

目录

目录	1
术语	3
综合监控实时数据系统集群方案	4
1. 综述	5
1.1. 概念	5
1.2. 目的	5
1.3. 分类	6
2. 综合监控实时数据系统需求	错误!未定义书签。
2.1. 系统需求	错误!未定义书签。
2.1.1. 数据需求	错误!未定义书签。
2.1.2. 接口需求	错误!未定义书签。
2.2. 系统模型	错误!未定义书签。
2.2.1. FEP TCP Server	错误!未定义书签。
2.2.2. HMI TCP Server	错误!未定义书签。
2.2.3. RTD TCP Server	错误!未定义书签。
3. 集群方案	7
3.1. 心跳机制	7
3.1.1. 心跳原理	7
3.1.2. 消息格式	8
3.2. 系统启动与主备切换	9
3.3. 车站控制器（SC）集群	错误!未定义书签。
3.3.1. 心跳	错误!未定义书签。
3.3.1.1. 心跳启动	错误!未定义书签。
3.3.1.2. 心跳故障切换	错误!未定义书签。
3.3.2. 系统启动	错误!未定义书签。
3.3.3. 故障切换	错误!未定义书签。
3.3.3.1. 主SC故障	错误!未定义书签。
3.3.3.2. 主前置处理机（FEP）故障	错误!未定义书签。
3.3.3.3. 主SC和主FEP故障	错误!未定义书签。
3.4. 前置处理机（FEP）集群	错误!未定义书签。
3.4.1. 心跳	10
3.4.2. 系统启动	10
3.4.3. 故障切换	10
3.4.3.1. 主FEP故障	10

3.4.3.2.	主RTDB故障	11
3.4.3.3.	主FEP与主RTDB故障	11
3.5.	实时数据库系统（RTDB）集群	12
3.5.1.	心跳	12
3.5.2.	系统启动	12
3.5.3.	故障切换	13
3.5.3.1.	主RTDB故障	13
3.5.3.2.	主Cache故障	14
3.5.3.3.	主RTDB与主Cache故障	15
3.5.4.	主模式下的数据处理	15
3.5.5.	备模式下的数据处理	16
3.5.5.1.	检查表	16
3.5.5.2.	检查表更新	17
3.5.5.3.	检查表清理	18
3.6.	数据缓存（CACHE）集群	18
3.6.1.	心跳	18
3.6.2.	系统启动	18
3.6.3.	故障切换	19
3.6.4.	主模式下的数据处理	20
3.6.5.	备模式下的数据处理	20
3.6.5.1.	检查表	20
3.6.5.2.	检查表更新	22
3.6.5.3.	检查表清理	22

术语

ISCS	Integrated Supervision and Control System
RTDB	Real-time Database
SC	Station Computer
FEP	Front End Processor

综合监控实时数据系统集群方案

1. 综述

1.1. 概念

集群（cluster）：简单的说就是通过网络连接起来的一组服务器（每一台服务器作为一个节点 node），它们作为一个整体向用户提供应用服务。

一个理想状态的集群应该是这样的，对用户来说，是不知道（也没有必要）对其提供服务的系统到底是由多少节点组成，集群就是一个系统。对集群系统管理人员来说，在不影响集群系统正常提供服务的前提下，能够随意增加、修改和删除集群的节点。

1.2. 目的

与单机系统相比，集群系统有极大的优势。集群系统的巨大优势如下：

- 高可扩展性

随着系统用户的增加或系统接入数据的增大，导致系统性能下降时，可以方便地扩充集群系统（增加新的服务器），来提高系统的服务能力。

- 高可用性

一个节点失效，其它节点可以接替它的工作，有效防止单点失效。

- 高性能

如果是负载均衡集群系统，任务是分配到集群的不同节点处理的，系统可以同时为大量用户提供服务。

- 高性价比

集群中的节点可以采用廉价硬件构造高性能的系统。

- 高可维护性

可以方便的对集群中的节点进行维修和替换，在某些情况下，也可以删除集群中的某些节点。

1.3. 分类

根据集群的设计目的不同，可以把集群划分为两大类：高可用性集群和高性能集群。

高可用性集群就是采用集群技术，来实现系统的高可用性。一般采用容错系统（主从服务器模式）和负载均衡系统（集群的所有节点都处于激活状态，平均分摊系统任务）。

高性能集群系统就是通过高吞吐技术、高并发技术、分布式等技术实现任务的快速处理。

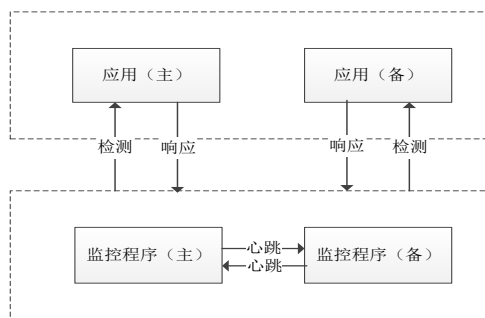


图 3 心跳机制

监控程序由主监控程序和备用监控程序构成。主/备监控程序之间通过心跳互相检测。当主监控程序故障时，备用监控程序自动升级为主监控程序（图 4）。

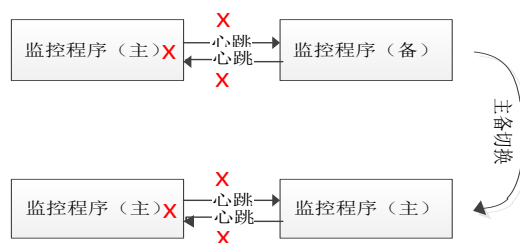


图 4 监控程序主备切换

应用程序的主备切换。应用程序（主）发生故障后，监控程序通知应用程序（备）升级为主。

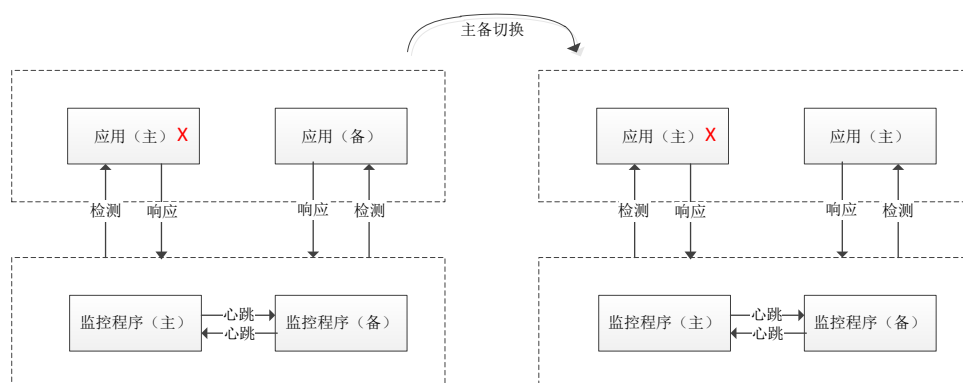


图 4 故障重连

2.1.2. 消息格式

心跳消息有两种：心跳消息和身份消息。它们的格式（图 5），表示的意义不同。



图 5 消息格式

其中：

- id
应用程序 ID。
- role
应用程序角色，取值[‘M’ ， ‘S’]。‘M’ 表示主机， ‘S’ 表示备机。

监控程序向应用程序和监控程序发送心跳消息。应用程序向监控程序发送身份消息。

2.2. 系统启动与主备切换

监控程序启动。第一个启动的监控程序作为主监控程序，第二个启动的作为备用监控程序。即监控程序启动时，检查是否已经有监控程序在运行，如果有则本监控程序作为备用监控启动；如果没有则本监控程序作为主监控程序运行。

应用程序启动。应用程序启动时，按照自己配置的身份启动。即主应用程序启动时以主应用程序身份启动，备应用程序启动时，以备用应用程序身份启动。在异常情况下，应用程序都以主（或备）应用身份启动时，应用的主备身份由监控程序负责裁定。裁定原则监控程序把第一个启动的应用作为主，第二个启动的作为备。

2.3. FEP 集群

FEP 由主 FEP 和备 FEP 构成。

主 FEP。采集设备或系统数据，同时向主 RTDB 和备 RTDB 发送采集到的数据。处理主 RTDB 下发的命令。

备 FEP。采集设备或系统数据，同时向主 RTDB 和备 RTDB 发送采集到的数据。不处理来自主/备 RTDB 的任何命令。

主/备 FEP 之间不相互检测。由监控程序负责主/备 FEP 身份的切换。

2.3.1. 心跳

FEP 心跳详情参照 2.1.2。

2.3.2. 系统启动

正常启动。主 FEP 先启动，备 FEP 后启动，监控程序不对 FEP 做主备切换。备 FEP 先启动，主 FEP 后启动，监控程序不对 FEP 做主备切换。

异常启动。两个 FEP 都为主或备。在 FEP 以当前身份启动后，监控程序把第一个启动的 FEP 升级为主，第二个启动的 FEP 升级为备。

2.3.3. 故障切换

在 FEP 层，系统发生故障，需要进行主备系统切换的情况有三种：一，主 FEP 故障，备 FEP 切换成主 FEP。二，主 RTDB 故障，主 FEP 把 RTDB 服务切换到备 RTDB。三，主 FEP 与主 RTDB 同时故障，备 FEP 切换成主 FEP，主 FEP 把 RTDB 服务切换到备 RTDB。

2.3.3.1. 主 FEP 故障

主 FEP (A) 发生故障，备 FEP (B) 自动切换成主 FEP，主 FEP (B) 自动建立 FEP 到主、备 RTDB 的网络连接。FEP (A) 恢复后，自动切换成备 FEP (图 17)。

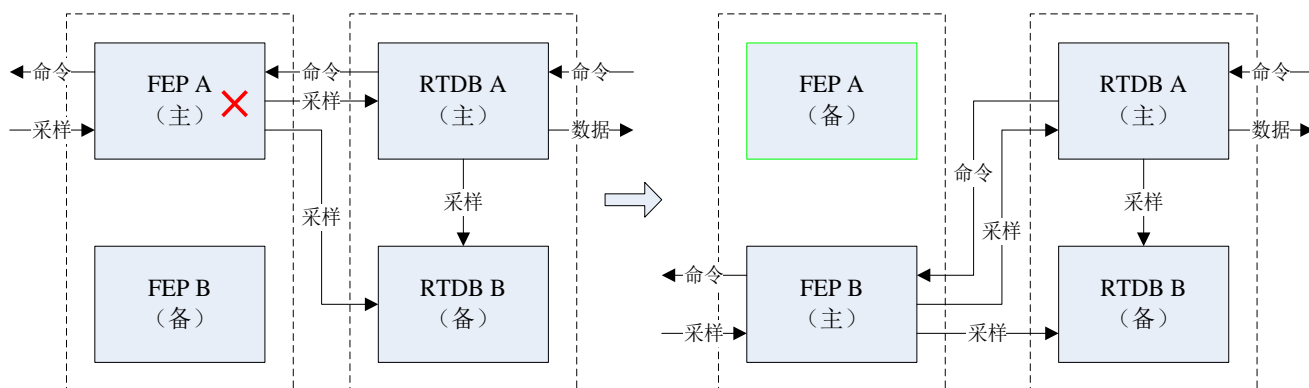


图 17 FEP 集群故障切换--主 FEP 故障

2.3.3.2. 主 RTDB 故障

主 RTDB (A) 发生故障，主 FEP (A) 自动把主 RTDB (A) 服务切换到备 RTDB (B)，RTDB 的主备切换在 RTDB 集群中说明 (图 18)。

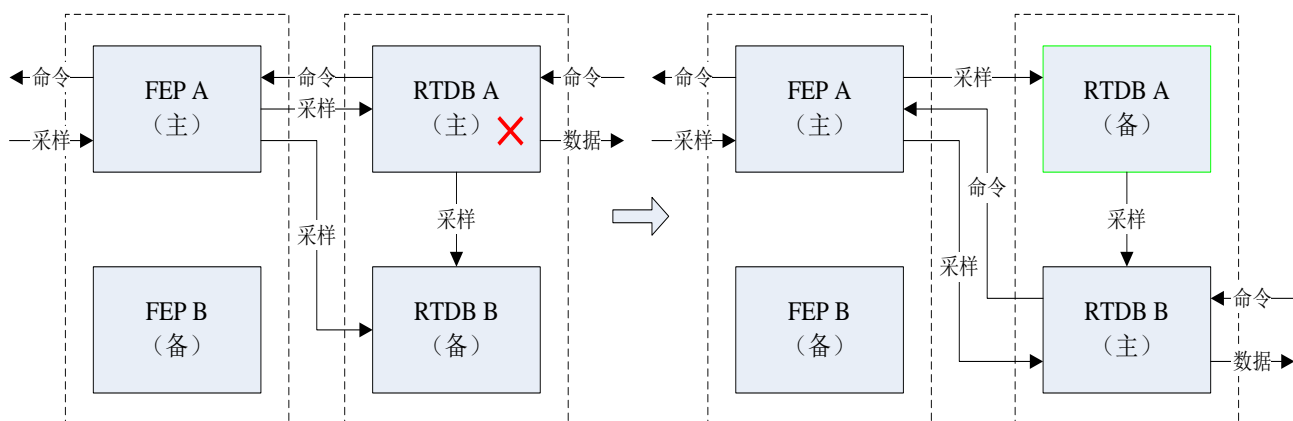


图 18 FEP 集群故障切换--主 RTDB 故障

2.3.3.3. 主 FEP 与主 RTDB 故障

主 FEP (A) 与主 RTDB (A) 同时发生故障时。首先，备 FEP (B) 自动切换成主 FEP；然后，主 FEP (B) 把主 RTDB (A) 服务切换到备 RTDB (B)，RTDB 的主备切换在 RTDB 集群中说明。FEP (A) 恢复后，自动切换成备 FEP (图 19)。

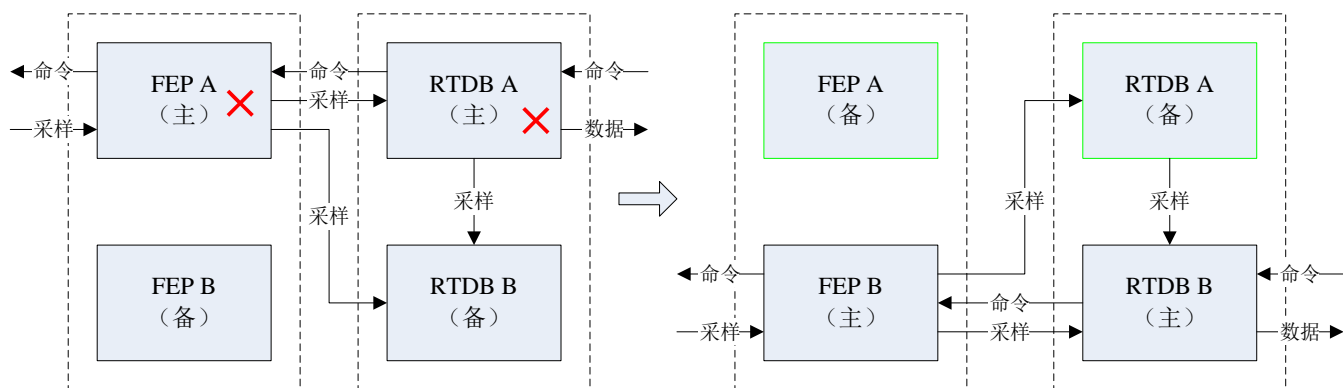


图 19 FEP 集群故障切换--主 FEP 与主 RTDB 故障

2.4. 实时数据库系统（RTDB）集群

RTDB 是一个主备模式的集群，该集群作为综合监控系统集群的一个集群节点。主 RTDB 对 HMI 提供命令服务，对主备 Cache 提供数据服务。备 RTDB 不提供命令和数据服务，仅提供主备 RTDB 数据同步处理。

2.4.1. 心跳

RTDB 心跳详情参照[SC心跳](#)。

2.4.2. 系统启动

当主 RTDB 启动时，在完成系统初始化之后，首先检测网络中是否有主 RTDB 存在。如果已经存在主 RTDB，则自动把自己降级为备，作为备 RTDB 启动；否则，则正常启动。

当备 RTDB 启动时，在完成系统初始化之后，首先检测网络中是否有主 RTDB 存在。如果不存在主 RTDB，则自动把自己升级为主，作为主 RTDB 启动；否则，则正常启动。

正常情况下，主 RTDB（A）主动建立 RTDB 到主备 TSDB、主 RTDB 到备 RTDB 的网络连接。备 RTDB（B）不主动与主备 Cache、主 RTDB 建立网络连接（见图 20）。

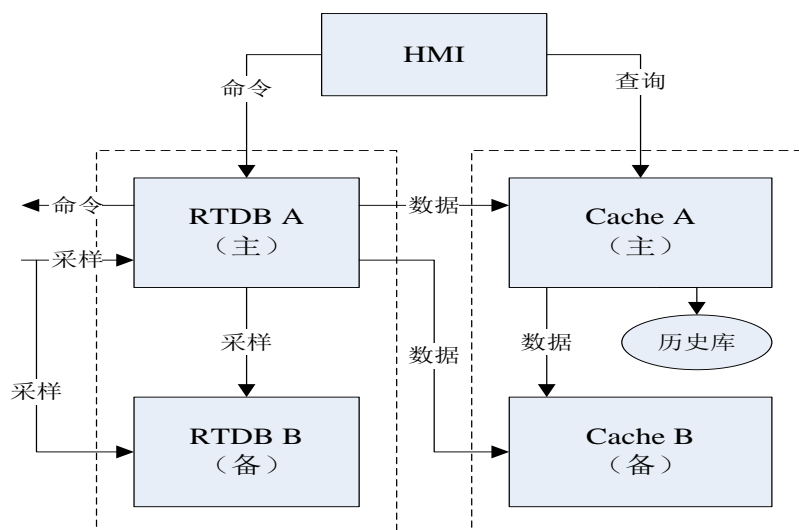


图 20 RTDB 集群启动

2.4.3. 故障切换

在 RTDB 层，发生故障需要进行主备切换的情况有三种：一、主 RTDB 故障，备 RTDB 切换成主 RTDB。二、主 Cache 故障，主 RTDB 把 Cache 服务切换到备 Cache。三、主 RTDB、主 Cache 同时故障，备 RTDB 切换成主 RTDB；主 RTDB 把 Cache 服务切换到备 Cache。

2.4.3.1. 主 RTDB 故障

主 RTDB (A) 发生故障时，备 RTDB (B) 自动切换成主 RTDB，主 RTDB (B) 自动建立与 RTDB (A)、主 Cache (A) 的网络连接。故障 RTDB (A) 恢复后，自动切换成备 RTDB (图 21)。

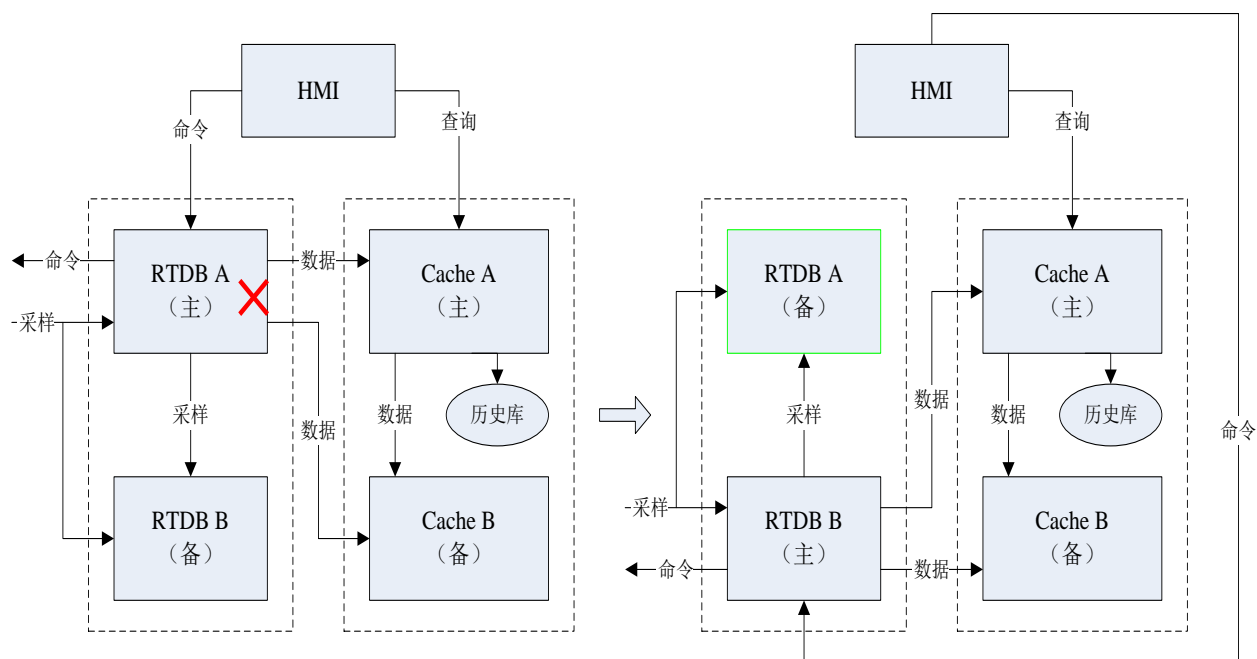


图 21 RTDB 故障切换--主 RTDB 故障

2.4.3.2. 主 Cache 故障

主 Cache (A) 发生故障时，主 RTDB (A) 自动把主 Cache (A) 服务切换到备 Cache (B)，Cache 的主备切换在 Cache 集群中说明 (图 22)。

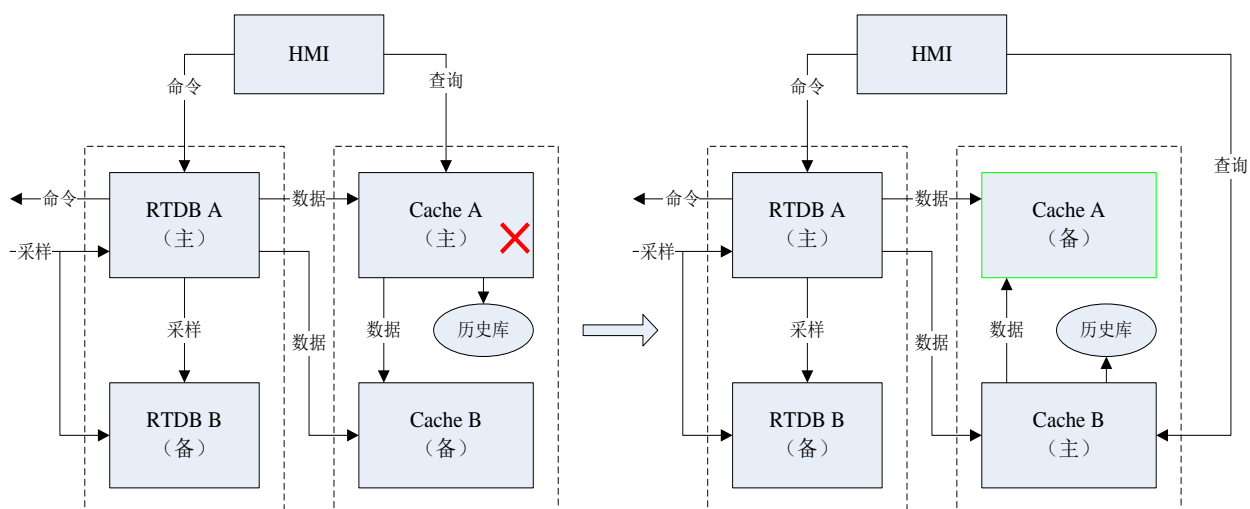


图 22 RTDB 故障切换--主 CACHE 故障

2.4.3.3. 主 RTDB 与主 Cache 故障

主 RTDB (A) 与主 Cache (A) 同时发生故障时。首先，备 RTDB (B) 自动切换成主 RTDB；然后，主 RTDB (B) 把主 Cache (A) 服务切换到备 Cache (B)，并与 RTDB (A) 建立网络连接。Cache 的主备切换在 Cache 集群中说明。RTDB (A) 恢复后，自动切换成备 RTDB (图 23)。

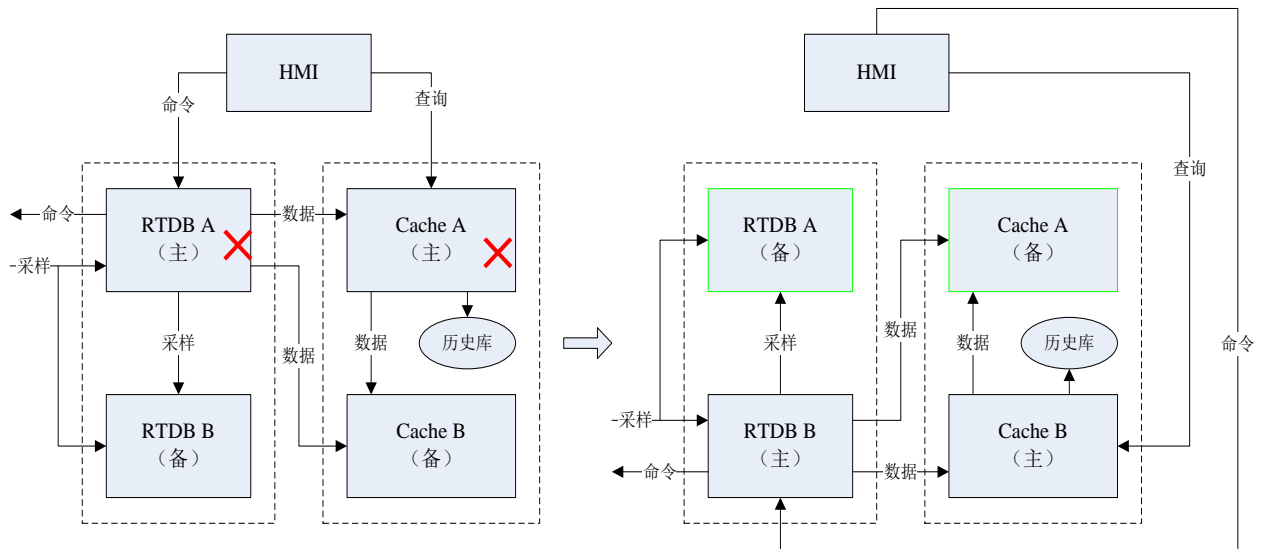


图 23 RTDB 故障切换--主 RTDB 与主 Cache 故障

2.4.4. 主模式下的数据处理

在主模式下，RTDB 提供两大功能：采样数据的处理、监控命令处理和数据同步。

采样数据处理流程：

- 1) 从主 FEP 接收采样数据。
- 2) 把采样数据同步发送给备 RTDB。
- 3) 把采样数据同步更新到内核实时数据区。
- 4) 对采样数据进行分析处理，并把分析结果同步更新到内核实时数据区。
- 5) 把采样数据和分析的结果数据上传给主、备 Cache。

监控命令处理流程：

- 1) 从 HMI 接收命令，并同步更新到内核实时数据区的数据。

-
- 2) 检查命令的执行环境，并把结果同步更新到内核实时数据区。
 - 3) 检查命令执行环境的检查结果，如果不允许执行命令，终止命令执行，转步骤 5。
 - 4) 执行命令，并把命令执行的结果同步更新到内核的实时数据区。
 - 5) 命令结果返回。

当 HMI 命令修改了 RTDB 的运行参数、数据分析处理条件参数、控制参数等，这个改变要同步更新到备 RTDB。

数据处理和命令处理的详情见文档《综合监控实时数据库系统点表设计》。

2.4.5. 备模式下的数据处理

在备模式下，RTDB 提供两大功能：数据同步和数据分析。

RTDB 的数据同步通过主 FEP、主 RTDB 和备 RTDB 共同实现的。首先，主 FEP 把采样数据分别上传给主、备 RTDB；其次，主 RTDB 从主 FEP 接收到上传的采样数据要发送给备 RTDB；最后，备 RTDB 把主 FEP 与主 RTDB 发送的采样数据进行校验和归并，然后对采样数据进行分析（与主 RTDB 相同，只是备 RTDB 不向 Cache 上传分析结果）。

2.4.5.1. 检查表

RTDB 使用检查表来检查和归并重复采样数据。检查表是 ID、数据映射表，ID 与数据一一对应。在检查表中，ID 是唯一的。通过 ID 可以唯一索引到数据。检查表记录内容为：

{id, data}

data 的内容为：

{sources, checked_count, created_time, modified_time, raw_record}

其中：

➤ id

检查表的 id，通过 raw_record 计算得到。具体算法待定，要求不同的 raw_record 计算的结果 id 不重复。

➤ data

检查表的数据项，由多个数据域组成，其中：

-
- sources

raw_record 的来源标识，掩码值及其含义：

- 0x00000000 无
- 0x00000001 FEP
- 0x00000002 RTDB
- 0x00000003 FEP & RTDB

- checked_count

数据项被检查过的次数。

- created_time

表记录创建时间。

- modified_time

表记录被修改的时间。

- raw_record

原始记录。在 RTDB 中是一条采样记录（即一个数据点的一次采样，内容为设备状态值、采样时间、设备地址等），是 FEP 或 RTDB 上传的数据。

2.4.5.2. [检查表更新](#)

RTDB 从 FEP 收到采样数据（一条或多条采样记录），首先用 ID 生成算法，计算出记录的 ID（比如“XXX”）；然后用生成的“XXX”在检查表检索记录。如果记录已经存在，则更新记录的 sources 值（sources = sources | 0x00000001），更新 modified_time 为当前时间；然后检查 sources 值，如果 sources 同时包含 FEP 源和 RTDB 源，则把记录的 raw_record 更新到采样数据就绪表，并把记录从检查表移除。如果记录不存在，则新建一条记录；为 sources 添加 FEP 源，设置 created_time 和 modified_time 为当前时间，把采样记录赋给 raw_record。

RTDB 从主 RTDB 收到采样数据，处理过程同上（不同之处是 **sources** 添加 RTDB 源）。

在检测表更新完成后，RTDB 对采样数据就绪表中的采样数据进行分析处理（过程同主 RTDB，只是结果不会上传给 **Cache**）。

2.4.5.3. 检查表清理

RTDB 定时对检查表进行清理，以清除得不到更新的记录。检查的指标有记录被检索次数、更新超时、存活时间超时。

检查表中的记录，每被检查一次，**checked_count** 值就会加 1。如果 **checked_count** 达到一定次数（比如 3 次），认为该记录已经有效（无需再等待记录合并等），把记录的 **raw_record** 加入采样数据就绪表，并从检查表移除本记录。

如果检测表记录的创建时间超时（比如 3 秒），或最近一次更新时间超时（比如 2 秒），则认为该记录有效，把记录的 **raw_record** 加入采样数据就绪表，并从检查表删除本记录。

在检查完检测表后，RTDB 对采样数据就绪表中的采样数据进行分析处理（过程同主 RTDB，只是结果不会上传给 **Cache**）。

2.5. 数据缓存（**CACHE**）集群

2.5.1. 心跳

RTDB 心跳详情参照[SC心跳](#)。

2.5.2. 系统启动

当主 **Cache** 启动时，在完成系统初始化之后，首先检测网络中是否有主 **Cache** 存在。如果已经存在主 **Cache**，则自动把自己降级为备，作为备 **Cache** 启动；否则，则正常启动。

当备 **Cache** 启动时，在完成系统初始化之后，首先检测网络中是否有主 **Cache** 存在。如果不存在主 **Cache**，则自动把自己升级为主，作为主 **Cache** 启动；否则，则正常启动（见图 24）。

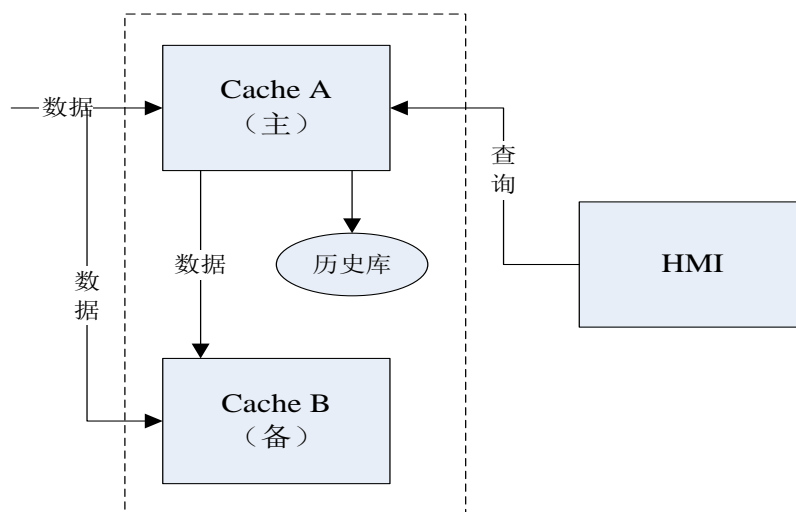


图 24 Cache 集群启动

2.5.3. 故障切换

在 Cache 层，发生故障进行主备切换的情况只有一种，主 Cache (A) 故障，备 Cache (B) 自动切换成主 Cache。其中，HMI 需要在主 Cache (A) 发生故障后，自动把对 Cache 的查询切换到 Cache (B)。Cache (A) 恢复后，自动降为备 Cache (图 25)。

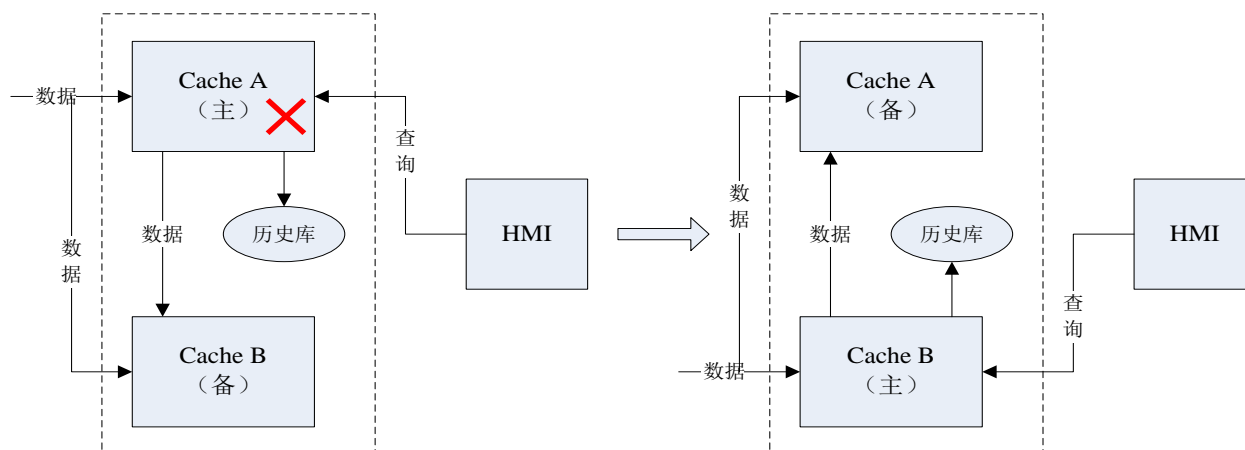


图 25 Cache 集群故障切换

2.5.4. 主模式下的数据处理

在主模式下，Cache 提供数据缓存、写历史库、数据查询（实时数据查询、缓存数据查询）、数据发布订阅。

Cache 收到 RTDB 上传的数据后，处理流程如下：

- 1) 把数据同步发送给备 Cache。
- 2) 把数据插入到对应的缓存队列。
- 3) 根据订阅信息，把相关数据发布给订阅者。

Cache 写历史库是在 RTDB 上传的数据，更新到缓存队列时触发的。流程处理流程：

- 1) 统计数据队列中未写入历史库的数据数量，如果未达到设定值，转步骤 4。
- 2) 把数据写入历史库。
- 3) 更新数据状态为已入库，并把状态同步到备 Cache。
- 4) 本次写历史结束。

2.5.5. 备模式下的数据处理

在备模式下，Cache 提供数据缓存和数据同步。

Cache 的数据同步需要主 RTDB、主 Cache 和备 Cache 共同完成。首先，主 RTDB 分别给主、备 Cache 上传数据；其次，主 Cache 收到主 RTDB 上传的数据要发送给备 Cache；最后，备 Cache 把从主 RTDB 与主 Cache 接收的数据进行检查合并，然后存入缓存队列。

2.5.5.1. 检查表

Cache 使用检查表来检查和归并重复的原始采样数据和处理的结果数据。检查表是 ID、数据映射表，ID 与数据一一对应。在检查表中，ID 是唯一的。通过 ID 可以唯一索引到数据。检查表记录内容为：

{id, data}

data 的内容为：

{sources, checked_count, created_time, modified_time, raw_record}

其中：

➤ id

检查表的 id，通过 raw_record 计算得到。具体算法待定，要求不同的 raw_record 计算的结果 id 不重复。

➤ data

检查表的数据项，由多个数据域组成，其中：

- sources

raw_record 的来源标识，掩码值及其含义：

- 0x00000000 无
- 0x00000001 RTDB
- 0x00000002 Cache
- 0x00000003 RTDB & Cache

- checked_count

数据项被检查过的次数。

- created_time

表记录创建时间。

- modified_time

表记录被修改的时间。

- raw_record

原始记录。Cache 中是一条采样原始记录或处理的结果记录。

2.5.5.2. 检查表更新

Cache 从 RTDB 收到数据记录（一条或多条记录），首先用 ID 生成算法，计算出记录的 ID（比如“XXX”）；然后用生成的“XXX”在检查表检索记录。如果记录已经存在，则更新记录的 sources 值（sources = sources | 0x00000001），更新 modified_time 为当前时间；然后检查 sources 值，如果 sources 同时包含 RTDB 源和 Cache 源，则把记录的 raw_record 更新到数据就绪表，并把记录从检查表移除。如果记录不存在，则新建一条记录；为 sources 添加 RTDB 源，设置 created_time 和 modified_time 为当前时间，把采样记录赋给 raw_record。

Cache 从主 Cache 收到数据记录，处理过程同上（不同之处是 sources 添加 Cache 源）。

在检测表更新完成后，Cache 把数据就绪表中的数据插入数据缓存队列。

2.5.5.3. 检查表清理

Cache 定时对检查表进行清理，以清除得不到更新的记录。检查的指标有记录被检索次数、更新超时、存活时间超时。

检查表中的记录，每被检查一次，checked_count 值就会加 1。如果 checked_count 达到一定次数（比如 3 次），认为该记录已经有效（无需再等待记录合并等），把记录的 raw_record 加入数据就绪表，并从检查表移除本记录。

如果检测表记录的创建时间超时（比如 3 秒），或最近一次更新时间超时（比如 2 秒），则认为该记录有效，把记录的 raw_record 加入数据就绪表，并从检查表删除本记录。

在检查完检测表后，Cache 把数据就绪表中的数据插入缓存队列。