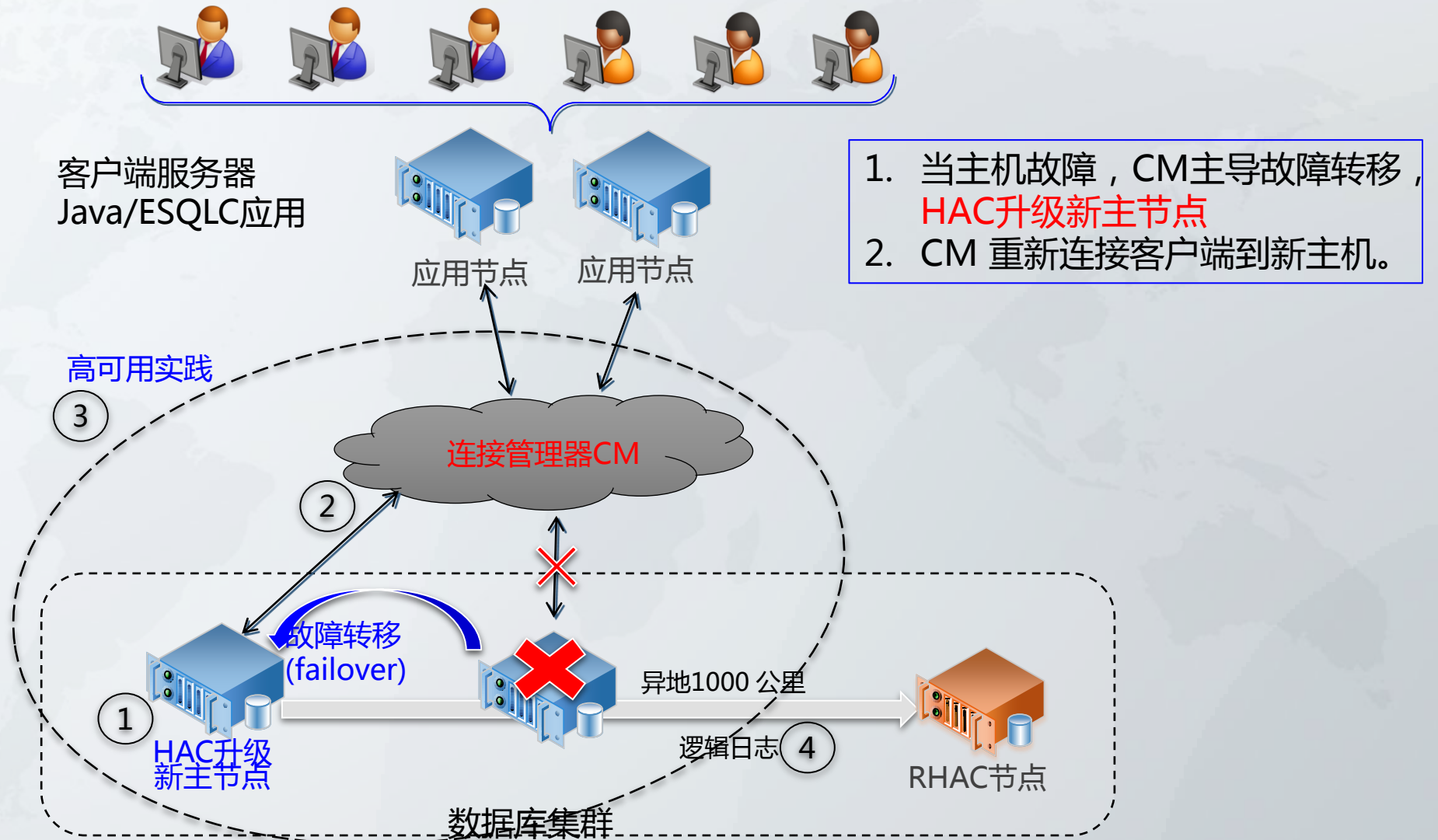
# 故障转移FOC

## 故障转移

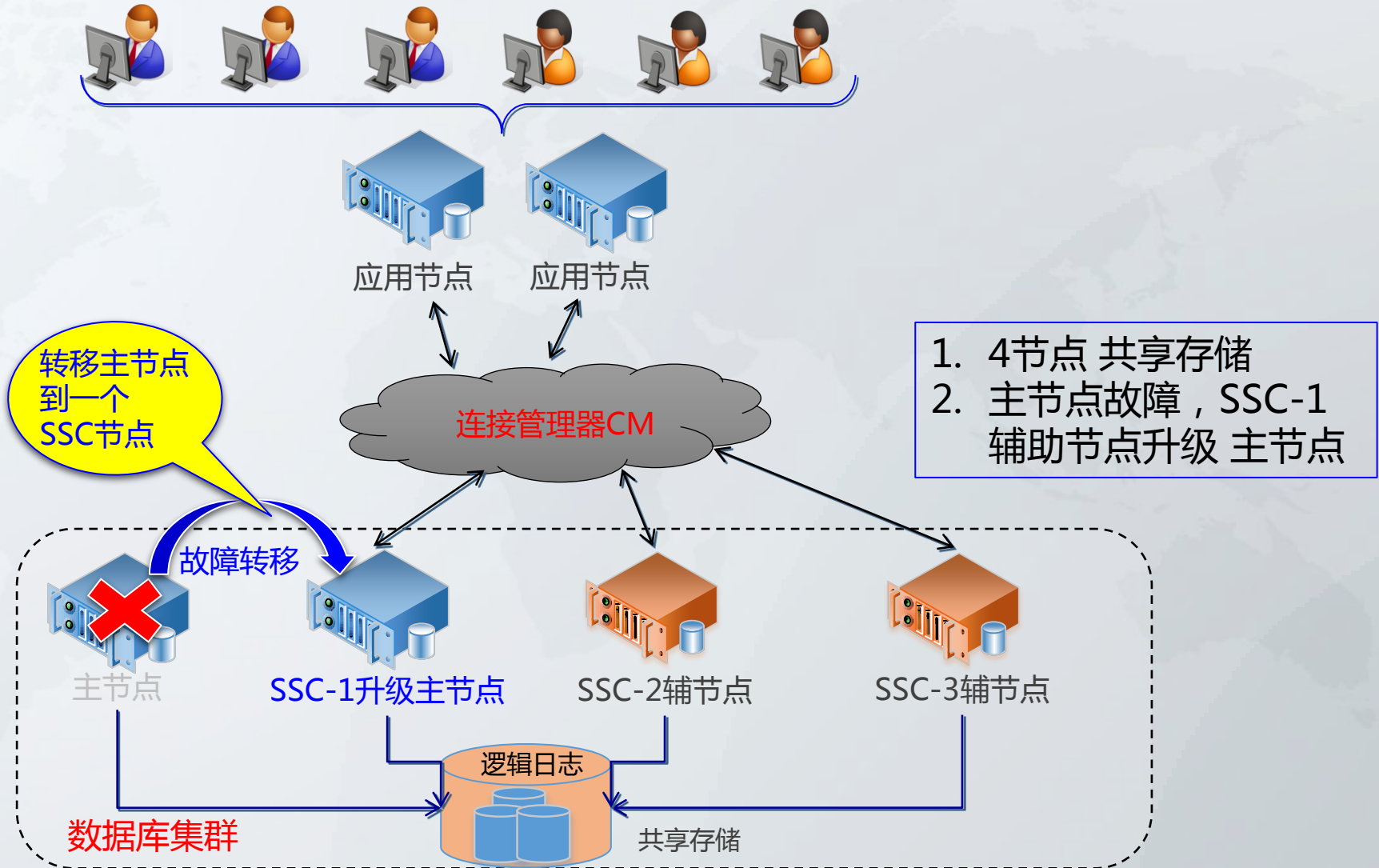
- CM 发现高可用集群中Primary节点不可用后，会根据既定的规则把集群中某个辅节点切换为Primary节点
- 故障转移规则
  - FOC ORDER= *ENABLED/DISABLED* PRORITY= *n* TIMEOUT= *n*
  - ORDER
    - ◆ DISABLED，禁用故障转移
    - ◆ ENABLED，启用故障转移
    - ◆ Primary节点 ONCONFIG配置文件中 HA\_FOC\_ORDER 参数设置了故障转移顺序，则使用该顺序
    - ◆ 如果HA\_FOC\_ORDER参数未设置，则故障切换顺序为SSC->HAC->RHAC(默认配置)
    - ◆ 如果禁用故障转移规则，则只能手工切换
  - PRORITY 指定处理故障转换的CM的优先级，1为最高
  - TIMEOUT 指定开始故障转移前等待的秒数



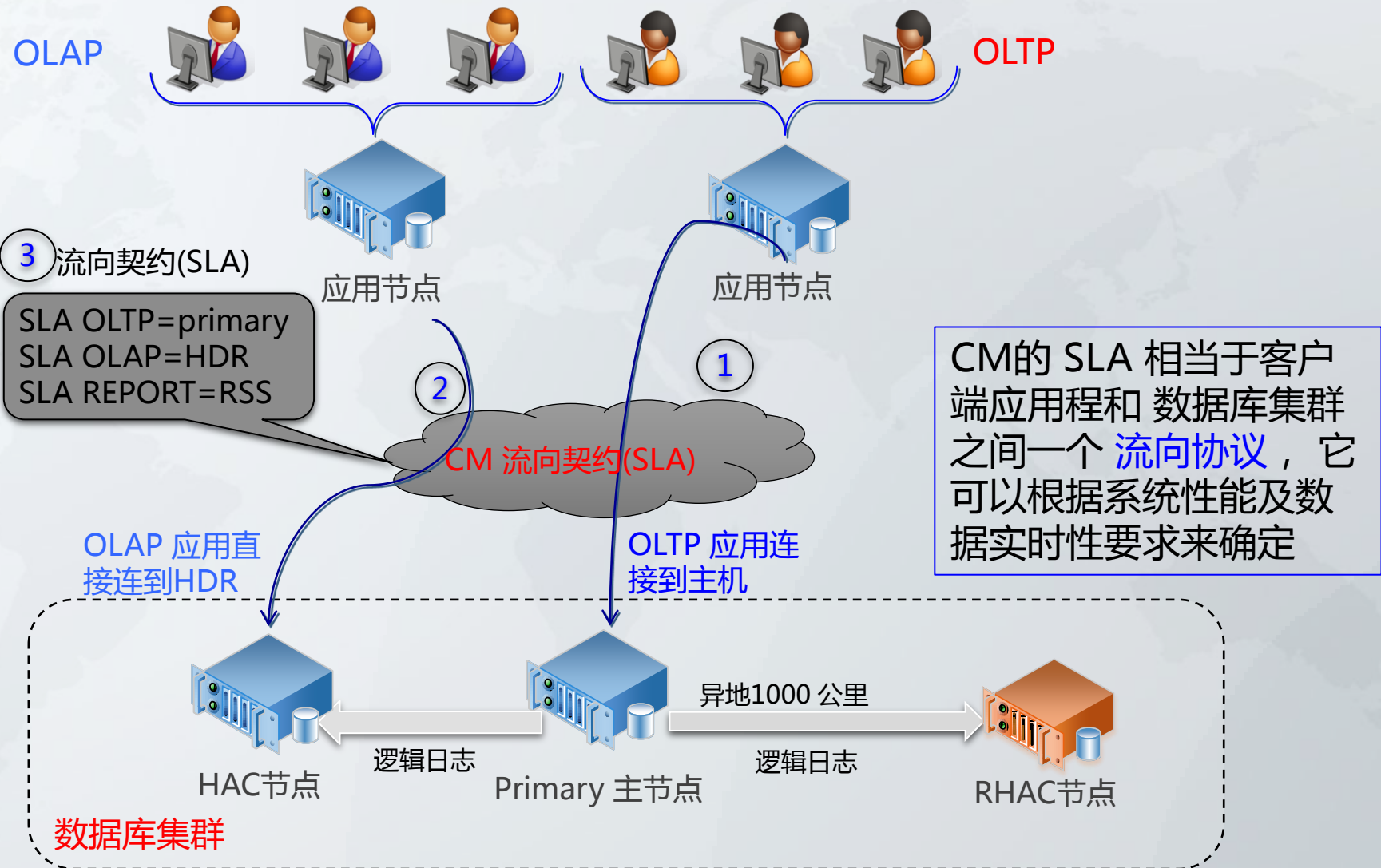
# 数据库集群 – HAC高可用



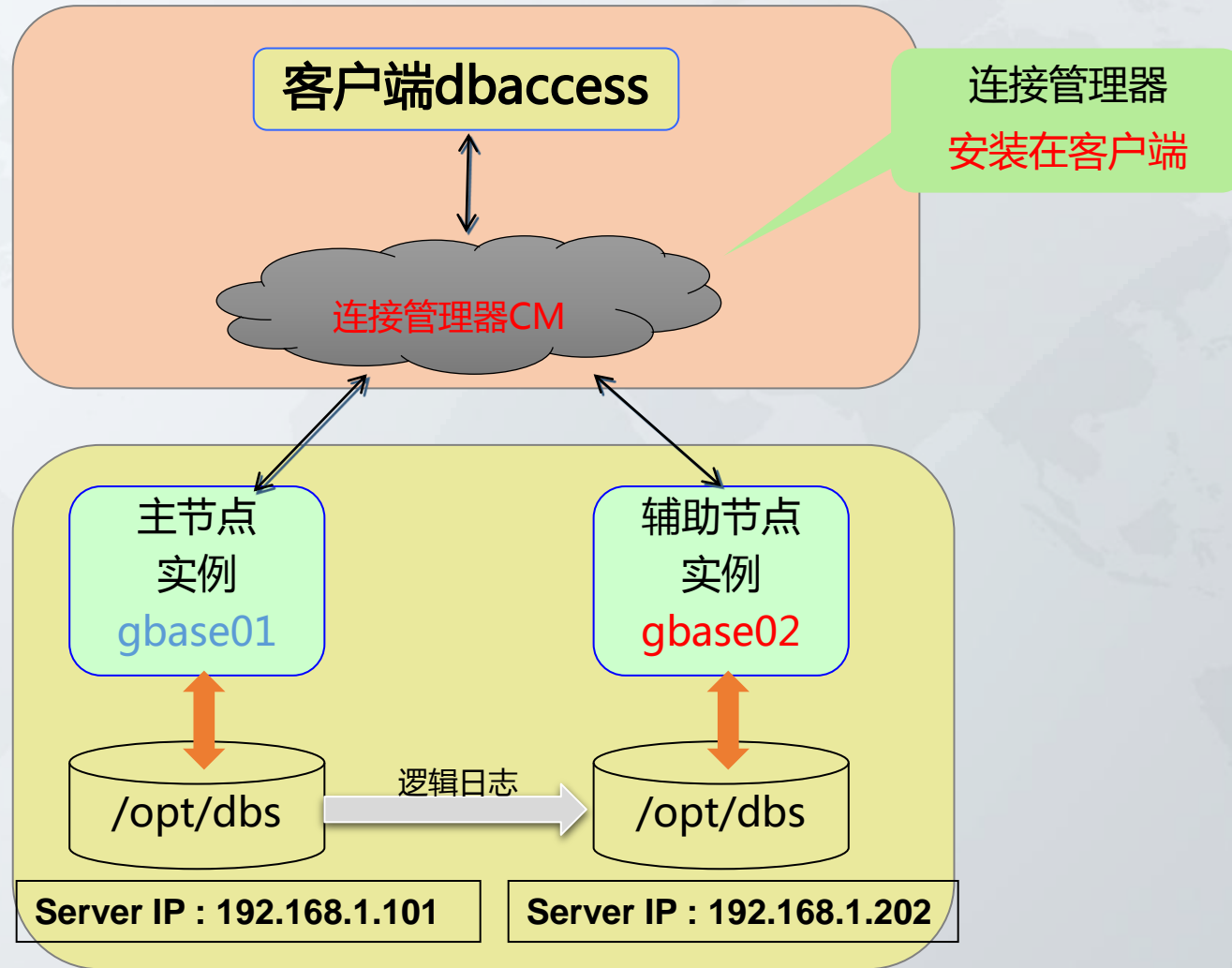
# 数据库集群 – SSC共享存储高可用



# 数据库集群 – SLA流向协议

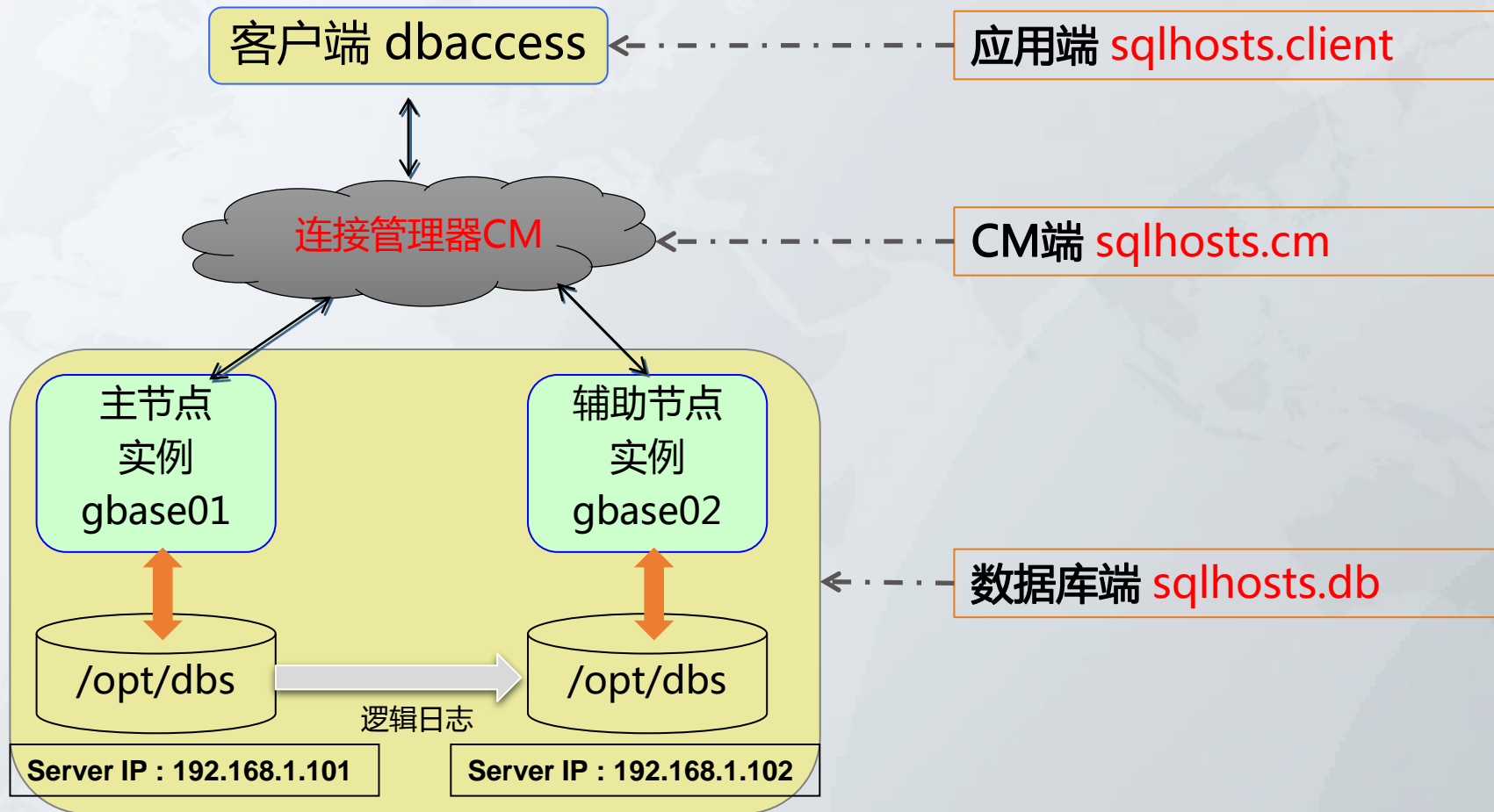


# 连接管理器CM - 存在客户端



# 高可用环境 - 使用各自的 SQLHOSTS

应用端，CM端，数据库端 有各自GBASEDBTSQLHOSTS环境变量



# 独立的 SQLHOSTS配置文件

› 客户应用端，CM端，数据库端有各自GBASEDBTSQLHOSTS环境变量

## # 应用端 sqlhosts.client

oltp_sla	group	-	-	i=1,e=oltp2_sla
oltp1_sla	onsoctcp	cm1server	cmport1	g=oltp1_sla
oltp2_sla	onsoctcp	cm2server	cmport2	g=oltp2_sla

## # CM端(节点1) sqlhosts.cm

g_db	group	-	-	i=10,e=gbase02
gbase01	onsoctcp	node01	9088	g=g_db
gbase02	onsoctcp	node02	9088	g=g_db

oltp_sla	group	-	-	i=1,e=oltp2_sla
oltp1_sla	onsoctcp	cm1server	cmport1	g=oltp1_sla
oltp2_sla	onsoctcp	cm2server	cmport2	g=oltp2_sla

report_sla	group	-	-	i=2,e=report2_sla
report1_sla	onsoctcp	cm1server	cmport1	g=report_sla
report2_sla	onsoctcp	cm2server	cmport2	g=report2_sla

## # 数据库端 sqlhosts.db

g_db	group	-	-	i=10,e=gbase02
gbase01	onsoctcp	node01	9088	g=g_db
gbase02	onsoctcp	node02	9088	g=g_db

# 目录

一	网络连接
二	SQLHOSTS
三	连接管理器详解
四	连接管理器示例

# 连接管理器示例 – 环境变量

## 连接管理器的环境变量

```
export GBASEDBTSQLHOSTS=$GBASEDBTDIR/etc/sqlhosts.sdscm  
export GBASEDBTSERVER=g_db
```

```
# sqlhosts.sdscm
```

```
g_db      group      -          -          i=10,e=gbase02  
gbase01   onsoctcp    192.168.1.101  9088  g=g_db  
gbase02   onsoctcp    192.168.1.101  9089  g=g_db  
  
oltp_sla  onsoctcp    192.168.1.101  18888  
report_sla onsoctcp    192.168.1.101  17777
```

## CM 的 sqlhosts.sdscm 给出 5 个连接方式

- g\_db - 直接连接到组
- gbase01,gbase02 – 直接连接到数据库服务器gbase01,gbase02
- oltp\_sla,report\_sla – 经由CM才连接到数据库集群中的数据库服务器



# 连接管理器示例 – 设定

- ▶ 在这个 CM 的范例中，首先确认 CM 本身的环境变量
  - GBASEDBTSQLHOSTS=\$GBASEDBTDIR/etc/sqlhosts.sdscm
  - GBASEDBTSERVER=g\_db
- ▶ GBASEDBTSQLHOSTS=sqlhosts.sdscm 是 CM 的连接配置文件，包括“数据库集群”的与自己本身的SLA连接的位置。
  - GBASEDBTSERVER是CM 所代表“数据库集群”的名称：g\_db。
  - GBASEDBTSERVER也同时定义在CMCONFIG: sdscm.cfg 配置文件中。
  - sqlhosts.sdscm 内的CM 连接名称: oltp\_sla 与report\_sla 同时也是定义在CMCONFIG 配置文件中。
  - sqlhosts.sdscm 与 sdscm.cfg 两个配置文件串起 CM 与 “数据库集群” 之间的关系。

# 连接管理器示例 - 配置文件

```
# GBASEDBTSERVER
g_db          group    -          -          i=10,e=gbase02
gbase01      onsoctcp  192.168.1.101  9088      g=g_db
gbase02      onsoctcp  192.168.1.101  9089      g=g_db

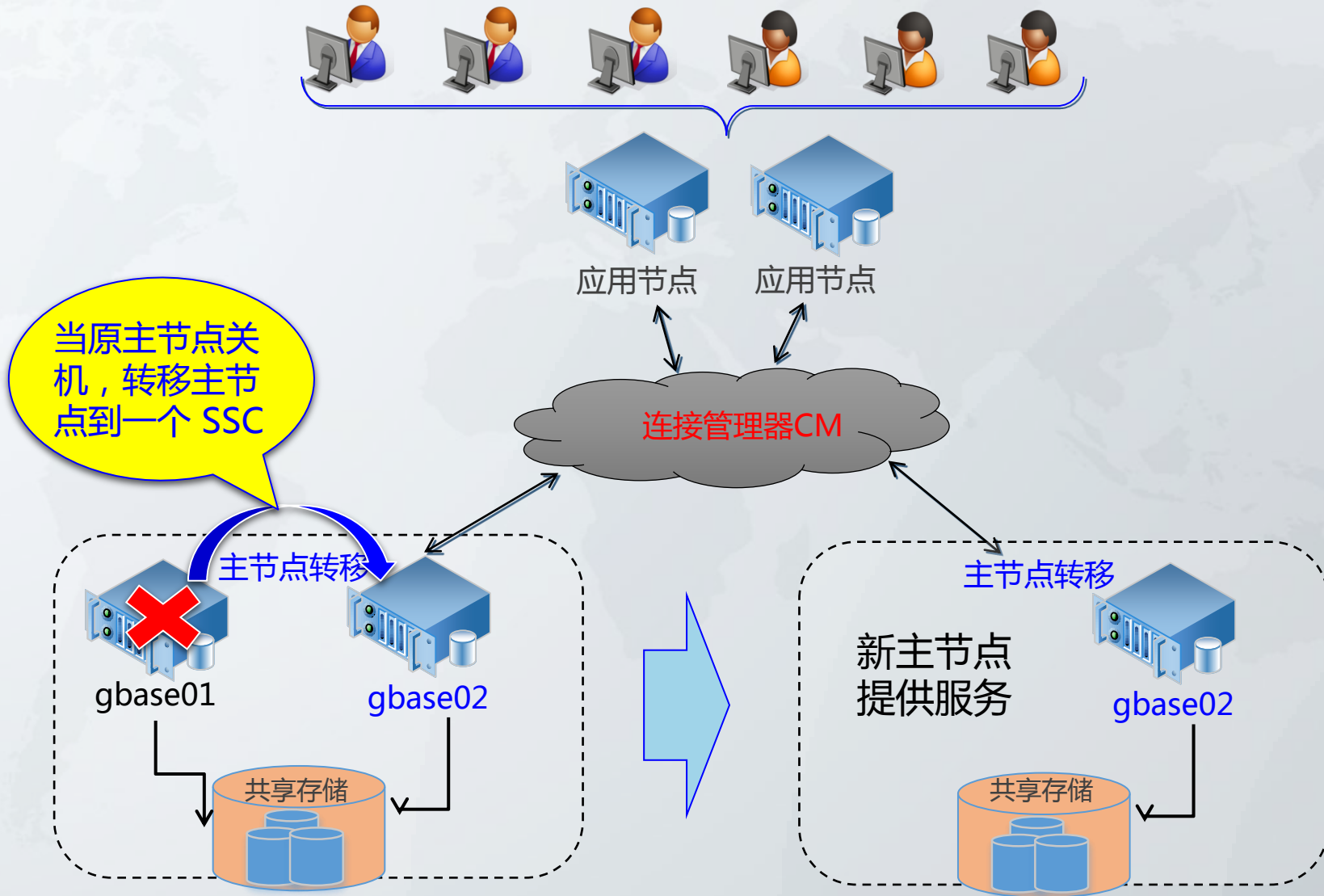
# SLAs For CM
oltp_sla     onsoctcp  192.168.1.101  18888
report_sla   onsoctcp  192.168.1.101  17777
```

CM 的SQLHOSTS配置文件

```
# cmconfig
NAME      cm_sds
LOGFILE   /home/gbase/8t/tmp/cm_sds.log
CM_TIMEOUT 40
SECONDARY_EVENT_TIMEOUT 25
EVENT_TIMEOUT 30
SQLHOSTS  LOCAL
LOCAL_IP  127.0.0.1
CLUSTER  main
{
    GBASEDBTSERVER          g_db
    SLA      oltp_sla      DBSERVERS=PRI
    SLA      report_sla    DBSERVERS=SDS,PRI
    FOC      ORDER=SDS,HDR PRIORITY=1
}
```

CM 的CMCONFIG配置文件

# 连接管理器示例 – 故障转移图示



# 连接管理器示例 – 故障转移过程

- › SSC 主节点 gbase01关机

```
[GBASEDBT@gbase01 gbase01]$ onmode -ky
```


- › SSC 辅助节点 gbase02升级为主节点的过程

```
[GBASEDBT@gbase02 gbase02]$ onstat -  
GBase 8s Database Server Version 12.10.FC4G1TL -- Updates Blocked (SDS) -- Up 00:29:59 -- 164460 Kbytes  
  
[GBASEDBT@gbase02 gbase02]$ onstat -  
GBase 8s Database Server Version 12.10.FC4G1TL -- Fast Recovery (SDS) -- Up 00:30:07 -- 164460 Kbytes  
  
[GBASEDBT@gbase02 gbase02]$ onstat -  
GBase 8s Database Server Version 12.10.FC4G1TL -- Fast Recovery -- Up 00:30:14 -- 164460 Kbytes  
Blocked:HA_CONV_STD  
  
[GBASEDBT@gbase02 gbase02]$ onstat -  
GBase 8s Database Server Version 12.10.FC4G1TL -- On-Line -- Up 00:30:22 -- 164460 Kbytes
```


- › gbase02升级为主节点的过程在 30秒以内完成

# 连接管理器总结

- ▶ GBase 8s 的连接是指客户端(应用服务器) 到服务器端的连接方式；  
GBase 8s 连接的前提是网络连接；
- ▶ SQLHOSTS 是GBase 8s 的连接的灵魂，客户端(Client), CM, 数据库集群可以有各自的 SQLHOSTS，当然也可以共用一个，SQLHOSTS 的连接位置决定：
  - 客户端(应用服务器) -> CM -> 服务器端
- ▶ CM 连接管理器基本上是多加的另一层次，它利用SQLHOSTS将连接动态化而创造出，故障转移(failover)功能。



# Q&A



Thank you !

**GBASE<sup>®</sup>**

南大通用数据技术股份有限公司  
General Data Technology Co.,Ltd

版权所有© GBASE 2022