计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(11):96−102 [doi: 10.15888/j.cnki.csa.006659]

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

面向 OpenStack/Ceph 的虚拟机备份系统研究①

杨皓森[1], 胡晓勤[1], 黄传波[2]

[1](四川大学 计算机学院, 成都 610065)
[2](成都云祺科技有限公司, 成都 610041)
通讯作者: 胡晓勤, E-mail: huxiaoqin@scu.edu.cn

摘　要: 针对 OpenStack 面临的虚拟机容灾备份问题, 提出了一种基于 Ceph 存储快照的虚拟机容灾备份系统, 备份时对虚拟机存储在 Ceph 中的磁盘生成快照, 再根据备份要求计算有效数据或者变化数据, 保存虚拟机的配置信息以及磁盘数据, 恢复时自动创建相同配置的虚拟机并将当前快照点的磁盘数据恢复到该虚拟机中. 实验表明, 该方法比 OpenStack 的快照备份方法能有效节省备份时间和存储空间, 并且可以实现后者不具有的增量备份、多磁盘备份等功能.

关键词: 云计算; 虚拟化; OpenStack; 容灾备份; Ceph

引用格式: 杨皓森,胡晓勤,黄传波.面向 OpenStack/Ceph 的虚拟机备份系统研究.计算机系统应用,2018,27(11):96–102. http://www.c-s-a.org.cn/1003-3254/6659.html

Virtual Machine Backup System for OpenStack/Ceph

YANG Hao-Sen[1], HU Xiao-Qin[1], HUANG Chuan-Bo[2]

[1](College of Computer Science, Sichuan University, Chengdu 610065, China)
[2](Chengdu Vinchin Technology Co. Ltd., Chengdu 610041, China)

Abstract: Aimed at virtual machine disaster backup and recovery problem faced by OpenStack, the authors design and carry out a virtual machine backup system based on Ceph storage snapshot. During backup process, generate a snapshot of the virtual machine disk stored in Ceph, and according to the backup requirement, calculate disk valid data or modified data, save the config information of the virtual machine and disk data. During recovery process, automatically create the same configured virtual machine, and restore the data of current snapshot point to the corresponding disk. The experimental results show that proposed system can effectively save backup time and storage space compared with OpenStack's snapshot backup method, and implement the functions of incremental backup and multi-disk backup that the latter does not have.

Key words: cloud computing; virtualization; OpenStack; disaster decovery; Ceph

近年来, 虚拟化与云计算的浪潮席卷了整个 IT 行业, 引领了 IT 基础设施的发展以及产业的革新. OpenStack[1]是目前使用最为广泛的开源云计算平台, 可帮助企业实现构建自己的云基础架构服务. OpenStack 可利用多种虚拟化平台提供虚拟机服务, 如 Vmware、KVM[2]等, 虚拟机在 OpenStack 中会作为云主机展示给用户.

Ceph[3]是一种性能优秀、服务稳定且可扩展的开源分布式存储系统, 作为软件定义存储 (SDS)[4]领域的代表, 可以整合多种传统存储方式并进行统一配置管理, 与 OpenStack 形成了良好的搭配, 也是其目前使用最普遍的块存储方式.

① 收稿时间: 2018-04-27; 修改时间: 2018-05-17; 采用时间: 2018-05-24; csa 在线出版时间: 2018-10-24

根据 2017 年 4 月的 OpenStack 用户调查报告显示, 拥有的虚拟机数量 100 台以上的环境比例高达 74%[5]. 由于用户大量的数据和业务运行在虚拟机之上, 保障虚拟机的数据安全成为了 OpenStack 面临的迫切问题[6].

虚拟机备份与简单的快照不同, 它要求能保存虚拟机任意快照点的数据和状态, 可在本地或异地存储这些数据, 在生产环境发生灾难或者人工操作失误时, 能够利用原数据进行恢复, 降低损失. 其它大型虚拟化厂商, 如 Vmware、Redhat、XenServer 等, 市面上都有针对自身产品的虚拟机备份方案, 而 OpenStack 由于其作为云计算平台的复杂性, 并且没有完善相应的接口, 因此在虚拟机备份功能上一直进展缓慢, 其快照备份功能, 对于有具体环境备份需求的用户来说, 功能过于单一, 且时间和存储空间开销较大, 因此在实际部署中并不适用.

为此, 本文提出一种面向 OpenStack/Ceph 的虚拟机备份系统, 基于 Ceph 存储快照, 计算出虚拟机磁盘对象的有效数据或变化数据区域, 读取磁盘数据并按快照点的顺序保存至备份服务器, 同时保存原虚拟机的硬件配置、元数据等信息; 在恢复时, 创建一个配置相同的新虚拟机并将所选快照点的数据恢复到新的磁盘. 实验结果表明, 该系统比 OpenStack 的虚拟机快照备份功能, 可有效降低备份时间, 节省备份数据存储空间, 可以实现后者不具有的增量备份[7]等功能, 同时满足对虚拟机根磁盘、临时磁盘、挂载云硬盘的数据备份. 该方法不需修改 OpenStack 原生内容, 不会对用户的生产环境造成影响.

1 相关研究

1.1 OpenStack 现有备份功能

OpenStack 是基于多个模块协同工作的云计算平台, 其内部接口参照亚马逊 AWS[8]. OpenStack 虚拟机的磁盘数据保存于根磁盘 (Root Disk)、临时磁盘 (Ephemeral Disk) 和挂载的云硬盘 (Volumes).

OpenStack 现有的的虚拟机快照备份功能是对虚拟机根磁盘进行转换和拷贝, 不支持增量备份, 备份时间长, 冗余数据多, 虚拟机的临时磁盘、挂载的云硬盘无法得到有效备份. 新的虚拟机只能利用上传的根磁盘镜像创建, 一旦出现灾难或者人工失误操作, 容易造成大量数据丢失. 而 OpenStack 提供的 Cinder-backup 服务目前只能实现对云硬盘的备份且有诸多限制条件, 也无法满足对虚拟机整机的有效保护.

1.2 Ceph RBD 快照

Ceph 是一个多节点的分布式系统, 提供统一的存储访问接口. Ceph 的节点可分为 Monitor 节点与 OSD 节点, OSD 节点用于存储和查询对象, Monitor 节点用于维护集群成员的状态.

按照模块划分, Ceph 最底层模块是 RADOS (Reliable, Autonomic Distributed Object Store), 通过 CRUSH 算法[9]保证数据均衡存储于各个 OSD 节点. 在 RADOS 之上有多个模块对其功能进行了封装与拓展, 其中 OpenStack 主要使用到的模块为 Ceph RBD (Reliable Block Device), 即 Ceph 的块存储服务[3].

以 Ceph 为块存储后端的 OpenStack 环境, 每个磁盘均对应一个 RBD 对象, 磁盘格式采用 RAW[10], RAW 格式磁盘在 Ceph 中只保存已划分空间部分, 节省了存储空间. Ceph RBD 支持对块设备生成快照[11], 采用 COW (Copy-On-Write) 机制, 即写时拷贝, 可在任意时间点以秒级速度创建快照, 不对块设备的使用造成影响, 使每个快照点的数据都能保存并且不会复制冗余数据, 其功能类似于 QCOW2 磁盘的快照功能[12].

2 系统设计

鉴于虚拟机备份对 OpenStack 环境的安全有着重要的作用, 而其快照备份方法不能满足备份的需要, 本文提出一种面向 OpenStack/Ceph 的虚拟机备份系统. 系统分为服务端和代理端, 包含 6 个模块: 磁盘处理模块、通信模块、备份模块、恢复模块、数据存储模块、虚拟机管理模块, 如图 1 所示, 备份模块和恢复模块通过调用虚拟机管理模块获取原虚拟机配置信息、创建新虚拟机等, 通过通信模块与磁盘处理模块通信, 传输磁盘有效数据或增量数据, 然后调用数据存储模块写入或读取备份数据. 其中代理端安装在环境里每个计算节点, 服务端安装在单独的备份服务器上.

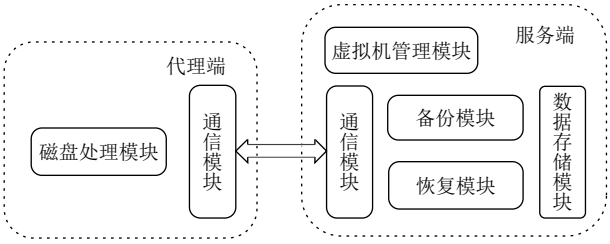

图 1 系统模块图

本文备份系统涉及到 OpenStack 环境的存储、管理、公开三种网络, 如图 2 所示. 管理网络用于服务端与 OpenStack 控制节点通信, 管理控制 OpenStack 环境, 管理网络还用于获取虚拟机信息、传输备份恢复数据; 存储网络用于代理端读取、写入、查询存储在 Ceph 环境中的磁盘数据; 公开网络使用户可以从外部访问备份服务器, 控制备份和恢复任务.

3 系统实现

3.1 备份模块

备份模块通过调用其它基础模块, 控制备份任务的流程. 当 OpenStack 环境加入到备份系统时, 会首先通过虚拟机管理模块获取所有虚拟机的状态列表, 用户可以选择某台或多台虚拟机创建备份任务, 备份任务可以分为全量备份、增量备份两种方式, 备份模块有不同的处理方式.

任务创建完成后, 后续流程均由备份模块控制, 用户不需要再操作, 备份的具体步骤如下.

1) 根据任务类型和状态, 如果是全量备份任务, 则直接执行全量备份; 如果是增量备份任务, 第一个备份点也是全量备份, 后续增量备份点均依赖于前一个备份点执行;

2) 通过虚拟机管理模块获取虚拟机的详细配置以及磁盘列表, 包括根磁盘、临时磁盘以及挂载的云硬盘, 获取所有磁盘的存储路径;

3) 对于磁盘列表中的每一个磁盘, 根据磁盘路径, 调用磁盘处理模块创建当前时间点 T_n 的快照, 记录快照的 ID 等信息;

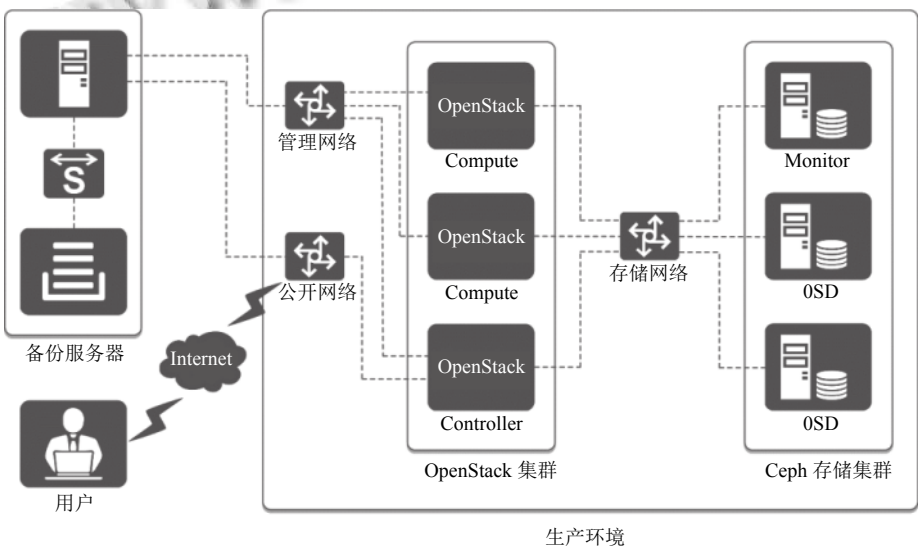

图 2 系统网络拓扑

4) 通过磁盘处理模块, 如果是全量备份点, 根据 T_n 时刻的快照计算每个磁盘的全量数据 Bitmap, 如果是增量备份点, 结合 T_n–1 和 T_n 时刻的快照计算每个磁盘的增量数据 Bitmap;

5) 根据备份任务类型, 选择不同的 Bitmap 用于数据备份, 调用通信模块进行传输, 调用数据存储模块保存磁盘数据;

6) 等待所有磁盘数据备份完成, 将虚拟机的配置信息保存至数据库或文件;

7) 如果是全量备份任务, 删除 T_n 时刻的快照; 如果是增量备份任务, 删除 T_n–1 时刻的快照, 保留 T_n 时刻的快照, 备份任务完成.

3.2 恢复模块

与备份模块相似, 恢复模块也是通过调用其它模块, 控制恢复的流程. 主要包含创建虚拟机、写磁盘数据两部分. 新虚拟机名、可用域和网络等设置为可选项. 通过创建相同配置的新虚拟机并将备份数据写入覆盖到新虚拟机的磁盘, 可以实现整机数据恢复. 数据恢复过程中虚拟机需要是关机状态, 整个恢复的步骤如下.

1) 用户选择需要恢复哪台虚拟机以及哪个时间点;

2) 通过虚拟机管理模块获取可恢复到的环境和租户, 验证用户在该租户内是否有权限;

3) 获取所选租户内可选网络和可用域, 允许用户配置新虚拟机的名称、网络和可用域等选项;

4) 根据保存的原机云主机类型和云硬盘挂载信息, 调用虚拟机管理模块创建与原虚拟机硬件配置相同的新虚拟机, 保证与原机磁盘数量、大小、挂载顺序一致, 然后获取新虚拟机的磁盘列表和路径信息;

5) 按照磁盘列表, 将新虚拟机与原虚拟机的磁盘一一对应, 针对每一对磁盘, 从数据存储模块, 根据所选时间点原磁盘的 Bitmap 信息依次读取有效数据块, 再调用磁盘处理模块写入覆盖新磁盘相同的偏移位置;

6) 等待所有磁盘数据恢复完成, 任务成功.

3.3 通信模块

通信模块基于 TCP/IP 协议, 提供备份服务器与计算节点的通信基础, 负责备份服务器与不同计算节点之间通信数据包的转发和分配, 该模块提供对三种数据的传输, 分别是虚拟化平台的命令和返回信息、磁盘操作的处理和回馈、备份恢复数据的请求和传输.

为满足不同的消息请求, 可以通过构造自定义的多级包头加以区分, 比如操作消息可以通过在包头定义操作码, 备份服务端和代理端再通过分析操作码对数据部分做不同的操作. 在传输时要保证数据传输的正确性, 可以构造待发送的数据包队列, 平衡处理各节点间的通信任务.

3.4 虚拟机管理模块

虚拟机管理模块用于处理与 OpenStack 云计算平台的连接、验证权限, 获取云计算平台的相关信息, 创建、删除虚拟机, 获取虚拟机配置, 磁盘列表和路径等. 该模块是系统与 OpenStack 环境通信的基础.

OpenStack 包含有 Keystone, Cinder, Nova, Glance 等各种组件, 每个组件提供不同的服务, 占用不同的端口[13]. 要与这些服务通信, 需要先获取权限, OpenStack 将用户划分为不同的权限级别, 且用户与租户相关, 需选择一个具有 Admin 权限的用户并通过 Keystone 服务认证.

完成认证后, 可通过 Nova 等服务获取所有虚拟机的详细信息, 包括虚拟机的硬件配置以及运行状况, 磁盘列表和路径, 用于备份和恢复任务. 创建新虚拟机时需保证与原机的配置相同, 结合用户可选的虚拟机名称、网络等信息, 再通过 Nova 和 Cinder 服务创建完成.

3.5 磁盘处理模块

磁盘处理模块位于每一个计算节点, 用于管理与 Ceph 环境的连接、生成快照、计算快照点 Bitmap、读取和写入磁盘数据等.

本系统逻辑上将磁盘划分为固定大小的数据块, 对每个磁盘生成一个 Bitmap, 每一个 bit 对应一个数据块. Ceph 提供了返回单个快照点内全部有效数据段或两个快照点间变化数据段的接口, 可计算出全量数据 Bitmap 或增量数据 Bitmap, 前者描述完整磁盘的有效数据, 后者描述磁盘两个快照点之间的变化数据.

本系统在计算 Bitmap 时将磁盘划分为多个较长的数据段, 一个数据段包含若干数据块, 通过 Ceph 获取每个数据段内的有效数据或变化数据信息, 再计算生成每个数据段的 Bitmap, 最终合并出磁盘的完整 Bitmap. 数据段的长度可采用 Ceph 的分段上限. 计算数据段的 Bitmap 方法如图 3 所示.

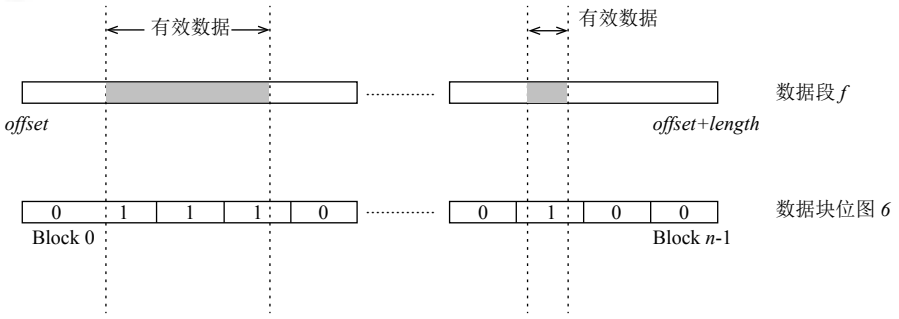

图 3 计算数据段 Bitmap

以某个数据段 f 为例, 其在文件中的偏移为 *offset*, 长度为 *length*, 对应 n 个数据块, 数据块编号从 0 到 n–1, 则分块长度为 *length*/n. 对于某一段有效数据, 假设其偏移和长度分别为 S 和 L, 对于第 i 个数据块, 如果存在:

$$\frac{length}{n}*i<S<\frac{length}{n}*(i+1),\ i\in(0,n-1)$$

或

$$S\le\frac{length}{n}*i<S+L,\ i\in(0,n-1)$$

则说明数据块内包含该段有效数据, 将 Bitmap 内第 i 个 bit 置为 1. 计算完成所有数据段之后, 即可合并出整个磁盘快照点的 Bitmap. 备份时, 只需对其中 bit 为 1 的数据块, 计算出其在磁盘文件中的偏移, 再从 Ceph 中按块的偏移和长度读取即可, 避免了对无效数据区的备份, 节约了备份时间和空间, 采用增量数据 Bimtap 传输数据可以实现增量备份的效果.

3.6 数据存储模块

数据存储模块负责保存虚拟机磁盘快照点的备份数据, 保存数据的方式采用固定的结构和方法. 存储结构上, 需按照存储根目录、虚拟机、快照时间点和磁盘的层次, 快照时间点和磁盘通过生成 ID 的方式加以区分. 每一个快照点内保存有所有磁盘的存储数据文件、索引文件、元数据等信息. 目录结构如图 4 所示.

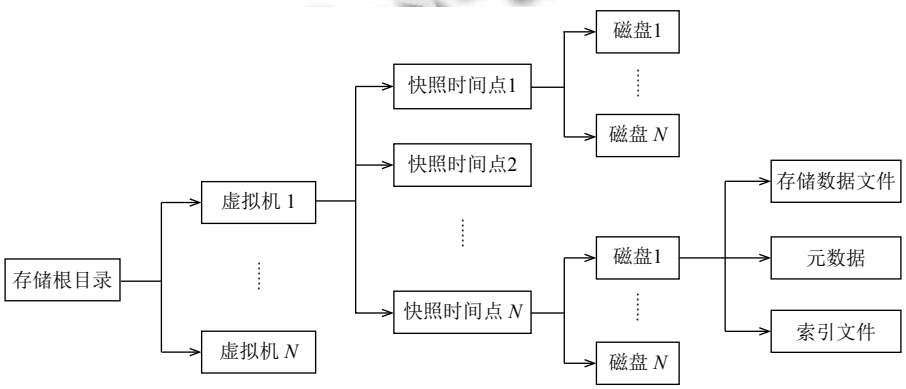

图 4 存储目录结构

存储数据文件保存全量备份点的有效数据块, 或增量快照点的变化数据块, 所有数据块拥有同样的块大小, 按顺序写入存储数据文件中. 元数据保存了磁盘 ID、依赖快照点 ID 等信息. 索引文件是根据磁盘快照点计算的 Bitmap 生成的, 每一行的格式为"bit|offset", bit 是指该数据块是否为磁盘当前快照点的有效数据块或增量数据块, 与 Bitmap 里的值相同, *offset* 指数据块在存储数据文件中的偏移, 如果当前没有保存该块, 则偏移为 0.

当进行备份时, 需要写入数据. 首先按照目录结构创建正确的路径; 针对每一个磁盘, 通过磁盘处理模块计算该快照点的全量或增量 Bitmap; 循环处理每一个 bit 位, 如果 bit 为 1, 则传输对应的数据块, 按顺序写入存储数据文件, 如果 bit 为 0, 则不需要传输, 同时应记录该 bit 的值和数据块在存储数据文件中的偏移, 并保存至索引文件, 不需传输的块偏移置为 0; 最后, 记录磁盘的依赖快照点、磁盘大小等元数据信息.

当进行恢复时, 针对每一个磁盘, 根据元数据信息获取完整的快照链关系, 然后合并所有索引文件的 Bitmap 信息, 合并方式为某数据块如果在任意时间点的 bit 为 1, 那么合并后该数据块 bit 也置为 1. 根据合并后的 Bitmap, 对 bit 为 1 的数据块, 从最近时间点开始读取, 如果该时间点未保存有该数据块, 即索引文件的偏移为 0, 则向前一个时间点继续读取, 以此类推. 最终可得到完整的磁盘数据.

以一个磁盘的增量快照链为例, 如图 5 所示, 增量点 T_2 依赖于增量点 T_1, 增量点 T_1 依赖于全量点 T_0, 每个快照点保存有索引文件、元数据、存储数据文件. 按照从左到右的顺序编号, 有数据块 0 到 15. 备份时, 根据 T_2 时刻计算出的增量数据 Bitmap, 数据块 5、10、11 的数据有变化, 所以当前只需要备份这三个数据块, 将数据写入到有效数据文件, 并记录偏移 *offset*.

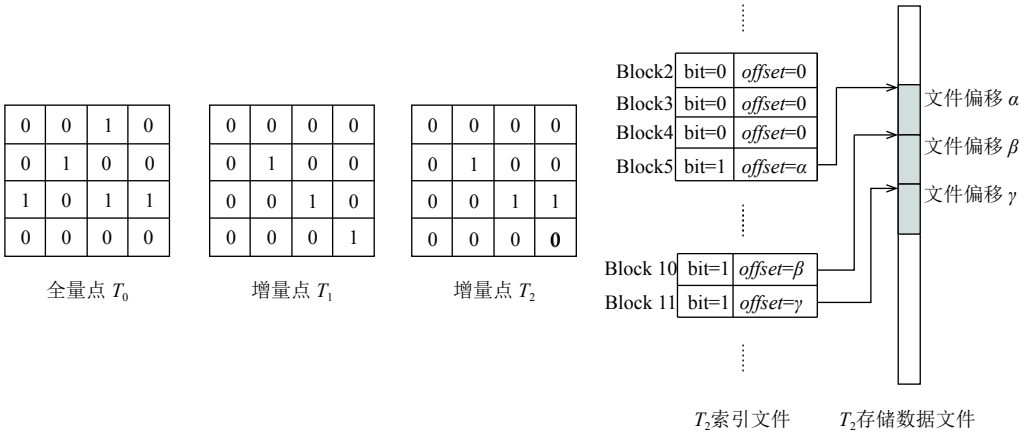

图 5 备份数据存储方式

恢复时, 首先根据快照链关系, 合并 T_0、T_1、T_2 的 Bitmap, 得到 T_2 时刻磁盘的完整数据 Bitmap, 即数据块 2、5、8、10、11、15 的 bit 都为 1, 再根据完整数据 Bitmap 从存储数据文件读取数据块, 优先从最近时间点开始读, 如果没有查询到该数据块, 则从依赖快照点中读, 以此类推. 图 5 例中, 首先会从增量点 T_2 读取数据块 5、10、11, 然后从增量点 T_1 读取数据块 15、从全量点 T_0 读取数据块 2、8, 最终即可得到 T_2 时刻磁盘的完整数据.

4 实验分析

4.1 数据存储模块

实验环境参照图 2 的网络拓扑, OpenStack 版本号为 Mitaka, 虚拟机磁盘格式为 RAW; 备份服务端安装在独立主机上, 代理端安装于 OpenStack 所有计算节点上. 实验采用 OpenStack 虚拟机快照备份功能作为对比方法, 实验对象选用云环境内不同操作系统、不同系统配置的虚拟机, 配置包含单磁盘、多磁盘等多种方式, 本文系统分块大小为 2 MB.

4.2 实验结果分析

为测试备份系统虚拟机的增量数据备份, 创建一台拥有 10 GB 根磁盘的虚拟机, 操作系统为最小化安装的 Centos7.3, 先用 dd 命令写入 5.1 GB 非 0 数据扩充根磁盘大小, 在 T_0 时刻做一次全量备份, 由于都是提取的有效数据, 两种方法全量备份的效果相同, 然后自 T_1 时刻起向根磁盘连续写入测试数据并做增量备份, 如表 1 所示. 测试数据采用不同版本的 Linux 内核 tar.gz 压缩文件, 版本号依次为 Linux-2.6.39.2、Linux-3.6.9、Linux-3.8.2、Linux-3.12.69、Linux-4.6.3、Linux-4.9.88.

实验结果表明, 本文提出的备份系统可针对变化数据使用增量备份, 而 OpenStack 快照备份只能对根磁盘做全量备份, 故备份大小约等于根磁盘已分配大小. 根磁盘增量数据备份传输的大小和时间对比如图 6 所示, 无论是节省备份时间还是存储空间, 本文系统均优于 OpenStack 的快照备份方式, 实验中本文系统平均节约了 88.72% 的备份时间和 97.14% 的备份存储空间, 在变化数据相对于磁盘已分配空间较小时, 本文系统的优势更加明显.

表 1 根磁盘增量数据备份

时刻	写入测试数据大小 (MB)	写入后的磁盘空间已分配大小 (MB)	OpenStack 快照备份		本文系统		时间节省比例 (%)	存储节省比例 (%)
			传输大小 (MB)	时间花费 (s)	传输大小 (MB)	时间花费 (s)		
T_1	91.56	6249.59	6256	73	168	9	87.67	97.31
T_2	99.12	6348.71	6368	74	174	7	90.54	97.28
T_3	102.07	6450.78	6480	73	176	10	86.30	97.28
T_4	110.21	6538.25	6576	73	186	6	91.78	97.17
T_5	129.86	6668.11	6704	76	208	10	86.84	96.90
T_6	135.27	6803.38	6848	74	212	8	89.19	96.90

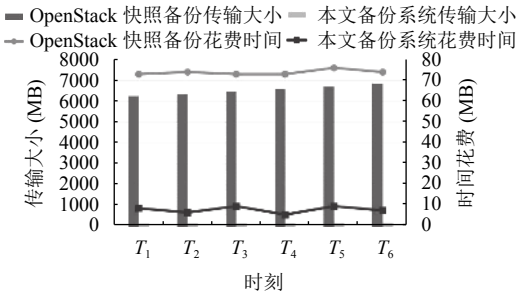

图 6 根磁盘增量备份对比

为测试多磁盘的备份情况, 创建一台拥有 10 GB 根磁盘、10 GB 临时磁盘、10 GB 云硬盘的虚拟机, 对比 OpenStack 快照备份方法, 实验全量备份、增量备份任务, 实验结果表明, 本文提出的备份系统可针对多磁盘虚拟机实现有效保护, 对临时磁盘、云硬盘均可实现全量和增量备份, 备份效果与根磁盘相同. 与 OpenStack 快照备份的功能对比如表 2 所示, 在恢复测试中, 每个时间点所有磁盘的数据均能正确恢复到新虚拟机.

表 2 多磁盘备份功能对比

备份方法	根磁盘	临时磁盘	云硬盘
OpenStack 快照备份	全量	不支持	不支持
本文备份系统	全量/增量	全量/增量	全量/增量

5 结语

本文针对以 Ceph RBD 作为后端块存储方法的 OpenStack 环境, 设计并实现了一种利用 Ceph 快照特点的虚拟机备份系统, 实验结果表明, 相比于 OpenStack 的虚拟机快照功能, 本系统可以实现后者不具有的虚拟机整机的数据保护功能, 包括对虚拟机配置信息的保存、多磁盘数据的备份, 可以实现跨用户、跨租户的恢复, 通过计算变化数据可以实现增量备份, 有效节约了备份时间和存储空间. 但本系统的增量备份会保留一个快照, 备份速度和增量计算精度与分块大小相关, 备份数据也可以通过存储网络传输, 后续可以做进一步优化和改进.

参考文献

1 Kumar R, Gupta N, Charu S, *et al*. Open source solution for cloud computing platform using OpenStack. International Journal of Computer Science and Mobile Computing, 2014, 3(5): 89–98.

2 Habib I. Virtualization with KVM. Linux Journal, 2008, 2008(166): 8.

3 Weil S A, Brandt S A, Miller E L, *et al*. Ceph: A scalable, high-performance distributed file system. Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation. Seattle, WA, USA. 2006. 307–320.

4 Carlson M, Yoder A, Schoeb L, *et al*. Software defined storage. Santa Clara, CA, USA: Storage Networking Industry Assoc, 2014. 20–24.

5 OpenStack Foundation. A snapshot of OpenStack users' perspectives and deployments. Openstack User Survey (Apr. 2017), 2017.

6 Garfinkel T, Rosenblum M. When virtual is harder than real: Security challenges in virtual machine based computing environments. Proceedings of the 10th Conference on Hot Topics in Operating Systems. Santa Fe, NM, USA. 2005.

7 McDowall RD. Computer (In)security–2: Computer system backup and recovery. The Quality Assurance Journal, 2001, 5(3): 149–155. [doi: 10.1002/(ISSN)1099-1786]

8 Varia J, Mathew S. Overview of Amazon Web services. Amazon Web Services, 2014.

9 Weil SA, Brandt SA, Miller EL, *et al*. CRUSH: Controlled, scalable, decentralized placement of replicated data. Proceedings of 2006 ACM/IEEE Conference on Supercomputing. Tampa, FL, USA. 2006. 122.

10 许艳军, 姜进磊, 王博, 等. 几种虚拟机镜像格式及其性能测评. 计算机应用, 2013, 33(S1): 22–25.

11 Singh K. Learning Ceph. Birmingham: Packt Publishing, 2015. 112–116.

12 McLoughlin M. The QCOW2 image format. http://people.gnome.org/~markmc/qcow-image-format.html. (2008-09-11), [2018-03-07].

13 OpenStack API. Complete reference. http://developer.openstack.org/api-guide/quick-start/. [2018-07-17].