第九节：分布式恢复（二）---进阶篇

4.配置分布式恢复

4.1尝试连接次数

对于从二进制日志进行状态转移，组复制会限制加入节点尝试从引导节点资源池连接到引导节点时进行的尝试次数。如果在没有成功连接的情况下达到连接重试限制，则分布式恢复过程将以错误终止。此限制指定了加入节点尝试连接到引导节点的总数。例如，如果2个组成员是合适的引导节点，并且连接重试限制设置为4，则连接成员在达到限制之前会尝试2次连接到每个引导节点。

默认连接重试限制为10。可以使用group_replication_recovery_retry_count系统变量配置此设置。以下命令将最大连接尝试次数设置为5：

```
mysql> SET GLOBAL group_replication_recovery_retry_count= 5;
```

4.2连接尝试的间隔

对于基于二进制日志的状态传输，系统变量group_replication_recovery_reconnect_interval定义了分布式恢复过程中重新连接引导节点的时间间隔。注意，如果最大重试次数设置为4，集群内有2个候选引导节点，则会先连续2次分别尝试连接这两个候选引导节点（不会使用系统变量group_replication_recovery_reconnect_interval设置的间隔时间，因为这两个候选引导节点之间并没有相互的强关联影响因素，所以没有必要在占满这两个候选引导节点之前就执行重试等待）。一旦加入节点尝试与所有的候选引导节点都执行了连接尝试之后（假设这里2个候选引导节点都在同时做连接尝试），那么，比起候选节点数量来讲，多余的重试连接次数（4-2=2次）就会按照系统变量group_replication_recovery_reconnect_interval配置的时间间隔（单位秒）对分布式恢复程序进行休眠。

默认的连接重试间隔为60秒，可以动态更改此值。以下命令将分布式恢复加入节点连接重试间隔设置为120秒：

```
mysql> SET GLOBAL group_replication_recovery_reconnect_interval= 120;
```

对于远程克隆操作，此间隔不适用。在开始尝试从二进制日志进行状态转移之前，组复制仅对每个合适的引导节点进行一次连接尝试。

4.3设置新加入节点为online状态

当分布式恢复已成功完成从引导节点到加入节点的状态转移时，可以将加入成员标记为集群中的online节点并准备加入。默认情况下，这是在加入节点收到并应用了所有丢失的事务之后完成的。 可以允许加入成员在收到并验证所有丢失的事务之后（但尚未应用它们）将其设置为online状态。如果要执行此操作，使用group_replication_recovery_complete_at系统变量来指定备用设置TRANSACTIONS_CERTIFIED。

5.分布式恢复的容错性

组复制的分布式恢复过程具有许多内置操作，以确保恢复的过程中出现任何问题时的容错能力。

从当前视图中的online状态集群节点列表中随机选择用于分布式恢复的引导节点。选择随机的引导节点意味着当多个外部节点加入该集群时，不太可能一次选择集群内同一个server。从MySQL 8.0.17开始，为了从二进制日志进行状态转移，所有的加入节点在集群内仅能选择同一个引导节点，引导节点的版本低于或等于加入节点的MySQL Server补丁版本。对于较早的发行版本，集群内所有online节点都可以成为引导节点。对于远程克隆操作，仅选择运行与自身相同修补程序版本的引导节点。请注意，当加入节

点在操作结束重新启动时，它将与新的引导节点建立连接以从二进制日志进行状态转移，该二进制日志可能与用于远程克隆操作的原始节点不是同一个节点。

在以下情况下，组复制在分布式恢复中检测到错误，自动切换到新的引导节点，然后重试状态传输：

连接错误-与候选引导节点的连接存在身份验证问题或其他问题。

复制错误-用于从二进制日志进行状态转移的复制线程之一（接收者线程或应用线程）失败。由于这种状态转移方法使用现有的MySQL复制框架，因此某些暂时性错误可能会导致接收方或应用线程中的错误。

远程克隆操作错误-远程克隆操作失败或在完成之前已停止。

引导节点离开集群，或在状态转移过程中停止引导节点上的组复制。

performance_schema.Replication_applier_status_by_worker表显示导致最后重试的错误。在上述这些情况下，尝试使用新的候选引导节点进行错误之后的重新连接。如果发生错误，则选择其他引导节点意味着新的候选节点可能没有相同的错误。如果安装了克隆插件，则组复制将首先尝试使用每个合适的支持在线克隆的引导节点进行远程克隆操作。如果所有这些尝试均失败，则组复制将尝试使用所有合适的引导节点从二进制日志中依次进行状态传输（如果可能）。

ps:对于远程克隆操作，在远程克隆操作开始从引导节点传输数据之前，先删除用户创建的表空间和接收者（加入成员）上的数据。如果远程克隆操作开始但未完成，则加入成员可能会保留其原始数据文件的一部分，或者没有用户数据。如果在完全克隆数据之前停止了克隆操作，则将引导节点转移的数据从接收者中删除。可以通过重试克隆操作来纠正这种情况，组复制会自动执行此操作。

在以下情况下，分布式恢复过程无法完成，并且加入节点离开集群：

事务被清理-任何online成员的二进制日志文件中都没有加入成员所需的事务，并且无法通过远程克隆操作获取数据（因为未安装克隆插件，或者因为尝试了全部克隆可能的引导节点，但失败了。因此，加入节点无法获得集群的全部数据）。

事务冲突-加入节点已经包含了集群中不存在的一些事务。如果执行远程克隆操作，则这些事务将被删除并丢失，因为会删除加入成员上的数据目录。如果从引导节点的二进制日志进行状态转移，则这些事务可能与集群的事务冲突。

已达到连接重试限制-加入节点已进行了连接重试限制所允许的所有连接尝试。可以使用group_replication_recovery_retry_count系统变量进行配置。

没有更多的引导节点-加入节点依次尝试与每个支持在线克隆的引导节点进行远程克隆操作（如果已安装克隆插件），但是没有成功地尝试从二进制日志中通过每个合适的在线方式进行状态转移。

加入节点离开集群，或者在状态转移进行期间，在加入节点上停止组复制。

如果加入节点无意中离开了集群，那么在上面列出的任何情况下，除了最后一个，它都会继续执行group_replication_exit_state_action系统变量指定的操作。

6.分布式恢复原理

当Group Replication的分布式恢复过程正在从二进制日志执行状态转移时，要在特定的时间点之前将加入节点与引导节点同步，则加入节点和引导节点将使用GTID。但是，GTID仅提供一种手段来了解加入节点所缺少的事务。它们无助于标记加入集群的server必须赶上的特定时间点，也无法传达认证信息。这是二进制日志视图标记的工作，它标记二进制日志流中的视图更改，并且还包含其他元数据信息，从而为加入节点提供缺少与认证相关的数据。

本节说明视图更改的作用和视图更改标识符，以及从二进制日志执行状态转移的步骤。

6.1视图和视图变更

视图：对应于组中活跃成员的当前配置，换句话说，它是在特定的时间点所有组成员达成一致状态的配置。

视图更改：当对集群配置进行修改（例如成员加入或离开）时，将发生视图变更。任何组成员身份更改都会导致在相同的逻辑时间点向所有成员传达独立的视图更改。

视图标识符：唯一地标识视图，每当视图更改发生时，都会生成它。

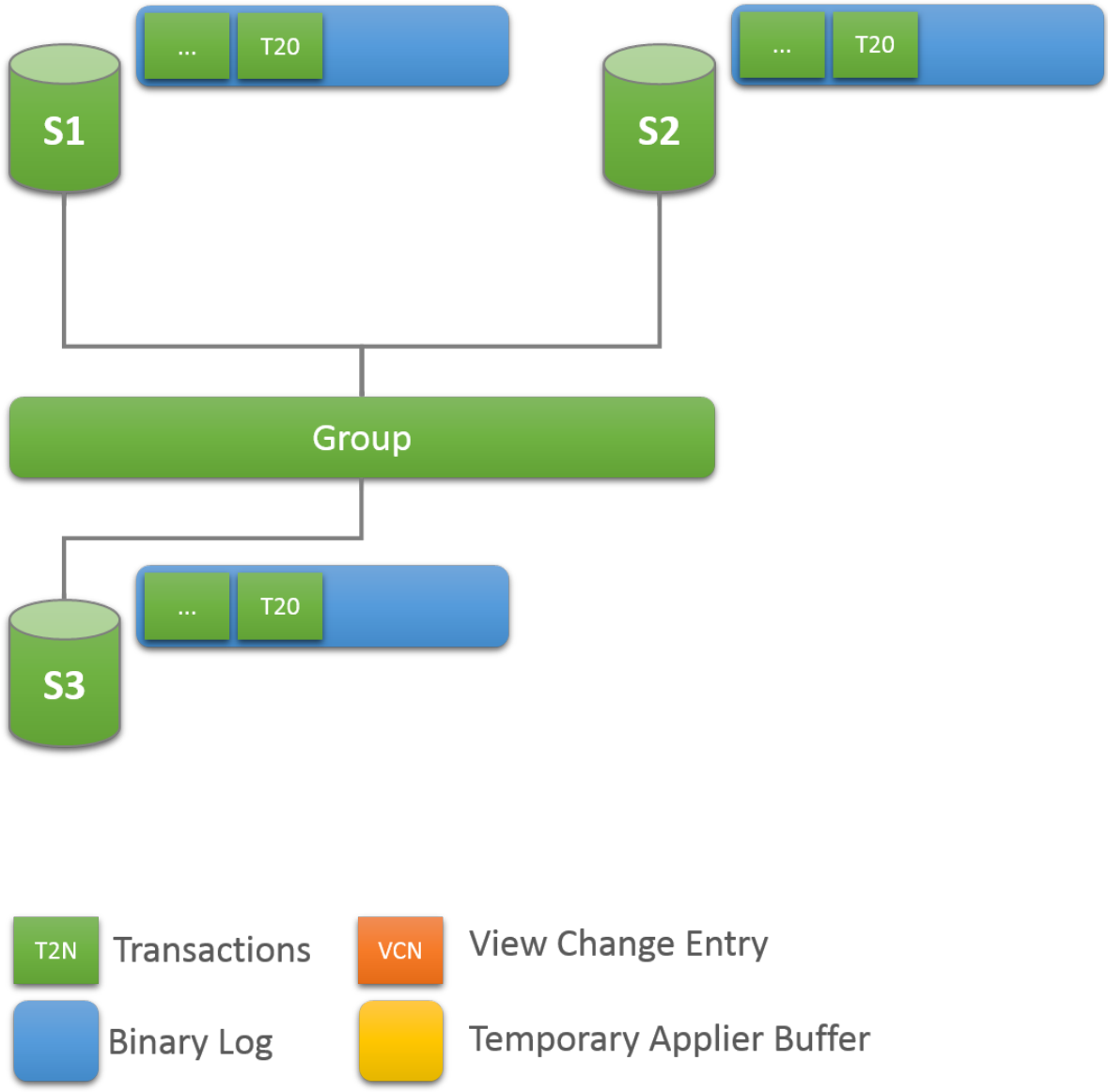
在组通信层，视图更改及其关联的视图标识符标记了节点加入之前和之后交换的数据之间的边界。这个概念是通过二进制日志事件实现的。记录视图标识符以划分组成员资格发生更改之前和之后传输的事务。

视图标识符本身由两部分组成：一个随机生成的部分和一个单调递增的整数。（例如：view_id=15692965051216743:3）随机生成的部分是在创建集群时生成的，并且在集群中至少有一个成员时保持不变。每次视图更改发生时，整数都会递增。通过使用这两个不同的部分，视图标识符就可以唯一标识由于Server加入组或组成员脱离组而导致的组成员资格变更，还可以标记在完全关闭组时所有成员退出组的情况，这将确保二进制日志中的数据标记保持唯一，以便在完全关闭组后不会重用相同的标识符，以防止将来的分布式恢复出现问题。

6.2稳定状态下的集群

集群内所有server均处于online状态，并且正在处理来自该集群传入的事务。在事务复制方面，某些server可能会有些落后，但最终它们会达成最终的一致性。该集群充当一个分布式的数据库副本。

示意图如下所示：

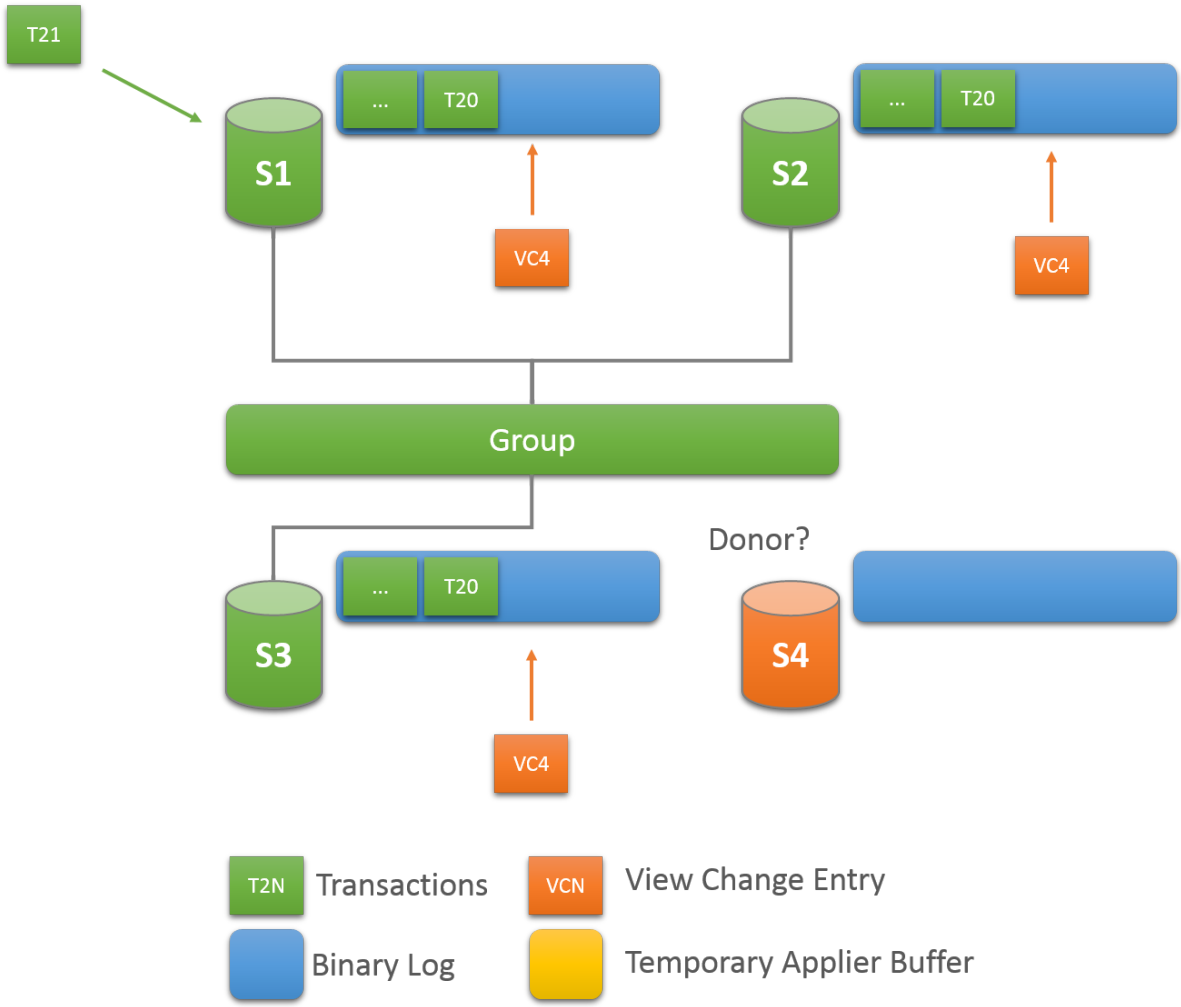


6.3集群有新节点加入，视图发生变更

每当新节点加入该集群并因此执行视图变更时，集群内的每个online server都会将视图更改日志事件排队等待执行。之所以要排队，是因为在更改视图之前，在server上还有旧的视图更改事务队列尚未应用完成。将视图变更日志时间排在这些属于旧视图的事务之后可以确保正确标记什么时候发生了视图变更。

同时，新加入集群的Server从视图声明的online节点列表中选择一个合适的引导节点。如下图所示，Server S4申请加入集群时生成视图4（VC4），集群中的所有online节点将视图变更日志事件写入二进制日志中（如果有节点存在应用延迟，则会先将View_change_log_event事件缓存在队列里排队，在该事件之前的事务属于旧视图，在该事件之后的事务属于新视图）。

示意图如下所示：

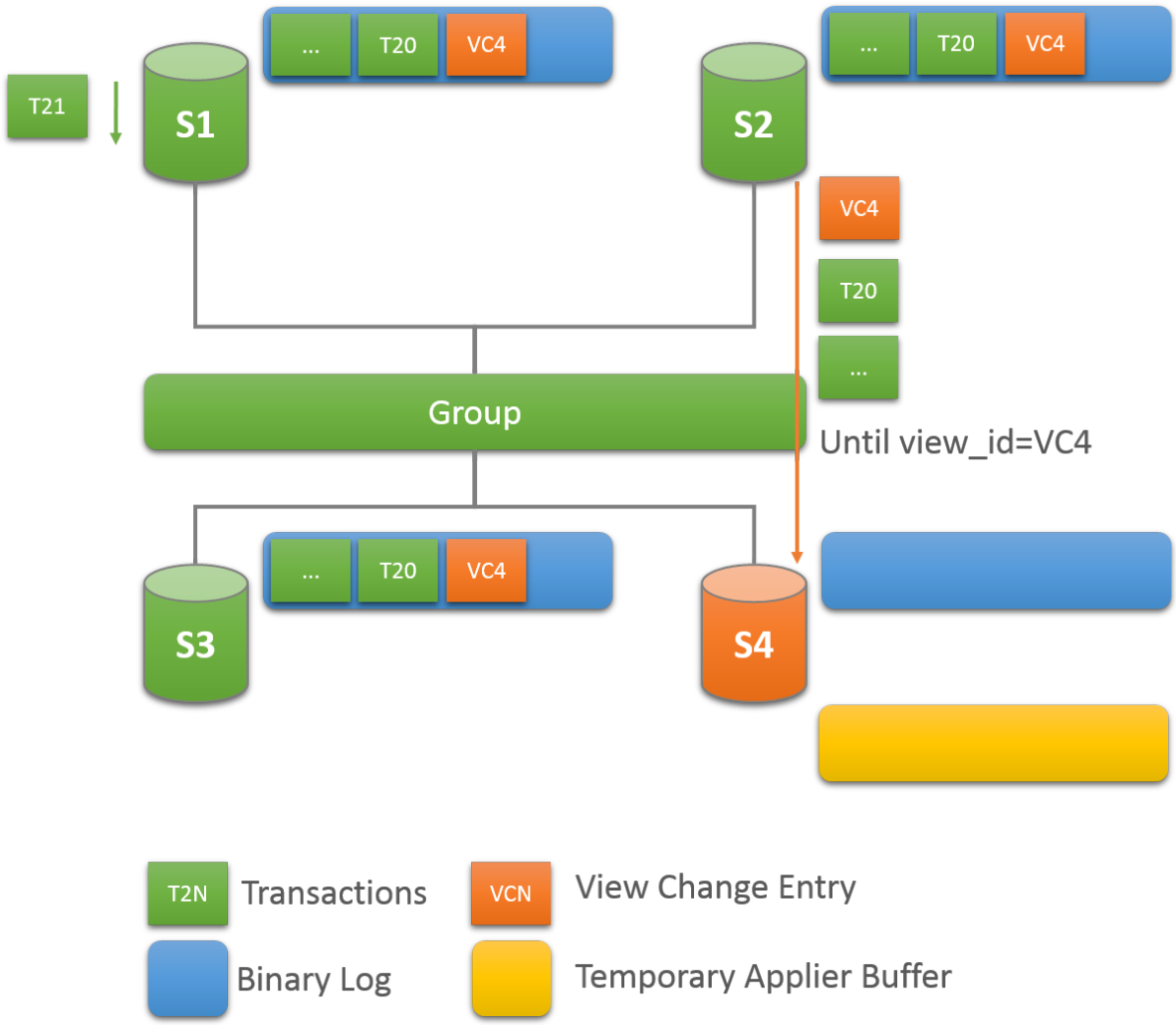


6.4状态转移---追赶数据

如果集群中的节点和加入节点都设置好了克隆插件，如果加入节点与集群之间的事务差异量超过了系统变量group_replication_clone_threshold设置的阈值，组复制就会通过远程克隆操作执行分布式恢复；如果加入节点在集群中的任何节点的二进制日志文件中都找不到所需的事务，组复制也会通过远程克隆操作执行分布式恢复。在远程克隆操作期间，加入节点上的现有数据将会被全部删除，并替换为引导节点的数据副本。当远程克隆操作完成且加入节点完成数据库实例的重启操作之后，将从引导节点执行基于二进制日志的状态传输，以获取在远程克隆操作期间该集群中新应用的事务；如果加入节点与集群之间的事务差异不大，或者没有安装克隆插件，则组复制直接从引导节点执行基于二进制日志进行状态传输。

对于从引导节点执行基于二进制日志的状态传输，会在加入节点（这里为Server S4）和引导节点之间建立一个异步复制机制的专用通道，然后开始状态传输。与引导节点的这种交互将一直持续，直到Server S4中的applier线程处理视图变更日志事件（这里为VC4）为止，该事件对应于Server S4进入该组时触发的视图变更。这个时候，集群内的所有节点读取到VC4时，通过VC4都能够清楚地知道在这之前的事务属于old view，在这之后的事务属于new view。

示意图如下：

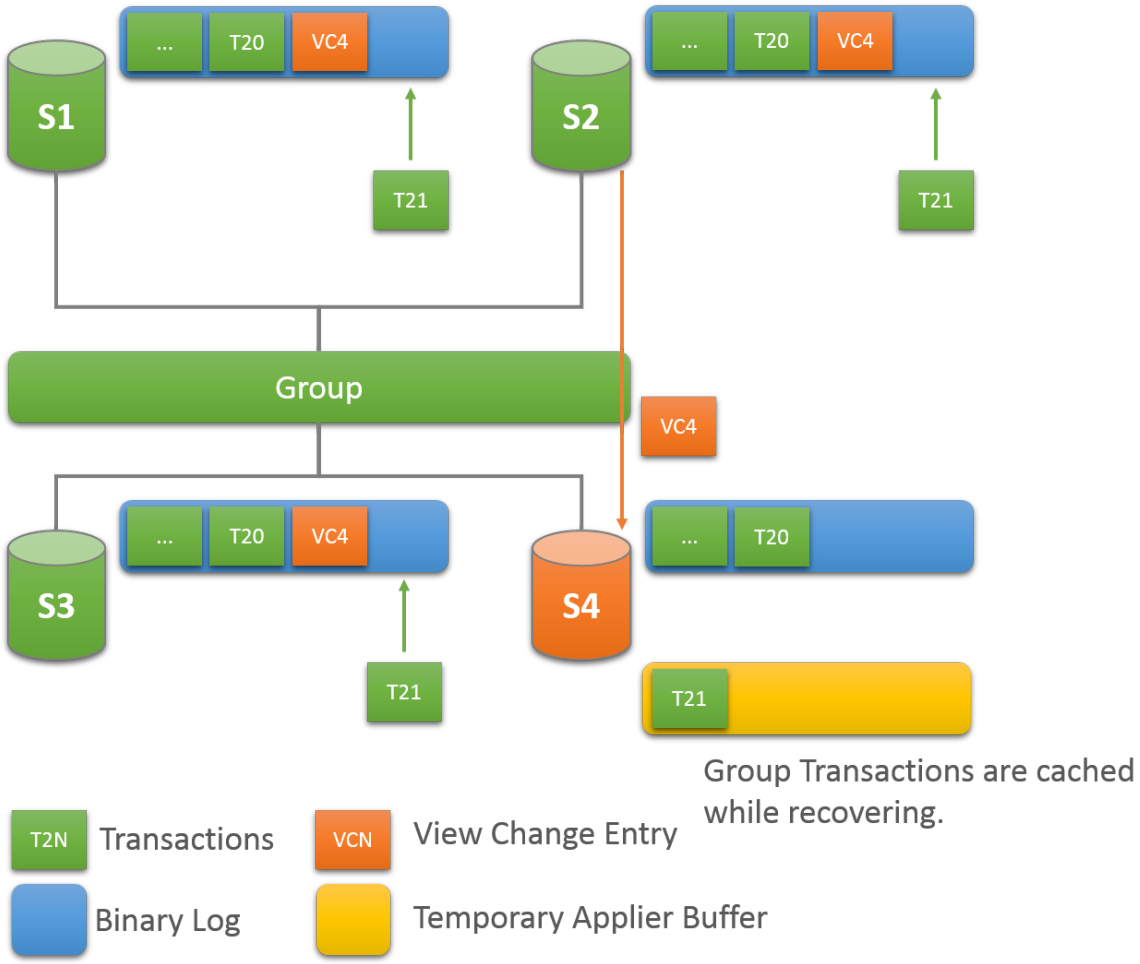


6.5应用缓存队列中的事务---追赶集群最新数据

由于视图标识符（VC4）在同一逻辑时间会传输给集群中的所有节点，所以Server S4知道应该在哪个视图标识符（VC4）处停止复制（注意，这里说的停止复制指的是停止在Server S4与引导节点之间建立的专用的异步复制通道）。这避免了复杂的GTID SET计算，因为视图标识符（VC4）清楚地标记（界定）了哪些数据属于哪个组视图。

当Server S4从引导节点复制数据的过程中，它也同时缓存来自集群的新执行的事务。最终，当Server S4停止与引导节点之间的异步复制连接之后，它将应用那些被缓存的事务（这是实现Server S4加入组的过程中，在该阶段不阻塞写业务的关键特性）。

示意图如下：



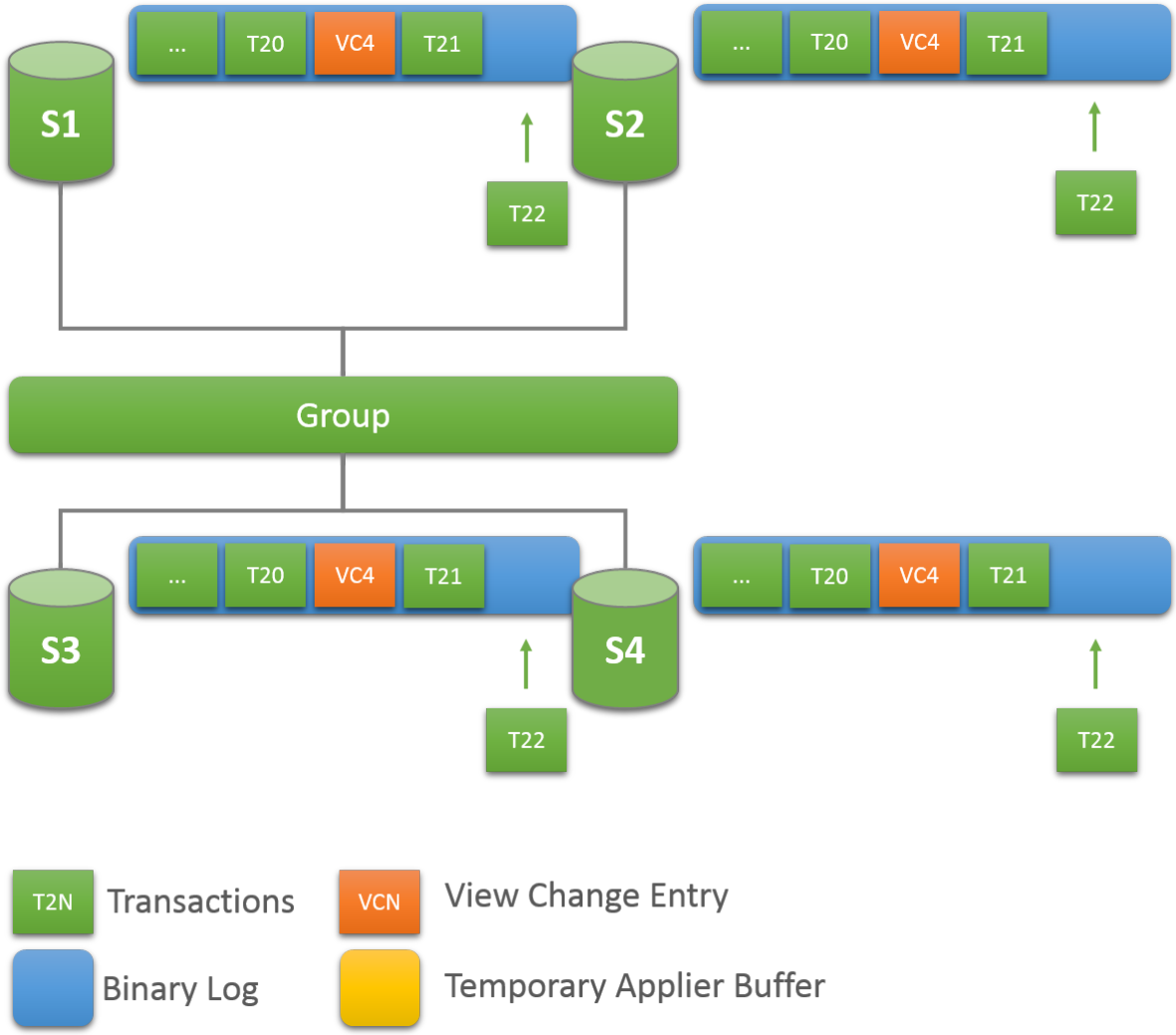
6.6状态转移完成---数据追平

当加入节点（Server S4）使用预期的视图标识符识别了视图变更日志事件（VC4）时，它与引导节点之间的连接将终止，并开始应用自身缓存中的增量事务。VC4 除了在二进制日志中充当新旧视图的分隔标记之外，它还扮演另一个角色。当Server S4成员进入集群时，它传递所有服务器感知到的认证信息，就是最后一次视图变更。如果没有VC4，Server S4将没有所需的信息来验证（检测冲突）后续的事务。

追赶的持续时间是不确定的，因为它取决于工作负载和整个过程中集群内新进入的事务速率。因为，这个过程是完全在线的，Server S4在追赶集群数据的过程中不会阻塞集群中的任何其他成员写入新的数据。因此，当Server S4在执行此阶段过程中，后续新写入集群的事务可能会堆积，堆积的事务多少取决于工作负载。

当Server S4应用完成缓存中的事务（缓存队列为空）且其存储的数据与组中其他成员达到一致时，其公共状态将更改为ONLINE。

示意图如下：


...
T20
VC4
T21
S1
S2
T22
T22
Group
S3
S4
T22
T22
T2N
Transactions
VCN
View Change Entry
Binary Log
Temporary Applier Buffer