数据中心下基于相似文件协助的 P2P 镜像分发技术研究

韦立超[1+] 符永铨[1] 刘 锋[1] 彭宇行[1]

[1]国防科学技术大学并行与分布处理重点实验室, 长沙 410073

+Corresponding author: Phn +86-15616021926, E-mail: chaoge_nudt@126.com

Research on the Similar-File Assisted P2P based Image Delivery Technique in the Data Center Environment

Wei Lichao[1+], Fu Yongquan[1], Liu Feng[1], and Peng Yuxing[1]

[1](National Laboratory of Parallel and Distributed Processing, National University of Defense Technology, Changsha, 410073)

Abstract To enable the elastic Infrastructure as a Service, the virtual-machine deployment platform in the data center has to start the virtual machine fast, otherwise, the quality of experiences of users will degrade significantly. The open-source virtual machine server platforms typically push images from the storage server to client server directly, which suffers from the performance bottlenecks with increasing concurrent connections. The Peer-to-Peer (P2P) approach that collaboratively transmits images among clients avoids the performance bottlenecks. Unfortunately, existing P2P methods disseminate the same image among all clients, which can't fully exploit the redundant chunks that are prevalent in the data center. The paper addresses the limitation by proposing a new extended BitTorrent based similar image dissemination-assisted protocol that is able to disseminate redundant chunks among clients. Tracker searches top-k most similar images of target image and deploy seed services proactively on the servers that host these images. We improve the piece-download algorithm to exchange chunks not only from peers downloading the same image, but also from those downloading similar images. We design and implement three algorithms, namely the similar-image searching algorithm, the seed selection algorithm, and the same-piece downloading algorithm. A test based experiment shows that our method increases the downloading rate by 10% to 70% compared to the standard BT protocol, moreover, the overhead is modest on the Tracker server.

Key words: VM deployment; similar image; P2P

摘要 为有效提供弹性的基础设施即服务，数据中心虚拟机运维平台需要快速启动虚拟机，否则用户体验将严重下降。开源的虚拟机平台通常采取服务器直接推送的方式，但随并发连接的增长容易产生性能瓶颈。P2P 方法使得客户端之间协同传输镜像，避免了性能瓶颈。不幸的是，在现有的 P2P 方式中只有下载同一镜像的客户间才能交换数据，没有充分挖掘数据中心中流行的冗余块资源。为克服上述限制，本文提出一个基于相似镜像协助分发的扩展 BitTorrent 协议，通过 Tracker 搜索与目标镜像最相似的 top-k 个镜像文件，并在这些相似镜像文件所在服务器主动部署种子节点。本文改进了文件块下载算法，使得下载节点不仅与下载相同镜像的其它节点交换数据，还能从相似镜像种子节点下载数据。我们设计并实现了三种算法，分别是相似镜像检索算法、相同块编号映射算法、相似文件下载算法。实验结果证明，改进的 BitTorrent 协议较标准 BitTorrent 协议虚拟机镜像分发速度提升 10%~70%。并且，Tracker 服务器的开销适中，性能也得到保证。

关键词 虚拟机部署；相似镜像；P2P

中图法分类号 TP399

收稿日期：2015-07-01

基金项目：国家自然科学基金重点项目(61402509)

云计算基础设施服务(IaaS)利用虚拟机运维平台为用户提供弹性可扩展的虚拟机资源。为了提供弹性的虚拟机资源，虚拟机运维平台需要快速启动虚拟机，否则，用户体验将显著下降。在主流的 Openstack、CloudStack 开源虚拟机运维平台中，宿主机(安装虚拟机镜像的主机)需要先从中心服务器下载虚拟机镜像到本地，然后启动虚拟机实例。从用户发出虚拟机创建指令到所有虚拟机都正常启动的间隔(称为响应时间)是重要的性能度量标准。虚拟机镜像的传输时间占据了响应时间很大的部分[1]，因此优化镜像文件传输时间成为快速部署虚拟机的重要问题。随着用户数量及业务的迅速增长，直接从中心服务器下载镜像文件的方式会增加存储服务器的负载，造成性能瓶颈和单点故障问题，导致虚拟机镜像传输时间增大，严重降低了服务体验和服务质量。

已有研究采取 P2P 加速镜像文件的传输(如 Snowball Tree[2]，P2P Network[3]，VMTorrent[5])，宿主机之间通过共享目标镜像最终完成镜像文件的下载。这种方式无需依赖集中式服务器支持，很好地克服了直接拉取方式的性能瓶颈和单点故障问题，并具有优异的可扩展性。

在 P2P 镜像分发中，下载同一文件的节点属于同一群体，只有同一群体内的节点才能相互交换数据，不同群体的节点间不交换数据。这种文件级的传输忽略了镜像之间的相似性[4]。文件被划分为大小相同的块时，不同镜像之间存在许多相同的块，已有的 P2P 文件共享协议在下载过程中无法挖掘相似镜像间相同的块资源，影响了下载速度，增加了镜像文件的分发时间。

针对已有镜像文件分发方法存在的上述不足，本文以 BitTorrent(简称 BT)文件共享协议为基础，提出基于相似镜像文件协同传输的改进 BT 协议。基本思想是 BT 中的 Tracker 节点主动发现数据中心的相似镜像资源，在资源所在节点主动部署相似镜像的种子节点，为镜像下载节点上传相同的块资源，提高下载节点获取目标镜像的传输速度。本文以开源的 BT 协议为基础设计并实现了相似镜像协作分发功能。系统部署测试表明，本文提出的方法在保证 Tracker 服务器性能的情况下能显著提高虚拟机镜像的分发速度，缩短了虚拟机镜像的传输时间 10%~70%。

1　问题模型与相关工作

本节从概念定义、BT 镜像分发和网络带宽限制三方面介绍了基于 BT 的镜像数据分发模型。最后介绍了镜像分发研究的一些相关工作。

1.1　概念定义

为便于描述，遵循已有 P2P 系统的术语，本文对重要的概念做解释如下：

块：BT 系统把目标文件划分为大小相同的块(piece)，块在整个目标文件中按顺序排列，每个块都有编号，节点之间使用块编号来定位数据。

目标镜像：peer 正在下载的镜像文件。

Tracker：负责收集下载者信息的服务器，并将此信息提供给其他下载者，使下载者们相互连接起来交换数据。

peer：指拥有部分文件块的正在下载的节点。

seed：指拥有一个完整文件的节点。

torrent 文件：内容提供者为共享文件创建的一个元数据文件，描述共享文件的文件名、大小、文件块总数、每个文件块的校验值和 Tracker 地址。

相似文件：当有一个文件的块集合为 A，目标文件的块集合为 B，且 A∩B≠Ø，则称该文件为目标文件的相似文件。

群体：指 BT 中下载同一共享文件的节点组织成的一个 P2P 覆盖网。

相似度：目标文件和相似文件相同块的个数为 M，目标文件的块的个数为 N，a = M/N*100%，则称 a 为相似文件与目标文件的相似度。

1.2　BT 镜像分发原理与不足

基于 BitTorrent 的镜像数据分发过程如图 1，除 Tracker、peer 和 seed 节点外，图中 Web 服务器负责保存 torrent 文件发布到 Web 服务器。peer 从 Web 服务器下载 torrent 文件后，解析文件信息及 Tracker 地址，然后连接 Tracker，得到其他 peer 的信息，接着与其它 peer 建立连接，获取文件的各个片断，同时向其它 peer 上传文件片断。

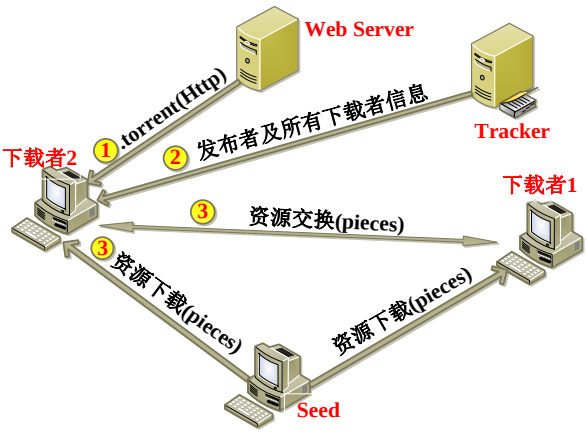

图 1　基于 BT 的镜像数据分发

peer 进程解析 torrent 文件，计算出种子特征码 infohash，每个 torrent 文件的 infohash 都是不同的，它是目标文件唯一的身份标志，也称为目标文件的文件标识符，Tracker 及 peer 都是以 infohash 来识别不同的目标文件。

peer 进程解析 torrent 文件，计算出种子特征码 infohash，每个 torrent 文件的 infohash 都是不同的，它是目标文件唯一的身份标志，也称为目标文件的文件标识符，Tracker 及 peer 都是以 infohash 来识别不同的目标文件。

peer 周期地连接 Tracker，将自己的目标镜像的 infohash 和下载进度告知 Tracker，Tracker 统计客户端提交的信息，并返回其他 peer 信息。最后，BT 客户端与其他 peer 建立邻居关系，相互交换镜像的数据块，直至下载完成。

基于 BT 的镜像分发系统中各个节点功能和地位相等，避免了因过度占用中心服务器资源而造成网络瓶颈或瘫痪的问题。但其在数据中心环境中依然存在以下不足：

(1) BT 中一个群体所提供的服务能力和节点规模成正相关关系，在中小型下载任务较多的情况下[6]，较低的网络服务能力使 peer 的下载带宽得不到充分利用，导致镜像分发速度缓慢。而中小型的虚拟机部署任务在数据中心非常普遍。

(2)下载任务中的 seed 较少时，在下载起始阶段，peer 间没有可供交换的数据块，只能从 seed 下载，这样就造成网络中资源稀少，peer 初始下载速度缓慢。

(3)没有充分利用相似镜像的块资源。[7]指出大多数虚拟机镜像首先使用裸操作系统构造基本的镜像，在该基本镜像上安装应用程序或打补丁后就创建了一个新的镜像文件。大部分自定义镜像都是基于类似的支持系统版本衍生而来。从基本镜像衍生的不同镜像文件共享大量重复的公用数据，即镜像文件具有相似性。比如同一 Linux 发行版的不同版本的虚拟机镜像的鉴别数据块一般高度重叠[8]。同一群体中的节点只能共享同一镜像，无法共享相似镜像中的相同块资源来加速下载。

1.3 网络带宽限制

本文假设多租户的数据中心环境，为了避免 BT 占用过多的带宽资源，每个服务器节点对 BT 传输的上传带宽和下载带宽进行限制。由于数据中心环境下服务器硬盘较大，存储数据较为频繁，而向外发送行为相对不多，故本文默认限定 BT 的上传带宽低于下载带宽。

1.4 相关工作

文献[3]、[5]改变 BT 的块选择策略，优先下载启动虚拟机需要的块数据，虽然减少了文件的传输量，但需要复杂的配置文件，消耗了大量的时间，并且也忽略了相似镜像中的相同块资源。文献[9]将压缩后的镜像文件传输到宿主机后再解压缩。该方式减少了网络传输量，但增加了压缩和解压缩的开销，可扩展性也不好。文献[10]提出模板预拷贝策略减少镜像模板的传输时间，对部署频率较高及用户感兴趣的镜像模板预先存储在一个预拷贝模板库中以供下次使用。该方法增加了存储开销，且虚拟机镜像会动态更新，导致部署频率高的镜像较少，部署性能不高。

文献[6]利用虚拟机镜像的相似性设计了一种虚拟机镜像分发网络 VDN，VDN 使用分布式文件系统作为镜像存储结构，节点缓存下载到的文件块，下次使用镜像时可以不必完全下载镜像。这种存储系统中的分块设置相对缓解了服务器压力，但启动虚拟机数量巨大时，存储系统依然会成为数据传输的瓶颈，且 VDN 实现复杂。

文献[11]、[12]都采用 P2P 辅助的内容分发系统，通过集中调度动态分配种子节点给用户集群的上传带宽，从而获得最大的聚集下载带宽。前者设计了一种基于内容分发的 Antfarm 系统，通过测试种子上传带宽与用户集群的聚集下载带宽的响应曲线，并使用优化算法计算出种子带宽分配的最优解。但用户集群是动态的，获取响应曲线的代价很大，不实用，且 Antfarm 不支持分布式数据中心环境，无法保证性能。不同的是，后者提出了一种新型的内容分发指标 CPM，系统通过 CPM 计算节点上传的数据块在各个用户群体的传播速率来决定节点应给哪个用户群体分配上传带宽，最大化系统的下载带宽。该方法实现复杂，开销较大，其集中式调度策略与 P2P 的分布式工作原理并不兼容。

数据中心网络环境安全、稳定和均匀，文献[13]改进了 BT 协议，舍弃 Tit-For-Tat 阻塞算法，每个节点尽全力贡献自己的上传带宽。另外，节点收到数据块后并不进行块的完整性校验，仅在下载完成后对整个文件进行完整性校验。虽然数据中心下发生错误的概率较低，如果发生错误则整个文件都要重新下载，很浪费时间。

2 基于相似文件协助的 BT 协议

2.1 基本思想

针对 BitTorrent 无法利用相似镜像资源的不足，本文提出了基于相似文件协助的 BT 协议(后文统称为改进 BT 协议)，主动发现相似文件节点，将这些节点作为提供相似镜像的种子节点，为 peer 节点上传有用的块数据，从而提高系统的文件资源和服务能力，达到加速虚拟机镜像分发速度的目的。

改进 BT 协议保留了 BT 原有的基于文件同一群体内部节点间交换数据的功能，当有下载任务时，系统就主动发现相似镜像的种子节点。

跨镜像数据块分发协议的原理如图 2 所示。

镜像 1 和镜像 2 是两个相似的镜像，Peer1、Peer2 所属的群体下载镜像 1，Tracker 收到镜像 1 的下载任务时，查找发现 Peer3 节点拥有的镜像 2 与镜像 1 是相似的。此时 Tracker 部署 Peer3 为镜像 2 的种子节点。镜像 1 和镜像 2 在逻辑上被划分为大小相等的块，镜像 1 和镜像 2 都有片断 a、片断 b 和片断 c，而且片断 a 是镜像 1 的第一个 piece，位置编号即为 0，在镜像 2 的位置编号为 1。同理也能得到块 b、块 c 分别在镜像 1 和镜像 2 的位置编号。于是，镜像 1 和镜像 2 共享数据块的位置编号映射就为[(0，1)，(1，3)，(2，0)]。如右图所示，当 Peer1、Peer2 与 Peer3 建立连接后，节点 Peer1 向 Peer3 请求片断 a 和 Peer2 向 Peer3 请求片断 c 时，Peer3 都能够满足其请求。

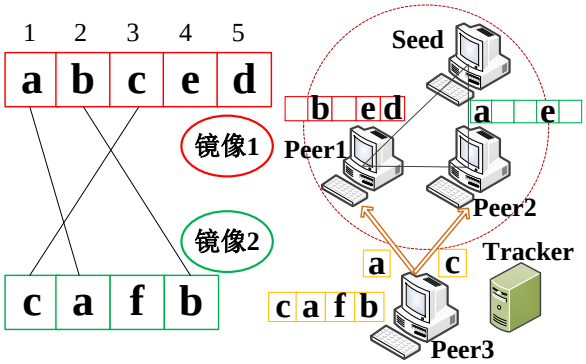

图 2　基于相似文件协助的 BT 协议原理图

2.2　工作流程

改进 BT 协议的整体流程如图 3：Web 服务器通过共享存储获取镜像 torrent 文件，预先计算镜像之间的相似度，进行 top-k 排序。一个 peer 联系 Tracker 后，Tracker 主动部署目标镜像的 top-k 相似镜像种子节点，同时向 peer 返送相似镜像的种子节点。peer 访问共享存储计算目标镜像与 top-k 相似镜像的相同块的位置编号关系，连接相似镜像的种子节点后选择感兴趣的块进行分布式下载。而主动部署的镜像种子则上传相应的请求块。

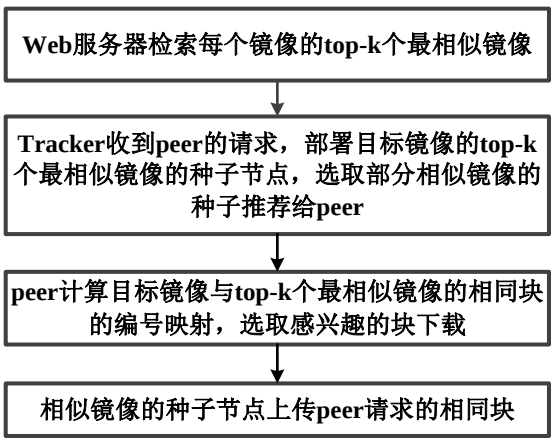

图 3　改进 BT 协议工作流程

为了支持基于相似镜像的主动传输，本文提出的改进 BT 协议在 BT 基础上添加了三个模块：

（1）相似镜像检索模块：Web 服务器解析存储的 torrent 文件，搜索每个镜像文件的相似镜像，计算每个镜像与其相似镜像的相似度和相同块的编号映射。根据相似度对相似镜像排序。

（2）相似镜像的种子推荐模块：Tracker 主动部署目标镜像最相似的 top-k(可调参数，默认为 5)个镜像的种子节点，并分别从种子列表、相似镜像的种子列表和 peer 列表随机选择部分节点返回给客户。客户与 Tracker 返回的节点列表建立连接并交换数据。

（3）相同块下载模块：peer 连接 Tracker，从 Tracker 返回的映射文件中读取相似度最高的 top-k 个镜像文件与目标镜像的相同块的映射。peer 收到 Tracker 提供的相似镜像的种子列表后，发起连接，根据目标镜像与相似镜像的相同块的映射向相似镜像的种子请求数据块。

改进 BT 协议在兼容原协议和保证可扩展性的同时，具有以下优势：

- 根据下载任务进行灵活的相似镜像种子节点发现与部署。
- 下载节点在继承 BT 协议传输的同时实现了共享相似镜像文件的下载功能。

2.3　基于相似镜像种子协助下载的实现

本节介绍基于相似镜像种子协助下载的改进 BT 协议的实现，从相似镜像检索、相似镜像的种子推荐和相同块下载三个方面分别介绍了各自的实现流程和算法设计。最后对 Tracker、seed 和 peer 端的负载进行了分析。

2.3.1 相似镜像检索

Peer 若要从相似镜像的种子节点下载数据块，首先要检索目标镜像的相似镜像，并从中选择与目标镜像最相似的 top-k 个镜像。由于 Web 服务器只负责提供 torrent 文件，其负载相对较小，故由 Web 服务器计算并更新每个镜像与其相似镜像的信息，该信息和 torrent 文件存储到一个共享存储，如 samba 服务器。Tracker 和 peer 通过网络存储挂载的方式实现共享。

为检索每个镜像的 top-k 个最相似的镜像，Web 服务器初始化时建立一个保存每个镜像与其相似镜像相关信息的文件 SF。Web 服务器初始化时先判断 SF 是否为空，若为空就搜索每个镜像的相似镜像。

由于 torrent 目录是动态的，为方便更新和维护每个镜像的相似镜像的信息，本文采用字典 *Dict* 存储每个镜像与其相似镜像的相关信息。字典的键为镜像的文件标识符 infohash，值是一个嵌套字典，嵌套字典的键为该镜像的相似镜像的 infohash，值是一个列表，列表的元素由相似度

和 torrent 文件名构成。另外嵌套字典还有一个键为“top-k”，值为一个由 top-k 个最相似的镜像的 infohash 组成的列表。本文设计了相似镜像检索算法构建 SF 文件，算法如下：

算法 1 相似镜像检索算法

输入：torrent 文件目录 *TD*
输出：相似镜像的信息文件 *SF*
描述：

```
1.  for each torrent file F ∈ TD
2.      count = 0
3.      计算 F 的 infohash 值为 Fi
4.      for each torrent file F′∈ TD and F′ ≠ F
5.          计算 F′的 infohash 值为 F′i
6.          if Dict[Fi] not has key F′i
7.              for each piece p ∈ F
8.                  if p ∈ F′
9.                      count ← count + 1
10.                 end if
11.             end for
12.         Dict[Fi][F′i] ← count
13.         end if
14.         if Dict[F′i] not has key Fi
15.         Dict[F′i][Fi] ← count
16.         end if
17.     end for
18.     选出与 F 最相似的 top-k 个镜像，并组成列表 list
19.     Dict[Fi][‘top-k’] ← list
20. end for
21. save Dict to SF
```

假设总共有 N 个 torrent 文件，每个 torrent 文件平均有 m 个 piece，每个 torrent 文件的相似镜像平均有 q 个。计算每个 torrent 文件与其相似镜像的相似度需遍历其余 N-1 个 torrent 文件的每个数据块，从而得到 Dict 的时间复杂度为 $O(N^2*m)$。计算 N 个镜像的 top-k 个最相似镜像需对每个镜像的 q 个相似镜像全排序，故查询 N 个镜像的 top-k 个相似镜像的时间复杂度为 O(N*qlogq)，故检索每个相似镜像的复杂度为 $O(N^2*m)+O(N*qlogq)$。算法 1 的空间复杂度为 Dict 的大小，为 O(N*q)字节。

Web 服务器会周期地访问 torrent 目录。当 torrent 目录更新时，如果有新的 torrent 文件产生时，Web 服务器遍历其他 torrent 文件，搜索新 torrent 文件的相似文件，并将它们的相关信息更新到 SF 文件。如果删除了无效的 torrent 文件，Web 服务器同时将其与相似文件的相关信息从 SF 文件中删除。

2.3.2 top-k 相似镜像种子推荐

为了利用相似镜像加速下载，系统需要选择最有效的种子。本文利用 Tracker 部署与目标镜像最相似的 top-k 个镜像的种子节点，并将这些相似镜像的种子节点地址告知 peer，从而最大化下载效率。具体实现如下：

为主动部署相似镜像的种子节点，Tracker 需先发现相似镜像所在节点的 IP 地址，从而可以向该节点发送对相似镜像做种的命令。

Tracker 服务器维护一个以镜像 infohash 为键，以存储节点的 IP 为值的字典来记录每个镜像的存储位置。该位置存储字典初始化时为空，随着节点下载镜像不断更新，一个镜像可能对应多个存储节点，可将节点 IP 地址追加到相应 infohash 的键值，即字典的一个键对应多个值。

peer 向 Tracker 发送下载请求时，Tracker 从共享存储中的 SF 文件读取目标镜像的 top-k 个最相似镜像。然后根据镜像的位置存储字典得到拥有此 top-k 个相似镜像的所有节点的 IP。随后 Tracker 向上述节点发送对相似镜像做种的命令使之成为种子节点，开始共享相似镜像的数据。

相似镜像的种子节点在 Tracker 注册后，Tracker 根据返回给客户的种子数、相似镜像的种子数和其他 peer 的数目占所有返回给客户的节点数的比例等于所有节点中种子数、相似镜像的种子数和下载节点占的比例的原则以及用户要求返回的节点数，从目标镜像的种子列表、相似镜像的种子列表和 peer 列表中分别随机选取相应数量的节点返回给客户。

2.3.3 相同块下载

peer 获取相似镜像的种子节点之前，peer 先访问共享存储中的 SF 文件，得到目标镜像的最相似的 top-k 个相似镜像的 infohash，然后计算目标镜像与各个相似镜像的相同块的编号映射。并将相似镜像的 infohash 和相同块的编号映射以键值对存入字典 dict 中。算法如下：

算法 2 相同块编号映射算法

输入：top-k 个 infohash 列表 *list*；目标镜像的 infohash 值 *T*
输出：存储相同块编号的字典 *dict*
描述：

```
1.  for each infohash U ∈ list
2.      for each piece p ∈ U
3.          if p ∈ T
4.              得到 p 在 T 的编号 index1
5.              得到 p 在 U 的编号 index2
6.              元组(index1, index2) 添加到列表 L
7.          end if
```

```
8.      dict[U] ← L
9.      end for
10.  end for
11.  return dict
```

在算法 2 中，list 的元素个数为 top-k，假设每个相似镜像平均有 m 个 piece，目标镜像与相似镜像平均有 q 个相同块。那么该算法的时间复杂度为 O(m*q*top-k)。空间复杂度为 dict 的大小，为 O(q*top-k)个字节。

在 BT 网络中，节点间以块为单位交换数据，节点间通信的消息中封装块的编号信息。由于相同块在目标镜像和相似镜像的编号可能不同，故 peer 向相似镜像的种子发送块请求时，需转换为其在对方镜像文件的编号。本文根据最少优先下载原则选择要下载的数据块，即 peer 选择本身所缺少的且邻居节点上数量最少的数据块优先下载，用 NumHave(i)表示拥有文件块 i 的节点数。为实现跨镜像的数据传输，本文设计并实现了相同块的下载算法，见算法 3。

算法 3 相同块下载算法

输入：存储相同块编号映射的字典 *dict*；下载节点 *P*；相似镜像的种子节点 *S*；相似镜像的 infohash 值 *U*

输出：相同块的内容

描述：

```
1.  init Min=2**30
2.  while P 未下载完 do
3.      for each (index2, index2) ∈ dict[U]
4.          if P 无 index2 and S 有 index2
5.              P 对 S 感兴趣
6.              break
7.          end if
8.      end for
9.      if P 对 S 感兴趣
10.         for each (index1, index2) ∈ dict[U]
11.             if S 有 index2 and NumHave(index1) <
                Min
12.                 Min ← NumHave(index1)
13.             end if
14.         end for
15.         P send message(index1, index2) to S
16.     end if
17.     S 发送 index2 块的内容和 index1 到 P
18.     P 对收的块进行 hash 校验后保存到编号
        index1 位置处
19. end while
```

在算法 3 中，相似镜像的种子 S 中若有 peer P 还没下载的相同块，则 P 对 S 感兴趣。然后 P 根据最少优先原则选择一个块，并将该块的原编号和在相似镜像的编号封装在请求消息中发送给相似镜像的种子。相似镜像的种子取出正确的块后，将块的原编号和块内容上传给 P。

假设算法 3 中下载节点 P 和相似镜像的种子节点 S 的相同块个数为 m，其时间复杂度为 O(m)。

2.3.4 负载分析

Tracker 作为中心服务器，保证其性能对系统的可扩展性很重要。为尽量不增加 Tracker 负载，改进的 BT 协议由 Web 服务器来执行检索相似镜像和计算相同块的编号映射的任务。Tracker 需完成部署相似镜像的种子节点的功能，这会增加部分带宽开销。另外，Tracker 还需维护一个镜像位置存储字典以记录镜像的存储节点。假设系统中有 N 个镜像，平均每个镜像所在节点数为 H，节点 IP 所需空间为常数 K 字节，则构建镜像位置存储字典需要 K*N*H 字节的内存空间。随着系统规模增大，Tracker 的负载也是线性增加的。

seed 端只需要接受 peer 的块下载请求并将块内容发送给 peer，故改进的 BT 协议中，seed 的负载基本没有发生变化。而 peer 端在下载之初需要计算目标镜像与 top-k 个最相似镜像的相同块的编号映射，由算法 2 知其增加的开销对 peer 的下载并没有显著的影响。

3 实验与测试

本节测试使用原 BT 协议和改进 BT 协议分发镜像的平均下载速度以及 Tracker 的负载情况。

3.1 实验环境

本文采取一个服务器机柜开展实验。机柜共有 19 台主机，其中一台作为 Tracker 服务器，10 台用来做下载节点，一台作为种子节点，剩余的 7 台作为相似镜像的种子节点。每台主机的配置为 Intel Xeon E5-1620 3.6GHz CPU、48GB RAM、1TB 硬盘和 1GBps 以太网卡，CentOS 6.4 操作系统和 V2.6 Python 编译器。

3.2 实验与性能分析

本实验通过调整上传/下载带宽比、目标镜像和相似镜像的相似度、相似镜像的种子数、peer 的规模和 Tracker 的性能等参数，检验改进 BT 协议和原 BT 协议的下载性能。

3.2.1 改变上传/下载带宽比的加速实验

为模拟多租户环境下带宽限速环境，通过实验检验节点上传带宽对 BT 下载性能的影响。

实验选定 1 个种子节点，10 个 peer 同时下载目标文件，设定节点下载带宽为 100MB/s，并改变节点上传/下载带宽比。对每种上传/下载带宽比进行 5 次测试，改变相似镜像的种子节点数，

相似镜像和目标镜像的相似度为 50%。测量下载节点的平均下载速度。实验结果如图 5 所示：

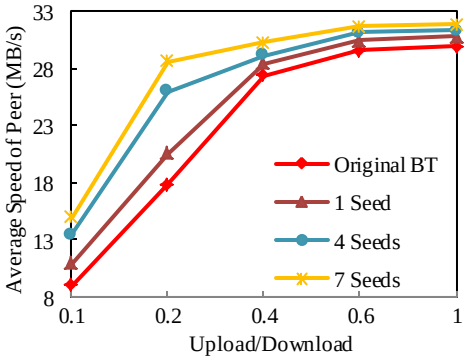

图 5　不同上传/下载带宽比 peer 平均下载速度

实验表明，固定下载带宽，当上传/下载带宽比较小时，peer 的平均下载速度随相似镜像的种子增多而有很明显的提升；当上传/下载带宽较大时，相似镜像的种子增多时，peer 的平均下载速度提升幅度很小。这主要因为随着上传带宽增大，在带宽充足的情况下 BT 的带宽利用率不高。

3.2.2　改变相似镜像的种子数的加速实验

设定下载任务中有一个种子节点，10 个 peer 同时下载目标镜像，每个节点的上传/下载带宽为 10/100MB/s，相似镜像与目标镜像的相似度为 50%。先测试标准 BitTorrent 系统中 peer 的平均下载速度。然后加入不同数量相似镜像的种子节点，测试改进的 BitTorrent 系统中 peer 的平均下载速度。实验结果如图 6 所示：

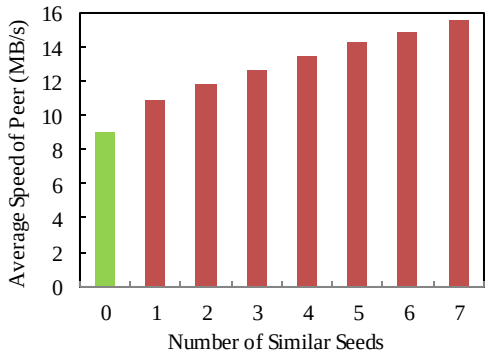

图 6　不同数量相似镜像种子 peer 平均下载速度

相似镜像的种子节点数为零时，就是标准 BT 中 peer 的平均下载速度，如图第一个柱状体所示。从统计结果看出，采用改进 BT 协议的 peer 平均下载速度较于标准 BitTorrent 提升了 21%-66%。且相似镜像的种子越多，peer 平均下载速度越快，下载完成时间越少。

3.2.3　改变相似度的加速实验

设定有一个种子节点，有十个下载节点同时下载目标镜像，节点的上传/下载带宽为 10/100MB/s，选取五种相似镜像，其与目标镜像的相似度分别为 10%、20%、50%、70%和 100%。对于每种相似镜像分别做实验，每次实验向目标镜像的节点群中加入 5 个该相似镜像的种子，测量 peer 的平均下载速度。测量结果如图 7 所示：

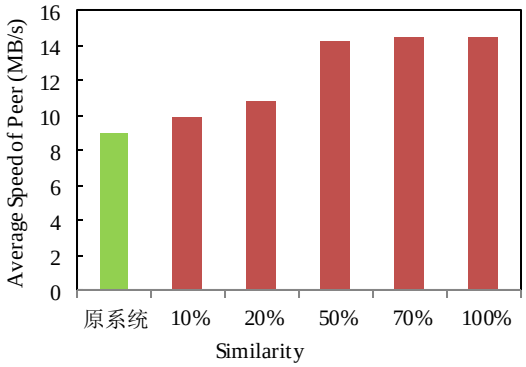

图 7　相似度对 peer 平均下载速度的影响

从图 7 中可以看出，采用改进 BT 协议时，加入不同相似镜像的种子，peer 平均下载速度较之标准 BitTorrent 协议均有提升。且目标镜像与相似镜像相似度越高，peer 平均下载速度越快。当相似度大于一定的值时，peer 下载的加速效果变化不大。这是因为当相似镜像与目标镜像的相同块较多时，peer 还未从相似镜像的种子节点获取全部的相同块就已经完成下载了。

3.2.4　改变下载节点规模的加速实验

设定下载任务中有 1 个种子节点，加入 2 个相似镜像的种子节点，相似镜像与目标镜像的相似度为 50%，每个节点的上传/下载带宽为 10/100MB/s。改变 peer 的数量，测量 peer 的平均下载速度。实验结果如图 8 所示：

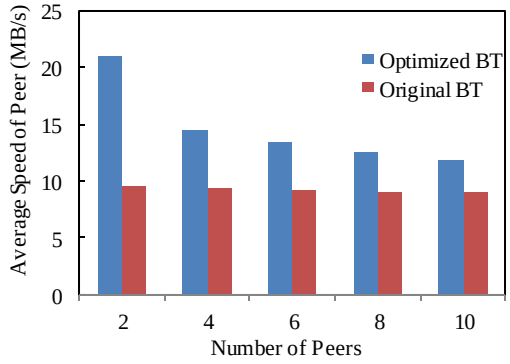

图 8　不同节点规模的 peer 平均下载速度

上图表明，其他条件不变时，采用改进 BT 协议后 peer 的平均下载速度较之标准 BT 系统中 peer 的平均下载速度均有明显的提升。并且下载节点的规模越小，加速的效果越好。这是因为 peer 越少，种子分配给每个下载节点的带宽越多，从而下载节点的获得的下载带宽越多。

4　结束语

本文以虚拟机镜像存储的相似性为依据，针对数据中心下采用 P2P 方式分发虚拟机镜像难以充分挖掘相同的块资源的缺点，设计了跨镜像

数据块分发协议，下载节点能够向拥有相似镜像的种子节点请求相同的数据块。本文指出 BT 协议以块为单位分发数据，通过下载正确的数据块并最终组合成目标文件。BT 协议的该特性是跨镜像数据块分发协议成功的核心因素。基于相似文件协助的 BT 协议显著提高了系统中服务提供者的数量，增加了系统中目标镜像的块资源。实验证明，优化的 BT 协议和原 BT 协议相比在实际运行中能显著提高平均下载速率。

参考文献

[1] Schmidt M, Fallenbeck N, Smith M, et al. Efficient Distribution of Virtual Machines for Cloud Computing[C]// Proceedings of the 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing. IEEE Computer Society, 2010:567-574.

[2] Zhang Z, Lu X, Peng Y, et al. A Reality Check of Multiple Snowball Tree File Dissemination in Large Scale Cloud Cluster[C]. In IVCE, ISORC Workshops. 2012:76-80.

[3] Chen Z, Zhao Y, Miao X, et al. Rapid Provisioning of Cloud Infrastructure Leveraging Peer-to-Peer Networks[C]. In Distributed Computing Systems Workshops, ICDCS Workshops. 2009:324-329.

[4] Li P, Ge W. Research on Dynamic Deployment for Virtual Machine in Cloud Computing[J]. Computer Measurement & Control, 2013.

[5] Reich J, Sherman A, Laadan O, et al. VMTorrent: Scalable P2P Virtual Machine Streaming[C]. In CoNEXT'12. 2012:289-300.

[6] Peng C, Kim M, Zhang Z, et al. VDN: Virtual Machine Image Distribution Network for Cloud Data Centers[J]. Proceedings - IEEE INFOCOM, 2012, 131(5):181- 289.

[7] Razavi K, Ion A, Kielmann T. Squirrel: Scatter Hoarding VM Image Contents on IaaS Compute Nodes[C]// Proceedings of the 23rd international symposium on High-performance parallel and distributed computing. ACM, 2014:265-278.

[8] Jayaram K, Peng C, Zhang Z, et al. An Empirical Analysis of Similarity in Virtual Machine Images [C]. In Proceedings of the Middleware 2011 Industry Track Workshop. 2011: 6.

[9] 王海斌. 动态虚拟集群部署与管理[D]. 吉林大学, 2011.

[10] 袁金艳. 多虚拟机快速部署机制的研究[D]. 华中科技大学, 2008. DOI:10.7666/d.d072323.

[11] Chowdhury M, Zaharia M, Ma J, et al. Managing Data Transfers in Computer Clusters with Orchestra.[J]. Acm Sigcomm Computer Communication Review, 2011, 41(4):98-109.

[12] Peterson R S, Sirer E G. Antfarm: Efficient Content Distribution with Managed Swarms[C]// Proceedings of the 6th USENIX symposium on Networked systems design and implementation. USENIX Association, 2012.

[13] Peterson R S, Wong B, Sirer E G. A Content Propagation Metric for Efficient Content Distribution.[J]. Acm Sigcomm Computer Communication Review, 2011, 41(4):326-337.

韦立超(1991-),男,安徽阜阳人,硕士生,主要研究领域为云计算应用,P2P 网络.