Tair2.3部署说明

* 程序配置介绍

安装好tair后（路径：/usr/local/tair）,在目录下有一个sbin目录与etc目录，现在分别对这两个目录下的文件作介绍。

Tair 所需的G++版本为4.4.7

Tair的集群部署方式为单机房单集群

sbin目录下主要程序功能介绍：

**tair\_cfg\_svr**：tair控制服务器程序

1) 通过维护和dataserver心跳来获知集群中存活节点的信息  
2) 根据存活节点的信息来构建数据在集群中的分布表。  
3) 提供数据分布表的查询服务。  
4) 调度dataserver之间的数据迁移、复制。

**tair\_server**：tair数据服务器程序

1) 提供存储引擎  
2) 接受client的put/get/remove等操作  
3) 执行数据迁移，复制等  
4) 插件：在接受请求的时候处理一些自定义功能  
5) 访问统计

**inval\_server**：tair跨机房集群管理服务器程序

1) 接收来自client的invalid/hide等请求后，对属于同一组的集群（双机房独立集群部署方式）做delete/hide操作，保证同一组集群的一致。  
2) 集群断网之后的，脏数据清理。  
3) 访问统计。

**tairclient**：tair轻量级客户端测试程序

1) 在应用端提供访问tair集群的接口。  
2) 更新并缓存数据分布表和invalidserver地址等。  
3) LocalCache，避免过热数据访问影响tair集群服务。  
4) 流控

etc目录下主要配置文件介绍：

**configserver.conf ：**属于控制服务器配置文件，主要配置端口与相关输出目录等参数。

[public]项：第一行config\_server为master服务器，第二行为slaver服务器，如果只配置一台config\_server，需将第二行屏蔽。

[configserver]项：port为控制服务器端口号。注意：dev\_name属性，该属性需要设置服务器IP所对应网卡的名称，通过# ifconfig查看后再设置。

**group.conf：**属于控制服务器配置文件，主要用于注册DataSever服务器的IP和Port，设置数据服务器分组信息，配置数据库等参数。

[group\_1]项：项名称可根据分组需要修改，当前默认组为group\_1；每个group配置文件可以配置多个group，这样一组configserver就可以同时服务于多个 group 了。不同的 group 用group name区分；

\_data\_move 当这个配置为1的时候, 如果发生了某个dataserver宕机，则系统会尽可能的通过冗余的备份对数据进行迁移。注意，如果 copy\_count 为大于1的值，则这个配置无效，系统总是会发生迁移的。只有copy\_count为1的时候。该配置才有作用。

min\_data\_server\_count 这个是系统中需要存在的最少data server的个数。当系统中可正常工作的data server的个数小于这个值的时候，整个系统会停止服务，等待人工介入。

\_build\_strategy 在分配各个桶到不同的data server上去的时候所采用的策略，目前提供两种策略，配置为1 则是负载均衡优先，分配的时候尽量让各个 data server 的负载均衡， 配置为 2 的时候，是位置安全优先，会尽量将一份数据的不同备份分配到不同机架的机器上。 配置为3的时候，如果服务器分布在多个机器上，那么会优先使用位置安全优先，即策略2。 如果服务器只在一个机架上，那么退化成策略1，只按负载分布。

\_build\_diff\_ratio 这个值只有当\_build\_strategy 为2的时候才有意义。 实际上是用来表示不同的机架上机器差异大小的。当位置安全优先的时候，如果某个机架上的机器不断的停止服务，必然会导致负载的极度不平衡。当两个机架上机器数量差异达到一定程度的时候，系统也不再继续工作，等待人工介入。

\_pos\_mask 机架信息掩码，程序使用这个值和由ip以及端口生成的64为的id做与操作，得到的值就认为是位置信息。比如当此值是65535的时候是十六进制 0xffff。因为ip地址的64位存储的时候采用的是网络字节序，最前32位是端口号，后32位是网络字节序的ip地址。所以0xffff 这个配置， 将认为10.1.1.1 和 10.2.1.1 是不同的机架。

\_copy\_count 这个表示一条数据在系统中实际存储的份数。如果tair被用作缓存，这里一般配置1， 如果被用来做存储，一般配置为3，当系统中可工作的data server的数量少于这个值的时候，系统也会停止工作。比如\_copy\_count 为3，而系统中只有 2 台data server。这个时候因为要求一条数据的各个备份必须写到不同的data server上，所以系统无法完成写入操作，系统也会停止工作的。

\_bucket\_number这个是hash桶的个数，一般要 > dataserver的数量(10倍以上)。数据的分布，负载均衡，数据的迁移都是以桶为单位的。

\_accept\_strategy这个默认为0，dataserver重新连接上configserver的时候，需要手动touch group.conf。如果设置成1，则当有dataserver重新连接configserver的时候，不需要手动touch group.conf。 configserver会自动接入该dataserver。

\_areaCapacity\_list 这是每一个area的配额信息。这里的单位是 byte。 需要注意的是，该信息是某个 area 能够使用的所有空间的大小。举个具体例子；当copy\_count为3 共有5个dataserver的时候，每个dataserver上该area实际能使用的空间是这个值/(3 \* 5)。因为fdb使用mdb作为内部的缓存， 这个值的大小也决定了缓存的效率。

**dataserver.conf**：属于数据服务器配置文件，用户设置数据服务器存储引擎及相关各引擎参数。

[public]项：请与configserver.conf保持一致。

storage\_engine 这个可以配置成：

非持久化：mdb（Memcached）持久化：fdb（Firebird）、kdb（Kyoto Cabinet）、 ldb（LevelDB）。其中memcached和Firebird是关系型存储数据库，而Kyoto Cabinet和LevelDB是Nosql数据库。

mdb\_type 这个是兼容以前版本用的，现在都配成mdb\_shm就可以了。

mdb\_shm\_path 这个是用作映射共享内存的文件。

port为dataserver的工作端口。

heartbeat\_port为dataserver的心跳端口。

process\_thread\_num 工作线程数，实际上启动的线程会比这个数值多，因为有一些后台线程。真正处理请求的线程数量是这里配置的。

slab\_mem\_size 所占用的内存数量。这个值以M为单位，如果是mdb，则是mdb能存放的数据量，如果是fdb，此值无意义。

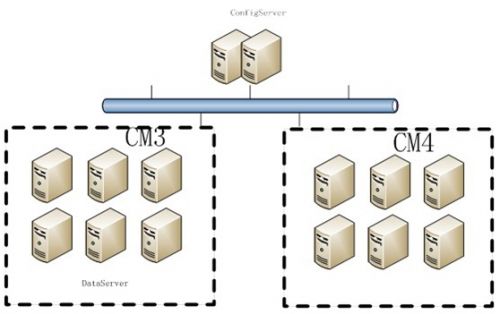
**invalserver.conf：**集群管理配置，暂无研究。

* 集群部署

tair通过多种集群部署方式，来满足各类应用的容灾需求。  
 下面所述的双机房可以扩展到多机房，现阶段基本还是采用的双机房。  
 现总共有4种方式：  
 mdb存储引擎适用于双机房单集群单份，双机房独立集群，双机房单集群双份。  
 ldb存储引擎适用于双机房主备集群，双机房单集群单份。

**双机房单集群单份**

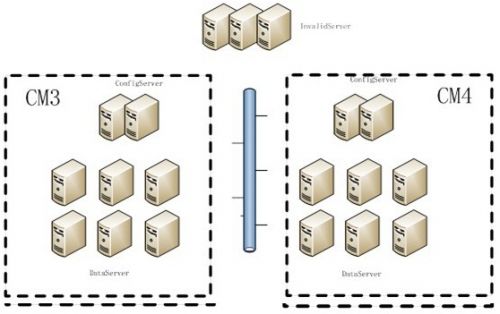
双机房单集群单备份数是指，该tair集群部署在两个机房中（也就是该tair集群的机器分别在两个机房）， 数据存储份数为1， 该类型集群部署示意图如下所示。数据服务器（Dataserver）分布在两个机房中，他们都属于同一集群。



使用场景：  
1) 后端有无数据源都可。  
2) 后端有数据源，且更新比例很高的场景。  
优点：  
1) 服务器存在于双机房，任一机房宕机保持可用。  
2) 单份数据，无论应用在哪个机房，看到的都是同一个数据。  
缺点：  
1) 应用服务器会跨机房访问。如上图，并假设应用服务器在cm3和cm4，那么cm3的应用服务器也可能调用到cm4的tair机器，cm4的亦然。  
2) 当一边机房出现故障时，tair中的数据会失效一半（一半这个数值是按两边机房tair机器数相同估计的，如果不相同，则按对应比例估算）  
该部署方式，应用在删除数据时，只需要调用delete即可，无需调用invalid。当然，调用invalid也可，这种方式下会直接退化到delete。

**双机房独立集群**

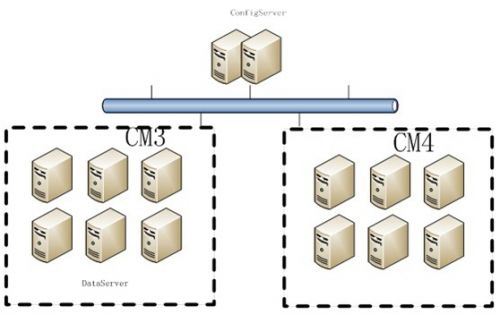
双机房独立集群是指，在两个机房中同时部署2个独立的tair集群，这两个集群没有直接关系。下图是一个典型的双机房独立集部署示意图，可以看到，cm3 和cm4各有一个完整的tair集群（2个configserver+多个dataserver）。图中还多了一个invalidserver的角色， invalidserver接收客户端的invalid或者hide请求后，会对各机房内的集群进行delete或者hide操作，以此保障tair中的 数据和后端数据源保持一致的。



适用场景：  
1) 后端必须要有数据源，否则则退化成单机房集群，tair集群本身不做同步。  
2) 读写比不能过小，不然可能会带来tair命中率降低。例如某个key，在数据库中被频繁更新，那么此时应用必须调用invalid来确保tair和DB的 一致性。此时应用读tair一直会不命中，导致整体命中率低，可能造成DB压力比较大。 如果依然有疑问的话，请联系 tair答疑。  
优点：  
1) 每个机房拥有独立tair集群，应用在哪个机房就访问相同机房的tair集群，不会出现跨机房调用和流量。  
2) 单边机房故障，不会影响业务访问tair命中率。  
缺点：  
1) 后端必须要有数据源，也就是这种部署方式下，tair必然是当作传统意义上的cache存在的。因为tair mdb集群之间本身不会做数据同步，多集群间一致性保证依赖于后端数据源，如DB。  
2) 当后端数据源数据发生更新后，业务不能直接把数据put到tair，而是先需要调用invalid接口来失效这些对等集群中的数据（来保持各tair集群 的数据和后端数据源的一致性）。之后业务可以把数据put到当前tair集群（注意：只会put到本机房的tair集群，不会put到对端集群）或者在读tair时发生not exist的时候从后端数据源取来放入tair。

**双机房单集群双份**

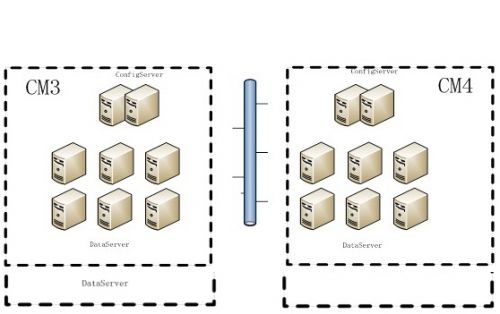
双机房单集群双份，是指一个tair集群部署在2个机房中，数据保存2份，并且同一数据的2个备份不会放在同一个数据服务器上。根据数据分布策略的不同， 还可以将同一份数据的不同备份分布到不同的机房上。该类型的集群部署方式与双机房单集群单份数据的部署方式一样。其不同之处，数据保存份数不一样。该类型 集群部署方式示意图如下图所示，数据服务器分别部署在两个不同的机房里，所有的数据服务器都被相同的配置服务器管理，在逻辑上，他们构成一个独立的集群。



现只有tbsession集群使用了这种部署方式。  
适用场景：  
后端无数据源，临时数据的存放，非cache。  
cache类应用推荐使用双机房独立集群和双机房单集群单份部署方式。  
优点：  
1) 数据存放两份，数据安全性有一定保障。但由于存储引擎是mdb，数据存放在内存中，无法绝对保证数据不丢失。  
2) 当一边机房故障时，另外一边机房依然可以服务，并且数据不丢失。  
缺点：  
1) 如果机房间网络出现异常情况，依然有较小几率丢失数据。

**双机房主备集群**

这种部署方式中，存在一个主集群和一个备份集群，分别在两个机房中。如下图所示，不妨假设CM3中部署的是主集群，CM4中部署的是备份集群。那么，在正 常情况下，用户只使用主集群，读写数据都与主集群交互。主备集群会自动同步数据（不需要业务去更新两边），保证两个机房数据的最终一致性。当一个机房发生 故障后，备集群会自动切换成主集群，提供服务，保证系统可用性。



适用场景：  
该方式只在ldb存储引擎中存在，也就是业务将tair当作最终存储使用。我们会在当前主集群存两份数据，并由tair异步将数据更新到备集群，确保数据安全和服务可用。  
优点：  
1) 数据安全和服务可用性高。

2) 用户调用方便，无需考虑多集群间数据一致性的问题。