分布式云灾备平台搭建及性能分析①

杨 涌, 刘磊锋, 陈勇源, 李子介

(中国科学院重庆绿色智能技术研究院 高性能计算应用研究中心, 重庆 400714)

摘 要: 云存储服务, 作为云计算的衍生产物, 目的是为网络海量数据的存储提供有效的解决方案, 节约存储成本和系统资源, 提供一个完善的备份、容灾的数据中心, 并能够保证数据安全性、容错性. 现阶段云灾备模型局限于有限的网络位置, 使用虚拟化技术, 依托本地服务器实现, 与传统云灾备模型不同, 介绍了一种基于 DHT 的云灾备模型, 可适用于广域网的、普适的数据级灾备解决方案; 最后, 在本地云计算集群中对该方案进行模拟, 验证该模型的可行性.

关键词: 云存储; 云计算; 灾备; DHT; 分布式存储; 纠删码

Distributed Cloud Disaster Recovery Platform Structure and Performance Analysis

YANG Yong, LIU Lei-Feng, CHEN Yong-Yuan, LI Zi-Ji

(Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China;)

Abstract: Cloud storage service is derivative products of cloud computing intended to provide an effective solution for mass data storage networks, while save storage costs and system resources. It can provide a complete backup, disaster recovery data center, in order to ensure data security, fault tolerance and efficient storage functions. At present, cloud disaster recovery models are using virtualization technology, which relies on the local server implementations. This paper introduces a model of DHT-based cloud disaster recovery, which is applicable to WAN universal data-level disaster recovery. Finally, in order to verify the feasibility of the model, the scheme is simulated in the cloud computing cluster.

Key words: cloud storage; cloud computing; disaster recovery; DHT; distributed storage; erasure code

云灾备是利用虚拟化的、易于扩展的云存储资源池提供数据级和应用级容灾的解决方案. 云灾备是灾备领域的一个新兴概念, 能为企业提供一套行之有效的存储备份解决方案. 云灾备能够提供一种服务, 由客户付费使用, 灾备服务提供商将根据客户需要提供针对性的存储备份服务模式. 采用这种模式, 客户可以利用服务提供商的优势网络资源、技术资源、丰富的灾备项目实施经验和成熟的运维管理流程, 快速实现自身的灾备目标, 降低运维成本和工作强度, 大幅度降低建设成本.

云灾备的基础问题是数据存储, 即在新兴的云存储架构上储存数据, 而不是传统本地存储, 因此, 需要引入适合云存储架构的路由算法来解决数据传输和存储等问题. DHT(Distributed Hash Table, 分布式哈希表)算法就是使用分布式哈希函数来解决结构化的分布式存储问题[1]. 分布式哈希表实际上是一张很大的散列表, 每个节点被分配给一个属于自己的散列块, 并成为这个散列块的管理者. DHT 及其发现技术为 P2P 网络中资源的组织与查找提供了一种全新的算法思想, 在对等网络结构下的数据定位和查找等方面得到了广泛应用. 目前, 典型的 DHT 协议包括有美国 MIT 的 Chord、UC Berkeley 的 Tapestry 和 CAN、纽约大学的 Kademlia; 其中, Kademlia 协议根据对等网络中两个节点标志的异或(XOR), 建立起一种全新的 DHT 拓扑结

① 基金项目:重庆市“121”科技支撑示范工程-知识产权交易服务协同创新与关键技术攻关项目(cstc2012jcsf-jfzhX0006)
收稿时间:2014-05-23;收到修改稿时间:2014-06-23

构, 相比其他算法, 大大提高了路由查询速度, 已被许多 P2P 软件使用[2-5].

目前研究云灾备的国外项目主要是 Oceanstore 项目、PASIS 项目、POTSHARDS 项目等. 其中 Oceanstore 项目系统利用加密的方式保证存储数据和端到端通信的机密性,对用户进行访问的数据使用复制的方式制作数据冗余提高数据的可用性和访问效率,并使用拜占庭容错协议进行数据的拜占庭容错设计.

PASIS 是由卡内基梅隆大学的并行数据实验室的 Gregory Ganger 等人从 2000 年开始的可容忍拜占庭错误的可生存存储系统项目. 该系统认为整个系统中的客户端和服务器端都是不可信的,客户端有可能在数据分发时发送错误的数据,而存储节点也面临崩馈错误和拜占庭错误带来的影响. 传统的基于复制的拜占庭容错协议无法解决崩馈错误对数据恢复带来的影响, 因此, Gregory Ganger 等人利用纠删码设计了一种 block 级别的能够容忍拜占庭错误的分布式数据存储协议,通过该协议可以有效的避免不可信的客户端和存储服务器对数据恢复带来的影响.

POTSHARDS 项目是由加利福尼亚大学圣克鲁兹学院的 Mark W. Storer 和 Kevin M. Greenan 等人使用秘密共享方案开发的一种新的分布式存储协议,该协议使用两层秘密共享的方法对数据进行处理,通过这种方式,信息窃取者无法定位某一个数据对应的秘密份额,而恶意的节点也难以通过获取足够的秘密份额的方式恢复用户的数据.

国内的项目如 Upstore 是由北京大学的田敬和代亚非等人从 2004 年始研究的一个具有开放框架的存储平台. Upstore 期望能够在不可靠的存储节点上构建一种可靠的存储协议,它采用了副本和纠删码两种冗余策略保证数据的可用性和可恢复性. 田敬和代亚非等人在 Upstore 项目的研究中提出了一种新的纠删码设计(SEC)方案, 通过在纠删编码的同时引入数据加密方案,能够同时保证数据的可用性和私密性. S2-S0S 是由中科院荆继武等人提出的一种用于存储敏感数据的拜占庭容错存储系统. S2-BQS 使用 PSS(Perfect Secret Sharing)方法和拜占庭 Quorum 系统相结合保证数据的机密性,可靠性和可恢复性. 与 S3-BQS 相比, 在故障节点数目比较少时, S2-BQS 能够使用较小的计算负载,保证所有被拆分的数据块的信息安全.

1 云灾备关键技术研究

1.1 高性能可扩展重复数据删除技术

在提高重复数据删除性能方面, 可以使用减轻磁盘瓶颈技术. 在重复数据删除系统中, 为了节约成本, 一些系统仅具有少量的内存, 因而不能支持所有的数据索引一次性地进入内存进行检测, 从而导致了大量的磁盘访问, 这成为性能下降的最主要因素. 针对这种情况, Data Domain 重复数据删除文件系统中采用了减轻磁盘瓶颈的 3 种技术, 它们分别是: (1)摘要向量, 一种内存中紧凑的数据结构, 用于辨别新的块. (2)基于流的块排列, 一种用于提高磁盘上的被连续访问块的访问局部性的数据排列方法. (3)局部性保持, 保持了重复块的指纹值的局部性从而达到缓存的高命中率. 应用这 3 种技术, 可实现高吞吐率、低开销的相同块删除存储系统.

1.2 云存储安全技术

在云灾备应用环境中, 用户的数据存放在由云服务提供商管理和维护的服务器上, 不再受用户的直接控制, 增加了数据的潜在风险. 可以说数据安全已成为限制云灾备在企业中进一步推广和应用的关键因素, 为了数据安全考虑, 完整性检查和持有性证明技术应用而生. 完整性检查是指检查从 CSP 读回的数据和之前写入的数据是否一致, 即数据是否被篡改. 完整性检查是写文件时使用某种单向哈希函数对数据计算得到一个哈希值, 存放在本地可靠存储中. 读文件时进行同样计算得到哈希值并和本地的哈希值比较. 为了降低完整性检查的复杂度, 可以采用 Merkle 哈希树的方法, 将文件分成若干数据块, 最底层的树叶节点对应数据块的哈希值, 次底层节点是每两个哈希值的哈希值, 由此逐层递归构造出一个二叉树, 根节点对应最终的哈希值. 此时检查一个数据块完整性的复杂度由 O(n)降为 O(log n), 其中 n 为数据块个数[6].

上述方法可以验证 CSP 返回的数据的完整性. 然而在很多情况下, 用户需要知道其数据是否始终由 CSP 完好保存并可获取. 当用户在云中存储大量数据时, 如果用户每次将所有数据下载到本地, 用上述完整性验证方法检查数据是否完好, 这种做法显然是不可行的. 为此研究者提出了持有性证明, 即 CSP 可以通过某种方法向用户证明其仍然完好的持有用户数据, 并且数据是可获取的, 而不需要提供完整数据.

这些方法可以分为两类: 基于 RSA 公钥密码算法的和基于对称密码算法的. 基于 RSA 的方法利用了基于 RSA 的哈希函数的同态性, 该方法的最大优点是允许用户发起无限次的检查, 缺点是由于需要进行有限域上以文件数据块为指数的指数运算, 计算开销较大, 尤其在文件预处理阶段. 基于对称密码算法的持有性证明的基本思想是首先将文件加密并用纠错码编码, 然后在编码后的文件的一些随机位置插入和文件数据不可区分的“岗哨”.

1.3 操作系统虚拟化技术

除了数据级的灾备, 还应提供系统级的灾备. 即在将数据复制到云端的同时, 也将受保护的应用程序的状态复制到云端, 当灾难发生时可以立即切换到云端的应用程序运行, 保证业务连续性. 系统级灾备是通过操作系统虚拟化和检查点实现的. 检查点用来捕获进程某一时刻的运行状态, 从而实现进程迁移. 进程迁移既可以是用户应用程序进程到云灾备中心的迁移, 也可以是云灾备中心内部的虚拟机池间进程迁移, 以实现根据前端用户的需求自动地调节灾备服务提供商有限的硬件与软件资源, 动态地、弹性的反应前端业务对灾备的需求.

当程序因故障中断, 如果不能保留其中间运行状态, 恢复后从头运行将会带来极大的消耗. 检查点技术能够解决这个问题. 通过保留各个进程的运行状态, 恢复时能够复原到最近一次保留的数据映像.

传统的检查员机制是基于库的检查点机制. 例如以静态库的形式实现, 或通过加载动态链接库来追踪程序运行过程中的数据变化. 也有一些检查点机制实现于内核级别甚至硬件级别. 例如通过在文件系统层之上引入一个中间层来实现保留文件系统状态的检查点机制; 或者借助 Fuse 内核模块实现的支持检查点机制的文件系统, 通过 Fuse 侦测、拦截内核级别的文件系统操作并将控制权传递给用户, 从而能够在用户空间对文件系统状态进行保留.

随着操作系统虚拟化技术的发展, 基于虚拟容器的检查点技术也得到了很好的应用. 虚拟容器是通过系统虚拟化技术构建出来的一个进程运行的较独立的上下文环境. 虚拟容器检查点技术能够有效保护容器内运行的应用程序和服务而不需要对应用进行修改.

2 基于DHT的分布式云灾备模型设计

本文在通过深入研究与分析云计算和分布式存储相关领域工作的基础上, 设计提出了一种基于DHT的分布式云灾备模型, 该模型包括 4 层, 其中分为:物理设备层、虚拟化层、Kaemlia 路由协议层和应用数据管理层. 基于 DHT 的分布式云灾备模型构架如图 1 所示[7-9].

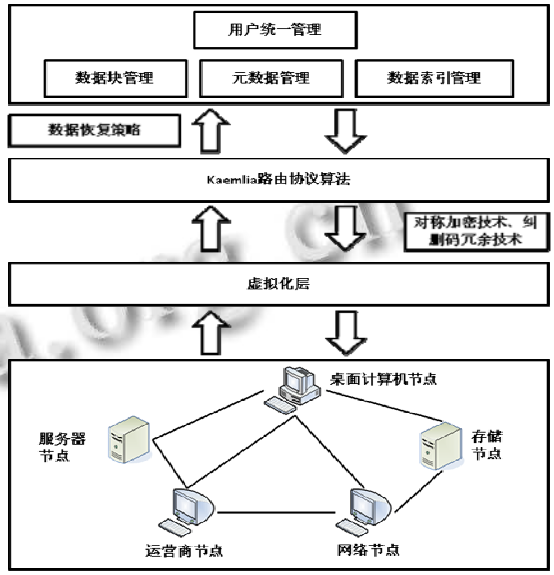

图 1 基于 DHT 的分布式云灾备模型构架

(1)物理设备层

主要负责存储节点的物理规划和布局, 从硬件角度解决数据存储资源的收集和规划. 理论上隶属因特网范围内的所有计算机存储系统都能成为该灾备云网落的节点之一. 同时, 该层还需要配置一定带宽的通信网络, 满足大流量数据传输策略. 本层是该云灾备模型的基本组成元素, 贡献给整个系统的是存储空间、计算资源以及物理通信网络, 除此之外该层提供重复数据删除功能.

(2)虚拟化层

虚拟化层采用硬件虚拟化和操作系统虚拟化技术, 将物理设备层服务器统一虚拟整合为一个服务资源池, 根据各个业务系统的需求, 为其分配适合的 CPU、内存和存储资源. 就相当于多台服务器同时运行了, 利用率大大提高. 虚拟化层虚拟出多台节点, 供 Kaemlia 路由协议层查找调度.

(3)Kaemlia 路由协议层

采用 Kaemlia 结构化路由算法, 实现对松散网络节点资源的结合和利用, 在保证系统底层存储资源物理互通的前提下, 实现逻辑上 DHT 网络的覆盖, 其中路由算法实现对存储节点的快速查找等. 同时, 该层

也将负责使用对称加密技术实现存储加密和取数据解密的功能. 通常使用分组密码或者序列密码对上层已分块的数据包进行密钥控制加密. 此加密方式的简单、高效、安全等特性是该系数据存储的核心安全机制. 除此之外, 该层还要负责数据的纠删冗余性检测, 以达到一定的容错性. 传统的用于分布式系统的纠删码, 如 RS 纠删码、阵列纠删码、LDPC 纠删码等都可以作为该云灾备系统的选择[10-12].

(4)应用数据管理层

在整个系统中位于结构的最上层, 包含数据管理、用户管理、数据分块、元数据管理和恢复策略等灾备核心业务. 首先提供给云灾备用户文件存储服务的接口和认证方式, 预估给每个用户的云存储空间为 100GB 或者更大, 主要依据底层存储总资源的大小和用户的信用等级, 并能根据需要动态调整; 然后是对用户个人信息的认证和数据前端加密, 保证多用户各自的独立目录空间, 给予必要用户数据安全性; 最后是针对数据的分块操作, 利用高响应缓存来管理副本和元数据, 文件数据按照固定大小分块加密封装传至下层 DHT 网络的各个存储节点. 在完全副本冗余和删冗余等技术的协助下实现数据块高效的索引和存取操作. Kaemlia 路由协议层将数据的路由信息发送给该应用数据管理层, 应用数据管理层集中管理这些元数据信息. 对元数据管理使用动态子数分割管理, 具体如图 2 所示.

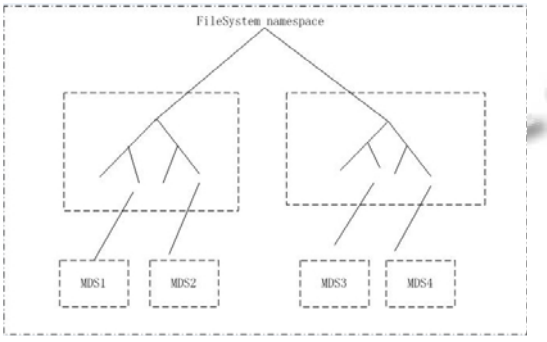

图 2　元数据管理使用动态子数分割管理示意图

该管理策略是: 动态地将缓存的元数据信息分布到一组节点上, 每个 MDS 统计自己目录条目的热度, 使用一个计数器来统计, 当条目被访问时, 计数器加 1, 同时也会影响其祖先节点, 为每个元数据服务节点(MDS)提供一个表述负载均衡分布情况的权值树. 定期检查MDS的负载, 需要时迁移部分合适大小的子树以维持系统的负载均衡. 元数据操作是采用加锁机制, 保证数据的一致性. MDS 使用日志更新数据日志记录, 这样能够在一个 MDS 失效后帮助另一个节点恢复失效节点临界区的信息.

3　平台搭建及性能测试

由于目前的实验条件限制, 针对基于 DHT 的分布式云灾备模型的设计思想, 在本地云服务器集群搭建了一套灾备测试平台, 用于验证该灾备模式及实现思想的可行性和可用性.

3.1　测试平台配置与搭建

由于目前的实验条件限制, 采用虚拟化技术在原有的云计算节点上虚拟出若干个装有不同操作系统的虚拟服务器, 用于模拟分布式云灾备模型中提到的物理设备层; 几个云计算节点作为转发节点, 转发节点之间实现负载均衡, 每个转发节点根据 Kaemlia 路由协议选择不同的虚拟服务器进行文件块转发.

该测试平台设备配置如下:

① 云灾备平台服务器: 采用曙光 A620r-G, 操作系统: SUSE 11 SP2 XEN; 应用软件: 元数据和客户端.

② 云计算管理节点: 采用曙光 A620r-G; 操作系统: SUSE 11 SP2, 应用软件: 转发软件.

③ 客户端: 联想启天 M5659; 操作系统: SUSE 11 SP2 XEN 应用软件: 灾备云平台客户端软件.

④ 性能测试客户端: Acer 星锐 4752; 操作系统: Windows 7; 应用软件: Spotlight.

测试网络拓扑如下图 3 所示.

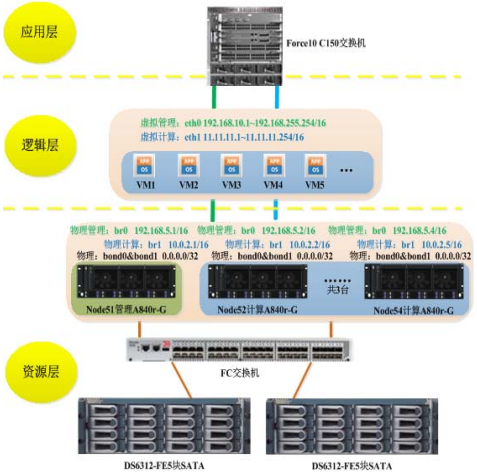

图 3　测试网络拓扑图

3.2 测试方法

本测试在本地云存储环境下, 根据需求将本地云存储资源虚拟化为四台不同操作系统的存储节点; 然后, 数据发送端将数据发送给指定的存储节点, 由于目的存储节点可能会处于不同的工作状态(开机、关机), 因此, 按实现实际判断存储位置进行数据传输, 在此过程中, 测量云灾备系统的工作性能:

① 响应时间(按数据量累进): 通过云平台服务器, 传输数据至储存服务器的平均响应时间.

② 文件备份性能.

③ 文件还原性能.

3.3 测试结果

① 响应时间

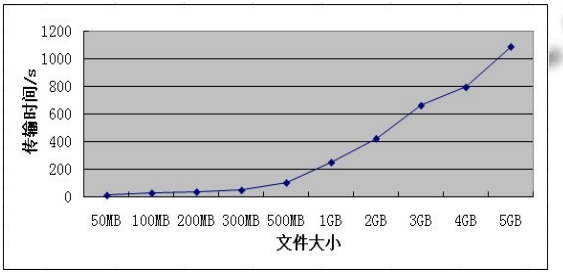

图 4 数据传输响应时间

从图中可以看出随着数据量增大, 传输时间基本呈线性增长, 符合正常逻辑.

② 文件备份

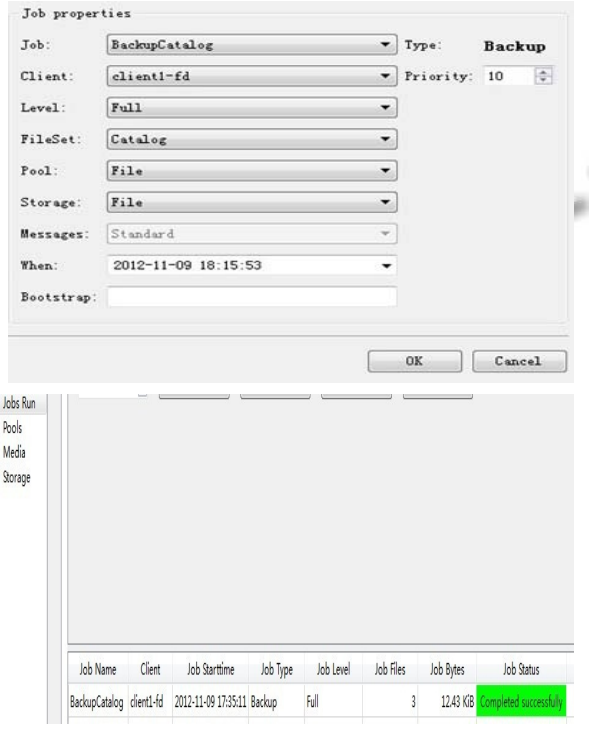

图 5 备份界面

③ 文件还原

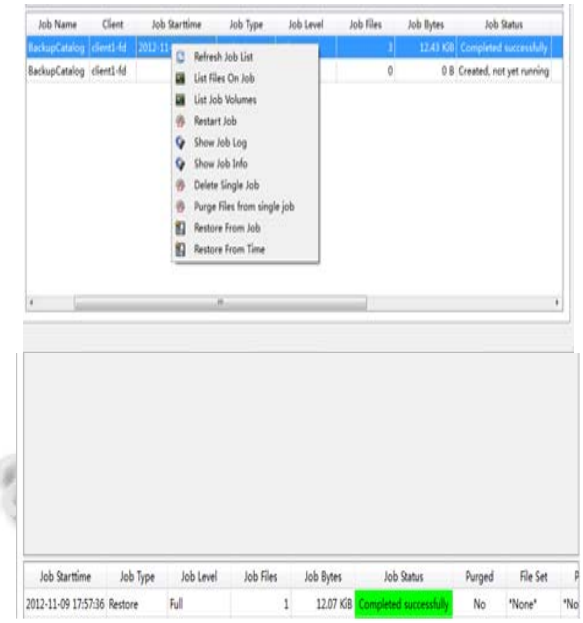

图 6 文件还原界面

从图 5、图 6 所示, 在云服务器集群搭建了一套灾备测试平台已经基本具备了灾备系统需要具备的备份和还原功能.

4 结语

本文通过系统介绍一种分布式云灾备平台架构, 并在本地云服务器集群上进行系统搭建和测试, 通过测试, 系统基本达到文件的备份和还原功能, 在底层存储节点宕机时, 路由协议层仍能有效地找到其它存储节点进行正常的备份. 不足之处是目前系统的存储节点之间没有冗余, 当已经备份的文件所在的存储节点坏掉后, 很难进行还原, 下一步准备针对这个问题进行研究和完善.

参考文献

1 Eric R. Introduction to distributed Hash tables. IETF-65 Technical Plenary. 2006.

2 孙知信,骆冰清,陈亚当,卜凯.一种基于多维 DHT 空间映射的 P2P 安全拓扑方案.中国科学,2013,43(3):343–360.

3 Dabek F, Li J, Sit E, Robertson J, Kaashoek F, Morris R. Designing a DHT for low latency and high throughput. Proc. of NSDI. SanFrancisco, USA. 2004.

4 陈贵海,吴帆, 李宏兴,邱彤庆.基于 DHT 的 P2P 系统中高可用数据冗余机制.计算机学报,2008, 31 (10).

5 杨楠.基于Kademlia的P2P网络资源定位模型改进[学位论文].武汉:湖北工业大学,2010.
6 陈钊.基于云灾备的数据安全存储关键技术研究[学位论文].北京:北京邮电大学,2012.
7 Reese G. Cloud Application architectures: Building applications and infrastructurein the cloud. O'Reilly Media, 2009.
8 冯丹.网络存储关键技术的研究及进展.移动通信,2009, 33(11):35–39.
9 Plank JS. Erasure codes for storage applications. 4th Usenix Conference on File and Storage Technologies, 2005.
10 陶钧,沙基昌,王晖.基于 Erasure Code 的分割文件 P2P 存储结构设计.国防科技大学学报,2008,30(6):57–62.
11 彭荣华.基于 DHT 的存储系统中纠删码技术研究.西安电子科技大学,2013.
12 Plank JS, Ding Y. Note: Correction to the 1997 tutorial on reed-solomon coding. Software, Practice & Experience, 2005, 35(2).