

阿里云 专有云企业版 对象存储 OSS

技术白皮书

产品版本：V3.12.0

文档版本：20200619

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务时间约十分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意： 权重设置为0，该服务器不会再接受新请求。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	单击 设置 > 网络 > 设置网络类型 。
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面，单击 确定 。
Courier字体	命令。	执行cd /d C:/window命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid Instance_ID
[]或者[a b]	表示可选项，至多选择一个。	ipconfig [-all -t]
{ }或者[a b]	表示必选项，至多选择一个。	switch {active stand}

目录

法律声明	I
通用约定	I
1 什么是对象存储	1
1.1 基本定义	1
1.2 产品优势	1
1.3 应用场景	2
2 技术优势	4
3 产品架构	6
3.1 系统架构	6
3.2 数据流转流程	8
4 功能原理	10
4.1 功能组件	10
4.2 功能特性	10
4.3 基本概念	12

1 什么是对象存储

1.1 基本定义

阿里云对象存储OSS（Object Storage Service）是阿里云提供的海量、安全、低成本、高可靠的云存储服务。

OSS可以被理解成一个即开即用，无限大空间的存储集群。相比传统自建服务器存储，OSS在可靠性、安全性、成本和数据处理能力方面都有着突出的优势。使用OSS，您可以通过网络随时存储和调用包括文本、图片、音频和视频等在内的各种非结构化数据文件。

OSS将数据文件以对象（Object）的形式上传到存储空间（Bucket）中，提供键值对（Key-Value）形式的对象存储服务。您可以根据Object的名称（Key）获取该Object的内容。

您可以进行以下OSS相关的操作：

- 创建存储空间，并向存储空间中上传文件。
- 获取已上传文件的地址，进行文件的分享和下载。
- 修改存储空间或文件的属性或元信息，为其设置相应的权限。
- 在对象存储OSS控制台进行基础和高级OSS操作。
- 通过开发工具包SDK或直接在应用程序中调用RESTful API，进行基础和高级OSS操作。

1.2 产品优势

阿里云对象存储OSS，是阿里云提供的海量、安全、低成本、高可靠的云存储服务。本文将OSS与传统的自建存储进行对比，让您更好的了解OSS。

OSS与自建存储对比的优势

对比项	对象存储OSS	自建服务器存储
可靠性	• 规模自动扩展，不影响对外服务。 • 数据自动多重冗余备份。	• 受限于硬件可靠性，易出问题，一旦出现磁盘坏道，容易出现不可逆转的数据丢失。 • 人工数据恢复困难、耗时、耗力。

对比项	对象存储OSS	自建服务器存储
安全	- 提供企业级多层次安全防护。 - 多用户资源隔离机制，支持同城容灾。 - 提供多种鉴权和授权机制，以及白名单、防盗链、主子账号、STS临时授权访问功能。	- 需要另外购买清洗和黑洞设备。 - 需要单独实现安全机制。
数据处理能力	提供图片处理功能。	需要额外采购，单独部署。

OSS的其他优势

- 方便、快捷的使用方式

提供标准的RESTful API接口（部分接口与Amazon S3 API兼容）、丰富的SDK包、客户端工具、控制台。您可以像使用文件一样方便地上传、下载、检索、管理用于Web网站或者移动应用的海量数据。

- 不限文件数量和大小。您可以根据所需存储量无限扩展存储空间，解决了传统硬件存储扩容问题。
- 支持流式写入和读出。特别适合视频等大文件的边写边读业务场景。
- 支持数据生命周期管理。您可以自定义将到期数据批量删除。

- 强大、灵活的安全机制

灵活的鉴权、授权机制。提供STS和URL鉴权和授权机制，以及白名单、防盗链、主子账号功能。

- 丰富的图片处理服务

支持JPG、PNG、BMP、GIF、WebP、TIFF等多种图片格式的转换，以及缩略图、剪裁、水印、缩放等多种操作。

1.3 应用场景

本文介绍对象存储OSS的主要应用场景。

图片和音视频等应用的海量存储

OSS可用于图片、音视频、日志等海量文件的存储。各种终端设备、Web网站程序、移动应用可以直接向OSS写入或读取数据。OSS支持流式写入和文件写入两种方式。

网页或者移动应用的静态和动态资源分离

利用BGP带宽，OSS可以实现超低延时的数据直接下载。

离线数据归档存储

根据OSS低成本和高可用的特性，可以将企业内部长期需要离线归档的数据转存至OSS。

2 技术优势

对象存储OSS是阿里云提供的海量、安全、低成本、高可靠的云存储服务，本文介绍OSS的技术优势。

多功能

OSS支持以下功能：

- 支持对象的简单上传、表单上传、追加上传、下载、删除、列举、复制，获取对象的元数据、创建多段上传任务。
- 支持列举存储空间、创建存储空间、删除存储空间、列举存储空间内对象、获取存储空间的元数据。
- 支持构建全网统一的对象存储空间，支持跨域多中心的存储空间复制功能。
- 支持生命周期管理、定义和管理存储空间内所有对象或对象的某个子集的生命周期、变更容量和变更归属。
- 支持同城容灾，同城容灾采用存储空间同名的模式，根据配置自动开启集群级别的容灾。即用户创建一个主存储空间后，会自动的创建一个同名的备存储空间，并自动开启主到备存储空间的同步。
- 支持将存储空间配置成静态网站托管模式，并通过存储空间域名访问该静态网站。
- 支持防盗链，OSS支持设置基于HTTP header中表头字段Referer的防盗链方法。
- 支持跨域访问。提供日志记录功能，可追查访问来源，并提供多维度的统计分析功能。
- 采用全冗余架构，无单点故障。
- 支持大文件的分片并发上传和下载，支持断点续传。

高性能

支持集群规模达到上万节点吞吐量。

安全性

OSS支持多种数据安全功能，保障您的数据安全：

- 提供权限控制ACL（Access Control List），用于控制用户的访问权限。支持三种访问权限：私有、公共读、公共读写。
- 提供主子账号功能。按照部门组织架构为员工、应用、系统进行对象存储的资源访问控制，为每个员工、应用、系统创建单独的登录密码或访问密钥。默认情况下用户没有任何操作对象存储资源的授权，可以通过访问策略机制对用户进行细粒度的授权或通过安全令牌服务进行临时授权。
- 支持服务器端和客户端加密、传输加密HTTPS。

- 对象存储提供防盗链功能，可屏蔽恶意来源的访问。
- 结合攻击防御系统，能够有效抵御DDoS攻击、CC攻击，保障业务正常运行。
- 用户存储数据通过跨区域复制功能实时同步到指定区域，实现数据异地容灾，从容应对极端灾难保证业务流畅，为重要数据加上多重保险。
- 支持同城容灾，同城容灾采用存储空间同名的模式，根据配置自动开启集群级别的容灾。即用户创建一个主存储空间后，会自动的创建一个同名的备存储空间，并自动开启主到备存储空间的同步。

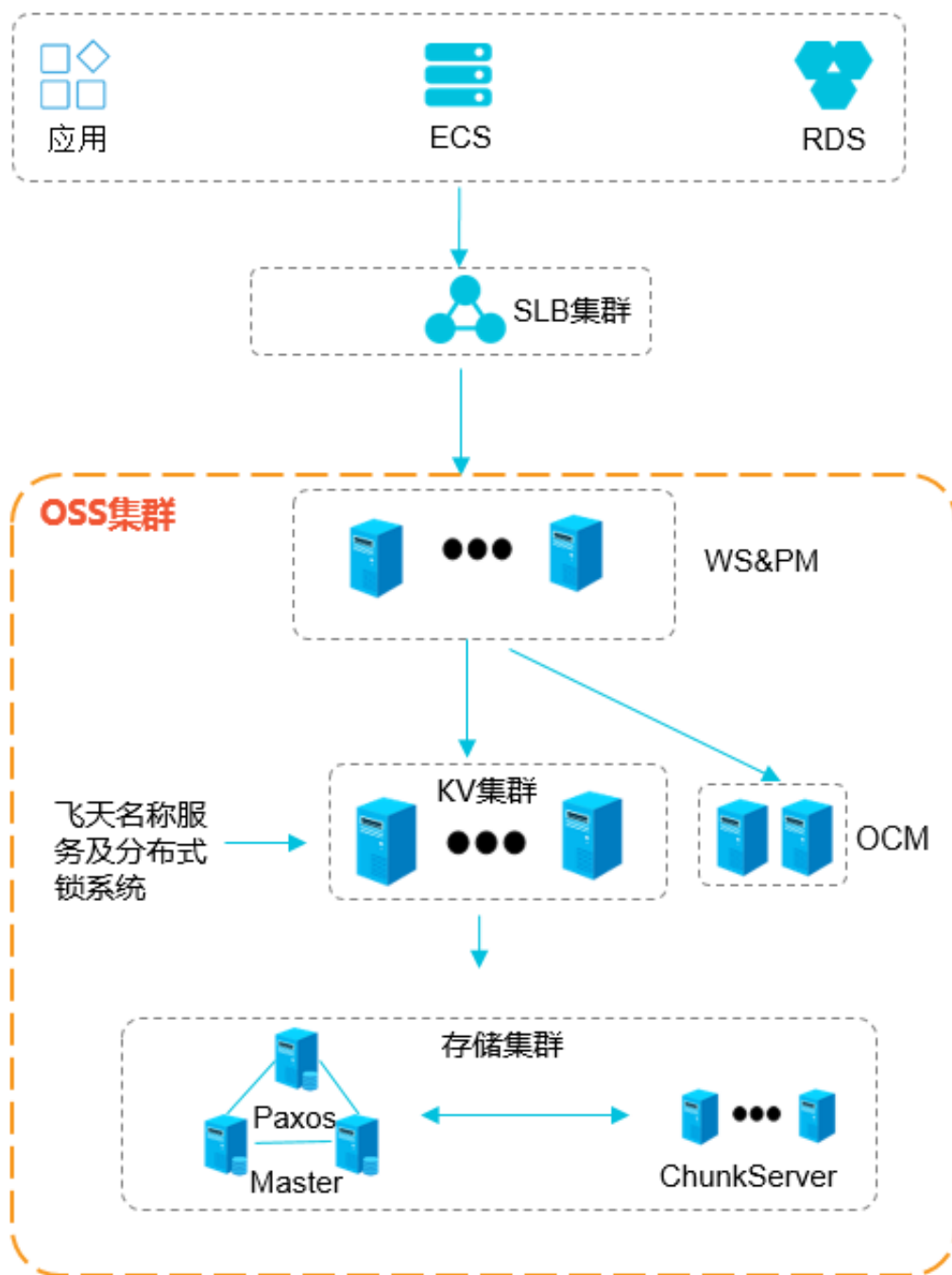
3 产品架构

3.1 系统架构

对象存储OSS是构建在阿里云飞天平台上的一种存储解决方案。其基础是飞天平台的分布式文件系统、分布式任务调度等基础设施。这些基础设施提供了OSS以及其他阿里云服务所需的分布式调度、高速网络、分布式存储等重要特性。

OSS的架构如下图所示。

图 3-1: OSS架构图



OSS的架构分为三层：协议接入层、分区层、持久存储层。

- 协议接入层

- WS: WebServer 网页服务器，使用的是开源的tengine组件，对外提供HTTP和HTTPS服务。
- PM: ProtocolModule 协议模块将HTTP请求转换为后端KV或者其他模块的操作。并负责接收用户使用RESTful协议发来的请求，进行安全认证。如果认证通过，用户的请求将被转发到Key-Value引擎继续处理；如果认证失败，直接返回错误信息给用户。

- 分区层

分区层，负责数据结构化处理，即按照Key来查找或存储数据，并支持大规模并发的请求。当协调服务集群变更导致服务被迫改变运行物理位置时，可以快速协调找到接入点。同时负责对象文件索引管理、对象文件转换成底层持久数据文件。

- 分布式协调服务，主要负责命名服务可看做nameservice及分布式锁服务。
- KV分为KVMaster和KVServer。KVMaster主要负责管理和调度分区，KVServer负责分区的索引及数据存储。

- 持久层

持久存储层，即大规模分布式文件系统。元数据存储 Master 上，Master 之间采用分布式消息一致性协议(Paxos)保证元数据的一致性。从而实现高效的文件分布式存储和访问，保证数据在系统中有3个备份以及在软硬件错误发生以后的故障恢复。

3.2 数据流转流程

本文介绍用户从访问OSS到获取相应数据的数据流转流程。

OSS数据流程：用户 → RESTful API → 负载均衡-Web服务器 → 协议模块 → KV Engine → 分布式存储，详情如下：

1. 用户使用不同的客户端，例如浏览器或者SDK，发出符合对象存储 API规范请求，到对象存储的Endpoint，对象存储的Endpoint会被解析到负载均衡的LVS VIP上，VIP的后端绑定着真实的WebServer (WS)。请求最终会落到其中的某一台WS上。
2. 请求到对象存储的Web服务器后，协议模块 (PM) 会将用户请求进行解析。首先会进行权限验证，权限验证不通过会直接返回用户相应的错误码。
3. 如果Object读写的请求被协议模块验证通过后，会被解析成对KV Engine的读写操作，进入分区层。
4. 分区层负责数据结构化处理，即按照Key来查找或存储数据，并支持大规模并发的请求。当协调服务集群变更导致服务被迫改变运行物理位置时，可以快速协调找到接入点。
5. 分区层的KV Engine存入的数据最终会被写入到持久存储层。

6. 持久存储层，即大规模分布式文件系统。元数据存储 Master 上，Master 之间采用 分布式消息一致性协议（Paxos）保证元数据的一致性。从而实现高效的文件分布式存储和访问，保证数据在系统中有 3 个备份以及在软硬件错误发生以后的故障恢复。

4 功能原理

4.1 功能组件

对象存储OSS是构建在飞天云操作系统上的一种存储解决方案，本文介绍OSS的功能组件。



OSS主要分为以下三大模块：

- 接入层：包括API、SDK以及用户管理控制台。
- 应用层：包括存储空间、对象管理、图片处理及安全模块。
- 基础设施层：盘古、伏羲、女娲等关联系统。

4.2 功能特性

对象存储OSS是阿里云提供的海量、安全、低成本、高可靠的云存储服务，本文介绍OSS的功能特性。

Bucket和Object管理

- 存储空间概览

展示请求者所拥有的所有Bucket，在通过HTTP访问OSS服务地址时将默认展示您所拥有的所有Bucket。

- 创建Bucket

一个用户默认最多创建100个Bucket。新创建的Bucket 命名要符合Bucket命名规范。

创建Bucket可能存在以下情况：

- 如果创建的Bucket不存在，系统按照Bucket名称创建Bucket，并返回成功标志。
- 如果要创建的Bucket已存在，且请求者是Bucket所有者，则保留原来Bucket，并返回成功标志。
- 如果要创建的Bucket已存在，且请求者不是Bucket所有者，则返回失败标志。

- 删除Bucket

删除Bucket的操作需要满足以下条件：

- Bucket存在。
- 访问者对Bucket有删除权限。
- Bucket为空。

- 列举Bucket中的所有Object

根据Bucket名称列出此Bucket下的所有Object信息，访问者必须具有对相应Bucket的操作权限，当访问的Bucket不存在时返回错误信息。

OSS支持前缀查询，可以设置一次最大返回的文件数量（最大支持设置1000）。

- 上传Object

上传Object到指定的 Bucket 空间下。在满足如下条件下 Object 会上传成功：Bucket存在，且访问者拥有对Bucket相应的操作权限。当Bucket中存在同名Object文件时将会覆盖掉原来的Object文件。

- 删除Object

根据Object名称删除某个特定Object，访问者必须有此Object相应的操作权限。

- 获取Object

取得Object内容信息或者元信息，访问者需要对Object有相应的操作权限。

- 访问Object

OSS支持通过URL方式访问某文件。

安全管控

- 设置并查询Bucket访问权限

用户可以设置和查看Bucket的访问权限。Bucket的访问权限可以为以下任意一种：

- Private（私有权限）：只有该存储空间的创建者或者授权对象可以对该存储空间内的文件进行读写操作，其他人在未经授权的情况下无法访问该存储空间内的文件。
- Public-read（公共读，私有写）：只有该存储空间的创建者可以对该存储空间内的文件进行写操作，任何人（包括匿名访问）可以对该存储空间中的文件进行读操作。
- Public-read-write（公共读写）：任何人（包括匿名访问）都可以对该存储空间中的文件进行读写操作。

- 日志及监控操作

用户可以选择开启Bucket的日志记录功能，一旦开启，OSS会按照小时粒度推送日志。用户可以在Apsara Stack控制台OSS首页查询存储空间、流量、请求等信息。

- 防盗链

为了防止您存储于OSS的数据被其他人盗链，OSS提供了防盗链功能，即设置HTTP header中的Referer字段。您可以通过OSS控制台为您的存储空间设置Referer字段的白名单以及是否允许Referer字段为空的请求访问。例如，对于一个名为oss-example的存储空间，设置其Referer白名单为http://www.aliyun.com。则只有Referer为http://www.aliyun.com的请求才能访问oss-example这个存储空间中的对象。

4.3 基本概念

本部分将向您介绍对象存储OSS中涉及的几个基本概念，以便于您更好地理解OSS产品。

对象（Object）

Object是OSS存储数据的基本单元，也被称为OSS的文件，由元信息（Object Meta）、用户数据（Data）和文件名（Key）组成。Object由存储空间内部唯一的Key来标识。Object元信息是一个键值对（Key-Value），表示了Object的一些属性，比如最后修改时间、大小等信息，用户也可以在元信息中存储一些自定义的信息。

Object的生命周期是从上传成功到被删除为止。在整个生命周期内，Object信息不可变更。重复上传同名的Object会覆盖之前的Object，因此，OSS不支持修改文件的操作。



说明：

如无特殊说明，OSS文档中的对象、文件、Object意义相同。

存储空间 (Bucket)

存储空间是您用于存储Object的容器，所有的Object都必须隶属于某个存储空间。您可以设置和修改存储空间属性用来控制访问权限、生命周期等，这些属性设置直接作用于该存储空间内所有Object，因此您可以通过灵活创建不同的存储空间来完成不同的管理功能。

- 同一个存储空间的内部是扁平的，没有文件系统的目录等概念，所有的Object都直接隶属于其对应的存储空间。
- 每个账号可以拥有多个存储空间。
- 存储空间名称在OSS范围内必须是全局唯一的，一旦创建之后无法修改名称。
- 存储空间内部的Object数目没有限制。

强一致性

Object操作在OSS上具有原子性，操作要么成功要么失败，不会存在有中间状态的Object。OSS保证用户在完成上传后读到的Object是完整的，OSS不会返回给用户一个部分上传成功的Object。

Object操作在OSS上同样具有强一致性，用户一旦收到了一个上传（PUT）成功的响应，该上传的Object就已经立即可读，并且数据的三份副本已经写成功。不存在一种上传的中间状态，即read-after-write却无法读取到数据。对于删除操作也是一样的，用户删除指定的Object成功之后，该Object立即变为不存在。

强一致性方便了用户架构设计，可以使用跟传统存储设备同样的逻辑来使用OSS，修改立即可见，无需考虑最终一致性带来的各种问题。

OSS与文件系统的对比

OSS是一个分布式的Object存储服务，提供的是一个Key-Value对形式的Object存储服务。用户可以根据Object的名称唯一地获取该Object的内容。虽然用户可以使用类似test1/test.jpg的名字，但是这并不表示该Object是保存在test1目录下面的。对于OSS来说，test1/test.jpg仅仅只是一个字符串，和a.jpg这种并没有本质的区别。因此不同名称的Object之间的访问消耗的资源是类似的。

文件系统是一种典型的树状索引结构，一个名为test1/test.jpg的文件，访问过程需要先访问到test1这个目录，然后再在该目录下查找名为test.jpg的文件。因此文件系统可以很轻易的支持文件夹的操作，例如重命名目录、删除目录、移动目录等，因为这些操作仅仅只是针对目录节点的操作。这种组织结构也决定了文件系统访问越深的目录消耗的资源也越大，操作拥有很多文件的目录也会非常慢。

对于OSS来说，可以通过一些操作来模拟类似的功能，但是代价非常昂贵。例如重命名目录，希望将test1目录重命名成test2，那么OSS的实际操作是将所有以test1/开头的Object都重新复制成以test2/开头的Object，这是一个非常消耗资源的操作。因此在使用OSS的时候要尽量避免类似的操作。

OSS保存的Object不支持修改（追加写Object需要调用特定的接口，生成的Object也和正常上传的Object类型上有差别）。用户哪怕是仅仅需要修改一个字节也需要重新上传整个Object。而文件系统的文件支持修改，比如修改指定偏移位置的内容、截断文件尾部等，这些特点也使得文件系统拥有广泛的适用性。但另外一方面，OSS能支持海量的用户并发访问，而文件系统会受限于单个设备的性能。

因此，将OSS映射为文件系统是非常低效的，也是不建议的做法。如果一定要挂载成文件系统的话，建议尽量只做新建文件、删除文件、读取文件这几种操作。使用OSS应该充分发挥其优点，即海量数据处理能力，优先用来存储海量的非结构化数据，例如图片、视频、文档等。