

IBM solidDB
V7.0

高可用性用户指南



注意

在使用本信息及其支持的产品之前，请阅读第 201 页的『声明』中的信息。

第一版、第四修订版

此版本适用于 IBM solidDB V7.0 FP5（产品编号 5724-V17）及所有后续发行版和修订版，直到在新版本中另有声明为止。

© Oy IBM Finland Ab 1993, 2013

目录

图	v
表	vii
更改摘要	ix
关于本手册	xi
图解约定	xi
印刷约定	xii
语法表示法约定	xiii
1 IBM solidDB HotStandby 简介	1
1.1 HotStandby 的关键功能	2
1.1.1 HotStandby API (HSB 管理命令)	3
1.1.2 基本 HotStandby 服务器方案	3
1.1.3 服务器 HotStandby 状态	7
1.1.4 HotStandby 中的复制方式	10
1.1.5 耐久性和记录	13
1.1.6 对只读工作负载进行负载均衡	14
1.1.7 HotStandby 和 SMA	15
1.1.8 HotStandby 和高级复制	16
1.2 性能和 HotStandby	17
1.3 高可用性控制器 (HAC)	19
1.3.1 常见的故障	21
1.3.2 控制数据库服务器进程	22
1.3.3 外部引用实体 (ERE)	22
1.3.4 在 HAC 中联网	23
1.3.5 HAC 日志记录	24
1.4 高可用性管理器 (样本)	24
2 HotStandby 入门	27
2.1 HotStandby 快速启动过程	27
2.2 使用 HAC 的 HotStandby 快速启动过程	28
2.2.1 启动和停止 HA 控制器	30
2.3 启动顺序总结	31
2.4 HotStandby 样本	33
3 管理和配置 HotStandby	35
3.1 HotStandby 管理基础	35
3.1.1 查询 HotStandby 配置参数	36
3.1.2 修改 HotStandby 配置参数	37
3.2 配置 HotStandby	37
3.2.1 定义主节点和辅助节点 HotStandby 配置	37
3.2.2 设置 HotStandby 服务器等待时间以帮助检测已断开的连接或者不可用的连接	38
3.2.3 定义事务耐久性级别	39
3.2.4 定义 HotStandby 数据库复制操作的名称和位置	40
3.2.5 定义主服务器在辅助服务器发生故障期间的行为	40
3.2.6 确保主服务器和辅助服务器的参数值协调	40

3.2.7 确定主服务器设置是否优先于辅助服务器设置	41
3.3 配置 HA 控制器和 HA 管理器	42
3.4 使用 ADMIN COMMAND (HotStandby API) 管理 HotStandby	43
3.4.1 管理任务概述	43
3.4.2 执行 HotStandby 恢复和维护	44
3.4.3 切换服务器状态	44
3.4.4 停止 HotStandby 操作	48
3.4.5 使主服务器与辅助服务器同步	49
3.4.6 连接 HotStandby 服务器	57
3.4.7 检查 HotStandby 状态	58
3.4.8 验证 HotStandby 服务器状态	60
3.4.9 选择使哪个服务器成为主服务器	61
3.4.10 将 HotStandby 服务器更改为非 HotStandby 服务器	62
3.5 性能调整	63
3.5.1 使用安全级别和耐久性级别来调整复制性能	63
3.5.2 调整 netcopy 性能	63
3.5.3 调整数据库同步复制性能	64
3.6 适用于将 solidDB 与 HotStandby 配合使用的特殊注意事项	64
3.6.1 事务隔离级别和内存表	64
3.6.2 网络分区和双重主服务器	64
3.6.3 用尽用于事务日志的空间	65
3.6.4 在辅助服务器中进行调速和多处理	66
3.7 适用于更低成本和更高安全性的配置	67
3.7.1 降低成本: N + 1 个备用服务器和 N + M 个备用服务器方案	67
3.7.2 提高可靠性: 2N + 1 备用服务器和 2N + M 备用服务器方案	68
3.7.3 solidDB HSB 如何支持 N+1 (N+M) 和 2N+1 (2N+M) 方法	68
3.7.4 将 HAC 与备用服务器配合使用	69
4 对应用程序使用 HotStandby	71
4.1 连接 HotStandby 服务器	71
4.1.1 选择连接类型	72
4.2 透明连接	72
4.2.1 定义“透明连接”连接	72
4.2.2 透明连接中的故障透明性	84
4.2.3 透明连接中的负载均衡	86
4.2.4 处理 TC Info 矛盾	89
4.3 基本连接	89
4.3.1 从应用程序重新连接至主服务器	90
4.3.2 重新连接至辅助服务器	94
4.4 定义应用程序与服务器之间的超时	94
4.4.1 应用程序读取超时选项	94
4.4.2 在 Connect 参数中指定 -C 选项	95
4.5 使用 HotStandby 配置 SMA	96
4.6 利用 HotStandby 配置高级复制	97

5 处理高可用性控制器 (HAC) 时发生的故障	101
5.1 主数据库失败	101
5.2 辅助数据库失败	101
5.3 主节点失败	102
5.4 辅助节点失败	103
5.5 HotStandby 链路失败	103
5.6 服务器对外部客户机无响应	104
6 升级 (迁移) HotStandby 服务器	107
6.1 冷迁移过程	107
6.2 热迁移过程	107
6.3 热迁移过程 (使用 netcopy)	110
附录 A. HotStandby 配置参数	115
A.1 服务器端参数	116
A.1.1 [Cluster] 节	116
A.1.2 [HotStandby] 节	116
A.2 客户端参数	119
A.2.1 Com 节	119
A.2.2 TransparentFailover 节	120
A.3 高可用性控制器 (HAC) 参数	120
A.4 高可用性管理器 (HAM) 配置参数	125
A.5 配置文件示例	126
A.5.1 solid.ini 配置文件	126
A.5.2 solidhac.ini 配置文件	126
A.5.3 HAManager.ini 配置文件	129
附录 B. HotStandby 的错误代码	131
B.1 HotStandby 错误和状态码	131
B.2 高可用性控制器错误和状态码	140
B.3 有关 HotStandby 的 solidDB 数据库错误	141
B.4 solidDB 表错误	143
B.5 solidDB 通信错误	144

附录 C. HotStandby 和 HAC ADMIN COMMAND	145
C.1 HotStandby 命令 (管理命令)	145
C.2 高可用性控制器命令 (ADMIN COMMAND)	152
附录 D. 服务器状态过渡	155
D.1 HotStandby 状态过渡图	155
附录 E. HotStandby 系统事件	163
附录 F. 看守程序样本	165
F.1 使用看守程序的 HotStandby 配置	165
F.1.1 看守程序的工作方式	166
F.1.2 系统设计问题	167
F.1.3 看守程序配置	168
F.1.4 使用样本看守程序	169
F.2 故障情况和看守程序操作	169
F.2.1 主服务器已关闭	170
F.2.2 辅助服务器已关闭	172
F.2.3 看守程序已关闭	175
F.2.4 主服务器和辅助服务器之间的通信链路已断开	178
F.2.5 看守程序和主服务器之间的通信链路已断开	181
F.2.6 看守程序和辅助服务器之间的通信链路已断开	183
F.2.7 看守程序与主服务器之间以及主服务器与辅助服务器之间的通信链路已断开	185
F.2.8 看守程序与辅助服务器之间以及主服务器与辅助服务器之间的通信链路已断开	188
F.3 solid.ini 配置文件的 [Watchdog] 节	191
索引	195
声明	201



1. HotStandby 体系结构	2	18. HotStandby 与高级复制: 副本数据库的故障转移	99
2. HotStandby 服务器方案	3	19. HotStandby 服务器状态过渡	157
3. HotStandby 切换到新的主服务器 (也就是旧的辅助服务器)	6	20. 使用了看守程序的异构 HotStandby 配置	168
4. 服务器故障转移和同步复制示例	7	21. 主服务器已关闭的情况和补救措施	171
5. 同步 HotStandby 配置	11	22. 辅助服务器已关闭的情况和补救措施	174
6. 使用 HotStandby 的 SMA 透明连接的体系结构	16	23. 看守程序已关闭的情况和补救措施	177
7. 由 HotStandby 以及主服务器和副本服务器组成的方案	17	24. 主服务器与辅助服务器之间的链路已断开的情况和补救措施	179
8. 高可用性控制器体系结构	20	25. 看守程序与主服务器之间的链路已断开的情况和补救措施	182
9. “外部引用实体”组件	23	26. 看守程序与辅助服务器之间的链路已断开的情况和补救措施	184
10. 高可用性管理器	25	27. 看守程序与主服务器之间以及主服务器与辅助服务器之间的通信链路已断开的情况和补救措施	186
11. 启动顺序总结	32	28. 看守程序与辅助服务器之间以及主服务器与辅助服务器之间的通信链路已断开的情况和补救措施	189
12. 状态切换	45		
13. 手动完全复制过程	52		
14. 示例: TC 连接	82		
15. 示例: 使用多主目录服务器的 TC 连接	83		
16. 示例: 使用 SMA 的 HotStandby 配置	97		
17. HotStandby 与高级复制: 主数据库的故障转移	98		

表

1. HotStandby 图解约定	xi	22. HotStandby 参数	116
2. 印刷约定	xii	23. 客户端通信参数	119
3. 语法表示法约定	xiii	24. TransparentFailover 参数	120
4. 对服务器状态的描述	8	25. HAC 配置参数: [HAController] 节	121
5. 安装顺序步骤	32	26. HAC 配置参数: [LocalDB] 节	123
6. 管理任务	43	27. HAC 配置参数: [RemoteDB] 节	125
7. 连接状态返回值	59	28. HAC 配置参数: [ERE] 节	125
8. 服务器状态	61	29. 高可用性管理器配置参数	125
9. 用于监视辅助服务器中多处理的 pmon 计数器	66	30. 有关 HotStandby 的 solidDB 服务器错误	131
10. 选择连接类型	72	31. solidDB HotStandby 错误	136
11. TC Info 缩写	73	32. solidDB HSB 错误和消息	137
12. 连接字符串选项	74	33. 高可用性控制器错误和状态码	140
13. 可能存在的 TC Info 属性组合	79	34. solidDB 数据库错误	142
14. 连接请求错误	81	35. solidDB 表错误	143
15. 警告	81	36. solidDB 通信错误	144
16. 连接切换请求	84	37. HotStandby 命令 (管理命令)	145
17. 通信链路故障	85	38. 高可用性控制器命令 (ADMIN COMMAND)	152
18. 会话状态保留	85	39. 服务器状态过渡表	158
19. 连接字符串选项	89	40. HotStandby 事件	163
20. HOTSTANDBY_CONNECTSTATUS 状态值	92	41. 看守程序参数	191
21. 集群参数	116		

更改摘要

对修订版 04 的更改

- 在高可用性控制器 (HAC) 参数这一节中增加了新的高可用性控制器 (HAC) 参数
 - `HAController.NetcopyErrorLevel`
 - `HAController.NetcopyWarningLevel`

修订版 03 的更改

- 在高可用性控制器命令 (ADMIN COMMAND) 这一节中增加了先前未记录的高可用性控制器 (HAC) 命令:
 - `hac shutdowndb`
 - `hac restartdb`

修订版 02 的更改

- 在 [HotStandby] 节这一节中增加了新参数 `HotStandby.NetcopyReceiveBufferSize`。
- 在热迁移过程这一节中增加了新的简化热迁移过程。从 7.0 IF2/FP3 开始，热迁移过程不再需要使用 `netcopy` 命令来使服务器在升级期间同步。

修订版 01 的更改

- 更新了使用 HotStandby 配置 SMA TC 和透明连接信息的语法 - ODBC 这两节，更新的信息描述了将 SMA 与透明连接配合使用时如何指定负载均衡方法。将 SMA 与 TC 配合使用时，必须将负载均衡方法设置为 `LOCAL_READ`。如果指定了 `READ_MOSTLY` 或 `WRITE_MOSTLY`，那么将使用网络连接而不是 SMA 连接。
- 更新了热迁移过程：在热迁移中将使用选项 `-xconvert` 或 `-xautoconvert`，而不是 `-xmigratetohsbg2`。
- 更新了透明连接信息的语法 - ODBC 这一节：增加了加密属性 `[USE_GSKIT={YES|NO}]`。

关于本手册

IBM® solidDB®高可用性 (HotStandby) 组件会增强数据库系统依赖性以减少停机时间。HotStandby 使用一种“热备用”方法。在这种方法中，第二个数据库服务器与主服务器并行运行并且保留数据的最新副本。如果主数据库服务器失败，那么高可用性控制器 (HAC) 将以对于应用程序透明的方式切换到辅助服务器，并且不会丢失已落实的事务，对于性能的影响也最小。切换速度非常快 - 可以在几百毫秒之内完成切换，这取决于您所在的硬件和软件环境的特征。

本指南只包含特定于 HotStandby 组件的信息。有关 solidDB 数据库的常规管理和维护信息，请参阅《IBM solidDB 管理员指南》。

本指南假定读者已经具备数据库管理系统 (DBMS) 的一般知识，并且熟悉 SQL 和 solidDB。

图解约定

本文档中包含多个服务器图，用于描述 HotStandby 环境中的不同情况。

下表提供了服务器图的图解要点：

表 1. HotStandby 图解约定

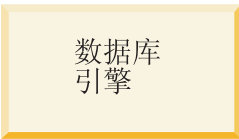


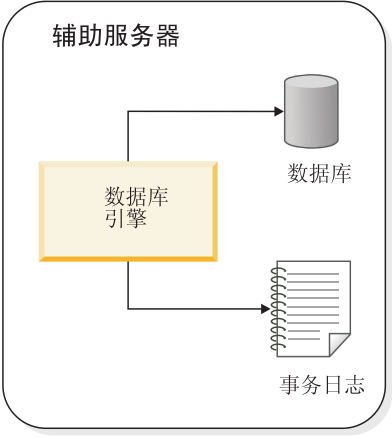
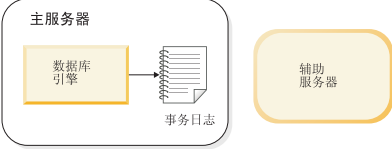
符号	描述
 数据库引擎	矩形表示正在执行的程序，即，数据库服务器（引擎）本身。
 数据库	柱面表示通常存储在磁盘上的数据。也有可能某些数据或所有数据存储在内存中。
 事务日志	此符号表示事务日志（事务日志），用于数据库恢复和 HotStandby 中。

表 1. HotStandby 图解约定 (续)

符号	描述
 <p>辅助服务器</p> <p>数据库引擎</p> <p>数据库</p> <p>事务日志</p>	<p>圆角矩形表示一个具有数据和事务日志的完整服务器。如果圆角矩形中存在辅助服务器或主服务器这两个词语，那么此服务器是 HotStandby 服务器。</p>
 <p>主服务器</p> <p>数据库引擎</p> <p>事务日志</p> <p>辅助服务器</p>	<p>在某些情况下，为了简单起见，将省略用于表示数据库中的数据的数据的柱面。在某些情况下，将更进一步简化此符号以只显示圆角矩形。两个图标都是服务器的简化表示法。</p>

印刷约定

solidDB 文档使用下列印刷约定:

表 2. 印刷约定

格式	适用于
数据库表	此字体用于所有普通文本。
非 NULL	采用此字体的大写字母指示 SQL 关键字和宏名称。
solid.ini	这些字体指示文件名和路径表达式。
SET SYNC MASTER YES; COMMIT WORK;	此字体用于程序代码和程序输出。示例 SQL 语句也使用此字体。
run.sh	此字体用于样本命令行。
TRIG_COUNT()	此字体用于函数名。
java.sql.Connection	此字体用于接口名称。
LockHashSize	此字体用于参数名、函数自变量和 Windows 注册表条目。
<i>argument</i>	此类强调词指示用户或应用程序必须提供的信息。
管理员指南	这种样式用于引用其他文档或者同一文档中的章节。新术语和强调的问题也按此样式书写。

表 2. 印刷约定 (续)

格式	适用于
文件路径表示	除非另有声明，否则文件路径按 UNIX 格式表示。斜杠 (/) 字符表示安装根目录。
操作系统	如果文档包含有关操作系统之间的差别的内容，那么首先提到的是 UNIX 格式。Microsoft Windows 格式位于 UNIX 格式之后并括在括号中。其他操作系统将单独列出。对于不同的操作系统还可能有不同的章节进行描述。

语法表示法约定

solidDB 文档使用下列语法表示法约定：

表 3. 语法表示法约定

格式	适用于
INSERT INTO <i>table_name</i>	语法描述采用此字体。可替换部分采用此字体。
solid.ini	此字体指示文件名和路径表达式。
[]	方括号指示可选项；如果是粗体文本，那么必须将方括号包含在语法中。
	竖线，用于将语法行中的两个互斥选项分隔开。
{ }	大括号用于对语法行中的一组互斥选项进行定界；如果是粗体文本，那么必须将大括号包括在语法中。
...	省略号指示可以多次重复使用自变量。
· · ·	由三个点组成的一列表示这是先前代码行的延续。

1 IBM solidDB HotStandby 简介

高可用性 (HA) 系统的目标是要使系统可以容忍故障。为了实现高可用性, solidDB HotStandby 组件允许辅助服务器 (备用服务器) 与主服务器 (活动的服务器) 并行运行, 并保留主服务器中数据的最新副本。

HotStandby 组件会实现内部状态机, 该状态机可使服务器知道 HA 状态。HA 状态机能够保持数据库一致性。例如, 当服务器处于辅助服务器状态 (即, 从主服务器接收事务流) 时, 会禁用辅助数据库的更新。

可以使用看守程序来控制服务器的可用性或状态。solidDB 服务器提供了称为 solidDB 高可用性控制器 (HAC) 的看守程序实现。

在内部, HAC 使用一组 HotStandby 命令 (HotStandby API) 来控制服务器状态。诸如此类的解决方案能够实现具有更高可靠性的系统。即使数据库服务器失败, 也不会使您的站点完全停止。在 solidDB 支持的任何引擎配置中 (例如, 具有主控服务器和副本服务器的 solidDB 高级复制), HotStandby 允许辅助数据库在仅仅几百毫秒之内就可以替换失败的数据库。

HotStandby 体系结构

HotStandby 包括以下组件:

- 看守程序 (例如, HAC)
- HotStandby API (HSB 管理命令)
- 主 solidDB 服务器和辅助 solidDB 服务器

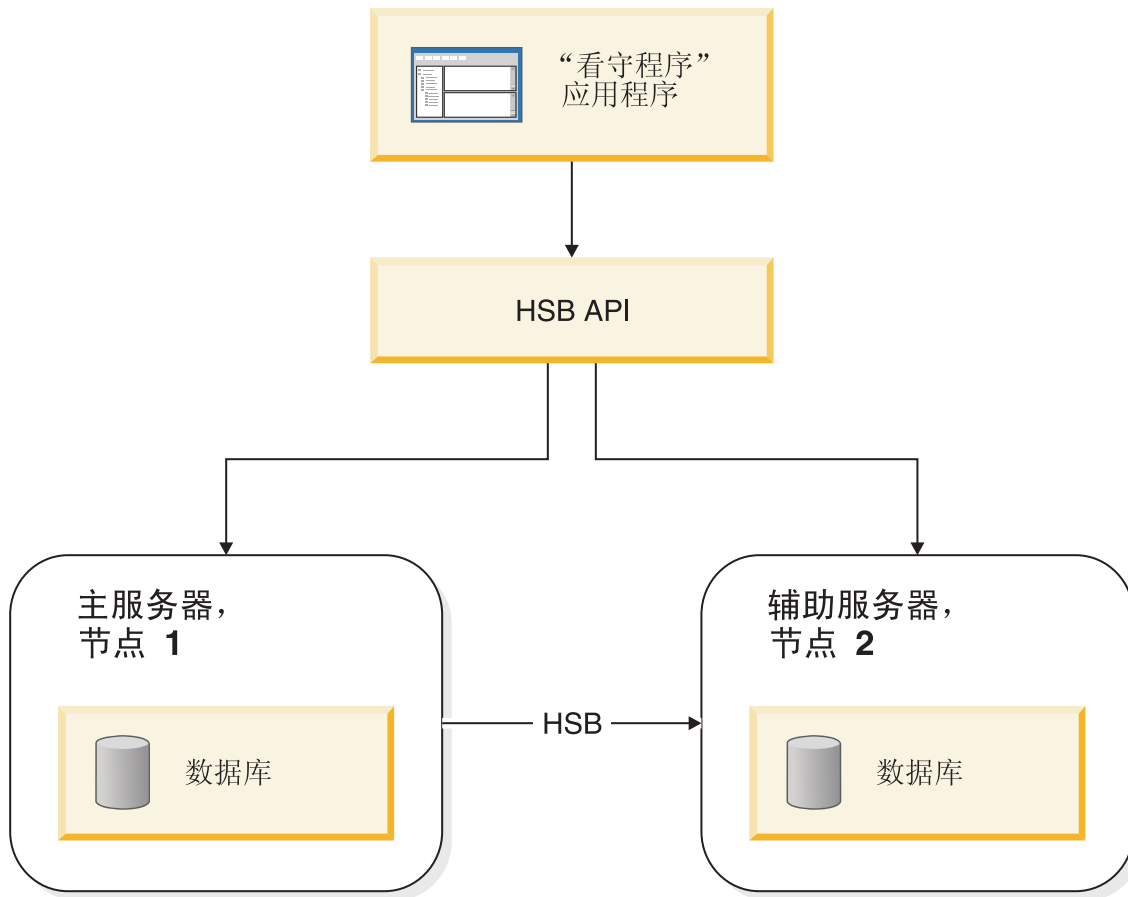


图 1. HotStandby 体系结构

操作原则

HotStandby (HSB) 在主服务器和辅助服务器这两个节点之间执行同步事务复制。

主服务器节点（主服务器）包含活动数据库。辅助服务器节点（辅助服务器）包含此活动数据库的准确而且是最新的副本。如果主服务器失败，它还可以接替主服务器。

辅助服务器将接收来自主服务器的更新。当主服务器失败时，辅助服务器可以接替它成为新的主服务器。提供辅助服务器的另一个好处在于，辅助服务器还可以对来自客户机的只读请求（例如，SELECT 语句）作出响应。这样就可以将一些工作负载分散在两台服务器上，而不是由一台服务器来承担。

注：“热备用”这个术语 (hot standby) 是指一项常规技术，即，当第一台服务器发生故障时，另一台服务器可以立即接替工作。而“HotStandby”（它是一个单词，首字母大写）是指 solidDB 用于实现此常规技术的特定组件。通常将 HotStandby 缩写为 HSB。

同样，看守程序也是指一项技术，用于监控两个数据库的状态，必要时可以切换状态。HAC 就是 solidDB 看守程序的实现。您还可以使用 HSB API 来编写您自己的看守程序。solidDB 软件包还包括可供您用作起始点的看守程序样本。

1.1 HotStandby 的关键功能

HotStandby 的关键功能包括故障转移支持、事务记录和负载均衡。

1.1.1 HotStandby API (HSB 管理命令)

高可用性行为由一个 API 进行控制，该 API 基于 solidDB 管理命令的子集。

这个命令子集由命令前缀 **hotstandby** (缩写为 **hsb**) 进行识别。可以使用任何支持 SQL 的工具 (例如, **solsql**) 或者诸如 ODBC 或 JDBC 等程序化接口来发出这些命令。HotStandby 管理命令具有以下语法:

```
admin command 'hotstandby hsb-command options';
```

或者

```
admin command 'hsb hsb-command options';
```

hotstandby 管理命令用于控制 solidDB HSB 服务器的高可用性状态, 或检索状态信息。可以手动发出这些命令, 也可以通过程序发出这些命令。solidDB HA 管理工具、高可用性控制器 (HAC) 和看守程序样本以编程方式使用这些命令。

1.1.2 基本 HotStandby 服务器方案

在基本 HotStandby 服务器方案中, 有两个数据库服务器 - 主服务器和辅助服务器。每个服务器都具有自己用于存储数据库的磁盘驱动器, 并且都具有自己的事务日志 (Txn 日志)。

图 2 说明了基本 HotStandby 服务器方案。主服务器写入它的事务日志并转发至辅助服务器, 以便辅助服务器可以对它的数据库副本进行相同更改。HSB 中不会主动涉及到辅助服务器上的事务日志, 但是会维护该事务日志, 以便辅助服务器可以恢复已落实但尚未写入主数据表的数据。

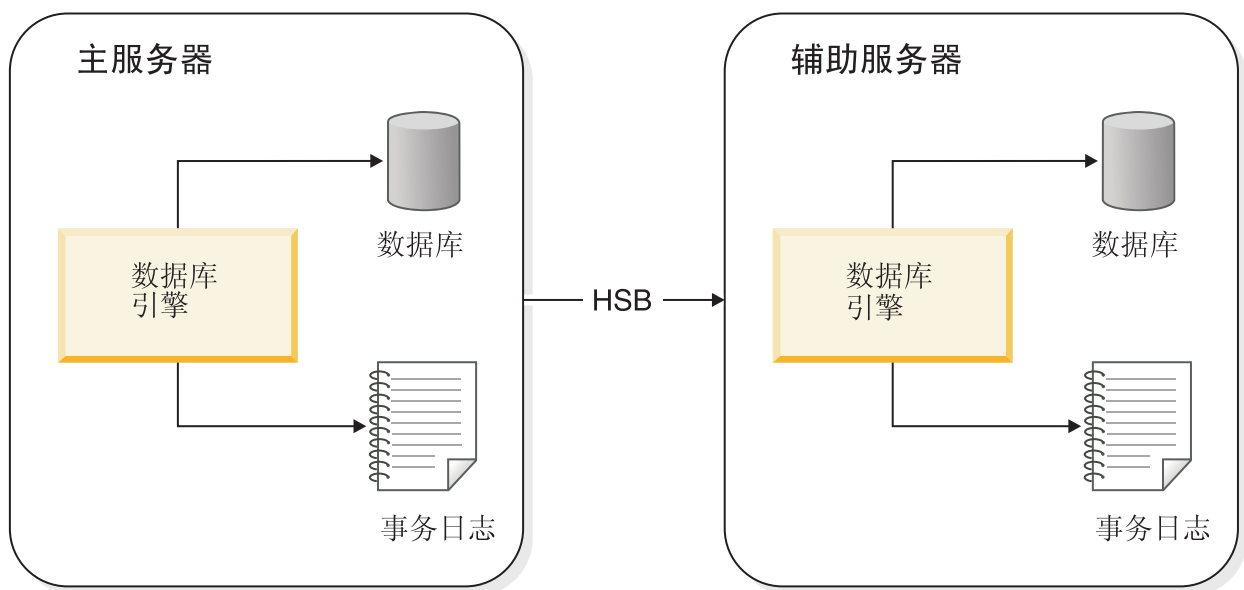


图 2. HotStandby 服务器方案

脉动信号

solidDB HSB 在内部使用一项称为脉动信号的技术来监视服务器之间的连接。

活动服务器与备用服务器之间将发送一系列保持活动消息。这两个服务器都会不断地向对方发送单向消息“我在此处”。将按固定的时间间隔发送这些消息。期望来自另一个服务器的消息在预定义的时间段内到达。在 solidDB 中，脉动信号技术称为 *ping*。

要点: 在 solidDB 中，脉动信号技术称为 *ping*，尽管没有发送 *ping* 请求。脉动信号技术与因特网协议网络中使用的 Ping 协议不同。

事务日志和 HotStandby

HotStandby 使用主服务器事务日志，该事务日志中包含主服务器上已落实事务的副本。在非 HotStandby 服务器中，当服务器异常关闭时，此事务日志可用来恢复数据。

在 HotStandby 主服务器中，还会将日志数据发送至辅助服务器，以便它知道要更新的数据。辅助数据库运行连续的前滚过程，该过程将接收日志数据并且使辅助服务器上数据的副本与主服务器上数据的副本保持同步。

如果主服务器失败，那么看守程序将让辅助服务器成为主服务器。在新的主服务器开始运行之后，客户机可以与其建立连接并继续工作。客户机会继续看到在主服务器关闭之前已落实的所有数据。（客户机必须重新启动在原始主服务器关闭时已启动但未完成的所有事务。）

为 HSB 环境中的客户机提供了一种称为透明连接 (TC) 的特殊类型的客户机连接，用于处理故障转移和切换。有关更多信息，请参阅第 71 页的 4，『对应用程序使用 HotStandby』。

如果辅助服务器失败，那么主服务器可以继续运行。它会继续将数据写入事务日志并且保留此事务日志，直到主服务器与辅助服务器互相重新建立连接，并且主服务器已将此日志发送至辅助服务器为止。主服务器保留日志的确切时间长度取决于 `solid.ini` 配置参数 `General.CheckpointDeleteLog` 和 `General.BackupDeleteLog` 的设置。

1. 如果 `General.CheckpointDeleteLog=Y`，那么主服务器将从辅助服务器关闭开始或者从最新检查点开始（以这两者中较早者为准）保留所有事务日志。
2. 如果 `General.CheckpointDeleteLog=N` 并且 `General.BackupDeleteLog=Y`，那么主服务器将从辅助服务器关闭开始或者从最新备份开始（以这两者中较早者为准）保留所有事务日志。
3. 如果 `General.CheckpointDeleteLog=N` 并且 `General.BackupDeleteLog=N`，那么服务器将无限期地保留日志。

在发生故障的数据库服务器再次可用时，可以将它配置为新的辅助数据库服务器（未发生故障的服务器已成为当前主服务器）。

如果主服务器失败，那么服务器之间的职责将互换，即，原来的辅助服务器将接替成为主服务器，而原来的主服务器在被修复之后将返回系统中成为新的辅助服务器。每当发生故障时，服务器之间的职责就会互换。正是因为任一服务器都可以作为主服务器这一事实，使得系统在经历多次故障之后仍能幸存，并且将无限期地继续运行。

注: 如果主服务器很长时间都无法与辅助服务器联系，那么事务日志会填满所有可用的磁盘空间。通过应用适当的配置参数设置，可以避免出现由于日志文件较大而用完磁盘空间的情况。有关更多信息，请参阅第 65 页的 3.6.3，『用尽用于事务日志的空间』。

您可以使用 HSB 来缩短硬件和软件升级期间的当机时间。可以让一个服务器作为主服务器运行，而升级另一个服务器。

还可以使用 `HotStandby` 来帮助选择速度与安全的定制均衡。HSB 参数 `HotStandby.SafenessLevel` 和 `HotStandby.2SafeAckPolicy` 用于控制辅助服务器确认事务的方式。这些参数与记录相关的 `Logging.DurabilityLevel` 参数一起，允许您指定速度与安全的组合。在非 HSB 服务器上，某些参数设置可以提高性能。（有关更多信息，请参阅第 63 页的 3.5，『性能调整』中对耐久性级别和安全参数的讨论。）

还可以通过使用 `HotStandby.SafenessLevel=auto` 设置，将安全级别配置为根据耐久性级别动态更改。

故障转移

在故障转移中，辅助服务器将切换为新的主服务器。

有多种原因会造成辅助服务器切换为新的主服务器：

1. 主服务器发生故障 -
2. 您要管理主服务器。
3. 当系统上不存在现有主服务器时，您必须选择一个主服务器。

通过在辅助服务器上发出以下命令将辅助服务器切换为新的主服务器：

```
ADMIN COMMAND 'hotstandby set primary alone';
```

进行故障转移时，新的主服务器中将包含旧的主数据库中最新落实的数据。可以在辅助数据库中找到主数据库中已落实的所有数据。如果使用了“透明连接 (TC)”，那么在故障转移时就不会丢失连接。但是，正在进行的事务将异常终止，必须重新执行此事务。有关更多信息，请参阅第 84 页的 4.2.2，『透明连接中的故障透明性』。新的主服务器可以单独运行，并继续将事务和数据写入它的数据库和事务日志。

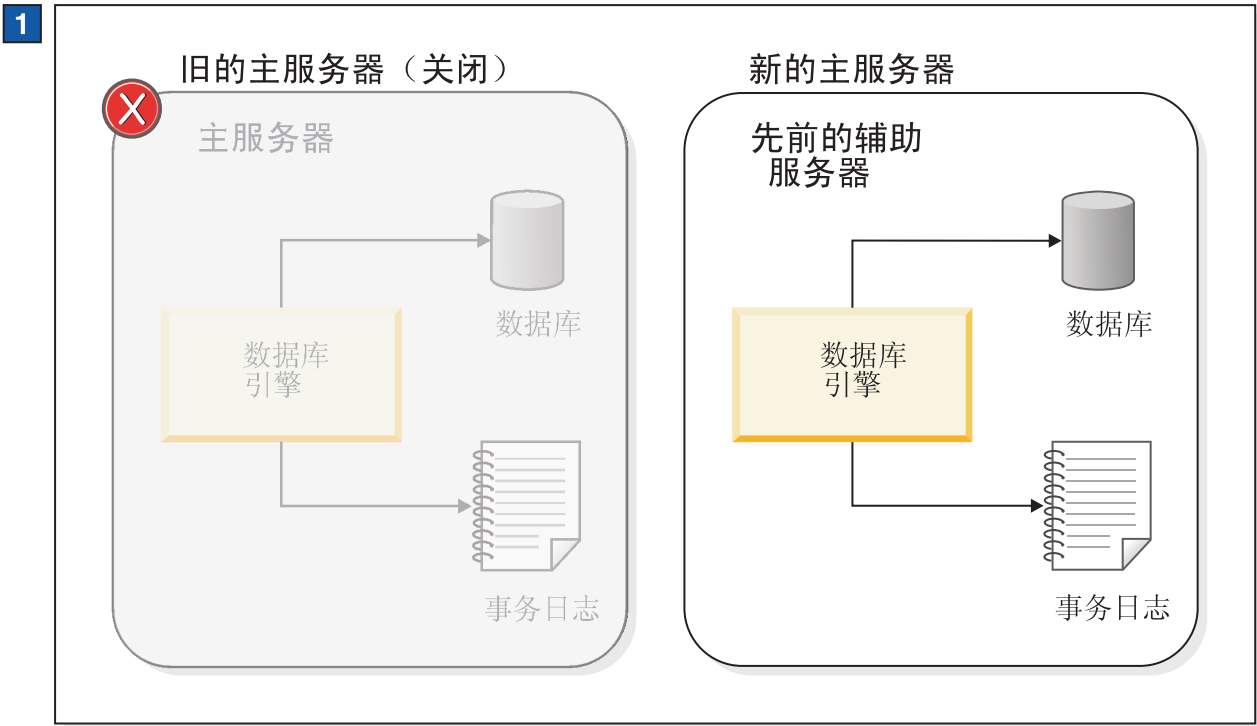


图 3. HotStandby 切换到新的主服务器 (也就是旧的辅助服务器)

1. 在旧的主服务器失败之后，原来的辅助服务器变成了新的主服务器。

服务器同步复制

如果存在现有主服务器，那么旧的主服务器在重新联机时，将成为新的辅助服务器。在此阶段，由于新事务将落实到新的主数据库中，因此新的辅助服务器中的信息落后于新的主服务器中的信息。为了使新的辅助服务器中的信息也是最新的，在连接服务器之后，会自动将新的主服务器的事务日志数据发送至新的辅助服务器。所有暂挂的更改将从事务日志写入新的辅助服务器，以便辅助服务器可以与主服务器保持同步。第 7 页的图 4 中说明了服务器同步复制。

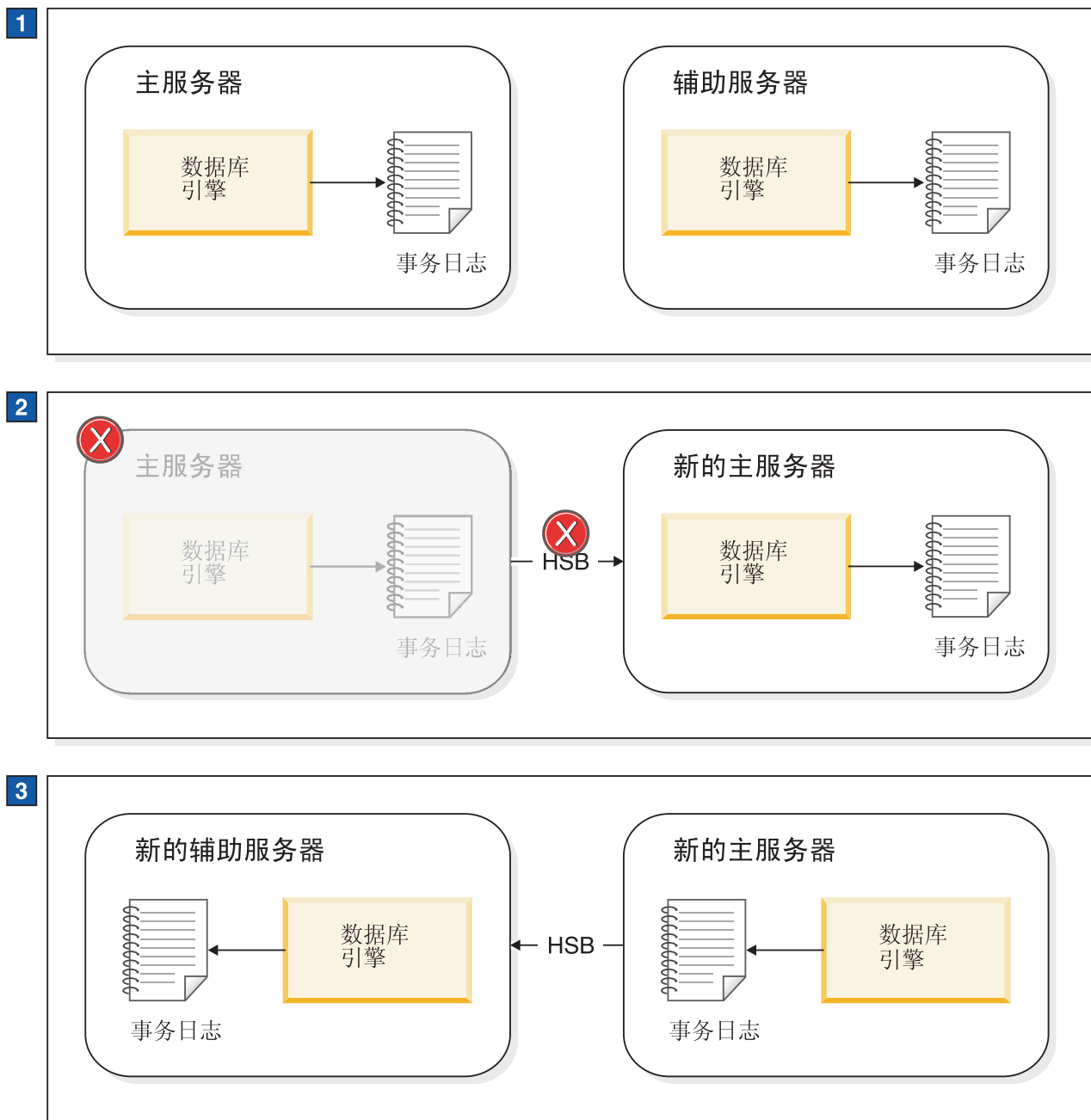


图 4. 服务器故障转移和同步复制示例

1. 正常操作: 主服务器将数据发送至辅助服务器。
2. 当主服务器失败时, 辅助服务器将接替它成为新的主服务器。新的主服务器会将事务信息保存在它的事务日志中, 以便它稍后可以将数据发送至新的辅助服务器。
3. 在将旧的主服务器作为新的辅助服务器启动之后, 会将新的主服务器的事务日志中的信息发送至新的辅助服务器, 以便新的辅助服务器可以执行同步复制。

1.1.3 服务器 HotStandby 状态

在 HotStandby 系统中, 每个服务器都处于其中一种可能存在的状态, 此状态描述了服务器的当前行为。

例如，当主服务器和辅助服务器正在进行通信和同步时，它们处于 PRIMARY ACTIVE 和 SECONDARY ACTIVE 状态。

或者，如果主服务器与辅助服务器失去联系，那么主服务器将自动切换到 PRIMARY UNCERTAIN 状态。在该状态下，它不会接受新的事务。用户或者 HAC（后者是更典型的情况）可以将主服务器切换到 PRIMARY ALONE 状态。在该状态下，主服务器充当一个独立服务器，它接受新事务并存储它们以便稍后发送至辅助服务器。

对服务器状态的描述

组成 HotStandby (HSB) 对的两个服务器都具有可以查询和处理的状态。

表 4. 对服务器状态的描述

状态	描述
PRIMARY ACTIVE	已连接服务器，并且此服务器（主服务器）正在接受读/写事务并正在将数据发送至辅助服务器。辅助服务器必须处于 SECONDARY ACTIVE 状态。
PRIMARY ALONE	对等服务器之间未互相连接。对等服务器可能已启动，但是未连接该服务器，因此它不接受任何事务。对等服务器可能处于 SECONDARY ALONE 状态。 此服务器（主服务器）正在积极接受和执行读/写事务，并且正在收集这些事务以便稍后发送至辅助服务器。

表 4. 对服务器状态的描述 (续)

状态	描述
PRIMARY UNCERTAIN	<p>服务器已异常断开连接，并且 AutoPrimaryAlone 配置参数设置为 No。在 PRIMARY UNCERTAIN 状态下，未经确认的任何事务都将保持暂挂状态，这意味着在 HAC 将此服务器更改为另一种状态之前，此服务器不会落实或回滚事务。</p> <p>操作员可以选择执行以下三项操作：将主服务器重新连接至辅助服务器；将主服务器设置为 PRIMARY ALONE 状态；将主服务器设置为 SECONDARY ALONE 状态。</p> <ol style="list-style-type: none"> 1. 如果主服务器已重新连接至辅助服务器，就会在主服务器上落实事务。 2. 如果状态更改为 PRIMARY ALONE，就会在主服务器上落实已打开的事务。 3. 如果状态更改为 SECONDARY ALONE，那么已打开的事务将保持暂挂状态。将在服务器更改为另一种状态之后最终解决这些事务。例如，如果服务器更改为 SECONDARY ACTIVE 状态，那么将根据同步复制结果异常终止或落实被阻塞的事务。如果服务器状态更改为 STANDALONE 或 PRIMARY ALONE，就会落实被阻塞的事务。 <p>如果您希望主服务器在与辅助服务器失去联系时就自动进入 PRIMARY ALONE 状态而不是 PRIMARY UNCERTAIN 状态，那么请阅读关于 AutoPrimaryAlone 配置参数的描述。</p> <p>注：</p> <p>HAC 可以始终使服务器从 PRIMARY UNCERTAIN 状态切换到 SECONDARY ALONE 状态，从而获得最高安全性。这样可以防止存在双重主服务器的可能性。但是，这样也会阻止用户更新服务器上的数据。有关更多信息，请参阅第 64 页的 3.6.2，『网络分区和双重主服务器』。</p>
SECONDARY ACTIVE	<p>对等服务器之间已互相连接，并且此服务器正在接受来自主服务器的入局事务日志数据。将在辅助服务器上执行这些事务，以使它与主服务器具有相同的数据。还会将这些事务写入辅助服务器的事务日志，以便辅助服务器在发生故障时本身可以恢复数据。此外，客户机还可以在处于 SECONDARY ACTIVE 状态的服务器上执行只读事务。当服务器处于 SECONDARY ACTIVE 状态时，此服务器的对等服务器必须处于 PRIMARY ACTIVE 状态。</p>

表 4. 对服务器状态的描述 (续)

状态	描述
SECONDARY ALONE	辅助服务器已与其对等服务器断开连接。它仅接受读取请求。可以通过在辅助服务器或主服务器上发出 HotStandby connect 命令来将服务器连接至对等服务器。
STANDALONE	服务器没有 HSB 状态（主服务器或辅助服务器），且以常规独立服务器的运行方式运行。同时也按正常方式处理和除去事务日志；不会为辅助服务器保存事务日志。要恢复 HSB 操作，必须将服务器设置为 PRIMARY ALONE 或 SECONDARY ALONE，并且主服务器必须执行 netcopy 或 copy 操作以将数据库的完整副本发送至辅助服务器。
OFFLINE	<p>服务器是以“netcopy 侦听方式”（也称为“备份服务器方式”）启动的。在此方式下，服务器将等待来自处于 PRIMARY ALONE 状态的服务器的人局 netcopy。在服务器成功完成 netcopy 之后，服务器将更改为 SECONDARY ALONE 状态。</p> <p>因为处于 OFFLINE 状态的服务器不接受客户机连接，所以您无法直接观察 OFFLINE 状态。如果您尝试连接至处于 OFFLINE 状态的服务器，那么将返回错误代码 14552 Server is in backup server mode, no connections are allowed。处于 OFFLINE 状态的服务器将只会对 netcopy 操作作出响应。</p>

1.1.4 HotStandby 中的复制方式

HotStandby 复制协议的目的在于将事务结果从主服务器安全地运送到辅助服务器。solidDB 同时提供了同步 (*2-safe*) 和异步 (*1-safe*) 复制协议。与各种耐久性（记录）级别一起，复制协议可以用于调整 HotStandby 系统，以使它的性能与持久性之间达到必需的均衡。

- **1-safe**: 首先在主服务器中落实事务，然后再将此事务发送到辅助服务器中。
- **2-safe**: 在未得到辅助服务器确认之前不落实事务（这是缺省值）。

可以在以下三个级别控制安全级别:

- 全局 - **HotStandby.SafenessLevel1** 参数
- 会话 - SET SAFENESS 语句
- 事务 - SET TRANSACTION SAFENESS 语句

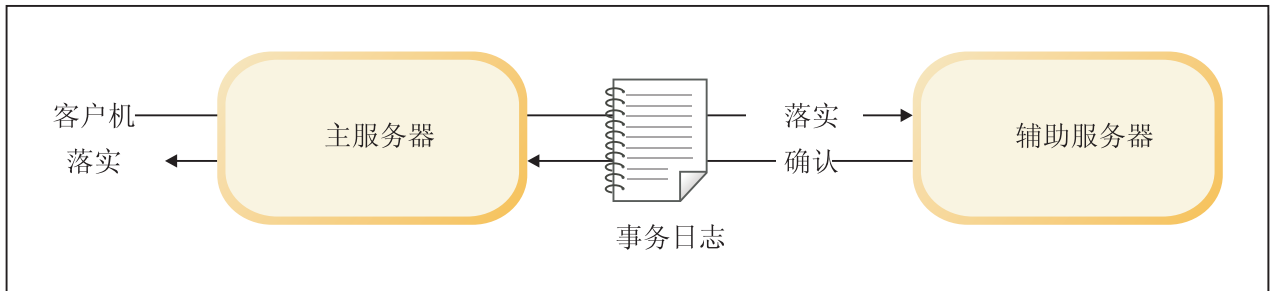
使用 **2-safe** 复制的同步 HotStandby

为了确保主服务器和辅助服务器具有相同的数据，solidDB 主要使用同步 *HotStandby* 模型。这种方法称为 *2-safe* 复制方法；在让用户知道已经落实了数据之前会将数据写入两个位置。

在落实对主数据库中的事务所作的更改之前，主服务器会将事务数据发送至辅助服务器。辅助服务器必须将确认发送给已落实或至少已接收数据的主服务器。否则，主服务器将超时，并且其状态会从 PRIMARY ACTIVE 更改为 PRIMARY UNCERTAIN。在这种情况下，主服务器将无法回滚或落实事务。HAC 可以将主服务器设置为 PRIMARY ALONE 状态，此状态允许主服务器继续接收事务并独立于辅助服务器执行操作。它将落实已发送至辅助服务器的暂挂事务并继续接受新的事务。

注：一旦辅助服务器已经落实了（或者至少已经接收到）事务日志条目，它就会发送一条确认消息。此配置可以防止在发生单一故障时丢失事务。此外，可以选择保留基于文件的事务日志，以便于在整个系统发生故障时恢复数据库。

主服务器和辅助服务器都在运行



主服务器运行，辅助服务器关闭

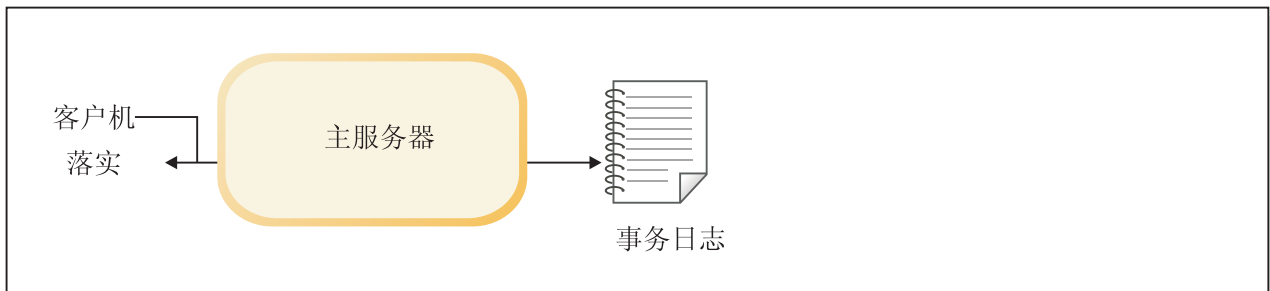


图 5. 同步 HotStandby 配置

发送数据时执行的基本步骤

使用同步复制发送数据的过程包括以下步骤：

1. 主服务器将数据（采用记录级别格式）写入主节点中的事务日志。
2. 当主服务器遇到 commit 语句时，会将所有已更改的数据发送至辅助服务器。

注：如果辅助服务器在事务启动之后到主服务器发送数据之前这段时间内失败，那么主服务器将回滚此事务。

3. 辅助服务器将确认落实消息。确认的时间取决于 **HotStandby.2SafeAckPolicy** 配置参数的设置。通过最快的备用方式 (*2-safe received*)，辅助服务器一接收到落实消息就立即向主服务器发送确认消息。使用最安全的备用方式 (*2-safe durable*) 时，辅助服务器在已执行事务并将其持久地写入自己的事务日志之后才发送确认消息。

当主服务器接收到确认消息时，主服务器会通知用户已经落实了数据。

4. 如果主服务器未接收到来自辅助服务器的确认消息（例如，由于发生网络故障或节点故障），那么主服务器将超时并切换到 **PRIMARY UNCERTAIN** 状态。由于主服务器不知道辅助服务器中的最新事务的状态，因此无法回滚或落实事务。主服务器不知道发生了下面哪种情况：
 - 在落实事务之前，辅助服务器已当机。
 - 辅助服务器已经落实了事务，但是主服务器未接收到确认消息（例如，由于发生网络故障而导致未接收到）。

当服务器处于 **PRIMARY UNCERTAIN** 状态时，会阻塞用户尝试落实的当前事务和新事务，并且用户可能会感觉到服务器不响应。

5. 如果 **HAC** 检测到辅助服务器已关闭或者网络发生了故障，那么它可以将主服务器切换到 **PRIMARY ALONE** 状态。主服务器在设置为 **PRIMARY ALONE** 时，它会落实已发送至辅助服务器的暂挂事务并继续接受新的事务。
6. 更改将被累积到事务日志文件中，直到辅助服务器恢复运行或者主服务器用尽磁盘空间为止。如果服务器用尽了用于保存事务日志的磁盘空间，那么主服务器将更改为只读方式。
7. 如果辅助服务器停止运行较长时间，并且服务器可能用尽用于保存事务日志的磁盘空间，那么您可能要将主服务器从 **PRIMARY ALONE** 状态切换到 **STAND-ALONE** 状态。这表示事务日志不会存储自从与辅助服务器失去联系以来发生的所有事务，因此，辅助服务器仅仅通过从主服务器中读取事务日志将无法进行同步复制。如果无法通过事务日志使辅助服务器达到最新状态，那么唯一的办法就是使辅助服务器与主服务器同步，以将主服务器的数据库文件复制到辅助服务器。这可以通过执行 **HotStandby netcopy** 命令来实现。
8. 如果要执行 **hotstandby netcopy** 命令，主服务器必须处于 **PRIMARY ALONE** 状态。执行 **hotstandby netcopy** 命令之后，无论此命令成功还是失败，主服务器都将保持处于 **PRIMARY ALONE** 状态。
9. 如果要使主服务器再次开始将其事务发送给辅助服务器，必须使用 **hotstandby connect** 命令将主服务器显式连接至辅助服务器。在主服务器连接至辅助服务器之后，主服务器将在 **PRIMARY ACTIVE** 状态下运行。

在服务器之间建立连接之后，服务器开始执行同步复制 - 此时，所有暂挂更改将从事务日志自动写入辅助服务器，以便辅助服务器与主服务器保持同步。在服务器执行同步复制之前，主服务器与辅助服务器将交换信息并确定在何处开始执行同步复制，以便在辅助服务器上不会将事务落实两次。

相关任务:

第 53 页的『通过网络将主数据库复制到辅助服务器』

要将主服务器中的数据库文件的副本发送至辅助服务器，请使用 **netcopy** 命令。辅助服务器必须已经在运行。

第 56 页的『将数据库文件从主服务器复制到指定目录』

如果辅助服务器用于数据库的目录对于主服务器可视，那么您可以使用 **hotstandby copy** 命令将数据库从主服务器复制到辅助服务器。

第 57 页的 3.4.6, 『连接 HotStandby 服务器』

使用 1-safe 复制的异步 HotStandby

（可选）可以使用从主服务器到辅助服务器的异步复制。这种复制方法称为 *1-safe* 复制。

采用 1-safe 复制时，在主服务器中落实了事务之后，会立即确认这些事务。这将显著提高性能。落实事务之后，采用异步方式将它们发送至辅助服务器。这种方法也存在一定弊端，当主服务器中发生故障时，已在传输中的一些事务可能会丢失。

可以动态选择这两种复制方法中的任一方法，甚至可以对每个会话或事务选择复制方法。也可以控制 1-safe 复制涉及到的复制延迟。

1.1.5 持久性和记录

持久性级别会控制 solidDB 处理事务记录的方式。solidDB 服务器支持三种持久性级别：严格（安全）、宽松（快速）和自适应。宽松持久性会产生最佳性能，而严格持久性会在最大程度上减少事务的丢失。自适应持久性级别仅适用于 HotStandby 配置中。

- **严格持久性**：如果一旦落实了事务就将此事务写入事务日志，这就称为“严格持久性”。这种持久性提供了最高安全性。
- **宽松持久性**：如果允许服务器延迟到它不太忙或者可以同时写入多个事务时才写入事务，这就称为“宽松持久性”（或者“宽松记录”）。如果在不属于 HSB 对的服务器中使用宽松持久性，那么当该服务器异常终止时，存在丢失最新的一些事务的风险。但是，如果该服务器属于 HSB 对，那么事务的副本位于另一个服务器（辅助服务器）上。并且，即使主服务器在记录事务之前发生故障，也不会丢失事务。因此，将宽松持久性与 HSB 配合使用时，宽松持久性会导致安全性稍微降低。宽松持久性也可以提高系统的性能，尤其是在服务器负载由许多较小的写事务组成的情况下。
- **自适应持久性**：自适应持久性仅适用于 HotStandby 主服务器。自适应持久性意味着，如果服务器处于 PRIMARY ACTIVE 状态（正在将事务发送至辅助服务器），那么它将使用宽松持久性。在任何其他状态下，将使用严格持久性。因此，当 HSB 处于活动状态时，您将获得高性能（安全性稍微降低）；如果只有一个服务器在运行，您还将保持较高的安全性。仅当使用 2-Safe 复制时，才能有效地启用自适应持久性（这是缺省情况）。

自适应持久性可以显著提高性能，同时又能使数据在故障情况下具有较高的安全程度。它可以提高整个系统的吞吐量并且可以缩短等待时间，即，用户在被告知已经落实了事务之前必须等待的时间。

可以利用 **Logging.DurabilityLevel** 参数将持久性级别设置为服务器缺省值，或利用 SET [TRANSACTION] DURABILITY 语句来根据会话或事务进行设置。

关于复制协议，STRICT 对应于 2-safe，RELAXED 对应于 1-safe。

1-Safe 复制

使用 1-safe 复制时，在主服务器中完成了落实处理之后，会立即确认 commit 语句。在已将控制权返回给应用程序之后，会将已落实的事务异步传输至辅助服务器。

传输事务时的延迟可能从几毫秒到几百毫秒。由于主服务器中的等待时间急剧缩短，因此 1-safe 复制将显著提高性能。1-safe 复制的不足之处在于，在发生故障的情况下，一些事务可能会在故障转移时丢失。

可以使用参数 **HotStandby.SafenessLevel=1safe** 来为服务器设置 1-safe 复制。**HotStandby.SafenessLevel** 参数的值可以是 1safe、2safe 和 auto；缺省值是 2safe。

还可以使用 SET 命令来动态控制安全级别：

```
SET SAFENESS {1SAFE| 2SAFE| DEFAULT}
```

SET SAFENESS 设置当前会话的安全级别，直到它更改为止。

```
SET TRANSACTION SAFENESS {1SAFE| 2SAFE| DEFAULT}
```

SET TRANSACTION SAFENESS 设置当前事务的安全级别。落实之后，安全级别可以重新恢复为会话设置的值、启动值或者系统缺省值（即，2-safe）。

选项 DEFAULT 表示会话的当前设置。

如果 **HotStandby.SafenessLevel** 参数设置为 auto（即，“自动”），那么您可以通过程序化耐久性控件（例如，SET DURABILITY RELAXED）来控制安全级别。

2-safe 确认策略

当启用了 2-safe 复制时（这是缺省情况），主服务器要在接收到辅助服务器已具有事务的确认消息之后，才会让客户机知道已经成功地落实了此事务。

有三种不同的确认策略：

- 2-safe received: 当辅助服务器接收到数据时，它才会发送确认消息（这是缺省情况）。
- 2-safe visible: 辅助服务器已更新它的数据副本，并且更改现在是可视的。换句话说，已连接至辅助服务器的客户机应用程序将能够查看更新。
- 2-safe durable: 辅助服务器将在它已使数据成为持久数据之后（即，在它已落实数据并且将数据写入磁盘之后）进行确认。

选择 2-safe 确认策略

使用 2-safe received 时速度将更快。使用 2-safe durable 时将更安全。由于仅当主服务器和辅助服务器都处于活动状态（即，都在应用事务）时这些确认策略才适用，因此，即使 2-safe received 也认为是安全的。仅当两台服务器在一秒钟之内几乎同时失败，才会存在丢失事务的风险。

通过使用 2-safe received 可以缩短等待时间，等待时间就是从开始落实到用户接收到落实确认之间的这段时间。2-safe received 策略对总体吞吐量的影响很小。

1.1.6 对只读工作负载进行负载均衡

除了内置的透明故障转移功能之外，solidDB JDBC 和 ODBC 驱动程序中的“透明连接”功能也支持对只读工作负载进行负载均衡。

负载均衡基于以下事实：存在两个同时运行的已同步 (HotStandby) 数据库。所有读查询都提供相同的结果，而不管是在主数据库还是在辅助数据库中执行查询。

激活了负载均衡时，JDBC 或 ODBC 驱动程序将使用两个物理连接（每个数据库各有一个）并将查询负载分配给工作负载连接。将根据查询类型（例如，读或写）和数据库服务器中的当前负载来选择工作负载连接。

负载均衡实现基于以下原则：

- 如果隔离级别是“落实读”，那么可以在任一数据库中执行只读查询。
- 在主数据库中执行需要较高隔离级别（对于更新操作，请选择可重复读）的读查询。

- 始终在主数据库中执行写查询。
- 在主数据库中执行同一事务中任何写操作之后的读查询。通过使用主数据库，可以确保已更新的行对于后续读操作是可视的。
- 确保数据库的内部读/写级别一致性，以便在落实写事务之后，不会将辅助数据库用于通过同一连接进行的读操作，直到辅助数据库对于该写事务是最新的为止。通过以一致的方式使用同一连接，可以确保使用 1-safe 或 2-safe received HotStandby 复制协议时，下一个读事务看不到上一个写事务中的已落实数据。

solidDB 会自动选择工作负载连接；负载均衡对于应用程序是自动且透明的。尤其是以执行读操作为主的应用程序可以轻松地在两个数据库服务器之间进行负载均衡，并使用两个服务器的全部处理器容量。

通过将 PREFERRED_ACCESS 连接属性设置为值 READ_MOSTLY，使用 solidDB ODBC 驱动程序来激活负载均衡；或者通过将名为 solid_preferred_access 的属性设置为值 READ_MOSTLY，使用 solidDB JDBC 驱动程序来激活负载均衡。连接至数据库的每个应用程序都可以选择使用负载均衡功能，也可以选择将主数据库用于所有查询（缺省值）。您还可以通过改变服务器的事务隔离级别，在会话或事务级别动态地控制负载均衡。

相关概念:

第 86 页的『静态负载均衡配置』

第 87 页的『动态控制负载均衡』

使用负载均衡 (READ_MOSTLY 或 LOCAL_READ) 时，您可以通过程序将指定的工作负载服务器从辅助服务器更改为主服务器。

第 72 页的 4.2.1, 『定义“透明连接”连接』

将使用非标准 ODBC 连接字符串设置或 JDBC 连接属性来指定“透明连接”。

1.1.7 HotStandby 和 SMA

通过 solidDB HotStandby 组件可使 SMA 服务器节点具有高可用性。

在使用 HotStandby 的 SMA 安装中，每个节点上可以有一个或多个 SMA 应用程序。可以将应用程序与数据库的连接作为常规 SMA 连接 (SMA 基本连接) 或透明连接 SMA 连接 (SMA TC) 进行配置。通过这两种连接类型，主节点上的应用程序使用 SMA 连接在本地执行读取和写入操作，并且辅助节点上的应用程序使用 SMA 连接在本地执行读取操作。另外，通过 SMA TC 连接，可在主服务器上利用网络连接从辅助节点上的应用程序执行写入事务。而且，如果已利用 SMA TC 连接启用了负载均衡选项，那么应用程序可以活跃到活跃的方式进行运作；在每个节点上，数据库访问的完整功能可用。

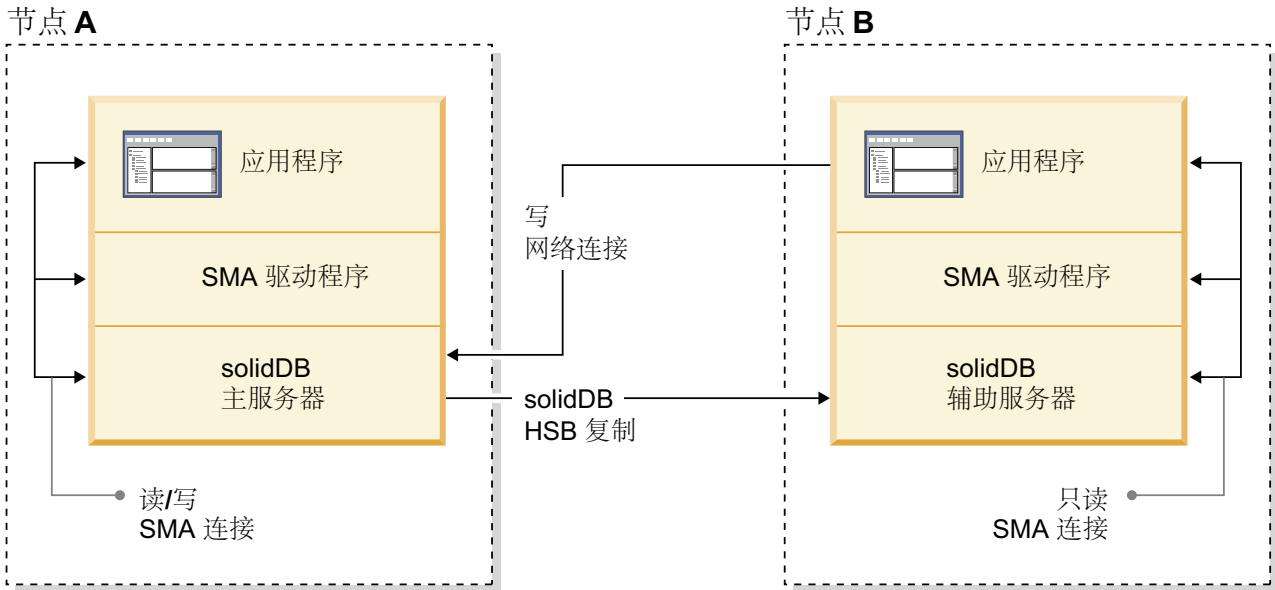


图 6. 使用 HotStandby 的 SMA 透明连接的体系结构

通过 SMA TC，每个节点上的应用程序必须能够利用 SMA 连接来连接到本地服务器，并利用基于网络的连接来连接到远程服务器。

故障转移和转换处理

- 只要其中一个服务器处于 PRIMARY ACTIVE、PRIMARY ALONE 或 STAND-ALONE 状态，那么在进行切换和故障转移之后，仍会保持连接句柄。
- 如果 SMA 服务器失败，那么应用程序可能也会失败。要确保这种故障情况下的高可用性，您的系统需要包括应用程序级别的故障转移机制，以将应用程序所提供的服务从发生故障的应用程序实例移至另一个应用程序实例。

1.1.8 HotStandby 和高级复制

可以将 solidDB HotStandby 组件与 solidDB 高级复制技术结合起来使用。高级复制技术将定期进行双向数据同步，从而允许您创建一个同时包含主服务器和副本服务器的分布式系统。借助 HotStandby，可以使分布式系统的任何数据库服务器具备高可用性。

第 17 页的图 7 显示了一个简单的分布式系统，它包含一个主数据库和两个副本数据库。每个副本数据库至少包含主数据库的一部分数据。借助 HotStandby 复制，每个数据库服务器都已经具备容错能力。将在数据库服务器层次结构的主服务器之间进行高级复制。如果任何主数据库服务器发生问题，那么失败的节点可以进行 HotStandby 故障转移，从而使该节点的辅助服务器成为新的主服务器。现在就可以继续对新的主服务器进行高级复制。

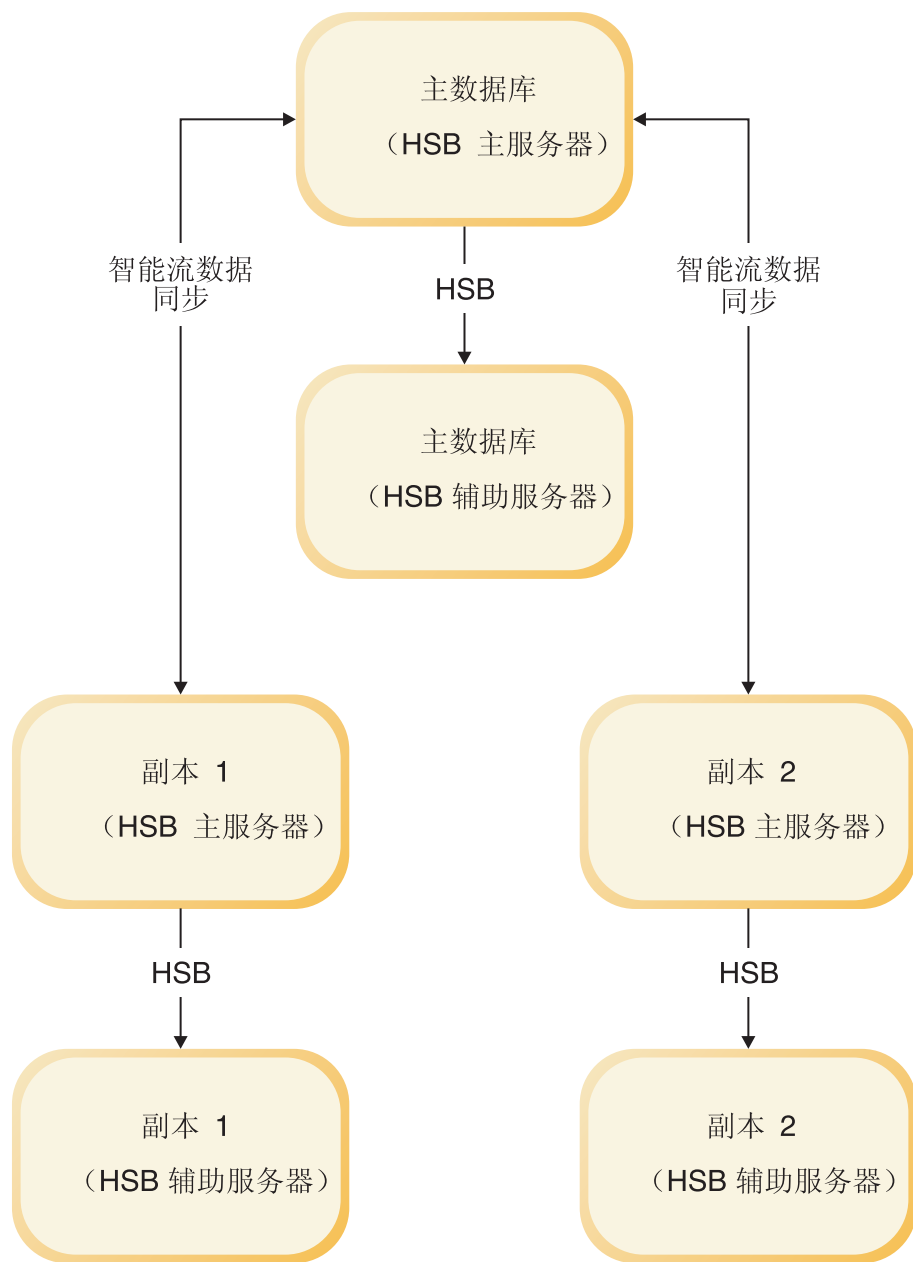


图 7. 由 HotStandby 以及主服务器和副本服务器组成的方案

1.2 性能和 HotStandby

调整 HotStandby 系统的性能时，需要进行多方面的考虑。使用 HA（冗余）系统时，以可靠且一致的方式将已更改的数据复制到另一个节点的影响起重要作用。

本节描述了 HotStandby 设置中的性能注意事项。但是，在考虑如何调整 HotStandby 设置之前，请记住优化 SQL 查询和模式设计。常常可以通过改进 SQL 查询和索引等来提高性能。实际的吞吐量和响应时间取决于许多因素，其中包括（但不限于）网络速度、网络上其他流量的大小、SQL 语句的复杂程度以及每个事务的 SQL 语句数。一些常规环境因素（例如，内存量和磁盘速度）也会影响性能。

在 HotStandby 设置中，请考虑以下主要的性能元素：

- 等待时间或响应时间 - 完成单个读取或写入操作所需的最短时间。
- 吞吐量 - 双节点系统可以处理的总查询量或事务量。
- 数据安全 - 系统是否可以确保将每个事务安全地持久储存在相同节点（至磁盘）上或（通过网络）持久储存到下一个节点上？
- 故障转移时间 - 系统在发生单节点故障后，继续提供其服务的最短时间（包括错误检测时间）。
- 恢复时间 - 系统在解决故障之后恢复到 HA 状态的最短时间，以及如何自动恢复。

在 solidDB HotStandby 设置中，可以使用下列配置和设置选项来优化 HotStandby 性能：

- 要在单一故障之后保留事务时，自适应耐久性非常有用。
- 在可接受失败期间发生次要事务丢失时，1-Safe 复制协议非常有用。
- 2-Safe 复制协议和适用的 2-Safe 确认策略会强制使安全达到最高程度。
- 负载均衡会将只读事务定向至辅助服务器。

借助 HotStandby 中的负载均衡功能，客户机可以连接至辅助服务器并执行只读操作。在某些情况下，通过将只读客户机连接至辅助服务器并在辅助服务器中执行读操作，可以分散负载并提高系统的总体性能。负载均衡对于诸如报告生成查询或“数据仓储”查询之类的工作很有用，在这些查询中，您希望读取若干条记录，但又不希望更改其中的任何一条记录。

- 内部并行性

为了确保让系统利用并行性，应考虑将事务分散在多个连接上，而不是通过同一个连接来提交所有事务。

当您使用 HotStandby (HSB) 组件时，每个包含写操作的事务都将执行两次，在主服务器上执行一次，在辅助服务器上又执行一次。随后，在某些情况下，单个事务在使用 HSB 时所花的时间可能是未使用 HSB 时的两倍。但是，这并不意味着总体吞吐量会降低 50%。服务器具有很高的并行度，当辅助服务器在处理一个事务时，主服务器可以处理另一个事务。

注：并行运行的查询越多，服务器需要的内存就越多。因此，通过添加连接和并行运行查询并不一定会增大吞吐量，尤其是在没有大量内存的系统中。您可能必须通过试验才能找到同时运行的最佳查询数。

总之，除非您需要尽可能获得最高级别的安全性，否则可以使用以下配置设置来提高性能：

- 通过将 **Logging.DurabilityLevel** 设置为 2 来使用自适应记录。
- 通过将 **HotStandby.2SafeAckPolicy** 设置为 1 来使用 2-safe received 方式。

即使您使用不太安全的设置（自适应耐久性和 2-safe received 方式），您仍然会受到 HotStandby 的保护，除非至少发生了两个故障（例如当两个服务器几乎同时宕机时）。要防止发生电源故障，必须将每个服务器都连接至不间断电源 (UPS)。此外，与任何数据库系统一样，应该备份重要数据并将它们归档在一个单独的位置。HotStandby 并不能代替您备份数据。

提示：可以在 HSB 对的任一服务器上运行备份 (**ADMIN COMMAND backup**)。通常，辅助服务器上具有更多资源可用于创建备份。

相关概念:

第 63 页的 3.5, 『性能调整』

1.3 高可用性控制器 (HAC)

高可用性控制器 (HAC) 是用于 solidDB HotStandby 配置的一个自动冗余管理程序。它通过检测故障、执行故障转移到备用单元并在必要时重新启动失败进程来维护数据库服务的可用性。

数据库节点中发生硬件问题、数据库进程失败或者 HSB 链路断开都可能会导致故障。HAC 将监视服务器的 HSB 状态。当发生故障时, HAC 将确保未受此故障影响的服务器充当主服务器角色, 并确保此服务器已准备好接受事务负载。

换句话说, HAC 充当一个看守程序的角色。在 HAC 的实现中, solidDB 的事件机制用来监视服务器状态。每当 HSB 服务器的状态更改时, 它就会向 HAC 发送一个事件, 然后 HAC 将推断可能需要使用 HotStandby 管理命令 API 执行哪些操作。

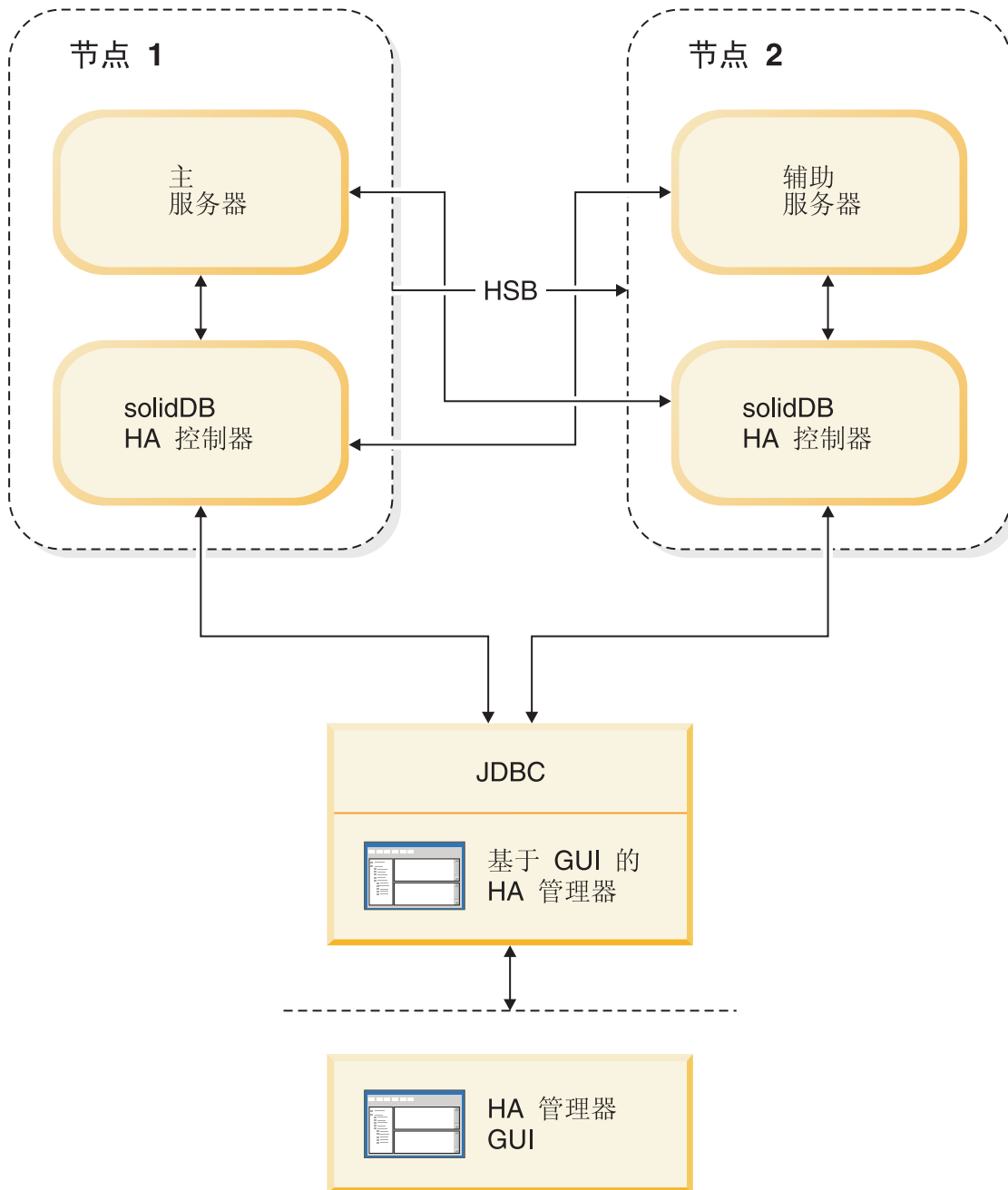


图 8. 高可用性控制器体系结构

HAC 有以下两个主要用途:

1. 可以将 HAC 用作看守程序来自动维护数据库服务的可用性。在此方式下（称为“自动”方式），HAC 将执行下列操作:
 - 启动、重新启动和终止数据库服务器进程（可选）
 - 监视服务器的状态以及它们之间的 HSB 链路
 - 推断必需执行的操作
 - 执行操作

2. HAC 还可以用作 HSB 系统的监视器，以监视 HSB 服务器的状态更改（由其他人触发）和报告该系统的状态。此方式称为“管理”方式。在此方式下，HAC 不会执行任何 HSB 状态过渡或修改 HSB 系统。

可通过 `solidhac.ini` 配置文件来配置高可用性控制器。在启动 HAC 之前，该配置文件必须在 HAC 工作目录中。可以使用命令行选项 `-c` 来指定工作目录。

`solidDB` 软件包包括 HAC 配置文件模板和示例，该模板包括可用的所有配置参数和注释。`solidhac_template.ini` 位于 `samples\hac\` 目录的根目录中。

还为 HAC 提供了一个称为高可用性管理器 (HAM) 的图形用户界面 (GUI) 组件样本。它包含在样本目录 `samples\hac\` 中。HA 管理器采用与 HA 控制器相同的方式进行配置。`samples\hac\` 目录包含适用于高可用性管理器的配置文件样本 (`HAManager.ini`)。

1.3.1 常见的故障

HAC 将监视 HotStandby 服务器的运行状况和状态。在故障情况下（例如，数据库进程故障或者计算机节点故障），HAC 将执行故障转移和其他必需的状态过渡以维护数据库服务的最佳可用性。

对于所考虑的所有故障，假定它们都是在一种正常并且完全可运行的状况（即，两个 HSB 服务器分别处于 PRIMARY ACTIVE 和 SECONDARY ACTIVE 状态）下发生的。HAC 仅处理单个故障。换句话说，假定在系统从前一个故障恢复之前不能再发生故障。但是，HAC 可以处理许多预定义的多故障情况。

只要是考虑单个故障，HAC 就会维护几乎不间断的数据库服务。如果发生了多个故障，HAC 将尝试避免出现错误的系统状态（例如，双重主服务器）。

HAC 可以处理的故障包括：

- 单个故障
 - 主 (ACTIVE) 数据库服务器进程失败
 - 辅助 (ACTIVE) 数据库服务器进程失败
 - 主节点失败
 - 辅助节点失败
 - 如果使用了“外部引用实体”，那么 HAC 还可以处理 HotStandby 链路故障，即，两个 HotStandby 数据库进程之间的连接断开。有关“外部引用实体”的更多信息，请参阅第 22 页的 1.3.3，『外部引用实体 (ERE)』。
 - 服务器对外部客户机无响应
- 双重故障
 - 在从前一个故障恢复时，HAC 将识别主数据库与辅助数据库之间同步过程中发生的错误。
 - HAC 也会注意一些不常见的故障，例如，在前一个故障之后，当服务器建立 HSB 链路时发生的服务器进程故障。

有关故障和恢复方案的详细描述，请参阅第 101 页的 5，『处理高可用性控制器 (HAC) 时发生的故障』。

1.3.2 控制数据库服务器进程

可以配置 HAC 来启动数据库进程和重新启动失败进程。

HAC 实例与本地数据库服务器断开连接时，它会调用 `solidhac.ini` 配置文件中指定的启动脚本。此脚本由用户提供。随程序包一起提供了一个示例脚本。

要点: HAC 实例假定服务器正在运行，并且在启动脚本终止时将作出响应。由于 HAC 不会处理服务器启动时发生的故障，因此，除非服务器接受连接，否则该脚本不能退出。

每当数据库服务器发生故障或者因其他某些原因而消失时，HAC 就会重新启动数据库服务器，以下情况除外：

- 在操作系统的进程列表中提供了数据库进程。
- 使用 HA 管理器关闭了数据库服务器。
- 使用 **ADMIN COMMAND 'hsb shutdowndb'** 命令关闭了数据库服务器。

1.3.3 外部引用实体 (ERE)

当数据库节点之间的通信链路发生故障时，两个数据库服务器可能都认为对方已关闭，这是您必须准备处理的一种故障情况。此故障会导致出现双重主服务器的情况（这就是所谓的裂脑），并且稍后使数据库同步时可能会丢失事务。要避免 HAC 作出错误决定，您可以使用网络引用设备（外部引用实体 (ERE)）来检查网络运行状况。例如，如果一台计算机中的网络适配器发生故障，那么 HAC 可以检测到这种情况，并且能够设置正确的数据库节点继续作为主数据库，而另一个数据库节点继续作为辅助数据库。

如果使用了 ERE，那么 HAC 将通过对 ERE 执行 ping 操作来检查 HotStandby 节点与 ERE 设备之间的物理链路的状态。如果与离得最近的 ERE 的物理链路不起作用，那么本地 HAC 会将本地服务器设置为 **SECONDARY ALONE** 状态。如果离得最近的物理链路起作用，并且没有连接可用于其他服务器，那么本地 HAC 将断定本地服务器就是继续提供服务的服务器并将它设置为 **PRIMARY ALONE**。因此，HotStandby 节点（与它相对应的 HotStandby 节点和离得最近的 ERE 之间的连接断开）成为辅助服务器。这样，就可以防止在发生网络故障的情况下出现双重主服务器（“裂脑”）的情况。

有关为 ERE 配置 HAC 的更多信息，请参阅第 42 页的 3.3，『配置 HA 控制器和 HA 管理器』。

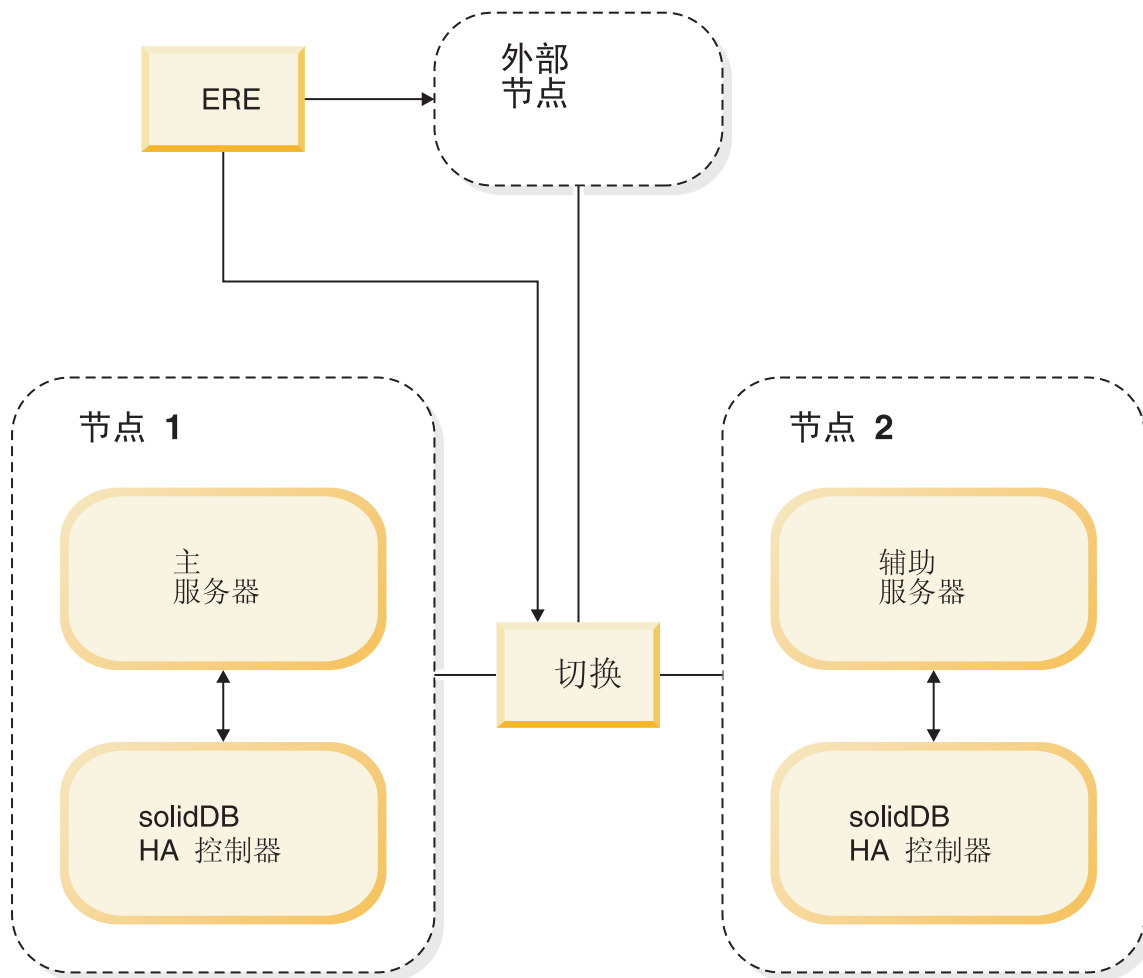


图 9. “外部引用实体”组件

上图说明了 ERE 可能用于的两个位置:

- 组开关
- 位于集群之外的网络中的任何计算机。如果在集群中使用了冗余网络（即，使用了两组完全相同的网络控制器、电缆和交换机），请在集群外部定义 ERE。

要点: 如果认为 HotStandby 链路不可靠（包括所有使用了 ERE 的情况），那么必须将以下 HotStandby 服务器参数设置为出厂值:

HotStandby.AutoPrimaryAlone=no

ERE 必须使用保持活动消息使用的同一 HSB 链路。

1.3.4 在 HAC 中联网

要优化 HotStandby 设置，请在每个 HSB 服务器中使用单个逻辑网络接入（IP 地址）来完成所有通信。服务器可以将不同的端口号用于不同用途。

您可以使用多个或冗余的较低层的网络组件（网络接口卡、电缆和交换机），这些组件在网络接口 API 层是透明的。

仅使用一个逻辑网络接入时，HAC（带有 ERE）会检测对 HSB 事务复制和数据库客户机应用程序都有影响的网络中断。如果使用了单独的接口，那么 HAC 无法检测数据库客户机通信中的故障，这是因为 HAC 只监视 HSB 链路的运行状况。

但是，无论底层的网络接入实现方案如何，如果网络接入因任何原因而被认为不可靠，就可以使用 HAC 的 ERE 功能部件。

1.3.5 HAC 日志记录

HAC 将日志记录写入 HAC 工作目录中的 hacmsg.out 文件。

该日志包含有关以下 HAC 操作的信息：

- 警告
- 不可恢复错误
- 与配置相关的信息
- 与初始化相关的信息
- 所有输入事件
- HotStandby 状态更改
- 用户命令，导致系统中发生更改
- HAC 状态更改
- HAC 方式更改（自动/管理）
- 事件，导致系统中发生状态更改

HAC 日志文件的最大大小为 64 MB。当超过大小限制时，hacmsg.out 会重命名为 hacmsg.bak，并且会创建一个新的 hacmsg.out。这些文件最多包含 128 MB 的最新日志。

1.4 高可用性管理器（样本）

solidDB 软件包包括称为高可用性管理器 (HAM) 的样本程序。HAM 是一个带有 GUI 的 Java 程序，用于显示 HotStandby 服务器的状态和 HAC 的状态。它还提供了用于管理 HAC 的基本功能，例如，通过切换 HotStandby 服务器的角色以及暂挂和恢复 HAC 来管理 HAC。

下图显示了 HAM 界面。



图 10. 高可用性管理器

高可用性管理器通过 `HAManager.ini` 配置文件进行配置。

可以使用高可用性管理器来执行下列操作：

- 在 HotStandby 服务器之间执行切换
- 在“自动”与“管理”HAC 方式之间切换
 - 在“管理”方式下，HAC 仅监视 HSB 集群，并且用户可以在 HSB 服务器上执行管理任务
 - 在“自动”方式下，HAC 充当看守程序，用于处理故障和维护数据库服务的可用性。
- 关闭 HotStandby 数据库服务器进程
- 启动 HotStandby 数据库服务器进程
- 暂挂高可用性控制器
- 恢复高可用性控制器

注：

仅当 `solidhac.ini` 配置文件中的 `HACController.EnableDBProcessControl` 参数设置为 Yes 时，才能关闭和启动数据库服务器进程。

2 HotStandby 入门

本节提供了有关设置两个 solidDB HotStandby 服务器（一个主服务器和一个辅助服务器）的逐步指示信息。

入门步骤假定您已经在 一台或两台计算机上安装了 solidDB。

您可以使用一台计算机来设置 HotStandby 的评估配置；可以在一台计算机上运行 solidDB 的两个实例，并将其中一个实例设置为主服务器，而将另一个实例设置为辅助服务器。

2.1 HotStandby 快速启动过程

通过完成快速启动过程，可以使 HSB 系统达到为应用程序服务的状态。

关于此任务

除了 HSB 服务器之外，快速启动过程不会采用任何其他组件。例如，不需要 HAC。第 30 页的 2.2.1, 『启动和停止 HA 控制器』中描述了用于 HAC 的类似逐步完成的过程。solidDB HotStandby 程序包中还包含一个样本看守程序。要使用此样本看守程序，需要为它提供配置设置。

要设置和运行 HSB 服务器对，需要两台已联网的计算机。要设置 HotStandby 服务器（它们没有任何其他 solidDB 组件），请遵循以下过程。

过程

1. 配置主节点和辅助节点。

HotStandby 要求您至少在主服务器和辅助服务器上 `solid.ini` 配置文件的 `[HotStandby]` 节中配置下列参数：

- **HSBEnabled=Yes**

如果您省略了 **HSBEnabled** 参数，或者在您打算用作 HotStandby 服务器的服务器中将此参数的值设置为 `no`，那么当此服务器启动时将是一个非 HotStandby 服务器。

- **Connect=connect string for the opposite HSB server**

如果您省略 **Connect** 参数，那么服务器将作为 HotStandby 服务器启动，但是您必须先使用 `ADMIN COMMAND` 提供连接字符串，然后服务器才能进行连接。

2. 采用您将用来启动任何 solidDB 服务器的方式来启动两个 HSB 服务器。

这两个服务器从它们自己的 `solid.ini` 文件中读取 HotStandby 配置信息。这两个服务器在启动之后的状态都是 `SECONDARY ALONE`。

3. 选择将成为主服务器的服务器，并通过发出以下命令将所选服务器的状态切换为 `PRIMARY ALONE`：

```
ADMIN COMMAND 'hsb set primary alone';
```

提示: 在 HotStandby ADMIN COMMAND 中, 可以使用缩写 **hsb** 来代替 **hotstandby**。

- 通过在任一服务器中发出以下命令将主服务器连接至辅助服务器:

```
ADMIN COMMAND 'hsb connect'
```

如果要验证连接是否成功, 请发出下列命令:

```
ADMIN COMMAND 'hsb state'
```

主服务器会答复其状态为 PRIMARY ACTIVE。但是, 如果主服务器的状态是预期之外的其他状态 (例如, PRIMARY ALONE), 那么可以通过发出以下命令来检查 **hsb connect** 的状态:

```
ADMIN COMMAND 'hsb status connect'
```

如果获得的结果为 ACTIVE, 那么连接进程仍然处于活动状态。如果结果为 BROKEN, 那么在连接服务器之前, 必须使它们的数据库同步。

- 通过在主服务器中发出下列命令使数据库同步:

```
ADMIN COMMAND 'hsb netcopy'
```

通过发出下列命令来检查数据库复制进程的状态:

```
ADMIN COMMAND 'hsb status copy'
```

- 只要 **hsb status copy** 的结果为 SUCCESS, 就说明数据库之间已同步, 因此可以连接服务器了。在任一服务器中重新发出以下命令:

```
ADMIN COMMAND 'hsb connect'
```

按照步骤 4 中的指示, 验证连接是否成功。

- 开始使用应用程序。

相关概念:

第 37 页的 3.2.1, 『定义主节点和辅助节点 HotStandby 配置』

要配置 HotStandby, 至少必须在这两个节点上将 **HotStandby.HSBEnabled** 参数设置为 yes, 并定义这两个节点之间的连接设置。

第 37 页的 3.2, 『配置 HotStandby』

通过使用主节点和辅助节点处的 solid.ini 配置文件来配置

HotStandby。[HotStandby] 一节包含特定于 HotStandby 的所有配置参数。还必须设置其他节和参数 (例如, **Com.Listen** 参数)。

2.2 使用 HAC 的 HotStandby 快速启动过程

本节描述了使用 solidDB 高可用性控制器 (HAC) 的 HotStandby 的快速启动过程。

关于此任务

此过程与第 27 页的 2.1, 『HotStandby 快速启动过程』类似。但是, 此过程不仅仅只是设置两个 HSB 服务器, 它还会指导您设置一个高可用 HSB 系统, 其中的高可用性由高可用性控制器 (HAC) 来强制实现。通过遵循以下过程, 您将准备好一个具有容错能力的 HSB 系统来为应用程序服务。

要设置和运行高可用性 HSB 系统, 需要两台已联网的计算机。将在每个 HSB 服务器上设置 HAC 的一个实例。

过程

1. 请按照 第 27 页的 2.1, 『HotStandby 快速启动过程』中的步骤来配置主服务器和辅助服务器。

完成之后, 其中一个服务器处于 PRIMARY ACTIVE 状态, 另一个服务器处于 SECONDARY ACTIVE 状态。

2. 在主服务器和辅助服务器上配置 HAC。HAC 从工作目录中的 solidhac.ini 文件读取其配置。需要以下必需配置参数:

[HAController] 节

- **Listen**=<listen address, 'tcp' and chosen port #>
- **Username**
- **Password**
- **DBUsername**
- **DBPassword**

[LocalDB] 节

- **Connect**=<connect address, protocol, ip/hostname, port #>
- **StartScript** (如果 **EnabledDBProcessControl**=Yes (这是缺省值), 那么此参数是必需的)

[RemoteDB] 节

- **Connect**=<connect address, protocol, ip/hostname, port #>

3. 按照第 30 页的 2.2.1, 『启动和停止 HA 控制器』中的指示启动两个节点上的 HAC。

HAC 会自动根据服务器的先前角色以及日志位置来查明哪个服务器将成为新的主服务器。在第 61 页的 3.4.9, 『选择使哪个服务器成为主服务器』中更详细地描述了该机制。在某些特殊情况下 (例如, 当使用空数据库启动时), 两个服务器都可以作为新的主服务器的候选者。在这些情况下, 如果 solidhac.ini 的 [LocalDB] 节中的 **PreferredPrimary** 参数设置为 Yes, 那么 HAC 将选择它的本地服务器。

4. 在完成第二个步骤之后, 应当有一个服务器处于 PRIMARY ACTIVE 状态, 另一个服务器处于 SECONDARY ACTIVE 状态。您可以通过在辅助服务器上发出下列命令来切换角色:

```
ADMIN COMMAND 'hsb switch primary'
```

或者, 在主服务器上执行以下命令:

```
ADMIN COMMAND 'hsb switch secondary'
```

此后, HAC 就会在必要时切换服务器的角色。至少有一个服务器执行读/写事务。

5. 开始使用应用程序。

相关概念:

第 42 页的 3.3, 『配置 HA 控制器和 HA 管理器』

高可用性控制器 (HAC) 部署在每个 HotStandby 服务器节点上; 可以通过 solidhac.ini 配置文件对其进行配置。高可用性管理器 (HAM) 是通过 HAManager.ini 配置文件来配置的。

相关参考:

第 120 页的 A.3, 『高可用性控制器 (HAC) 参数』

本节描述 solidhac.ini 配置文件中的高可用性控制器 (HAC) 配置参数。

第 126 页的 A.5.2, 『solidhac.ini 配置文件』

高可用性控制器 (HAC) 配置文件 (solidhac.ini) 的样本摘录。

2.2.1 启动和停止 HA 控制器

在 HAC 工作目录中必须具有 solidhac.ini 配置文件, 才能启动 HAC。有关配置 HAC 的简要描述, 请参阅第 28 页的 2.2, 『使用 HAC 的 HotStandby 快速启动过程』。有关配置 HAC 的更多信息, 请参阅第 42 页的 3.3, 『配置 HA 控制器和 HA 管理器』。要了解参数列表以及详细的参数描述, 请参阅第 120 页的 A.3, 『高可用性控制器 (HAC) 参数』和 第 126 页的 A.5.2, 『solidhac.ini 配置文件』。

注: 根据您使用的平台不同, HAC 二进制文件名为 solidhac 或 solidhac.exe。在示例中, 为了更清楚明了, 将使用 solidhac.exe。同样, 样本 solidDB 启动脚本的名称为 start_solid.sh 或 start_solid.bat。在示例中, 将使用 start_solid.bat。

启动 HAC

用于启动 HAC 的语法是:

```
solidhac.exe [-c working_directory | -?]
```

? 参数或者除了 -c 之外的任何其他参数将输出用法消息。

HAC 启动之后, 它就会开始侦听由 solidhac.ini 配置文件中的 **HAController.Listen** 参数所指定的端口。例如, 该端口用于在 HAC 与 HA 管理器之间传输命令, 以及用于发出特定于 HAC 的 ADMIN COMMAND。

停止 HAC

要发出该命令, 必须先连接至 HAC, 并指定 solidhac.ini 配置文件中用 **HAController.Listen** 参数定义的端口。例如, 您可以使用 **solsql** 或 ODBC 接口来连接至 HAC。

可以通过执行下列命令来停止 (终止) HAC:

```
ADMIN COMMAND 'hacontroller shutdown'
```

示例: 启动 HAC

如果 solidhac.exe 位于 c:\solid\hac 目录下 (这也是当前目录), 并且使用 c:\solid\run\server1 作为 HAC 的工作目录, 请使用下列命令来启动 HA 控制器:

```
solidhac.exe -c c:\solid\run\server1
```

或者

```
solidhac.exe -c ..\run\server1
```

相关参考:

第 152 页的 C.2, 『高可用性控制器命令 (ADMIN COMMAND)』

2.3 启动顺序总结

下图和下表以并排方式提供了带有 HAC 的安装和不带 HAC 的安装的顺序, 以便进行比较。在下图中, 带有 HAC 的安装顺序位于左边, 不带 HAC 的安装顺序位于右边。图下方的表中说明了两种安装类型的图中的每个带有编号的步骤。

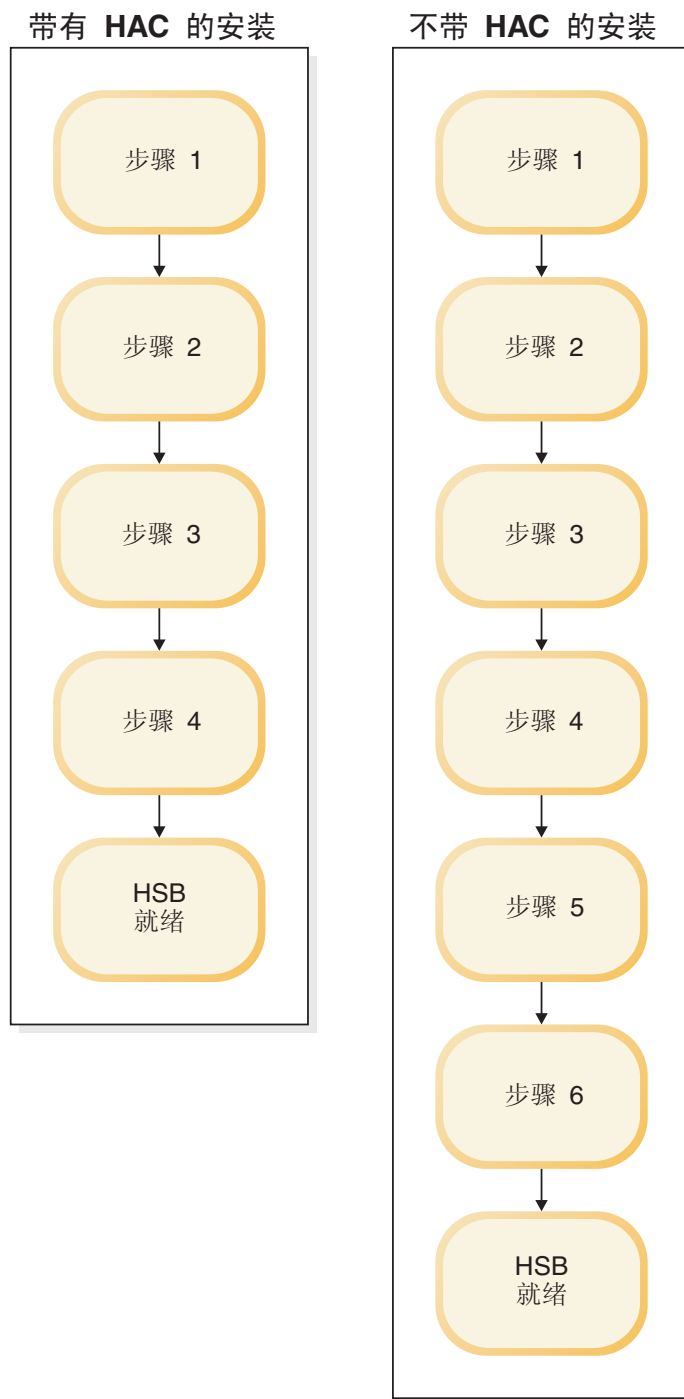


图 11. 启动顺序总结

表 5. 安装顺序步骤

带有 HAC 的安装	不带 HAC 的安装
步骤 1. 配置 HSB 服务器	

表 5. 安装顺序步骤 (续)

带有 HAC 的安装	不带 HAC 的安装
步骤 2. 配置 HA 控制器	步骤 2. 在两个节点中启动 HSB 服务器
步骤 3. 在两个节点中启动 HA 控制器	步骤 3. 将数据库服务器的状态切换到 PRIMARY ALONE
步骤 4. 必要时, 切换 HSB 角色	步骤 4. 将主服务器连接至辅助服务器
	步骤 5. 如果连接失败, 那么请启动从主数据库复制到辅助数据库
	步骤 6. 执行 netcopy 之后, 重新进行连接
HSB 已就绪。您可以开始使用应用程序了。	

2.4 HotStandby 样本

solidDB 服务器软件包包括一些样本, 这些样本用于演示使用 HAC 或样本看守程序实现的 HotStandby 组件。

使用 HAC 的 HotStandby

可在 `samples\hac` 目录中找到使用高可用性控制器 (HAC) 的 HotStandby 样本。可在下列文件中找到有关该样本的详细信息: `samples\hac\readme.txt`

使用看守程序的 HotStandby 样本

可在 `samples\hsb` 目录中找到使用看守程序的 HotStandby 样本。可在下列文件中找到有关该样本的详细信息: `samples\hsb\readme.txt`

3 管理和配置 HotStandby

本节描述如何使用与 HotStandby 相关的参数及 ADMIN COMMAND 来维护 HotStandby 安装（包括 HSB 服务器和 HAC 实例）。

此描述是对 一节中信息的补充。

3.1 HotStandby 管理基础

与 HotStandby 相关的组件使用配置参数和 ADMIN COMMAND 进行管理。

配置参数

这些参数按照 solid.ini 配置文件中的节类别进行分组。使用 HotStandby 时，您主要使用 solid.ini 配置文件的 [HotStandby] 节。其他节可用于配置 solidDB 服务器的总体行为。

将使用 solidhac.ini 配置文件来配置高可用性控制器 (HAC)。solidhac.ini 配置文件中的参数也会按照不同的节类别进行分组。

将使用 HAManager.ini 配置文件来配置高可用性管理器 (HAM)。HAManager.ini 中的参数标识 HA 控制器实例，以便高可用性管理器对它们具有访问权。

可以采用下列方法来更改配置参数：

- 手动编辑配置文件 solid.ini、solidhac.ini 和 HAManager.ini。

注：服务器仅在启动期间才读取配置文件，因此，对任何配置文件所作的任何更改都将在下一次启动相应程序之后才会生效。

- 使用以下 ADMIN COMMAND 语法来动态地更改正在运行的 solidDB 服务器的设置：

```
ADMIN COMMAND 'parameter section_name.param_name=value';
```

例如：

```
ADMIN COMMAND 'parameter hotstandby.2SafeAckPolicy=2';  
ADMIN COMMAND 'parameter com.listen="tcp sf_server 1315"';
```

注：所有配置参数都不能动态地更改。

HotStandby ADMIN COMMAND (HotStandby API)

HSB API 是作为 solidDB SQL 的语法扩展进行提供的，它采用了 ADMIN COMMAND 的格式：

```
ADMIN COMMAND hotstandby hsb-command options
```

或者

```
ADMIN COMMAND hsb hsb-command options
```

可以通过任何支持 SQL 的交互式工具（例如，`solsql`）来发出 HSB 命令，也可以通过 ODBC 或 JDBC 以编程方式发出 HSB 命令。

HotStandby 和访问权

没有特定于 HotStandby 的访问权，常规管理员访问权已足够。如果要执行 HotStandby 管理命令，那么需要 `SYS_ADMIN_ROLE` 或 `SYS_CONSOLE_ROLE` 访问权。

HotStandby 和 solidDB 工具

solidDB 数据管理工具可以与 HotStandby 服务器配合使用。

用于发出 ADMIN COMMAND 的控制台工具

您可以使用 solidDB SQL 编辑器 (`solsql`) 和 solidDB 远程控制 (`solcon`) 来发出特定于 HotStandby 的管理命令 (`ADMIN COMMAND 'hotstandby <option>'`)。

使用 `solsql` 时，必须用单引号将命令名引起来并且命令的末尾有分号。例如：

```
ADMIN COMMAND 'hotstandby status connect';
```

使用 `solcon` 时，命令名不能带有 `ADMIN COMMAND` 前缀和引号，并且命令末尾不能有分号。例如：

```
hotstandby status connect
```

导入和导出工具

导入和导出工具：solidDB 快速装入器 (`solloado` 或 `solload`)、solidDB 导出 (`solexp`) 以及 solidDB 数据字典 (`soldd`) 也可以与 HotStandby 配合使用。

高可用性管理器 (HAM) 样本

solidDB 软件包还包含一个样本工具：高可用性管理器 (HAM)，该工具可用于监视 solidDB HotStandby 状态和控制 HSB 服务器及高可用性控制器 (HAC)。HAM 只能与 HA 控制器配合使用。

3.1.1 查询 HotStandby 配置参数

可以使用标准参数处理命令来查询 HotStandby 参数的值和属性。

命令为：

```
ADMIN COMMAND '[describe] parameter[section_name[.parameter_name]]';
```

例如：

```
ADMIN COMMAND 'parameter logging.durabilitylevel';
```

```
RC TEXT
```

```
-- -----
```

```
0 Logging DurabilityLevel 3 3 2
```

```
ADMIN COMMAND 'parameter hotstandby.MaxLogSize';
```

```
RC TEXT
```

```
-- -----
```

```
0 HotStandby MaxLogSize 10000000 0 0
```

结果集显示以下三个值：

- 当前值 - 动态设置此值，或者继续使用缺省值或出厂值。
- 缺省值 - 最初是从 `solid.ini` 文件中读取的，或者继续使用出厂值。
- 出厂值 - 是在产品中提供的。

3.1.2 修改 HotStandby 配置参数

通常，要更改某个参数的值，可以通过在 `solid.ini` 配置文件中更改此值，然后重新启动服务器来完成更改。但是，还可以使用 `ADMIN COMMAND` 来更改大多数 HotStandby 参数。

关于此任务

用于修改参数的语法为：

```
ADMIN COMMAND 'parameter section_name.parameter_name=value [temporary]';
```

除非使用了 **temporary** 选项，否则，对参数所作的所有更改都将保存在下一个检查点的 `solid.ini` 文件。也可以使用以下命令来执行该保存操作：

```
ADMIN COMMAND 'save parameters [file_name]';
```

缺省情况下，此命令将重写缺省 `solid.ini` 文件。通过使用 `file_name` 选项，可以将输出定向到另一个位置。

3.2 配置 HotStandby

通过使用主节点和辅助节点处的 `solid.ini` 配置文件来配置 HotStandby。[HotStandby] 一节包含特定于 HotStandby 的所有配置参数。还必须设置其他节和参数（例如，**Com.Listen** 参数）。

3.2.1 定义主节点和辅助节点 HotStandby 配置

要配置 HotStandby，至少必须在这两个节点上将 **HotStandby.HSBEnabled** 参数设置为 `yes`，并定义这两个节点之间的连接设置。

[HotStandby] 节中至少需要的配置信息是：

- **HotStandby.HSBEnabled** 参数必须设置为 `yes`。

如果 **HotStandby.HSBEnabled** 未设置为 `yes`，那么服务器将作为非 HotStandby 服务器启动，并且不使用 HotStandby 复制。

- 必须设置 **HotStandby.Connect** 参数。此参数定义用来连接至另一服务器（主服务器或辅助服务器）的网络名。如果未在 `solid.ini` 文件中设置此参数，那么服务器只能运行在不需要连接的状态下（例如 `PRIMARY ALONE`、`SECONDARY ALONE` 和 `STANDALONE`）。当服务器在运行时，可以使用 `ADMIN COMMAND 'parameter'` 命令来设置或更改此参数。
- 每个服务器的 **HotStandby.Connect** 字符串必须与另一服务器的 **Com.Listen** 字符串匹配。

示例：节点 1 最小配置

```
[Com]
Listen=tcp 2315
```

```
[HotStandby]
HsbEnabled=yes
;The server on Node 1 connects to the server on Node 2
;using the connect string 'tcp 2325':
Connect=tcp 2325
```

示例: 节点 2 最小配置

```
[Com]
Listen=tcp 2325

[HotStandby]
HsbEnabled=yes
;The server on Node 2 connects to the server on Node 1
;using the connect string 'tcp 2315':
Connect=tcp 2315
```

3.2.2 设置 HotStandby 服务器等待时间以帮助检测已断开的连接或者不可用的连接

HotStandby 服务器使用超时参数来控制它在断定现有连接已断开或者无法建立新连接之前将等待的时间。

关于此任务

超时参数包括:

- **HotStandby.PingTimeout**
- **HotStandby.PingInterval**
- **HotStandby.ConnectTimeout**

如果处于 PRIMARY ACTIVE 或 SECONDARY ACTIVE 状态的 HotStandby 服务器尝试与另一个服务器联系, 并且在指定的时间内未接收到应答, 那么其状态将更改为 PRIMARY UNCERTAIN、PRIMARY ALONE 或 SECONDARY ALONE。

过程

要控制服务器的等待时间, 可以执行下列操作:

- 设置 **PingTimeout** 参数, 以指定服务器在更改为 PRIMARY UNCERTAIN 状态之前必须等待的时间。
- 设置 **PingInterval** 参数, 以指定服务器两次发送“ping”消息(用于指示此服务器在正常运行)之间的时间间隔。
- 设置 **ConnectTimeout** 参数, 以指定服务器在尝试与另一个服务器建立新连接(例如, 通过 **ADMIN COMMAND 'hotstandby connect'** 操作来建立连接)时必须等待的时间。

PingTimeout 和 PingInterval 参数 [HotStandby]

[HotStandby] 节中的可选参数 **PingTimeout** 和 **PingInterval** 用于控制 ping 操作。

ping 操作实质上是由一个数据库服务器发送至另一个数据库服务器的一条表示“我还活着”的消息。HotStandby 设置中的 ping 系统是被动的脉动信号系统。启用该系统时, 主服务器和辅助服务器会定期互相发送 ping 消息。有关更多信息, 请参阅第 3 页的『脉动信号』。

- **HotStandby.PingTimeout** 指定一个服务器在断定另一个服务器已关闭或者不可访问之前应等待的时间。缺省值为 4000 (即, 4 秒)。
- **HotStandby.PingInterval** 指定两次 ping 操作之间的时间间隔(以毫秒计)。缺省值为 1000 (即, 1 秒)。

例如, 如果 **PingInterval** 为 10 秒, 那么服务器之间每隔 10 秒钟就会互相发送 ping 消息。如果 **PingTimeout** 为 20 秒, 而一个服务器 (S1) 在 20 秒之内未接收到来自另

一个服务器 (S2) 的 ping 消息, 那么 S1 断定 S2 已关闭或者不可访问。于是, 服务器 S1 会切换到另一种状态, 例如, 从 PRIMARY ACTIVE 状态切换到 PRIMARY UNCERTAIN 状态。

如果参数的值不相同, 那么将优先采用在执行 **hsb connect** 命令期间在主服务器中设置的值。这些值在切换期间不会更改。但是, 可以使用 **ADMIN COMMAND 'parameter'** 命令动态地更改这些值。

如果 **PingTimeout** 设置为 0, 那么将禁用 ping 操作。

Ping 操作需要很少的开销, 并且 solidDB 服务器设置为快速对丢失 ping 消息作出响应。可以将 **PingInterval** 值设置为一个相当短的时间间隔 (例如, 1 秒甚至更短)。

如果快速检测故障转移很重要, 请将 **PingTimeout** 值设置为一个相对短的时间。但是, 此值越小, 产生假警报的机会也越多。如果您的网络流量很大, 从而导致在接收到 ping 消息之前产生延迟, 那么您可能需要将 **PingTimeout** 参数设置为一个较大值以避免产生假警报。

注: 某些联网软件也具有 ping 操作, 但是 **HotStandby.PingTimeout** 和 **HotStandby.PingInterval** 参数仅适用于 solidDB 服务器 ping 操作, 而不适用于常规的网络 ping 操作。

ConnectTimeout 参数 [HotStandby]

在某些网络实现中, 连接操作可能无限期地不作出响应。一种原因可能是, 虽然远程服务器是一个已知节点, 但是在尝试连接期间不可用。通过指定连接超时值, 可以设置 HotStandby 连接操作等待与远程服务器建立连接的最长时间 (以秒计)。

使用 **HotStandby.ConnectTimeout** 参数来设置连接超时值。单位是毫秒。缺省值为 0, 它意味着不存在超时。可以将它设置为其他值, 例如:

```
[HotStandby]
; Set ConnectTimeout to 20 seconds (20000 milliseconds).
ConnectTimeout=20000
```

HotStandby.ConnectTimeout 参数与以下管理命令配合使用:

- **hotstandby connect**
- **hotstandby switch primary**
- **hotstandby switch secondary**

3.2.3 定义事务耐久性级别

事务耐久性级别使用 **Logging.DurabilityLevel** 参数进行设置。该参数具有三个不同的值: 宽松 (1)、自适应 (2) 和严格 (3) 耐久性。

“自适应”耐久性仅用于 HotStandby 服务器。“自适应”耐久性意味着:

- 如果主服务器和辅助服务器已建立连接并且都在正常运行 (处于 PRIMARY ACTIVE 和 SECONDARY ACTIVE 状态), 那么服务器将使用宽松耐久性。
- 在所有其他情况下 (例如, PRIMARY ALONE 和 STANDALONE), 服务器将使用“严格”耐久性。

有关严格耐久性与宽松耐久性之间的差别的说明, 或者有关 **Logging.DurabilityLevel** 参数的更多信息, 请参阅《IBM solidDB 管理员指南》。

3.2.4 定义 HotStandby 数据库复制操作的名称和位置

可选的 **HotStandby.CopyDirectory** 参数会定义 HotStandby **copy** 操作复制到的目录的名称和位置。

HotStandby 复制操作是通过以下命令指定的:

```
ADMIN COMMAND 'hotstandby copy [directory_name]';
```

HotStandby.CopyDirectory 参数没有缺省值, 因此, 如果在 `solid.ini` 文件中未指定目录, 那么必须在 `copy` 命令中提供该目录。如果您为 **HotStandby.CopyDirectory** 参数提供相对路径, 那么该路径将相对于主服务器的工作目录。

如果您使用 **ADMIN COMMAND 'hotstandby netcopy'** 命令来执行 HotStandby 数据库复制操作, 那么不需要 **HotStandby.CopyDirectory** 参数。

要点: 不推荐使用 `copy` 命令。请改为使用 `netcopy` 命令。

3.2.5 定义主服务器在辅助服务器发生故障期间的行为

可以使用 **HotStandby.AutoPrimaryAlone** 参数来控制主服务器在与辅助服务器失去联系之后的行为: 是自动切换到 PRIMARY ALONE 状态还是继续保持 PRIMARY UNCERTAIN 状态。

如果 **AutoPrimaryAlone** 设置为 Yes, 那么当主服务器与辅助服务器失去联系时, 主服务器自动切换到 PRIMARY ALONE 状态。PRIMARY ALONE 状态意味着主服务器继续接受事务。如果 **AutoPrimaryAlone** 设置为 No, 那么当主服务器与辅助服务器失去联系时, 主服务器自动切换到 PRIMARY UNCERTAIN 状态。

缺省情况下, **AutoPrimaryAlone** 设置为 No。

```
[HotStandby]
AutoPrimaryAlone = No
```

在 PRIMARY UNCERTAIN 状态下, 主服务器将无法接受新的事务或者落实当前处于活动状态的事务。在 HAC、看守程序或者系统管理员指示主服务器切换到 PRIMARY ALONE 状态之前, 它不会切换到此状态。

如果 **AutoPrimaryAlone** 设置为 No, 那么服务器可以通过执行 **ADMIN COMMAND 'hotstandby set primary alone'** 命令来设置为 PRIMARY ALONE 状态。**ADMIN COMMAND 'hotstandby set primary alone'** 命令不会更改配置文件中 **AutoPrimaryAlone** 的值。

如果您将缺省值更改为 Yes, 那么主服务器的状态将从 PRIMARY ACTIVE 更改为 PRIMARY ALONE 而不是更改为 PRIMARY UNCERTAIN。

3.2.6 确保主服务器和辅助服务器的参数值协调

在主服务器和辅助服务器上, 某些 HotStandby 参数或其他参数必须相同, 某些参数必须不相同。

某些参数应该在主服务器和辅助服务器上都相同。要确保在进行故障转移之后, 原始辅助服务器成为新的主服务器, 并且与旧的主服务器具有相同的行为, 那么必须将某些 HotStandby 参数设置为在主服务器和辅助服务器上具有相同的值。但是, 使用相同

的值并不是无条件的要求，系统也不会强制使用相同的值。如果您使用不同的值，虽然服务器不会发生故障，但客户机可能会看到不同的行为。

未包含在 [HotStandby] 节中、但是与此节间接相关的某些参数在主服务器和辅助服务器上也必须相同。例如，**Logging.DurabilityLevel** 参数在主服务器和辅助服务器上通常必须相同。

例如，要确保可以唯一地标识服务器并且服务器可以互相通信，某些参数在主服务器和辅助服务器上必须不相同。

在主服务器和辅助服务器上，以下 HotStandby 参数必须相同：

- [HotStandby]
 - **2SafeAckPolicy**
 - **AutoPrimaryAlone**
 - **ConnectTimeout**
 - **HSBEnabled**
 - **PrimaryAlone**（不推荐使用此参数；但是如果使用此参数，那么必须相同）
- [IndexFile]
 - **FileSpec** 必须兼容，这意味着 **FileSpec** 参数的数目必须相同，并且相应的 **FileSpec** 参数的大小必须匹配。
 - **BlockSize**
- [Logging]
 - **BlockSize**

以下参数必须不相同：

- [HotStandby]
 - **Connect**

以下参数可以相同也可以不相同，这取决于环境（例如，计算机上的磁盘驱动器配置）：

- [General]
 - **BackupDirectory**
- [HotStandby]
 - **CopyDirectory**

还有一些非 HSB 参数的设置会影响 HSB 性能。例如，可以使用 **Logging.DurabilityLevel** 参数的设置来优化 HotStandby 的性能。请参阅第 13 页的 1.1.5, 『持久性和记录』以及《《IBM solidDB 管理员指南》》中对 **Logging.DurabilityLevel** 的描述。

3.2.7 确定主服务器设置是否优先于辅助服务器设置

主服务器和辅助服务器的某些参数必须相同。如果未将这些参数设置为相同的值，那么可以预期每个服务器都会使用在服务器的 solid.ini 文件中定义的值。但是，并不是必需这样做。

甚至对于一些用于控制辅助服务器行为的参数（例如，**2SafeAckPolicy**），这些参数在主服务器上的值也是确定该行为的值。总的原则是，所有安全和耐久性参数都由主服务器控制。例如，主服务器读取它的 **2SafeAckPolicy** 值，并将此值发送至辅助服务器供辅助服务器使用。仅当辅助服务器成为主服务器时，才会使用存储在辅助服务器的 `solid.ini` 文件中的值。

对于以下参数，在主服务器上设置的值优先于在辅助服务器上设置的值：

- **HotStandby.SafenessLevel**
- **HotStandby.2SafeAckPolicy**
- **Logging.DurabilityLevel**
- **HotStandby.NetcopyRpcTimeout**

发出 `hsb connect` 命令时，将优先使用以下主服务器参数：

- **HotStandby.PingTimeout**
- **HotStandby.PingInterval**

3.3 配置 HA 控制器和 HA 管理器

高可用性控制器 (HAC) 部署在每个 HotStandby 服务器节点上；可以通过 `solidhac.ini` 配置文件对其进行配置。高可用性管理器 (HAM) 是通过 `HAManager.ini` 配置文件来配置的。

HA 控制器

HAC 配置文件 `solidhac.ini` 必须位于 HAC 工作目录下。`solidhac.ini` 配置文件中的参数归入下列各节：

- [HAController]
- [LocalDB]
- [RemoteDB]
- [ERE]

有关这些参数的描述，请参阅第 120 页的 A.3，『高可用性控制器 (HAC) 参数』

所有配置参数也显示在 第 126 页的 A.5.2，『`solidhac.ini` 配置文件』中的 `solidhac.ini` 示例文件中。

HA 管理器

HAM 配置文件 `HAManager.ini` 必须位于 HAM 工作目录下。

有关这些参数的描述，请参阅第 125 页的 A.4，『高可用性管理器 (HAM) 配置参数』

所有配置参数也显示在 第 129 页的 A.5.3，『`HAManager.ini` 配置文件』中的 `HAManager.ini` 示例文件中。

3.4 使用 ADMIN COMMAND (HotStandby API) 管理 HotStandby

HotStandby API (HSB API) 用于监视和控制服务器进程。例如, solidDB 高可用性控制器 (HAC) 使用 HSB API。另一个示例是包含在 solidDB 服务器软件包中的“看守程序”样本程序。

HSB API 是作为 solidDB SQL 的语法扩展进行提供的, 它采用了 ADMIN COMMAND 的格式:

```
ADMIN COMMAND hotstandby hsb-command options
```

或者

```
ADMIN COMMAND hsb hsb-command options
```

可以通过任何支持 SQL 的交互式工具 (例如, `solsql`) 来发出 HSB 命令, 也可以通过 ODBC 或 JDBC 以编程方式发出 HSB 命令。

可以使用 HSB 命令来编写自己的应用程序, 以管理 solidDB 中的高可用性 (例如, 为了实现与外部集群管理软件的集成)。

3.4.1 管理任务概述

HotStandby 管理包括诸如恢复和维护之类的任务。

表 6. 管理任务

主题	描述	页面
执行 HotStandby 恢复和维护任务	描述在发生系统故障 (原因是通信链路中断或者 HotStandby 服务器不可运行) 时要执行的 HotStandby 任务。这些任务包括: <ul style="list-style-type: none">• 切换服务器状态• 停止 HotStandby 操作• 使主服务器与辅助服务器同步• 连接 HotStandby 服务器	第 44 页的 3.4.2, 『执行 HotStandby 恢复和维护』
通过网络将主数据库复制到新的辅助服务器	描述当 HotStandby 配置中新增加了远程服务器 (即, 它是新的辅助服务器) 或者远程服务器数据已损坏并且必须替换时如何创建数据库的远程 (网络) 副本。	第 53 页的 『通过网络将主数据库复制到辅助服务器』
检查 HotStandby 状态	描述如何检查主服务器和辅助服务器的 HotStandby 状态信息。	第 58 页的 3.4.7, 『检查 HotStandby 状态』
验证 HotStandby 服务器状态	描述如何检查 HotStandby 服务器的状态 (是主服务器、辅助服务器还是独立服务器)。	第 60 页的 3.4.8, 『验证 HotStandby 服务器状态』
将 HotStandby 服务器更改为非 HotStandby 服务器	描述如何将为 HotStandby 配置的服务器设置为一个正常的非 HotStandby 服务器。	第 62 页的 3.4.10, 『将 HotStandby 服务器更改为非 HotStandby 服务器』

3.4.2 执行 HotStandby 恢复和维护

在发生系统故障之后或者在服务器维护期间，您可能必须执行一些 HotStandby 任务，例如，切换服务器状态、停止 HotStandby 操作、使主服务器与辅助服务器同步以及连接 HotStandby 服务器。

过程

1. 执行下面的某些或者所有操作：

a. 切换服务器状态。

这包括将主服务器设置为 PRIMARY ALONE 状态，它在这种情况下会继续将事务累积在事务日志中（以便稍后可以将它们发送至辅助服务器）或者关闭 HotStandby。

b. 使服务器同步，以确保主数据库和辅助数据库完全相同。

c. 如果通信链路因某种原因而中断，那么请将主服务器连接至辅助服务器。

2. 可以按如下所示使用 HAC 和 HA 管理器来执行相同的步骤：

a. 按 HA 管理器中的**切换按钮**。如果需要关闭服务器，那么请按 HA 管理器中的**关闭按钮**。

b. 通过按 HA 管理器中的**管理按钮**将 HAC 实例设置为“管理”方式。

c. 通过按 HA 管理器中的**自动按钮**将 HAC 实例设置为“自动”方式。

下一步做什么

有关如何将应用程序重新连接至辅助数据库或主数据库的详细信息，请参阅第 90 页的 4.3.1，『从应用程序重新连接至主服务器』。

要点：

如果使用了 HAC，请使用 HA 管理器来执行管理步骤，或者在启动管理任务之前将 HAC 实例设置为“管理”方式。

3.4.3 切换服务器状态

HotStandby 组件要求在必要时自动切换或者由用户手动切换服务器状态。

在生产使用中，通过使用自动切换状态（即，执行故障转移）来选择服务器状态。例如，自动状态切换可以是 solidDB 高可用性控制器 (HAC) 的职责。

切换意味着颠倒主服务器与辅助服务器的角色（前提是它们都在运行）。切换可以用于各种维护目的。

故障转移是当主服务器失败时，辅助服务器所采取的操作，以充当主服务器的角色。

执行切换

可通过在辅助服务器处发出 ADMIN COMMAND 'hotstandby switch primary'; 命令，或在主服务器处发出 ADMIN COMMAND 'hotstandby switch secondary'; 命令来切换服务器的角色。

无论两个服务器之间是否建立了连接，都可以使用 **switch** 命令。如果服务器之间已建立连接，那么它们的状态将颠倒；旧的辅助服务器将成为新的主服务器，而旧的主服务器将成为新的辅助服务器。如果服务器之间未建立连接，那么旧的辅助服务器将成为新的主服务器，而另一个服务器的状态保持不变。

下图说明了当服务器之间建立了连接的情况下发出 **hsb switch secondary** 或 **hsb switch primary** 命令时将发生的情况。只能在处于 **SECONDARY** 状态（例如，**SECONDARY ACTIVE**）的服务器上发出 **hsb switch primary** 命令，而 **hsb switch secondary** 命令只能用在处于 **PRIMARY** 状态（例如，**PRIMARY ACTIVE**）的服务器上。

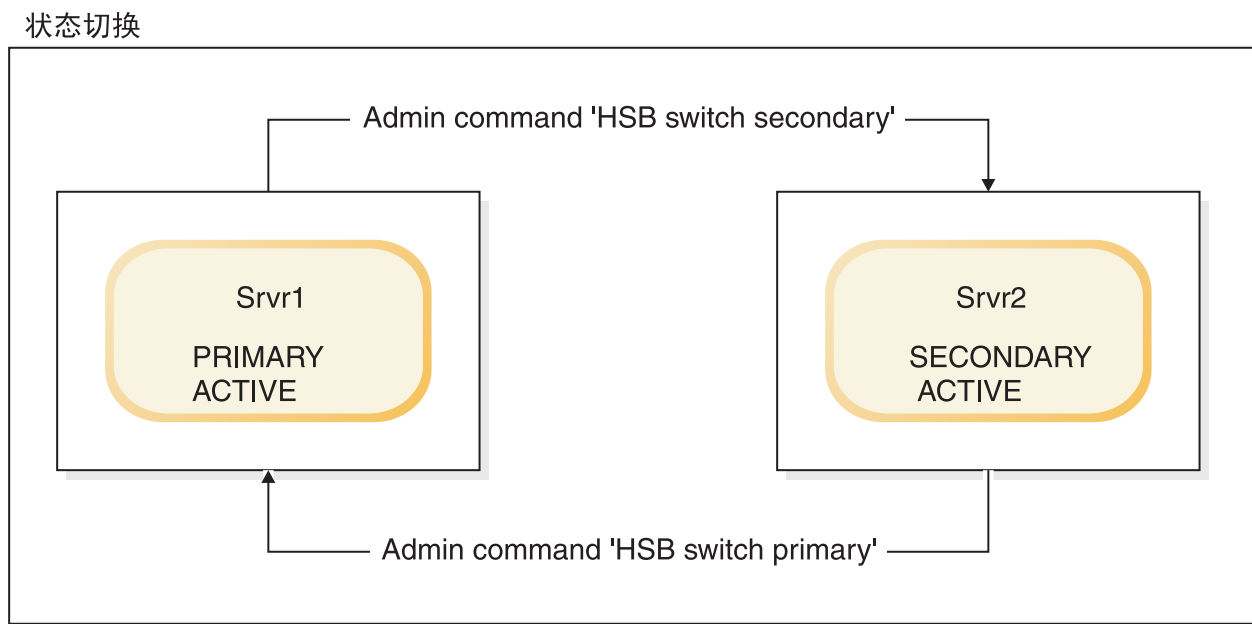


图 12. 状态切换

发出 **hotstandby switch primary** 命令以将辅助服务器 (Srvr2) 切换为主服务器时，如果辅助服务器 (Srvr2) 未连接至另一个服务器 (Srvr1)，那么会返回错误。

如果两个服务器之间建立了连接，那么它们将切换状态。换句话说，旧的主服务器 (Srvr1) 将成为新的辅助服务器，而旧的辅助服务器 (Srvr2) 将成为新的主服务器。

如果旧的辅助服务器 (Srvr2) 无法连接至另一个服务器 (Srvr1)，那么这两个服务器都将切换为 **SECONDARY ALONE** 状态。即使 **AutoPrimaryAlone** 配置参数设置为 **yes**，新的主服务器也会切换到 **SECONDARY ALONE** 状态而不是 **PRIMARY ALONE** 状态。

将辅助服务器切换为主服务器:

可以通过发出下列命令将辅助服务器切换至 **PRIMARY** 状态: **ADMIN COMMAND 'hotstandby switch primary';**。

发出 **hotstandby switch primary** 命令时，它会启动一个切换状态的进程。如果成功启动了切换进程，那么将显示以下消息: **Started the process of switching the role to primary**。可以使用 **hotstandby status switch** 命令来监视切换状态。

如果您在发出 **hotstandby switch primary** 命令之后发出落实命令，那么落实命令将失败并产生错误: replicated transaction is aborted。

所有事务在切换期间都终止。但是，管理命令 (ADMIN COMMAND) 不是用于处理事务的命令，无法回滚。

注：如果尚未打开一个新事务，那么管理命令将强制启动一个新事务。要避免留下打开的事务，或者要避免事务开始时间与预期时间不同，请在发出管理命令之后发出 **COMMIT WORK** 命令。

如果某个配置错误导致两个服务器都处于 **PRIMARY** 状态（例如，都处于 **PRIMARY ALONE** 状态），那么可以使用 **hotstandby switch secondary** 命令将其中一个服务器切换回 **SECONDARY** 状态。

- 如果这两个服务器具有相同的数据，那么在这两个服务器上可以继续执行正常操作。
- 如果这两个服务器没有相同的数据，那么主服务器将拒绝来自辅助服务器的连接操作并发出下列消息: 14525: HotStandby databases are not properly synchronized。

未启动 HotStandby 复制。在这种情况下，需要在辅助服务器中完全复制主数据库。首先必须决定哪个数据库是正确的。

此外，如果发生 14525 错误，那么数据库状态不会更改。这两个服务器仍然都处于在发出该命令之前的状态。

相关任务:

第 58 页的『显示切换状态信息』

可以使用 **ADMIN COMMAND 'hotstandby status switch'** 命令来验证两个 HotStandby 服务器之间是否发生了状态切换。

将主服务器切换为辅助服务器:

可以通过发出 **ADMIN COMMAND 'hotstandby switch secondary'** 命令将主服务器切换到 **SECONDARY** 状态。

如果两个服务器已切换状态，而您要将它们切换回其原始状态，那么 **ADMIN COMMAND 'hotstandby switch secondary'** 命令很有用。例如，当新的辅助服务器恢复运行时，您可以将其状态切换回 **PRIMARY**，并将新的主服务器切换回 **SECONDARY**。

发出 **hotstandby switch secondary** 命令时，它会启动一个切换状态的进程。如果成功启动了切换进程，那么将显示以下消息: Started the process of switching the role to secondary。可以使用 **hotstandby status switch** 命令来监视切换状态。

执行 **hotstandby switch secondary** 命令时，如果服务器之间尚未互相连接，那么旧的主服务器会尝试连接至旧的辅助服务器。

如果两个服务器之间建立了连接，那么它们将切换状态。旧的主服务器成为新的辅助服务器，而旧的辅助服务器成为新的主服务器。

相关任务:

第 58 页的『显示切换状态信息』

可以使用 **ADMIN COMMAND 'hotstandby status switch'** 命令来验证两个 HotStandby 服务器之间是否发生了状态切换。

执行故障转移

通过在辅助服务器中执行 **ADMIN COMMAND 'hotstandby set primary alone'** 命令来执行故障转移。

在处理了先前从主服务器中接收到的所有暂挂事务之后，服务器便立即进入新的状态。不会丢失任何事务，并且数据库状态反映主服务器在发生故障之前的状态。但是，如果使用的安全级别为 1-safe，那么某些事务在故障转移期间可能会丢失。

运行处于 PRIMARY ALONE 状态的新的主服务器

尽管主服务器在处于 PRIMARY ALONE 状态时与辅助服务器的连接已中断，但是您可以运行主服务器并连续更新事务日志。在辅助服务器恢复运行之后，处于 PRIMARY ALONE 状态的服务器可以继续将事务发送至辅助服务器。

过程

可以使用以下三种方法将服务器设置为 PRIMARY ALONE 状态:

- 发布以下命令:

```
ADMIN COMMAND 'hotstandby set primary alone';
```

- 使用下列命令在主服务器或辅助服务器上执行受控制的断开连接操作:

```
ADMIN COMMAND 'hotstandby disconnect';
```

如果您通过在辅助服务器上执行 **ADMIN COMMAND 'shutdown';** 命令来执行受控关闭，那么辅助服务器会在关闭之前隐式断开连接，并且主服务器将安全地切换到 PRIMARY ALONE 状态。

- 将 **HotStandby.AutoPrimaryAlone** 参数设置为 yes，以将缺省状态设置为 PRIMARY ALONE 状态。

当 **HotStandby.AutoPrimaryAlone=yes** 时，如果与辅助服务器的连接中断，那么服务器将自动进入 PRIMARY ALONE 状态。否则，在服务器发生故障之后，除非 HAC、管理员或者看守程序发出了 **ADMIN COMMAND 'hotstandby set primary alone'** 命令，否则服务器将保持 PRIMARY UNCERTAIN 状态。缺省情况下，**HotStandby.AutoPrimaryAlone** 参数设置为 no，这表示在辅助服务器发生故障时，在 PRIMARY ACTIVE 状态下运行的主服务器将自动切换到 PRIMARY UNCERTAIN 状态。

结果

在发生下列其中一种情况之前，服务器将保持 PRIMARY ALONE 状态:

- 成功地连接至辅助服务器。
- 服务器用尽了用于保存事务日志的磁盘空间。
- 达到了日志大小限制 (**MaxLogSize**)。
- 另一个命令将服务器切换到另一种状态 (例如，STANDALONE)。
- 主服务器已关闭。

要点: 请不要在关闭主服务器的同时又命令辅助服务器切换到 PRIMARY ALONE 状态。这两个操作有冲突, 反而可能会导致辅助服务器进入 SECONDARY ALONE 状态。在正常运行过程中, 这两个操作不会同时发生。

此外, 甚至在测试阶段, 请不要尝试使用关闭操作来模拟主服务器故障, 这是因为关闭操作并不能代替故障。关闭操作是一个复杂的分布式操作, 对主服务器和辅助服务器都会产生影响。此外, 在执行关闭操作之后, 作为新的辅助服务器启动的主服务器与新的主服务器之间无法进行同步复制。要关闭主服务器, 正确的顺序是:

1. 执行切换。
2. 关闭新的辅助服务器。
3. 新的主服务器自动切换到 PRIMARY ALONE 状态。

使辅助服务器恢复联机状态

过程

要使辅助服务器恢复联机状态, 请将主服务器与辅助服务器连接起来。有关更多信息, 请参阅第 57 页的 3.4.6, 『连接 HotStandby 服务器』。

在使辅助节点恢复联机状态之后, 它可能需要执行同步复制。随着时间的过去, 主服务器中已经累积了一些更改。主服务器被设置为 PRIMARY ALONE 状态, 主服务器将事务和数据写入事务日志。

在将辅助服务器再次连接至主服务器时, 会将主服务器中的暂挂更改从事务日志写入辅助服务器以保持同步。将更改写入辅助服务器之后, 辅助服务器将处于 SECONDARY ALONE 状态, 主服务器将处于 PRIMARY ALONE 状态。如果您发出 **ADMIN COMMAND 'hsb status connect'** 命令, 那么返回的消息会指示服务器是否正在执行同步复制。

注: 如果已使用 **hotstandby set standalone** 命令将主服务器设置为 STANDALONE 状态, 那么必须先将整个数据库从主服务器复制到辅助服务器, 然后才能使辅助服务器进入 SECONDARY ACTIVE 状态。有关更多信息, 请参阅第 49 页的 3.4.5, 『使主服务器与辅助服务器同步』。

结果

在辅助服务器处理了暂挂更改之后, 主服务器和辅助服务器的状态会自动更改为 PRIMARY ACTIVE 和 SECONDARY ACTIVE。

3.4.4 停止 HotStandby 操作

您可能要临时停止主服务器中的 HotStandby 操作, 例如, 如果您要使辅助服务器停止运行以对它进行升级, 而主服务器没有足够的磁盘空间来存储辅助服务器停止运行期间累积的事务日志。

关于此任务

(有关更多详细信息, 请参阅第 65 页的 3.6.3, 『用尽用于事务日志的空间』。)

要在主服务器中停止 HotStandby:

过程

1. 如果服务器之间当前已建立连接, 那么请将它们断开连接。
2. 将使用下面这一系列命令将主服务器设置为 STANDALONE 状态:

```
ADMIN COMMAND 'hotstandby disconnect'; -- 如果服务器之间已建立连接
ADMIN COMMAND 'hotstandby set standalone';
```

主服务器继续作为非 HotStandby 服务器运行。

注:

在您停止存储要发送至辅助服务器的事务日志之后，就不再能够仅仅通过再次连接主服务器和辅助服务器就在它们之间进行同步复制。相反，当您恢复 HotStandby 操作时，必须手动使两个服务器同步。

下一步做什么

如果要永久停止将该服务器用作 HotStandby 服务器，请参阅第 62 页的 3.4.10，『将 HotStandby 服务器更改为非 HotStandby 服务器』。

3.4.5 使主服务器与辅助服务器同步

要启动 HSB 复制，主服务器和辅助服务器上的数据库必须完全相同。辅助数据库必须是主数据库的精确副本。使 HotStandby 系统的数据库完全相同的过程称为 *HotStandby 同步*。

例如，在以下情况下，必须使主服务器与辅助服务器同步：

- 辅助服务器是新的，并且启动时还没有主数据库的副本。
- 辅助服务器暂时未运行，并且其数据副本不是最新的。
- 主服务器和辅助服务器同时在 Primary Alone 状态下运行，因此具有相冲突的数据。
- 辅助服务器的磁盘驱动器发生故障，或者文件已损坏而必须替换。

可以通过以下两种主要方法使两个服务器上的数据同步：*同步复制*和*完整复制*。

同步复制

当且仅当主服务器中存储了服务器之间断开连接期间辅助服务器“已丢失”的所有事务的副本时，才能使用同步复制。如果主服务器中存储了所有这些事务，那么当它重新连接至辅助服务器时，它会自动将这些事务转发至辅助服务器，以便辅助服务器可以与它进行*同步复制*。

solidDB 服务器仅在处于 PRIMARY ALONE 状态时才会存储事务（以转发至辅助服务器），在它处于 STANDALONE 状态或者正在作为非 HotStandby 服务器运行时，则不会存储事务。因此，如果主服务器自从上次与辅助服务器建立连接以来一直处于 STANDALONE 状态或者作为非 HotStandby 服务器运行，那么它就并非具有所有事务，因此无法执行同步复制。相反，必须执行完整复制。

没有显式的同步复制命令。当您使用 ADMIN COMMAND 'hotstandby connect' 命令连接服务器时，这些服务器会自动尝试执行同步复制。

在主服务器与辅助服务器建立连接之后，它们将自动进行检查以了解主服务器的事务日志中是否有要发送至辅助服务器的数据。如果存在数据，那么服务器之间将自动尝试进行同步复制。

在同步复制过程中，主服务器和辅助服务器将分别保持 PRIMARY ALONE 和 SECONDARY ALONE 状态。客户机可以继续提交查询和落实事务。同步复制过程对于客户机应用程序是透明的。

如果服务器认为即使在将事务从主服务器复制到辅助服务器之后主数据库与辅助数据库也不完全相同，那么将产生一条错误消息。

如果同步复制失败（或者您提前知道将不会进行同步复制，例如因为主服务器处于 STANDALONE 状态），那么您必须执行完全复制。

仅当辅助服务器在某个时间已经在 SECONDARY ACTIVE 状态下运行时才会进行同步复制。如果您具有全新的辅助服务器，那么即使主服务器先前是在 PRIMARY ALONE 状态下运行并且已经存储了自从它启动以来的所有事务，您也必须执行完全复制，以便为辅助服务器提供数据库的初始副本。

有关同步复制过程的更多信息，请参阅第 48 页的『使辅助服务器恢复联机状态』。

完全复制

顾名思义，完全复制就是将主服务器中的所有数据复制到辅助服务器中。通过复制数据库文件本身来执行完全复制。

完全复制用于下列情况：

- 辅助服务器是全新的，并且要获取主数据库的初始副本。
- 主服务器在不处于 PRIMARY ALONE 状态的情况下写入了事务，因此无法进行同步复制。
- 辅助数据库已损坏或者丢失。
- 辅助服务器无盘并且已发生故障。在发生故障之后启动无盘辅助服务器时，它要求使用 **hotstandby netcopy** 命令来复制整个数据库。与基于磁盘的辅助服务器不同，无盘辅助服务器无法读取事务日志并应用在它无法运行期间进行的更改。
- 主服务器具有进行同步复制所需要的所有数据，但是进行同步复制预期比复制当前数据文件要花更长时间。

注意：

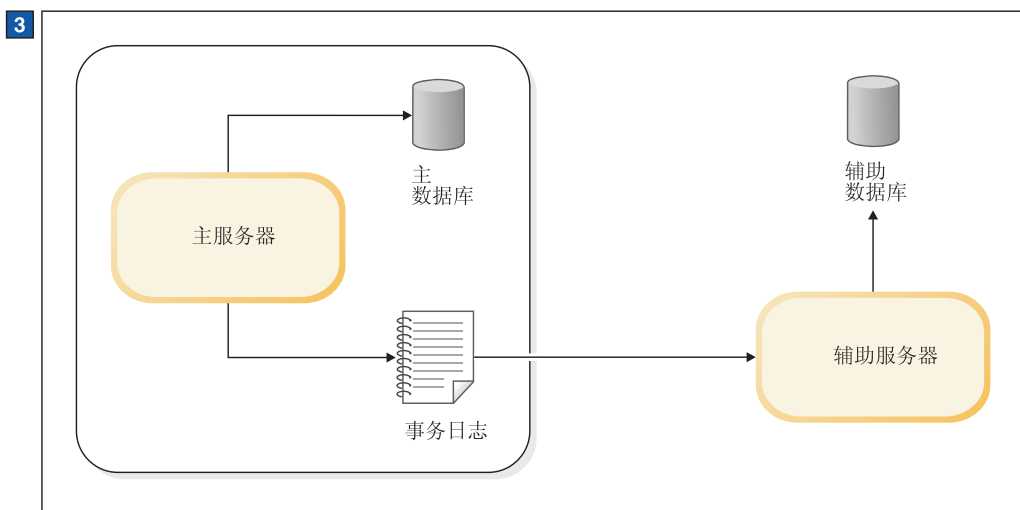
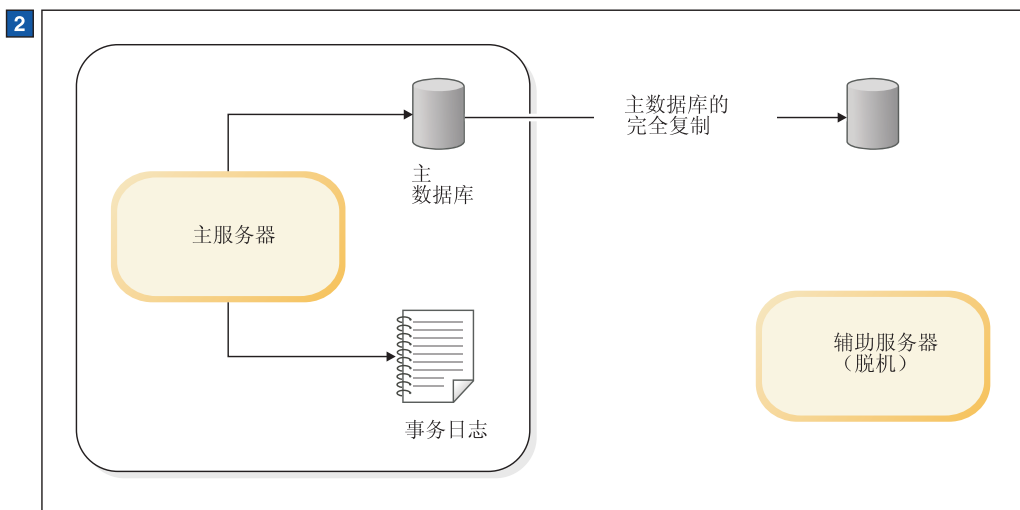
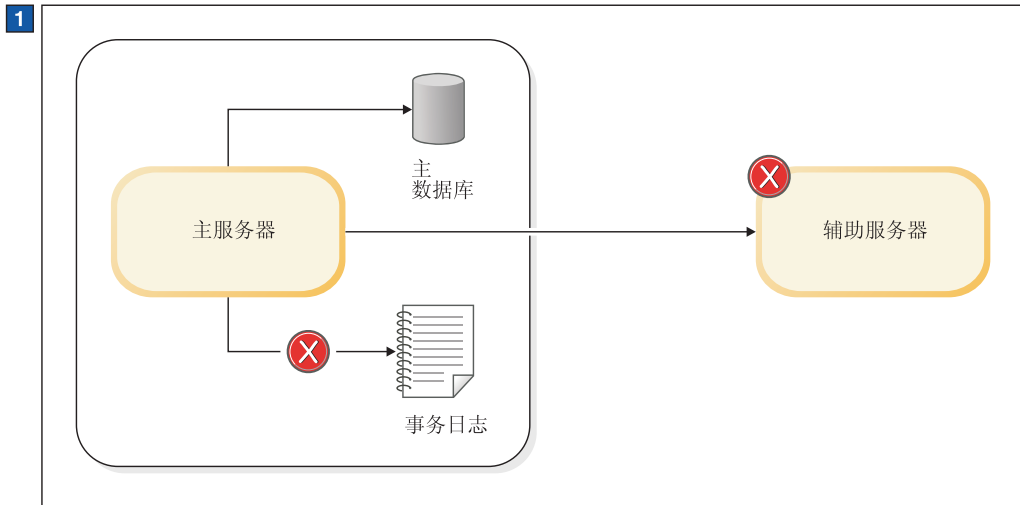
如果辅助服务器具有旧的数据库文件，那么完全复制会覆盖这些旧文件。如果辅助服务器上的文件因任何原因而包含不是来源于主服务器的数据（例如，如果两个服务器同时都在 PRIMARY ALONE 状态下运行），那么该数据会丢失。

有两个 HotStandby 命令可以用来执行完全复制，即，它们将数据库文件从主服务器复制到辅助服务器。可以使用以下任一命令：

```
ADMIN COMMAND 'hotstandby netcopy'; ADMIN COMMAND 'hotstandby copy [<directory _name>]';
```

netcopy 操作通过网络将数据库复制到正在运行并且可以通过网络接收文件的辅助服务器。**copy** 操作将数据库文件复制到对于主服务器可视的指定磁盘驱动器目录。在执行 **copy** 操作期间，辅助服务器一定不能在运行。通常，使用 **netcopy** 命令比使用 **copy** 命令更合适。因此，大多数示例都以 **netcopy** 为例，而不以 **copy** 为例。

第 56 页的『将数据库文件从主服务器复制到指定目录』和第 53 页的『通过网络将主数据库复制到辅助服务器』中描述了 **copy** 和 **netcopy** 命令。



1. 辅助服务器已停止很长时间，因此主服务器停止使用事务日志来存储用于辅助服务器的数据。日志仍然用于进行本地恢复。
2. 已将主数据库复制到辅助服务器节点，并且主服务器开始写入事务日志。
3. 复制了数据库，并且主服务器将对事务日志文件所作的更改发送至辅助服务器。

注:

上图非常简单地说明了事务日志的用法。在此图的第一部分中, 当主服务器与辅助服务器未建立连接时, 主服务器实际上会继续将数据写入事务日志, 但是仅保留足够进行恢复的数据, 这些数据不足以允许辅助服务器对断开连接以来进行的所有更改进行同步复制。

使用看守程序使服务器同步

看守程序还可以使用手动使服务器同步的命令来自动使服务器同步。

如果执行同步复制就足够了, 那么看守程序只需要监视辅助服务器以了解它何时启动, 然后发出将主服务器连接至辅助服务器的命令。如果需要执行完全复制, 那么看守程序可以指示主服务器执行 **netcopy** 操作。完全复制会覆盖辅助服务器上的所有数据。

通过网络将主数据库复制到辅助服务器

要将主服务器中的数据库文件的副本发送至辅助服务器, 请使用 **netcopy** 命令。辅助服务器必须已经在运行。

开始之前

要点: 要启动 **netcopy**, 主服务器必须处于 PRIMARY ALONE 状态。

关于此任务

在以下两种主要情况下, 应使用 **netcopy** 命令来为辅助服务器创建数据库的副本:

- 您要为以前从未有过的全新辅助服务器创建数据库。(此方法也用来将数据库复制到无盘辅助服务器, 这是因为发生故障之后, 该服务器会失去它的整个数据库, 必须将它视为全新的辅助服务器。)
- 您要替换现有的辅助数据库(例如, 已损坏的辅助数据库)

过程

要启动 **netcopy**, 请发出以下命令:

```
ADMIN COMMAND 'hotstandby netcopy';
```

当主服务器执行 **netcopy** 时, 它将使用在 `solid.ini` 的 [HotStandby] 节中指定的连接字符串。

有关用于定义连接字符串的 **Connect** 参数的详细信息, 请参阅第 37 页的 3.2.1, 『定义主节点和辅助节点 HotStandby 配置』。

当您执行 **hotstandby netcopy** 命令时, 在将主数据库的副本发送至辅助服务器之前会创建数据库检查点。

在执行 **netcopy** 期间, 主服务器继续接受事务和连接(但是, 将拒绝任何对服务器状态进行更改的 ADMIN COMMAND。)辅助服务器将不再继续接受事务和连接。启动 **netcopy** 时, 如果辅助服务器有任何已打开的连接或事务, 那么它会回滚这些已打开的事务并与其客户机断开连接, 然后开始接收 **netcopy**。辅助服务器在接收 **netcopy** 时, 仅与主服务器通信。

成功完成 **netcopy** 之后, 辅助服务器的状态会更改为 SECONDARY ALONE (如果它尚未处于该状态)。

在执行 **netcopy** 操作期间, 主服务器仍保持在 PRIMARY ALONE 状态。在完成

netcopy 之后，主服务器继续保持在相同状态。必须先连接主服务器和辅助服务器，然后才能恢复完全热备用操作。连接服务器时，会将主服务器置于 **PRIMARY ACTIVE** 状态。有关连接服务器的信息，请参阅第 57 页的 3.4.6，『连接 HotStandby 服务器』。

为辅助服务器创建新的数据库

通常，当您启动 solidDB 服务器时，如果还没有数据库，那么该服务器会提示您创建新的数据库。但是，如果该服务器是辅助服务器，那么它必须使用主数据库的副本，而不是创建它自己的数据库。因此，当您启动一个没有现成数据库的辅助服务器时，必须为它提供一个命令行参数，以告诉它等待接收来自主服务器的数据库副本。此命令行参数为 **-x backupserver**。例如，使用以下命令来启动辅助服务器：

```
solid -x backupserver
```

-x 与 **backupserver** 之间的空格是可选的。

-x backupserver 命令行参数将服务器置于 *netcopy* 侦听方式（也称为备份侦听方式）。在 *netcopy* 侦听方式下，辅助服务器唯一可以执行的操作就是接收来自主服务器的数据库副本。辅助服务器不会对任何其他命令作出响应，甚至不会接受来自客户机程序（例如，**solsql**、您的应用程序或看守程序）的连接请求。

如果存在辅助数据库，那么可以采用正常方式来启动该服务器。该服务器将在 **SECONDARY ALONE** 状态下启动。

在使用 **-x backupserver** 启动了辅助服务器之后，或者辅助服务器处于 **SECONDARY ALONE** 状态时，您可以在主服务器上执行 **netcopy** 命令。

首先确保主服务器处于 **PRIMARY ALONE** 状态。然后在主服务器上发出以下命令：

```
ADMIN COMMAND 'hsb netcopy';
```

在主服务器上，**hotstandby netcopy** 命令使用由 **solid.ini** 配置文件中的 **Connect** 参数定义的连接字符串来连接至辅助服务器。在连接服务器之后，主服务器会通过网络链路来复制数据库文件。在 *netcopy* 侦听方式下，辅助服务器仅在接收到新的数据库副本（通过在主服务器中执行 **hotstandby netcopy** 命令）之后才会尝试打开辅助数据库。

按照以下过程来创建辅助数据库副本：

1. 确保您已经配置 **solid.ini** 文件，以使它对于 HotStandby 配置有效。有关用于定义连接字符串的 **Connect** 参数的更多信息，请参阅第 37 页的 3.2.1，『定义主节点和辅助节点 HotStandby 配置』。

连接字符串用来从主服务器连接至辅助服务器以及通过网络来复制数据库文件。

2. 启动主服务器。
3. 在 *netcopy* 侦听方式下通过执行以下命令来启动辅助服务器：

```
solid -x backupserver
```

或者，启动具有现成数据库的辅助服务器。

4. 如果主服务器尚未处于 **PRIMARY ALONE** 状态，那么请将它设置为此状态：

```
ADMIN COMMAND 'hotstandby set primary alone';
```

5. 在主服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby netcopy';
```

6. 完成 **netcopy** 命令之后，可以通过发出以下命令来连接两个服务器并启动（或者恢复）完整热备用操作：

```
ADMIN COMMAND 'hotstandby connect';
```

当辅助服务器通过网络链路接收到数据库的新副本时，它将打开辅助数据库。在使用 **hsb connect** 命令连接服务器之后，辅助服务器将在正常的 SECONDARY ACTIVE 状态下运行，并且可以接受来自主服务器的用户事务。

如果使用了 HAC，那么将主服务器数据库复制到辅助服务器的过程如下所示：

1. 确保这两个服务器具有正确的 **Connect** 参数。
2. 确保主节点中的 HAC 在 `solidhac.ini` 中设置了 **PreferredPrimary=Yes** 参数值，而辅助节点中的 HAC 未设置此参数值。有关配置 HAC 的更多信息，请参阅第 42 页的 3.3，『配置 HA 控制器和 HA 管理器』。
3. 启动 HAC 实例，或者将 HAC 设置为“自动”方式（后一项操作可选）。

注： 如果已将 **netcopy** 发送至处于 SECONDARY ALONE 状态的服务器，那么将使用所复制的数据库来覆盖现有数据库。如果要使数据库重新同步或者要修复已损坏的辅助数据库，那么 **netcopy** 很有用。

替换辅助服务器上的现有数据库

尽管 **netcopy** 主要用来将数据库发送至一个以前从未具有数据库的辅助服务器，但是也可以在其他情况下使用 **netcopy**。例如，如果辅助服务器的数据库副本已损坏（例如，在磁盘驱动器发生故障之后），那么可以使用 **netcopy** 命令将主数据库的副本发送至辅助服务器。

开始之前

如果您要替换现有数据库，那么辅助服务器不必处于“**netcopy** 侦听方式”；换句话说，您不必使用 **-x backupserver** 来启动辅助服务器。

过程

1. 请确保主服务器处于 PRIMARY ALONE 状态，并且辅助服务器处于 SECONDARY ALONE 状态。
2. 对主服务器发出以下命令：

```
ADMIN COMMAND 'hotstandby netcopy';
```

在完成 **netcopy** 之后，主服务器处于 PRIMARY ALONE 状态，而辅助服务器自动进入 SECONDARY ALONE 状态（如果它尚未处于该状态）。

如果您在辅助服务器处于 SECONDARY ALONE 状态时执行 **netcopy**，并且所有客户机都已连接至辅助服务器（以执行只读查询），那么辅助服务器将回滚所有打开的事务并断开所有客户机连接。一旦完成了 **netcopy**，辅助服务器仍保持 SECONDARY ALONE 状态。

3. 服务器不会自动连接：要连接服务器，请发出以下命令：

```
ADMIN COMMAND 'hotstandby connect';
```

验证 netcopy 状态

当您启动 **netcopy** 命令时，它将在后台异步运行。当 **netcopy** 完成时，服务器不会显示消息 - 相反，您必须监视 **netcopy** 的状态。

关于此任务

即使由于发生诸如网络错误之类的问题而导致 **netcopy** 失败，服务器也不会显示消息。

过程

在主服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby status copy';
```

注：此命令使用关键字 **copy** 而不是 **netcopy**。将使用同一个命令来执行 **copy** 和 **netcopy** 操作。

此命令将显示一条状态消息，指示 **netcopy** 是已成功、仍在执行还是已失败。

将数据库文件从主服务器复制到指定目录

如果辅助服务器用于数据库的目录对于主服务器可视，那么您可以使用 **hotstandby copy** 命令将数据库从主服务器复制到辅助服务器。

开始之前

仅当主服务器和辅助服务器都可以看到一些相同的磁盘驱动器，从而都能够读/写一些相同的目录时，才能执行此任务。

注意：

在发出 **hotstandby copy** 命令之前，请关闭辅助服务器。当主服务器正在写入数据库文件时，辅助服务器不能尝试访问此文件。

注：当您发出 **hotstandby copy** 命令时，主服务器必须处于 PRIMARY ALONE 状态，并且在执行此命令期间（以及之后），主服务器仍会保持该状态。

关于此任务

hotstandby copy 命令与 **hotstandby netcopy** 命令之间的主要差别是：仅当辅助服务器正在运行时才能使用 **netcopy** 命令。仅当辅助服务器未在运行时才能使用 **copy** 命令。这两种数据库复制方法在性能方面没有重大区别。

过程

1. 要使用 **hotstandby copy** 命令来复制文件，请在主服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby copy[directory_name]';
```

其中 *directory_name* 是您要将文件复制到的目录的名称。目录名的格式取决于操作系统。目录名是可选的。如果您未指定目录名，那么服务器将使用由 **HotStandby.CopyDirectory** 参数指定的值。

当您发出 **hotstandby copy** 命令时，主服务器会先创建一个检查点并生成主数据库的副本，然后将该副本发送至辅助服务器。

由于主服务器处于 PRIMARY ALONE 状态，因此在执行 **copy** 命令期间，主服务器通常会继续进行事务处理，并且主服务器将事务存储在事务日志中，以便稍后可以将它们转发至辅助服务器。

2. 执行复制操作之后，辅助服务器仍然处于关闭状态。必须启动辅助服务器，然后发出 **hotstandby connect** 命令才能将主服务器与辅助服务器连接起来。

通过使用管理命令 **hotstandby connect** 将主数据库与辅助服务器连接起来之后，主服务器和辅助服务器将自动执行“同步复制”以使辅助服务器中的信息也是最新的。

启动辅助服务器并执行同步复制:

完成复制之后，必须使用新复制的数据库来启动辅助服务器。

过程

1. 启动辅助服务器。
2. 在主服务器中使用 **hotstandby connect** 命令将主服务器连接至辅助服务器。

```
ADMIN COMMAND 'hotstandby connect';
```

有关 **hotstandby connect** 命令的更多信息，请参阅3.4.6, 『连接 HotStandby 服务器』。

结果

在主服务器连接至辅助服务器之后，主服务器和辅助服务器将自动开始执行同步复制。在同步复制期间，主服务器通过将主服务器的事务日志复制到辅助数据库来使辅助数据库达到最新状态。然后，辅助服务器前滚事务日志并更新其数据库副本。

3.4.6 连接 HotStandby 服务器

关于此任务

将使用 **HotStandby.Connect** 参数来指定主服务器用来连接至辅助服务器的连接字符串。

可以在主节点和辅助节点中通过发出以下命令来查看当前的连接设置:

```
ADMIN COMMAND 'hotstandby cominfo';
```

过程

如果主服务器与辅助服务器之间的连接已断开或者尚未建立，那么必须在主节点或辅助节点中发出以下命令:

```
ADMIN COMMAND 'hotstandby connect';
```

例如，在执行 **netcopy** 之后，您通常将连接服务器。

由于 HotStandby 服务器中没有自动连接机制，因此，当服务器之间的连接已断开时，高可用性控制应用程序必须发出该命令。

发出此命令之后，如果主服务器与辅助服务器之间的连接成功，那么将显示一条确认消息。如果主服务器与辅助服务器之间已建立连接，但是尚未将事务日志完全复制到辅助服务器，那么主服务器会返回以下消息: Started the process of connecting the servers.

如果主服务器的状态为 PRIMARY UNCERTAIN 或 PRIMARY ALONE，那么当您发出了该命令并且连接成功时，主服务器的状态会更改为 PRIMARY ACTIVE。如果连接不成功，那么状态仍然为 PRIMARY UNCERTAIN 或 PRIMARY ALONE。

下一步做什么

有关在主服务器和辅助服务器中查询连接状态的更多信息，请参阅第 58 页的 『显示连接状态信息』。有关将应用程序重新连接至主服务器的更多信息，请参阅第 90 页的 4.3.1, 『从应用程序重新连接至主服务器』。

3.4.7 检查 HotStandby 状态

可以从主服务器和辅助服务器中请求获取有关 HotStandby 状态的信息。

过程

要检查状态，请在主服务器或辅助服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby status option';
```

其中 *option* 可以是下列其中一项：

选项	描述
catchup	指示服务器是否正在执行同步复制。同步复制是在主服务器连接至辅助服务器之后执行的。在同步复制期间，主服务器将发送累积的事务日志，以便辅助服务器可以应用更改。此选项的值可以为“ACTIVE”和“NOT ACTIVE”。
connect	指出上次尝试连接服务器是否成功。
copy	指出上次尝试执行 copy/netcopy 是否成功。
switch	指出上次尝试将服务器切换到 PRIMARY ACTIVE 或 SECONDARY ACTIVE 状态是否成功。

示例

```
ADMIN COMMAND 'hotstandby status catchup';
```

显示切换状态信息

可以使用 **ADMIN COMMAND 'hotstandby status switch'** 命令来验证两个 HotStandby 服务器之间是否发生了状态切换。

关于此任务

要检查 HotStandby 切换状态信息：

过程

在主服务器或辅助服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby status switch';
```

- 当两个服务器之间先前未发生切换时，将显示以下消息：NO SERVER SWITCH OCCURRED BEFORE。
- 当切换进程仍然处于活动状态时，将显示以下消息：ACTIVE。
- 当成功完成了最近的切换进程时，将显示以下消息：SUCCESS。
- 当最新尝试切换失败时，将显示以下消息：ERROR *number*，其中 *number* 标识切换期间所发生错误的类型。

显示连接状态信息

可以查询主服务器与辅助服务器之间的连接状态信息。此功能等价于 SQL 函数 HOTSTANDBY_CONNECTSTATUS，可以在应用程序代码中使用此函数。

过程

要检查连接状态，在主服务器或辅助服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby status connect';
```

返回值可能是：

表 7. 连接状态返回值

错误代码	文本	描述
0	已连接	连接处于活动状态。从主服务器和辅助服务器中返回。
14007	正在连接	主服务器正在连接至辅助服务器。从主服务器和辅助服务器中返回。
14008	同步复制	主服务器已连接至辅助服务器，但是尚未完全复制事务日志。从主服务器和辅助服务器中返回。
14010	正在断开连接	服务器正在断开连接。
14537	已断开连接	连接已断开。从主服务器和辅助服务器中返回。

显示连接信息

可以使用 **ADMIN COMMAND 'hotstandby cominfo'** 命令来查询 HotStandby 服务器用来连接至另一个服务器的连接信息。

过程

要显示连接信息，请在主服务器或辅助服务器中发出以下命令：

```
ADMIN COMMAND 'hotstandby cominfo';
```

该命令返回使用 **HotStandby.Connect** 参数定义的连接信息。

显示角色的开始时间

有时候，知道服务器何时进入当前角色（即，成为主服务器或辅助服务器）非常重要。

可以使用带有 **primarystarttime** 和 **secondarystarttime** 选项的 **ADMIN COMMAND 'info'** 命令来检索有关主服务器角色或辅助服务器角色的开始时间的信息。

例如：

```
admin command 'info primarystarttime';
RC TEXT
-- ----
0 2005-06-09 14:22:18
admin command 'info secondarystarttime';
RC TEXT
-- ----
0 2005-06-09 18:24:44
```

所报告的时间是服务器角色成为主服务器角色或辅助服务器角色的时间。STAND-ALONE 状态被认为是主服务器角色的一种状态。

主服务器的开始时间在发生以下状态过渡时设置:

- SECONDARY ALONE => PRIMARY ALONE
- SECONDARY ALONE => STANDALONE
- SECONDARY ACTIVE => PRIMARY ACTIVE

辅助服务器的开始时间在发生以下状态过渡时设置:

- 服务器是在 SECONDARY ALONE 状态下启动的
- OFFLINE (使用 `-x backupserver` 启动) => SECONDARY ALONE
- PRIMARY ALONE => SECONDARY ALONE
- STANDALONE => SECONDARY ALONE
- PRIMARY ACTIVE => SECONDARY ACTIVE

如果当前角色与查询发生矛盾, 那么查询将返回空字符串。例如, 如果角色是 SECONDARY, 而发出了 `info primarystarttime` 命令, 那么此命令将返回一个空字符串。

3.4.8 验证 HotStandby 服务器状态

可以使用 `ADMIN COMMAND 'hotstandby state'` 命令来检查 HotStandby 服务器的状态。

过程

要检查 HotStandby 服务器的当前状态, 请在此服务器中发出以下 HotStandby 命令:

```
ADMIN COMMAND 'hotstandby state';
```

结果

要了解该命令返回的可能状态的描述, 请参阅第 8 页的『对服务器状态的描述』。

注:

1. 如果服务器已配置为 HotStandby 服务器, 那么启动该服务器时, 该服务器会在 SECONDARY ALONE 状态下启动。
2. 服务器无法返回状态 OFFLINE, 这是因为当服务器处于 OFFLINE 状态时, 您无法与其建立连接并发出任何查询 (例如, `ADMIN COMMAND 'hotstandby state'`)。
3. 如果在未配置为 HotStandby 的服务器上发出了 `ADMIN COMMAND 'hotstandby state'`, 那么将返回以下错误消息:

```
14527: 这是一个非 HotStandby 服务器
```

如果使用了 HAC 和 HA 管理器, 那么 HA 管理器会显示两个服务器的 HSB 状态。每秒钟将定期查询信息。

下一步做什么

有关在执行管理操作和故障诊断操作时发生的 HotStandby 状态过渡的摘要, 请参阅第 155 页的附录 D, 『服务器状态过渡』。

服务器状态组合情况

并不是所有服务器状态组合情况都可以实现。例如，如果主服务器处于 PRIMARY ACTIVE 状态，那么辅助服务器只能处于 SECONDARY ACTIVE 状态。

下表列出了当一个 HotStandby 服务器的相关联服务器处于特定状态时，此服务器可以具有的状态。

表 8. 服务器状态

服务器的状态	相关联的服务器可能处于的状态
PRIMARY ACTIVE	SECONDARY ACTIVE
PRIMARY ALONE	PRIMARY ALONE * PRIMARY UNCERTAIN SECONDARY ALONE STANDALONE *
PRIMARY UNCERTAIN	PRIMARY ALONE PRIMARY UNCERTAIN SECONDARY ALONE STANDALONE
SECONDARY ACTIVE	PRIMARY ACTIVE
SECONDARY ALONE	PRIMARY ALONE PRIMARY UNCERTAIN SECONDARY ALONE STANDALONE
STANDALONE	PRIMARY ALONE * PRIMARY UNCERTAIN SECONDARY ALONE STANDALONE *

* 如果一个服务器处于 PRIMARY ALONE 状态或 STANDALONE 状态，那么另一个服务器不能处于 PRIMARY ALONE 或 STANDALONE 状态。这是因为如果单独对两个服务器都进行了更改，那么将无法将两个数据库合并成一个数据库。

3.4.9 选择使哪个服务器成为主服务器

在某些错误恢复情况下，您可能不知道必须使哪个服务器成为主服务器。可以使用 **hsb logpos** 命令来确定要使哪个服务器成为主服务器。

关于此任务

例如，在两个数据库都发生故障的情况下，可以使用此过程；在服务器之间互相失去联系之前是主服务器的服务器现在并不一定应该成为主服务器。

过程

1. 确保两个服务器都处于 SECONDARY ALONE 状态。
2. 连接至两个服务器。
3. 在每个服务器中发出以下命令：

```
ADMIN COMMAND 'hsb logpos';
```

此命令会返回上次操作时的日志操作标识和服务器的角色 (PRIMARY、SECONDARY 和 STANDALONE)。管理命令在成功后会返回错误代码 0、一个字符串和服务器先前的角色。

例如:

```
ADMIN COMMAND 'hsb logpos';
RC TEXT
-- ----
0 000000000000000000000871:PRIMARY
```

应用程序必须将该字符串视为一个不透明的值,因为它未定义结构。

4. 对字符串值进行比较。

例如,在 C 语言中,使用 strcmp() 函数。

从原则上来说,具有较大日志操作标识值的服务器已接受更多事务,因此该服务器应该成为主服务器。但是,如果您在 HSB 连接失败之后已对两个服务器进行了更新,那么比较日志操作标识值将不再可靠。

如果使用了 HAC 且 STRINGS 值相等,那么 LocalDB.PreferredPrimary HAC 参数将定义本地服务器是否将成为主服务器。

5. 通过在将成为主服务器的服务器上使用以下命令来选择主服务器:

```
ADMIN COMMAND 'hsb set primary alone';
```

6. 通过使用以下命令将 HotStandby 服务器互相连接起来:

```
ADMIN COMMAND 'hsb connect';
```

结果

- 如果 **hsb connect** 命令成功,那么辅助服务器将与主服务器之间进行同步复制,并且 HotStandby 对将再次起作用。
- 如果 **hsb connect** 命令失败,那么必须分别同步节点。有关更多信息,请参阅第 49 页的 3.4.5,『使主服务器与辅助服务器同步』。

要点: 此过程并不确保具有更大字符串值的服务器是另一个服务器的超集。这两个服务器可能各自都接受了另一个服务器未接受的事务。例如,这两个服务器可能一直在 PRIMARY ALONE 状态下运行。为了检测这两个服务器都不是对方的超集这种可能性,在发出 **hsb connect** 命令时,服务器会对信息进行比较。如果这两个服务器都不是超集,那么 **hsb connect** 命令将失败并显示适当的错误消息。

3.4.10 将 HotStandby 服务器更改为非 HotStandby 服务器

可以通过编辑 solid.ini 文件的 [HotStandby] 节将主服务器或辅助服务器更改为一个正常的非 HotStandby 服务器。

过程

1. 除去 **HSBEnabled** 参数 (或将它设置为 no)。
2. (可选) 除去或者剔除 **Connect** 参数。
3. 要使 solid.ini 文件中的更改生效,请重新启动服务器。

示例

如果您要让服务器临时停止充当 HotStandby 服务器，但是希望它稍后继续充当 HotStandby 服务器，那么可以让 `solid.ini` 文件保持不变，而将服务器的状态更改为 `STANDALONE`。

3.5 性能调整

3.5.1 使用安全级别和耐久性级别来调整复制性能

在执行正常操作期间的数据复制性能取决于耐久性级别和安全级别的设置。此外，当使用 `2-safe` 复制时，`2-safe` 方式中使用的确认策略会影响等待时间，应用程序将感觉到这一点。有关更多信息，请参阅第 17 页的 1.2，『性能和 HotStandby』。

3.5.2 调整 netcopy 性能

`netcopy` 命令允许将主数据库复制到远程辅助服务器。当两个服务器的其中一个或全部是无盘服务器时，还可以使用 `netcopy` 命令将数据库从主服务器复制到辅助服务器。将通过网络链路来复制主数据库文件。

将使用 `HotStandby.Connect` 参数来指定用来连接至辅助服务器以执行 `netcopy` 操作的连接字符串。

控制 netcopy 块大小

将数据库文件从主服务器复制到辅助服务器时，通过增大或减小该文件的块大小，可以调整 `netcopy` 的性能（以及备份的性能）。可以使用 `General.BackupBlockSize` 参数来定义块大小。一般来说，块大小越大，`netcopy` 和备份的速度就更快，但代价是，在执行 `netcopy` 或备份时，服务器对其他请求作出响应的的时间可能会变长。

缺省情况下，`General.BackupBlockSize` 参数设置为 64K。

可以按字节来设置该值，也可以使用后缀“M”和“K”来设置该值。例如：

```
[General]
BackupBlockSize = 32K
```

或者

```
[General]
BackupBlockSize = 32768
```

注：

- `General.BackupBlockSize` 的最小值是服务器块大小（使用 `IndexFile.BlockSize` 参数进行定义）。
- 最大值是 8MB。如果参数值超过最大值，那么将使用缺省值 (64K)。
- `General.BackupBlockSize` 的值必须是服务器的数据库块大小（使用 `IndexFile.BlockSize` 参数进行定义）的倍数。

相关任务:

第 53 页的『通过网络将主数据库复制到辅助服务器』

要将主服务器中的数据库文件的副本发送至辅助服务器，请使用 `netcopy` 命令。辅助服务器必须已经在运行。

3.5.3 调整数据库同步复制性能

当发生故障的辅助服务器恢复运行并且已连接至主服务器时，HotStandby 通过一个称为 *HotStandby 数据库同步复制* 的自动化进程继续将主服务器事务日志文件内容发送至辅助节点。

通过调整服务器花在同步复制上的时间（相对于花在维护当前客户机数据库查询上的时间），可以调整数据库同步复制的性能。同步复制时间由

HotStandby.CatchupSpeedRate 参数控制，该参数以百分数的形式定义服务器花在同步复制上的时间。例如:

```
[HotStandby]
CatchupSpeedRate = 90
```

如果将 **HotStandby.CatchupSpeedRate** 的值指定为 90，那么服务器花大约 90% 的时间来进行同步复制，花大约 10% 的时间来对用户查询进行响应。

HotStandby.CatchupSpeedRate 值越大，同步复制就越快。但是，较高的同步复制速率也会影响其他活动（例如，用户查询）。缺省情况下，**HotStandby.CatchupSpeedRate** 设置为 50。

3.6 适用于将 solidDB 与 HotStandby 配合使用的特殊注意事项

将 solidDB 与 HotStandby 配合使用时，必须特别注意事务隔离级别、事务日志空间、未计划的网络分区以及调速。

3.6.1 事务隔离级别和内存表

如果您已连接至辅助服务器并且要从内存表中读取数据，那么会自动将事务隔离级别设置为“落实读”。即使您已将事务级别指定为“可重复读”或“可序列化”，事务级别也是“落实读”。

此外，如果使用了负载均衡，那么隔离级别在缺省情况下为“落实读”。

3.6.2 网络分区和双重主服务器

在某些情况下，两个服务器都可能处于 PRIMARY ALONE 状态。如果存在双重主服务器，可能会导致严重的无法恢复的错误。

在出现双重主服务器的情况下，如果每个服务器都落实了另一个服务器未落实的任何事务，那么您无法使这两个服务器重新同步，这是因为无法合并数据库以创建具有正确信息的单个数据库。实际上，在存在双重主服务器的情况下，在“错误主服务器”数据库中落实的事务将丢失。存在双重主服务器还可能会导致其他错误。

双重主服务器问题很可能由网络分区引起。在网络分区情况下，某些（但不是全部）网络连接已断开，并且您的单个网络实际上已分成多个单独的子网。每个子网都可以

在其内部进行通信，但不能与其他子网进行通信。因此，两个服务器将相互断开连接，但是仍然已启动并正在运行，并且在某些情况下可能仍然可以与某些客户机进行通信。

可以通过在位于 HSB 系统外部的节点中使用单个实例看守程序来避免出现双重主服务器情况。通过使用这样的看守程序，易于决定必须将哪个服务器设置为主服务器并确保客户机仅看到一个主服务器。

尽管 HAC 由两个与 HSB 服务器在相同节点中运行的实例组成，但是，将 HAC 与 ERE 配合使用时，不可能出现双重主服务器的情况。

即使您遇到了双重主服务器的情况，也不会具有不一致的数据，除非有人能够在原始辅助服务器上执行写操作（在此服务器已切换到 PRIMARY ALONE 状态之后）。如果已将原始辅助服务器与网络的其余部分断开连接，那么无人能够对其写入。原始主服务器是辅助服务器的超集，并且（在重新连接服务器并且允许辅助服务器对原始主服务器上所作的更改进行同步复制之后）您可以获得单个一致数据集。

尽管很少出现双重主服务器的情况，但这种情况真的出现时则很危险。必须规划您的环境，以便可以防止数据出现不一致的情况。例如，使用 ERE。

如果您为两个服务器的其中一个或全部将配置参数 **AutoPrimaryAlone** 设置为 Yes，那么出现双重主服务器机会将增大。将 **AutoPrimaryAlone** 参数设置为 Yes 时，意味着系统可以快速对故障作出响应，但是它也意味着系统不再有任何独立的观察器或观察员（HAC、看守程序或人员）来防止出现双重主服务器的情况。如果您对网络可靠性有任何顾虑，那么请让 **AutoPrimaryAlone** 参数保持使用出厂值（即，No）。

3.6.3 用尽用于事务日志的空间

使用 HotStandby 时，如果您使服务器处于 PRIMARY ALONE 状态，那么一定要小心不要让它用尽用于事务日志的磁盘空间。在非 HotStandby 服务器中，如果您频繁地执行检查点操作，那么事务日志不会增大，这是因为每次执行检查点操作之后，服务器都会删除旧的事务日志。

尤其是，非 HotStandby 服务器将删除在执行检查点操作之前生成的日志和数据更改。有关执行检查点操作的更多信息，请参阅《solidDB 管理员指南》。

但是，在一个处于 PRIMARY ALONE 状态下运行的 HotStandby 服务器中，服务器必须保留自从主服务器与辅助服务器失去联系以来已累积的事务日志。如果辅助服务器已关闭很长时间，那么此服务器可能会保留大量事务日志数据；然而，如果服务器不处于这种状态，那么在每次执行检查点操作之后都会删除这些事务日志数据。在最糟糕的情况下，如果辅助服务器无法及时恢复正常，并且又没有足够的磁盘空间来存储发生的所有事务，那么主服务器的事务日志会填满所有可用磁盘空间。当磁盘变满时，服务器会切换到只读方式。

要防止事务日志用完磁盘空间，可以使用参数 **MaxLogSize** 来控制日志大小。在日志达到指定的日志总大小之后，服务器会在下一个检查点自动切换到 STANDALONE 状态。在无盘服务器中，服务器继续保持 PRIMARY ALONE 状态，即使根本没有执行磁盘写入操作。

如果将服务器设置为 STANDALONE 状态，那么它不再保留自从主服务器与辅助服务器失去联系以来已累积的所有事务日志。在没有完整事务日志的情况下，仅仅通过将

主服务器连接至辅助服务器并允许辅助服务器通过读取旧日志来进行“同步复制”将无法使系统同步。必须使用 **copy** 或 **netcopy** 命令将整个数据库从主服务器复制到辅助服务器。

如果使用了 HAC，那么它将识别上述情况，并通过自动执行必需的操作使服务器处于活动状态。

3.6.4 在辅助服务器中进行调速和多处理

为了保持主服务器与辅助服务器中的数据最新，主服务器会写入它的事务日志并将其转发至辅助服务器，以便辅助服务器可以对它的数据库副本进行相同更改。如果辅助服务器的处理步调无法与主服务器的处理步调保持一致，那么 **solidDB** 调速机制会降低主服务器上的处理速度。从应用程序的立场来说，调速会导致响应时间增加。

从 V7.0 开始，辅助服务器可以使用多个线程来处理写入负载。辅助服务器在一个流中接收来自主服务器的操作。会对操作进行解析并将每个事务附加到执行队列中；执行队列的数量与已打开的并行事务的数量相同。为了确保以正确的顺序来执行有冲突的操作，已使用了行锁定。

您可以利用下列性能计数器 (**pmon**) 来监视线程的使用情形：

表 9. 用于监视辅助服务器中多处理的 **pmon** 计数器

perfmon 变量	描述
HSB grpcommits	最新组落实中的事务数 将在作为一个包发送至辅助服务器的单日志脉冲串中对事务落实进行分组。 只能在主服务器上使用此计数器。
HSB secondary ops in packet	辅助服务器从主服务器发出的最新日志记录包中接收到的日志记录数。
HSB secondary trx count	辅助服务器从主服务器接收到的已打开的事务数
HSB secondary locks	辅助服务器上的行级别锁定数
HSB secondary lock reqs	辅助服务器上的锁定请求数
HSB secondary lock waits	自服务器启动以来，辅助服务器上的锁定等待数
HSB secondary op waits	辅助服务器上的操作（事务）已等待继续执行的次数
HSB secondary buffers	辅助服务器已从主服务器接收到的缓冲区日志记录包数
HSB secondary serial mode count	辅助并行执行程序已切换至序列方式（而不是运行于并行方式）的次数
HSB secondary dispatch queuelength	辅助服务器上最新分派线程（要分派的操作）的大小

也可以使用 **HotStandby.SecondaryThreads** 参数来优化辅助服务器的多处理级别。最优的线程数视环境而定，并需要利用您环境的典型装入进行实验。从原则上来说，计算机的处理能力（核心）越强，您可以设置的 **HotStandby.SecondaryThreads** 参数值越高。但是，过度使用线程并不一定会提高性能。

3.7 适用于更低成本和更高安全性的配置

HotStandby 解决方案使用成对的主服务器和辅助服务器来提供真正的高可用性。但是，使用成对的服务器可能不是每个使用方案的最佳方案。如果不需要几乎即时进行故障转移，那么可能就没有必要为每个主服务器配备一个辅助服务器。另一种极端情况是，某些商业案例可能对可靠性要求特别高，也需要保证资源的安全，从而购买备用服务器的备用服务器。“备用服务器的备用服务器”表示，不仅要为每个主服务器购买辅助服务器，而且还要购买一个或多个备用服务器，以便在主服务器发生故障而由其辅助服务器替换时，如果无法快速修复原始主服务器，那么可以将一个备用服务器用作新的辅助服务器。

为了帮助您降低成本或者提高可靠性，HotStandby 支持各种替代模型，以替代所谓的 N+N 或者 2N 标准热备用模型（在标准模型中，主服务器和辅助服务器的数量相同（都为 N））。替代模型包括：

- “N + 1 个备用服务器”或“N + M 个备用服务器”：这是用于独立服务器的“备用节点”方案。有 N 个“主”服务器以及一个或多个备用服务器，但是没有辅助服务器。发生故障的主服务器由备用服务器替换。独立服务器案例不是真正的热备用方案。可以将它称为暖备用：虽然提供了备用计算机，但是它没有数据库的副本。
- 2N + 1 个备用服务器或者 2N + M 个备用服务器：这是用于 HotStandby 的“备用节点”方案。有 N 个 HotStandby 对，即，每个主服务器都有一个辅助服务器。此外，有 M 个备用服务器，其中 M 至少应为 1，但是通常小于 N。当一个主服务器或辅助服务器失败时，就会将一个备用服务器作为新的辅助服务器。因此，主服务器决不会长时间独立运行，即使它原来的伙伴服务器失败也是如此。

3.7.1 降低成本：N + 1 个备用服务器和 N + M 个备用服务器方案

在“N + 1 个备用服务器”和“N + M 个备用服务器”方案中，有 N 个“主”服务器，每个主服务器都在 STANDALONE 状态下运行，即，不必将它们连接至辅助服务器。此外，有 M 个备用服务器，其中 M 至少应为 1，但是通常小于 N。如果“主”服务器失败，其中一个备用服务器就会替换它。将数据从原始服务器复制到备用服务器，使原始服务器将脱机，并将备用服务器配置为充当原始服务器。任何备用服务器都可以替换任何主服务器（备用服务器并不专用于特定的主服务器）。而且，不会即时进行故障转移。

N + 1 方法称为单一备用服务器方案，N + M 方法称为多备用服务器方案。

这两个方法都要求您具有原始服务器数据的副本。如果原始服务器的磁盘驱动器已损坏并且不存在数据的备份，那么单一备用服务器方案和多备用服务器方案不能防止丢失数据。

N+M 方法在以下情况下很有用：

1. 您要使用备用节点来处理预定维护，而不是因意外故障而进行维护。
2. 您具有可靠的备份，可以将它快速复制到备用服务器中。
 - a. 您已将备份保存在磁带或其他某些安全位置。
 - b. 您要使用 solidDB 高级复制技术，并且您可以通过从发生故障的高级复制主控服务器或副本服务器中读取数据来复制或重新创建足够多的数据。
3. 单个数据块不是非常关键或者不是唯一的。

- a. 例如，如果您需要的是“计算马力”（负载分布能力）而不是特定数据，那么通过复制标准数据库或“种子”数据库，或者通过从客户机中获取数据，也许能够满足您的需要，然后继续运行。
- b. 同样，如果所有服务器都具有差不多相同的数据，并且几乎完全在响应“读”请求，很少响应或者不响应写请求，那么您可以从任何一台计算机中复制有用的数据库。例如，如果您正在运行许多服务器（它们都使用相同的因特网路由表或电话簿信息），那么您可以使用任何服务器来恢复数据。

3.7.2 提高可靠性: 2N + 1 备用服务器和 2N + M 备用服务器方案

正常的 HotStandby 操作非常可靠。如果主服务器和辅助服务器都使用单独的可靠电源，那么它们几乎同时发生故障的可能性很低。但是，如果您甚至要降低电源故障的风险，或者无法快速修复发生故障的服务器，那该怎么办呢？理想情况下，当主服务器发生故障并且您使用辅助服务器替换它时（或者当辅助服务器发生故障时），您希望使用一个新的辅助服务器来替换旧的辅助服务器，以便您可以使用一个完整的服务器对来继续运行。

这种情况称为 $2N + 1$ 个备用服务器（或者 $2N + M$ 个备用服务器）方案。您具有 N 个主服务器， N 个辅助服务器，并且至少有一个备用服务器可以替换任何已发生故障或者已转换为主服务器的辅助服务器。备用服务器并不专用于特定服务器（或者 HSB 服务器对），并且备用服务器需要具备某些配置之后才能替换失败的服务器。

3.7.3 solidDB HSB 如何支持 N+1 (N+M) 和 2N+1 (2N+M) 方法

必须使备用服务器看起来与它要替换的服务器一样。通常，这意味着：

1. 您必须将数据复制到备用服务器中。
2. 您必须让备用服务器与它要替换的服务器在同一网络地址或者在客户机程序知道用来通信的另一个地址进行“侦听”。
3. 此外，在 $2N+1$ ($2N+M$) 方案中，您还必须让新的辅助服务器和当前的主服务器知道如何互相进行通信。换句话说，您必须让每个服务器知道用来连接至对方的地址。

solidDB 有以下两项功能用来满足这样的需要：

- 可以在不关闭备用服务器的情况下将数据复制到备用服务器。
 - 可以动态地设置某些配置参数。
1. 尽管通常通过关闭服务器，更新 `solid.ini` 配置文件，然后重新启动服务器这种方法来设置 solidDB 配置参数，但是也可以通过执行类似于以下示例的 `ADMIN COMMAND` 来更改某些配置参数（例如，“`com.listen`”和“`hotstandby.connect`”参数）：

```
ADMIN COMMAND 'parameter com.listen="tcp SpareServer1 1315"';  
ADMIN COMMAND 'hsb parameter connect "tcp srvr27 1316"';
```

这意味着可以动态地配置备用服务器以替换另一个服务器，而不必先关闭备用服务器。同样，主服务器可以知道其新的辅助服务器的连接字符串。

提示： 执行这些命令时，不会将已更新的参数值写入 `solid.ini` 文件。要确保服务器在下一次重新启动之后具有新值，请在 `solid.ini` 文件中更新这些参数值。

要点：

必须使用 **-x backupserver** 命令行选项来启动备用服务器，以使它准备好接收来自主服务器的 netcopy。有关 **-x backupserver** 选项的更多信息，请参阅第 54 页的『为辅助服务器创建新的数据库』。

2. **netcopy** 命令将数据库复制到已启动并且正在运行的服务器。

a. 设置“connect”参数的新值:

```
ADMIN COMMAND 'hsb parameter connect "tcp srvr27 1316";
```

b. 执行 **netcopy** 命令:

```
ADMIN COMMAND 'hsb netcopy';
```

c. 通过执行以下命令将当前主服务器与新的辅助服务器连接起来:

```
ADMIN COMMAND 'hsb connect';
```

3.7.4 将 HAC 与备用服务器配合使用

HAC 对于备用服务器方案提供了有限支持。可以在备用服务器中使用 HAC，但是必须先更新主服务器 HAC 的连接信息 **RemoteDB.Connect**，然后主节点中的 HAC 才能开始监视备用服务器的 HSB 状态。必须更新 `solidhac.ini` 文件并重新启动主节点中的 HAC。

同样，备用节点中的 HAC 需要主服务器的连接信息。如果事先不知道连接信息，那么必须在 `solidhac.ini` 文件中添加此信息。将此信息添加到 `solidhac.ini` 文件之后，必须重新启动 HAC。

4 对应用程序使用 HotStandby

要使用 HotStandby，应用程序设计必须解决应用程序如何连接至 HotStandby 服务器以及应用程序如何处理故障和故障转移。

根据应用程序所在的位置（与其使用的数据库相比较而言）以及故障转移情况下的首选自动化级别，应用程序连接至 HotStandby 服务器对的方式可能会有所不同。

通常，当应用程序使用数据库时，该应用程序会连接一个数据库，并将该数据库用于所有查询、读取和写入操作。如果该数据库变得不可用，那么连接会中断，并且应用程序必须等待，直到它可以再次连接至同一数据库为止。使用 HotStandby 对时，会有两个数据库，它们都是实时数据库且处于运行中，彼此相互监视且所具有的内容相同。应用程序可以使用更多的可用选项并具有更多的责任来确保应用程序向其用户提供提供的服务在即使其中一个数据库节点发生故障时也能够继续提供。

4.1 连接 HotStandby 服务器

可以对 HotStandby 环境中的应用程序使用两种连接方法，即基本连接和透明连接 (TC)。通过基本连接，客户机以显式方式连接每个 HSB 服务器。使用透明连接时，客户机仅建立一个称为“TC 连接”的逻辑连接。

这两种连接类型在 solidDB ODBC 和 JDBC 驱动程序以及 SMA 服务器中都受支持。连接类型定义在连接字符串内。通过基本连接，会使用标准 solidDB 连接语法。通过透明连接，会使用特定于 TC 连接的语法。

基本连接

使用“基本连接”时，应用程序必须考虑通过使用特定服务器地址单独连接至 HotStandby 配置中的每个服务器。如果发生了故障转移，那么活动连接将断开，应用程序必须重新连接至新的主服务器。

透明连接

使用“透明连接”时，应用程序不必处理连接至任何特定服务器，或者在发生了故障转移的情况下重新建立连接。应用程序将保持一个称为 TC 连接的逻辑连接（句柄）。简单地说，透明连接使得应用程序不必考虑服务器及其地址的多重性。

只要所指定的服务器集中有任何服务器处于 PRIMARY ACTIVE、PRIMARY ALONE 或 STANDALONE 状态，那么在进行故障转移和切换之后仍会保持连接句柄。在进行故障转移和切换时，驱动程序将执行一项称为连接切换的内部操作。会将连接切换告知应用程序，因为应用程序必须根据故障透明性级别来重建某些会话状态。

使用基于网络的连接的透明连接

通过基于网络的连接，应用程序和服务器可以位于相同或不同的节点上。

应用程序可以使用负载均衡功能向辅助服务器发送只读负载。

使用 SMA 连接的透明连接

使用 SMA 时，有两个应用程序，每个 HotStandby 节点上有一个应用程序。主

节点上的应用程序将 SMA 连接用于读取和写入操作。辅助节点上的应用程序使用 SMA 连接在本地执行读取操作；来自辅助节点上应用程序的写入事务在主服务器上利用网络连接来执行。

从原则上来说，服务器故障转移和切换的处理方式与基于网络的处理方式相同。但是，如果应用程序因 SMA 服务器失败而失败，那么将需要特定于应用程序的高可用性处理。

4.1.1 选择连接类型 过程

以下兼容性矩阵可以帮助您通过对所选择的连接信息指示受支持的功能来选择连接类型：

表 10. 选择连接类型

功能	独立配置	HSB 配置
基本连接	Yes (BC Info)	Yes (BC Info)
透明故障转移	No	Yes (TC Info)
负载均衡	No	Yes (TC Info)

4.2 透明连接

当使用 solidDB 透明连接时，solidDB ODBC 或 JDBC 驱动程序会对应用程序隐藏两个 HSB 服务器的存在。透明连接还提供了主服务器与辅助服务器之间的透明负载均衡。

驱动程序提供了已映射至内部活动连接的单个逻辑 TC 连接。在理想情况下，当主服务器和辅助服务器都在活动状态下运行时，驱动程序还会维护一个备用连接，即，与辅助服务器的连接。此连接将设置为事件等待方式，它已准备好接收 HSB 状态更改事件。这些事件是有关驱动程序将使用的故障转移和切换的主要信息源。在某些情况下（例如，PRIMARY ALONE 操作），将缺少备用连接，但是每当有可能，驱动程序就会尝试执行它。

备用连接的处理对于应用程序完全透明。此外，将通过特殊的错误代码向应用程序通知所产生的任何连接切换（即，更改 TC 连接至内部活动连接的映射）。

要点： solidDB 工具，例如 solidDB SQL 编辑器 (`solsql`)，不支持 TC 连接。

4.2.1 定义“透明连接”连接

将使用非标准 ODBC 连接字符串设置或 JDBC 连接属性来指定“透明连接”。

透明连接信息的语法 - ODBC

当使用 solidDB 透明连接时，客户机将仅建立一个称为“TC 连接”的逻辑连接。使用 ODBC 时，TC 连接在 TC 信息中进行指定。TC 信息在两个 HSB 配置中都将执行透明故障转移和负载均衡。

使用 ODBC 应用程序时，可以采用下列方法来指定 TC 连接：

- SQLConnect 函数：

```
rc = SQLConnect(comHandle, "<solidDB_TC_Info>", ...
```

- 客户端 solid.ini 文件:

```
[Com]
Connect = <solidDB_TC_Info>
```

solidDB TC 信息的语法是:

```
<solidDB_TC_Info> ::= {[<failure_transparency_level_attribute>]
[<preferred_access_attribute>] [<encryption_attribute>]
<connect_target_list>} | <cluster_info>

failure_transparency_level_attribute ::= TF_LEVEL={NONE |
CONNECTION | SESSION}

preferred_access_attribute ::= PREFERRED_ACCESS={WRITE_MOSTLY |
READ_MOSTLY | LOCAL_READ}

encryption_attribute ::= USE_ENCRYPTION={YES|NO} [USE_GSKIT={YES|NO}]

connect_target_list ::= [SERVERS=]<connect_string>[, <connect_string > ...]

cluster_info ::= CLUSTER= <connect_string>[, <connect_string>...]
```

可以使用下列缩写。

表 11. TC Info 缩写

缩写	相应的语法
TF	TF_LEVEL
CO	CONNECTION
SES	SESSION
PA	PREFERRED_ACCESS
RM	READ_MOSTLY
WM	WRITE_MOSTLY
LR	LOCAL_READ
S	SERVERS

故障透明性属性

故障透明性用于处理对故障的掩盖。故障透明性级别是使用 TC Info 的 TF_LEVEL 属性设置的。有三个故障透明性级别:

1. NONE - 已禁用故障透明性。这是缺省值。
2. CONNECTION - 将保持服务器连接，也就是说，在进行故障转移或切换时不需要重新连接。
3. SESSION - 将保持某些具有非缺省值的会话属性。此外，还将保留预编译语句。但是，打开的游标会关闭，并且正在进行的事务会异常终止。

首选访问属性 - 负载均衡

首选访问属性 (PREFERRED_ACCESS) 指示是否应用了只读负载的负载均衡。可使用下列级别:

- WRITE_MOSTLY - 无负载均衡（缺省值）。所有事务都在主服务器上执行。
- READ_MOSTLY - 通过在主服务器与辅助服务器之间分发只读事务来执行负载均衡，如 **Cluster.ReadMostlyLoadPercentAtPrimary** 所定义

- LOCAL_READ - 通过在可能时于本地执行只读事务来执行负载均衡。始终会将只读事务转移到作为主服务器或辅助服务器的本地服务器。如果找不到本地服务器，那么将使用主服务器的已指定的工作负载连接（定义于连接字符串中的第一个基于网络的连接）。写入事务始终在主服务器中执行。

LOCAL_READ 通常与 SMA 设置配合使用，在 SMA 设置中，每个节点上至少有一个 SMA 应用程序。

要点： 将 SMA 与透明连接（TC）配合使用时，如果将负载均衡方法设置为 READ_MOSTLY 或 WRITE_MOSTLY（缺省值），那么将使用网络连接而不是 SMA 连接。因此，将 SMA 与 TC 配合使用时，请始终将负载均衡方法设置为 LOCAL_READ。

连接目标列表和集群信息 - 服务器地址

solidDB TC 信息包括一个服务器地址列表（称为连接目标列表）。如果 HotStandby 配置包括 SMA，那么必须使用特定于 SMA 的连接目标列表。

基于网络的应用程序的连接目标列表语法为：

```
[SERVERS:] <network_connect_string>, <network_connect_string>
```

SMA 应用程序的连接目标列表语法为：

```
[SERVERS:] <sma_connect_string>, <network_connect_string>
```

network_connect_string

基于网络的连接的连接字符串的格式如下所示：

```
protocol_name [options] [host_computer_name] server_name
```

其中，

- options 可以是下列各项的任何组合：

表 12. 连接字符串选项

选项	描述	协议
-4	指定客户机仅使用 IPv4 协议进行连接。	TCP/IP
-6	指定客户机仅使用 IPv6 协议进行连接。 在 Windows 环境中，如果使用 IPv6 协议，那么此选项是必需的。	TCP/IP
-isource_address	当系统缺省源 IP 地址绑定不符合应用程序需求时，指定显式连接套接字源地址。 source_address 可以是 IP 地址或主机名。	TCP/IP
-z	对此连接启用数据压缩功能	All
-c milliseconds	指定登录超时（缺省值特定于操作系统）。在指定的时间过后，登录请求将失败。	TCP/IP
-r milliseconds	指定连接（即读取）超时。如果在指定的时间内未接收到任何响应，那么网络请求将失败。值 0（缺省值）将超时设置为无限（操作系统的缺省超时适用）。	TCP/IP
-ofilename	打开网络跟踪工具并定义跟踪输出文件的名称 有关详细信息，请参阅《IBM solidDB 管理员指南》中的「网络跟踪工具」。	All

表 12. 连接字符串选项 (续)

选项	描述	协议
-plevel	<p>在给定级别 (0-5) 上对服务器执行 ping 操作。</p> <p>客户机始终可以在级别 1 (0 为无操作/缺省值) 上使用 solidDB Ping 工具。仅当已将服务器设置为至少在相同的级别上使用 Ping 工具时, 才能使用级别 2、3、4 或 5。</p> <p>有关详细信息, 请参阅《IBM solidDB 管理员指南》中的『Ping 工具』。</p>	All
-t	<p>打开网络跟踪工具</p> <p>有关详细信息, 请参阅《IBM solidDB 管理员指南》中的『网络跟踪工具』。</p>	All

- 如果客户机与服务器在不同机器上运行, 那么 TCP/IP 协议和命名管道协议需要 *host_computer_name*。
- *server_name* 取决于通信协议:
 - 在 TCP/IP 协议中, *server_name* 是服务端口号 (例如“2315”)。
 - 在其他协议中, *server_name* 是名称 (例如“soliddb”或“chicago_office”)。

有关不同通信协议中语法的详细信息, 请参阅《IBM solidDB 管理员指南》中的『通信协议』。

注:

- *protocol_name* 和 *server_name* 必须与服务器在其网络侦听名称中正在使用的那些值相匹配。
- 如果在连接时提供连接字符串, 那么必须用双引号将其引起来。
- 连接字符串的所有组件都不区分大小写。

sma_connect_string

基于 SMA 的连接的连接字符串的格式如下所示:

```
sma protocol_name port_number | pipe_name
```

使用 SMA 时, 每个节点上的应用程序必须能够利用 SMA 连接来连接到本地服务器, 并利用基于网络的连接来连接到远程服务器。这表示服务器地址列表将采用下列格式:

```
connect_target_list ::= [SERVERS=]<sma_connect_string>, <network_connect_string>
```

```
cluster_info ::= CLUSTER <sma_connect_string>, <network_connect_string>
```

例如:

```
sma tcp 2315, tcp 192.168.255.1 1315
```

驱动程序将从左到右扫描整个列表, 并设法找到主服务器和辅助服务器。因此, 必须将首选配置放在列表开头。列表中的其余部分可能包含一些备用地址, 可能会在系统生存期内的某个时间激活这些备用地址。应让此列表保持简短; 因为在发生错误的情况下, 要经过较长时间之后才会将错误返回给应用程序。将逐个尝试每个地址, 这涉及到所指定的登录超时 (基于网络的连接)。列表中的地址数不受限制。

如果未指定 TF_LEVEL 或 PREFERRED_ACCESS 属性 (或者 TF_LEVEL=NONE), 那么连接行为将后退到使用“基本连接”时的状态。如果给定了多个连接字符串, 那么将与列表中第一个接受连接请求的服务器建立连接。

使用多主目录服务器配置服务器地址

如果您的设置使用多主目录服务器对应用程序与服务器之间以及服务器自身之间的连接部署不同的网络，那么您需要使用 **HotStandby.TCConnect** 参数来定义 TC 连接的服务器地址。

从应用程序连接透视图，使用 **HotStandby.TCConnect** 参数指定的地址优先于使用 **HotStandby.Connect** 参数定义的地址。因此，TC 连接将使用通过 **HotStandby.TCConnect** 参数指定的服务器地址，而服务器之间的 HotStandby 连接将使用通过 **HotStandby.Connect** 参数定义的服务器地址。

有关多主目录服务器配置的示例，请参阅第 82 页的『示例：使用多主目录服务器的 TC 连接』。

CLUSTER

CLUSTER 关键字会将 TF_LEVEL 自动设置为 SESSION，并将 PREFERRED_ACCESS 自动设置为 READ_MOSTLY。

例如，下列 TC 信息和 CLUSTER 字符串可以互换：

```
TF_LEVEL=SESSION PREFERRED_ACCESS=READ_MOSTLY
SERVERS=tcp srv1.acme.com 1315, tcp srv2.acme.com 1315
CLUSTER=tcp srv1.acme.com 1315, tcp srv2.acme.com 1315
```

提示： 集群配置还可定义于客户机端 solid.ini 文件中，在这种情况下，SQLConnect 中的连接字符串可以使用逻辑名。例如：

```
rc = SQLConnect(comHandle, "Cluster1", ...
[Data Sources]
Cluster1=
  TF_LEVEL=SESSION
  PREFERRED_ACCESS=READ_MOSTLY
  SERVERS=
    tcp -c 1000 srv1.dom.acme.com 1315,
    tcp srv2.dom.acme.com 1315,
    tcp srv3.dom.acme.com 1316
```

加密属性

加密属性控制是否对连接密码进行加密。USE_ENCRYPTION 关键字控制是否使用加密。USE_GSKIT 控制是否将 IBM Global Security Kit (GSKit) 用于加密。

- 如果 USE_ENCRYPTION=NO，那么会禁用加密。
- 如果 USE_ENCRYPTION=YES 并且 USE_GSKIT=NO，那么会将 solidDB 内置 DES 算法用于加密。
- 如果 USE_ENCRYPTION=YES 并且 USE_GSKIT=YES，那么会将 GSKit 用于加密。

SQLConnect 示例

```
rc = SQLConnect(comHandle, "TF=CONNECTION
  USE_ENCRYPTION=YES PA=READ_MOSTLY
  SERVERS=
    tcp -c 1000 srv1.dom.acme.com 1315,
    tcp srv2.dom.acme.com 1315,
    tcp srv3.dom.acme.com 1316", ...
```

```
rc = SQLConnect(comHandle, "CLUSTER=
tcp -c 1000 srv1.dom.acme.com 1315,
tcp srv2.dom.acme.com 1315,
tcp srv3.dom.acme.com 1316", ...
```

SMA 设置: Node1 (srv1.dom.acme.com) 的应用程序上的 SQLConnect

```
rc = SQLConnect(comHandle, "TF=CONNECTION
PA=LOCAL_READ
SERVERS=
sma tcp 1315,
tcp srv2.dom.acme.com 2315", ...
```

SMA 设置: Node2 (srv2.dom.acme.com) 的应用程序上的 SQLConnect

```
rc = SQLConnect(comHandle, "TF=CONNECTION
PA=LOCAL_READ
SERVERS=
sma tcp 2315,
tcp srv1.dom.acme.com 1315", ...
```

客户机端 solid.ini 示例

注: 由于布局原因, 将下列示例中的 Com.Connect 参数值分隔为几行。在您的 solid.ini 文件中, 整个参数条目必须在一行中。

```
[Com]
Connect = TF=CONNECTION USE_ENCRYPTION=YES
PA=READ_MOSTLY
SERVERS=
tcp -c 1000 srv1.dom.acme.com 1315,
tcp srv2.dom.acme.com 1315,
tcp srv3.dom.acme.com 1316
```

SMA 设置: Node1 (srv1.dom.acme.com) 上的 solid.ini 文件

```
[Com]
Connect = TF=CONNECTION USE_ENCRYPTION=YES
PA=LOCAL_READ
SERVERS=
sma tcp 1315,
tcp srv2.dom.acme.com 2315
```

SMA 设置: Node2 (srv2.dom.acme.com) 上的 solid.ini 文件

```
[Com]
Connect = TF=CONNECTION USE_ENCRYPTION=YES
PA=LOCAL_READ
SERVERS=
sma tcp 2315,
tcp srv1.dom.acme.com 1315
```

使用 JDBC 的透明连接

通过 JDBC, 使用非标准连接属性来启用透明连接。服务器地址列表作为 JDBC 连接字符串的一部分进行提供。

注: 在 JDBC 中使用透明连接时, 必须谨慎处理语句对象的显式删除操作。垃圾回收器将不会检测未引用的语句对象。

故障透明性级别 (solid_tf_level)

可使用 solid_tf_level 连接属性来启用故障透明性。值是字符串: 您可以将其指定为助记符 (例如, NONE) 或数字 (0, 对应于 NONE)。为了使表达清晰, 首选使用助记符。

有三个故障透明性级别:

1. NONE | 0 - 已禁用故障透明性。这是缺省值。
2. CONNECTION | 1 - 将保持服务器连接, 也就是说, 在进行故障转移或切换时不需要重新连接。
3. SESSION | 3 - 将保持某些具有非缺省值的会话属性。此外, 还将保留预编译语句。但是, 打开的游标会关闭, 并且正在进行的事务会异常终止。

首选访问属性 (**solid_preferred_access**)

首选访问属性指示是否分配了只读负载。使用 `solid_preferred_access` 连接属性来启用首选访问属性。值是字符串: 您可以将其指定为助记符或数字。为了使表达清晰, 首选使用助记符。

可使用下列级别:

- WRITE MOSTLY | 0 - 将读工作负载定向至主服务器。这是缺省值。WRITE MOSTLY 还会将连接设置为 WRITE MOSTLY 方式。通过指定数值不能完成此任务。
- READ MOSTLY | 1 - 按照 `Cluster.ReadMostlyLoadPercentAtPrimary` 参数的定义, 将读工作负载定向至辅助服务器和主服务器。写事务将移交给主服务器。

缺省情况下, `Cluster.ReadMostlyLoadPercentAtPrimary` 参数设置为 50, 这意味着百分之五十的读负载将定向至主服务器。

重新连接超时 (**solid_tf1_reconnect_timeout**)

`solid_tf1_reconnect_timeout` 属性会指定连接的重新连接超时 (以毫秒计)。缺省值为 10000 毫秒 (也就是 10 秒)。

服务器地址

作为扩展 JDBC 连接字符串的一部分提供了服务器地址列表:

```
conStr= "jdbc:solid://host_name:port [,host_name:port].../user_name/password";
```

地址列表中的地址数不能超过 20。

使用多主目录服务器配置服务器地址

如果您的设置使用多主目录服务器对应用程序与服务器之间以及服务器自身之间的连接部署不同的网络, 那么您需要使用 `HotStandby.TCConnect` 参数来定义 TC 连接的服务器地址。

从应用程序连接透视图, 使用 `HotStandby.TCConnect` 参数指定的地址优先于使用 `HotStandby.Connect` 参数定义的地址。因此, TC 连接将使用通过 `HotStandby.TCConnect` 参数指定的服务器地址, 而服务器之间的 HotStandby 连接将使用通过 `HotStandby.Connect` 参数定义的服务器地址。

SMA 连接 (**solid_shared_memory**)

使用 SMA 配置时, 必须将非标准属性 `solid_shared_memory` 设置为 yes。还必须定义将在给定端口上使用本地服务器。

示例: 使用故障透明性级别 CONNECTION 的 TC 连接

```

...
String conStr = "jdbc:solid://srv1.acme.com:1323,srv2-acme.com:1423/dba/dba";
Properties prop = new Properties();
prop.setProperty("solid_tf_level", "CONNECTION");
...
Connection c = DriverManager.getConnection(conStr, prop);
...

```

示例: 使用 SMA 的 TC 连接

```

...
String conStr = "jdbc:solid://localhost:1323,srv2-acme.com:1423/dba/
dba?solid_shared_memory=yes";
Properties prop = new Properties();
prop.setProperty("solid_tf_level", "CONNECTION");
Connection c = DriverManager.getConnection(conStr, prop);
...

```

示例: 使用负载均衡的 TC 连接

```

...
String conStr = "jdbc:solid://12.345.67.88:1323,12.345.67.89:1423/dba/dba";
Properties prop = new Properties();
prop.setProperty("solid_tf_level", "CONNECTION");
prop.setProperty("solid_preferred_access", "READ_MOSTLY");
...
Connection c = DriverManager.getConnection(conStr, prop);
...

```

示例: 使用负载均衡并定义了 JDBC URL 的 TC 连接

```

jdbc:solid://12.345.67.88 1323, 12.345.67.89 1423/
dba/dba?solid_tf_level=1?solid_preferred_access=READ_MOSTLY

```

TC 属性组合

下表总结了可能存在的 TC 属性组合并提供了所产生的连接能力:

表 13. 可能存在的 TC Info 属性组合

PREFERRED _ ACCESS:	TF_LEVEL: 未指定 或者 NONE	TF_LEVEL: CON- NECTION	TF_LEVEL: SES- SION
未指定	<ul style="list-style-type: none"> 不支持故障转移或切换 不进行负载均衡 (基本连接) 	<ul style="list-style-type: none"> 透明故障转移 (不保留会话状态) 透明切换 仅适用于主服务器中的工作负载 不进行负载均衡 	<ul style="list-style-type: none"> 透明故障转移 (保留会话状态) 透明切换 仅适用于主服务器中的工作负载 不进行负载均衡
WRITE_MOSTLY (缺省值)	<ul style="list-style-type: none"> 不支持透明故障转移 透明切换 仅适用于主服务器中的工作负载 不进行负载均衡 	<ul style="list-style-type: none"> 透明故障转移 (不保留会话状态) 透明切换 仅适用于主服务器中的工作负载 不进行负载均衡 	<ul style="list-style-type: none"> 透明故障转移 (保留会话状态) 透明切换 仅适用于主服务器中的工作负载 不进行负载均衡

表 13. 可能存在的 TC Info 属性组合 (续)

PREFERRED_ACCESS:	TF_LEVEL: 未指定或者 NONE	TF_LEVEL: CONNECTION	TF_LEVEL: SESSION
READ_MOSTLY	<ul style="list-style-type: none"> 不支持透明故障转移 透明切换 辅助服务器和主服务器中的工作负载 负载均衡 	<ul style="list-style-type: none"> 透明故障转移 (不保留会话状态) 透明切换 辅助服务器和主服务器中的工作负载 负载均衡 	<ul style="list-style-type: none"> 透明故障转移 (保留会话状态) 透明切换 辅助服务器和主服务器中的工作负载 负载均衡
LOCAL_READ	<ul style="list-style-type: none"> 不支持透明故障转移 透明切换 主服务器使用 SMA 连接在本地执行的读/写操作 辅助服务器使用 SMA 连接在本地执行的读操作以及使用网络连接在主服务器上执行的写操作 	<ul style="list-style-type: none"> 透明故障转移 (不保留会话状态) 透明切换 主服务器使用 SMA 连接在本地执行的读/写操作 辅助服务器使用 SMA 连接在本地执行的读操作以及使用网络连接在主服务器上执行的写操作 	<ul style="list-style-type: none"> 透明故障转移 (保留会话状态) 透明切换 主服务器使用 SMA 连接在本地执行的读/写操作 辅助服务器使用 SMA 连接在本地执行的读操作以及使用网络连接在主服务器上执行的写操作

连接错误处理

对 TC 连接发出连接请求之后，如果至少找到了一个合适的服务器并且已连接至此服务器，那么就认为此连接请求成功。

服务器应处于下列任一状态：PRIMARY ACTIVE、PRIMARY ALONE 或 STAND-ALONE。否则，就认为连接失败。将扫描地址列表一次。

可能有各种不同的原因会造成连接请求失败。下列错误情况涵盖了大多数原因：

表 14. 连接请求错误

SQLSTATE	本机代码	消息文本和描述
08001	25217	<p>Client unable to establish a connection</p> <p>描述: 驱动程序已使用 TC 连接信息来查找合适的服务器并与其建立连接。但是, 由于下列原因之一而失败:</p> <ul style="list-style-type: none"> 找不到地址列表中所列示的主机 找到了主机, 但是登录时发生超时 找到了主机, 但是登录时被拒绝 找到了主机, 但是不处于 PRIMARY/STANDALONE 状态
HY000	21307	<p>Invalid connect info...</p> <p>描述: 在基本连接字符串或者 TC 连接信息 (数据源信息) 中发现了语法错误。</p>
HY000	21300	<p>Protocol ... not supported.</p> <p>描述: TC 连接信息开头的字符串“TC”存在拼写错误 (或者在基本连接字符串中提供的协议名称不正确)。</p>

在某些情况下接受连接时发出了警告。

表 15. 警告

SQLSTATE	本机代码	消息文本和描述
0100	25218	<p>Connected to Standalone or Primary Alone server.</p> <p>已经尝试为 TF_LEVEL 或 PREFERRED_ACCESS 设置任何非缺省值, 并且只有一个服务器可用。在这种情况下, 故障透明性和负载均衡都不可用。</p>

示例: TC 连接

此示例显示适用于典型 HotStandby 设置的透明连接配置, 在该设置中应用程序与服务 器位于同一网络内。

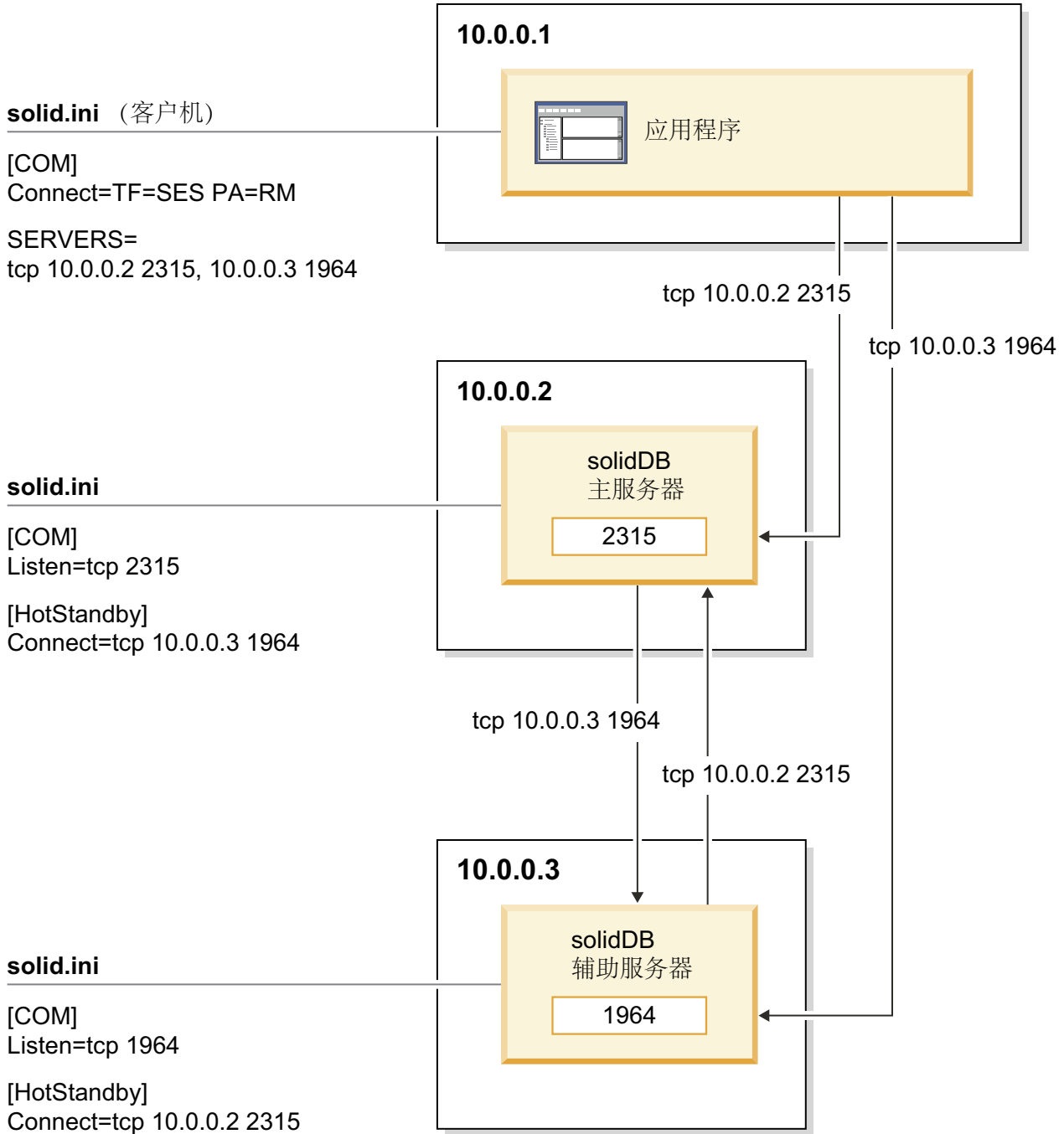


图 14. 示例: TC 连接

示例: 使用多主目录服务器的 TC 连接

此示例显示适用于 HotStandby 设置的透明连接配置，其中主服务器和辅助服务器都是多主目录服务器。在应用程序和服务器处于不同网络内的设置中，或者应用程序不能或不应使用用于服务器之间的 HotStandby 连接的相同网络来连接服务器的设置中，这是典型情况。

如果要区分 TC 连接与 HotStandby 连接，需要使用两个用来定义服务器地址的参数：

- **HotStandby.Connect** 参数会指定用于主服务器与辅助服务器之间的 HotStandby 连接的服务器地址。
- **HotStandby.TCConnect** 参数会指定用于 TC 连接的服务器地址。

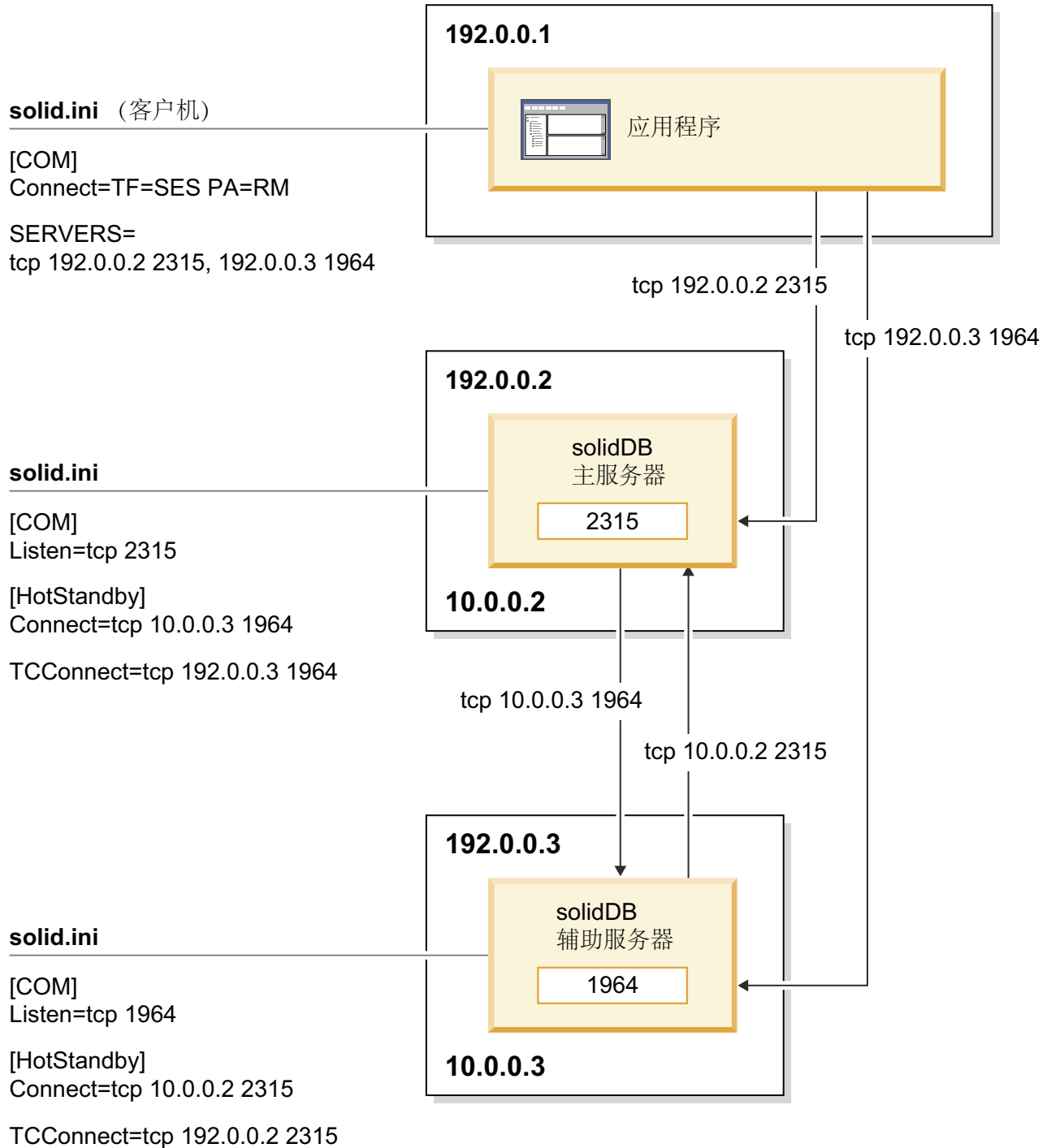


图 15. 示例：使用多主目录服务器的 TC 连接

4.2.2 透明连接中的故障透明性

故障透明性用于处理对故障的掩盖。故障透明性级别使用 `TF_LEVEL` 属性进行设置。

有三个故障透明性级别：

`TF_LEVEL=NONE`

已禁用故障透明性（缺省值）。

`TF_LEVEL=CONNECTION`

将保持服务器连接，也就是说，在进行故障转移或切换时不需要重新连接。

`TF_LEVEL=SESSION`

大部分非缺省会话级别设置将予以保留。此外，还将保留预编译语句。但是，打开的游标会关闭，并且正在进行的事务会异常终止。

连接切换的处理原则

连接切换是指驱动程序更改活动服务器连接这种情况。通常，在故障转移到辅助服务器或者在服务器之间切换时就会进行连接切换。

更具体地说，从下列其中一个事件中检测到需要进行连接切换：

- 辅助服务器中使得状态更改为 `PRIMARY ALONE`（故障转移）或 `PRIMARY ACTIVE`（切换）的事件。这是执行连接切换的主要方式（也是速度最快的方式）。
- 指示主服务器中发生了状态更改。
- 活动连接上发生链路故障。
- 活动连接上发生连接超时。

驱动程序通过以下两个步骤来执行连接切换：

1. 检测到需要进行连接切换。对于暂挂的请求或以下请求，驱动程序将返回以下连接切换错误：

表 16. 连接切换请求

SQLSTATE	本机代码	消息文本和描述
HY000	25216	Connection switch, some session context may be lost 描述：驱动程序已发现需要进行连接切换。期望客户机发出事务回滚调用以最终确定连接切换。在每个连续的网络请求调用时都将接收到此错误代码和消息，直到发出了回滚调用为止

2. 客户机程序发出回滚命令（ODBC: `SQLEndTran()` with `SQL_ROLLBACK`; JDBC: `Connection.rollback()`）。如果成功完成了回滚，那么已将新的活动连接映射至可以使用的 TC 连接。

注：一些连续的 ODBC 调用可能会返回连接切换错误。因此，必须规定当任何 ODBC 网络请求返回此错误时始终都进行回滚。如果在执行事务过程中发生此错误，那么必须重新执行此事务。

另一方面，如果无法建立新的活动连接，那么将返回以下错误代码：

表 17. 通信链路故障

SQLSTATE	本机代码	消息文本和描述
08S01	14503	<p>Communication link failure</p> <p>描述：驱动程序未能建立新的活动连接。TF 连接已断开，客户机需要使用数据源信息重新建立连接才能进行后续操作。</p>

在接收到回滚调用之后，驱动程序将使用一些备用方法来查找新的活动连接。在最简单的情况下，它将使用备用连接。如果此连接处于不正确的状态，那么驱动程序将等待两秒钟以等待正确的事件达到。如果事件未达到以及其他情况下，驱动程序将后退到 TC 连接信息中的地址列表，并且最多将在 10 秒钟之内反复执行连接序列。如果所有操作都失败，那么驱动程序将返回上述错误。

此错误导致的结果就是连接断开，正如应用程序所看到的。对此连接发出的任何后续请求都将导致相同错误。

会话状态的保留

在驱动程序执行连接切换之后，某些会话上下文可能会丢失，应用程序必须重建这些会话上下文。保留的状态数量由透明故障转移级别指定，用 TC Info 属性 TF_LEVEL 来表示透明故障转移级别。

对于 TF 级别 CONNECTION，不会保留任何状态。对于 SESSION 级别，将保留大多数会话状态。会话状态的保留是通过将必需数据高速缓存在驱动程序中来实现的。如果要获得更高的透明性级别，需要进行高速缓存的请求的响应时间就会越长，并且驱动程序中的内存使用量也会增大。

无论对于哪个 TF 级别，进行故障转移时都会应用下列各项：

- 由于进行事务回滚而丢失了当前事务的更新。
- 丢失了打开的游标及其位置。

表 18. 会话状态保留

TF_LEVEL	保留的状态
CONNECTION	不保留会话状态。

表 18. 会话状态保留 (续)

TF_LEVEL	保留的状态
SESSION	<p>预编译语句</p> <ul style="list-style-type: none"> • 保留预编译状态。 <p>保留了下列语句的作用:</p> <ul style="list-style-type: none"> • SET CATALOG • SET SQL INFO • SET SQL SORTARRAYSIZE • SET SQL CONVERTORSTOUNIONS • SET SQL JOINPATHSPAN • SET LOCK TIMEOUT <seconds> • SET IDLE TIMEOUT • SET OPTIMISTIC LOCK TIMEOUT • SET STATEMENT MAXTIME • SET ISOLATION LEVEL • SET DURABILITY • SET SAFENESS • SET SCHEMA • SET SYNC USER • SET SYNC MODE <p>保留了下列标准 ODBC 属性:</p> <ul style="list-style-type: none"> • SQL_ATTR_ACCESS_MODE <p>(SET READ ONLY 和 SET READ WRITE)</p> <ul style="list-style-type: none"> • SQL_ATTR_CURRENT_CATALOG <p>(与上面的 SET CATALOG 作用相同)</p> <ul style="list-style-type: none"> • SQL_ATTR_AUTOCOMMIT

4.2.3 透明连接中的负载均衡

通过 TC，驱动程序使用两种方法来转移事务负载；一种方法用来处理读取密集型负载，另一种方法用来处理写入密集型负载。为了负载均衡，会将逻辑 TC 连接映射至更低级别的、称为工作负载连接的服务器连接。工作负载连接可以随时间而变化，它对于应用程序通常无关紧要。但是，如果工作负载连接是必需的，那么可以通过一种方法来识别当前工作负载连接的内容。

静态负载均衡配置

负载均衡方法包括:

PREFERRED_ACCESS=WRITE_MOSTLY - 无负载均衡 (缺省值)

通过 WRITE_MOSTLY，所有事务都在主服务器中执行。这对应于典型 HotStandby 操作。WRITE_MOSTLY 方法对写入密集型负载非常有用。

PREFERRED_ACCESS=READ_MOSTLY - 通过在主服务器与辅助服务器之间分发只读事务来执行负载均衡

通过 `READ_MOSTLY`，可以在辅助服务器和主服务器中执行只读事务。写入事务始终在主服务器中执行。

对于只读事务，会基于 `Cluster.ReadMostlyLoadPercentAtPrimary` 参数设置来选择指定的工作负载服务器，该设置会指定转移到主服务器的总主要读负载的百分比。

`Cluster.ReadMostlyLoadPercentAtPrimary` 参数的缺省值是 50，这表示缺省情况下，50% 的连接使用主服务器，50% 的连接使用辅助服务器。对于大多数混合型负载来说，这是更合适的值。如果此值设置为 0，那么所有负载都将转移到辅助服务器中。在非常读取密集型（或者只读）应用程序使用 `PREFERRED_ACCESS=READ_MOSTLY` 并且（同时）写入密集型应用程序使用 `PREFERRED_ACCESS=WRITE_MOSTLY` 时，将此值设置为 0 是合适的。

通过 `READ_MOSTLY`，主服务器告诉驱动程序为了工作负载连接而连接至哪个服务器。如果将给定连接的负载转移至辅助服务器，并且发出了写操作，那么就会将负载移交给主服务器并在主服务器中执行事务。在落实事务之后，又会将负载转移回辅助服务器。如果辅助服务器失败，那么连接将从辅助服务器故障转移到主服务器。

PREFERRED_ACCESS=LOCAL_READ - 通过在可能时于本地执行事务来执行负载均衡

通过 `LOCAL_READ`，始终会将只读事务转移到作为主服务器或辅助服务器的本地服务器。写入事务始终在主服务器中执行。

`LOCAL_READ` 方法通常与 `SMA` 设置配合使用。主服务器上的应用程序将 `SMA` 连接用于读取和写入事务。辅助服务器上的应用程序将 `SMA` 连接用于读取操作，将基于网络的连接用于主服务器上的写入操作。

要点：将 `SMA` 与透明连接（`TC`）配合使用时，如果将负载均衡方法设置为 `READ_MOSTLY` 或 `WRITE_MOSTLY`（缺省值），那么将使用网络连接而不是 `SMA` 连接。因此，将 `SMA` 与 `TC` 配合使用时，请始终将负载均衡方法设置为 `LOCAL_READ`。

隔离级别和负载均衡

仅在“落实读”隔离级别执行负载均衡操作。如果服务器的隔离级别（启动）缺省值设置为其他值，那么设置 `PREFERRED_ACCESS=READ_MOSTLY` 和 `PREFERRED_ACCESS=LOCAL_READ` 会将此会话的隔离级别强制设置为“落实读”。可以将隔离级别动态重置为更高级别（例如“可重复读”），但是这会禁用负载均衡。

自动落实和负载均衡

要使用负载均衡，必须禁用会话自动落实。

动态控制负载均衡

使用负载均衡（`READ_MOSTLY` 或 `LOCAL_READ`）时，您可以通过程序将指定的工作负载服务器从辅助服务器更改为主服务器。

在会话级别，可以使用下列语句将工作负载连接服务器更改为主服务器：

- `SET WRITE`
- `SET ISOLATION LEVEL REPEATABLE READ`

- SET ISOLATION LEVEL SERIALIZABLE

如果此语句是事务的第一个语句，那么它将立即生效，否则将从下一个事务开始生效。

在事务级别，下列语句会在一个事务的时间内将工作负载连接服务器更改为主服务器：

- SET TRANSACTION WRITE
- SET TRANSACTION ISOLATION LEVEL REPEATABLE READ
- SET TRANSACTION ISOLATION LEVEL SERIALIZABLE

受影响的事务是通过使用此语句启动的事务，否则将为下一个事务。在主服务器中执行事务之后，工作负载连接服务器将恢复为对会话指定的那个工作负载连接服务器。

使用 SET [TRANSACTION] READ WRITE 语句可以恢复 SET [TRANSACTION] WRITE 语句的效果。另外，下列隔离级别语句具有相同的效果：

- SET ISOLATION LEVEL READ COMMITTED
- SET TRANSACTION ISOLATION LEVEL READ COMMITTED

注：使用 READ_MOSTLY 时，将使用 **Cluster.ReadMostlyLoadPercentAtPrimary** 参数来定义主服务器与辅助服务器之间读负载的分配比例。缺省情况下，该参数的值设置为 50，这意味着百分之五十的读负载将定向至主服务器。更改 **Cluster.ReadMostlyLoadPercentAtPrimary** 参数的值时，需要重新启动服务器。

相关信息：

第 116 页的 A.1.1，『 [Cluster] 节 』

故障转移透明性和负载均衡

当设置了故障透明性（**TF_LEVEL** 的值不为 NONE）并且启用了负载均衡（**PREFERRED_ACCESS=READ_MOSTLY** 或 **PREFERRED_ACCESS=LOCAL_READ**）时，所应用的故障转移策略如下所示：

1. 主服务器故障：所有负载都转移到新的处于 PRIMARY ALONE 状态的主服务器。
2. 辅助服务器故障：所有负载都转移到主服务器（处于 PRIMARY ALONE 状态）。
3. 服务器之间的连接断开；服务器处于 PRIMARY ALONE 和 SECONDARY ALONE 状态：如果有一个只读事务正在辅助服务器中执行，那么也会在辅助服务器中成功落实此事务。所有后续事务都将转移到主服务器中（处于 PRIMARY ALONE 状态）。

当恢复正常的热备用操作时（服务器分别处于 PRIMARY ACTIVE 和 SECONDARY ACTIVE 状态），就会在主服务器与辅助服务器之间重新均衡负载。

注：即使未启用故障透明性（**TF_LEVEL=NONE**），某些基本故障转移功能仍然可用：当辅助服务器失败时，从辅助服务器故障转移到主服务器。所有其他故障都将导致通信链路失败。因此，在大部分设置为 **TF_LEVEL=NONE** 的故障情况下，应用程序必须使用相同的 TC 信息进行重新连接。如果要避免重新连接，请在使用负载均衡时启用故障透明性。

在实现了负载均衡的情况下执行存储过程

除非在过程声明中通过使用 SQL 标准子句 *SQL_data_access_indication* 将 SQL 存储过程指定为只读过程，否则所有 SQL 存储过程都将在主服务器中执行。

```
<SQL_data_access_indication> ::=  
    NO SQL |  
    READS SQL DATA |  
    CONTAINS SQL |  
    MODIFIES SQL DATA
```

只有关键字 `MODIFIES SQL DATA` 才损害事务移交。这是缺省的行为。

为了避免不必要地移交只读过程和函数，请使用下列其中一个值：

- `NO SQL`
- `READS SQL DATA`
- `CONTAINS SQL`

4.2.4 处理 TC Info 矛盾

TC Info 的属性可能会与“使实际服务可用”产生矛盾。在这些情况下，将保证建立连接，但是会发出 `SUCCESS_WITH_INFO` 警告。

在下列情况下将处理矛盾：

- 指定了 `PREFERRED_ACCESS`，但是未启用 HSB。将启用基本连接。
- 指定了 `TF_LEVEL`，但是未启用 HSB。将启用基本连接。

4.3 基本连接

使用“基本连接”时，应用程序必须处理通过使用特定服务器地址单独连接至 HotStandby 或集群配置中的每个服务器。如果发生了故障转移，那么活动连接将断开，应用程序必须重新连接至新的主服务器。

基本连接语法

基本连接使用标准的 `solidDB` 连接字符串语法：

```
protocol_name [options] [host_computer_name] server_name
```

其中，

- *options* 可以是下列各项的任何组合：

表 19. 连接字符串选项

选项	描述	协议
-4	指定客户机仅使用 IPv4 协议进行连接。	TCP/IP
-6	指定客户机仅使用 IPv6 协议进行连接。 在 Windows 环境中，如果使用 IPv6 协议，那么此选项是必需的。	TCP/IP
- <i>source_address</i>	当系统缺省源 IP 地址绑定不符合应用程序需求时，指定显式连接套接字源地址。 <i>source_address</i> 可以是 IP 地址或主机名。	TCP/IP
-z	对此连接启用数据压缩功能	All
-c <i>milliseconds</i>	指定登录超时（缺省值特定于操作系统）。在指定的时间过后，登录请求将失败。	TCP/IP

表 19. 连接字符串选项 (续)

选项	描述	协议
-r <i>milliseconds</i>	指定连接（即读取）超时。如果在指定的时间内未接收到任何响应，那么网络请求将失败。值 0（缺省值）将超时设置为无限（操作系统的缺省超时适用）。	TCP/IP
-o <i>filename</i>	打开网络跟踪工具并定义跟踪输出文件的名称 有关详细信息，请参阅《 <i>IBM solidDB 管理员指南</i> 》中的『网络跟踪工具』。	All
-p <i>level</i>	在给定级别 (0-5) 上对服务器执行 ping 操作。 客户机始终可以在级别 1 (0 为无操作/缺省值) 上使用 solidDB Ping 工具。仅当已将服务器设置为至少在相同的级别上使用 Ping 工具时，才能使用级别 2、3、4 或 5。 有关详细信息，请参阅《 <i>IBM solidDB 管理员指南</i> 》中的『Ping 工具』。	All
-t	打开网络跟踪工具 有关详细信息，请参阅《 <i>IBM solidDB 管理员指南</i> 》中的『网络跟踪工具』。	All

- 如果客户机与服务器在不同机器上运行，那么 TCP/IP 协议和命名管道协议需要 *host_computer_name*。
- *server_name* 取决于通信协议：
 - 在 TCP/IP 协议中，*server_name* 是服务端口号（例如“2315”）。
 - 在其他协议中，*server_name* 是名称（例如“soliddb”或“chicago_office”）。

有关不同通信协议中语法的详细信息，请参阅《*IBM solidDB 管理员指南*》中的『通信协议』。

注:

- *protocol_name* 和 *server_name* 必须与服务器在其网络侦听名称中正在使用的那些值相匹配。
- 如果在连接时提供连接字符串，那么必须用双引号将其引起来。
- 连接字符串的所有组件都不区分大小写。

例如:

```
Connect=tcp srv1.dom.acme.com 1315
```

4.3.1 从应用程序重新连接至主服务器

为 HotStandby 准备客户机应用程序

已经与主服务器断开连接的客户机程序必须能够重新连接至新的主服务器（也就是旧的辅助服务器）。您必须编写客户机应用程序以便能够完成下列任务:

1. 识别主服务器不再可用于写入事务。
2. 连接至另一个服务器或者切换为使用先前创建的连接。
3. 考虑当前已中断的事务是否已丢失/异常终止，并且必须新的主服务器上重新执行。

获取辅助服务器地址

最容易获取辅助数据库服务器的连接信息的方法是使用 **ADMIN COMMAND 'hotstandby cominfo'** 命令，此命令将为 HSB 对中的另一个服务器提供连接信息。

关于此任务

过程

1. 当应用程序首次连接至主服务器时，应用程序可以执行 **ADMIN COMMAND 'hotstandby cominfo'** 命令并存储结果。注意，当 **cominfo** 命令返回一个值时，它并不意味着主服务器与辅助服务器当前已建立连接。“cominfo”命令仅仅返回在 `solid.ini` 配置文件的 **Connect** 参数中指定的值，或者返回最近使用 **hsb parameter connect** 命令指定的值。如果需要检查主服务器与辅助服务器之间的连接状态，那么可以使用 **ADMIN COMMAND 'hotstandby status connect'**。
2. 稍后，如果主服务器失败，那么应用程序可以使用所存储的信息来连接至辅助服务器（新的主服务器）。

在客户机应用程序中检测 HotStandby 服务器故障

要使用 HotStandby (HSB) 组件，应用程序必须知道何时从失败的主服务器切换到辅助服务器（也就是新的主服务器）。这可以通过几种方法来实现。最佳方法是仅检查您调用函数之后所产生的返回码，以了解您是否接收到错误并指出您应当切换到另一个服务器。

还可以监视服务器的状态（例如，检查主服务器以了解其状态是否已更改为 **PRIMARY UNCERTAIN**）。

如果发生下列错误，那么您应当尝试切换到另一个服务器：

- 10013: 事务是只读的
- 10041: 数据库是只读的
- 10047: 复制事务异常终止
- 11002: 磁盘已满
- 11003: 文件写入失败，超出了配置
- 14501: 操作失败
- 14502: RPC 参数无效
- 14503: 通信错误
- 14506: 服务器已关闭（例如，由于它当前是 **HSB netcopy** 操作的目标而关闭）
- 14510: 通信写入操作失败
- 14511: 通信读取操作失败
- 14518: 连接已断开
- 14519: 用户被拒绝（例如，由于某些管理操作而被拒绝）
- 14529: 操作超时
- 20009: 会话错误，写入操作失败
- 21306: 未找到服务器，连接失败
- 21308: 连接已断开（写入失败并提供了代码...）
- 21318: 操作失败（不常见的返回码）

ODBC 应用程序

对于无法建立连接的 ODBC 应用程序（例如，由于数据库服务器无法运行而无法建立连接）将返回以下错误消息：

- `SQLState = 08001` - 客户机无法建立连接

此外，将产生以下 `solidDB` 通信错误消息：

- 21306 - 未找到服务器 “`server_name`”，连接失败。

如果在双向操作（例如，执行查询并访存结果）中连接失败（例如，由于发生网络故障而连接失败），那么将返回以下错误消息：

- `SQLState = 08S01` - 通信链路故障

JDBC 应用程序

对于无法建立连接的 JDBC 应用程序（例如，由于数据库无法运行而无法建立连接）将返回以下错误消息：

- `SQLState = 08001` - 无法连接至数据源。

如果在双向操作（例如，执行查询并访存结果）中连接失败（例如，由于发生网络故障而连接失败），那么将返回以下错误消息：

- `SQLState = 08S01` - 通信链路故障

注：

对于同一个错误代码 (08001)，ODBC 和 JDBC 将提供不同的错误消息。

将应用程序切换到新的主服务器

在应用程序检测到它无法将事务发送至“旧的主服务器”之后，应用程序必须轮询旧的主服务器和旧的辅助服务器，直到找到处于 `PRIMARY ACTIVE`、`PRIMARY ALONE` 或 `STANDALONE` 状态的服务器为止。

通过让应用程序尝试连接至服务器并在建立连接之后检查服务器状态来完成轮询。连接成功之后，客户机可以使用 SQL 函数 `HOTSTANDBY_STATE` 来请求服务器状态，在第 93 页的『使用 `HOTSTANDBY_STATE` 函数』这一节中描述了此函数。

注意：

切换之后，所有打开的数据库对象（例如，预编译语句）以及打开的游标和事务将不再处于活动状态。因此，必须再次初始化这些对象。此外，如果您先前使用的是临时表或瞬态表（`solidDB` 主内存引擎功能部件），那么这些表在新的主服务器上将为空。

使用 `HOTSTANDBY_CONNECTSTATUS` 函数

要在从应用程序重新连接至主服务器时验证连接状态信息，可以使用 `HOTSTANDBY_CONNECTSTATUS` 函数。此函数等价于管理命令 `hotstandby status connect`。

此函数没有自变量，并且将返回下列状态值之一：

表 20. `HOTSTANDBY_CONNECTSTATUS` 状态值

状态	描述
已连接	连接处于活动状态。从主服务器和辅助服务器中都将返回此状态。

表 20. *HOTSTANDBY_CONNECTSTATUS* 状态值 (续)

状态	描述
正在连接	主服务器正在连接至辅助服务器。从主服务器和辅助服务器中都将返回此状态。
同步复制	主服务器已连接至辅助服务器，但是尚未完全复制事务日志。从主服务器和辅助服务器中都将返回此状态。
已断开连接	连接已断开。从主服务器和辅助服务器中都将返回此状态。

使用 `HOTSTANDBY_STATE` 函数

要实现使用应用程序来轮询主服务器和辅助服务器，可以使用 `HOTSTANDBY_STATE` 函数来完成。此函数等价于管理命令 `hotstandby state`。在当前 HotStandby 已连接至服务器的情况下，此函数允许应用程序请求当前 HotStandby 的状态。

注：此函数没有自变量。有关此函数可能返回的每种状态的描述，请参阅第 60 页的 3.4.8, 『验证 HotStandby 服务器状态』。

样本伪码

无论应用程序是否支持 HSB，它都应该具有错误处理方法，以允许此应用程序重放失败/异常终止的事务。

在非 HSB 环境中，事务可能会因并行冲突（乐观表）或死锁（悲观表）而异常终止。应用程序必须发现这些错误情况，并且自动重试此事务或者要求交互用户重新执行此事务。

如果应用程序中已经具有代码用来处理失败事务或者异常终止的事务，那么扩展此代码以利用 HSB 就相对比较容易。

在一个非常简单的示例中，具有不了解 HA 的应用程序处理的正确错误处理的应用程序伪码看起来类似于以下内容：

```
BEGIN TRANSACTION
EXECUTE APPLICATION LOGIC
PREPARE & EXECUTE STATEMENTS
COMMIT TRANSACTION
IF ERROR OCCURRED
    IF ERROR == concurrency conflict or deadlock
        GO TO BEGIN TRANSACTION
    END IF
    other error handling
END IF ;
```

很容易改进上述应用程序以使它了解 HA。您必须增加一些代码，以使应用程序可以完成下列任务：

- 连接至两个服务器中的任何一个，而不是只能连接至一个服务器；并且
- 发生错误时，找出当前处于下列其中一种状态的服务器：`PRIMARY ACTIVE`、`PRIMARY ALONE` 或 `STANDALONE`。

伪码看起来应该类似于以下内容：

```

BEGIN TRANSACTION
EXECUTE APPLICATION LOGIC
PREPARE & EXECUTE STATEMENTS
COMMIT TRANSACTION
IF ERROR OCCURRED
    IF ERROR == server unavailable for write transactions
        FIND CURRENT PRIMARY SERVER
        GO TO BEGIN TRANSACTION
    END IF
    IF ERROR == concurrency conflict or deadlock
        GO TO BEGIN TRANSACTION
    END IF
    IF ERROR == something else
        other error handling
    END IF
END IF

```

用于查找当前主服务器的逻辑也非常简单。只需同时检查两个服务器当前的状态（必要时尝试重新连接），如果其中任何一个服务器处于 **PRIMARY ACTIVE**、**PRIMARY ALONE** 或 **STANDALONE** 状态，那么将此服务器设置为当前的主服务器。如果两个服务器都不满足此条件，那么稍等一会儿，然后重新检查当前服务器状态。

4.3.2 重新连接至辅助服务器

在某些情况下，您可能需要连接至当前辅助服务器（如果它已经启动）。应用程序可以向辅助服务器提交只读查询；这在有时候可以帮助您均衡服务器之间的工作负载。

应用程序只能以只读方式连接至辅助数据库。请注意，客户机可以通过使用服务器的 `solid.ini` 配置文件的 `[HotStandby]` 节中的下列参数值来连接至辅助服务器（只能以只读方式进行连接）：

- 主服务器中的 **Connect** 参数
- 辅助服务器中的 **Listen** 参数

还可以使用以下命令来获取服务器的伙伴的连接信息：

```
ADMIN COMMAND 'hotstandby cominfo';
```

因此，如果您已连接至当前的主服务器，那么可以通过使用 `cominfo` 查询来获取当前辅助服务器的地址。

4.4 定义应用程序与服务器之间的超时

本节描述如何通过使用 `solid.ini` 的 **Connect** 参数或者适用于 ODBC 的 `SQLConnect` 函数的连接字符串来配置应用程序读取超时和连接超时设置。

这些超时值将应用于服务器与客户机应用程序之间的连接，这些客户机应用程序包括 `solidDB SQL 编辑器 (solsql)`、`solidDB 远程控制 (solcon)` 和 `HA 管理器`。

4.4.1 应用程序读取超时选项

应用程序读取超时选项可帮助检测低级别网络 RPC 读取操作中发生的故障。超时设置应用于物理网络中的读取操作（仅适用于 TCP/IP 协议）。

此 RPC 读取超时（在 ODBC 和 JDBC 中称为连接超时）可采用下列方法进行配置（以毫秒计）。缺省超时值为 0 - 无限（操作系统缺省超时适用），

- 客户机端 **Com.ClientReadTimeout** 参数

例如:

```
[Com]
;Set RPC read timeout to 1000 milliseconds (one second)
ClientReadTimeout=1000
```

- 带有 **-r** *milliseconds* 选项的客户机端 **Com.Connect** 参数

例如:

```
[Com]
;Set RPC read timeout to 1000 milliseconds (one second)
Connect=TCP -r1000 1313
```

注: 对于客户机应用程序 (例如看守程序), 在 **Com.Connect** 参数中使用 **-r** 选项来提供 RPC 读取超时 (也称为连接超时) 更方便。否则, 某些网络故障类型可能会导致无限期等待。

注:

[Com] 节、[Watchdog] 节和 [Hotstandby] 节中的 **Connect** 参数具有不同的用途。您务必要编辑正确的那个 **Connect** 参数。

- **SQLConnect** 函数 (**-r** 选项) 的连接字符串

例如:

```
SQLConnect (hdbc, "TCP -r1000 1313", SQL_NTS,
"dba", SQL_NTS, "dba", SQL_NTS);
```

在上述示例中, 常量 **SQL_NTS** 指示先前的字符串 (服务器名称、用户名或密码) 是作为以 **null** 结束的标准字符串传递的。

4.4.2 在 **Connect** 参数中指定 **-C** 选项

可以在 **solid.ini** 文件的 [Com] 节和 [Watchdog] 节中使用的 **Connect** 参数中指定连接超时 (也称为登录超时) 值。此连接超时仅适用于 TCP/IP 协议。

语法为:

```
Parameter = tcp -cnumber-of-milliseconds [machine name] port_number
```

其中 *Parameter* 为 **Connect** 或 **Listen**。

如果没有为连接超时提供任何值, 那么服务器将使用特定于操作系统的缺省值。

注:

对于客户机应用程序 (例如, 看守程序), 很方便在 **Connect** 参数中使用 **-c** 选项提供连接超时值。否则, 某些网络故障类型可能会导致等待很长时间之后才检测到故障。

例如:

应用程序节点:

```
[Com] ;The server listens to port 1320, and the Connection timeout is 1000 ms.
Listen = tcpip -c1000 1320
```

4.5 使用 HotStandby 配置 SMA

将 SMA 与透明连接 (TC) 配合使用时，主服务器与辅助服务器上的应用程序必须使用特定于 SMA 的 TC 连接信息语法来连接数据库。

关于此任务

通过 SMA TC，每个节点上的应用程序必须能够利用 SMA 连接来连接到本地服务器，并利用基于网络的连接来连接到远程服务器。

适用于使用 HotStandby 的 SMA 的 TC 连接目标列表的格式如下所示：

```
connect_target_list::=[SERVERS:]sma_connect_string, network_connect_string
```

其中，

```
sma_connect_string::= sma_protocol_name port_number | pipe_name
```

```
network_connect_string::= protocol_name IP_address | host_computer_name  
                           port_number | pipe_name
```

另外，还需要将负载均衡方法设置为 LOCAL_READ (PREFERRED_ACCESS=LOCAL_READ)。

要点：将 SMA 与 TC 配合使用时，如果将负载均衡方法设置为 READ_MOSTLY 或 WRITE_MOSTLY（缺省值），那么将使用网络连接而不是 SMA 连接。因此，将 SMA 与 TC 配合使用时，请始终将负载均衡方法设置为 LOCAL_READ。

过程

1. 设置两个 HotStandby 服务器。
2. 在这两个服务器上设置 SMA。
3. 对于这两个应用程序，请使用特定于 SMA 的连接目标列表语法和负载均衡属性 PREFERRED_ACCESS=LOCAL_READ 来定义 TC 连接。
4. 编译并启动应用程序。

示例

host1 上应用程序的连接信息，其中 solidDB 正在侦听端口 1964：
PREFERRED_ACCESS=LOCAL_READ SERVERS=sma tcp 1964, tcp host2 2315

host2 上应用程序的连接字符串，其中 solidDB 正在侦听端口 2315：
PREFERRED_ACCESS=LOCAL_READ SERVERS=sma tcp 2315, tcp host1 1964

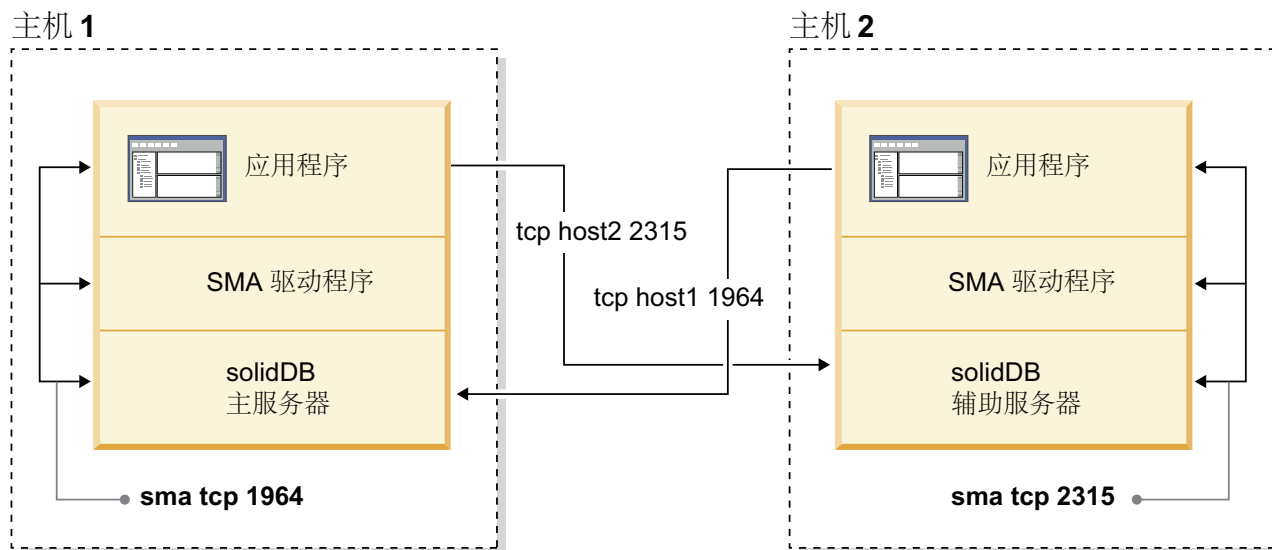


图 16. 示例: 使用 SMA 的 HotStandby 配置

4.6 利用 HotStandby 配置高级复制

通过 solidDB HotStandby 组件可以使高级复制系统的任何节点都具有高可用性。

当高级复制系统的主数据库和副本数据库正在使数据同步时，在数据库服务器对的主服务器之间也将进行同步。换句话说，主数据库的主服务器与副本数据库的主服务器进行通信。请参阅第 17 页的图 7。

数据库服务器在任何时候都可以故障转移到其辅助服务器，当数据库服务器与另一个使用高级复制的服务器同步数据时也是如此。如果在同步期间进行故障转移，那么将停止执行同步消息，必须在完成故障转移之后继续完成此进程。有关在发生错误之后如何进行同步的详细信息，请参阅 *IBM solidDB Advanced Replication User Guide*。

如果包含主数据库的服务器实现了与 solidDB HotStandby 之间容错，那么主数据库的副本必须知道用于两个主控服务器的连接字符串。为此，在每个副本数据库中执行以下语句：

```
SET SYNC CONNECT 'connect_string_to_server_1, connect_string_to_server_2'
TO MASTER master_nodename
```

在下图中，灰色箭头表示先前与原来的主服务器建立连接；而黑色箭头表示与新的主服务器（也就是旧的辅助服务器）建立的新连接。如果与旧的主服务器的同步失败，那么将使用备用连接。

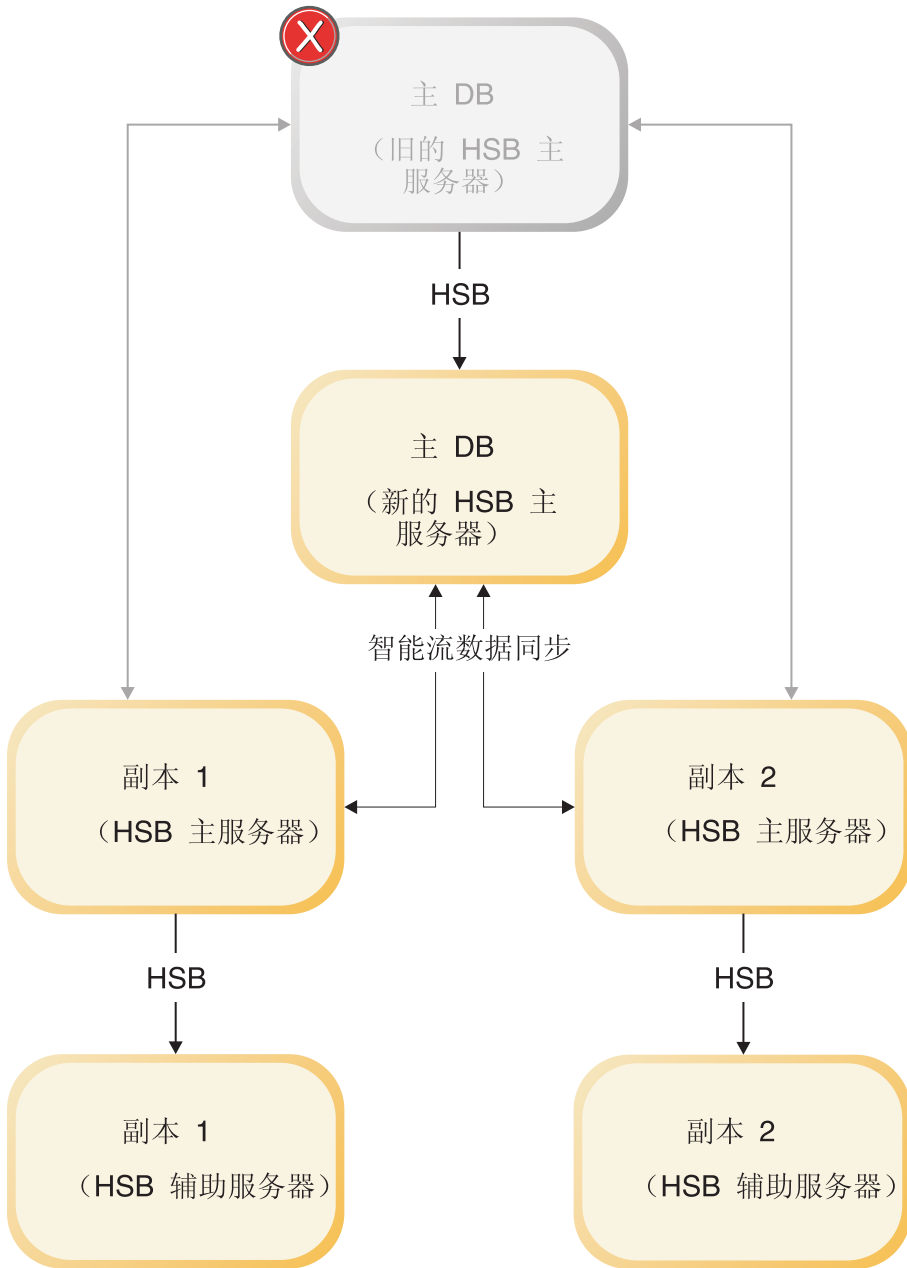


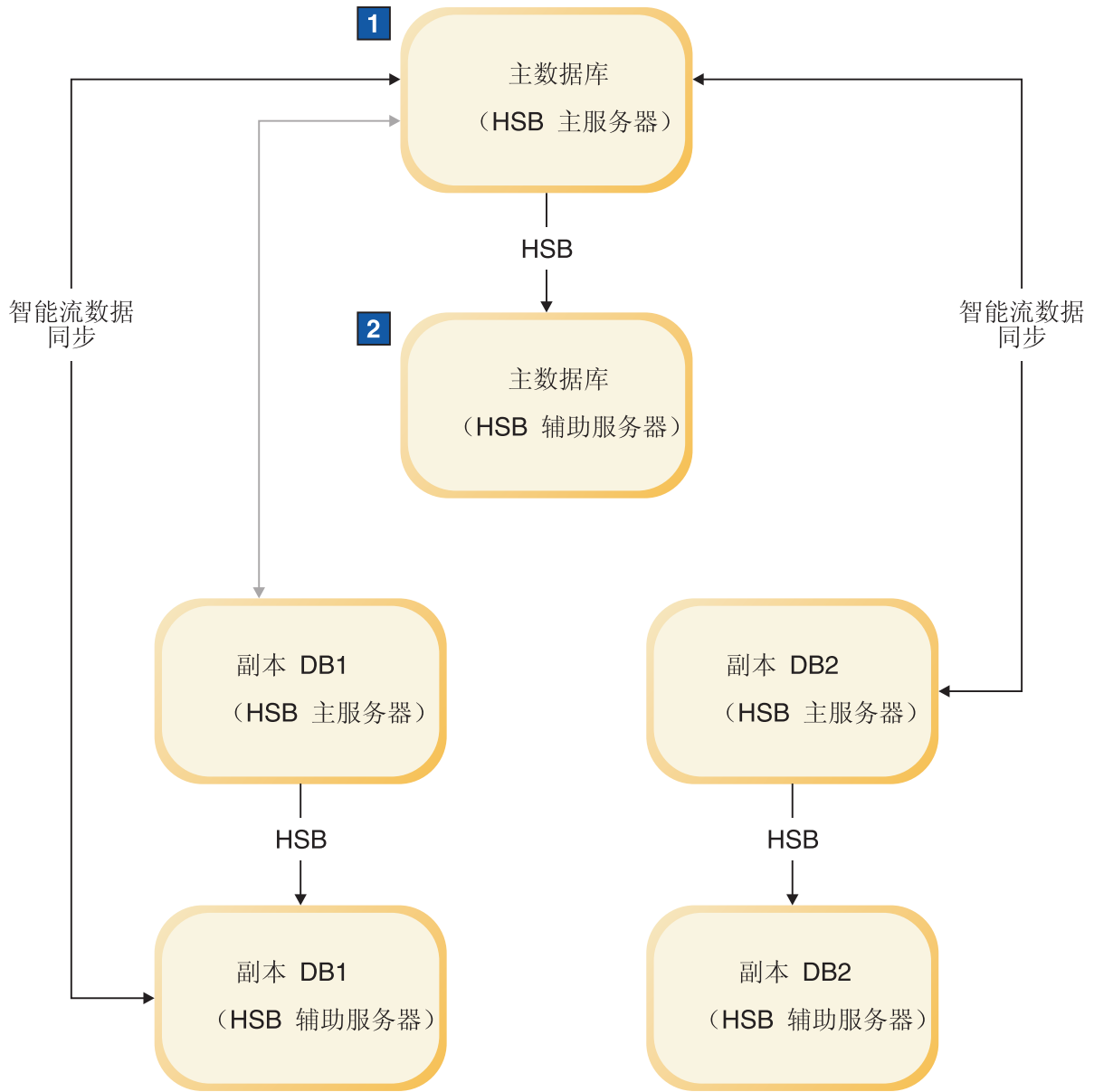
图 17. HotStandby 与高级复制: 主数据库的故障转移

如果使用 solidDB HotStandby 的服务器是一个包含副本数据库的服务器，并且如果主控服务器使用远程过程调用 (CALL *procedure_name* AT *node_name*) 以在副本服务器中运行过程（例如，启动同步），那么必须将两个副本服务器的连接字符串都告知主控服务器。通常，主服务器使用远程过程调用来启动与副本数据库同步。要将副本服务器对的连接字符串告知主服务器，请在主数据库中执行以下语句：

```
SET SYNC CONNECT 'connect_string_to_server_1,
connect_string_to_server_2' TO REPLICa replica_nodename
```

或者，可以将此语句保存在副本服务器中，然后在下一次进行同步时将它传播至主服务器。在这种情况下，请使用以下语句：

```
SAVE SET SYNC CONNECT 'connect_string_to_server_1,
connect_string_to_server_2' TO REPLICa replica_nodename
```

1. ReplicaDB 中的故障转移
2. SET SYNC CONNECT 'tcp machine4 1315' REPLICA TO replicaDB1

图 18. HotStandby 与高级复制: 副本数据库的故障转移

如果主控服务器从不在副本服务器中执行远程过程调用, 那么不需要上述语句。

5 处理高可用性控制器 (HAC) 时发生的故障

本节描述了使用高可用性控制器 (HAC) 时可能会发生的故障场景及其典型的恢复过程。HAC 会隐式处理各种故障场景。但是，不同的故障或初始化方案（简称管理方案）可由管理员或者看守程序类型的软件程序来处理。

HAC 是一种看守程序类型的程序，用于监视主服务器和辅助服务器，并在必要时发出命令以更改这些服务器的状态。例如，HAC 可以确定主服务器或辅助服务器本身发生故障的时间或者这些服务器之间的通信链路的断开时间。

进行恢复的目的是为了使失败组件重新恢复运行。有时，恢复期间可能又会发生其他故障。这些故障通常会导致系统处于一种可用性受限的状态（只有一个服务器已启动），从而期待人员干预。下面是不会自动处理的典型恢复时故障：

- 发生故障的数据库遭到破坏，从而无法重新启动此数据库。
- 没有足够的可用磁盘空间，无法执行同步复制。

5.1 主数据库失败

故障情况

节点 1 上的主数据库（处于 PRIMARY ACTIVE 状态）失败。

节点 2 上的辅助数据库（处于 SECONDARY ACTIVE 状态）与节点 1 上的主数据库建立连接时发生故障。

恢复

从主数据库故障进行恢复时，辅助服务器将替换主服务器。将按如下所示自动继续进行恢复：

1. 发生连接故障时，节点 2 上的辅助数据库将自动进入 SECONDARY ALONE 状态。
2. 节点 2 上的辅助数据库上的 HAC 实例断定节点 1 上的主数据库已失败，并将节点 2 上的辅助数据库设置为 PRIMARY ALONE 状态。
3. 在执行上述任务的同时，节点 1 上的主数据库上的 HAC 实例将重新启动主数据库，主数据库将进入 SECONDARY ALONE 状态。
4. 节点 2 上的辅助数据库上的 HAC 实例将启动连接主数据库与辅助数据库的过程。
5. 执行同步复制。

（可选）连接过程中包括一项 **netcopy** 操作，它会将节点 2 上的辅助数据库复制到节点 1 上的主数据库。

5.2 辅助数据库失败

故障情况

节点 2 上的辅助数据库（处于 SECONDARY ACTIVE 状态）失败。

节点 1 上的主数据库（处于 PRIMARY ACTIVE 状态）与节点 2 上的辅助数据库建立连接时发生故障。

恢复

从辅助数据库故障进行恢复时，辅助服务器将重新启动。将按如下所示自动继续进行恢复：

1. 发生连接故障时，节点 1 上的主数据库将自动进入 PRIMARY UNCERTAIN 状态；如果启用了 **AutoPrimaryAlone** 参数，那么此数据库将进入 PRIMARY ALONE 状态。
2. 节点 1 上的主数据库上的 HAC 实例断定节点 2 上的辅助数据库已失败。
3. 如果在步骤 1 中已将主数据库设置为 PRIMARY UNCERTAIN 状态，那么 HAC 现在会将它设置为 PRIMARY ALONE 状态。
4. 在执行上述任务的同时，节点 2 上的辅助数据库上的 HAC 实例将重新启动辅助数据库，辅助数据库将进入 SECONDARY ALONE 状态。
5. 节点 1 上的主数据库上的 HAC 实例将启动连接主数据库与辅助数据库的过程。
6. 执行同步复制。

（可选）连接过程中包括一项 **netcopy** 操作，它会将主数据库复制到辅助数据库。

5.3 主节点失败

故障情况

主节点（节点 1）失败。

节点 2 上的辅助数据库（处于 SECONDARY ACTIVE 状态）与节点 1 上的主数据库（处于 PRIMARY ACTIVE 状态）建立连接时发生故障。

恢复

从主节点故障进行恢复时，主服务器将重新启动。将按如下所示自动继续进行恢复：

1. 发生连接故障时，辅助数据库将自动进入 SECONDARY ALONE 状态。
2. 节点 2 上的辅助数据库上的 HAC 实例断定节点 1 上的主数据库已失败。
3. 节点 2 上的辅助数据库上的 HAC 实例将辅助数据库设置为 PRIMARY ALONE 状态。
4. 重新启动了主节点（节点 1）。
5. 重新启动了主数据库（节点 1）上的 HAC 实例。
6. 主数据库（节点 1）上的 HAC 实例断定主数据库未运行。
7. 主数据库（节点 1）上的 HAC 实例将重新启动主数据库并将它设置为 SECONDARY ALONE 状态。
8. 节点 2 上的辅助数据库上的 HAC 实例将启动连接主数据库与辅助数据库的过程。
9. 执行同步复制。

（可选）连接过程中包括一项 **netcopy** 操作，它会将主数据库复制到辅助数据库。

5.4 辅助节点失败

故障情况

辅助节点（节点 2）失败。

节点 1 上的主数据库（处于 PRIMARY ACTIVE 状态）与节点 2 上的辅助数据库（处于 SECONDARY ACTIVE 状态）建立连接时发生故障。

恢复

从辅助节点故障进行恢复时，辅助服务器将重新启动。将按如下所示自动继续进行恢复：

1. 发生连接故障时，节点 1 上的主数据库将自动进入 PRIMARY UNCERTAIN 状态。
2. 节点 1 上的主数据库上的 HAC 实例断定节点 2 上的辅助数据库已失败。
3. 节点 1 上的主数据库上的 HAC 实例将主数据库设置为 PRIMARY ALONE 状态。
4. 重新启动了辅助节点（节点 2）。
5. 重新启动了辅助数据库（节点 2）上的 HAC 实例。
6. 辅助数据库（节点 2）上的 HAC 实例断定辅助数据库未运行。
7. 辅助数据库（节点 2）上的 HAC 实例将重新启动辅助数据库并将它设置为 SECONDARY ALONE 状态。
8. 节点 1 上的主数据库上的 HAC 实例将启动连接主数据库与辅助数据库的过程。
9. 执行同步复制。

（可选）连接过程中包括一项 **netcopy** 操作，它会从主数据库复制到新的辅助数据库。

5.5 HotStandby 链路失败

故障情况

HotStandby 链路失败。

节点 1 上的主数据库（处于 PRIMARY ACTIVE 状态）和节点 2 上的辅助数据库（处于 SECONDARY ACTIVE 状态）互相建立连接时都发生了故障。

恢复

从 HotStandby 链路故障进行恢复时，HAC 实例将对“外部引用实体 (ERE)”执行 ping 操作，以便弄清楚是发生了网络故障还是相对应的服务器已失败。将按如下所示自动继续进行恢复：

1. 发生连接故障时，主数据库和辅助数据库将分别自动进入 PRIMARY UNCERTAIN（节点 1）和 SECONDARY ALONE（节点 2）状态。
2. 两个 HAC 实例直接连接至远程服务器都失败。
3. 两个 HAC 实例都使用操作系统的 ping 实用程序对 ERE 执行 ping 操作。
4. 如果 ping 操作失败，那么本地服务器将保持或者设置为 SECONDARY ALONE 状态。

5. 如果 ping 操作成功，那么成功的 HAC 将尝试连接至远程数据库服务器。
6. 如果连接至远程数据库服务器失败，那么 HAC 将断定它的网络连接是可操作的，并将本地服务器设置为 PRIMARY ALONE 状态。
7. 主数据库上的 HAC 实例将尝试重新与辅助数据库建立连接。
8. 一旦网络变得可操作并且连接成功，那么主数据库和辅助数据库将分别自动进入 PRIMARY ACTIVE 和 SECONDARY ACTIVE 状态。

5.6 服务器对外部客户机无响应

故障情况

与服务器的连接失败或永久挂起。服务器分别处于 PRIMARY ACTIVE 和 SECONDARY ACTIVE 状态，但客户机无法连接这些服务器，也无法执行事务。

恢复

无论无响应服务器的状态为何，HAC 都会执行 `solidhac.ini` 配置文件中利用 **LocalDB.UnresponsiveActionScript** 参数进行配置的脚本。该脚本以单个参数开始，包括无响应 solidDB 进程的进程标识 (pid)。

典型的解决方案就是终止根据进程标识来识别的进程。在此类情况下，恢复操作将自动执行，如下所示：

如果无响应的服务器处于主数据库角色：

1. 连接失败时，辅助数据库（节点 2）会自动过渡到 SECONDARY ALONE 状态。
2. 辅助数据库（节点 2）上的 HAC 实例会注意到主数据库进程（节点 1）已终止，并将辅助数据库（节点 2）设置成 PRIMARY ALONE 状态。
3. 在执行上述任务的同时，主数据库（节点 1）上的 HAC 实例会重新启动旧的主数据库，这个旧的主数据库将进入 SECONDARY ALONE 状态。
4. 辅助数据库（节点 2）上的 HAC 实例会启动用于连接主数据库和辅助数据库的进程。
5. 执行同步复制。

（可选）连接过程中包括一项从辅助数据库（节点 2）到主数据库（节点 1）的 **netcopy** 操作。

如果无响应的服务器处于辅助数据库角色：

1. 连接失败时，主数据库（节点 1）将自动过渡到 PRIMARY UNCERTAIN 状态；如果启用了 `AutoPrimaryAlone` 参数，那么主数据库将自动过渡到 PRIMARY ALONE 状态。
2. 主数据库（节点 1）上的 HAC 实例会注意到辅助数据库进程（节点 2）已终止。
3. 如果在步骤 1 中已将主数据库设置为 PRIMARY UNCERTAIN 状态，那么 HAC 现在会将它设置为 PRIMARY ALONE 状态。
4. 在执行上述任务的同时，辅助数据库（节点 2）上的 HAC 实例会重新启动旧的辅助数据库，这个旧的辅助数据库将进入 SECONDARY ALONE 状态。
5. 主数据库（节点 1）上的 HAC 实例会启动用于连接主数据库和辅助数据库的进程。
6. 执行同步复制。

(可选) 连接过程中包括一项 **netcopy** 操作, 它会将主数据库复制到辅助数据库。

6 升级（迁移）HotStandby 服务器

迁移过程涉及到更新软件版本。对于一个高可用系统（例如，solidDB HotStandby），可以执行冷迁移或热迁移。

冷迁移表示先关闭整个系统（两个服务器），再升级系统，然后使用新的软件和配置数据重新启动。

热迁移表示在升级服务器所需的时间内，无需使整个系统脱机，就升级 HotStandby 服务器对。一个服务器可以在另一个服务器进行升级期间继续运行。

要点：使用热迁移时，尽管不会关闭整个系统，但是，如果您使用的是基本连接，那么用户或应用程序可能必须与一个服务器断开连接，然后连接至另一个服务器。如果您使用的是透明连接，那么可以采用对用户和应用程序透明的方式执行热迁移。

6.1 冷迁移过程

关于此任务

迁移步骤如下：

过程

1. 将服务器断开连接并关闭它们。
2. 安装新的软件版本。
3. 如果要升级到新版本，请将新的许可证文件 (solid.lic) 从“许可证证书”映像中复制到主服务器和辅助服务器。
4. 必要时，更新 solid.ini 文件。
5. 使用命令行参数 **-x autoconvert** 启动主服务器，此参数将指示服务器将现有数据库转换为新格式。
6. 将主服务器设置为 PRIMARY ALONE 状态。
7. 执行从主服务器到辅助服务器的“hsb netcopy”。
8. 连接服务器。

6.2 热迁移过程

从 版本级别 7.0 IF2/7.0 FP3 的 solidDB 服务器开始，支持此热迁移过程。对于 FP3 之前的版本，请使用热迁移过程（使用 netcopy）。

开始之前

1. 如果应用程序尚未设计为自动进行故障转移，请通知用户这些应用程序将断开它们的连接，并将需要重新连接至新的主服务器。
2. 准备系统和软件以进行升级：
 - a. 由于每个 solidDB 服务器在部分升级操作期间将单独运行（具体来说，就是在 PRIMARY ALONE 状态下运行），因此，请确保这两台计算机都处于健康状态。

例如，这两台计算机都应该具有足够的可用磁盘空间、可靠的网络连接以及在发生电源故障时可以使用的 UPS 设备。

- b. 每个服务器至少有一小段时间（当另一个服务器进行升级时）将在 PRIMARY ALONE 状态下运行。当服务器处于 PRIMARY ALONE 状态时，它会将事务存储在事务日志中。必须具有足够的磁盘空间可供日志文件用来存储在另一个服务器进行升级期间（其中包括另一个服务器在重新启动之后进行同步复制时所需的时间）将产生的所有事务。
- c. 确保每台计算机上都具有新的软件版本的副本。

注： solidDB 以单个安装文件的形式提供。如果您不想在生产环境节点上运行安装程序，可以在独立节点上安装 solidDB，再将适用于您的安装的可执行文件、库和驱动程序手动复制到生产节点。

提示： 根据您是要升级到新的版本级别（例如，从 6.5 升级到 7.0）还是要应用修订包（例如，7.0 FP2 至 7.0 FP3），升级过程有所不同。过程中描述了这些差别。

3. 如果您使用了看守程序，请将其暂时关闭，以便升级过程中，它不会发出与您所发出的命令相冲突的命令。例如，在使主服务器与辅助服务器断开连接之后，在对辅助服务器进行升级之前，您不希望看守程序尝试重新连接这两个服务器。

关于此任务

在下列步骤中，S1（“OP”）和 S2（“OS”）表示原来的主服务器和辅助服务器。当您完成此过程时，每个服务器的状态都会更改。

过程

1. S1: 使主服务器与辅助服务器断开连接:
ADMIN COMMAND 'hsb disconnect';
2. S2 OS: 关闭服务器 S2（辅助服务器）:
ADMIN COMMAND 'shutdown force';
3. S1 OP: 如果服务器 S1（主服务器）尚未自动切换到 PRIMARY ALONE 状态，请指示该服务器在此状态下运行。验证服务器 S1（主服务器）是否处于 PRIMARY ALONE 状态。
ADMIN COMMAND 'hsb set primary alone';
ADMIN COMMAND 'hsb state';
4. S2 OS: 将服务器 S2（原始辅助服务器）升级到新的 solidDB 版本级别。

注： solidDB 以单个安装文件的形式提供。如果您不想在生产环境节点上运行安装程序，可以在独立节点上安装 solidDB，再将适用于您的安装的可执行文件、库、驱动程序和许可证文件手动复制到生产节点。必要时，还请更新 solid.ini 文件。

5. S2 OS: 启动服务器 S2（原始辅助服务器）。
 - 如果要升级到新的版本级别（例如，从 6.5 升级到 7.0），请使用 **-x autoconvert** 命令行选项启动 solidDB 服务器。

```
solid -x autoconvert
```
 - **-x autoconvert** 选项指示服务器接受现有数据库并将其转换为新的版本级别。
 - 如果要应用新的修订包级别（例如，7.0 FP2 至 7.0 FP3），那么可以正常启动 solidDB 服务器。不需要转换数据库。

服务器 S2 将在 SECONDARY ALONE 状态下启动。

6. S1 OP: 将主服务器连接至辅助服务器:

```
ADMIN COMMAND 'hsb connect';
```

注: 如果辅助服务器正在运行更高版本的服务器, 那么不能在辅助服务器上发出 **hsb connect** 命令。

此步骤会启动一个进程, 辅助服务器通过该进程对辅助服务器关闭期间发生的数据更改进行同步复制。

7. S1 OP: 等到同步复制完成之后才继续执行后续操作。
8. S2: 在服务器之间已建立连接并进行同步复制之后, 请执行角色切换以准备关闭当前主服务器 (S1 OP):

```
ADMIN COMMAND ' hsb switch primary';
```

9. S2: 使主服务器与辅助服务器断开连接:

```
ADMIN COMMAND 'hsb disconnect';
```

10. S1 OP: 关闭服务器 S1 OP (现在处于辅助方式):

```
ADMIN COMMAND 'shutdown force';
```

要点: **force** 选项会异常终止所有已打开的事务。

11. S2 OS: 如果新的主服务器 S2 (旧的辅助服务器) 尚未自动切换到 PRIMARY ALONE 状态, 请将该服务器设置为在此状态下运行。验证服务器 S2 是否处于 PRIMARY ALONE 状态。

```
ADMIN COMMAND 'hsb set primary alone';  
ADMIN COMMAND 'hsb state';
```

12. S1 OP: 将服务器 S1 (原始主服务器) 升级到新的 solidDB 版本级别。

注: solidDB 以单个安装文件的形式提供。如果您不想在生产环境节点上运行安装程序, 可以在独立节点上安装 solidDB, 再将适用于您的安装的可执行文件、库、驱动程序和许可证文件手动复制到生产节点。必要时, 还请更新 solid.ini 文件。

13. S1 OP: 启动服务器 S1。

- 如果要升级到新的版本级别 (例如, 从 6.5 升级到 7.0), 请使用 **-x autoconvert** 命令行选项启动 solidDB 服务器。

```
solid -x autoconvert
```

-x autoconvert 选项指示服务器接受现有数据库并将其转换为新的版本级别。

- 如果要应用新的修订包级别 (例如, 7.0 FP2 至 7.0 FP3), 那么可以正常启动 solidDB 服务器。不需要转换数据库。

服务器 S1 OP 将在 SECONDARY ALONE 状态下启动。

14. S2 OS: 连接服务器:

```
ADMIN COMMAND 'hsb connect';
```

hsb connect 命令将新的主服务器连接至新的辅助服务器并启动一个进程, 新的辅助服务器通过该进程对它关闭期间发生的数据更改进行同步复制。

如果此步骤失败, 请使用 **hsb netcopy** 命令将整个数据库复制到辅助服务器, 然后从步骤 12 重新开始。

结果

在新的辅助服务器与新的主服务器之间执行同步复制之后，系统应该正常运行。新的主服务器和新的辅助服务器都已升级，并且都具有最新数据。

下一步做什么

在升级之后，您可能要运行一些测试查询，以确保一切正常。

1. 测试主服务器和辅助服务器是否都在正确运行。例如，您可以选择下面这一系列操作：

在主服务器上：

```
ADMIN COMMAND 'hsb state';  
ADMIN COMMAND 'hsb status catchup';
```

发出任何类型的只读查询。

在辅助服务器上：

```
ADMIN COMMAND 'hsb state';  
ADMIN COMMAND 'hsb status catchup';
```

发出任何类型的只读查询。

2. 如果您使用了看守程序，请将它重新启动。

相关任务：

6.3, 『热迁移过程（使用 netcopy）』

将 solidDB 服务器升级到版本级别 7.0 FP2 时，必须使用热迁移过程（使用 netcopy）。

6.3 热迁移过程（使用 netcopy）

将 solidDB 服务器升级到版本级别 7.0 FP2 时，必须使用热迁移过程（使用 netcopy）。

开始之前

1. 如果应用程序尚未设计为自动进行故障转移，请通知用户这些应用程序将断开它们的连接，并将需要重新连接至新的主服务器。
2. 准备系统和软件以进行升级：
 - a. 由于每个 solidDB 服务器在部分升级操作期间将单独运行（具体来说，就是在 PRIMARY ALONE 状态下运行），因此，请确保这两台计算机都处于健康状态。例如，这两台计算机都应该具有足够的可用磁盘空间、可靠的网络连接以及在发生电源故障时可以使用的 UPS 设备。
 - b. 每个服务器至少有一小段时间（当另一个服务器进行升级时）将在 PRIMARY ALONE 状态下运行。当服务器处于 PRIMARY ALONE 状态时，它会将事务存储在事务日志中。必须具有足够的磁盘空间可供日志文件用来存储在另一个服务器进行升级期间（其中包括另一个服务器在重新启动之后进行同步复制时所需的时间）将产生的所有事务。
 - c. 确保每台计算机上都具有新的软件版本的副本。

注: solidDB 以单个安装文件的形式提供。如果您不想在生产环境节点上运行安装程序,可以在独立节点上安装 solidDB,再将适用于您的安装的可执行文件、库和驱动程序手动复制到生产节点。

提示: 根据您是要升级到新的版本级别(例如,从 6.5 升级到 7.0)还是要应用修订包(例如,7.0 FP2 至 7.0 FP3),升级过程有所不同。过程中描述了这些差别。

3. 如果您使用了看守程序,请将其暂时关闭,以便升级过程中,它不会发出与您所发出的命令相冲突的命令。例如,在使主服务器与辅助服务器断开连接之后,在对辅助服务器进行升级之前,您不希望看守程序尝试重新连接这两个服务器。

关于此任务

在下列步骤中,S1(“OP”)和 S2(“OS”)表示原来的主服务器和辅助服务器。当您完成此过程时,每个服务器的状态都会更改。

过程

1. S1: 使主服务器与辅助服务器断开连接:

```
ADMIN COMMAND 'hsb disconnect';
```
2. S2 OS: 关闭服务器 S2(辅助服务器):

```
ADMIN COMMAND 'shutdown force';
```
3. S1 OP: 如果服务器 S1(主服务器)尚未自动切换到 PRIMARY ALONE 状态,请指示该服务器在此状态下运行。验证服务器 S1(主服务器)是否处于 PRIMARY ALONE 状态。

```
ADMIN COMMAND 'hsb set primary alone';  
ADMIN COMMAND 'hsb state';
```
4. S2 OS: 将服务器 S2(原始辅助服务器)升级到新的 solidDB 版本级别。

注: solidDB 以单个安装文件的形式提供。如果您不想在生产环境节点上运行安装程序,可以在独立节点上安装 solidDB,再将适用于您的安装的可执行文件、库、驱动程序和许可证文件手动复制到生产节点。必要时,还请更新 solid.ini 文件。

5. S2 OS: 启动服务器 S2(原始辅助服务器)。
 - 如果要升级到新的版本级别(例如,从 6.5 升级到 7.0),请使用 **-x autoconvert** 命令行选项启动 solidDB 服务器。

```
solid -x autoconvert
```
 - **-x autoconvert** 选项指示服务器接受现有数据库并将其转换为新的版本级别。
 - 如果要应用新的修订包级别(例如,7.0 FP2 至 7.0 FP3),那么可以正常启动 solidDB 服务器。不需要转换数据库。

服务器 S2 将在 SECONDARY ALONE 状态下启动。

6. S1 OP: 使用 netcopy 命令将数据库从旧的主服务器(S1)复制到新的辅助服务器(S2):

```
ADMIN COMMAND 'hsb netcopy';
```
7. S1 OP: 验证 netcopy 命令是否已成功:

```
ADMIN COMMAND 'hsb status copy';
```
8. S1 OP: 将主服务器连接至辅助服务器:

ADMIN COMMAND 'hsb connect';

注: 如果辅助服务器正在运行更高版本的服务器, 那么不能在辅助服务器上发出 **hsb connect** 命令。

此步骤会启动一个进程, 辅助服务器通过该进程对辅助服务器关闭期间发生的数据更改进行同步复制。

9. S1 OP: 等到同步复制完成之后才继续执行后续操作。
10. S2: 在服务器之间已建立连接并进行同步复制之后, 请执行角色切换以准备关闭当前主服务器 (S1 OP):

ADMIN COMMAND ' hsb switch primary';

11. S2: 使主服务器与辅助服务器断开连接:

ADMIN COMMAND 'hsb disconnect';

12. S1 OP: 关闭服务器 S1 OP (现在处于辅助方式):

ADMIN COMMAND 'shutdown force';

要点: **force** 选项会异常终止所有已打开的事务。

13. S2 OS: 如果新的主服务器 S2 (旧的辅助服务器) 尚未自动切换到 PRIMARY ALONE 状态, 请将该服务器设置为在此状态下运行。验证服务器 S2 是否处于 PRIMARY ALONE 状态。

ADMIN COMMAND 'hsb set primary alone';
ADMIN COMMAND 'hsb state';

14. S1 OP: 将服务器 S1 (原始主服务器) 升级到新的 solidDB 版本级别。

注: solidDB 以单个安装文件的形式提供。如果您不想在生产环境节点上运行安装程序, 可以在独立节点上安装 solidDB, 再将适用于您的安装的可执行文件、库、驱动程序和许可证文件手动复制到生产节点。必要时, 还请更新 solid.ini 文件。

15. S1 OP: 启动服务器 S1。
 - 如果要升级到新的版本级别 (例如, 从 6.5 升级到 7.0), 请使用 **-x autoconvert** 命令行选项启动 solidDB 服务器。

solid -x autoconvert

-x autoconvert 选项指示服务器接受现有数据库并将其转换为新的版本级别。

- 如果要应用新的修订包级别 (例如, 7.0 FP2 至 7.0 FP3), 那么可以正常启动 solidDB 服务器。不需要转换数据库。

服务器 S1 OP 将在 SECONDARY ALONE 状态下启动。

16. S2 OS: 通过执行 Netcopy 操作将数据库从新的主服务器 (S2) 复制到新的辅助服务器 (S1) 中:

ADMIN COMMAND 'hsb netcopy';

17. S2 OS: 验证 netcopy 命令是否已成功:

ADMIN COMMAND 'hsb status copy';

18. S2 OS: 连接服务器:

ADMIN COMMAND 'hsb connect';

hsb connect 命令将新的主服务器连接至新的辅助服务器并启动一个进程, 新的辅助服务器通过该进程对它关闭期间发生的数据更改进行同步复制。

如果此步骤失败，请使用 **hsb netcopy** 命令将整个数据库复制到辅助服务器，然后从步骤 14 重新开始。

结果

在新的辅助服务器与新的主服务器之间执行同步复制之后，系统应该正常运行。新的主服务器和新的辅助服务器都已升级，并且都具有最新数据。

下一步做什么

在升级之后，您可能要运行一些测试查询，以确保一切正常。

1. 测试主服务器和辅助服务器是否都在正确运行。例如，您可以选择下面这一系列操作：

在主服务器上：

```
ADMIN COMMAND 'hsb state';  
ADMIN COMMAND 'hsb status catchup';
```

发出任何类型的只读查询。

在辅助服务器上：

```
ADMIN COMMAND 'hsb state';  
ADMIN COMMAND 'hsb status catchup';
```

发出任何类型的只读查询。

2. 如果您使用了看守程序，请将它重新启动。

附录 A. HotStandby 配置参数

本节探讨了可用于 HotStandby 的配置参数。

有四种类型的参数:

- 服务器端 `solid.ini` 配置文件中的服务器端参数
 - 服务器端 `solid.ini` 配置文件中有两个特定于 HotStandby 的节: `[Cluster]` 和 `[HotStandby]`。
 - 许多其他参数也可以影响 HotStandby 功能。
- 客户端 `solid.ini` 配置文件中的客户端参数
- `solidhac.ini` 配置文件中的高可用性控制器 (HAC) 配置参数。
- `solidhac.ini` 配置文件中的高可用性管理器 (HAM) 配置参数。

提示: 如果使用 `solidDB` 软件包中所提供的看守程序样本, 那么 `solid.ini` 还将在 `[Watchdog]` 节中包含特定于看守程序的参数。有关信息, 请参阅第 165 页的附录 F, 『看守程序样本』一节。

访问方式

参数的访问方式会定义是否可以动态更改此参数以及更改何时生效。当使用 `ADMIN COMMAND` 更改了参数的值时, 可能不会立即应用更改, 并且当服务器下一次启动时也可能不应用该更改。如果将参数值写入 `solid.ini` 文件, 那么它将在服务器下次启动后生效。

访问方式值

访问方式可能是:

- *RO* (只读): 不能更改值; 当前值始终与启动值完全相同。
- *RW*: 可以通过 `ADMIN COMMAND` 来更改值, 并且更改将立即生效。
- *RW/Startup*: 可以通过 `ADMIN COMMAND` 来更改值, 并且更改将在服务器下一次启动时生效。
- *RW/Create*: 可以通过 `ADMIN COMMAND` 来更改值, 并且将在创建了新数据库时应用更改。

相关概念:

第 37 页的 3.2, 『配置 HotStandby』

通过使用主节点和辅助节点处的 `solid.ini` 配置文件来配置

HotStandby。[HotStandby] 一节包含特定于 HotStandby 的所有配置参数。还必须设置其他节和参数（例如, `Com.Listen` 参数）。

A.1 服务器端参数

A.1.1 [Cluster] 节

表 21. 集群参数

[Cluster]	描述	出厂值	访问方式
ReadMostlyLoadPercentAtPrimary	定义当负载均衡设置为 READ_MOSTLY 时定向到主服务器的读负载的百分比。	50	RW/Startup

A.1.2 [HotStandby] 节

表 22. HotStandby 参数

HotStandby	描述	出厂值	访问方式
1SafeMaxDelay	在 1-Safe 复制中, 此参数是指将已落实的事务发送至辅助服务器之前的最大延迟时间（按毫秒计）。	5000	RW
2SafeAckPolicy	此参数指定辅助服务器从主服务器中接收到事务时进行确认的计时。 有效值为: <ul style="list-style-type: none">• 1 = 2-safe received。辅助服务器接收到数据时就进行确认。• 2 = 2-safe visible。辅助服务器将在数据“可视”时（即, 当辅助服务器执行了事务时）进行确认。• 3 = 2-safe durable。辅助服务器将在它已使数据成为持久数据之后（即, 在它已落实数据并且将数据写入磁盘之后）进行确认。 2-safe durable 是最安全的方法, 并且 2-safe received 的响应时间最短。但是, 2-safe received 方式实际上在大多数情况下都足以保证数据安全, 因此, 在数据安全与响应速度之间达到了最佳平衡。 仅当服务器正在使用 2-safe 复制时, 此参数才适用。 注: 尽管此参数可控制辅助服务器的行为, 但此参数是在主服务器设置的。将忽略在辅助服务器的 <code>solid.ini</code> 文件中的值。	1	RW
AutoPrimaryAlone	如果此参数设置为 yes, 那么在与辅助服务器的连接中断后, 服务器将自动进入 PRIMARY ALONE 状态（而不是 PRIMARY UNCERTAIN 状态）。	no	RW

表 22. HotStandby 参数 (续)

HotStandby	描述	出厂值	访问方式
CatchupSpeedRate	<p>当服务器正在执行同步复制时，它还会继续处理来自客户机的数据库请求。可以通过 CatchupSpeedRate 参数来提高“对应用程序请求作出响应”的重要性，同时降低同步复制的优先级；或者降低“对应用程序请求作出响应”的重要性，同时提高同步复制的优先级。</p> <p>使用由链路和辅助服务器吞吐量指定的最大可用速度所占的百分比来表示速率。此数目越大，表示更注重同步复制，处理客户机请求的重要性就会降低。有效值为 1 到 99。</p>	50	RW
Connect	<p>Connect 参数指示服务器对中另一个 HotStandby 服务器所在的地址。</p> <p>此参数的值是一个标准的 solidDB 连接字符串（基本连接）或特定于 TC 的连接字符串（透明连接）。</p> <p>使用此参数定义的连接字符串必须与使用 Com.Listen 参数定义的另一个 HotStandby 服务器的服务器侦听名称相匹配。</p> <p>如果您在打算用于 HotStandby 的服务器中省略此参数，那么可以使用 ADMIN COMMAND 来动态设置此参数。在服务器具有连接字符串之前，该服务器只能处于不涉及到 HotStandby 连接的状态（即，PRIMARY ALONE、SECONDARY ALONE 和 STANDALONE 状态）。</p> <p>如果 HSBEnabled 参数设置为 no，那么将忽略此参数。</p> <p>对于使用多主目录服务器的透明连接 (TC) 连接，可以将 Connect 参数覆盖为 TCConnect 参数。</p>	无出厂值。	RW
ConnectTimeout	<p>通过指定连接超时值，可以设置 HotStandby 连接操作等待与远程机器建立连接的最长时间（按秒计）。</p> <p>ConnectTimeout 参数与以下管理命令配合使用：</p> <ul style="list-style-type: none"> • hotstandby connect • hotstandby switch primary • hotstandby switch secondary <p>例如，要将超时值设置为 30 秒（30000 毫秒）：</p> <pre>[HotStandby] ConnectTimeout=30000</pre> <p>另请参阅 PingTimeout。</p>	<p>0（无超时）</p> <p>单位：1 毫秒</p>	RW

表 22. HotStandby 参数 (续)

HotStandby	描述	出厂值	访问方式
CopyDirectory	<p>[HotStandby] 节中的 CopyDirectory 参数定义当用户执行以下命令时所执行的 HotStandby 复制操作的名称和位置:</p> <pre>ADMIN COMMAND 'hotstandby copy';</pre> <p>例如, 此参数看起来可能为如下所示:</p> <pre>[HotStandby] CopyDirectory=C:\solidDB\secondary\dbfiles</pre> <p>如果您为 CopyDirectory 参数提供一个相对路径, 那么此路径是相对于用来保存主服务器的 solid.ini 文件的目录的路径。</p> <p>此参数没有出厂值, 因此, 如果在 solid.ini 文件中未指定目录, 那么必须在 copy 命令中提供目录。</p> <p>因为 ADMIN COMMAND 'hotstandby netcopy' 是较为灵活的解决方案, 所以它是建议的数据库复制方法。</p>	无出厂值	RW
HSBEnabled	<p>如果此参数设置为 yes, 那么服务器可以充当 HotStandby 主服务器或辅助服务器。如果此参数设置为 no, 那么服务器不能充当 HotStandby 服务器。</p> <p>如果将此参数设置为 yes, 那么当服务器首次启动时, 会将服务器的缺省初始状态隐式定义为 SECONDARY ALONE。有效值为 yes 和 no。</p> <p>要使用 HotStandby, 还必须通过在 solid.ini 文件中设置 Connect 参数或者使用 ADMIN COMMAND 来设置此参数的方式来指定此参数。</p>	no	RO
MaxLogSize	定义基于磁盘的 HSB 日志的最大大小。出厂值: 无限制	0 单位: 1 字节 k=KB m=MB	
MaxMemLogSize	当禁用基于文件的记录 (Logging.LogEnabled=no) 时, 这是在将内存日志发送至辅助服务器之前, 用来存放事务的内存日志大小。此值将影响在内存日志变满之前服务器可能保持处于 PRIMARY ALONE 状态的时间。	8M 单位: 1 字节 k=KB m=MB	RO
NetcopyReceiveBufferSize	定义 netcopy 期间辅助服务器中用于存储数据的缓冲区大小。如果缓冲区已满, 那么网络复制数据的写操作会调速。	32M 单位: 1 字节 k=KB m=MB	RW
NetcopyRpcTimeout	netcopy 操作的数据传输确认超时 (按毫秒计)	30000 单位: 1 毫秒	RW
PingInterval	<p>主服务器和辅助服务器会定期互相发送“ping”消息以确保它们仍然互相保持连接。(这些 ping 消息与主服务器发送至辅助服务器的事务信息无关。)</p> <p>此值等于服务器发送的两个连续 ping 操作之间的时间间隔 (按毫秒计)。</p>	1000 (1 秒) 单位: 1 毫秒	RW

表 22. HotStandby 参数 (续)

HotStandby	描述	出厂值	访问方式
PingTimeout	<p>此参数指定一个服务器在断定另一个服务器已关闭或者不可访问之前应等待的时间。</p> <p>经过所指定的时间（按毫秒计）之后，服务器将断定连接已断开并且将相应地更改状态。</p> <p>另请参阅 ConnectTimeout。</p>	<p>4000（4 秒）</p> <p>单位：1 毫秒</p>	RW
PrimaryAlone	不推荐使用此参数。请使用 AutoPrimaryAlone 参数。	no	RW
SafenessLevel	<p>此参数用于设置复制协议的安全级别。</p> <p>可以使用的值为：1safe、2safe 和 auto</p> <p>通过使用 auto 值，可以允许根据耐久性级别来动态更改安全级别。如果 SafenessLevel 参数设置为 auto，并使用 SET DURABILITY 命令或 DurabilityLevel 参数将耐久性设置为“宽松”，那么安全级别将设置为 1-safe；当您将耐久性级别设置为“严格”时，安全级别将设置为 2-safe。但是，如果 DurabilityLevel 参数设置为 2（自适应耐久性），那么 auto 设置将不起作用；安全级别始终将为 2-safe。</p>	2-safe	RW
SecondaryThreads	<p>此参数用于定义辅助服务器处理写操作时所使用的线程数。</p> <p>最优的线程数视环境而定。从原则上来说，有效值为 1 到 256。</p>	4	RW/Startup
TCConnect	<p>如果应用程序和服务器需要使用不同的网络来互相连接（例如，当使用多主目录服务器时），此参数用于定义透明连接（TC）连接的服务器对中另一个 HotStandby 服务器的地址。</p> <p>从应用程序连接透视图，使用此参数指定的地址优先于使用 HotStandby.Connect 参数定义的地址。因此，TC 连接将使用通过此参数指定的服务器地址，而服务器之间的 HotStandby 连接将使用通过 HotStandby.Connect 参数定义的服务器地址。</p>	无出厂值。	RW

A.2 客户机端参数

A.2.1 Com 节

客户机端 solid.ini 文件中的 [Com] 节包含可与 HotStandby 配合使用的 **Com.Connect** 参数。

表 23. 客户机端通信参数

[Com]	描述	出厂值
ClientReadTimeout	<p>此参数定义连接超时，即读超时（以毫秒计）。如果在指定的时间内未接收到任何响应，那么网络请求将失败。值 0 将超时设置为无限。此值可覆盖为连接字符串选项 -r 甚至 ODBC 属性 SQL_ATTR_CONNECTION_TIMEOUT。</p> <p>注：此参数仅适用于 TCP 协议。</p>	0（无限）

表 23. 客户端通信参数 (续)

[Com]	描述	出厂值
Connect	<p>如果未在连接参数中显式指定连接字符串, 那么 Connect 参数会定义客户端用于连接 solidDB 服务器的缺省网络名 (连接字符串)。当发出带有空数据源名称的 SQLConnect() 调用时, 也使用此值。</p> <p>标准 solidDB 连接字符串的格式是:</p> <pre>protocol_name [options] [host_computer_name] server_name</pre> <p>其中, options 和 server_name 取决于通信协议。 要点: 在 HotStandby 和 SMA 设置中, 会使用其他连接字符串属性来指定更多功能, 例如透明连接 (TC)。</p> <p>有关更多详细信息, 请参阅网络名和连接字符串语法。</p>	<p>tcp localhost 1964 (Windows)</p> <p>upipe SOLID (Linux 和 UNIX)</p>
ConnectTimeout	<p>ConnectTimeout 参数会定义登录超时 (以毫秒计)。</p> <p>此值可覆盖为连接字符串选项 -c 甚至 ODBC 属性 SQL_ATTR_LOGIN_TIMEOUT。</p> <p>注: 此参数仅适用于 TCP 协议。</p>	特定于操作系统
SocketLinger	<p>此参数控制在对接字连接发出 close 之后 TCP 套接字延迟 (SO_LINGER) 行为。发出 close() 时, 此参数指示系统是尝试传递任何已缓冲的数据 (yes) 还是废弃该数据 (no)。</p>	否
SocketLingerTime	<p>此参数定义套接字在 close 发出后处于不确定状态的时间间隔长度 (以秒计)。如果在正常关闭序列完成之前该时间间隔到期, 那么会执行异常关闭序列 (将废弃数据)。缺省值 0 表示使用系统缺省值 (通常为 1 秒)。</p>	0
跟踪	<p>如果此参数设置为 yes, 那么会将有关已建立网络连接的跟踪信息写入 TraceFile 参数所指定的文件。</p>	否
TraceFile	<p>如果 Trace 参数设置为 yes, 那么会将有关网络消息的跟踪信息写入此参数所指定的文件。</p> <p>跟踪文件会输出到服务器或客户端的当前工作目录, 具体取决于用于启动跟踪的端点。</p>	soltrace.out

A.2.2 TransparentFailover 节

TransparentFailover 参数是客户端参数。

表 24. TransparentFailover 参数

[TransparentFailover]	描述	出厂值
ReconnectTimeout	<p>此参数会指定在 TF 转换或故障转移情况下, 驱动程序在尝试重新连接主服务器之前应该等待的时间长度 (以毫秒计)。如果驱动程序找不到要重新连接的新主服务器, 那么会返回错误, 且 TF 连接会中断。</p>	10000
WaitTimeout	<p>此参数会指定驱动程序应该等待服务器切换状态的时间长度 (以毫秒计)。当驱动程序尝试重新连接服务器时, 该驱动程序可能会连接正处于中间 (切换中或不确定) 状态的服务器。</p>	10000

A.3 高可用性控制器 (HAC) 参数

本节描述 solidhac.ini 配置文件中的高可用性控制器 (HAC) 配置参数。

HAC 配置文件 `solidhac.ini` 分成不同的节；这些节在下面的节中进行了描述。各个参数是按照它们在配置文件中的相同顺序提供的。

有关 `solidhac.ini` 配置文件的示例，请参阅 第 126 页的 A.5.2, 『`solidhac.ini` 配置文件』一节。

`solidhac.ini` 配置文件中参数名称、值和节标题的格式遵循 `solid.ini` 配置文件中所使用的相同约定。

要更改 HAC 参数设置，编辑 `solidhac.ini` 文件并重新启动 HAC。

[HAController] 节

表 25. HAC 配置参数: [HAController] 节

参数名称	描述	必需	出厂值
Listen	<p>Listen 参数的值指定 HAC 用于与高可用性管理器 (HAM) 或 <code>solidDB</code> 实用程序 (例如, <code>solsql</code> 和 <code>solcon</code>) 进行通信的协议和端口。</p> <p>例如, <code>Listen=tcp 3135</code>.</p> <ul style="list-style-type: none"> 唯一受支持的协议是 TCP/IP (“tcp”)。 如果无法启动侦听 (例如, 因为该端口已被另一个进程使用), 那么该信息会写入日志文件 (<code>hacmsg.out</code>), 然后 HAC 会终止。 	X	
NetcopyErrorLevel	<p>此参数定义在将错误消息写入 <code>hacmsg.out</code> 文件, 且 HAC 切换到 ADMINISTRATIVE 模式之前, <code>hsb connect</code> 操作后的 <code>hsb netcopy</code> 操作可以失败的次数。</p> <p>在修订该问题后, 通过发出命令 <code>admin command 'hacontroller setautomatic'</code> 将 HAC 切换回 AUTOMATIC 模式。</p> <p>该参数值必须是正 64 位整数。</p>		10
NetcopyWarningLevel	<p>此参数定义在警告消息写入 <code>hacmsg.out</code> 文件之前, This parameter defines the number <code>hsb connect</code> 后的 <code>netcopy</code> 操作可以失败的次数。</p> <p>该参数值必须是正 64 位整数。</p>		3
StartInAutomaticMode	<p>此参数会指定 HAC 的初始方式; 定义 HAC 是否以自动方式开始执行。</p> <p>一旦 HAC 在运行, 它就可能处于下列方式之一: “自动”或“管理”。</p> <ul style="list-style-type: none"> 在 AUTOMATIC 方式 (yes) 下, HAC 将通过更改服务器的 HSB 状态, 并在必要时重新启动服务器进程来自动尝试使可用性达到最大值。 在 ADMINISTRATIVE 方式 (no) 下, HAC 将仅监视服务器的运行状况。 <p>您可以动态地更改此参数。</p> <p>可能的值为 yes 和 no。 注: 另请参阅 LocalDB 节中的 PreferredPrimary 参数; 仅当 StartInAutomaticMode 参数的值为 yes 时, PreferredPrimary 参数才有效。</p>		yes

表 25. HAC 配置参数: [HAController] 节 (续)

参数名称	描述	必需	出厂值
EnableDBProcessControl	<p>设置 EnableDBProcessControl=yes 时, 将允许 HAC 通过自动启动服务器以及通过为用户提供用于关闭和重新启动数据库进程的命令来管理本地服务器进程。</p> <p>仅当 HAC 处于 AUTOMATIC 方式时此值才有效。</p> <p>可能的值为 yes 和 no。 注: 设置 EnableDBProcessControl=yes 时, 将使 [LocalDB] 节中的 StartScript 参数成为必需参数。</p>		no
EnableAutoNetcopy	<p>设置 EnableAutoNetcopy=yes 时, 将允许 HAC 在使用 hsb connect 命令无法建立 HSB 链路时启动网络复制。</p> <p>可能的值为 yes 和 no。</p>		yes
RequiredConnectFailures	<p>此参数用于定义在认为服务器已失败之前需要的连接尝试连续失败的次数。</p> <p>当服务器状态未知, 或者因其他某些原因而使得 HAC 需要确保服务器状态正确时, 请使用非分块 SQLConnect (check) 命令。如果在这种情况下执行非分块 SQLConnects 失败, 那么请多次重复执行此操作, 直到认为所讨论的服务器无法作出响应为止。</p> <p>值可以是大于 1 的任何数值, 或者是无限制。</p>		1
CheckTimeout	<p>此参数用于定义两次连续非分块 SQLConnect 命令在“检查”方式下的超时 (按毫秒计)。</p> <p>超时值太小可能会导致“假正”问题。换句话说, 尽管服务器在运行, 但是它未能在超时时间段内作出响应, 看起来好像已经失败了一样。</p> <p>值可以是大于 1 的任何毫秒数, 或者是无限制。</p>		150
CheckInterval	<p>此参数会定义两次连续执行非分块 SQLConnect 命令的时间间隔。此值不会影响故障转移时间。通常在发生故障之后或者在系统启动期间执行检查 (轮询)。</p> <p>值可以是大于 1 的任何毫秒数, 或者是无限制。</p>		1000
Username	<p>HAC 的用户名。</p> <p>用户名必须以字母或下划线开头。请使用小写字母 a 至 z、大写字母 A 至 Z、下划线字符“_”以及数字 0 至 9。</p>	X	
Password	<p>由 Username 参数标识的用户的 HAC 密码。</p> <p>密码可以使用任何字母、下划线或数字开头。请使用小写字母 a 至 z、大写字母 A 至 Z、下划线字符“_”以及数字 0 至 9。</p>	X	
DBUsername	<p>HAC 连接至的本地 HotStandby 服务器的用户名。</p> <p>数据库用户应当具有 SYS_ADMIN_ROLE 或 SYS_CONSOLE_ROLE。</p>	X	
DBPassword	<p>由 DBUsername 参数标识的用户本地 HotStandby 服务器的密码。</p>	X	
ApplicationConnTestUsername	<p>定义应用程序连接测试 (EnableApplicationConnCheck 设置为 yes) 中所用连接的用户名。</p> <p>如果 EnableApplicationConnCheck 设置为 yes 并且未设置此参数的值, 那么将使用 DBUsername 参数的值。</p>		
ApplicationConnTestPassword	<p>定义应用程序连接测试 (EnableApplicationConnCheck 设置为 yes) 中所用连接的密码。</p> <p>如果 EnableApplicationConnCheck 设置为 yes 并且未设置此参数的值, 那么将使用 DBPassword 参数的值。</p>		

[LocalDB] 节

表 26. HAC 配置参数: [LocalDB] 节

参数名称	描述	必需	出厂值
Connect	<p>此参数用于定义本地数据库服务器的连接信息。HAC 在连接至本地服务器时使用此信息。</p> <p>连接信息由通信协议 (tcp) 和服务器端口组成, 例如: tcp 2125。</p>	X	
EnableApplicationConnTest	<p>如果设置为 yes, 那么将启用对应用程序连接的定期测试。应用程序连接测试会检查应用程序用于连接服务器的连接是否处于工作中。</p> <p>为了测试连接, HAC 使用与应用程序相同的连接信息来连接服务器, 并执行简单命令以确保该服务器有响应。</p> <p>应用程序连接测试可允许 HAC 监视 solidDB 服务器的外部可用性。如果无法建立与服务器的连接或者服务器对简单查询没有响应, 那么 HAC 会推断出该服务器所提供的服务不可用。如果系统中存在服务器进程, 那么 HAC 会调用使用 UnresponsiveActionScript 参数指定的脚本。</p> <p>要点: 在启用应用程序连接测试之前, 请确保已创建了测试所使用的用户帐户。如果应用程序连接测试因用户名或密码无效而无法连接 solidDB, 那么它将等待, 直到使用 ApplicationConnTestInterval 参数指定的时间到期为止。这时, 还会将错误输出到 hacmsg.out。</p> <p>另外, 如果 EnableApplicationConnTest 设置为 yes 并且两个服务器都缺少应用程序连接测试程序所使用的用户帐户, 那么 solidDB 的速度可能会明显变慢。如果要恢复, 请执行下列操作:</p> <ol style="list-style-type: none"> 1. 使用 ADMIN COMMAND 'hac suspend' 命令暂挂 HAC 操作。 2. 在两个服务器上创建用户帐户。 3. 使用 ADMIN COMMAND 'hac resume' 命令恢复 HAC 操作。 		no
ApplicationConnTestConnect	<p>此参数用于定义应用程序连接测试连接 (EnableApplicationConnCheck 设置为 yes) 的连接信息。</p> <p>此参数的值需要与应用程序连接服务器时所使用的连接信息相同。此值由通信协议 (tcp)、服务器地址和端口号组成, 例如: tcp -i10.0.0.101 2125。</p> <p>如果未指定 ApplicationConnTestConnect 参数, 那么将使用通过 Connect 参数定义的值。</p>		
EnableUnresponsiveActions	<p>如果设置为 yes, 那么当应用程序连接测试失败时, 将执行用户提供的脚本。</p> <p>此脚本通过 UnresponsiveActionScript 参数进行定义。</p>		no
RequiredAppConnTestFailures	<p>此参数用于定义将服务器视为无响应之前应用程序连接测试命令的执行次数。</p>		3
ApplicationConnTestTimeout	<p>此参数用于定义应用程序连接测试中所用连续命令的超时 (以毫秒计)。</p>		5000
ApplicationConnTestInterval	<p>此参数用于定义应用程序连接测试中所用连续的非分块命令之间的时间间隔 (以毫秒计)。</p>		3000

表 26. HAC 配置参数: [LocalDB] 节 (续)

参数名称	描述	必需	出厂值
StartScript	<p>此参数声明用来启动数据库进程的脚本的名称, 例如, /home/soliddb/start_solid.sh.</p> <p>如果使用 EnableDBProcessControl 参数将 HAC 配置成控制数据库进程, 那么此参数是必需参数:</p> <ul style="list-style-type: none"> • 如果 EnableDBProcessControl 设置为 yes, 那么此参数是必需参数。 • 如果 EnableDBProcessControl 设置为 no, 那么此参数无效。 <p>提示: 在 Linux 和 UNIX 环境中, 您可以对 sh 使用 -x 选项, 以在执行命令时打印这些命令及其参数。</p> <p>例如:</p> <pre>#!/bin/sh -x</pre> <p>缺省情况下, 会将启动脚本 (start_solid.sh) 的日志输出到 /tmp/hac_start_solid.err 文件中。缺省情况下, 会将启动 solidDB 进程的日志输出到 /tmp/hac_start_solid.out 文件中。</p>	请参阅描述	
UnresponsiveActionScript	<p>此参数用于定义当应用程序连接测试失败时, 包含所要采取的操作的脚本的名称和位置, 例如: /home/solid/terminate_solid.sh.</p> <p>调用脚本时, HAC 需要将 solidDB 进程标识指定为参数。如果 HAC 不知道 solidDB 进程标识, 那么将无法执行脚本。</p> <p>如果 EnableApplicationConnTest 和 EnableUnresponsiveActions 参数都设置为 yes, 那么此参数将是必需参数。</p> <p>示例:</p> <p>下列 terminate_solid.sh 脚本将在 Linux 和 UNIX 操作系统中终止进程标识为 1 的 solidDB 进程:</p> <pre>#!/bin/sh #terminate_solid.sh ulimit -c unlimited kill -6 \$1 sleep 30 kill -9 \$1</pre> <p>提示: 在 Linux 和 UNIX 环境中, 您可以对 sh 使用 -x 选项, 以在执行命令时打印这些命令及其参数。</p> <p>例如:</p> <pre>#!/bin/sh -x</pre> <p>操作脚本 (terminate_solid.sh) 的日志将输出到 /tmp/hac_terminate_solid.err 文件中。终止 solidDB 进程的日志将输出到 /tmp/hac_terminate_solid.out 文件中。</p>	请参阅描述	
PreferredPrimary	<p>如果两个服务器的 <i>logpos</i> 值相等, 那么此参数将定义本地服务器是否将成为主服务器。如果两个服务器的 PreferredPrimary 参数具有相同值, 那么第一个服务器将成为新的主服务器。</p> <p>仅当 StartInAutomaticMode 参数的值为 yes 时, 此参数才有效。</p> <p>可能的值为 yes 和 no。</p>		no

[RemoteDB] 节

表 27. HAC 配置参数: [RemoteDB] 节

参数名称	描述	必需	出厂值
Connect	此参数用于定义远程数据库服务器的连接信息。HAC 在连接至远程服务器时使用此信息。 远程数据库是通过提供通信协议、远程服务器 IP 地址及其端口来定义的，例如，tcp 192.168.3.123 2125。	X	

[ERE] 节

表 28. HAC 配置参数: [ERE] 节

参数名称	描述	必需	出厂值
EREIP	此参数用于标识外部引用实体的 IP 地址，例如，192.168.3.1。 另请参阅 RequiredPingFailures 参数。		
RequiredPingFailures	此参数定义在 HAC 断定服务器已与 ERE 断开连接并与客户机网络分隔开之前，必须连续失败的 ping 调用次数。 只能对 ERE 使用此参数。 值可以是大于 1 的任何数值，或者是无限制。		3

A.4 高可用性管理器 (HAM) 配置参数

本节描述 HAManager.ini 配置文件中的高可用性管理器配置参数。

HAManager.ini 配置文件中参数名称、值和节标题的格式遵循 solid.ini 配置文件中所使用的相同约定。

表 29. 高可用性管理器配置参数

参数名称	描述
Header_text	此参数用于为 HA 管理器定义标题文本。此值显示在用户界面中。 这是一个必需参数。
Server1_host Server2_host	这两个参数用于定义 HAC 实例的主机名。
Server1_name Server2_name	这两个参数用于定义 HAC 实例的名称。
Server1_pass Server2_pass	这两个参数用于定义 HAC 实例的密码。
Server1_port Server2_port	这两个参数用于定义 HAC 实例的端口。
Server1_user Server2_user	这两个参数用于定义 HAC 实例的用户名。
Window_title	此参数用于为 HA 管理器定义窗口标题。此值显示在用户界面中。 这是一个必需参数。

A.5 配置文件示例

下列各节提供了与 HotStandby 相关的不同配置文件的示例。

A.5.1 solid.ini 配置文件

下面是第一个 HotStandby 服务器的 solidDB 配置文件 (solid.ini) 的样本摘录:

```
[Com]
; The first server listens to the network with this
; name
Listen = tcp 1320
[HotStandby]
HSBEnabled=yes
; The first server connects to the second server
; using the following connect string.
Connect = tcp 188.177.166.12 1321
AutoPrimaryAlone=No
[Logging]
LogEnabled=yes
```

下面是第二个 HotStandby 服务器的 solidDB 配置文件 (solid.ini) 的样本摘录:

```
[Com]
; The second server listens to the network using the following
; connect string.
Listen = tcp 1321
[HotStandby]
HSBEnabled=yes
; The second server connects to the first server
; using the following connect string.
Connect = tcp 188.177.166.11 1320
AutoPrimaryAlone=No
[Logging]
LogEnabled=yes
```

A.5.2 solidhac.ini 配置文件

高可用性控制器 (HAC) 配置文件 (solidhac.ini) 的样本摘录。

```
=====
; NOTE : Copy this file as solidhac.ini
;        to solidhac working directory
;
; solidDB High Availability Controller inifile
=====

[HAController]
;** HAC connect info
;** HAC clients, HA Manager, for example, use this information.
;** Mandatory
;** Listen=tcp 3135
Listen=

;** Setting StartInAutomaticMode=Yes starts HAC in AUTOMATIC mode.
;** In AUTOMATIC mode, solidhac automatically tries
;** to maximize the availability by changing the HSB states of the
;** server, and restarting the server processes when necessary.
;** In contrast, it can be in ADMINISTRATIVE mode
;** in which HAC only monitors the health of the servers.
;**
;** This is dynamically changeable parameter.
;** Optional
```

```

;** Values : Yes/No, default = Yes
StartInAutomaticMode=

;** Setting EnableDBProcessControl=Yes allows solidhac
;** manage local db process by automatically starting
;** the db, and by providing the user with commands to
;** shutdown and restart db process.
;**
;** Optional
;** Effective only when HAC is in AUTOMATIC mode.
;** Values : Yes/No, default = No
EnableDBProcessControl=

;** Setting EnableAutoNetcopy=Yes allows solidhac to initiate
;** netcopy when HSB link cannot be established with 'hsb connect'.
;**
;** Optional
;** Effective only when HAC is in AUTOMATIC mode.
;** Values : Yes/No, default = Yes
EnableAutoNetcopy=

;** When server state is unknown, or HAC needs, for some other reason, to
;** ensure the state of server, non-blocking SQLConnect command is used.
;** If the execution of non-blocking SQLConnect in such a case fails,
;** it is repeated multiple (RequiredConnectFailures) times before
;** the server in question is considered as non-responsive.
;**
;** Optional
;** Values : 1..n, default=2
RequiredConnectFailures=

;** Timeout in milliseconds for non-blocking SQLConnect commands.
;** Too short interval can cause 'false positives', server seems
;** to be failed because it wasn't able to respond within the timeout period.
;**
;** Optional
;** Values : 1..n, default=150 (milliseconds)
CheckTimeout=

;** Interval between consecutive non-blocking SQLConnect commands.
;** The value doesn't affect on failover time. Checking (polling)
;** takes place typically after failure, or during system startup.
;**
;** Optional
;** default = 1000 (milliseconds)
CheckInterval=

;** HAC user identification
;** Mandatory
Username=
Password=

;** HSB server user identification
;** Mandatory
DBUsername=
DBPassword=

;** Identification for application connection test
;** These values are used when ApplicationConnectionTest
;** thread monitors the connection, and availability of
;** the server.
;** If values are not set, and
;** LocalDB.EnableApplicationConnCheck=Yes, then DBUsername, and
;** DBPassword are used.
;**
;** Optional
ApplicationConnTestUsername=

```

```

ApplicationConnTestPassword=

[LocalDB]
;** soliddb connect info
;** Mandatory
;** Connect=tcp 2125
Connect=

;** Enable periodical connection testing in the server.
;** In practice, HAC connects to the server, and executes
;** simple command(s) to ensure the responsiveness.
;**
;** Optional
;** default = No
EnableApplicationConnTest=

;** Connect info for applications, used in application connection test,
;** if it is enabled.
;**
;** Optional, if not specified, LocalDB.Connect is used.
;** ApplicationConnect=tcp 10.0.0.101 2125
ApplicationConnTestConnect=

;** Enables execution of the user-provided script when application
;** connection test fails. The script is defined with
;** UnresponsiveActionScript.
;**
;** Optional
;** default = No
EnableUnresponsiveActions=

;** Number of times the application connection test commands
;** are executed before the server is considered unresponsive.
;**
;** Optional
;** default = 3
RequiredAppConnTestFailures=

;** Timeout in milliseconds for consecutive application connection
;** test commands.
;**
;** Optional
;** default = 5000 (milliseconds)
ApplicationConnTestTimeout=

;** Interval between consecutive non-blocking application connection
;** test commands.
;**
;** Optional
;** default = 30000 (milliseconds)
ApplicationConnTestInterval=

;** The name of the script, which is used to initiate the db process.
;**
;** Optional, except if HAC controls db process (EnableDBProcessControl=Yes).
;** Value is not effective if EnableHACActions=No or EnableDBProcessControl=No
;** StartScript=/home/solid/start_solid.sh
StartScript=

;** The name and location of the script that contains the intended actions that
;** take place if application connection test fails.
;** When calling the script, HAC needs to specify the solidDB® process
;** identifier as a parameter. If HAC does not know the solidDB process id,
;** the script cannot be executed.
;**
;**
;** Optional, except if ApplicationConnectionTest=Yes, and

```

```

; ** EnableUnresponsiveAction=Yes
; **
; ** UnresponsiveActionScript=/home/solid/terminate_solid.sh
UnresponsiveActionScript=

; ** Setting PreferredPrimary=Yes moves local HSB server to as Primary
; ** in the case where either of the servers could start as Primary.
; ** If both servers have PreferredPrimary=No, or no value, first
; ** (new Primary) server wins.
; **
; ** Optional
; ** Value is not effective if EnableHACActions=No.
; ** Values : Yes/No, default no
PreferredPrimary=

[RemoteDB]
; ** soliddb connect info
; ** Mandatory
; ** Connect=tcp 192.168.3.123 2125
Connect=

[ERE]
; ** IP address of an ERE
; ** Optional
; ** Connect=192.168.3.1
EREIP=

; ** The number of consecutive ping calls that must
; ** fail before HAC concludes that the server is
; ** disconnected from ERE.
; **
; ** Optional
; ** Values : 1..n, default=3
; ** RequiredPingFailures=10
RequiredPingFailures=

```

A.5.3 HAManager.ini 配置文件

下面是高可用性管理器配置文件（HAManager.ini）的样本摘录：

```

;=====
;solidDB High-Availability Manager
; Configuration file HAManager.ini
; V. 0.3
; 2008-21-01
;=====
; ** HA Controller connect info, for example
;Server1_name = Server 1
;Server1_host = node1.acme.com
;Server1_port = 2220
;Server2_name = Server 2
;Server2_host = node2.acme.com
;Server2_port = 2220

; All the following lines are mandatory.
Window_title = HA Manager
Header_text = SolidDB HA Manager

; Display names, host addresses and port numbers
; of the SolidHAC (HA Controllers) instances
;Server 1 HA Controller
;-----
Server1_name = Server 1
Server1_host = localhost
Server1_port = 1234

```

```
Server1_user = foo
Server1_pass = bar
;
;Server 2 HA Controller
;-----
Server2_name = Server 2
Server2_host = 192.168.0.1
Server2_port = 1234
Server2_user = foo
Server2_pass = bar
```


附录 B. HotStandby 的错误代码

本节说明了与 HotStandby 相关的错误代码。

solidDB 错误代码的完整列表可在《IBM solidDB 管理员指南》中的附录『错误代码』内找到。

本节中说明的某些错误是 ADMIN COMMAND 结果集的 RC（返回码）列的值，而其他某些错误是作为 ODBC 或 JDBC 驱动程序的错误代码返回的。例如，B.1，『HotStandby 错误和状态码』和第 140 页的 B.2，『高可用性控制器错误和状态码』节中的大部分错误都是 ADMIN COMMAND 结果集值，而所有第 144 页的 B.5，『solidDB 通信错误』都由驱动程序返回。

B.1 HotStandby 错误和状态码

使用 HotStandby 命令时，会发生 solidDB HotStandby 错误（14009 - 147xx 和 307xx）。

有关 HotStandby 的 solidDB 服务器错误

本节列示了与 HotStandby 相关的 solidDB 服务器错误。在《IBM solidDB 管理员指南》的 *solidDB 服务器错误* 一节中，提供了有关“服务器”类中错误的完整列表。

表 30. 有关 HotStandby 的 solidDB 服务器错误

代码	类	输入	描述
14003	纸带机	返回码	ACTIVE 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: <ul style="list-style-type: none">• ADMIN COMMAND 'hotstandby status switch'• ADMIN COMMAND 'hotstandby status catchup'• ADMIN COMMAND 'hotstandby status copy' 含义：切换进程、同步复制进程、复制或网络复制进程仍然处于活动状态。
14007	纸带机	返回码	正在连接 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: <ul style="list-style-type: none">• ADMIN COMMAND 'hotstandby status connect' 含义：主服务器和辅助服务器正在建立连接。
14008	纸带机	返回码	同步复制 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: <ul style="list-style-type: none">• ADMIN COMMAND 'hotstandby status connect' 含义：主服务器已连接至辅助服务器，但是尚未完全复制事务日志。仅从主服务器中返回此消息。

表 30. 有关 HotStandby 的 solidDB 服务器错误 (续)

代码	类	输入	描述
14009	纸带机	返回码	<p>先前未进行服务器切换。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby status switch' <p>含义: 两个服务器之间从未发生切换进程。</p>
14501	纸带机	Error	<p>操作失败。</p> <p>含义: 操作失败, 服务器即将关闭。故障原因可能是由于对非 HotStandby 服务器发出了命令, 或者是对命令不适用于的主服务器或辅助服务器发出了命令。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby switch primary' • ADMIN COMMAND 'hotstandby switch secondary' • ADMIN COMMAND 'hotstandby cominfo' • ADMIN COMMAND 'hotstandby connect' • ADMIN COMMAND 'hotstandby status switch' • ADMIN COMMAND 'hotstandby set standalone' • ADMIN COMMAND 'hotstandby copy' • ADMIN COMMAND 'hotstandby netcopy'
14502	纸带机	Error	<p>RPC 参数无效</p> <p>含义: HSB 连接字符串中提供的某些连接信息是错误的, 并且连接至另一个服务器失败。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby status connect'
14503	纸带机	Error	<p>通信错误, 丢失连接。</p> <p>含义: 发生了通信错误, 未找到另一个服务器。连接至另一个服务器时发生故障。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby status connect'
14520	纸带机	Error	<p>服务器是 HotStandby 辅助服务器, 不允许建立连接。</p>
14522	纸带机	Error	<p>未指定 HotStandby 复制目录。</p> <p>含义: 未指定复制目录。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby copy' <p>要解决此问题, 要么在命令中指定目录, 例如:</p> <pre>ADMIN COMMAND 'hotstandby copy \Secondary\dbfiles\'</pre> <p>要么在 solid.ini 配置文件中设置 CopyDirectory 参数。</p>

表 30. 有关 HotStandby 的 solidDB 服务器错误 (续)

代码	类	输入	描述
14523	纸带机	Error	<p>切换进程已处于活动状态。</p> <p>含义: 在 HotStandby 服务器中, 切换进程已处于活动状态。如果您只需要完成当前切换, 那么请等待。如果您要尝试再次进行切换 (即, 切换回最初配置), 那么必须等待第一次切换完成之后才能开始第二次切换。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby switch primary' • ADMIN COMMAND 'hotstandby switch secondary' • ADMIN COMMAND 'hotstandby status switch'
14524	纸带机	Error	<p>HotStandby 数据库具有不同的基本数据库, 数据库时间戳记是不同的。</p> <p>含义: 数据库来源于不同的种子数据库。您必须使这些数据库同步。可能需要执行从主服务器的数据库网络复制到辅助服务器。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby connect' • ADMIN COMMAND 'hotstandby status switch'
14525	纸带机	Error	<p>HotStandby 数据库之间未正确地同步。</p> <p>含义: 数据库之间未正确地同步。您必须使这些数据库同步。可能需要使用命令行选项 -x backupserver 启动其中一个数据库服务器 (您要使其成为辅助服务器的那个数据库服务器), 然后使用 netcopy 命令将主服务器的数据库复制到辅助服务器。</p> <p>有关在接收到错误 14525 之后如何使主服务器和辅助服务器重新同步的更多信息, 请参阅当 hsb netcopy 命令运行了很长时间而导致 hsb 连接失败 (错误为 14525) 时, 如何使 HotStandby 主服务器和辅助服务器重新同步?。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby connect' • ADMIN COMMAND 'hotstandby status switch'
14526	纸带机	Error	<p>参数无效。</p> <p>含义: HotStandby ADMIN COMMAND 中使用的参数未知或者无效。</p> <p>所有 HotStandby 命令都可能会在 ADMIN COMMAND 的结果集中返回此错误。</p> <p>注意: 在下列 HotStandby 命令中, “参数无效”错误是一个语法错误, 这时, 无法将指定的主服务器或辅助服务器应用于切换:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby switch primary' • ADMIN COMMAND 'hotstandby switch secondary'
14527	纸带机	Error	<p>这是一个非 HotStandby 服务器。</p> <p>含义: 在一个未对 HotStandby 配置的服务器上执行了此命令。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby connect' • ADMIN COMMAND 'hotstandby status switch' • ADMIN COMMAND 'hotstandby switch primary' • ADMIN COMMAND 'hotstandby switch secondary' • ADMIN COMMAND 'hotstandby state'

表 30. 有关 HotStandby 的 solidDB 服务器错误 (续)

代码	类	输入	描述
14528	纸带机	Error	<p>两个 HotStandby 数据库都是主数据库。</p> <p>含义: 两个数据库都是主数据库。这是一个致命错误, 这是因为可能有互相冲突的更改。两个数据库都会被系统自动降级为辅助数据库。您必须决定哪个数据库是真正的主数据库, 然后使两个数据库同步。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby connect' • ADMIN COMMAND 'hotstandby status switch'
14535	纸带机	Error	<p>服务器已经是主服务器。</p> <p>含义: 您尝试切换为主服务器的服务器已经处于 PRIMARY 状态。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> •
14536	纸带机	Error	<p>服务器已经是辅助服务器。</p> <p>含义: 您尝试切换为辅助服务器的服务器已经处于 SECONDARY 状态。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby switch secondary'
14537	纸带机	Error	<p>HotStandby 连接已断开。</p> <p>含义: 从主服务器和辅助服务器中都返回此命令。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby status connect' • ADMIN COMMAND 'hotstandby connect' <p>可能会导致此问题的一个原因是辅助服务器的 solid.ini 文件中具有不正确的连接字符串。如果 netcopy 操作成功, 但是 connect 命令失败, 那么请检查连接字符串。Netcopy 操作不需要辅助服务器与主服务器建立单独的连接, 因此, 即使辅助服务器上的连接字符串错误, netcopy 操作也可能成功。</p>
14538	纸带机	Error	<p>服务器不是 HotStandby 主服务器。</p> <p>含义: 要发出此命令, 服务器必须是一个 HotStandby 主服务器。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby copy copy_directory' • ADMIN COMMAND 'hotstandby netcopy' • ADMIN COMMAND 'hotstandby connect' • ADMIN COMMAND 'hotstandby set primary alone' • ADMIN COMMAND 'hotstandby set standalone'

表 30. 有关 HotStandby 的 solidDB 服务器错误 (续)

代码	类	输入	描述
14539	纸带机	Error	<p>操作被拒绝。</p> <p>当发生下列情况之一时就会提供此错误代码:</p> <ul style="list-style-type: none"> 用户对主服务器发出了 netcopy 命令, 但是应当作为辅助服务器的服务器实际上并不处于 SECONDARY 状态, 或者未采用“netcopy 侦听方式”。(主服务器和“辅助”服务器可能都处于 PRIMARY ALONE 状态。) <p>要解决此问题, 可使用 -x backupserver 命令行选项重新启动“辅助”服务器, 然后再次尝试对主服务器发出 netcopy 命令。</p> <p>警告: 如果这两台服务器先前都处于 PRIMARY ALONE 状态, 以及如果这两台服务器都在处于 PRIMARY ALONE 状态时执行了事务, 那么这两台服务器可能各自具有另一方没有的数据。这是一个严重错误, 并且通过执行 netcopy 使它们恢复同步将导致覆盖已经在“辅助”服务器中落实的一些事务。</p> <ul style="list-style-type: none"> 当您使用回调函数时, 如果此回调函数拒绝关闭或接受 backup 或 netcopy 命令, 就会生成此消息。 <p>当您使用链接库访问时, 可以通过使用 SSCSetNotifier 函数来提供“回调”函数。当已经命令服务器关闭或者执行 netcopy 操作时, 将通知回调函数。如果您的应用程序因某种原因不希望遵从此命令, 那么回调函数可能会返回一个用于取消此命令的值。在这种情况下, 将产生错误 14539。</p> <p>要解决此问题, 请等到客户机代码完成它不希望中断的操作之后, 重试此命令 (例如, 关闭或 netcopy)。</p>
14540	纸带机	Error	服务器已经是一个非 HotStandby 服务器。
14541	纸带机	Error	solid.ini 中的 HotStandby 配置与 ADMIN COMMAND 'HSB SET STANDALONE' 相冲突。
14542	纸带机	Error	服务器采用备份服务器方式。操作被拒绝。
14543	纸带机	Error	命令无效。数据库是一个 HotStandby 数据库, 但是在 solid.ini 配置文件中未找到 [HotStandby] 节。
14544	纸带机	Error	操作失败。此命令在无盘服务器上不受支持。
14545	纸带机	Error	当主服务器的角色为 PRIMARY BROKEN 时, 只能将它设置为 PRIMARY ALONE。
14546	纸带机	Error	<p>切换失败。服务器或远程服务器无法从 PRIMARY ALONE 切换为辅助服务器。在切换之前首先应执行同步复制。</p> <p>含义: 当在处于 PRIMARY ALONE 状态的本地或远程主服务器中执行了将状态切换为 SECONDARY, 并且它检测到主服务器与辅助服务器不同步时, 就会返回此命令。必须将主服务器连接至辅助服务器并且等待同步复制进程完成之后才将辅助服务器切换为主服务器。</p> <p>将返回此错误的 HotStandby 命令:</p> <ul style="list-style-type: none"> ADMIN COMMAND 'hotstandby switch secondary'
14547	纸带机	Error	-R 选项 (读取超时) 的值缺少或者无效。

表 30. 有关 HotStandby 的 solidDB 服务器错误 (续)

代码	类	输入	描述
14548	纸带机	Error	<p>切换失败。无法将处于 STANDALONE 状态的服务器切换到 SECONDARY 状态。</p> <p>含义: 当在处于 STANDALONE 状态的本地或远程主服务器中执行了将状态切换为 SECONDARY, 并且它检测到主服务器与辅助服务器不同步时, 就会返回此命令。必须将主服务器连接至辅助服务器并且等待同步复制完成之后才将辅助服务器切换为主服务器。</p> <p>将返回此错误的 HotStandby 命令:</p> <ul style="list-style-type: none"> • ADMIN COMMAND 'hotstandby switch secondary'
14549	纸带机	Error	<p>HotStandby 事务处于活动状态。</p> <p>含义: 如果 HotStandby 连接已断开, 那么在关闭主服务器之前必须将它设置为 STANDALONE 方式或者切换到 SECONDARY 方式。</p>
14550	纸带机	Error	<p>仅当主服务器未连接至辅助服务器时, 才能更改 HotStandby 连接参数。</p>
14551	纸带机	Error	<p>已达到 START AFTER COMMIT 语句的最大数目。</p>
14552	纸带机	Error	<p>服务器采用备份服务器方式, 不允许建立连接。</p> <p>当客户机尝试与 solidDB 服务器建立连接, 而该服务器处于备份服务器方式 (也称为网络复制侦听方式) 时, 会返回错误 14552。备份服务器方式是特殊的服务器方式, 在此方式下, 已使用命令行选项 -xbackupserver 启动 solidDB 实例。此方式指示 solidDB 实例为辅助服务器, 由于在主服务器上发出了 netcopy 命令, 所以该辅助服务器正在等待或正在接收来自主服务器的数据库文件。</p>

solidDB HotStandby 错误

表 31. solidDB HotStandby 错误

代码	类	输入	描述
14700	HotStandby	Error	<p>拒绝了连接, 两个服务器都具有 PRIMARY 角色。</p> <p>含义: 如果两个节点都具有同一角色, 那么 'hsb connect' 命令将返回此错误。</p>
14701	HotStandby	Error	<p>已拒绝连接, 两台服务器都处于 SECONDARY 角色。</p> <p>含义: 如果两个节点都具有同一角色, 那么 'hsb connect' 命令将返回此错误。</p>
14702	HotStandby	Error	<p>操作失败, 同步复制处于活动状态。</p> <p>含义: 当两个服务器正在执行同步复制时, 如果在主服务器上发出下列任何命令, 就会产生此错误: 'hsb switch secondary'、'hsb set secondary alone'、'hsb set standalone'、'hsb connect'、'hsb copy' 或 'hsb netcopy'。</p> <p>当两个服务器正在执行同步复制时, 如果在辅助服务器上发出下列任何命令, 就会产生此错误: 'hsb switch primary'、'hsb set secondary alone'、'hsb set primary alone'、'hsb set standalone' 或 'hsb connect'。</p>
14703	HotStandby	Error	<p>操作失败, 复制处于活动状态。</p> <p>含义: 当主服务器正在执行复制或网络复制时, 下列命令将返回此错误: 'hsb switch secondary'、'hsb set secondary alone'、'hsb set standalone'、'hsb connect'、'hsb disconnect'、'hsb copy' 或 'hsb netcopy'。</p>
14704	HotStandby	Error	<p>仅当主服务器处于 PRIMARY ALONE 状态时才允许执行 HotStandby 复制或网络复制。</p> <p>含义: 如果服务器处于 PRIMARY ACTIVE 状态并且发出了 'hsb copy' 或 'hsb netcopy' 命令, 就会返回此错误。</p>

表 31. solidDB HotStandby 错误 (续)

代码	类	输入	描述
14705	HotStandby	Error	在此状态下, 不允许设置为 STANDALONE。 含义: 如果服务器处于 PRIMARY ACTIVE 状态并且您发出了 'hsb set standalone' 命令, 那么将产生此消息。
14706	HotStandby	Error	HotStandby 的读取线程方式无效, 仅支持方式 2。
14707	HotStandby	Error	不允许在 STANDALONE 状态下执行操作。
14708	HotStandby	Error	同步复制失败, 在日志文件中找不到同步复制位置。
14709	HotStandby	Error	启用了 HotStandby, 但是未定义连接字符串。
14710	HotStandby	Error	HotStandby 管理命令与入局管理命令冲突。
14711	HotStandby	Error	由于服务器即将关闭而失败。
14712	HotStandby	Error	服务器是辅助服务器。请使用主服务器来完成此操作。

solidDB HSB 错误和消息

表 32. solidDB HSB 错误和消息

代码	类	输入	描述
14007	HSB	信息	正在连接
14008	HSB	信息	同步复制
14009	HSB	信息	自从服务器启动以来, 未进行任何角色切换
14010	HSB	信息	正在断开连接
14522	HSB	信息	未指定 HotStandby 复制目录。
14537	HSB	信息	已断开连接
14704	HSB	Error	仅当主服务器处于 ALONE 状态时, 才允许 HotStandby 复制或网络复制
14712	HSB	Error	服务器是辅助服务器。请将主服务器用于此操作
30500	HSB	信息	已作为 HotStandby 主服务器启动
30501	HSB	信息	已作为 HotStandby 辅助服务器启动
30502	HSB	信息	上次使用数据库作为 HotStandby 辅助服务器启动时, 未正确关闭该数据库
30503	HSB	信息	强制使 HotStandby 主服务器作为辅助服务器启动
30504	HSB	信息	HotStandby 角色已切换至辅助角色
30505	HSB	信息	HotStandby 角色已切换至主角色
30506	HSB	信息	主服务器必须设置为 PRIMARY ALONE 状态或切换至辅助角色。
30507	HSB	信息	HotStandby 服务器已设置为 PRIMARY ALONE 状态。
30508	HSB	信息	HotStandby 服务器已设置为 SECONDARY ALONE 状态
30509	HSB	信息	HotStandby 切换至主服务器时失败, 发生错误 <i>error_code</i>
30510	HSB	信息	HotStandby 切换至辅助服务器时失败, 发生错误 <i>error_code</i>
30511	HSB	信息	无法将 HotStandby 启动至 <i>server_name</i> , 发生错误 <i>error_code</i>
30512	HSB	信息	无法将 HotStandby 角色切换为主角色, 发生错误 <i>error_code</i>
30513	HSB	信息	无法将 HotStandby 角色切换为辅助角色, 发生错误 <i>error_code</i>
30514	HSB	信息	这两个数据库都是作为辅助服务器启动的主服务器
30515	HSB	信息	这两个 HotStandby 数据库都是主数据库
30516	HSB	信息	无法将 HotStandby 启动至 <i>server_name</i> , 其他服务器已拒绝, 发生错误 <i>error_code</i>
30517	HSB	信息	已切换处于辅助角色的 HotStandby 角色
30518	HSB	信息	HotStandby 角色已切换至独立角色
30530	HSB	信息	正在开始将 HotStandby 同步复制数据发送至辅助服务器
30531	HSB	信息	已成功完成 HotStandby 同步复制
30532	HSB	信息	已异常结束 HotStandby 同步复制

表 32. solidDB HSB 错误和消息 (续)

代码	类	输入	描述
30533	HSB	信息	无法启动 HotStandby 同步复制。辅助服务器未正确地与主服务器同步，需要执行完全同步
30534	HSB	信息	已异常结束 HotStandby 同步复制，发生错误 <i>error_code</i>
30535	HSB	信息	已异常结束 HotStandby 同步复制，发生错误 <i>error_code</i>
30536	HSB	信息	由于通信错误，已异常结束 HotStandby 同步复制
30537	HSB	信息	已异常结束 HotStandby 同步复制，辅助服务器返回错误 <i>error_code</i>
30538	HSB	信息	HotStandby 同步复制大小 <value> 大于所配置的最大大小 <i>value</i> ，正在停止 HotStandby
30539	HSB	信息	HotStandby 同步复制期间文件发生错误，正在停止 HotStandby
30540	HSB	信息	正在开始从主服务器接收 HotStandby 同步复制数据
30541	HSB	信息	由于日志文件已毁坏，导致辅助服务器未正确地与主服务器同步。请重新启动辅助服务器并执行 HSB 网络复制。
30550	HSB	信息	与 HotStandby 辅助服务器的连接已中断
30551	HSB	信息	已连接至 HotStandby
30552	HSB	信息	已连接 HotStandby 辅助服务器
30553	HSB	信息	已连接 HotStandby 主服务器
30554	HSB	信息	与辅助服务器的 HotStandby 连接已中断，已打开的事务正在等待操作员解析事务状态。必须将主服务器设置为单独方式或切换至辅助方式。
30555	HSB	信息	HotStandby ping 操作超时
30556	HSB	信息	与 HotStandby 辅助服务器的连接已中断
30557	HSB	信息	HotStandby 数据库未正确同步
30558	HSB	信息	与辅助服务器的 HotStandby 连接已超时
30559	HSB	信息	HotStandby 连接已中断
30560	HSB	信息	HotStandby: <i>HotStandby_error_message</i>
30561	HSB	信息	已开始连接至 HotStandby
30562	HSB	信息	与 HotStandby 主服务器的连接已中断
30570	HSB	信息	已完成网络备份。
30571	HSB	信息	已开始接收网络备份。
30572	HSB	信息	已通过使用 HotStandby 复制/网络复制启动了数据库。
30573	HSB	信息	网络备份失败。
30574	HSB	信息	HotStandby 将线程数强制设置为 1
30575	HSB	信息	已配置 HotStandby 复制，但是找不到任何处于活动状态的许可证，因此未启动复制
30577	HSB	信息	HotStandby 连接操作失败
30579	HSB	信息	HotStandby 连接已处于活动状态。
30581	HSB	信息	无效事件 <i>event</i>
30582	HSB	信息	HotStandby 无法将服务器设置为 PRIMARY ALONE 状态。
30583	HSB	信息	HotStandby 复制失败。
30585	HSB	信息	数据库开始侦听网络复制。
30586	HSB	信息	HotStandby 同步复制， <i>catchup_phase</i> 日志位置: <i>log_position</i> <i>catchup_phase</i> 可以是: <ul style="list-style-type: none"> • HSB waitdurable • HSB catchup start • HSB write catchup • HSB write switch
30750	HSB	信息	HotStandby 连接已处于活动状态。
30752	HSB	信息	操作失败，断开连接处于活动状态。
30757	HSB	信息	已连接

表 32. solidDB HSB 错误和消息 (续)

代码	类	输入	描述
30758	HSB	信息	HotStandby 命令不正确。
30759	HSB	信息	HotStandby 服务器已设置为 STANDALONE 状态。
30760	HSB	信息	已启动从服务器断开连接的进程。
30761	HSB	信息	已启动将角色切换至主角色的进程。
30762	HSB	信息	已启动将角色切换至辅助角色的进程。
30763	HSB	信息	已启动连接服务器的进程。
30764	HSB	信息	已启动复制。
30765	HSB	信息	AutoPrimaryAlone 参数已设置为 Yes。
30766	HSB	信息	AutoPrimaryAlone 参数已设置为 No。
30767	HSB	信息	Connect 参数已设置为 <i>value</i> 。
30768	HSB	信息	HotStandby 连接已中断。
30769	HSB	信息	由于服务器之间的连接处于活动状态，所以操作失败。
30772	HSB	信息	必须在 INI 文件中定义 HotStandby 节点标识。
30774	HSB	信息	服务器已经处于 STANDALONE 状态。
30775	HSB	信息	CopyDirectory 参数已设置为 <i>value</i> 。
30776	HSB	信息	ConnectTimeout 参数已设置为 <i>value</i> 。
30777	HSB	信息	PingTimeout 参数已设置为 <i>value</i> 毫秒。
30779	HSB	信息	HotStandby 迁移处于活动状态
30782	HSB	信息	服务器已设置为 PRIMARY ALONE 状态。
30783	HSB	信息	服务器已设置为 SECONDARY ALONE 状态。
30784	HSB	信息	<i>parameter_name</i> 参数已设置为 <i>value</i> 。
30785	HSB	信息	<i>parameter_name</i> 参数已设置为 <i>value</i> 。
30786	HSB	信息	<i>parameter_name</i> 参数已设置为 <i>value</i> 。
30787	HSB	致命错误	pri_dologskip: 类型、日志位置和日志大小错误。 此错误是由 HSB 主服务器上的失败操作导致的。此错误将返回失败的操作及其在日志中的位置以及日志大小。跳过了复制日志中的操作。
30788	HSB	致命错误	pri_hsblogcopy_write: 类型、日志位置和日志大小错误。 此错误是由 HSB 主服务器上的失败操作导致的。写入复制日志文件失败。此错误将返回失败的操作及其在日志中的位置以及日志大小。
30789	HSB	致命错误	打开热备用复制日志文件失败。
30790	HSB	致命错误	为 HotStandby 日志分配内存失败。最大日志大小为 <i>logsize</i> 。 此错误与使用 HotStandby 的无盘数据库相关。在这些系统中，会将 HotStandby 日志写入内存。如果为日志文件分配更多内存失败，就会产生此错误。
30791	HSB	致命错误	HotStandby:solhsby:bad 类型 <i>type</i> 、日志位置 <i>log_pos</i> 和日志大小 <i>log_size</i>
30792	HSB	信息	这两个服务器都是辅助服务器。
30793	HSB	信息	已达到辅助服务器任务的最大数目 <i>value</i> 。 辅助服务器中用于入局日志操作的队列正在快速增长，以致这些操作无法执行，而且也无法向主服务器进行确认。 可以使用性能计数器 HSB 辅助服务器队列数 来监视该队列。

B.2 高可用性控制器错误和状态码

当使用特定的高可用性控制器命令时发生 solidDB 高可用性控制器错误 (17xxx)。

表 33. 高可用性控制器错误和状态码

错误或状态码	描述
17501	<p>HAC 即将关闭。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none">• ADMIN COMMAND 'hac shutdown' <p>含义: 高可用性控制器即将关闭。</p>
17502	<p>命令失败, HAC 已被暂挂。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none">• ADMIN COMMAND 'hac suspend'• ADMIN COMMAND 'hac gethsbdstate'• ADMIN COMMAND 'hac getdbstate'• ADMIN COMMAND 'hac shutdowndb'• ADMIN COMMAND 'hac restartdb'• ADMIN COMMAND 'hac switchdb'• ADMIN COMMAND 'hac statemachinestate'• ADMIN COMMAND 'hac getereip'• ADMIN COMMAND 'hac pingere' <p>含义: 命令执行失败, 高可用性控制器已被暂挂。</p>
17503	<p>未知命令。请输入 'hac commands' 以获取帮助。</p> <p>含义: ADMIN COMMAND 语法不正确。</p>
17504	<p>HAC 已经在运行。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none">• ADMIN COMMAND 'hac resume'
17506	<p>HSB 状态不允许进行切换。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none">• ADMIN COMMAND 'hac switchdb'
17507	<p>无法执行命令。</p> <p>在命令的结果集中可能会返回此状态的 ADMIN COMMAND:</p> <ul style="list-style-type: none">• ADMIN COMMAND 'hac shutdowndb'

表 33. 高可用性控制器错误和状态码 (续)

错误或状态码	描述
17509	重新启动数据库服务器失败。有关详细信息，请参阅 solmsg.out。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac restartdb'
17510	无法连接至数据库服务器。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac switchdb'
17511	数据库服务器未关闭。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac restartdb'
17513	切换失败。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac switchdb'
17514	在配置文件中未指定 ERE IP。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac getereip' • ADMIN COMMAND 'hac pingere'
17516	HAC 已处于活动状态。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac setactive' • ADMIN COMMAND 'hac pingere'
17517	HAC 已处于被动状态。 在命令的结果集中可能会返回此状态的 ADMIN COMMAND: • ADMIN COMMAND 'hac setpassive'

B.3 有关 HotStandby 的 solidDB 数据库错误

solidDB 数据库错误 (10002 - 10050) 由 solidDB 检测到并发送至客户机应用程序。它们可能要求执行管理操作。

本节列示了与 HotStandby 相关的 solidDB 数据库错误。在《IBM solidDB 管理员指南》的 *solidDB 数据库错误* 一节中，提供了有关“数据库”类中错误的完整列表。

表 34. solidDB 数据库错误

代码	类	输入	描述
10002	数据库	Error	操作失败。 含义：连接操作因发生意外错误而失败。很可能是因为服务器之间未正确地同步。
10013	数据库	Error	事务是只读事务。 含义：您已尝试在设置为只读的事务中写入，或者服务器临时（例如，在状态切换期间）设置为只读方式。不允许使用可更新的事务。
10019	数据库	Error	备份已处于活动状态。 含义：当一个备份或副本已经在运行时，您又尝试启动一个备份或副本。
10024	数据库	Error	备份目录“ <i>directory_name</i> ”非法。 含义：备份或副本目录是一个空字符串或者是一个句点（表示将在当前目录中创建备份或副本）。
10030	数据库	Error	备份或副本目录“ <i>directory_name</i> ”不存在。 含义：未找到备份或副本目录。请检查备份或副本目录的名称是否正确。
10045	数据库	Error	不能在 HotStandby 辅助服务器上执行此操作。 含义：不能在 HotStandby 辅助服务器上执行此操作。 为了成功执行所请求的操作，服务器必须是主服务器。
10046	数据库	Error	操作失败，数据字典操作处于活动状态。 含义：当前正在进行数据字典操作。
10047	数据库	Error	复制的事务异常终止。 含义：事务异常终止，例如，在状态切换时异常终止。当服务器状态从主服务器切换为辅助服务器时，所有活动的事务都将异常终止。
10048	数据库	Error	复制的事务包含数据字典更改，不允许执行正常的更新操作。 含义：HotStandby 方式将限制执行数据字典操作；例如，不能将 CREATE TABLE 与正常的更新操作混合在一起执行。 此消息在版本 4.1 和更高版本中已过时，这些版本允许您在使用 HSB 时在事务中执行 DML 和 DDL 操作。
10049	数据库	Error	远程服务器不是辅助服务器。 含义：您在命令中指定的服务器不处于 SECONDARY 状态。
10050	数据库	Error	复制操作更新了 BLOB 列。 含义：无法将 BLOB 列复制到辅助服务器。
10078	数据库	Error	用户回滚了事务。
10079	数据库	Error	无法除去文件规范。已经在使用文件。

表 34. solidDB 数据库错误 (续)

代码	类	输入	描述
10080	数据库	Error	HotStandby 辅助服务器无法执行从主服务器中接收到的操作。 含义: 发生此错误的原因可能是未使用 HotStandby copy 或 netcopy 命令复制主服务器中的数据库。
10081	数据库	Error	数据库文件不完整或者被破坏。 含义: 如果此文件位于 HotStandby 辅助服务器上, 那么请再次使用 hotstandby copy 或 hotstandby netcopy 命令从主服务器中发送此文件。
10082	数据库	Error	备份异常终止。
10083	数据库	Error	由于已经将落实操作发送至辅助服务器, 因此未能异常终止 HSB 事务。
10084	数据库	Error	表未被锁定。
10085	数据库	Error	已禁止执行检查点操作。
10087	数据库	Error	不允许将主内存表用于 HotStandby。
10088	数据库	Error	指定的锁定超时太长。
10089	数据库	Error	操作失败, 服务器处于 HSB PRIMARY UNCERTAIN 状态。

B.4 solidDB 表错误

solidDB 数据库表错误是由错误的 SQL 语句导致的, solidDB 将检测到这些错误。不需要执行管理操作。

本节列示了与 HotStandby 相关的 solidDB 表错误。在《IBM solidDB 管理员指南》的『solidDB 表错误』一节中, 提供了有关“表”类中错误的完整列表。

表 35. solidDB 表错误

代码	类	输入	描述
13068	表	Error	服务器正在关闭 含义: 由于服务器正在关闭, 因此您无法完成此操作。
13123	表	Error	“table_name”表不是空的 含义: 只有在表为空的情况下才能执行此操作。例如, 只有在表为空的情况下才能将基于磁盘的表更改为内存表 (或者将内存表更改为基于磁盘的表)。
13167	表	Error	只有 M 表才能是瞬态表 含义: 不能创建一个基于磁盘的瞬态表。例如, 以下 SQL 语句将产生此错误消息: CREATE TRANSIENT TABLE t1 (i INT) STORE DISK;
13170	表	Error	只有 M 表才能是临时表 含义: 不能创建一个基于磁盘的临时表。例如, 以下 SQL 语句将产生此错误消息: CREATE TEMPORARY TABLE t1 (i INT) STORE DISK;

B.5 solidDB 通信错误

solidDB 通信错误（21306 和 21308）是由网络错误导致的。这些错误要求执行管理操作。

本节列示了与 HotStandby 相关的 solidDB 通信错误。在《IBM solidDB 管理员指南》的『solidDB 通信错误』一节中，提供了有关“服务器”类中错误的完整列表。

表 36. solidDB 通信错误

代码	类	输入	描述
21306	Communication	Error	<p>未找到服务器“<i>server_name</i>”，连接失败。</p> <p>含义：未找到辅助服务器。</p> <ul style="list-style-type: none">• 检查服务器是否正在运行。• 检查网络名是否有效。• 检查服务器是否在侦听所指定的网络名。
21308	Communication	Error	<p>“连接已中断（<operation> 操作失败，错误代码为 <code>）”。</p> <p>例如，</p> <p>"连接已中断（TCP/IP 'Write' 操作失败，错误代码为 7）。”</p> <p>建议的操作：</p> <ul style="list-style-type: none">• 检查服务器是否正在运行。• 检查网络名是否有效。• 检查服务器是否在侦听所指定的网络名。

附录 C. HotStandby 和 HAC ADMIN COMMAND

本节描述了可用于 HotStandby 的 ADMIN COMMAND。

根据使用的工具，ADMIN COMMAND 语法可能会有所不同：

- **solidDB SQL 编辑器 (solsql)**

将命令名与 **solsql** 配合使用时，必须用单引号将命令名引起来。例如：

```
ADMIN COMMAND 'hotstandby switch primary';
```

```
ADMIN COMMAND 'hacontroller shutdown';
```

- **solidDB 远程控制 (solcon)**

将命令名与 **solcon** 配合使用时，命令名不得带有 ADMIN COMMAND 前缀和引号。

例如：

```
hotstandby switch primary
```

```
hacontroller shutdown
```

提示：可以将“hotstandby”缩写为“hsb”，将“hacontroller”缩写为“hac”。例如：

```
ADMIN COMMAND hsb switch primary
```

```
ADMIN COMMAND hac shutdown
```

在语法描述中，使用了可以采用的最短格式；省略了 ADMIN COMMAND 和引号，并使用了缩写“hsb”和“hac”。

注：如果命令中没有语法错误，那么 ADMIN COMMAND 始终返回一个成功的返回码 (0)。结果集的“RC”字段中包含此命令的实际结果代码。

有关 ADMIN COMMAND 的更多信息，请参阅《《IBM solidDB SQL 指南》》中的『ADMIN COMMAND』一节。

C.1 HotStandby 命令（管理命令）

表 37. HotStandby 命令（管理命令）

命令	说明
hsb cominfo	返回用于连接至另一个服务器的连接字符串。这通常是 HotStandby.Connect 参数的值，但是也可以使用以下命令设置此值： <pre>ADMIN COMMAND 'hsb parameter connect connect_string';</pre> 可以在应用程序中使用此信息来连接至其他服务器。

表 37. HotStandby 命令（管理命令）（续）

命令	说明
<p>hsb connect</p>	<p>如果主服务器与辅助服务器之间的连接已断开或者尚未建立连接，那么此命令会将主服务器连接至辅助服务器并启动 HotStandby 复制。由于没有在服务器之间建立自动连接的机制，因此，此命令始终需要连接服务器。在成功连接之后，主服务器的状态会自动从 PRIMARY ALONE 设置为 PRIMARY ACTIVE。如果连接不成功，那么状态仍然为 PRIMARY ALONE。</p> <p>可以在主服务器或辅助服务器上执行此命令。</p> <p>注： 当您执行此命令时，如果主服务器与辅助服务器已建立连接，但是尚未将事务日志完全复制到辅助服务器中，那么将显示下列消息：Catchup is active。</p>
<p>hsb copy [<i>directory_name</i>]</p>	<p>要点： 不推荐使用此命令。请改用 hsb netcopy 命令。</p> <p>可以使用 hsb copy 命令根据主数据库来创建初始辅助数据库。此命令会将数据库复制到主节点本地的一个目录（当然也会复制到辅助节点本地的一个目录）。完成复制之后，您就可以启动辅助服务器。将主服务器连接至辅助服务器之后，主服务器会将事务日志复制到辅助服务器，从而使辅助服务器也变成最新的。</p> <p>还可以使用此命令使主数据库与位于主节点本地目录中的辅助数据库同步（要在辅助数据库相当长一段时间内处于脱机状态才进行此操作）。请阅读第 49 页的 3.4.5, 『使主服务器与辅助服务器同步』。</p> <p>如果指定了可选的 <i>directory_name</i>，那么会将数据库文件复制到此目录中；否则，会将这些数据库文件复制到使用 solid.ini 配置文件的 [hotstandby] 节中的 copydirectory 参数指定的目录中。因为 hsb copy 命令不会复制 solid.ini 配置文件或者日志文件，所以建议您使此目录不同于正常的备份目录。</p> <p>主服务器只有处于 PRIMARY ALONE 状态时才能执行 hsb copy 命令。在执行此命令期间和执行之后，服务器将保持 PRIMARY ALONE 状态。完成此命令之后，您可以启动辅助服务器，并连接这两个服务器。</p>
<p>hsb disconnect</p>	<p>此命令指示服务器正常断开与 HSB 对另一个成员的连接。此命令在主服务器或辅助服务器上都有效。使用此命令的一个典型原因是在对其中一个服务器升级之前将两个服务器断开连接。（可以将另一个服务器设置为 PRIMARY ALONE 状态，以便它可以继续对客户机请求作出响应。）</p> <p>此命令通常会导致两个服务器都进入“ALONE”方式；即，主服务器从 PRIMARY ACTIVE 切换为 PRIMARY ALONE；而辅助服务器从 SECONDARY ACTIVE 切换为 SECONDARY ALONE。</p> <p>此命令在主服务器和辅助服务器上都有效。</p> <p>注： 使用 shutdown 命令 ADMIN COMMAND 'shutdown' 会导致服务器在关闭之前执行受控制的断开连接操作。如果辅助服务器已关闭（并且已断开连接），那么主服务器知道进入 PRIMARY ALONE 状态是安全的，并将进入此状态。</p>

表 37. HotStandby 命令 (管理命令) (续)

命令	说明
<p>hsb logpos</p>	<p>此命令会返回上次操作时的日志操作标识和服务器的角色 (PRIMARY、SECONDARY 和 STANDALONE)。</p> <p>下面显示了此命令的典型输出:</p> <pre>ADMIN COMMAND 'hsb logpos'; RC TEXT -- ---- 0 00000000000000000000000000871:PRIMARY</pre> <p>此命令可用于在下列情况下确定两个服务器中的哪个服务器应该成为主服务器: 例如在两个数据库都发生故障之后且您不知道哪个服务器应该成为主服务器时。(在连接中断之前是主服务器的服务器现在并不一定应该成为主服务器。)</p> <p>从原则上来说, 具有较大日志操作标识值的服务器已接受更多事务, 因此该服务器应该成为主服务器。但是, 如果您在 HSB 连接失败之后已对两个服务器进行了更新, 那么比较日志操作标识值将不再可靠。</p> <p>有关如何使用此命令的更多信息, 请参阅第 61 页的 3.4.9, 『选择使哪个服务器成为主服务器』。</p>
<p>hsb netcopy</p>	<p>此命令可用于将主数据库或者无盘内存数据复制到辅助服务器。这些数据库文件是利用主数据库 solid.ini 文件中的 HotStandby.Connect 参数所定义的连接通过网络链路进行复制的。</p> <p>例如, 您可能要使用此命令使主数据库与已经长时间处于脱机状态的辅助数据库进行同步。您还可以使用 hsb netcopy 来创建新的辅助数据库, 例如用于替换已损坏的辅助数据库、为新的 HotStandby 配置设置辅助数据库或将新的辅助数据库添加到现有配置中。</p> <p>主服务器必须处于 PRIMARY ALONE 状态才能发出此命令。</p> <p>完成此命令之后 (无论成功还是失败), 主服务器将保持 PRIMARY ALONE 状态。如果成功完成了复制, 那么辅助服务器将自动切换到 SECONDARY ALONE 状态。</p> <p>提示: 通常, 在执行 hsb netcopy 命令之后应接着执行 hsb connect 命令, 以连接主服务器和辅助服务器。将主服务器连接至辅助服务器之后, 主服务器会通过复制事务日志来自动使辅助服务器也变成最新的。</p> <p>有关如何使用 hsb netcopy 命令的更多详细信息, 请参阅第 49 页的 3.4.5, 『使主服务器与辅助服务器同步』和第 53 页的 『通过网络将主数据库复制到辅助服务器』。</p>

表 37. HotStandby 命令 (管理命令) (续)

命令	说明
<p>hsb parameter</p>	<p>要点: 不推荐使用此命令。</p> <p>此命令允许您设置特定于 HSB 的参数, 例如, AutoPrimaryAlone、Connect 和 PingTimeout。有关每个这些参数的完整描述, 请参阅第 115 页的附录 A, 『HotStandby 配置参数』。</p> <p>当您设置其中某个参数的值时, 此命令将立即生效, 但是在执行 shutdown 命令之前不会写入 solid.ini 配置文件。</p> <p>此命令的语法为:</p> <pre>ADMIN COMMAND 'hsb parameter param_name param_value';</pre> <p>此命令不使用等号。因此, 它不同于其他类似的命令 (建议您这样做):</p> <pre>ADMIN COMMAND 'parameter hotstandby.param_name = param_value';</pre>
<p>hsb role</p>	<p>要点: 不推荐使用此命令。请改用 hsb state 命令。</p> <p>在结果集中返回下列其中一个角色:</p> <ul style="list-style-type: none"> • PRIMARY, 如果所连接的服务器是一个正常的主服务器。在此角色下, 会将主服务器中的事务发送至辅助服务器。 • PRIMARY NOHSBLOG, 表示主服务器将接受事务并将它们存储在数据库中。但是, 它不会将事务存储在日志中以便稍后可以将它们发送至辅助服务器。要使辅助服务器与主服务器重新同步, 必须将主服务器中的整个数据库复制到辅助服务器。 • PRIMARY BROKEN, 如果主服务器已与辅助服务器断开连接。在主服务器中只能执行只读事务。 • PRIMARY ALONE, 如果主服务器在独自运行。与辅助服务器的连接已断开, 但是主服务器接受了事务并且已添加至事务日志, 以便稍后可以将它们发送至辅助服务器。 • PRIMARY CATCHUP, 如果正在进行同步复制。在同步复制期间, 在主服务器中发出 'hsb connect' 命令之后, 主服务器会自动将事务日志更改发送至辅助服务器。完成同步复制进程之后, 服务器的角色将自动切换为 PRIMARY。如果在连接之前主服务器的角色为 PRIMARY ALONE, 那么主服务器将继续接受事务。 • SECONDARY, 如果所连接的服务器是一个正常的辅助服务器。这意味着此服务器将从主服务器中接收并应用事务。 • SECONDARY BROKEN, 如果辅助服务器已与主服务器断开连接。 • SECONDARY CATCHUP, 如果在主服务器中发出 'hsb connect' 命令之后, 辅助服务器正在同步复制主服务器中所作的更改。完成同步复制进程之后, 辅助服务器的角色将自动切换为 SECONDARY。 <p>如果在一个未对 HotStandby 配置的服务器上发出了 hsb role 命令, 那么将返回下列错误消息: 14527: This is a non-HotStandby Server。</p> <p>此命令将与 SQL 函数 HOTSTANDBY_ROLE 返回相同的信息。</p>

表 37. HotStandby 命令 (管理命令) (续)

命令	说明
<p>hsb set primary alone</p>	<p>此命令将主服务器无条件地设置为 PRIMARY ALONE 状态。如果服务器当前处于下列其中一种状态，就可以使用此命令：PRIMARY ACTIVE、SECONDARY ACTIVE、SECONDARY ALONE 和 STANDALONE。</p> <p>可以使用此命令来实现快速故障转移。当辅助服务器处于 SECONDARY ACTIVE 状态时，如果接收到此命令，那么此服务器将不会尝试与主服务器进行通信。它将立即切换为 PRIMARY ALONE 状态。如果在辅助服务器检测到故障之前，就已将有关主服务器故障的信息传送到 HAC 中（延迟时间由 PingTimeout 和 PingInterval 参数指定），就会产生此行为。</p> <p>但是，如果在辅助服务器中执行此命令时主服务器是活动的并且处于 PRIMARY ACTIVE 状态（例如，不正确的故障检测就会导致这种情况），那么将强制使主服务器进入 PRIMARY UNCERTAIN 状态。然后，可以将它过渡到 SECONDARY ALONE 状态并重新进行连接，而不会丢失任何事务。</p> <p>注：用于执行故障转移的另一种方法是使用 hsb switch primary 命令。</p> <p>在 PRIMARY ALONE 状态下，与辅助服务器的连接已断开，但是此状态允许您运行主服务器并连续更新事务日志。PRIMARY ALONE 状态将一直保持到主服务器已关闭、成功地连接至辅助服务器或者服务器用尽了用于保存事务日志的磁盘空间为止。</p> <p>将服务器设置为 PRIMARY ALONE 状态时，该服务器不会自动尝试与另一个服务器重新建立连接。</p> <p>要点：在服务器上执行此命令之前，请确保服务器对中的另一个服务器尚未处于 PRIMARY ALONE 状态（或 STANDALONE 状态）。这是为了避免双重主服务器 - 有关更多详细信息，请参阅第 64 页的 3.6.2, 『网络分区和双重主服务器』。</p> <p>另请参阅 'hsb switch primary' 命令。</p>
<p>hsb set secondary alone</p>	<p>此命令将服务器状态设置为 SECONDARY ALONE。如果服务器当前处于下列其中一种状态，就可以使用此命令：PRIMARY ALONE、PRIMARY UNCERTAIN 和 STANDALONE。</p>
<p>hsb set standalone</p>	<p>发出此命令时，主服务器的状态将变为 STANDALONE。服务器将停止为辅助服务器存储事务。主服务器（处于 STANDALONE 状态）可以继续接受读/写事务。当辅助服务器在相当长一段时间内处于脱机状态，并且事务日志可能变得太大时，在主服务器中使用此选项就很有用。如果服务器当前处于下列其中一种状态，就可以使用此命令：PRIMARY ALONE 或 SECONDARY ALONE。</p>

表 37. HotStandby 命令（管理命令）（续）

命令	说明
<p>hsb state</p>	<p>返回服务器的状态:</p> <ul style="list-style-type: none"> • PRIMARY ACTIVE, 如果所连接的服务器是一个正常的主服务器。在此状态下, 会将主服务器中的事务发送至辅助服务器。 • STANDALONE, 表示主服务器将接受事务并将它们存储在数据库中。但是, 它不会存储事务以将它们转发至辅助服务器。 • PRIMARY UNCERTAIN, 如果主服务器已与辅助服务器断开连接, 且尚未切换到另一种状态 (例如, PRIMARY ALONE)。在主服务器中只能执行只读事务。 • PRIMARY ALONE, 如果主服务器在独自运行。与辅助服务器的连接已断开, 但是主服务器接受了事务并且已存储在事务日志, 以便可以将它们转发至辅助服务器。 • SECONDARY ACTIVE, 如果所连接的服务器是一个正常的辅助服务器。这意味着此服务器将从主服务器中接收并应用事务。 • SECONDARY ALONE, 如果辅助服务器已与主服务器断开连接。 <p>如果在一个未对 HotStandby 配置的服务器上发出了 ADMIN COMMAND 'hsb state' 命令, 那么将返回下列错误消息: 14527: This is a non-HotStandby Server.</p> <p>此命令将与 SQL 函数 HOTSTANDBY_STATE 返回相同的信息。有关此函数的详细信息, 请参阅第 92 页的『将应用程序切换到新的主服务器』中的使用 <i>HOTSTANDBY_STATE</i> 函数一节。</p> <p>有关在执行管理操作和故障诊断操作时所发生的 HotStandby 状态过渡的概述, 请参阅第 155 页的附录 D, 『服务器状态过渡』。</p>
<p>hsb status option</p>	<p>此命令将返回最后一个成功启动的操作的 HotStandby 状态信息。option 可以是下列任何一项:</p> <ul style="list-style-type: none"> • catchup • connect • copy • switch <p>有关更多详细信息, 请参阅下面对各个命令/选项 (例如, hsb status catchup) 的描述。</p> <p>status 命令将提供有关成功启动之后花了很长时间执行的操作的结果信息。如果启动某项操作失败 (例如, 由于状态不正确而失败), 那么 status 命令将不会返回此操作的状态, 而是返回所执行的前一个命令的状态。</p>
<p>hsb status catchup</p>	<p>此命令将指示服务器是否正在执行同步复制, 即, 辅助服务器在读取主服务器的事务日志并应用更改。</p> <p>可能的返回值为:</p> <ul style="list-style-type: none"> • ACTIVE • NOT ACTIVE

表 37. HotStandby 命令 (管理命令) (续)

命令	说明
hsb status connect	<p>此命令将返回状态信息:</p> <ul style="list-style-type: none"> • CONNECTED (已连接) - 连接处于活动状态。从主服务器和辅助服务器中都将返回此信息。 • CONNECTING (正在连接) - 主服务器与辅助服务器正在互相连接。从主服务器和辅助服务器中都将返回此信息。 • CATCHUP (同步复制) - 主服务器已连接至辅助服务器, 但是尚未将 HotStandby 主数据库日志完全复制到辅助服务器。从主服务器和辅助服务器中都将返回此信息。 • BROKEN (已断开) - 主服务器与辅助服务器之间的连接已断开。从主服务器和辅助服务器中都将返回此信息。 <p>注: 此命令将与 SQL 函数 HOTSTANDBY_CONNECTSTATUS 返回相同的信息。有关此函数的详细信息, 请参阅第 92 页的『将应用程序切换到新的主服务器』中的使用 HOTSTANDBY_CONNECTSTATUS 函数一节。</p>
hsb status copy	<p>此命令允许您检查上一个 hsb copy 或 hsb netcopy 命令的结果。即使您正在检查 hsb netcopy 的结果而不是 hsb copy 的结果, 此 status 命令也将始终使用 copy 关键字。</p> <p>返回的状态信息:</p> <ul style="list-style-type: none"> • SUCCESS (成功) - 成功完成了复制。 • ACTIVE (活动) - 仍然在完成复制进程。 • ERROR number (错误) - 复制失败, 并返回了错误代码 <i>number</i>。
hsb status switch	<p>此命令将返回 HotStandby 切换状态信息:</p> <ul style="list-style-type: none"> • ACTIVE (活动) - 仍然在完成复制进程。 • SUCCESS (成功) - 成功完成了复制。 • ERROR number (错误) - 复制失败, 并返回了错误代码 <i>number</i>。 • NO SERVER SWITCH OCCURRED BEFORE (先前没有发生服务器切换) - 先前没有发生切换。
hsb switch primary	<p>此命令会将数据库服务器切换为 PRIMARY 状态。此命令将启动一个切换进程, 可以使用 hsb status switch 命令来监视此进程。</p> <p>如果在执行此命令时服务器之间已建立连接, 那么服务器之间将只是颠倒状态 - 即, 旧的主服务器将从 PRIMARY ACTIVE 更改为 SECONDARY ACTIVE 状态, 而辅助服务器将从 SECONDARY ACTIVE 切换为 PRIMARY ACTIVE 状态。</p> <p>如果服务器之间未连接并且服务器处于 SECONDARY ALONE 状态, 那么当您将此服务器切换为主服务器时, 它最终将处于 PRIMARY ALONE 状态。新的主服务器将不会自动尝试连接至另一个服务器并切换为 PRIMARY ACTIVE 状态。</p> <p>因为在 SECONDARY ACTIVE 和 SECONDARY ALONE 状态下都可以执行此命令, 所以可以使用它来执行故障转移。但是, 因为此服务器始终将尝试与主服务器通信, 所以可能会发生网络超时。因此, 使用此方法比使用 hsb set primary alone 命令时的速度更慢。另一方面, 此方法可以更好地防止存在双重主服务器这种可能性。</p> <p>另请参阅 hsb set primary alone 命令。</p>

表 37. HotStandby 命令 (管理命令) (续)

命令	说明
hsb switch secondary	<p>此命令会将数据库切换为 SECONDARY 状态。所有活动的写入事务都将终止。</p> <p>注: 如果所连接的数据库服务器是主服务器, 那么它将成为辅助服务器。如果旧的辅助服务器可用, 那么此旧的辅助服务器将切换为主服务器 (请参阅 hsb switch primary 命令)。</p> <p>注: 如果在一个打开的事务中 (在 Windows 环境中, 是在启动此事务之后, 但是在执行 COMMIT 语句之前) 发出 hsb switch secondary 命令, 那么当您发出 COMMIT 语句时, 该语句将失败并发生错误: replicated transaction is aborted。将回滚在执行 switch 命令期间产生的所有事务, 其中包括执行此 switch 语句的事务。switch 本身是成功的 (即, 它未回滚), 这是因为 ADMIN COMMAND 不是用于处理事务的命令。但是, 如果尚未打开一个新事务, 那么管理命令将强制启动一个新事务。</p>

C.2 高可用性控制器命令 (ADMIN COMMAND)

如果要发出 HAC 命令, 您必须先使用 solidhac.ini 中的 **HAController.Listen** 参数所定义的端口来连接 HAC。例如, 您可以使用 **solsql** 或 ODBC 接口来连接 HAC。

表 38. 高可用性控制器命令 (ADMIN COMMAND)

命令	说明
hac restartdb 缩写: hac rsd	如果已使用 hac shutdowndb 命令终止 solidDB 服务器进程, 那么此命令指示 HAC 重新启动 solidDB 服务器进程。
hac resume 缩写: hac rs	当 HAC 处于暂挂方式 (利用 hac suspend 进行设置) 时, 此命令将恢复 HAC 操作。不会重新读取 solidhac.ini 文件, 即对 solidhac.ini 文件所做的任何更改都不会生效。
hac setadministrative 缩写: hac sad	此命令会将 HAC 设置为 ADMINISTRATIVE 方式。
hac setautomatic 缩写: hac sam	此命令会将 HAC 设置为 AUTOMATIC 方式。
hac shutdown 缩写: hac sd	此命令将终止 HAC 进程。
hac shutdowndb 缩写: hac sdd	此命令指示 HAC 关闭 solidDB 服务器进程。当使用此命令终止服务器进程时, HAC 不会尝试重新启动服务器进程。要重新启动服务器进程, 请使用 hac restartdb 命令。
hac suspend 缩写: hac sp	此命令会关闭所有 HAC 操作 (线程), HAC 用于侦听 ADMIN COMMAND 的线程除外。当 HAC 处于暂挂时, 可以利用 ADMIN COMMAND hac resume 来恢复操作。

表 38. 高可用性控制器命令 (ADMIN COMMAND) (续)

命令	说明
<p>trace { on off } hac</p> <p>缩写: tra { on off } hac</p>	<p>此命令会控制 HAC 操作的跟踪。跟踪信息会输出到 HAC 工作目录中的 <code>hactrace.out</code> 文件。</p> <p>通常, IBM 软件支持和开发团队使用 HAC 跟踪工具进行故障诊断。您也可以生成该跟踪, 以取得有关您正在调查的问题的信息, 但是如果没有 <code>solidDB</code> 源代码的知识, 那么对这些信息的使用相当有限。</p> <p>使用 HAC 跟踪工具对性能的影响最小。</p>

附录 D. 服务器状态过渡

本章描述可能会发生的状态过渡（例如，从 OFFLINE 过渡到 SECONDARY ALONE）。

第 8 页的『对服务器状态的描述』中对每种服务器状态进行了描述。

D.1 HotStandby 状态过渡图

本节中的图显示了可能会发生的状态过渡以及在哪些情况下可能会发生这些状态过渡。

例如，可以通过执行 `'hsb set primary alone'` 命令将服务器的状态从 PRIMARY UNCERTAIN 更改为 PRIMARY ALONE:

```
ADMIN COMMAND 'hsb Set Primary Alone';
```

当您使用此图时，请记住下列内容:

1. 未显示命令的完整语法。例如，它显示:

```
'hsb set primary alone'
```

而不是:

```
ADMIN COMMAND 'hsb Set Primary Alone';
```

2. 为 `'hsb copy'` 显示的状态过渡路径也适用于 `'hsb netcopy'`。
3. 执行某些命令时可能会失败。当一个命令既可能成功也有可能失败时，将同时显示这两种可能性。如果打算使用分支来描述命令失败时将发生的情况，那么将使用“失败”这个词: `'Disconnect' failed`。
4. 在某些情况下，行为取决于称为 **AutoPrimaryAlone** 的 `solid.ini` 配置参数的设置。通常使用缩写“APA”来表示此参数。
5. 当图中提到“事件”时，它指的是在内部生成的通知。这些事件与用户可以公告和等待的“事件”不同，如用于 `CREATE EVENT` 的 `SQL` 命令中所述。
6. 靠近此图的左上角，您将看到使用 `"-x backupserver"` 启动的文本。如果您要启动新的辅助服务器，并且您希望它通过“`netcopy`”命令从主服务器中获取数据库的副本，那么请从操作系统命令行中使用命令行选项 `-x backupserver` 来启动此服务器。此命令将让服务器等待从主服务器中发出的 `netcopy`。请注意，当服务器在等待接收 `netcopy` 时，此服务器将不会对有关其状态（或角色）的查询作出响应。例如，如果您发出以下命令:

```
ADMIN COMMAND 'hsb state';
```

那么此服务器将不会作出响应，因此您实际上看不到它返回“OFFLINE”状态。

7. “`rpc`”代表“远程过程调用”。“`rpc 已断开`”意味着主服务器与辅助服务器之间在未执行显式断开连接的情况下互相失去了连接。如果网络失败或者其中一个服务器已崩溃，就可能会失去连接。

8. 当一个箭头指向它的起始状态时，表示状态不变。例如，如果一个服务器处于 PRIMARY ALONE 状态，并且它尝试连接至另一个服务器但失败了，那么它将保持 PRIMARY ALONE 状态。

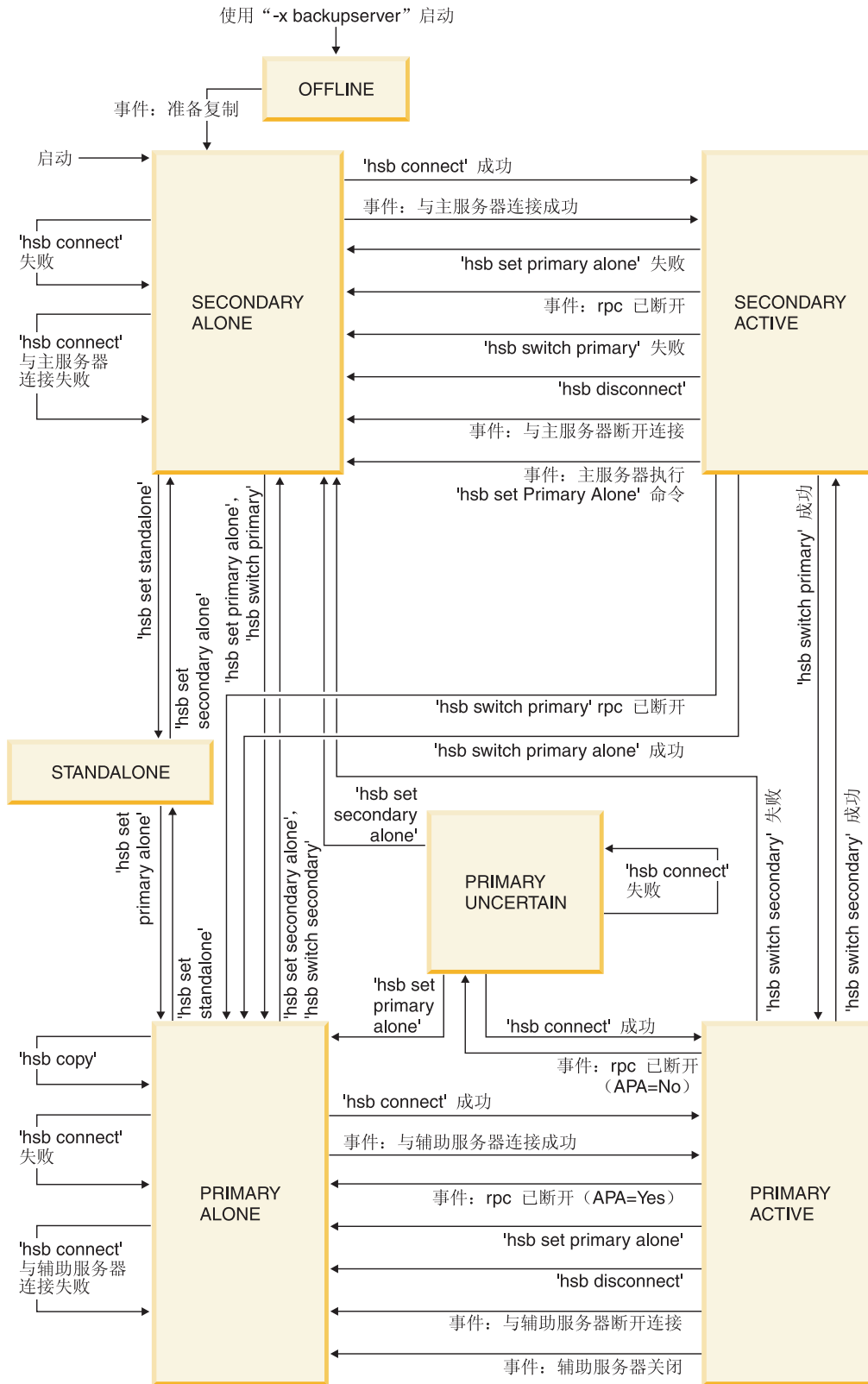


图 19. HotStandby 服务器状态过渡

下表显示了服务器状态以及 HotStandby 命令可以用来更改服务器状态的方式。

表 39. 服务器状态过渡表

服务器状态	如果发生此情况, 或者如果发出了此 HSB 命令...	那么服务器状态将变为...	如果此命令不成功, 那么状态为...
OFFLINE	如果主服务器执行 ADMIN COMMAND 'hotstandby netcopy' , 那么在复制数据库之后, 辅助服务器的状态将更改为 SECONDARY ALONE 。	SECONDARY ALONE	保持不变
PRIMARY ACTIVE	当 AutoPrimaryAlone = Yes 时发生了 HotStandby 超时 (自动)。 注意: 当辅助服务器关闭或者主服务器与辅助服务器之间的连接断开时, 就会自动产生 HSB 超时。	PRIMARY ALONE	(不适用)
PRIMARY ACTIVE	当 AutoPrimaryAlone = No 时发生了 HotStandby 超时 (自动)。 注意: 当辅助服务器关闭或者主服务器与辅助服务器之间的连接断开时, 就会自动产生 HSB 超时。	PRIMARY UNCERTAIN	(不适用)
PRIMARY ACTIVE	在主服务器中发出 ADMIN COMMAND 'hotstandby set standalone' 命令	STANDALONE	保持不变
PRIMARY ACTIVE	在主服务器中发出 ADMIN COMMAND 'hotstandby switch secondary' 命令, 或者在辅助服务器中发出 ADMIN COMMAND 'hotstandby switch primary' 。	SECONDARY ACTIVE	SECONDARY ALONE
PRIMARY ACTIVE	在主服务器中发出 ADMIN COMMAND 'hotstandby disconnect' 。	PRIMARY ALONE	PRIMARY ALONE

表 39. 服务器状态过渡表 (续)

服务器状态	如果发生此情况, 或者如果发出了此 HSB 命令...	那么服务器状态将变为...	如果此命令不成功, 那么状态为...
PRIMARY ALONE	<p>在主服务器中发出 ADMIN COMMAND 'hotstandby copy' 或 ADMIN COMMAND 'hotstandby netcopy' 命令。</p> <p>请注意, 主服务器的状态将保持不变。它将保持 PRIMARY ALONE 状态。要将状态更改为 PRIMARY ACTIVE, 必须发出“connect”命令: ADMIN COMMAND 'hotstandby connect';</p> <p>注意: 如果您使用的是无盘服务器, 但是您对辅助服务器没有文件访问权, 那么必须使用 netcopy 而不使用 copy。</p>	PRIMARY ALONE	PRIMARY ALONE
PRIMARY ALONE	<p>在主服务器中发出 ADMIN COMMAND 'hotstandby connect' 命令</p> <p>注意: 以上命令用来连接至现在已修正的辅助服务器, 或者连接至除了失败的辅助服务器之外的其他服务器。</p>	PRIMARY ACTIVE (在完成同步复制之后)	保持不变
PRIMARY ALONE	<p>在主服务器中发出 ADMIN COMMAND 'hotstandby set standalone' 命令, 或者发生事务日志已满的情况。</p>	STANDALONE	保持不变
PRIMARY ALONE	<p>在主服务器中发出 ADMIN COMMAND 'hotstandby set secondary alone' 或者 ADMIN COMMAND 'hotstandby switch secondary'。</p>	SECONDARY ALONE	SECONDARY ALONE

表 39. 服务器状态过渡表 (续)

服务器状态	如果发生此情况, 或者如果发出了此 HSB 命令...	那么服务器状态将变为...	如果此命令不成功, 那么状态为...
PRIMARY UNCERTAIN	在主服务器中发出 ADMIN COMMAND 'hotstandby set primary alone' 命令	PRIMARY ALONE	保持不变
PRIMARY UNCERTAIN	在主服务器中发出 ADMIN COMMAND 'hotstandby connect' 命令。 注: 以上命令用来连接至 现在已修正的辅助服 务器, 或者连接至除 了失败的辅助服务器 之外的其他服务器。	PRIMARY ACTIVE	保持不变
PRIMARY UNCERTAIN (连接至 辅助服务器时发生了 HSB 超时)	在主服务器中发出 ADMIN COMMAND 'hotstandby set standalone' 命令	STANDALONE	保持不变
PRIMARY UNCERTAIN	在主服务器中发出 ADMIN COMMAND 'hotstandby set secondary alone' 或 者 ADMIN COMMAND 'hotstandby switch secondary'。	SECONDARY ALONE	保持不变
SECONDARY ACTIVE	HSB 超时 (自动) 注: 当辅助服务器关 闭或者主服务器与辅 助服务器之间的连接 断开时, 就会自动产 生 HSB 超时。	SECONDARY ALONE	(不适用)
SECONDARY ACTIVE	在主服务器中发出 ADMIN COMMAND 'hotstandby switch secondary' 命令, 或 者在辅助服务器中发 出 ADMIN COMMAND 'hotstandby switch primary'。	PRIMARY ACTIVE	保持不变
SECONDARY ACTIVE	在辅助服务器中发出 ADMIN COMMAND 'hotstandby set primary alone' 命 令。	PRIMARY ALONE	保持不变

表 39. 服务器状态过渡表 (续)

服务器状态	如果发生此情况, 或者如果发出了此 HSB 命令...	那么服务器状态将变为...	如果此命令不成功, 那么状态为...
SECONDARY ACTIVE	在辅助服务器或主服务器中发出 ADMIN COMMAND 'hotstandby disconnect' 命令。	SECONDARY ALONE	SECONDARY ALONE
SECONDARY ALONE	在辅助服务器或主服务器中发出 ADMIN COMMAND 'hotstandby connect' 命令。	SECONDARY ACTIVE	保持不变
SECONDARY ALONE	在辅助服务器中发出 ADMIN COMMAND 'hotstandby set standalone' 命令。	STANDALONE	保持不变
SECONDARY ALONE	在辅助服务器中发出 ADMIN COMMAND 'hotstandby set primary alone' 或者 ADMIN COMMAND 'hotstandby switch primary' 命令。	PRIMARY ALONE	保持不变

附录 E. HotStandby 系统事件

本附录只包含特定于 HSB 的事件。有关其他类型的事件的讨论，请参阅其他手册，例如《IBM solidDB SQL 指南》。

每个 HotStandby 操作都将生成一个事件。可以使用诸如看守程序等应用程序来监视这些事件。

事件通过其名称来反映在服务器中发生了特定操作。需要在存储过程中使用特殊语句来接收事件。HotStandby 事件与 solidDB 创建和支持的其他事件没有区别。这些事件将发送给已注册为将在存储过程中接收事件的用户。有关公布、注册和等待事件的详细信息，请阅读《IBM solidDB SQL 指南》中的『存储过程、事件、触发器和序列』、『solidDB SQL 语法』以及《IBM solidDB SQL 指南》。

下表列示了当前可用于 HotStandby 的事件。请注意，大多数事件都包含五个参数，但并不是必需使用所有这些参数。

表 40. HotStandby 事件

HSB 事件	事件参数	事件原因
SYS_EVENT_HSBCONNECTSTATUS	ENAME WVARCHAR、POSTSRVTIME TIMESTAMP、UID INTEGER、NUMDATAINFO INTEGER 和 TEXTDATA WVARCHAR 对于 TEXTDATA，有效值可能是： TEXTDATA = { CONNECTED CONNECTING CATCHUP BROKEN}	主服务器与辅助服务器之间的连接状态更改
SYS_EVENT_HSBSTATESWITCH	ENAME WVARCHAR、POSTSRVTIME TIMESTAMP、UID INTEGER、NUMDATAINFO INTEGER 和 TEXTDATA WVARCHAR 对于 TEXTDATA，有效值可能是： TEXTDATA = { PRIMARY ACTIVE PRIMARY ALONE PRIMARY UNCERTAIN SECONDARY ACTIVE SECONDARY ALONE STANDALONE }	每次状态切换都会发送一个状态切换事件。

表 40. HotStandby 事件 (续)

HSB 事件	事件参数	事件原因
SYS_EVENT_NETCOPYEND	ENAME WVARCHAR, POSTSRVTIME TIME STAMP、UID INTEGER、NUMDATAINFO INTE- GER 和 TEXTDATA WVARCHAR 不使用参数。	HotStandby NETCOPY 操作已结束。 仅当用户正在使用共享内存访问或链接 库访问时，用户才能捕获到此事件。
SYS_EVENT_NETCOPYREQ	ENAME WVARCHAR, POSTSRVTIME TIME STAMP、UID INTEGER、NUMDATAINFO INTE- GER 和 TEXTDATA WVARCHAR 不使用参数。	请求了 HotStandby NETCOPY。 如果用户应用程序回调函数返回非零 值，那么将不执行网络复制。 仅当用户正在使用共享内存访问或链接 库访问时，用户才能捕获到此事件。

附录 F. 看守程序样本

本节讨论 solidDB 安装中包含的样本中提供的看守程序样本应用程序。

看守程序是一个独立程序，用于监视和控制主服务器和辅助服务器。看守程序将同时监视两个热备用服务器，并在必要时切换它们的状态。这就减轻了数据库管理员需要监视服务器的负担。

solidDB 提供了一个样本看守程序，可以将它用作构建能够满足您需要的定制看守程序的基础。此样本应用程序称为“看守程序”。在开始编程之前，请注意“看守程序”样本的下列功能。

- 打算将“看守程序”用作看守程序示例
- 看守程序使用轮询来保持它本身对于服务器状态是最新的
- 看守程序是一个单线程程序
- 看守程序通过 ODBC 使用 HSB API。可以将此 API 实现用作您自己的看守程序的模型。
- 看守程序没有用户界面。

如果您要使用看守程序，那么需要配置 solidDB 配置文件 (solid.ini) 中的 [WatchDog] 节，此文件位于看守程序的当前工作目录中。如果看守程序正在与主服务器或辅助服务器在同一目录中运行，那么您将只有一个 solid.ini 文件，服务器和看守程序将共享此文件。如果看守程序在一个单独的目录中运行，那么看守程序将具有它自己的 solid.ini 文件。

而且，此附录中还说明了特定于看守程序的 solid.ini 配置参数。这些参数是在 solid.ini 配置文件的 [WatchDog] 节中设置的。如果您编写自己的看守程序，那么不需要使用任何这些参数。

有关其他 solid.ini 参数的讨论，请参阅《IBM solidDB 管理员指南》。

F.1 使用看守程序的 HotStandby 配置

HotStandby 配置允许主服务器、辅助服务器和看守程序位于不同机器中并且使用不同的操作系统和 API，如第 167 页的 F.1.2, 『系统设计问题』中的示例所示。有关实现异构配置的详细信息，请阅读第 167 页的 F.1.2, 『系统设计问题』。

主数据库与辅助数据库之间的所有通信（包括使失败的系统恢复运行并且使主数据库与辅助数据库重新同步）都是在现有通信层（例如，TCP/IP）进行的。HotStandby 不需要使用辅助存储器或传输方法（例如，共享磁盘或 FTP 传输）。

要点：如果您正在辅助服务器所在机器上运行看守程序，那么务必将 **AutoPrimaryAlone** 参数设置为 no。在这种情况下，将 **AutoPrimaryAlone** 设置为 no 非常重要，这是因为它可以防止发生存在两个主服务器这种错误。主服务器可能处于 PRIMARY ALONE 状态，并且在发生服务器故障时看守程序可以将辅助服务器切换为 PRIMARY ALONE 状态。如果用户碰巧将旧的辅助服务器设置为新的主服务器，那么也可能发生此错误。有关双重主服务器的更多信息，请参阅第 64 页的 3.6.2, 『网络分区和双重主服务器』。

F.1.1 看守程序的工作方式

当主服务器关闭时，看守程序样本应用程序将通知您。在正常方式下，看守程序将在主服务器和辅助服务器中使用 **hotstandby status connect** 命令来检查服务器的连接状态。

看守程序定期在服务器之间执行此检查。时间间隔是在看守程序的 `solid.ini` 配置文件中的 **PingInterval** 参数设置的。

在尝试轮询给定次数之后，如果看守程序未从主服务器或/和辅助服务器节点中接收到响应，它就会认为 HotStandby 系统中发生了问题。尝试轮询的次数是在看守程序配置文件（`solid.ini` 文件中的 [Watchdog] 节）中的 **NumRetry** 参数中设置的。

看守程序还会观察主服务器和辅助服务器之间是否相连。如果主服务器或辅助服务器向看守程序返回成功连接状态，那么意味着主服务器和辅助服务器仍然相连。另一方面，如果返回一个错误，那么意味着主服务器和辅助服务器不再相连。

如果看守程序配置文件中的 **AutoSwitch** 参数设置为 YES，那么看守程序还将负责在主服务器发生故障时自动切换服务器的状态。例如，当主服务器关闭时，看守程序就会切换辅助服务器以使它成为新的主服务器并将它置于 PRIMARY ALONE 状态。如果 **AutoSwitch** 参数设置为 NO，看守程序本身不会更改服务器状态，但它会将一条消息写入看守程序日志以通知用户切换服务器状态。

要继续监视，看守程序将切换到故障方式，这意味着它将连续检查失败的服务器是否具有可以工作的连接。

故障方式

当看守程序样本应用程序知道 HotStandby 主服务器和辅助服务器已经相连时，看守程序将保持正常方式。如果其中一个服务器已失败，或者服务器之间的通信链路已失败，那么看守程序就会执行一些操作。如果执行操作之后未能使服务器相连，那么看守程序将进入故障方式。

在看守程序进入故障方式之后，它将等待系统管理员解决主服务器和辅助服务器存在的问题。同时，如果再次发生故障，看守程序不会处理此故障。看守程序中的这种局限性是故意这样设计的。在某些情况下，一系列故障甚至表面上看起来合适的响应都可能导致“具有两个主服务器”（处于 PRIMARY ALONE 或 STANDALONE 状态）这种错误。如果网络中短时间内发生故障，但是数据库服务器本身未发生故障，就特别容易产生这种错误。第 167 页的『编写用于处理多个故障的看守程序』中提供了产生两个主服务器的示例。

在故障方式下，看守程序将同时轮询主服务器和辅助服务器。当它能够连接至这两个服务器时，就会对这两个服务器发送 **hotstandby state** 命令，以了解它是否可与这两个服务器通信以及每个服务器所处的状态。

一旦看守程序能够与这两个服务器通信，它就会根据 `solid.ini` 参数 **DualSecAutoSwitch** 来决定下一步要执行的操作。如果 **DualSecAutoSwitch = Yes** 并且两个服务器都是辅助服务器，那么看守程序将自动选择其中一个辅助服务器作为新的主服务器并将它切换为主服务器。如果 **DualSecAutoSwitch = No**，那么系统管理员必须将一个服务器切换为主服务器。请注意，无论看守程序处于“正常”方式还是“故障”方式，**DualSecAutoSwitch** 参数都适用。

编写用于处理多个故障的看守程序

看守程序中可以采用两种方法来处理多个故障。您可以：

- 在每次发生故障（并由看守程序自动作出响应）之后，需要手动（人为）干预以检查所发生的情况。手动干预时可能需要执行一些操作，例如，重新启动服务器或者解决网络问题。看守程序通常也是采用这种方法，因为这样做可以降低存在两个主服务器的风险。
- 编写一个看守程序，可以用来处理随着时间推移而产生的多个故障。

此方法确实存在产生两个主服务器的风险，如以下示例中所示。

双重主服务器

在此示例中，Server1 最初是主服务器，Server2 最初是辅助服务器。

1. 发生了网络故障，Server1 变得不可访问。
2. 看守程序将 Server2 从 SECONDARY 状态切换为 PRIMARY ALONE 状态。
3. 发生了第二个网络故障，Server2 变得不可访问。
4. 修复了第一个网络故障，Server1 再次变得可访问。
5. 由于 Server1 可访问，而 Server2 不可访问，因此看守程序将 Server1 切换为 PRIMARY ALONE 状态。
6. 修复了第二个网络故障，Server2 再次变得可访问。
7. 此时，Server1 和 Server2 都处于 PRIMARY ALONE 状态。

F.1.2 系统设计问题

您用于配置 HotStandby 的方式（本地或远程配置，在一个或多个不同位置配置，通过因特网配置，使用看守程序来配置）会影响系统的可靠性和效率。本节的重点就是解决这些问题。

下图显示了一个异构系统的示例。在此系统中，主服务器和辅助服务器甚至不使用同一类型的硬件和操作系统。

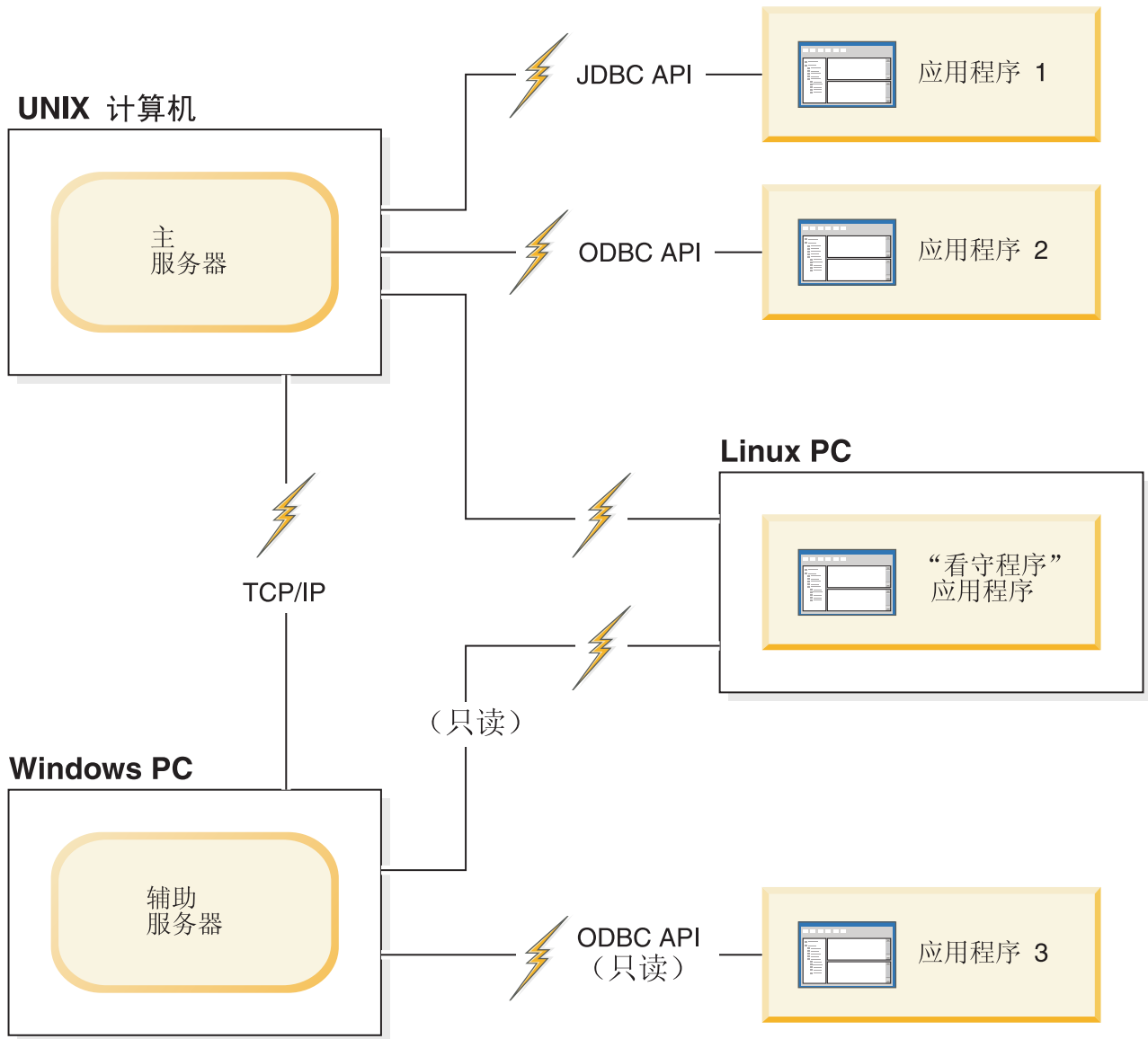


图 20. 使用了看守程序的异构 HotStandby 配置

F.1.3 看守程序配置

为了更有效率并且更准确地监视服务器状态，建议将看守程序作为任何 HotStandby 配置中的独立组件。

如果只有两台机器可用，那么不可能单独使用一台机器来运行看守程序。因此，在辅助服务器所在的机器上运行看守程序，并在主服务器和辅助服务器的 `solid.ini` 配置文件中将 **AutoPrimaryAlone** 参数设置为 `no`。请注意，将此参数设置为 `no` 极其重要，因为该值将防止发生具有两个主服务器的潜在错误。

注意:

如果两个服务器都处于一种允许写入的状态 (**PRIMARY ALONE** 或 **STANDALONE** 状态), 并且两个服务器的数据库是单独更新的, 那么将无法重新使这两个数据库同步。请确保看守程序不允许两个服务器同时处于 **PRIMARY ALONE** 或 **STAND-ALONE** 状态。请参阅第 64 页的 3.6.2, 『网络分区和双重主服务器』。

如果主服务器确实失败了, 那么看守程序能够将辅助服务器切换为新的主服务器。

将看守程序与辅助服务器放在同一台机器中存在一些缺点。这些缺点包括:

- 如果只是看守程序与主服务器之间的通信链路断开了, 那么此配置可能会导致主服务器与辅助服务器之间进行假切换。
- 通信链路成为“单一故障点”, 即, 可能会禁用整个系统的单一故障。(在大多数 HotStandby 配置中, 除非至少发生了两个故障, 否则不会禁用整个系统。)
- 即使发生了网络故障并且辅助服务器所在的机器无法与主服务器所在的机器通信, 用户和应用程序仍然能够访问主服务器, 并且从理论上讲可以继续对主服务器执行操作。但是, 主服务器将停止接受事务, 这是因为看守程序无法通知主服务器继续运行, 例如, 通过切换为 **PRIMARY ALONE** 状态而继续运行。

F.1.4 使用样本看守程序

关于此任务

最初, 应当在两个服务器都已启动并且相连的情况下启动看守程序。

过程

要启动看守程序, 切换至看守程序的当前工作目录, 并在命令提示符处发出以下命令:

```
watchdog
```

如果您在 `solid.ini` 文件尚未对 `connect1` 和 `connect2` 服务器 (它们都能够用作主服务器和辅助服务器) 指定用户名和密码, 那么看守程序将提示您指定这些用户名和密码。

结果

看守程序启动之后, 就会对两个服务器执行 `ping` 操作以检查哪个服务器是主服务器。除非在重试次数超过指定次数之后看守程序检测到服务器故障, 否则看守程序将保持正常方式。如果在看守程序最后一次向服务器发送重试尝试之后还是发生了故障, 那么看守程序就会切换到故障方式。当主服务器和辅助服务器都已启动并已重新连接后, 看守程序就会切换到正常方式。

F.2 故障情况和看守程序操作

本节描述典型的看守程序应当如何处理通常遇到的特定故障情况。

这些情况是发生服务器故障或者主服务器与辅助服务器之间或者其中一个服务器与看守程序之间的通信链路断开时发生的。

尽管管理员或软件程序都可以发出这些命令, 但是为了简单起见, 我们假定这些命令是由看守程序样本发出的。

F.2.1 主服务器已关闭

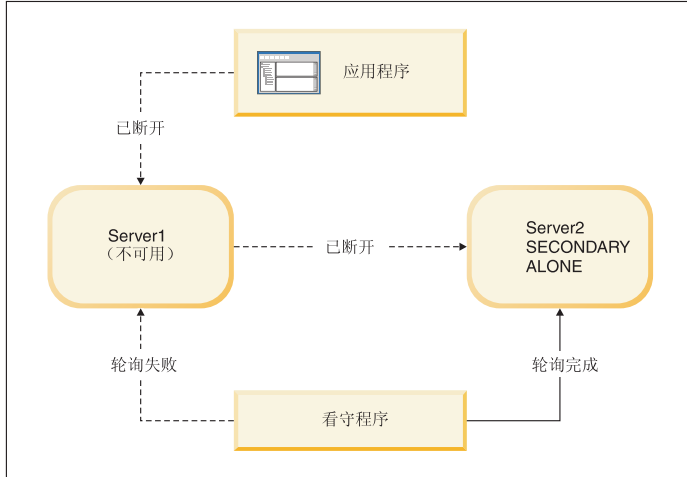
故障情况

与主服务器的所有连接都已断开。

补救措施

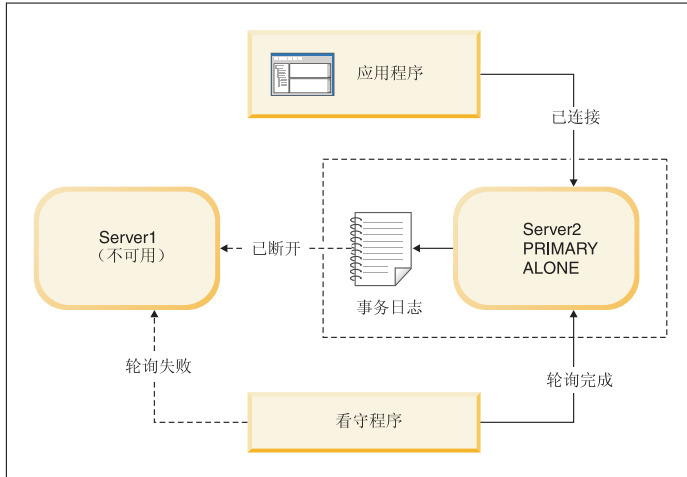
当主服务器已关闭时，将辅助服务器切换为新的主服务器并将新的主服务器设置为 PRIMARY ALONE 状态。然后，旧的主服务器可以成为新的辅助服务器。

故障情况



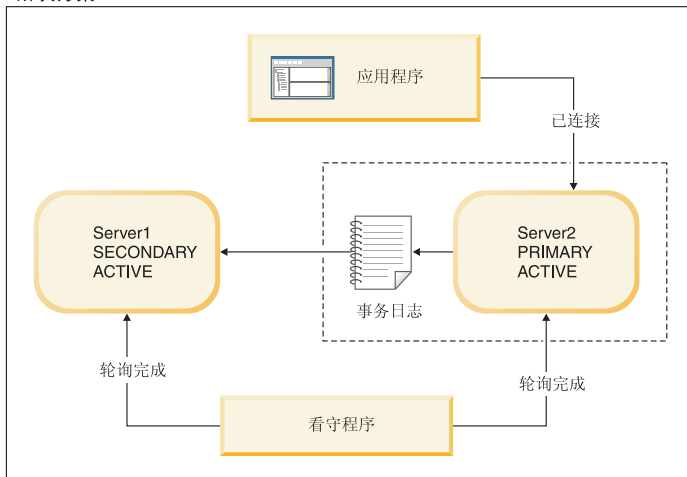
故障转移/补救措施

1



解决方案

2



3

图 21. 主服务器已关闭的情况和补救措施

1. 看守程序将指示 Server2 执行以下操作:

HSB SET PRIMARY ALONE

应用程序从 Server1 切换到 Server2。

HSB SET STANDALONE

2. 在修复 Server1 之后，Server1 将作为辅助服务器运行。

看守程序将指示 Server2 执行以下操作：

- HSB NETCOPY
- HSB CONNECT

3. 如果事务日志已填满，那么可能需要将 PRIMARY ALONE 服务器切换为 STAND-ALONE。在这种情况下，在重新连接服务器之前您还将需要执行 HSB NETCOPY。如果事务日志未填满，那么您必须跳过 NETCOPY 命令。

症状

应用程序无法连接至主服务器。此外，看守程序在主服务器上轮询失败。辅助服务器的 HSB 状态为 SECONDARY ALONE。

在主服务器已关闭的情况下如何恢复

关于此任务

为了允许“HotStandby”（辅助服务器）替换主服务器，请执行下列操作：

过程

1. 使用以下命令将新的主服务器设置为 PRIMARY ALONE 状态：

```
ADMIN COMMAND 'hotstandby set primary alone';
```

2. 将应用程序重新连接至新的主服务器。

3. 开始使用应用程序。

4. 修复旧的主服务器，并将它作为新的辅助服务器来启动。

5. 必要时，使用以下命令将数据库从新的主服务器复制到新的辅助服务器：

```
ADMIN COMMAND 'hotstandby netcopy';
```

有关详细信息，请阅读第 49 页的 3.4.5，『使主服务器与辅助服务器同步』。

6. 使用以下命令将新的主服务器重新连接至新的辅助服务器：

```
ADMIN COMMAND 'hotstandby connect';
```

F.2.2 辅助服务器已关闭

故障情况

与辅助服务器的所有连接都已断开。这可能是由于辅助服务器中发生故障，或者网络故障使得主服务器或者看守程序无法与辅助服务器通信而导致的。在本节中，认为辅助服务器已失败，但实际上问题可能与辅助服务器或网络有关。

补救措施

标准补救措施是将主服务器切换为 PRIMARY ALONE 状态。在辅助服务器再次启动之后，使它与主服务器同步。

当发现与辅助服务器的连接存在问题时，主服务器将执行下列操作：

1. 暂挂已打开的任何事务，既不落实它们，也不回滚它们（主服务器不会向客户机发送错误消息或者“成功”消息）；并且
2. 自动将它自己的状态从 PRIMARY ACTIVE 切换为 PRIMARY UNCERTAIN。

通常，在确定辅助服务器不可用之后，看守程序会将主服务器从 PRIMARY UNCERTAIN 状态切换为 PRIMARY ALONE 状态。在主服务器切换为 PRIMARY ALONE 状态之后，它可以继续接受事务并保持它们以便发送至辅助服务器。稍后，当辅助服务器再次运行时，可以向辅助服务器发送事务日志，以使它可与主服务器之间进行同步复制。

将主服务器设置为 PRIMARY ALONE 状态之后，主服务器将落实打开的事务。为了避免在辅助服务器尚未落实事务的情况下主服务器就落实事务这种可能性，会将事务保留在事务日志中，就好像从未将它们发送至辅助服务器一样。当辅助服务器重新运行并且开始进行同步复制时，主服务器将发送该事务日志，并且辅助服务器将检查每个事务。如果任何事务是重复的（即，在辅助服务器失败之前，辅助服务器已经落实了此事务），那么在辅助服务器上不会重新执行重复的事务。

看守程序或系统管理员在选择是否使主服务器处于 PRIMARY ALONE 状态或者选择替代操作时必须小心。如果看守程序或系统管理员选择的操作不是将主服务器切换为 PRIMARY ALONE 状态，那么必须考虑辅助服务器和主服务器可能不具有相同的数据，即，它们可能并不是都已回滚了事务。有可能失败的辅助服务器实际上已经落实了数据，但是在落实数据之后、在向主服务器发送确认信息之前崩溃了，而主服务器从未落实数据。在这种情况下，辅助服务器实际上可能比主服务器先落实数据，而不是在它之后落实数据。

同样，看守程序或管理员还必须小心，请不要允许两个服务器同时都进入 PRIMARY ALONE 状态。

下图分成三个框。第一个框显示了发生的情况，也就是主服务器和看守程序已经与辅助服务器失去联系。下一个框显示了如何让系统继续保持工作，直到彻底解决了问题为止。第三个框显示了解决问题之后（即，在已经修复了已断开的服务器之后，或者在恢复通信之后）的最终状态。

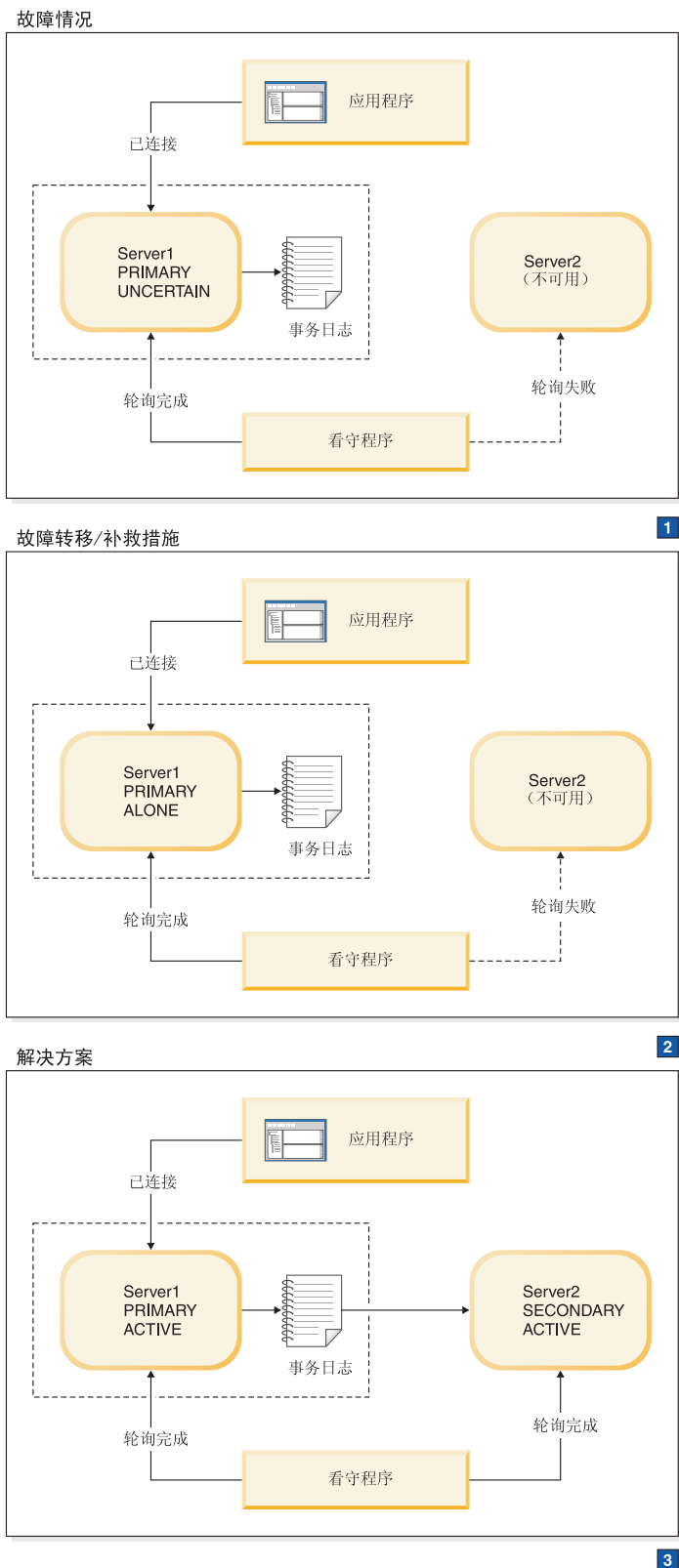


图 22. 辅助服务器已关闭的情况和补救措施

1. 看守程序将指示 Server1 执行以下操作:

```
HSB SET PRIMARY ALONE
```

HSB SET STANDALONE

- 在 Server2 重新开始运行之后，看守程序将指示 Server1 执行以下操作:

HSB NETCOPY

HSB CONNECT

- 如果事务日志已填满，那么可能需要将 PRIMARY ALONE 服务器切换为 STAND-ALONE。在这种情况下，在重新连接服务器之前您还将需要执行 HSB NETCOPY。如果事务日志未填满，那么您必须跳过 NETCOPY 命令。

症状

看守程序在辅助服务器上轮询失败。主服务器的状态为 PRIMARY ALONE 或 PRIMARY UNCERTAIN。

在辅助服务器已关闭的情况下如何恢复

关于此任务

为了允许主服务器继续接收事务并且独立于辅助服务器运行，请执行下列操作:

过程

- 如果主服务器处于 PRIMARY UNCERTAIN 状态，那么使用以下命令将主服务器设置为 PRIMARY ALONE 状态:

```
ADMIN COMMAND 'hotstandby set primary alone';
```
- 在已经修复并且重新启动辅助服务器以及/或者重新建立了辅助服务器的网络连接之后，请使用以下命令检查主服务器的状态:

```
ADMIN COMMAND 'hotstandby state';
```
- 如果主服务器的状态为 PRIMARY ALONE，那么请使用以下命令将主服务器重新连接至辅助服务器:

```
ADMIN COMMAND 'hotstandby connect';
```
- 如果主服务器的状态先前已更改为 STANDALONE，那么请执行下列操作:
 - 使用以下命令将数据库从新的主服务器复制到辅助服务器:

```
ADMIN COMMAND 'hotstandby netcopy';
```
 - 有关详细信息，请阅读第 49 页的 3.4.5，『使主服务器与辅助服务器同步』。
- 使用以下命令将主服务器重新连接至辅助服务器:

```
ADMIN COMMAND 'hotstandby connect';
```

当辅助服务器已关闭时的其他情况

如果应用程序从主服务器中接收到错误消息 10047 或 14537:

- 尝试连接至辅助服务器以检查它的状态是否已切换为新的主服务器。
- 如果它的状态不是 PRIMARY ACTIVE 或 PRIMARY ALONE，那么请参阅第 170 页的 F.2.1，『主服务器已关闭』中的情况。

F.2.3 看守程序已关闭

本节说明看守程序失败时发生的情况。

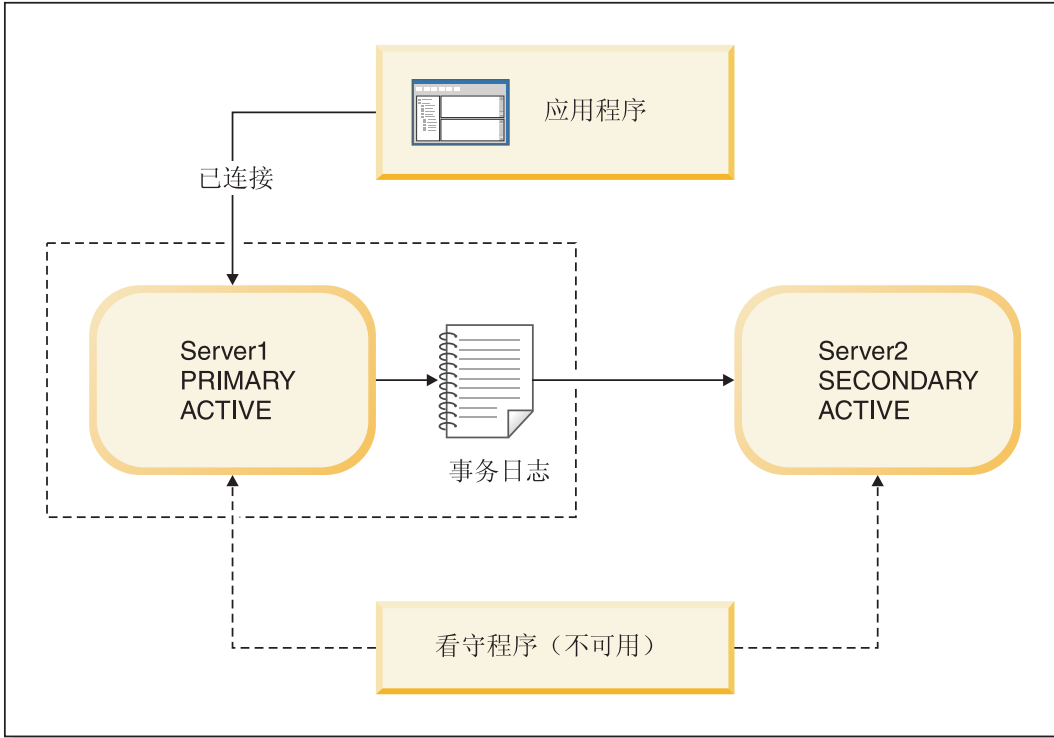
故障情况

与看守程序的所有连接都已断开。

补救措施

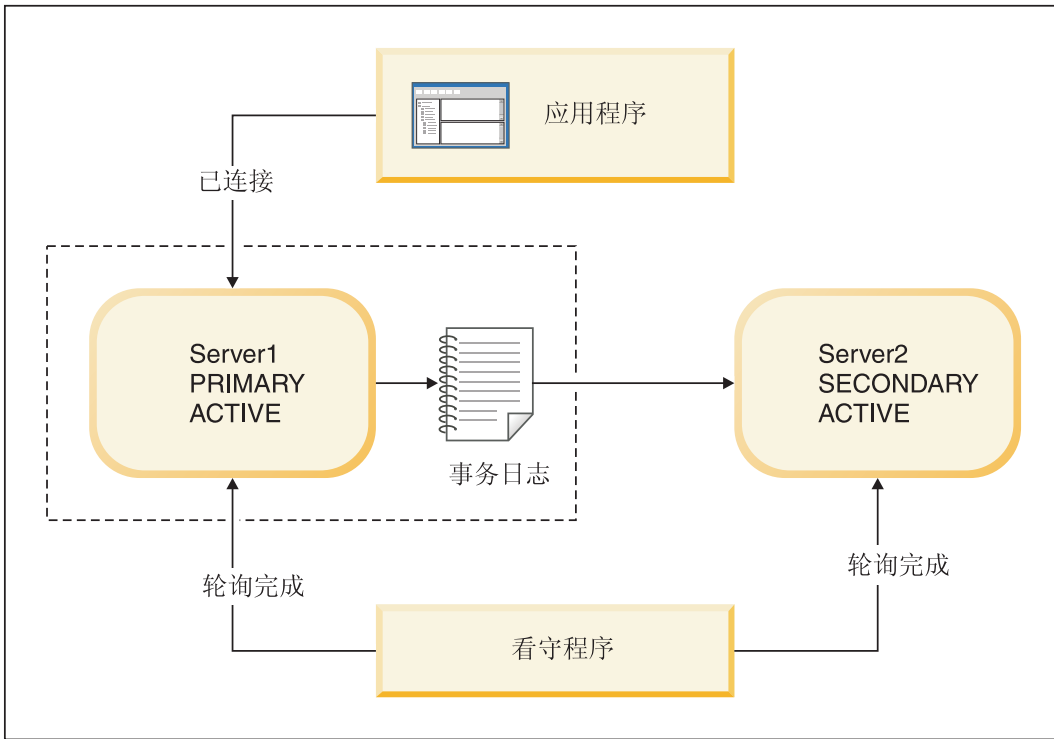
需要手动干预。当看守程序启动之后，务必要检查主服务器和辅助服务器以确认它们的状态。

故障情况



解决方案

1



1. 使看守程序重新开始运行或者修复网络。

图 23. 看守程序已关闭的情况和补救措施

症状

看守程序进程已关闭，或者看守程序与两个服务器的网络连接不可用。

其他情况

如果两个服务器已更改状态并且一个服务器不再工作，那么请参阅本节中的适当情况以获取指示信息。

在看守程序已关闭的情况下如何恢复

关于此任务

要从与看守程序的所有连接都已断开的情况下恢复：

过程

1. 允许主服务器和辅助服务器继续正常操作。
2. 一旦看守程序已启动，就让它通过以下命令检查每个服务器的状态：

```
ADMIN COMMAND 'hotstandby state';
```

F.2.4 主服务器和辅助服务器之间的通信链路已断开故障情况

主服务器与辅助服务器之间的连接已断开。

主服务器会将它本身切换为 **PRIMARY UNCERTAIN** 状态。（如果 **AutoPrimaryAlone** 设置为 **Yes**，那么服务器会将它本身切换为 **PRIMARY ALONE** 状态。）

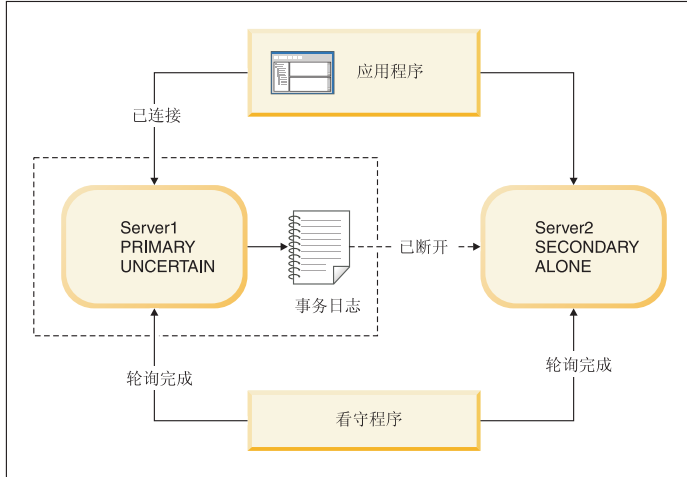
注：如果主服务器将落实消息发送至辅助服务器，然后检测辅助服务器的故障，那么主服务器将依赖看守程序或管理员来指示主服务器如何继续操作。这是因为在辅助服务器失败之前，主服务器将无法检测在辅助服务器中是否落实或回滚了事务。

在主服务器接收到来自看守程序或管理员的命令之前，它不再接受事务。在此阶段，为了让主服务器继续运行，看守程序或管理员可以将主服务器设置为 **PRIMARY ALONE** 状态。

补救措施

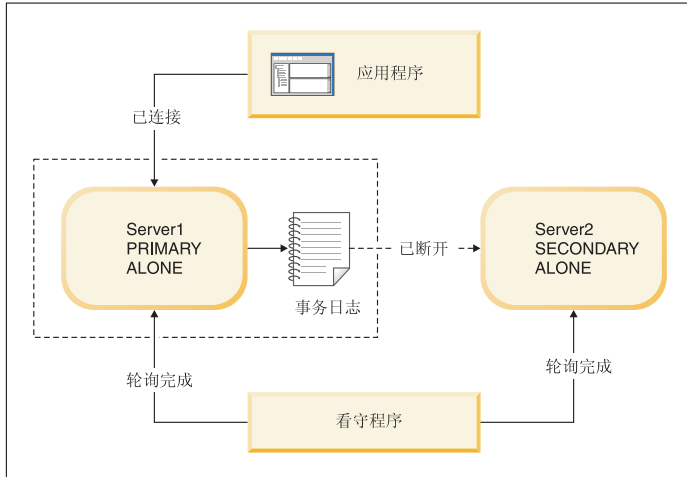
即使主服务器与辅助服务器之间的链路已断开，主服务器仍然可以继续运行。如果主服务器尚不处于 **PRIMARY ALONE** 状态，那么请将主服务器切换为 **PRIMARY ALONE** 状态。一旦恢复了主服务器与辅助服务器之间的链路，就使数据库同步。

故障情况



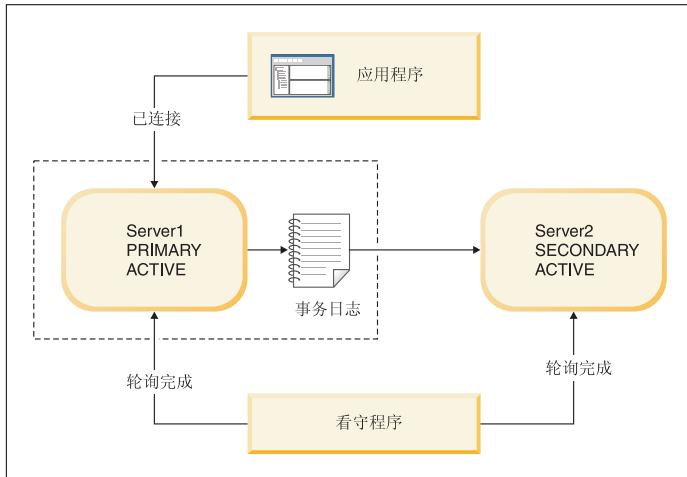
1

故障转移/补救措施



2

解决方案



3

图 24. 主服务器与辅助服务器之间的链路已断开的情况和补救措施

1. 看守程序将指示 Server1 执行以下操作:

HSB SET PRIMARY ALONE

HSB SET STANDALONE

2. 在修复主服务器与辅助服务器之间的连接之后，看守程序将指示 Server1 执行以下操作：

HSB NETCOPY

HSB CONNECT

3. 如果事务日志已填满，那么可能需要将 PRIMARY ALONE 服务器切换为 STAND-ALONE。在这种情况下，在重新连接服务器之前您还将需要执行 HSB NETCOPY。如果事务日志未填满，那么您必须跳过 NETCOPY 命令。

症状

主服务器未与辅助服务器连接，并且状态为 PRIMARY UNCERTAIN 或 PRIMARY ALONE。

当主服务器与辅助服务器之间的通信链路已断开的情况下如何恢复 关于此任务

要从主服务器与辅助服务器之间的连接已断开的情况下恢复：

过程

1. 修复主服务器与辅助服务器之间的网络连接。
2. 使用以下命令检查主服务器的状态：

```
ADMIN COMMAND 'hotstandby state';
```
3. 如果主服务器的状态为 PRIMARY ALONE，那么请使用以下命令将主服务器重新连接至辅助服务器：

```
ADMIN COMMAND 'hotstandby connect';
```
4. 如果主服务器的状态为 STANDALONE，那么请执行以下操作：
 - a. 将数据库从主服务器复制到辅助服务器。有关详细信息，请阅读第 49 页的 3.4.5，『使主服务器与辅助服务器同步』。

在使用 **ADMIN COMMAND 'hotstandby netcopy';** 命令之前，请确保辅助服务器已启动并且正在运行，并且已准备好接收 **netcopy** 命令。另外，务必将主服务器的状态设置为 PRIMARY ALONE。

- b. 使用以下命令将主服务器重新连接至辅助服务器：

```
ADMIN COMMAND 'hotstandby connect';
```

当主服务器与辅助服务器之间的通信链路已断开时的其他情况

如果应用程序从主服务器中接收到错误消息 10047 或 14537：

- 尝试连接至辅助服务器以检查它是否已切换为新的主服务器。
- 如果旧的辅助服务器未切换为新的主服务器，那么请参阅第 170 页的 F.2.1，『主服务器已关闭』中的情况。

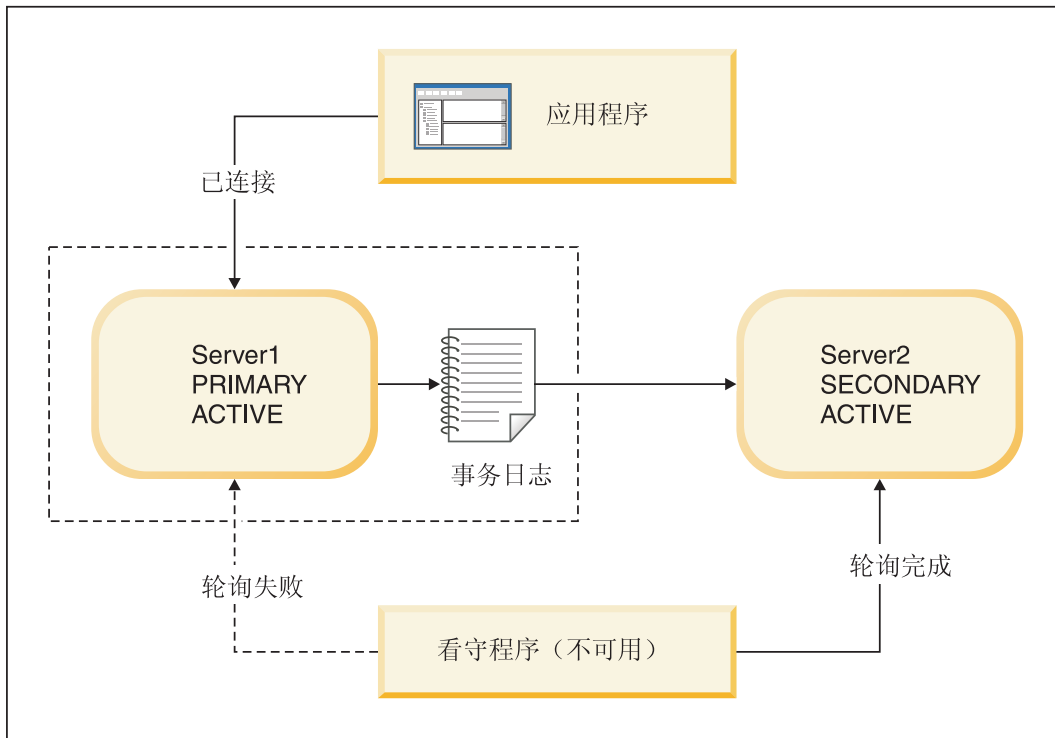
F.2.5 看守程序和主服务器之间的通信链路已断开故障情况

看守程序与主服务器之间的连接已断开。

补救措施

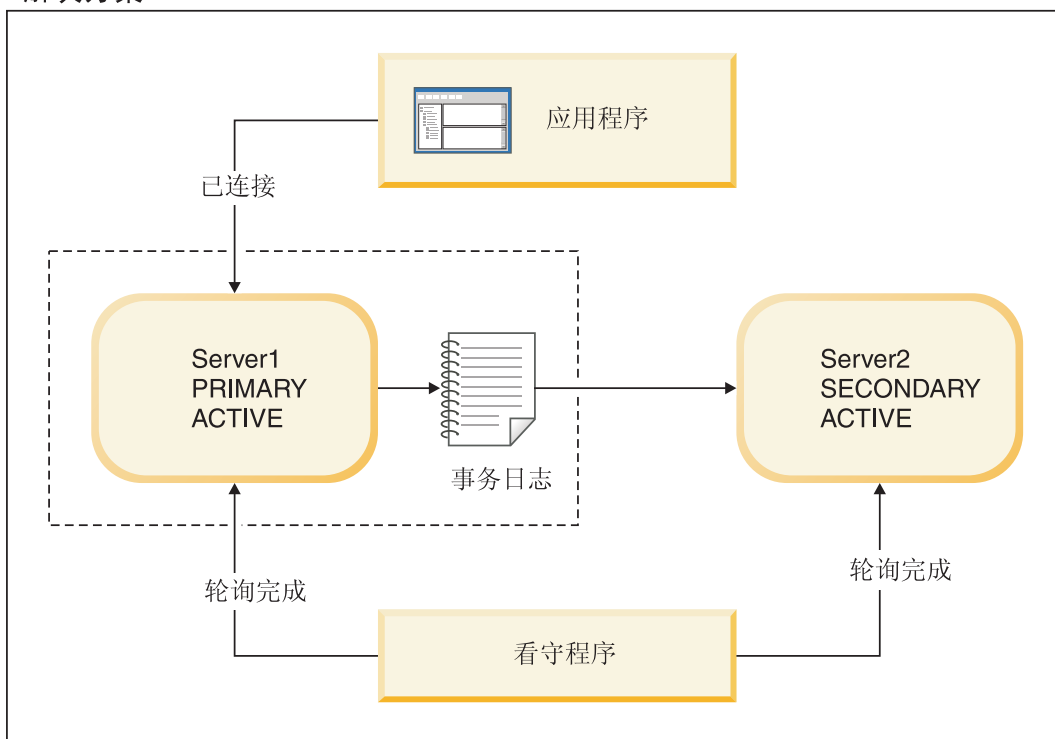
即使看守程序与主服务器之间的链路已断开，主服务器和辅助服务器仍然可以继续运行。修复了看守程序与主服务器之间的链路之后，务必检查主服务器和辅助服务器的状态。

故障情况



解决方案

1



1. 修复看守程序与 Server1 的网络连接。

图 25. 看守程序与主服务器之间的链路已断开的情况和补救措施

症状

看守程序在主服务器中轮询失败。但是，辅助服务器状态被报告为 `SECONDARY ACTIVE`。这意味着主服务器很可能处于正常状态，并且看守程序只是与主服务器失去联系而已。

其他情况

如果两个服务器的状态已更改并且一个服务器不再工作，那么请参阅本节中的适当情况以获取指示信息。

当看守程序与主服务器之间的通信链路已断开的情况下如何恢复

关于此任务

要从看守程序与主服务器之间的连接已断开的情况下恢复：

过程

1. 允许主服务器和辅助服务器继续正常操作。
2. 修复看守程序与主服务器之间的网络连接。
3. 一旦建立了网络连接之后，让看守程序使用以下命令来检查每个服务器的状态：

```
ADMIN COMMAND 'hotstandby state';
```

F.2.6 看守程序和辅助服务器之间的通信链路已断开

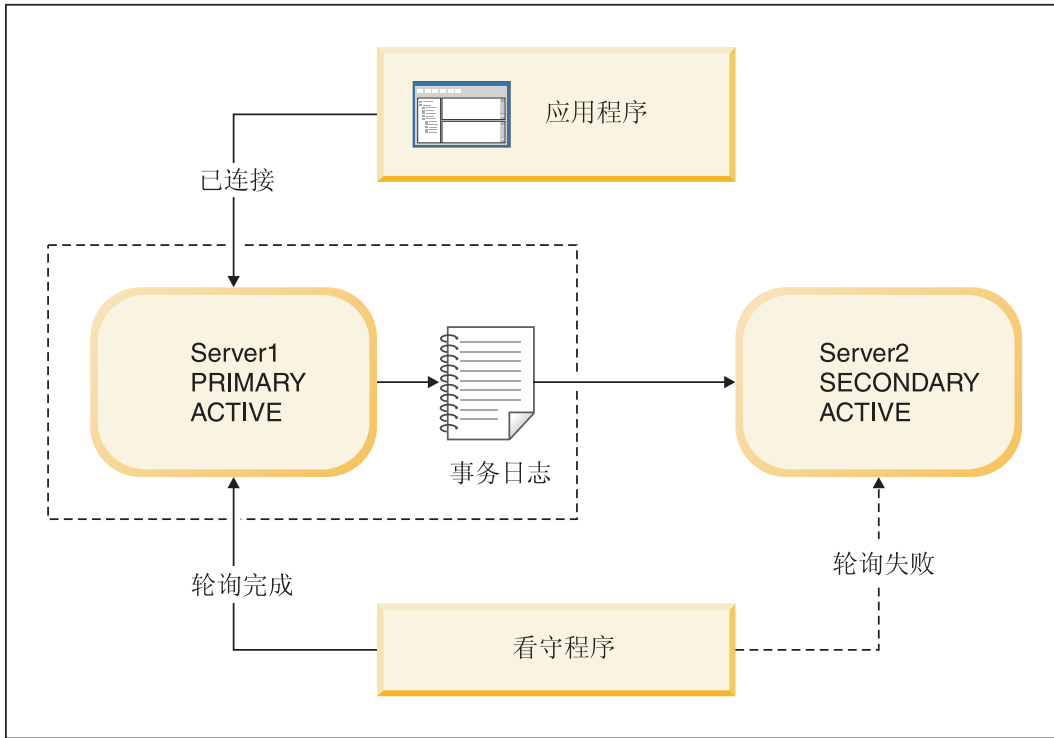
故障情况

看守程序与辅助服务器之间的连接已断开。

补救措施

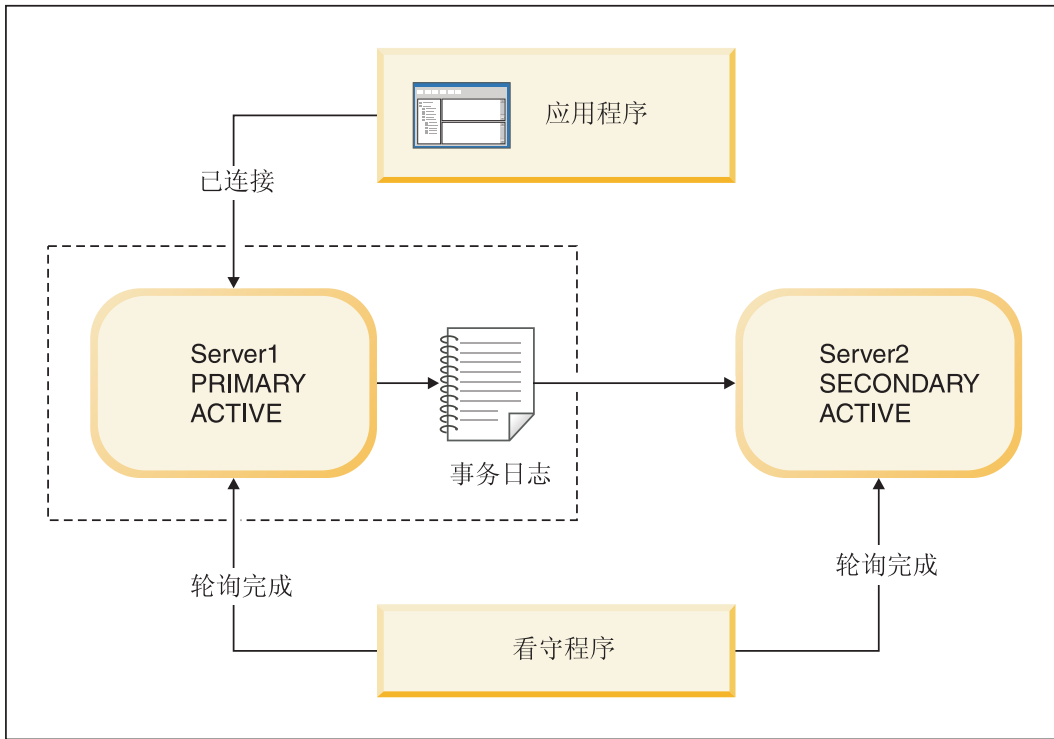
即使看守程序与辅助服务器之间的链路已断开，主服务器和辅助服务器仍然可以继续运行。修复了看守程序与辅助服务器之间的链路之后，务必要检查主服务器和辅助服务器以确认它们的状态。

故障情况



解决方案

1



1. 修复看守程序与 Server2 的网络连接。

图 26. 看守程序与辅助服务器之间的链路已断开的情况和补救措施

症状

看守程序在辅助服务器中轮询失败。

其他情况

如果两个服务器的状态已更改并且一个服务器不再工作，那么请参阅本节中的适当情况以获取指示信息。

当看守程序与辅助服务器之间的通信链路已断开的情况下如何恢复 关于此任务

要从看守程序与辅助服务器之间的连接已断开的情况下恢复：

过程

1. 允许主服务器和辅助服务器继续正常操作。
2. 修复看守程序与辅助服务器之间的网络连接。
3. 一旦建立了网络连接之后，让看守程序使用以下命令来检查每个服务器的状态：

```
ADMIN COMMAND 'hotstandby state';
```

F.2.7 看守程序与主服务器之间以及主服务器与辅助服务器之间的通信链路已断开

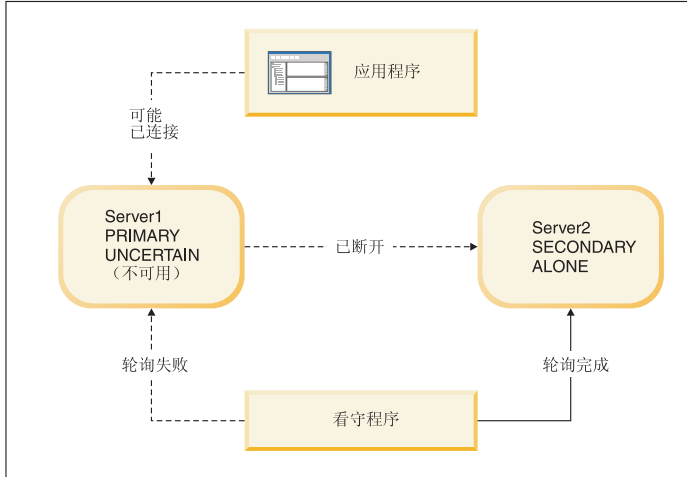
故障情况

看守程序与主服务器之间以及主服务器与辅助服务器之间的连接已断开。

补救措施

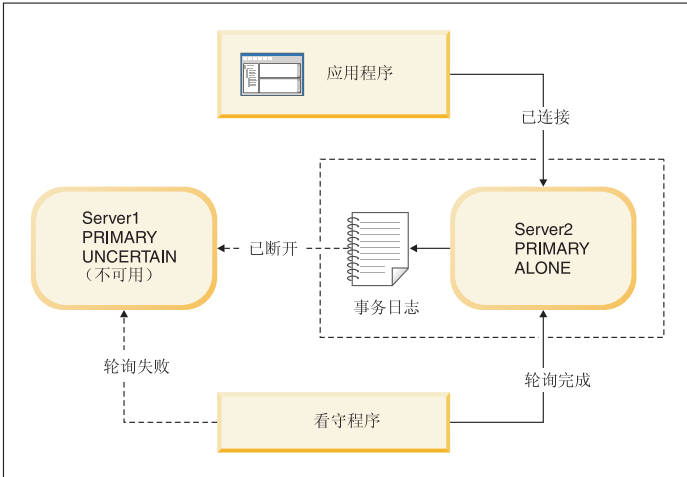
为了让看守程序继续监视主服务器，请将辅助服务器切换为新的主服务器并将此新的主服务器设置为 PRIMARY ALONE 状态。然后，设置新的辅助服务器并使它与主服务器同步。

故障情况



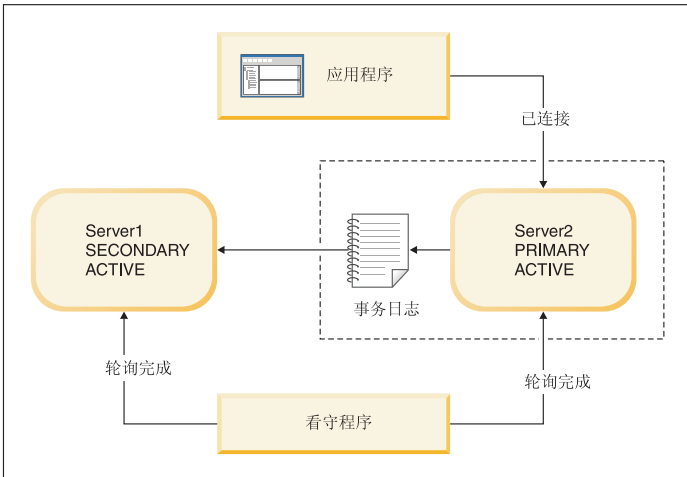
1

故障转移/补救措施



2

解决方案



3

图 27. 看守程序与主服务器之间以及主服务器与辅助服务器之间的通信链路已断开的情况和补救措施

1. Server1 的角色是 PRIMARY UNCERTAIN。但是，从看守程序的角度来看，Server1 是不可用而不是 PRIMARY UNCERTAIN。

看守程序将指示 Server2 执行以下操作:

```
HSB SET PRIMARY ALONE
```

应用程序从 Server1 切换到 Server2。

2. 两个服务器都认为它们自己是主服务器。如果程序或管理员（通过手动干预）将 Server1 从 PRIMARY UNCERTAIN 状态切换为 PRIMARY ALONE 状态，那么将有两个处于活动状态的主服务器，它们都可能在更新数据，将无法解决它们之间的差别。

在修复网络连接之后，看守程序将指示 Server1 执行以下操作:

```
HSB SWITCH SECONDARY
```

看守程序将指示 Server2 执行以下操作:

```
HSB NETCOPY
```

```
HSB CONNECT
```

3. 如果事务日志已填满，那么可能需要将 PRIMARY ALONE 服务器切换为 STAND-ALONE。在这种情况下，在重新连接服务器之前您还将需要执行 HSB NETCOPY。如果事务日志未填满，那么您必须跳过 NETCOPY 命令。

症状

看守程序在主服务器中轮询失败。辅助服务器与主服务器之间已经断开连接；因此，Server2 处于 SECONDARY ALONE 状态，如果可与主服务器联系，那么主服务器将报告其状态为 PRIMARY UNCERTAIN 或 PRIMARY ALONE。

这种情况开始假定应用程序可能已连接至旧的主服务器。但是，由于旧的主服务器处于 PRIMARY UNCERTAIN 状态，因此应用程序无法执行更新。请注意，连接至 Server1 的应用程序也有可能已经失去了它们的通信链路，因此不再知道旧的主服务器是否存在。

当看守程序与主服务器之间以及主服务器与辅助服务器之间的通信链路已断开的情况下如何恢复

要从看守程序与主服务器之间以及主服务器与辅助服务器之间的连接已断开的情况下恢复，请执行必需的步骤以使用热备用服务器（辅助服务器）来替换主服务器。

关于此任务

为了允许辅助服务器替换主服务器，请执行下列操作:

过程

1. 如果旧的主服务器处于 PRIMARY UNCERTAIN 状态或者与应用程序和辅助服务器断开连接，那么使用以下命令将辅助服务器设置为 PRIMARY ALONE 状态:

```
ADMIN COMMAND 'hotstandby set primary alone';
```
2. 将应用程序重新连接至新的主服务器。
3. 修复与旧的主服务器的网络连接或者已断开的连接。
4. 检查服务器状态。两个服务器现在必须正在运行。

5. 如果新的主服务器处于 STANDALONE 状态（例如，由于在修复连接期间新的主服务器的事务日志已填满）：
 - a. 使用以下命令将新的主服务器设置为 PRIMARY ALONE 状态：

```
ADMIN COMMAND 'hotstandby set primary alone';
```
 - b. 将数据库从新的主服务器复制到新的辅助服务器。有关详细信息，请阅读第 49 页的 3.4.5，『使主服务器与辅助服务器同步』。
6. 如果新的主服务器处于 PRIMARY ALONE 状态：
 - a. 使用以下命令将旧的主服务器切换到新的辅助服务器：

```
ADMIN COMMAND 'hotstandby switch secondary';
```
7. 使用以下命令将新的主服务器重新连接至新的辅助服务器：

```
ADMIN COMMAND 'hotstandby connect';
```

看守程序与主服务器之间以及主服务器与辅助服务器之间的通信链路已断开的其他情况

如果应用程序从新的主服务器中接收到错误消息 10047 或 14537:

- 尝试连接至旧的辅助服务器以检查它是否已切换为新的主服务器。
- 如果旧的辅助服务器未切换为新的主服务器，那么请对处于 PRIMARY ALONE 状态的原来的主服务器重新执行事务。

F.2.8 看守程序与辅助服务器之间以及主服务器与辅助服务器之间的通信链路已断开

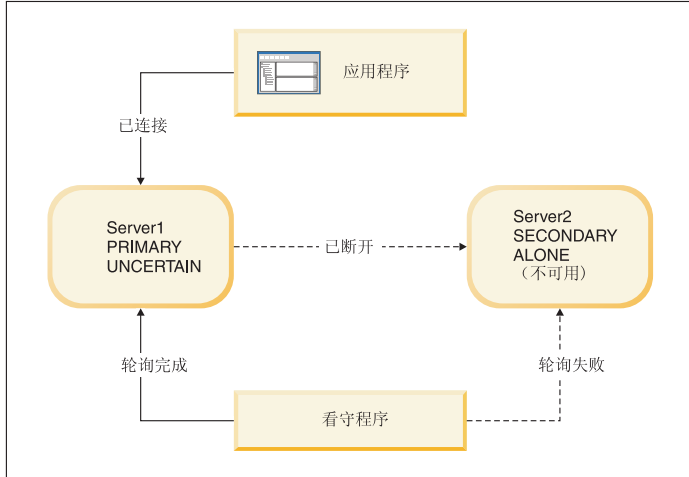
故障情况

看守程序与辅助服务器之间的连接以及主服务器与辅助服务器之间的连接已断开。

补救措施

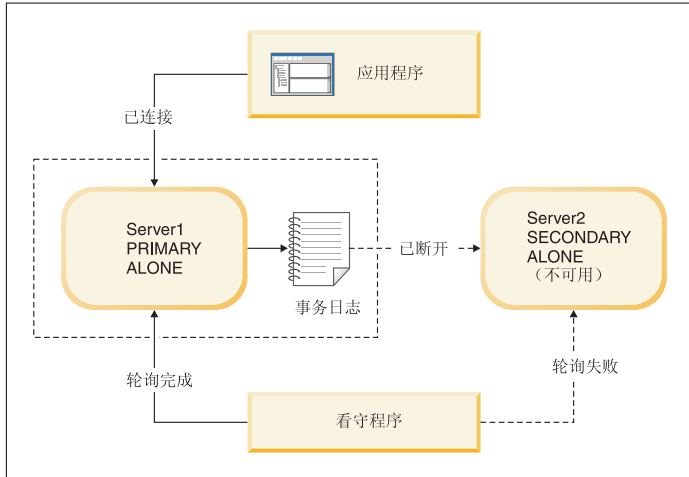
即使主服务器与辅助服务器和看守程序之间的链路已断开，主服务器仍然可以继续运行。如果主服务器尚未处于 PRIMARY ALONE 状态，那么请将它切换为此状态。然后，在辅助服务器再次启动之后，使它与主服务器同步。

故障情况



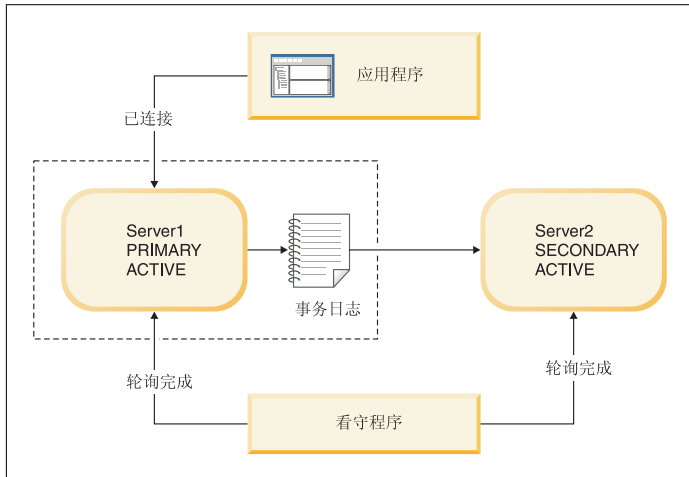
1

故障转移/补救措施



2

解决方案



3

图 28. 看守程序与辅助服务器之间以及主服务器与辅助服务器之间的通信链路已断开的情况和补救措施

1. Server2 看到它自己的角色为 SECONDARY ALONE，但是看守程序看不到 Server2，因为它认为 Server2 不可用。

看守程序将指示 Server1 执行以下操作:

```
HSB SET PRIMARY ALONE
```

2. 在修复与辅助服务器的连接之后, 看守程序将指示 Server1 执行以下操作:

```
HSB NETCOPY
```

```
HSB CONNECT
```

3. 如果事务日志已填满, 那么可能需要将 PRIMARY ALONE 服务器切换为 STAND-ALONE。在这种情况下, 在重新连接服务器之前您还将需要执行 HSB NETCOPY。如果事务日志未填满, 那么您必须跳过 NETCOPY 命令。

症状

看守程序在辅助服务器中轮询失败; 主服务器未与辅助服务器连接, 并且已切换为 PRIMARY UNCERTAIN 或 PRIMARY ALONE 状态。

当看守程序与辅助服务器之间以及主服务器与辅助服务器之间的通信链路已断开的情况下如何恢复

关于此任务

要从看守程序与辅助服务器之间的连接以及主服务器与辅助服务器之间的连接已断开的情况下恢复:

过程

1. 尝试修复连接。
2. 在修复连接之后, 使用 **ADMIN COMMAND 'hotstandby state'** 命令检查主服务器的状态。
3. 如果主服务器的状态为 STANDALONE:
 - a. 确保两个服务器都在运行。
 - b. 使用以下命令将主服务器的状态设置为 PRIMARY ALONE:

```
ADMIN COMMAND 'hotstandby set primary alone';
```
 - c. 使用以下命令将数据库从主服务器复制到辅助服务器:

```
ADMIN COMMAND 'hotstandby netcopy';
```

有关详细信息, 请阅读第 49 页的 3.4.5, 『使主服务器与辅助服务器同步』。

4. 使用以下命令将主服务器重新连接至辅助服务器:

```
ADMIN COMMAND 'hotstandby connect';
```

看守程序与辅助服务器之间以及主服务器与辅助服务器之间的通信链路已断开的其他情况

如果应用程序从主服务器中接收到错误消息 10047 或 14537:

- 尝试连接至辅助服务器以检查它是否已切换为主服务器。
- 如果辅助服务器未切换为新的主服务器, 那么请对处于 PRIMARY ALONE 状态的原来的主服务器重新执行事务。

F.3 solid.ini 配置文件的 [Watchdog] 节

看守程序的 solid.ini 文件包含 [Watchdog] 配置节以指定特定于看守程序的参数。

要点: solid.ini 文件的 [WatchDog] 节中的参数并非都是由 solidDB 预定义的。根据您是如何编写看守程序以及您是否希望它从 solid.ini 文件中读取参数信息，可以将此处定义的参数与您已经定义的参数混合使用。您也可以忽略这些参数。此处所显示的参数适用于 solidDB 提供的用 C 语言编写的样本看守程序。

表 41. 看守程序参数

[Watchdog]	描述	出厂值
AutoSwitch	<p>如果 AutoSwitch 参数设置为 yes，那么看守程序将自动执行下列操作：</p> <ol style="list-style-type: none"> 1. 如果辅助服务器失败，那么看守程序会让主服务器切换为 PRIMARY ALONE 状态（而不是保持 PRIMARY UNCERTAIN 状态）。 2. 如果主服务器失败，那么看守程序将自动发送下列命令： <pre>'hsb switch primary' 'hsb set primary alone'</pre> <p>以将原来的辅助服务器切换为新的主服务器。</p> <p>例如： <pre>[Watchdog] AutoSwitch = NO</pre> <p>此参数是可选的。</p> </p> 	Yes
Connect1	<p>[Watchdog] 节中的 Connect1 参数使看守程序能够连接至主服务器或辅助服务器。这是一个必需参数，用于定义 Connect1 服务器的协议和网络地址。</p> <p>例如： <pre>connect1 = tcp primarymachine 1313</pre> </p>	None
Connect2	<p>[Watchdog] 节中的 Connect2 参数使看守程序能够连接至主服务器或辅助服务器。这是一个必需参数，用于定义 Connect2 服务器的协议和网络地址。</p> <p>例如： <pre>connect2 = tcp secondarymachine 1313</pre> </p>	None
DualSecAutoSwitch	<p>如果 DualSecAutoSwitch = Yes 并且两个服务器都是辅助服务器，那么看守程序将自动选择其中一个辅助服务器作为新的主服务器并将它切换为主服务器。如果 DualSecAutoSwitch = No，那么系统管理员必须将一个服务器切换为主服务器。请注意，无论看守程序处于“正常”方式还是“故障”方式，DualSecAutoSwitch 参数都适用。</p>	Yes

表 41. 看守程序参数 (续)

[Watchdog]	描述	出厂值
NumRetry	<p>[Watchdog] 节中的 NumRetry 参数允许您指定在认为连接尝试发生响应故障或错误之前看守程序可以尝试连接至辅助服务器或主服务器的次数。</p> <p>例如:</p> <pre>[Watchdog] NumRetry = 3</pre> <p>重试次数就是除初始尝试之外的尝试次数。如果重试次数设置为 3, 那么总的尝试次数就是 4。请注意, 重试是立即进行的。当发生故障时, 看守程序不会等待两次重试之间的时间间隔 (例如, PingTimeout) 这段时间。</p> <p>此参数是可选的。</p>	0
Password1 Password2	请参阅下面对 Username1 和 Username2 参数的描述。	无出厂值。
Pessimistic	<p>将此参数设置为 Yes 可以提高看守程序的反应速度。</p> <p>当 Pessimistic = No 时, 看守程序会检查它与服务器之间的连接, 但是要等到其中一个服务器检测到有一个问题并更改其状态 (例如, 更改为 PRIMARY UNCERTAIN) 之后, 看守程序才会真正执行操作 (例如, 将服务器的状态更改为 PRIMARY ALONE)。</p> <p>当 Pessimistic = Yes 时, 一旦看守程序与其中一个服务器失去联系, 它就会立即执行操作; 看守程序不会等到另一个服务器更改状态才执行操作。这可以提高反应速度, 但是也可能会提高假报警 (例如, 由于发生网络问题而发出的报警) 的可能性。</p> <p>当 Pessimistic = Yes 时, 看守程序将按如下所示作出反应: 如果看守程序已经与主服务器失去联系, 那么看守程序会将辅助服务器切换为主服务器; 如果看守程序已经与辅助服务器失去联系, 那么它就会将主服务器设置为 PRIMARY ALONE。</p> <p>注意: 设置 Pessimistic = Yes 可能会导致额外的切换甚至导致双重主服务器。除非网络比服务器更不可能失败, 否则不应将此参数设置为 Yes。</p> <p>还可以使用可选的命令行开关“-p”来打开悲观行为。</p>	No
PingInterval	<p>[Watchdog] 节中的 PingInterval 参数允许您指定两次以正常的看守程序方式查询状态连接信息之间的时间间隔 (按毫秒计)。要检测服务器故障, 每隔 PingInterval 毫秒之后, 看守程序就会对主服务器和辅助服务器都发送 hotstandby status connect 命令。</p> <p>例如:</p> <pre>[Watchdog] PingInterval = 5000</pre> <p>此参数是可选的。</p> <p>请注意, 看守程序的 PingInterval 参数不同于服务器的 PingTimeout 参数。</p> <p>注意: 先前的样本看守程序要求按秒 (而不是毫秒) 指定 PingInterval。如果您要使用更旧的 solid.ini 文件, 那么应对它进行更新。</p>	1000 (1 秒)

表 41. 看守程序参数 (续)

[Watchdog]	描述	出厂值
<p>Username1 Username2</p>	<p>[Watchdog] 节中的 Username 和 Password 参数是可选的。这两个参数指定有权使用 connect1 服务器的用户名和密码。</p> <p>例如:</p> <pre>[Watchdog] Username1 = Tom Password1 = dr17xy Username2 = Jerry Password2 = M89tvt</pre> <p>如果由于安全原因而在 solid.ini 配置文件中未指定这些参数, 那么看守程序启动时将提示您输入用户名和密码。</p>	<p>无出厂值。</p>
<p>WatchdogLog</p>	<p>[Watchdog] 节中的 WatchdogLog 参数允许您指定看守程序日志的文件名。将在当前工作目录中创建看守程序日志。它用来记录看守程序消息, 提醒管理员需要发出看守程序命令。</p> <p>例如:</p> <pre>[Watchdog] WatchdogLog = Watchdog.log</pre> <p>请注意, 只要文件名中不包含特殊字符 (例如, 空白或某些标点符号), 就不需要用引号将文件名引起来。</p> <p>此参数是可选的。</p>	<p>Watchdog.log</p>

当使用参数

[Logging]
DurabilityLevel

时, **DurabilityLevel** 参数值将只影响主服务器。辅助服务器的记录方式由 [HotStandby] 节中的 **2SafeAckPolicy** 参数指定。

索引

[B]

备份 4
 侦听方式, netcopy 54

[C]

参数

 高可用性控制器

 CheckInterval 121
 CheckTimeout 121
 Connect [LocalDB] 121
 Connect [RemoteDB] 121
 DBPassword 121
 DBUsername 121
 EnableAutoNetcopy 121
 EnableDBProcessControl 121
 EREIP 121
 Listen 121
 Password 121
 PreferredPrimary 121
 RequiredConnectFailures 121
 RequiredPingFailures 121
 StartInAutomaticMode 121
 StartScript 121
 Username 121

 AutoPrimaryAlone 40, 45, 47

 AutoSwitch 191

 BackupDeleteLog 4

 CatchupSpeedRate 64

 CatchupStepsToSkip 64

 CheckInterval 121

 CheckpointDeleteLog 4

 CheckTimeout 121

 Connect 37, 62

 Connect [LocalDB] 121

 Connect [RemoteDB] 121

 Connect1 191

 Connect2 191

 ConnectTimeout 38, 39

 CopyDirectory 40

 DBPassword 121

 DBUsername 121

 DualSecAutoSwitch 191

 DurabilityLevel 39

 EnableAutoNetcopy 121

 EnableDBProcessControl 121

 EREIP 121

 Header_text 125

 HSBEnabled 37, 62

 LogEnabled 37

参数 (续)

 NumRetry 191

 Password 121

 Password1 191

 Password2 191

 Pessimistic 191

 PingInterval 38, 191

 PingTimeout 38

 PreferredPrimary 121

 ReadMostlyLoadPercentAtPrimary 86

 RequiredConnectFailures 121

 RequiredPingFailures 121

 Server1_host 125

 Server1_name 125

 Server1_pass 125

 Server1_port 125

 Server1_user 125

 Server2_host 125

 Server2_name 125

 Server2_pass 125

 Server2_port 125

 Server2_user 125

 StartInAutomaticMode 121

 StartScript 121

 Username 121

 Username1 191

 Username2 191

 WatchdogLog 191

 Window_title 125

查看当前的连接设置 57

创建

 辅助数据库 54

存储方式 115

[D]

当前值 36

等号 35

[F]

访问方式

 RO (只读) 115

 RW (读写) 115

 RW/Create 115

 RW/Startup 115

分区

 网络 64

服务器

 连接 57

服务器状态

- 切换服务器状态 44, 45
- 验证 60
- OFFLINE 8
- PRIMARY ACTIVE 8
- PRIMARY ALONE 8, 47
- PRIMARY UNCERTAIN 8
- SECONDARY ACTIVE 8
- SECONDARY ALONE 8
- STANDALONE 8

辅助服务器

- 恢复联机状态 48

负载均衡

- 动态控制 87
- 方法
 - PREFERRED_ACCESS=LOCAL_READ 86
 - PREFERRED_ACCESS=READ_MOSTLY 86
 - PREFERRED_ACCESS=WRITE_MOSTLY 86
- 透明连接 86

复制

- 从主服务器复制到本地辅助服务器 56
- 数据库的内容 54, 55, 56
- 通过网络将主数据库复制到辅助服务器 53
- 验证过程 53, 56

[G]

高可用性管理器

- 定义 24
- 配置 42
- 屏幕快照 24

高可用性控制器 43

- 命令 43
- 配置 30, 42
- 启动 30
- 日志记录 24
- 设置 30
- 停止 30
- 样本 33
- 原则 43
- solidhac.ini 30

故障透明性

- 选择连接类型
 - CONNECTION 84
 - NONE 84
 - SESSION 84

关闭

- HotStandby 48

管理

- HotStandby 59
 - 切换服务器状态 44
 - 状态信息 58

[H]

恢复

- 维护 44

[J]

基本连接 89

- 检查点 4

[K]

看守程序样本 33

可重复读

- 事务隔离级别 64

可序列化

- 事务隔离级别 64

[L]

连接

- 基本 89
- 透明故障转移 72
- 选择连接类型 72

落实读

- 事务隔离级别 64

[N]

内存表

- HotStandby 64

[P]

配置

- HotStandby
 - netcopy 性能 63

[Q]

启动顺序 31

切换

- 连接状态信息 59
- 切换状态信息 58

[R]

日志记录

- 高可用性控制器 24

[S]

升级

- 冷迁移 107

升级 (续)

冷迁移和热迁移 107

热迁移 107, 110

使主服务器与辅助服务器同步 49

事务

隔离级别

内存表 64

logs

用尽空间 65

数据库

复制内容 54, 55, 56

内存表 64

验证复制过程 56

双重主服务器 64

[T]

同步复制 49

透明连接 72

[W]

外部引用实体

描述 (description) 22

配置 121

网络分区 64

双重主服务器 64

[X]

显示

连接状态信息 59

切换状态信息 58

通信信息 59

[Y]

验证

复制过程 56

连接状态信息 92

样本

高可用性控制器 33

看守程序 33

应用程序开发

HotStandby

基本连接 89

切换到新的主服务器 92

用于事务日志的空间 65

运行处于 PRIMARY ALONE 状态的服务器 47

[Z]

执行恢复和维护 44

状态 58

状态 (续)

列表 59, 80, 84

显示连接状态信息 59

显示切换状态信息 58

显示通信信息 59

验证服务器状态 60

HotStandby 58

OFFLINE 155

PRIMARY ACTIVE 155

PRIMARY ALONE 155

PRIMARY UNCERTAIN 155

STANDALONE 48, 62, 155

[数字]

1SafeMaxDelay (参数) 116

2SafeAckPolicy (参数) 116

A

ADMIN COMMAND 'hotstandby cominfo'

查看连接设置 57

ADMIN COMMAND 'hotstandby connect'

连接 HotStandby 服务器 57

ADMIN COMMAND 'hotstandby copy'

复制数据库的内容 56

ADMIN COMMAND 'hotstandby netcopy'

复制数据库的内容 54, 55

ADMIN COMMAND 'hotstandby set primary alone'

运行处于 PRIMARY ALONE 状态的服务器 47

ADMIN COMMAND 'hotstandby set standalone'

停止 HotStandby 操作 48

ADMIN COMMAND 'hotstandby state'

验证服务器状态 60

ADMIN COMMAND 'hotstandby status connect'

显示连接状态信息 59

ADMIN COMMAND 'hotstandby status copy'

验证复制过程 56

ADMIN COMMAND 'hotstandby status'

查询 HotStandby 状态 58

ADMIN COMMAND 'hotstandby switch primary'

切换服务器状态 45

ADMIN COMMAND 'hotstandby switch secondary'

切换服务器状态 45

ApplicationConnTestConnect (HAC 参数) 121

ApplicationConnTestInterval (HAC 参数) 121

ApplicationConnTestPassword (HAC 参数) 121

ApplicationConnTestTimeout (HAC 参数) 121

ApplicationConnTestUsername (HAC 参数) 121

autoconvert

命令行选项 107, 110

AutoPrimaryAlone (参数) 40, 47, 116, 168

“hotstandby switch”命令 45

AutoSwitch (参数) 191

B

BackupDeleteLog (参数) 4

C

CatchupSpeedRate (参数) 64, 116

CatchupStepsToSkip (参数) 64

CheckInterval (参数) 121

CheckpointDeleteLog (参数) 4

CheckTimeout (参数) 121

ClientReadTimeout (参数) 119

CLUSTER 72

Connect (参数) 37, 62, 116, 120

Connect [LocalDB] (参数) 121

Connect [RemoteDB] (参数) 121

Connect1 (参数) 191

Connect2 (参数) 191

ConnectTimeOut (参数) 38, 39, 116, 120

CopyDirectory (参数) 40, 116

D

DBPassword (参数) 121

DBUsername (参数) 121

DualSecAutoSwitch (参数) 191

DurabilityLevel (参数) 39

E

EnableApplicationConnTest (HAC 参数) 121

EnableAutoNetcopy (参数) 121

EnableDBProcessControl (参数) 121

EnableUnresponsiveActions (HAC 参数) 121

ERE (外部引用实体) 22

EREIP (参数) 121

G

GUI

高可用性管理器 24

H

HA 管理器参数

Header_text 125

Server1_host 125

Server1_name 125

Server1_pass 125

Server1_port 125

Server1_user 125

Server2_host 125

Server2_name 125

Server2_pass 125

Server2_port 125

HA 管理器参数 (续)

Server2_user 125

Window_title 125

HAC 故障情况

辅助节点失败 103

辅助数据库失败 101

无响应的服务器 104

主节点失败 102

主数据库失败 101

HotStandby 链路失败 103

HAManager.ini 115

Header_text (参数) 125

HotStandby

关闭 62

管理 35

快速启动 27

配置 27, 35, 42

应用程序与服务器之间的超时 94

设置 27

停止操作 48

状态

检查 58

events

SYS_EVENT_HSBCONNECTSTATUS 163

SYS_EVENT_HSBSTATESWITCH 163

SYS_EVENT_NETCOPYEND 163

SYS_EVENT_NETCOPYREQ 163

HAC

快速启动 28

配置 28

设置 28

HotStandby 故障处理 101

hotstandby copy (ADMIN COMMAND) 155

hotstandby netcopy (ADMIN COMMAND) 155

HOTSTANDBY_CONNECTSTATUS (SQL 函数) 59, 92

HOTSTANDBY_STATE (SQL 函数) 92

hsb status ADMIN COMMAND

复制 145

连接 145

切换 145

同步复制 145

HSBEnabled (参数) 37, 62, 116

L

Listen (参数) 121

LogEnabled (参数) 37

logpos ADMIN COMMAND 145

logpos ADMIN COMMAND hotstandby 61

M

MaxLogSize (参数) 116

MaxMemLogSize (参数) 116

N

netcopy 155
 侦听方式 54
 调谐 63
 调整性能 63
 主服务器必须处于 PRIMARY ALONE 状态 11
 ADMIN COMMAND 'hotstandby netcopy' 54
NetcopyErrorLevel (HAC 参数) 121
NetcopyRpcTimeout (参数) 116
NetcopyWarningLevel (HAC 参数) 121
NumRetry (参数) 191

O

OFFLINE (状态) 155

P

Password (参数) 121
Password1 (参数) 191
Password2 (参数) 191
Pessimistic (参数) 191
ping 38
PingInterval (参数) 38, 116, 191
PingTimeout (参数) 38, 116
PreferredPrimary (参数) 121
PRIMARY ACTIVE (状态) 155
PRIMARY ALONE (状态) 47, 155
PRIMARY UNCERTAIN (状态) 155
PrimaryAlone (参数) 116

R

ReadMostlyLoadPercentAtPrimary (参数) 86, 116
RequiredAppConnTestFailures (HAC 参数) 121
RequiredConnectFailures (参数) 121
RequiredPingFailures (参数) 121
RO
 访问方式 115
RW
 访问方式 115
RW/Create
 访问方式 115
RW/Startup
 访问方式 115

S

Server1_host (参数) 125
Server1_name (参数) 125
Server1_pass (参数) 125
Server1_port (参数) 125
Server1_user (参数) 125
Server2_host (参数) 125

Server2_name (参数) 125
Server2_pass (参数) 125
Server2_port (参数) 125
Server2_user (参数) 125
SET TRANSACTION WRITE 87
SET WRITE 87
SocketLinger (参数) 120
SocketLingerTime (参数) 120
solidhac.ini 30, 115
SQL 函数
 HOTSTANDBY_CONNECTSTATUS 59, 92
 HOTSTANDBY_STATE 92
STANDALONE (状态) 48, 62, 155
StartInAutomaticMode (参数) 121
StartScript (参数) 121
SYS_EVENT_HSBCONNECTSTATUS (事件) 163
SYS_EVENT_HSBSTATESWITCH (事件) 163
SYS_EVENT_NETCOPYEND (事件) 163
SYS_EVENT_NETCOPYREQ (事件) 163

T

TC 连接 72
TC Info 72
 处理矛盾 89
 属性组合 79
 语法 72
 JDBC 语法 77
TF 连接 72
Trace (参数) 120
TraceFile (参数) 120

U

UnresponsiveActionScript (HAC 参数) 121
Username (参数) 121
Username1 (参数) 191
Username2 (参数) 191

W

WatchdogLog (参数) 191
Window_title (参数) 125

[特别字符]

-x autoconvert (命令行选项) 107, 110
-x backupserver (命令行选项) 54
= (等于)
 设置参数值时使用等号 35

声明

© Copyright Oy IBM Finland Ab 1993, 2013.

All rights reserved.

除非经过 IBM 书面明确地授权，否则不能以任何方式使用本产品中的任何部分。

本产品受美国专利

6144941、7136912、6970876、7139775、6978396、7266702、7406489、7502796 和 7587429 保护。

为此产品指定的美国出口管制分类编号是 ECCN=5D992b。

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节 (DBCS) 信息的许可证查询，请联系您所在国家的 IBM 知识产权部门，或将查询以书面的形式发送至：

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：INTERNATIONAL BUSINESS MACHINES CORPORATION“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及 (ii) 允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

此信息仅用于规划目的。在所描述产品可用之前，此处的信息有可能更改。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可：

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口 (API) 进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例尚未在所有条件下经过全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。样本程序“按原样”提供，而没有任何类型的保证。对于因使用样本程序所引起的任何损害，IBM 概不负责。

凡这些实例程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明：

©（贵公司的名称）（年）。此部分代码是根据 IBM Corp. 公司的样本程序衍生出来的。

© Copyright IBM Corp.（输入年份）。All rights reserved.

如果您要查看此信息软拷贝，那么可能不显示相片和颜色说明。

商标

IBM、IBM 徽标、ibm.com[®]、Solid[®]、solidDB、InfoSphere[®]、DB2[®]、Informix[®] 和 WebSphere[®] 是 International Business Machines Corp. 在全球许多管辖区域内注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。当前的 IBM 商标列表，可从 Web 站点 www.ibm.com/legal/copytrade.shtml 上“版权和商标信息”部分获取。

Java[™] 和所有基于 Java 的商标和徽标是 Oracle 和/或其附属机构的商标或注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的注册商标。

Microsoft 和 Windows 是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

其他产品和服务名称可能是 IBM 或其他公司的商标。



S151-1689-01

