10 | 对象存储：看似简单的存储服务都有哪些玄机？

2020-03-25 何恺铎 来自北京

《深入浅出云计算》

你好，我是何恺铎。今天起，我们展开来讲具体的 PaaS 服务。

我第一个要深入介绍的服务，当仁不让就是**对象存储**（Object Storage）了。因为它可以说是应用最广泛、最常见的基础性 PaaS 服务了，几乎每个云上的项目都会用到它。

对象存储，顾名思义，就是在云端，你可以存放任意对象的存储服务。你要注意，**这里的“对象”指的是任意的二进制对象，保存到云上通常是以二进制文件的形式，你不要和“面向对象编程”中的对象混淆起来。**

对象存储的历史，说起来和云计算一样悠久。AWS 著名的对象存储服务 S3（Simple Storage Service）早在 2006 年就发布了，甚至比它的虚拟机服务 EC2 还要早上几个月。

S3 对象存储服务从一开始发布，就以其简明易用、高可用低成本的特点，很快受到了市场的广泛欢迎。各个云计算厂商也纷纷跟进，推出了自己的对象存储产品。到现在，对象存储已经

是云计算领域的“标配”了。

说到这里你可能会问，对象存储听上去的确很简单，无非就像一个文件服务器而已，需要用单独的篇幅来展开介绍吗？

答案当然是**肯定的**。要知道，对象存储不但注重打造存储的核心能力，还建立了一整套成熟的管理控制机制，更能够方便地与各种应用程序集成。所以，它值得我们来好好看一看。

注：对象存储是如此的成功，以至于有时候人们会用“云存储”来称呼它。但理论上来说，云存储是一个更加宽泛的概念，可以包含多种云上存储产品。我们这里还是更严谨地称之为对象存储。

初识对象存储

那么，对象存储，究竟为我们提供了什么功能呢？

通俗地解释起来，你可以这样理解，对象存储是你在云上可以创建的一种“网盘”。这个网盘可以存储任意的二进制文件，包括结构化和非结构化数据。你可以随时上传下载，也可以修改和删除。当然，云上对象存储会保证你数据的可靠性、可用性和扩展性，你不需要操心这些细节。

那么，**同样是存储服务，对象存储和前面我们 IaaS 部分讲过的云硬盘有什么区别呢？**

这是好问题。这两者之间，虽然都是存储服务，也都有多副本的冗余机制，但还是有相当大的区别。

第一个主要区别，在于访问的接口与形式。

云硬盘其实是挂载到虚拟机的虚拟硬盘，它是通过实现操作系统级别的底层接口，作为虚拟机的**块存储设备**而存在。我们也必须连接到相关的虚拟机，才能访问它里面的数据。

而对象存储，本质是一个**网络化的服务**，调用方主要通过高层的 API 和 SDK 来和它进行交互。不管是面向外部公开互联网服务，还是和内部应用程序对接，对象存储都是通过提供像

HTTP 这样的网络接口来实现的。所以它的独立性很强，不需要依赖其他组件就可以运作。

这也正是我们把对象存储放在 PaaS 篇，而不是 IaaS 篇中讲解的原因。虽然它的功能很“基础”，但它的产品形态是非常典型的 PaaS，因为你不需要操心下面支撑它的具体机器和可用性等等问题，只需要依赖它，在它之上构建你的应用就行了。

注意：尽管有 S3FS、OSSFS 等工具也可以模拟磁盘并挂载到虚拟机，但它们也是基于对象存储的 API 进行了封装，并不改变对象存储是网络化服务的本质。

第二个主要区别，也是对象存储的一大特征，就是对象存储内本身不存在一个真正的文件系统，而是更接近一个键值（Key-Value）形式的存储服务。

这里的键就是对象的路径（路径中包含斜杠符号“/”），这里的值就是存储对象的二进制文件。

键值系统和云硬盘上经典文件系统的**核心差异**，就在于文件系统保存了更多的元数据，尤其是实现了目录结构和目录操作。而键值系统中，所谓的目录其实是多个对象共享的路径前缀，可以说是用前缀模拟出了目录。

这个键值系统的设计理念，给对象存储带来的好处就是简化了逻辑和设计，可以让云厂商把更多精力放在对象存储的分布式架构和服务高可用上面。

当然相应地，这样的设计也使得对象存储中的“目录”操作代价变高了，比如说目录的删除和重命名，我们就需要对目录下所有的对象文件进行修改或删除来模拟。所以，很多对象存储系统都默认不提供目录级别的操作功能，或是性能相对较差，这一点我们需要注意。

第三个主要区别，在于对象存储的巨大容量。

作为云计算最具代表性的服务之一，它的**可扩展性**（Scalability）是毋庸置疑的，对象存储能够轻松地容纳上 PB 的超大容量数据，这是任何的云硬盘所不能企及的。所以**对象存储是名副其实的大数据存储。**

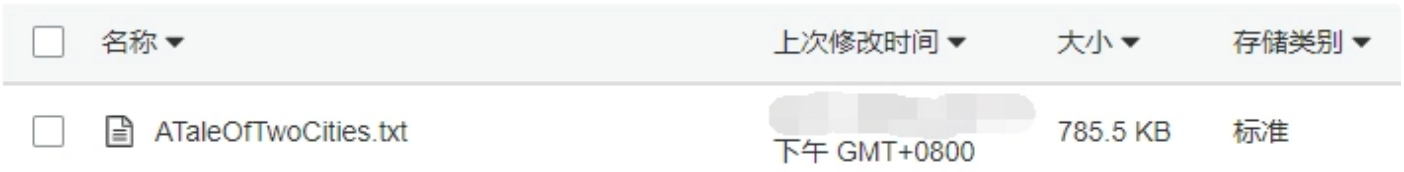
但从另一个角度说，对象存储和 HDFS 这样的大数据文件系统比起来，又有自己独到的优势：对象存储本身也是非常擅长和适合处理小文件的，即便是海量的小文件，对象存储也不会像 HDFS 那样处理起来捉襟见肘，可以说是“大小通吃”。

好的，现在我们不但把对象存储和云硬盘的区别搞清楚了，同时也理解了对象存储的最主要特征。

百闻不如一见，我们接下来进行**实操**。这次的实验我们使用国际版 AWS S3，当然你也可以使用阿里云 OSS 和 Azure Blob Storage 等类似服务进行体验。

首先，我们在 S3 门户创建一个基本的存储桶 geektime-hellocloud。这个存储桶，你可以认为是一个对象存储的基本容器，这里的名称一般要求全球唯一，在区域方面我们选择美国西部。

随后，我们点击进入这个存储桶实例，上传一个用于实验的文本文件，我们还是使用小说《双城记》的文本（ATaleOfTwoCities.txt）。成功上传后，能够看到文件已经存在于桶内。



点开这个文件，我们可以查看这个对象的一些基本属性，也能够进行一些基本操作。

在上图中点击“复制路径”按钮，你会得到一个 URL 为：

> **s3://geektime-hellocloud/ATaleOfTwoCities.txt**

这是使用 S3 标准协议下的对象路径，它也是对象的唯一标识。这个路径可以在所有支持 S3 协议的场景下使用，比如 AWS 的命令行工具。

下面展示了使用 AWS CLI 的 s3 命令，把我们这个文件下载到虚拟机当前目录的方法（事先我们已使用 aws configure 登录）：

复制代码

```
[ec2-user@ip-xx-xx-xx-xx s3test]$ aws s3 cp s3://geektime-hellocloud/ATaleOfTwoCi
download: s3://geektime-hellocloud/ATaleOfTwoCities.txt to ./ATaleOfTwoCities.txt
[ec2-user@ip-xx-xx-xx-xx s3test]$ ls
ATaleOfTwoCities.txt
```

注意，前面对象属性截图的底部（红框中的“对象 URL”），还提供了一个 HTTP 协议的对象路径，你一定不要把它和 S3 协议的路径混淆起来，因为这两者是用于不同的环境的。HTTP 协议的 URL 可以让通用的 Web 客户端直接访问这个对象。

不过现在如果我们直接请求这个 URL 的话，我们会吃一个闭门羹：

复制代码

```
[ec2-user@ip-xx-xx-xx-xx s3test]$ curl -I https://geektime-hellocloud.s3-us-west-
HTTP/1.1 403 Forbidden
...
```

这是因为在默认情况下，这个 URL 并不对公开互联网开放，你需要手动地在权限管理 Tab 中打开这个限制：

打开公有访问权限后再次实验，我们就能够成功地访问到文件的内容了：

复制代码

```
[ec2-user@ip-xx-xx-xx-xx s3test]$ curl -s https://geektime-hellocloud.s3-us-west-
The Project Gutenberg EBook of A Tale of Two Cities, by Charles Dickens

This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever.  You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
...
```

注意：打开对象存储的公开访问需要非常小心。历史上出现过非常多次因为误设置了公开权限而导致重要数据泄露的事故。一般来讲，更推荐使用更严格的基于身份认证的访问模式。

你看，对象存储是不是特别简明易用？而且得益于自带的冗余机制，它一般都有高达 99.999999999%（11 个 9）的数据可靠性，上传到其中的数据，几乎可以说是万无一失了。再结合它低成本的特点来看，对象存储也非常适合作为数据备份的场所。

对象存储的高级特性

学习了对象存储的基本知识和操作之后，接下来我们探讨一些它的高级特性。即便你是对象存储的熟手，这里面也很可能有一些你之前并不了解的门道。

第一个重要特性，是存储分层。

在生产环境下的对象存储，我们往往会存放大量的文件和数据，这些文件的访问频率其实是会有很大差异的。比如说，对于一些比较热门的下载文件，它可能经常需要被访问调用；而如果是一些明细的日志文件，写入后再次读取的机率通常不高，只有当排查问题时，我们才可能去访问翻看它。

所以为了应对不同的访问模式和频率，对象存储贴心地提供了分层的策略，你可以按照访问热度，设置**从热到冷**不同的存储级别（或者叫存储类型）。其中，存储级别为**热**的对象，存储空间占用的**成本稍高**，但访问读取不需要收取额外的费用；而存储级别越**冷**，则存储空间的单位**成本越低**，但访问读取需要收取一定的费用。到了极少访问的存档级别，数据的“解冻”可能还需要花费一些时间。

不同云的存储级别叫法有一些区别，我这里用一个表格给你做了大致的梳理：

访问频率	阿里云	AWS	Azure
高频	标准	Standard	热（Hot）
低频	低频访问	Standard-IA, OneZone-IA	冷（Cool）
极少访问	归档	Glacier, Glacier Deep Archive	存档（Archive）

所以，这些存储级别其实是一种在访问效率和存储成本之间的平衡。对象存储服务把这样的一个选择权开放出来，是一个非常有用的特性，能够让你根据具体的文件情况，因地制宜，选择不同的策略。而且这些策略既可以是存储桶级别的，也可以细到单个文件，非常灵活。

重要提示：同一个文件的存储类型是可以按需转换的，既可以从热到冷，也可以从冷到热。但你需要注意，这个切换动作本身可能会收取额外的费用，所以不应该经常地切换，这样会得不偿失。

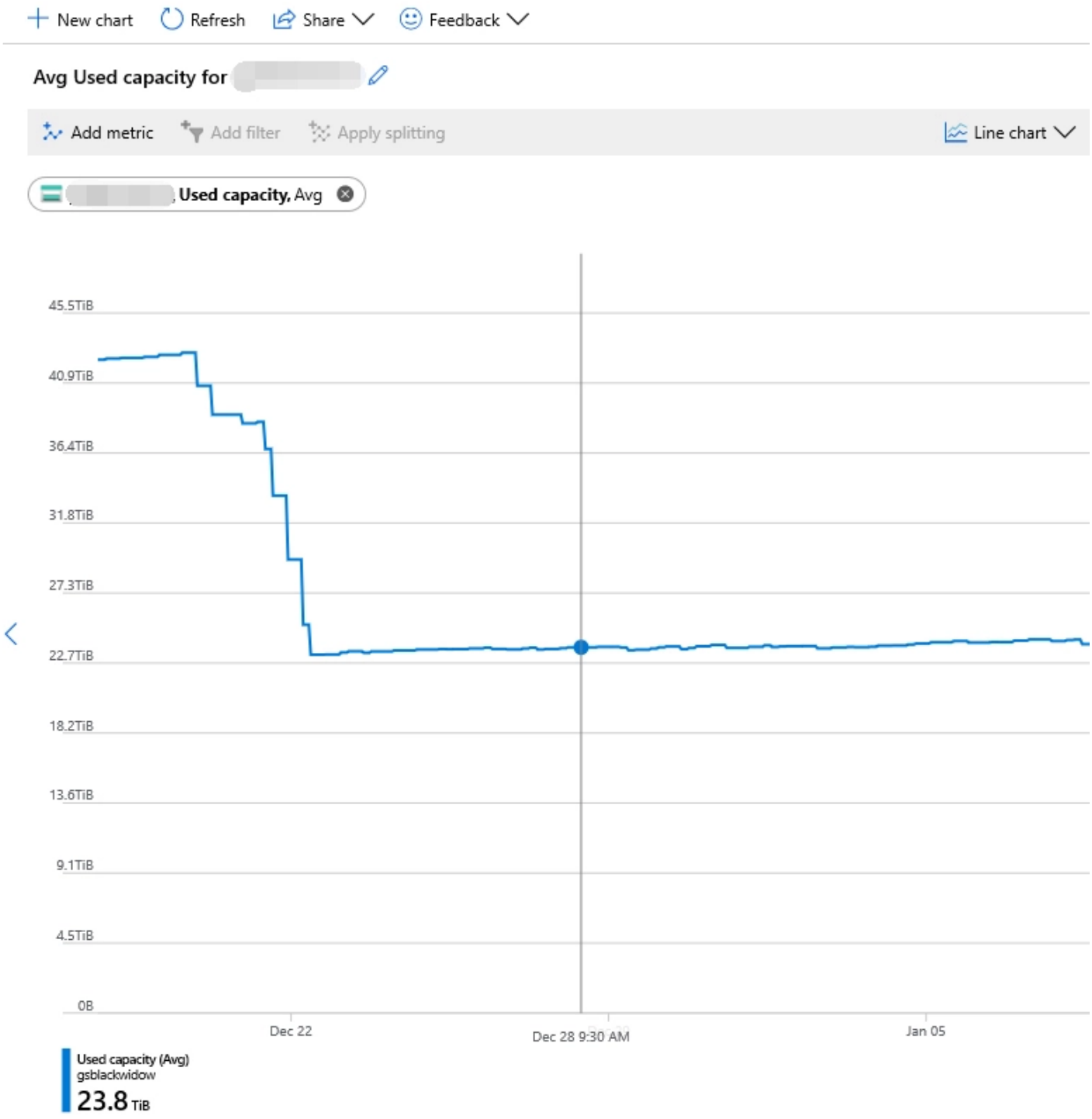
可以说，**存储分层的存在，让原本价格低廉的云上存储更加具有成本竞争力。**给你举个例子，现在归档层的存储费用，在典型情况下大约是每 GB 每月 1 分钱左右，是不是低得惊人？所以，很多用户上云的一个应用场景就是，把原本占用大量传统磁盘的备份文件，利用对象存储的归档能力长期保存。

第二个值得称道的特性，是生命周期管理。

随着时间的推移、业务的增长，你在对象存储中的内容肯定会越来越多。当总的体量和对象的个数到达一定级别的时候，你会发现对历史内容进行清理就成为了一件非常麻烦的事情。

这时候，生命周期管理功能就可以很好地帮助我们。因为它允许你设置一定的过期规则，当对象满足规则时（通常每天判断一次），可以自动地执行一些清理操作。比如，你可以对一个存储桶或目录进行设置，要求最后修改时间超过 60 天的文件自动切换到低频访问层，超过 180 天的文件则进行归档或删除。

我曾经就在某个生产环境中，启用了这个自动清理特性，立竿见影地节省了大量成本，如下图所示。



第三个特性，则是对象的版本管理（Versioning）。

这个很好理解。同一个对象可能会被修改更新，而启用这个特性后，对象存储系统就能够自动地帮助你记录这个对象之前的多个版本。这样，当有需要时，你可以按需进行回滚和恢复，能

避免不必要的损失。

此外，对象存储服务还有跨区域同步、访问日志分析等其他高级特性。前者可以帮助你自动对数据进行跨区域同步，常用于重要数据备份或热点数据分发，后者则对已经存放了海量数据的对象存储进行管理分析大有帮助。有兴趣的话，你都可以自己尝试一下。

对象存储的应用场景

我们的应用离不开数据，所以几乎到处都是对象存储可以发挥的场景。**一切需要保存数据的地方，不论是原始数据的保留备份、中间结果的临时落地，还是处理结果数据的永久保存，你都可以考虑对象存储是否适用。**

是的，在很多系统中，对象存储就是这样贯穿在整个系统数据流程的生命周期中，串联起了数据处理的各个环节。对象存储有时甚至还可以用来做简单的键值数据库，由于它的分布式设计，对它来说，承担大量的并发请求，也是小菜一碟。

对象存储还可以支撑大数据应用。现在各云厂商的对象存储服务，也普遍地作为分布式存储系统，与各家的大数据 PaaS 产品进行了深度的集成，也是云上各类数据湖解决方案的关键组成部分。我们后面讲到大数据 PaaS 服务时还会详细讨论。

最后，通过前面的实验，我们能看到，对象存储可以直接面向公开互联网，作为文件服务器对外提供服务。通过妥善设置对象的 HTTP 响应头，它甚至还能支撑起静态网站，免去我们创建虚拟机的麻烦。如果下载量比较大，且对带宽延时有更高要求的话，它又能无缝地与云上的 CDN 服务进行集成，作为 CDN 的回源站点。

课堂总结与思考

因为对象存储的高可用、低成本的特性，让它成为了云上最重要、最受欢迎的支柱性 PaaS 服务之一，也极大地助推了云计算本身的发展。它上手起来非常简单，而深入运用起来又很强大，可以说是产品设计上的最高境界了。

对象存储在实践中实在有太多妙用，等待着你去感受和发现。我建议你多多实际操作，探索一遍它的各个功能选项，这会比你单纯地阅读产品文档有更深入的体会。

今天给你的思考题是这样的，欢迎你在留言区和我互动：

> 将对象设置为完全公开是非常危险的，但如果我们要临时地分享一个对象，给特定的外部用户，应该怎样做呢？
>
> 假设你在本地数据中心，有大量的数据需要上传到云对象存储中，但互联网的带宽有限，上传需要很长的时间。对于这种情况有什么好办法吗？

好了，这一讲我们就到这里。如果你觉得有收获，也欢迎你把这篇文章分享给你的朋友。感谢阅读，我们下期再见。

精选留言 (12)

qinsi
2020-03-25

1. 链接中带过期时间并签名，超时后链接自动失效 2.邮寄硬盘

作者回复: 言简意赅的好答案！

共 2 条评论 >　　27

摇滚诗人M
2020-03-27

可以单独分享要分享的对象为公开，或者使用签名url。某公司和云存储之间带宽不够，可以加一条虚拟专用网到数据中心，带宽还不够的话，找运营商拉专线，最后云厂商有专用硬件可以上传大量数据的。

作者回复: Perfect.

7

ican_只会0到9
2020-03-25

将对象设置为完全公开是非常危险的，但如果我们要临时地分享一个对象，给特定的外部用户，应该怎样做呢？

通过特定的URL设置访问权限，分享给特定外部用户

2。假设你在本地数据中心，有大量的数据需要上传到云对象存储中，但互联网的带宽有限，上传需要很长的时间。对于这种情况有什么好办法吗？

结合CDN作为回源站点

作者回复: CDN是用于数据的“下发”，而不是往云上“上传”哦。

共 2 条评论 >

3

Joe Black

2020-04-14

对象存储的访问速度会不会不太理想呢？毕竟要写副本，还要走http协议。

作者回复: 看以什么标准来判断了。首先对象存储的吞吐能力其实相当不错，所以做大数据分析是没有什么问题的；主要操作延迟方面如果和内存数据库比可能差一些，但也不算低，在实时性要求不高的场合，一般也能够接受的。

胖子

2020-04-14

老师，单块云硬盘的最大容量由哪些因素决定的？我认为云硬盘的底层实现也是基于分布式架构的。

作者回复: 很好的问题。理论上云硬盘是可以用纯软件分布式实现，但因为实际场景下要考虑性能、延迟、高可用，而且块设备是走非常底层的协议，所以基于存储硬件加low-level冗余机制来实现更合理。这也是为什么云盘有容量上限的原因。各个厂商的实现是商业秘密，应该也各有差异，很可能会配套专用存储设备和定制芯片。

Regis

2020-04-09

老师，因为我们数据权限要求不能放到公有云上，所以最近正在搭建一个新的存储平台，因为对象存储的3副本机制会占用大量的成本，1PB的数据就需要购买3PB的存储空间，成本很高，所以需要分场景处理，一部分数据做对象存储，另外一部分数据打算搭建NAS或SAN存储，老师对这样的场景有什么好的建议吗？

作者回复: 私有云场景其实可以考虑采购存储厂商的一些存储硬件设备，这些设备能够提供容量和性能的保证，现在也吸取了云上存储分层等优点。你还可以考虑云厂商的“云存储网关”类的产品，帮助你私有云里的存储自动拓展到公有云上，一般也都支持上云自动加密，可以放心使用。


zhang

2020-03-27

老师，对象存储中的数据是如何归档的？ 归档之后为什么会便宜这么多呢？可以说归档存储的成本跟普通硬盘的性价比差不多了。

作者回复: 归档能做到非常便宜，是因为存储介质不同。云厂商一般都不会透露具体的实现方法，但一般认为可能用到了磁带、光盘或低性能廉价硬盘，总之是通过特殊存储介质来实现的。这也部分解释了，为什么归档层数据的恢复需要比较长的时间。


LindaWang

2020-03-26

1. 阿里云可以通过设计Bucket Policy来授权其他用户访问指定资源
2. 有的厂商会提供专门的工具，如GCS（Google Cloud Storage）会提供gsutil，通过-m参数，指定执行 并行copy (multi-threaded/multi-processing)

作者回复: 启用并行传输的确是一个最佳实践。不过提高并行度也有一个理论上限，那就是本地数据中心到互联网的带宽。很多时候这个带宽并不高，无法满足大数据量快速传输。这时，还有其他的解决方法吗？

共 2 条评论>

leslie

2020-03-25

我觉得这就像我们平时用的云盘啊

第一个问题：应当可以通过类似于云盘加密的方式吧；
第二个问题：是不是可以先传到云服务器，云服务器和云存储之间的架构类似于内网；这样会方便和快许多。
谢谢老师今天的分享，期待后续分享。

作者回复: 本地数据中心的数据传到云存储，和传到云虚拟机，都是通过互联网连接到云数据中心，带宽本质上区别不大的。所以这个方法不能解决问题。

许童童
2020-03-25

上传的话，可以先把数据压缩，上传成功后再在云端解开

戴斌
2020-03-25

我们也用到了阿里云的OSS对象存储存放一些用户上传的文件，为集群节点存储数据带来了遍历，扩容节点的时候不再考虑存储问题。

作者回复: 是的，这个场景很合适。

八哥
2020-03-25

CMS或者博客系统，上传的图片，应该需要公开访问，否则未登录的用户看不到图片了？

共 2 条评论 >