计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(7):104−109 [doi: 10.15888/j.cnki.csa.005867]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

# Ceph分布式系统的ISCSI高可用集群①

何汉东, 张　倩

(中电海康集团有限公司, 杭州 310013)

**摘　要**: 在分布式存储集群环境中, 为了兼容现有存储协议, 提高集群可扩展性, 通常都会支持ISCSI存储协议. 而现有ISCSI服务的高可用主要通过主从备份的方式实现, 该方式会导致资源利用不充分, 容易造成单节点负载过重等情况. 本文基于Ceph分布式存储系统的优势, 采用raft一致性算法等设计并实现ISCSI高可用集群. ISCSI集群使用自定义的节点选择策略实现服务负载均衡, 并通过raft分布式协议实现服务故障迁移. 实验表明, 本文提出的方案是有效的, 能够实现服务故障迁移, 保证集群负载均衡.

**关键词**: Ceph; ISCSI高可用; 分布式存储; 负载均衡

引用格式: 何汉东,张倩.Ceph分布式系统的ISCSI高可用集群.计算机系统应用,2017,26(7):104–109. http://www.c-s-a.org.cn/1003-3254/5867.html

## ISCSI High Availability Cluster in Ceph Distributed System

HE Han-Dong, ZHANG Qian

(CETHIK GROUP Co. Ltd., Hangzhou 310013, China)

**Abstract**: Commonly, the distributed storage cluster support ISCSI protocol for compatibility and scalability. But, the high availability of ISCSI services is mainly achieved through a master slave remote system which wastes resources and easily makes the server overload. This paper uses raft consistency algorithms to implement a high availability cluster of ISCSI service based on the advantages of Ceph. In order to implement the high availability, the cluster supports load balancing using sensible selection policy and services migration after detecting the failover of the server. The experiments show that the proposed scheme is effective, and can achieve services migration, and ensure the load balancing.

**Key words**: Ceph; high availability of ISCSI; distributed storage; load balancing

ISCSI(Internet Small Computer System Interface)服务是构建IP-SAN(Storage Area Network)的一种重要技术手段, 具有成本低廉、维护简单、易于扩展、使用广泛等特性[1,2]. ISCSI技术通过使用TCP/IP协议封装SCSI存储协议, 达到操作管理存储设备的作用[3].

但是, 在使用ISCSI服务时, 客户端需要通过一个固定的IP地址连接服务端. 当IP地址对应的服务端发生异常时, 客户端将不能正常使用存储服务. 经过研究发现, 针对这种情况, 业界普遍的一种解决方案是服务器冗余配置. 如图1所示, 方案使用冗余策略, 在客户端配置多路径选择、传输端实现路由器冗余、在服务器端实现ISCSI服务主从备份等方式实现ISCSI服务的高可用部署[4]. 而使用这种策略的原因与其ISCSI后端存储是传统存储阵列密切相关, 即所有的ISCSI服务部署在一个强大的Master存储服务器上, 数据同步备份到Slave存储服务器上. 当服务器故障时, 将所有服务迁移到Slave节点上.

许多文献基于该解决方案, 根据自身的需求与实际环境, 采用或修改部分策略, 使用不同的维护工具, 以实现ISCSI高可用. 如文献[4]通过改进心跳算法、实现故障接管和数据镜像同步等, 提高ISCSI的可靠性; 文献[5]使用网络块DRBD和PaceMaker工具维护

① 收稿时间: 2016-11-10; 收到修改稿时间: 2017-01-04

ISCSI服务, 以检测并迁移ISCSI服务.

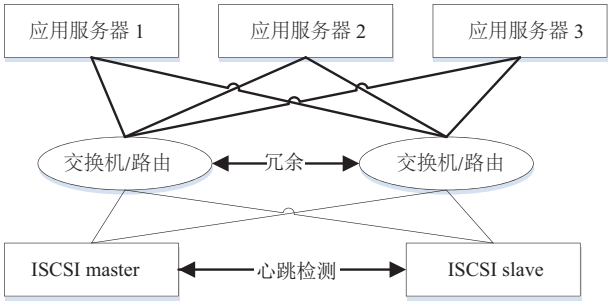


图1 ISCSI服务的IO路径及冗余策略

但在分布式存储系统中, 主从备份方式的ISCSI高可用部署并不能充分利用分布式存储带来的优势. 首先, 分布式存储系统管理的节点性能各异, 某一节点可能不足以支撑大量的ISCSI服务, 容易造成ISCSI服务节点频繁故障; 其次, 不能充分利用节点资源, 从而实现均衡负载. 然而, 针对分布式存储阵列, 业界仍然采用主从备份方式实现ISCSI服务的高可用, 暂时未发现基于ISCSI分布式集群架构相关的文献.

针对主从备份方式不能有效利用分布式节点资源的情况, 本文以Ceph[6]分布式存储管理系统为基础, 采用分布式思想设计并实现ISCSI服务集群. 本文设计的ISCSI集群充分利用节点资源, 将ISCSI服务均衡分布在各个存储节点上; 同时, 集群统一管理ISCSI服务, 自动监测节点状态, 实现服务故障迁移, 并保证上层业务I/O操作的连续性.

## 1 ISCSI高可用集群设计

Ceph是一个软件定义存储(SDS, Software Define Storage)的分布式管理系统, 其通过Paxos算法保证集群状态的一致性, 消除了单节点故障[7,8]. RBD是Ceph的块存储功能模块, 其性能优异, 可作为ISCSI服务的后端存储设备[9].

如图2所示为ISCSI服务的三层逻辑体系架构. 层级之间通过TCP/IP协议进行通信, 相对独立, 互不影响. 底层为物理存储层, 使用Ceph分布式集群可靠的存储服务. 中间层为ISCSI服务层, 提供高可用的ISCSI服务. 上层为用户层, 客户端通过ISCSI工具直连ISCSI Target服务, 实现远程操纵存储设备的目的.

### 1.1 负载均衡及节点选择策略

目前ISCSI服务的负载均衡一般通过多路径选择(multipath)实现. 这种方式需要在客户端进行复杂的配置, 极度依赖配置人员对于整个集群状态的掌握程度. 本文设计的ISCSI集群在服务部署和迁移时依据节点选择策略进行ISCSI服务部署, 从而将ISCSI服务均衡分布到集群节点上, 避免个别节点负载过高.

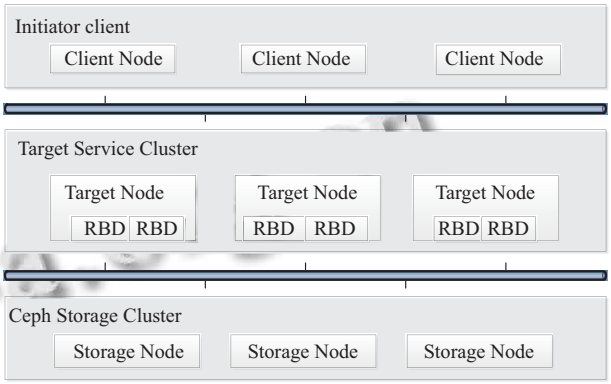


图2 分布式存储ISCSI高可用架构

如图3所示, ISCSI集群在部署ISCSI服务时, 主要通过predicate和priority两阶段进行节点的智能选择. 在predicate阶段, 主要判断节点是否能够通过如表1所示的硬性指标条件; 在priority阶段, 依据配置文件的策略配置, 使用对应的策略函数对符合predicate条件的节点进行优先值计算, 从而选择最符合的节点进行服务部署或迁移.

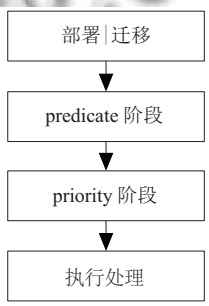


图3 节点选择策略流程

表1 predicate阶段判定的硬性指标

| 指标 | 描述 |
|---|---|
| 节点状态 | 节点是否处于正常工作状态 |
| 块设备状态 | 块设备是否未被占用 |
| IP端口状态 | IP和端口是否可以使用 |
| ISCSI状态 | ISCSI服务是否不存在 |
| 资源状态 | CPU、内存、网络带宽等是否满足ISCSI服务的运行标准 |

priority阶段支持三种选择策略: 最少ISCSI服务选择策略、Round Robin轮询选择策略和基于资源的优先级选择策略. 选择策略遵循如下前提定义:

1) ISCSI配置文件中的一个Target配置记作一个ISCSI服务;

2) $n$个分布式节点从小到大进行连续编号(从0开始), 编号$i$作为节点标记;

3) 假设一个ISCSI服务消耗服务器CPU、内存、网络带宽资源为$(x, y, z)$, 服务器总资源为$(a, b, c)$, 则可算出单个节点支持ISCSI服务的权重. 其计算公式为:

$$iscsi\_num_i = \min(a/x, b/y, c/z) \quad (1)$$

$$w_i = iscsi\_num_{\text{enable}}/iscsi\_num_i \quad (2)$$

其具体描述和使用场景如表2所示.

表2 priority的节点选择策略

| 节点选择策略 | 描述 | 使用场景和均衡效果 |
|---|---|---|
| 最少ISCSI服务 | 挑选ISCSI服务数量最少的正常节点, 如果节点的iscsi服务数量相同, 则选择编号最小的 | 场景: 每个节点性能相差不大的情况 效果: 在不发生故障情况下, 每个节点部署服务总数相差最多1个 |
| Round Robin | 从最小编号节点开始轮询, 若i为上次ISCSI服务部署的节点, 则选择i=(i+1)mod n作为服务部署节点. (如果节点故障, 则选择下一个有效节点) | 场景: 针对每个节点运行稳定且性能差不多的情况 效果: 有效降低易故障节点的ISCSI服务数量 |
| 基于资源的优先级 | 根据定义(3)计算每个节点支持的ISCSI服务总数和权重, 该算法基于权重选择节点进行服务部署 | 场景: 适用于节点性能间差距较大的情况 效果: 性能越好的节点部署的ISCSI服务越多, 节点服务占比大致相同 |

由于基于资源优先级部署策略较为复杂, 本文对其进行详细描述. 资源优先级策略依赖于“堆”数据结构, 其根据节点的权重$w_j$建立一个最大堆. 在ISCSI服务部署时, 遍历堆数组, 以$w_j$概率决定将ISCSI服务部署在该节点$j$上. 当选中部署节点后, 会导致该节点的权重$w_j$降低, 重新调整以该节点为根的子树, 使之成为堆. 当堆最大值$w_0$小于阀值threshhold时(比如0.5), 会导致首次命中的概率变小, 后续命中的概率变大, 不符合算法目的, 需要将概率随机输出反转, 使算法正确运行. 其算法伪代码描述如下所示:

```
// 最大堆数组A
for i=0 upto A.length
   w = A[i]
   if A[i] < threshold then
      w = 1 - A[i]
   choose = prand(w) // 根据w概率输出0或者1
   if choose then
        A[i] = iscsi_enable/iscsi_all
        MAX_HEAPIFY(A, i) //以i为根子树维护堆性质
```

针对实际物理环境, 用户通过配置文件指定priority阶段的选择策略. 集群通过读取配置文件参数, 使用特定策略选择, 从而优化ISCSI集群的每个节点的负载能力, 降低由于业务高负载引起的节点异常.

### 1.2 ISCSI高可用方案

本文的ISCSI高可用设计方案, 主要依靠故障节点发现和资源迁移(用户数据迁移、ISCSI服务迁移、IP迁移)两方面的设计保证服务的可用性和可靠性. 同时, 应用Raft[10]一致性算法来解决分布式集群中节点故障决策、数据存储一致性等各种决策问题.

#### 1.2.1 故障节点发现与数据存储

ISCSI服务节点状态通过心跳应答机制进行故障检测, 由Raft算法选举产生的leader节点作为裁决节点进行最终判定. 文献[10]详细描述了Raft算法在集群正常和异常情况下Leader节点选举与数据一致性维护的科学性原理.

本文利用Raft一致性算法少数服从多数的判定标准, 即只要超过半数节点(N/2+1个节点)的结果一致, 则leader节点就认为该结果是正确的这一准则, 来判断节点状态. 如果在一段固定时间内(如60秒), 某个节点被半数以上节点报告为异常, 则Leader认为该节点故障并踢出集群, 同时更新集群节点状态信息.

同时, 由于Leader节点的权威性, 本文设计的ISCSI集群配置数据的更改都只能通过Leader节点进行请求. 同时, 由Leader节点将改变的数据结果发送同步到各个节点上. Leader只有确保数据已经同步并写入到(N/2+1)个节点磁盘后, 即能够保证数据一致性后, 才会返回操作成功.

如图4所示, 在一个独立的物理节点中, 运行三个组件: agent、server和monitor.

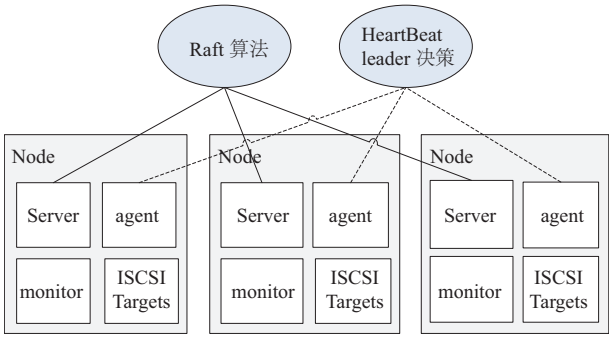


图4 ISCSI集群组件关系

Agent组件定时向所有节点发送心跳, 通过响应情况判断其他节点状态. 如果节点超时未返回响应, 则向当前的leader节点进行汇报. Leader节点接受所有节点agent的消息, 从而更新集群节点状态, 并将节点状态更新到server组件中. 当某个节点故障或者恢复后, leader设置该节点ISCSI服务迁移标识, 并应用1.1节中的迁移策略决定迁移的节点.

Server组件负责存储数据, 包括节点状态信息和ISCSI服务配置信息. 当server组件存储的数据发生变化时, 会向Leader进行报告, 由Leader节点判定该数据是否最新的, 是否能够同步到每个节点上, 并达到各节点数据新的一致性状态.

Monitor组件是ISCSI服务管理组件, 它定时向ISCSI集群获取节点的状态. 如果发现其所处的节点处于异常状态或者ISCSI集群超时未响应, 则认为节点本身存在异常, 然后将该节点上所有的ISCSI服务资源释放. 当monitor节点发现ISCSI服务需要迁移, 则通过ISCSI集群获取该节点的ISCSI配置, 并进入ISCSI服务迁移子程序.

### 1.2.2 服务资源迁移

ISCSI服务资源迁移总体上分为三个部分: 后端用户数据的迁移, ISCSI服务迁移和IP地址的迁移. 在迁移过程中, 首先获取待迁移节点的ISCSI服务配置文件, 根据配置文件将Ceph块设备映射到健康节点上, 然后重新部署ISCSI服务, 最后进行IP地址迁移. 服务迁移不仅包括故障节点的ISCSI迁移, 还包括节点重新加入集群时服务恢复的迁移.

该迁移过程主要由monitor组件负责, 其迁移过程伪代码如下所示:

```
status= request_status(leader) // 向leader请求节点状态
if status == “fail” then
    flag = can_transfer(cur_ip)
    if flag then
        transfer_all_service() // 服务作为整体迁移
    elif transfer_whatever then
        transfer_apart_service() // 服务拆分迁移
else if status == “recovery” then
    // 服务恢复, 先停止ip对应iscsi服务
    disable_iscsi_service(cur_ip)
    transfer_all_service()
```

在迁移决策中, 迁移节点的选择总体上依据节点选择策略进行. 值得一提的是, 故障节点上的所有ISCSI服务是作为一个整体迁移到同一个健康的节点上. 当任意节点都不能承受故障节点所有ISCSI服务时, 由transfer_whatever值判定是否将故障节点的服务拆分到各个节点上. 但是如前言所说, 用户是根据IP进行ISCSI服务连接的, 如果服务进行了拆分, 则需要用户重新通过ISCSI客户端工具进行连接. 本文设计的ISCSI在遇到这种情况下默认不进行任何迁移操作, 直至故障节点恢复正常. 同时, 可以通过配置文件将transfer_whatever设置为True来改变集群行为.

拆分迁移将服务一个个独立地重新部署在ISCSI集群节点中, 与新建服务流程大致相同, 在此不做详细描述. 服务整体迁移分为如下几个步骤:

首先进行用户数据的迁移. 如图5所示, RBD块由多个objects组成, 每个object对应着一个PG组, 通过PG与OSD的映射关系, 将数据实际存储在OSD上. 当数据进行迁移时, 只需从OSD重新获取数据并在健康节点上组织成RBD块设备, 就完成数据的迁移. 当故障节点恢复时, 有可能会对同一个块设备进行数据写入操作, 这样是会破坏块设备数据的. 因此, monitor组件将检测故障节点并进行故障节点的资源释放, 保护后端设备数据安全.

其次部署ISCSI服务. Monitor组件通过向Leader节点请求IP地址对应的ISCSI服务配置文件, 然后导入到当前节点的ISCSI配置程序. ISCSI配置程序根据配置文件信息, 重新生成ISCSI服务, 至此完成ISCSI服务迁移操作.

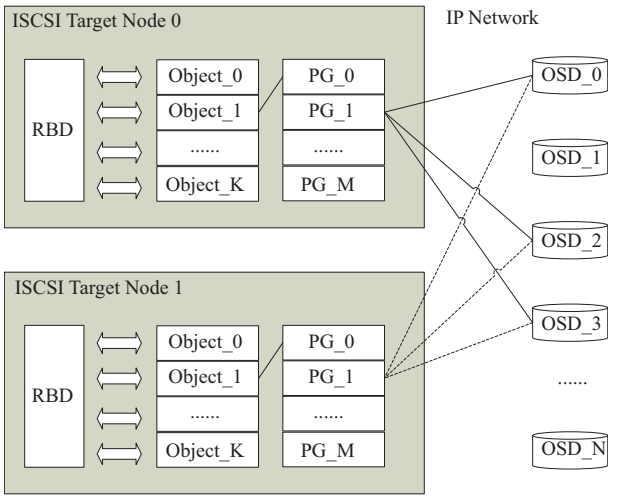


图5 Ceph RBD存储迁移示例

最后进行IP地址迁移. 每个服务节点对应一个虚拟IP地址. ISCSI集群动态绑定虚拟IP、ISCSI服务、存储节点三者, 并将其关系保存在server组件中. 只有当IP对应的节点发生异常时, IP地址才会进行迁移. 此时, 故障节点IP将临时迁移到某个指定健康节点上; 当故障节点重新加入ISCSI集群, monitor组件则会检查IP对应的节点是否与server组件记录的信息匹配. 如果发现IP对应的ISCSI服务只是临时迁移到某个健康节点上, 则monitor将释放健康节点上该IP对应的ISCSI服务, 并将IP以及其关联的ISCSI服务迁移回源节点上.

### 1.3 业务连续性管理

如图6所示, 客户端连接ISCSI服务主要是依靠TCP/IP网络, 只要维持服务对应的IP地址不变, 客户端在ISCSI迁移成功后重新建立Session会话, 并根据上层应用逻辑继续I/O业务流程. 虽然ISCSI服务连接是全程保持会话状态的, 但是当ISCSI节点异常或者集群服务迁移过程中, 会造成当前会话状态的丢失. 等到迁移完成后, 即使ISCSI客户端建立新的会话连接, 仍然很容易造成数据的不一致性.

为了保持ISCSI服务的连接会话, 可以使用分布式缓存服务器memcached存储ISCSI的回话状态. 如图7所示, 每次I/O操作都将记录在memcached服务器中, 直到写操作实际落盘后, 才删除memcached里的相关操作. 当服务迁移成功后, ISCSI服务端通过memcached遍历会话日志, 并重新向client请求未完成的I/O操作.

相比使用memcached这种实现复杂的机制, 一种更加简单有效的方法是直接关闭ISCSI服务以及RBD Cache, 从而保证会话数据的一致性. 虽然这样做会降低部分I/O性能, 但由于RBD是分布式存储设备, 关闭缓存对其性能影响有限, 而且几乎不需要实现成本.

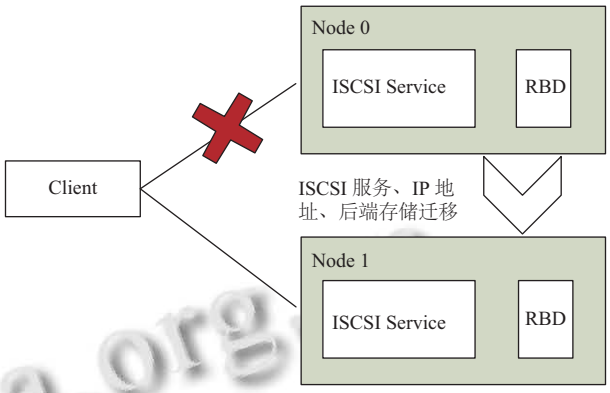


图6 ISCSI客户端重新建立会话

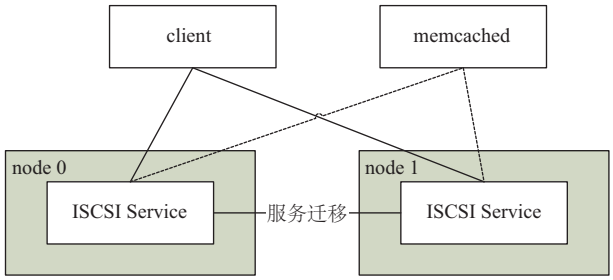


图7 利用memcached保持会话

如果对ISCSI性能要求不高的话, 本文建议直接关闭缓存. 在ISCSI集群的实现验证阶段也是通过关闭缓存的方式保证业务的连续性.

## 2 ISCSI高可用集群实现与验证

Consul是一个使用raft算法的通用服务框架, 集成了服务的注册与发现、节点故障检测和key-value数据库等功能. 通过Consul接口可以快速实现检测和存储ISCSI服务状态. Consul已经实现了类似server和agent组件的功能. 在此基础上实现monitor组件, 定时访问Consul集群, 请求当前的leader节点信息和ISCSI集群节点状态, 基于节点状态进行逻辑处理. 在ISCSI服务部署及迁移阶段, 只有leader节点的monitor组件才有权利进行节点选择, 然后更新到Consul的Key-Value数据库中. 从节点的monitor组件根据对应数据库信息进行实际服务迁移流程.

因此, ISCSI高可用集群的实现集成了Consul框

架、monitor组件以及ISCSI服务, 应用了Ceph分布式存储集群. 其实现部署图如图8所示.

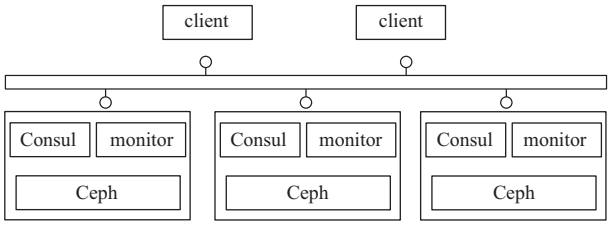


图8 ISCSI集群实现部署

每个节点基础配置如下: Intel Xeon E5系列CPU、3*2T的HDD硬盘、128GB内存、万兆网卡. client为普通的计算机, 可通过网络连接服务集群. 在ISCSI高可用集群的验证环境中, 集群节点间通过网络进行消息通信, 选用最少ISCSI服务策略作为ISCSI服务部署和迁移的目标节点选择策略, 同时关闭Ceph和ISCSI服务缓存, 以保证业务的连续性.

本文主要通过切断网络通信的方式, 模拟某个节点故障的情况, 从而验证ISCSI高可用集群的服务迁移功能.

表3 ISCSI高可用集群功能验证

| 验证方案 | 平台 | 验证结果 | 恢复间隔 |
|---|---|---|---|
| 断开Client端连接的服务节点的网络 | Linux Windows | 服务按照策略迁移到特定节点上, 期间client业务中断 | 约5 s |
| 将网络失败的服务节点恢复正常 | Linux Windows | iscsi服务迁移回源节点, 业务几乎不受影响 | 约2 s |

由于ISCSI高可用集群和Ceph集群是相互独立的, 而ISCSI集群需要等待Ceph集群恢复成功后, 才能进行恢复过程. 所以总故障恢复时间为:

故障恢复总时间=Ceph集群恢复时间+ISCIS集群恢复时间

ISCSI集群恢复时间=故障发现时间+ISCSI服务迁移时间

通过表3所示的验证结果, 可以认为ISCSI集群的服务迁移是有效的. 通过对比故障写入数据与源数据, 发现两份数据的消息摘要相同, 证明了写入业务的连续性.

## 3 结语

本文通过分析ISCSI服务在分布式存储环境下的状况, 提出了ISCSI高可用集群的设计方案. 该ISCSI高可用集群支持负载均衡策略选择, 应用raft一致性算法实现服务故障迁移等高可用功能. 同时, 对于如何保持无状态集群中的会话给出一套切实可行的方案. 最后, 通过开源Consul服务框架实现ISCSI高可用集群, 并进行了功能验证, 表明了ISCSI高可用集群的有效性与可用性.

## 参考文献

1 Satran J, Meth K, Sapuntzakis C, *et al*. Internet small computer systems interface (iSCSI).The Internet Society, 2004.
2 Aiken S, Grunwald D, Pleszkun AR, *et al*. A performance analysis of the iSCSI protocol. Proc. 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, 2003. San Diego, CA, USA. 2003. 123–134.
3 Chadalapaka M, Satran J, Meth K, *et al*. Internet Small Computer System Interface (iSCSI) Protocol (Consolidated). RFC7143,IETF, 2014.
4 Haas F. Replicate everything! highly available iSCSI storage with DRBD and pacemaker. Linux Journal, 2012, 2012(217): Article No.5.
5 王施人, 陕振, 张淑萍, 等. 基于iSCSI存储集群的设计与实现. 计算机工程与设计, 2010, 31(11): 2598–2601.
6 Weil SA, Brandt SA, Miller EL, *et al*. Ceph: A scalable, high-performance distributed file system. Proc. 7th Symposium on Operating Systems Design and Implementation. Seattle, Washington, USA. 2006. 307–320.
7 Lamport L. Paxos made simple. ACM SIGACT News (Distributed Computing Column), 2001, 32(4): 51–58.
8 Van Renesse R, Altinbuken D. Paxos made moderately complex. ACM Computing Surveys (CSUR), 2015, 47(3): Article No.42.
9 Gudu D, Hardt M, Streit A. Evaluating the performance and scalability of the ceph distributed storage system. Proc. 2014 IEEE International Conference on Big Data (Big Data). Washington DC, USA. 2014. 177–182.
10 Ongaro D, Ousterhout J. In search of an understandable consensus algorithm. Proc. 2014 USENIX Conference on USENIX Annual Technical Conference. Philadelphia, PA, USA. 2014. 305–320.