



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

SSE316 : 云计算技术 Cloud Computing Technology

陈壮彬

软件工程学院

<https://zbchern.github.io/sse316.html>



云数据存储

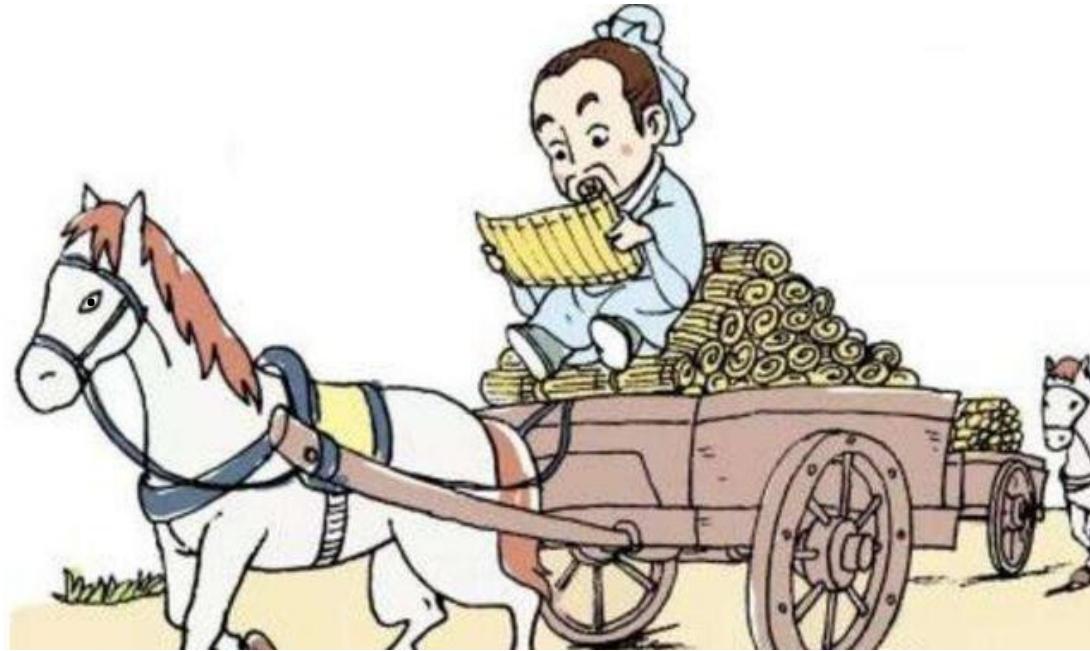
- ❖ 背景介绍
- ❖ 云数据存储的基本概念
- ❖ 分布式文件系统
- ❖ 常见的云存储服务



云数据存储

- ❖ 背景介绍
- ❖ 云数据存储的基本概念
- ❖ 分布式文件系统
- ❖ 常见的云存储服务

学富五车



曾考证过，3000多片竹简能写大约10万字，重量约12公斤，而古代马车的载重通常能达到200公斤左右，五车竹简就是1000公斤，算下来“学富五车”大概有800万字。

蔡伦造纸



学富五车 =

$\times 10^?$



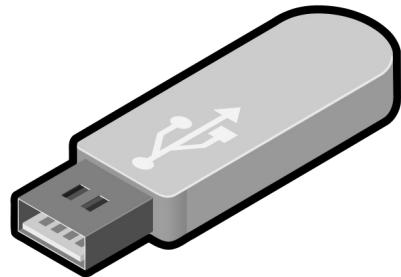
89.7万字

CD-ROM

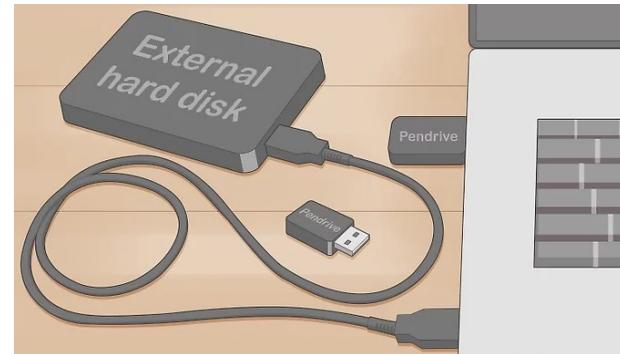


比尔·盖茨在1994年展示了一张CD-ROM可以容纳比一堆33万张纸更多的信息（700MB）。

移动存储

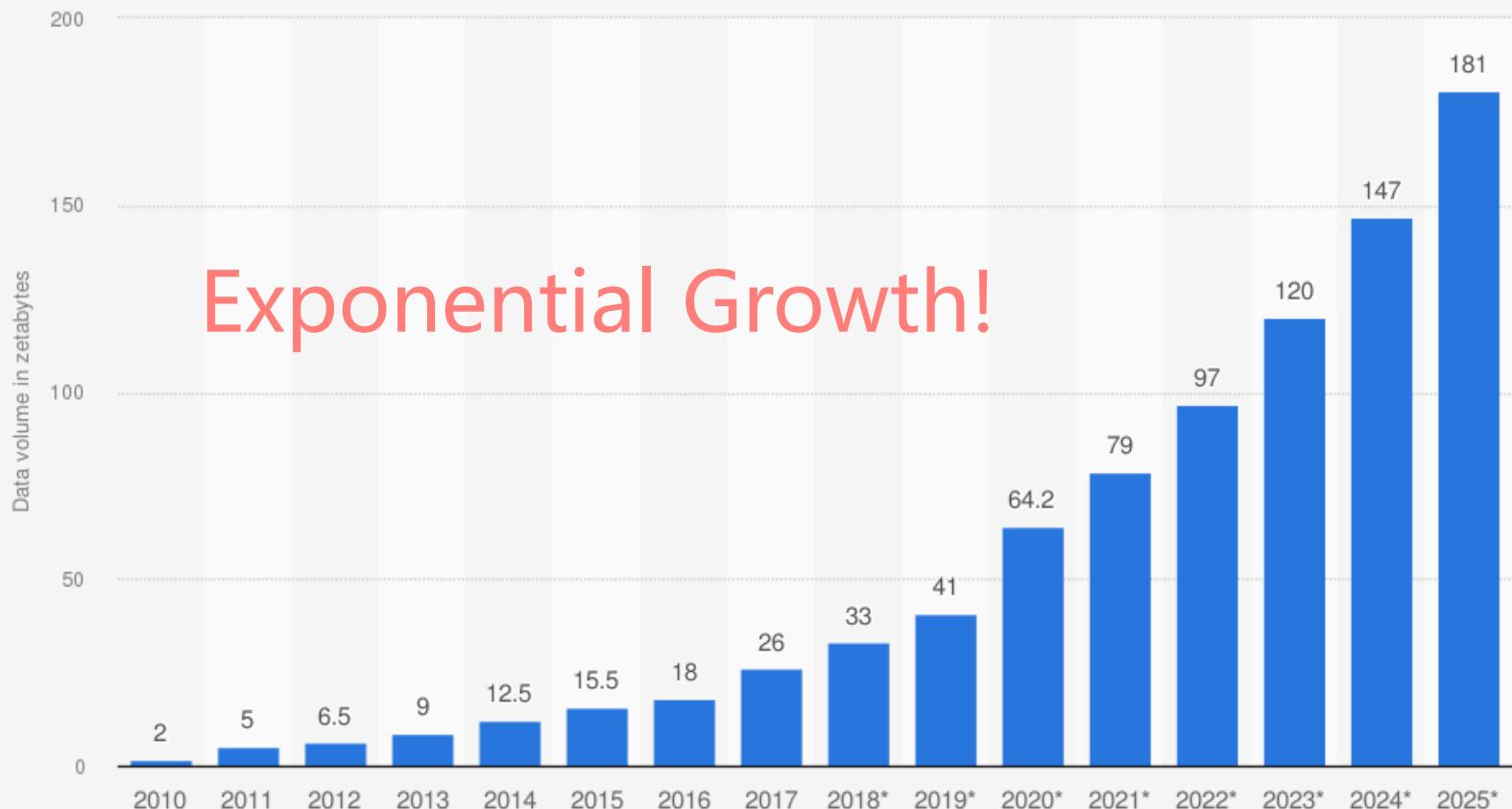


2GB - 2TB



1TB - 18TB

Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025 (in zettabytes)



Sources
IDC; Seagate; Statista estimates
© Statista 2022

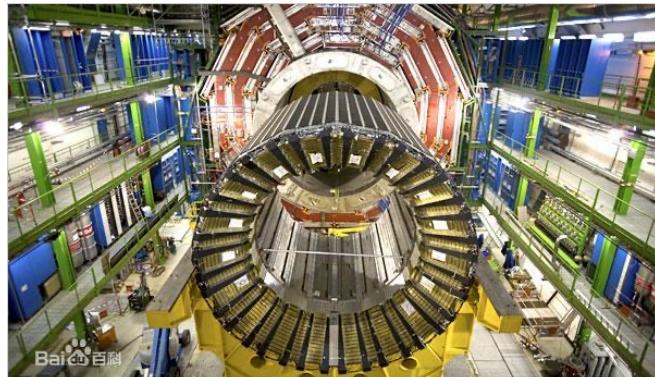
Additional Information:
Worldwide; 2010 to 2020

$$1 \text{ ZB} = 1024 \text{ EB} = 1024^2 \text{ PB} = 1024^3 \text{ TB} = 1024^4 \text{ GB}$$

超大规模数据的应用（1）



• 科学研究



大型强子对撞机



平方千米阵列射电望远镜



NASA宇宙观测

- ❖ 大型强子对撞机每年产生25PB的数据
- ❖ 电射望远镜每年产生600PB的数据

超大规模数据的应用（2）



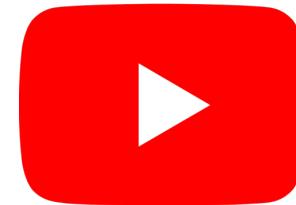
- 社交媒体平台



Weibo



Facebook



YouTube

- ❖ 2021年，Facebook的月活用户达28亿；每天新产生47.5亿条新内容，45亿个likes和3.5亿张图片
- ❖ 2021年，YouTube用户每分钟上传超过500小时的视频

超大规模数据的应用（3）



- 搜索引擎



Google



Bing



Baidu

- ❖ 截至2021年，谷歌检索超过130万亿的网页，需要EB级的存储
- ❖ 2021年，谷歌每天需要处理超过35亿的查询，等同于每秒4万次查询

磁盘技术的发展



Parameter	1956	2016
Capacity	3.75 MB	10 TB
Average access time	≈ 600 msec	2.5–10 ms
Density	200 bits/sq. inch	1.3 TB sq. inch
Average life span	≈ 2000 hours/MTBF	≈ 22,500 hours/MTBF
Price	\$9,200/MB	\$0.032/GB
Weight	910 Kg	62 g
Physical volume	1.9 m ³	34 cm ³

- ❖ 磁盘密度增加了 650×10^6 倍！
- ❖ 磁盘价格下降了 300×10^6 倍！
- ❖ 磁盘容量增加了 2.7×10^6 倍！



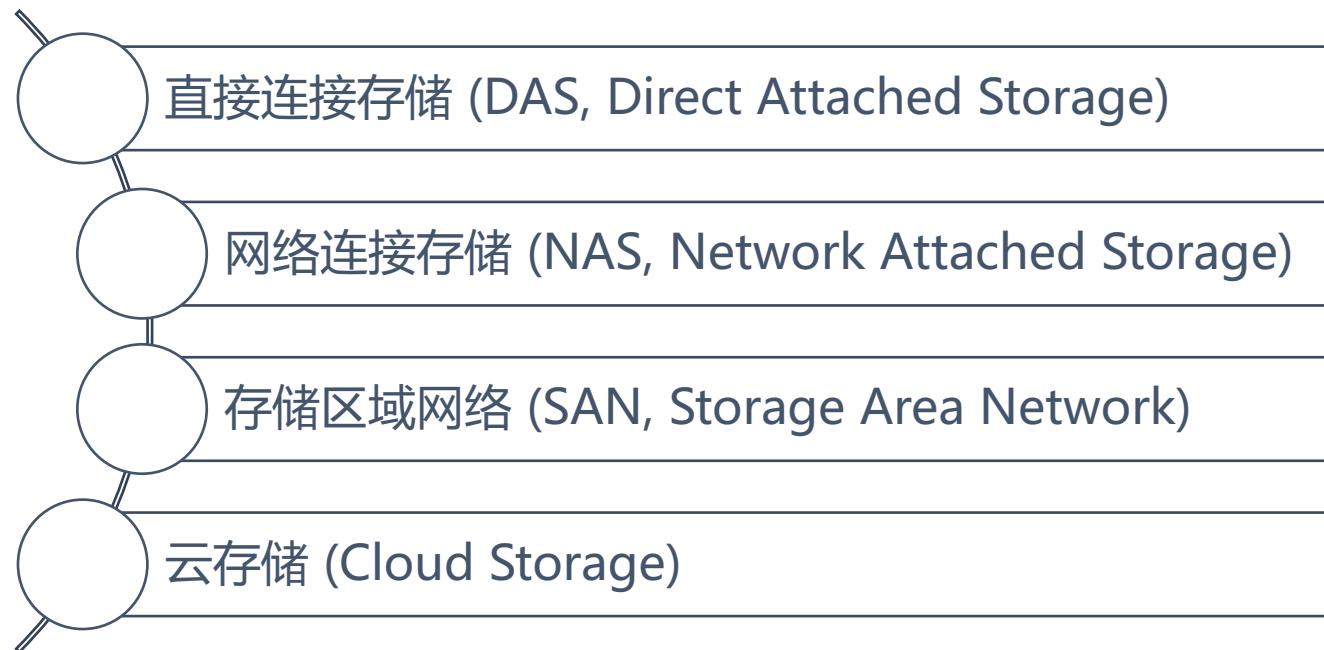
云数据存储

- ❖ 背景介绍
- ❖ 云数据存储的基本概念
- ❖ 分布式文件系统
- ❖ 常见的云存储服务

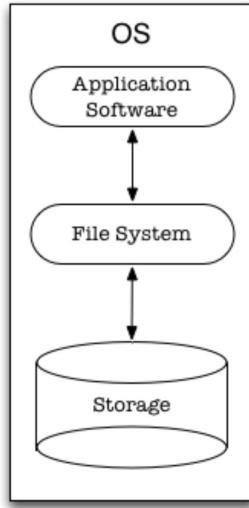
存储模型



存储模型 (Storage Model) 是一种描述计算机数据存储结构和管理方式的概念模型。常见的存储模型包括：



直接连接存储



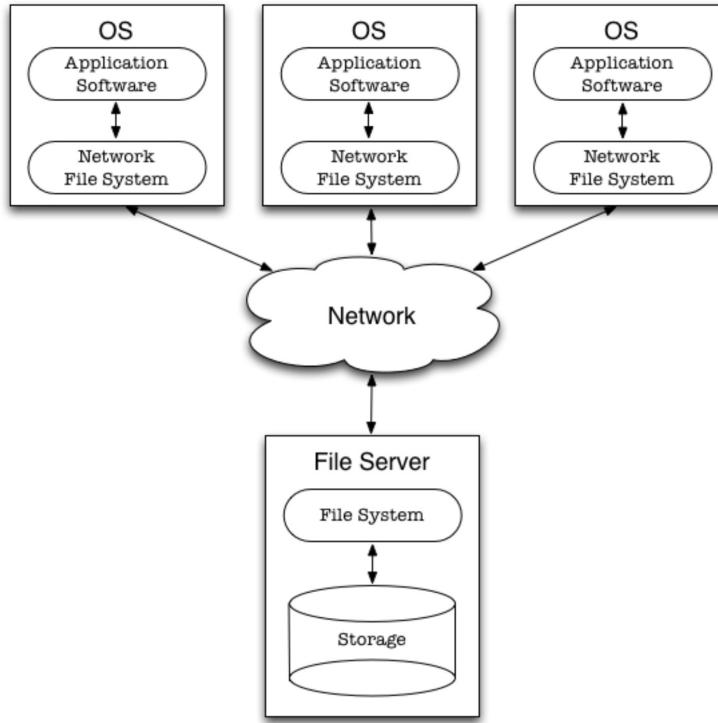
(a) Diagram



(b) A Maxtor IDE Drive

直接连接存储（ DAS ）：指的是直接连接到计算机或服务器的存储设备，如内部硬盘、外部硬盘或固态硬盘。

网络连接存储



网络连接存储（ NAS ）：是一种通过网络连接的独立存储设备，可以为多个计算机或设备提供文件级别的数据存储服务。

存储区域网络

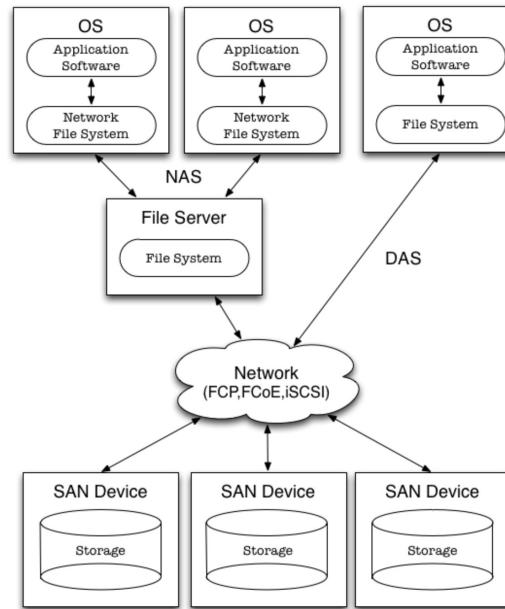
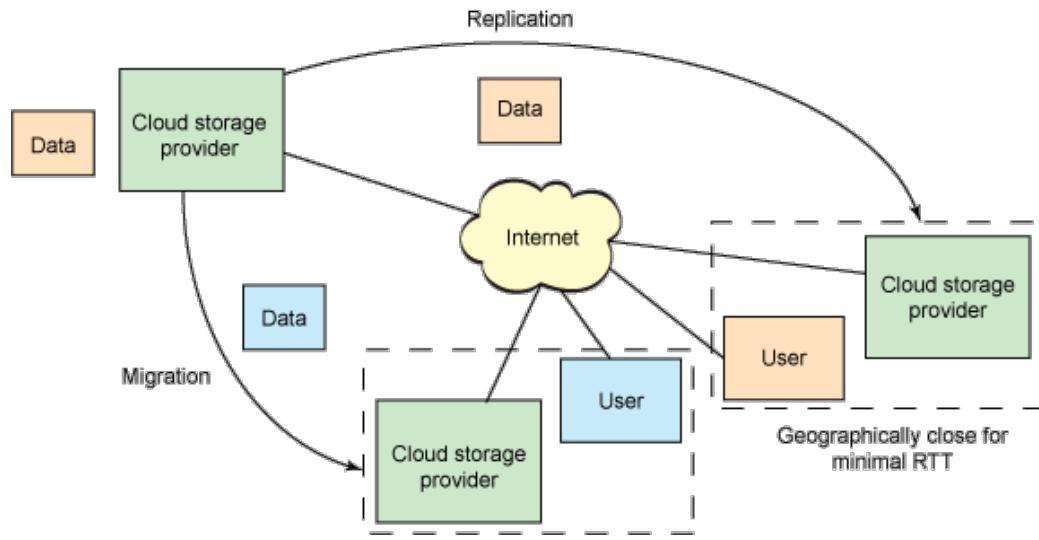


Figure 4.4: A SAN providing access to three devices; one host accesses parts of the available storage as if it was DAS, while a file server manages other parts as NAS for two clients.

存储区域网络（SAN）：一种专用的高速网络，提供了块级别的数据存储，具有高性能、高扩展性和高可靠性。然而，SAN的部署和管理通常更复杂，成本也较高。

云存储

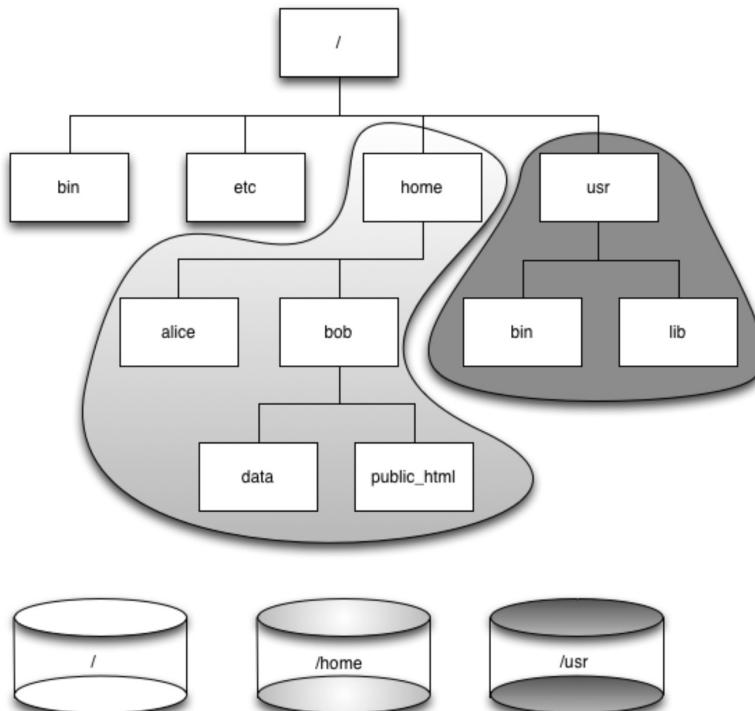


云存储 (Cloud Storage) : 一种将数据存储在远程服务器上并通过互联网访问的存储模式。只要有互联网连接，用户便可在任何地方存储、访问和共享数据。

文件系统



文件系统（File System）是一种用于组织、存储和检索计算机中文件的数据结构和管理方法。文件系统负责在存储设备上创建、删除和管理文件及目录。



Unix文件系统是一个树状结构，根位于/；不同的文件系统可以连接到不同的目录或装载点。本例中，/home 和/usr与/位于不同的磁盘上。



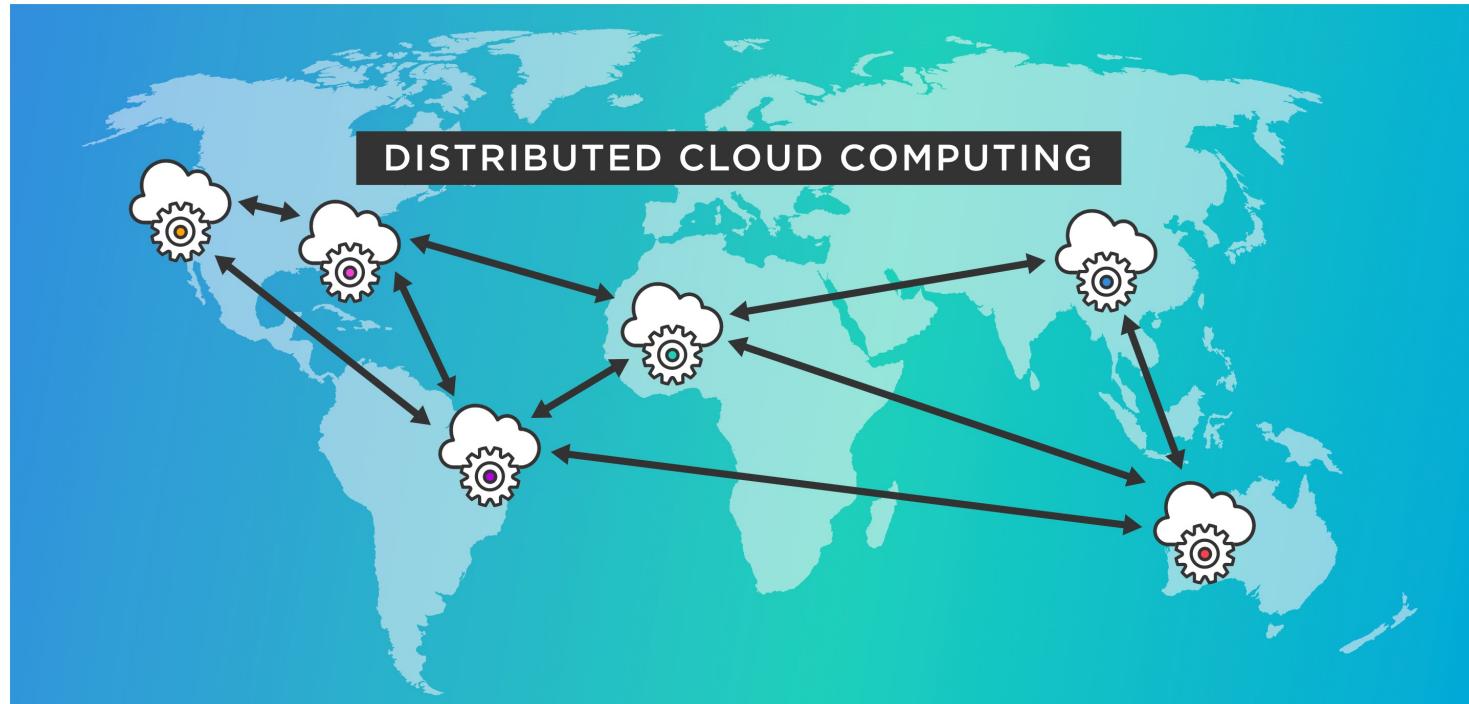
云数据存储

- ❖ 背景介绍
- ❖ 云数据存储的基本概念
- ❖ 分布式文件系统
- ❖ 常见的云存储服务

分布式文件系统



分布式文件系统（DFS, Distributed File System）**允许多台计算机通过网络共享文件和存储资源。**分布式文件系统将数据分散在多个物理位置，实现高度可扩展和容错能力。



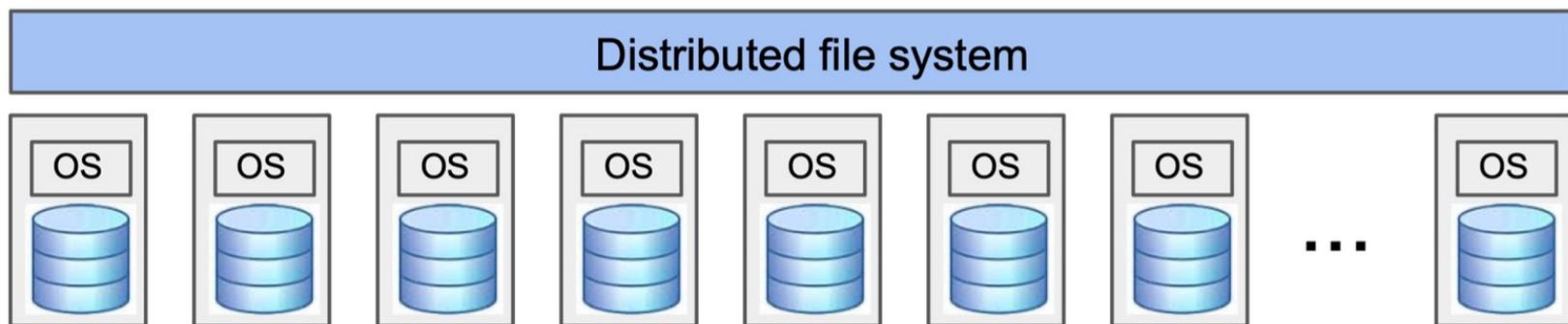
DFS的作用



通过网络共享文件：如果没有DFS，我们将只能通过电子邮件或使用Internet的FTP等应用程序交换文件

透明文件访问：访问远程文件不需要特殊的API，用户的程序可以**直接访问远程文件**，就像访问本地文件一样

简单的文件管理：管理DFS比管理多个本地文件系统更容易





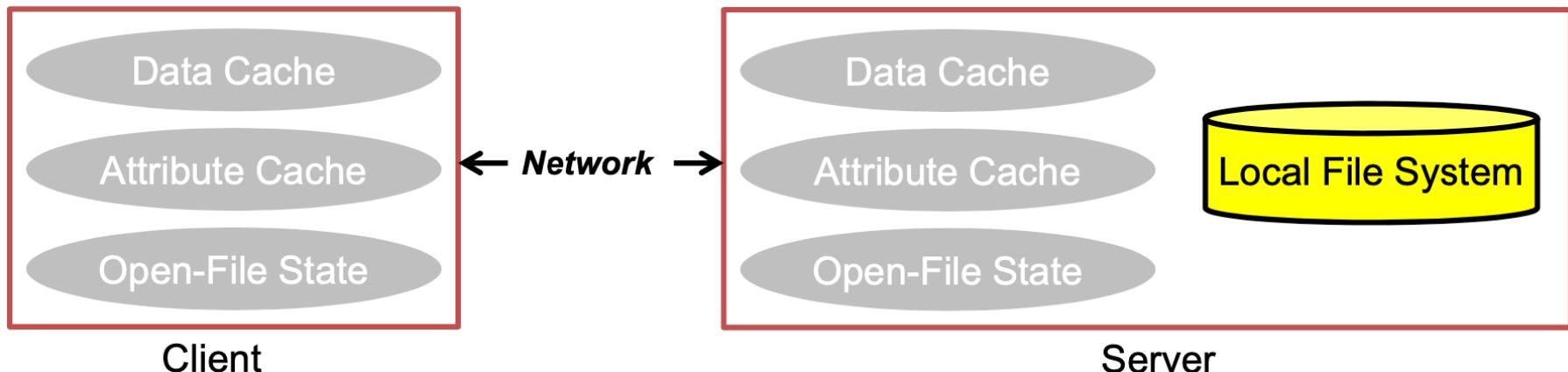
DFS的关键信息

- DFS所包含的信息通常可按如下分类
 - ✓ **The data state** : 文件的内容
 - ✓ **The attribute state (meta data)** : 文件的信息 (如文件的大小和访问控制列表)
 - ✓ **The open-file state** : 标识哪些文件已打开或正在使用 , 以及描述文件如何被锁定
- 设计DFS需要确定其各种信息的放置方式 :
 - ✓ 哪些信息放置在服务器端
 - ✓ 哪些信息放置在客户端

DFS关键信息的放置（1）



- ❖ Data state和meta data永久驻留在服务器的本地文件系统中
- ❖ 最近访问或修改的信息可能驻留在服务器和/或客户端缓存中
- ❖ Open-file state是暂时的，随着进程打开和关闭文件而改变



DFS关键信息的放置（2）



- 三个基本考虑问题
- **访问速度**：在客户端上缓存信息能否提升访问性能？
- **一致性**：如果客户端缓存了信息，各方是否共享相同的信息视图？
- **恢复**：如果一台或多台计算机崩溃，其他计算机会受到多大程度的影响？丢失了多少信息？



DFS几个重要层面

- 架构 (Architecture)
 - ✓ DFS通常是如何组织的 ?
 - 通信 (Communication)
 - ✓ DFS遵循何种通信模式 ?
 - ✓ DFS中的进程如何通信 ?
 - 命名 (Naming)
 - ✓ DFS中通常如何处理命名 ?
 - 同步 (Synchronization)
 - ✓ DFS采用何种文件共享语义 ?
 - 一致性和备份 (Consistency and Replication)
 - ✓ 客户端缓存和服务器端复制的各种功能是什么 ?
 - 容错 (Fault Tolerance)
 - ✓ DFS如何实现容错 ?
- 下节课内容



DFS几个重要层面

- 架构 (Architecture)
 - ✓ DFS通常是如何组织的 ?
- 通信 (Communication)
 - ✓ DFS遵循何种通信模式 ?
 - ✓ DFS中的进程如何通信 ?
- 命名 (Naming)
 - ✓ DFS中通常如何处理命名 ?

DFS的两种典型架构



客户机-服务器分布式文件
系统
Client-Server
Distributed File System

基于集群的分布式文件系
统
Cluster-based
Distributed File System

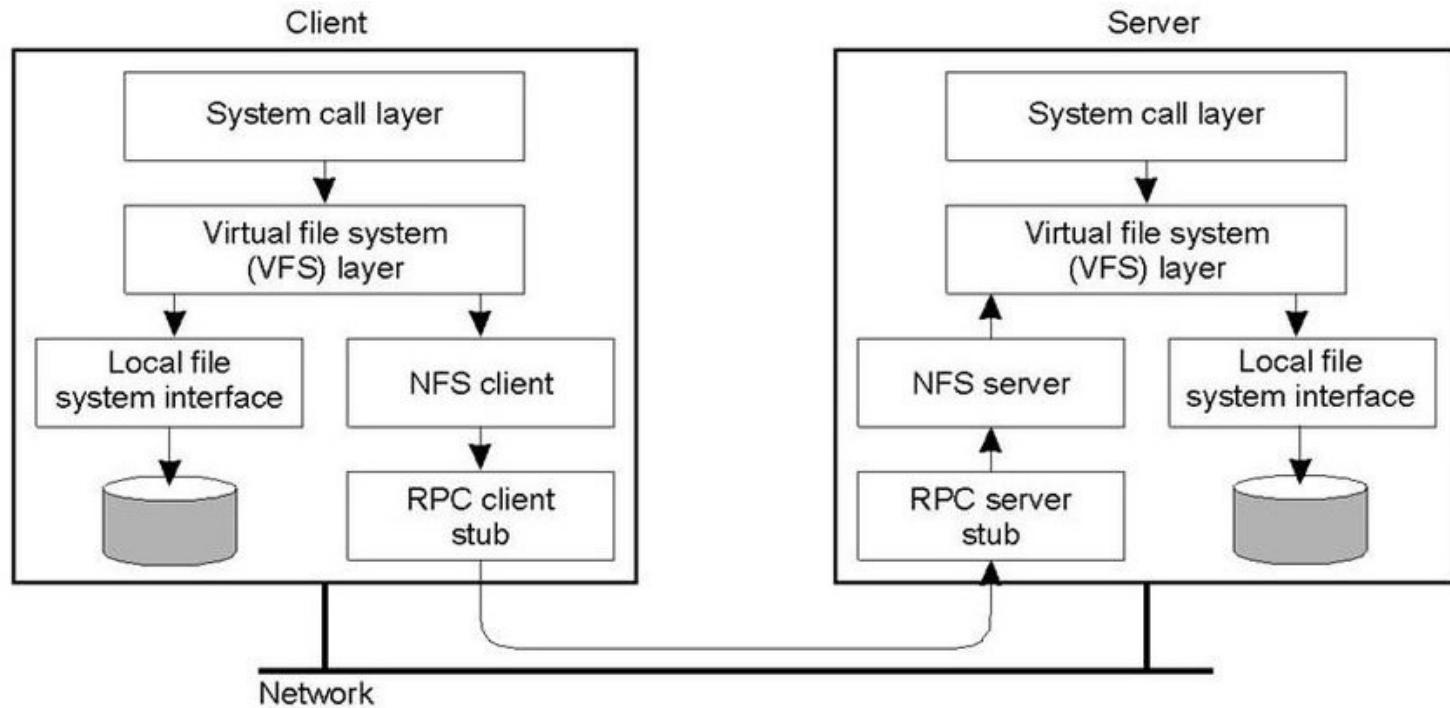


网络文件系统



- 许多分布式文件系统都是遵循“客户机-服务器”体系结构来组织的
- Sun微系统公司的网络文件系统（ Network File System ）就是基于 Unix 的系统最常用（也是最早）的分布式文件系统
- NFS附带了一个协议，该协议准确地描述了客户端如何访问存储在（远程）NFS文件服务器上的文件
- NFS允许不同操作系统和机器上的异构进程共享一个公共文件系统

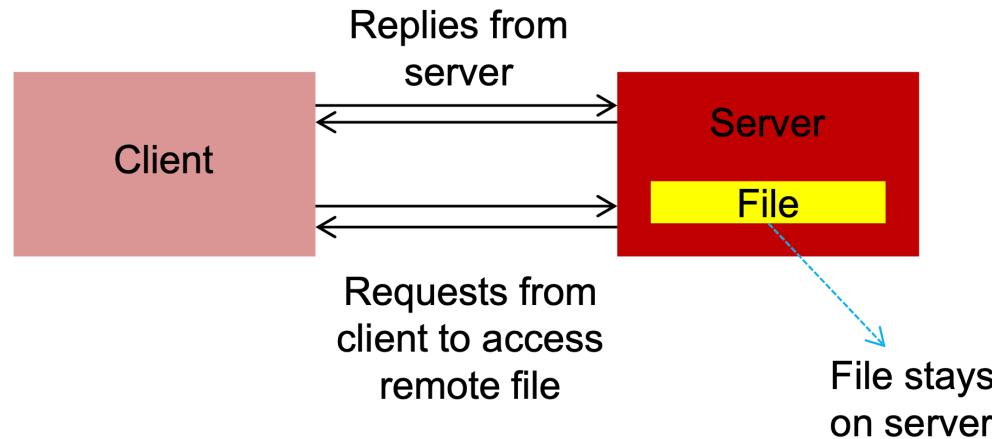
基本NFS架构



远程访问模型



- NFS和类似系统的底层模型是远程访问模型 (Remote Access Model)



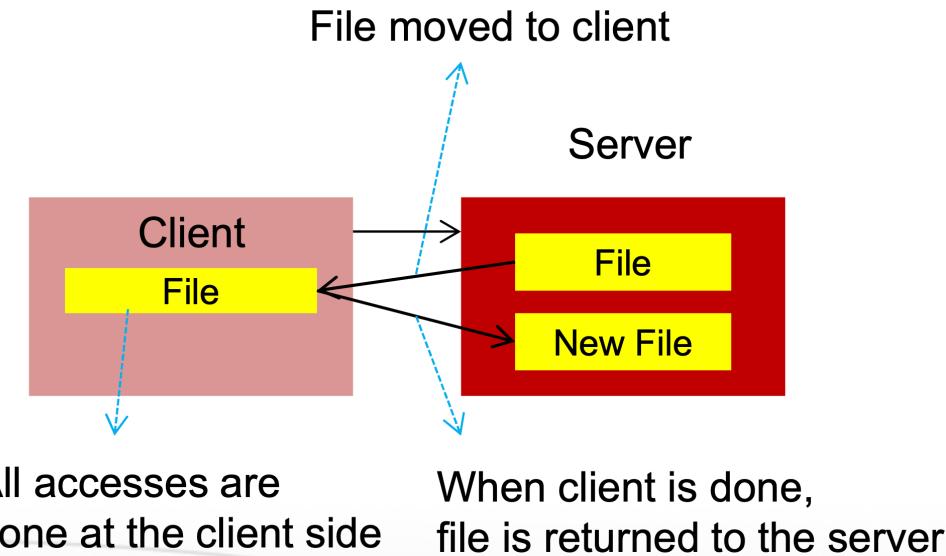
- 在这个模型中，客户机

- ✓ 可以透明地访问由远程服务器管理的文件系统
- ✓ 通常不知道文件的实际位置
- ✓ 调用文件系统的接口，类似于传统本地文件系统提供的接口



上传/下载模型

- 上传/下载模型 (Upload/Download Model) 则相反，允许客户端在从服务器下载文件后在本地访问该文件



- 当客户端下载、修改文件并将其放回时，可以通过这种方式使用 Internet 的 FTP 服务（文件传输协议，File Transfer Protocol）

DFS的两种典型架构



客户机-服务器分布式文件
系统
Client-Server
Distributed File System

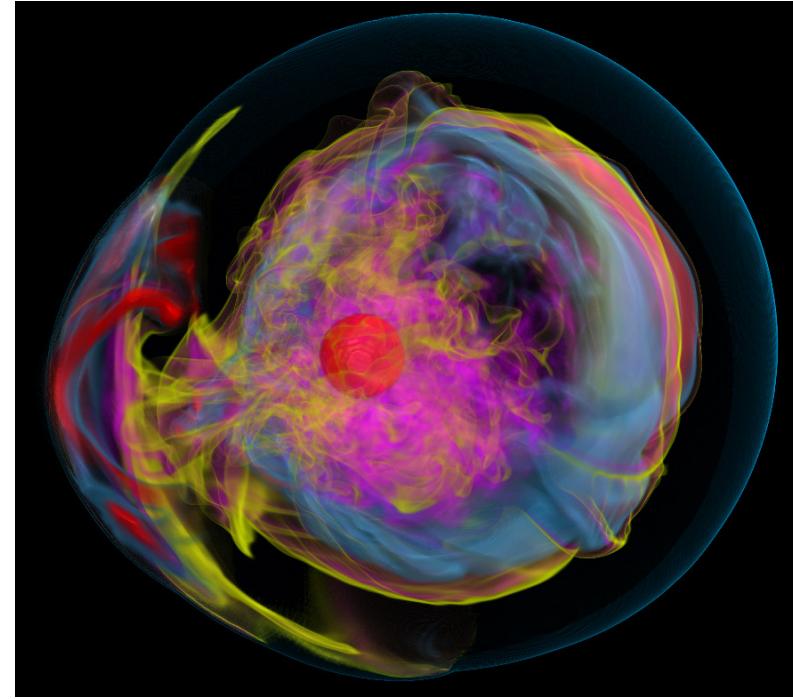
基于集群的分布式文件系
统
Cluster-based
Distributed File System



数据密集型应用



- 如今，大量数据密集型应用程序涌现
- 大多数数据密集型应用程序都属于两种类型的计算之一
 - 互联网服务（或云计算服务）
 - 高性能计算（HPC）
- 云计算和HPC应用程序通常在数千个计算节点上运行，并处理大数据



万亿级超新星的熵可视化

基于集群的分布式文件系统

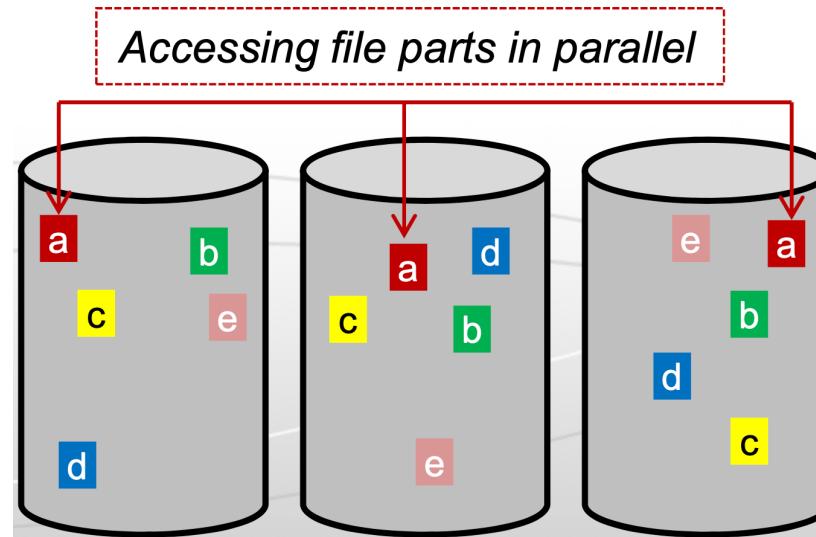


- 基于集群的分布式文件系统是支撑数据密集型应用的关键
- 为了使数据可以并发访问，集群式的文件系统使用文件条带化技术（file striping techniques）划分和分发大数据
- 集群式的文件系统可以是云计算或面向HPC的分布式文件系统
 - ✓ 云计算分布式文件系统：
 - Google File System (GFS)
 - AWS S3
 - ✓ 面向HPC的分布式文件系统：
 - Parallel Virtual File System (PVFS)
 - IBM的General Parallel File System (GPFS)

文件条带化技术



- 文件条带化是一种常用的算法，将单个文件分布在多个服务器上
- 因此，系统能并行获取单个文件的不同部分



文件条带化算法



- 如何在多台计算机上对文件进行条带化？
 - 循环分发 (Round-robin distribution)
 - 随机分发 (Random distribution)

循环分发 (1)



- 数据块以此写入服务器中

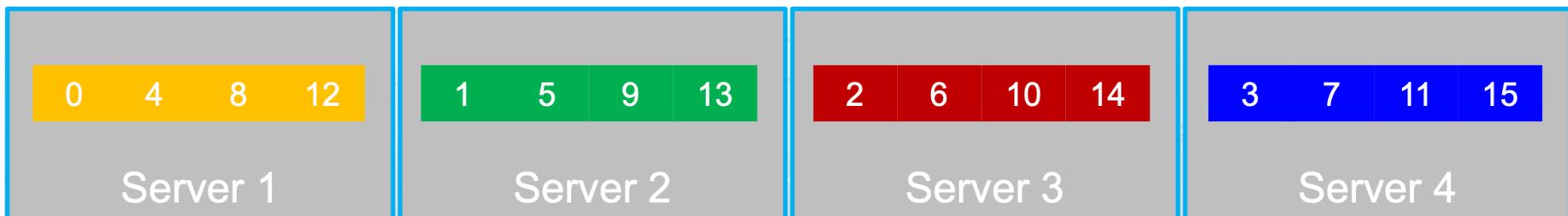
Logical File
→



Stripe Size



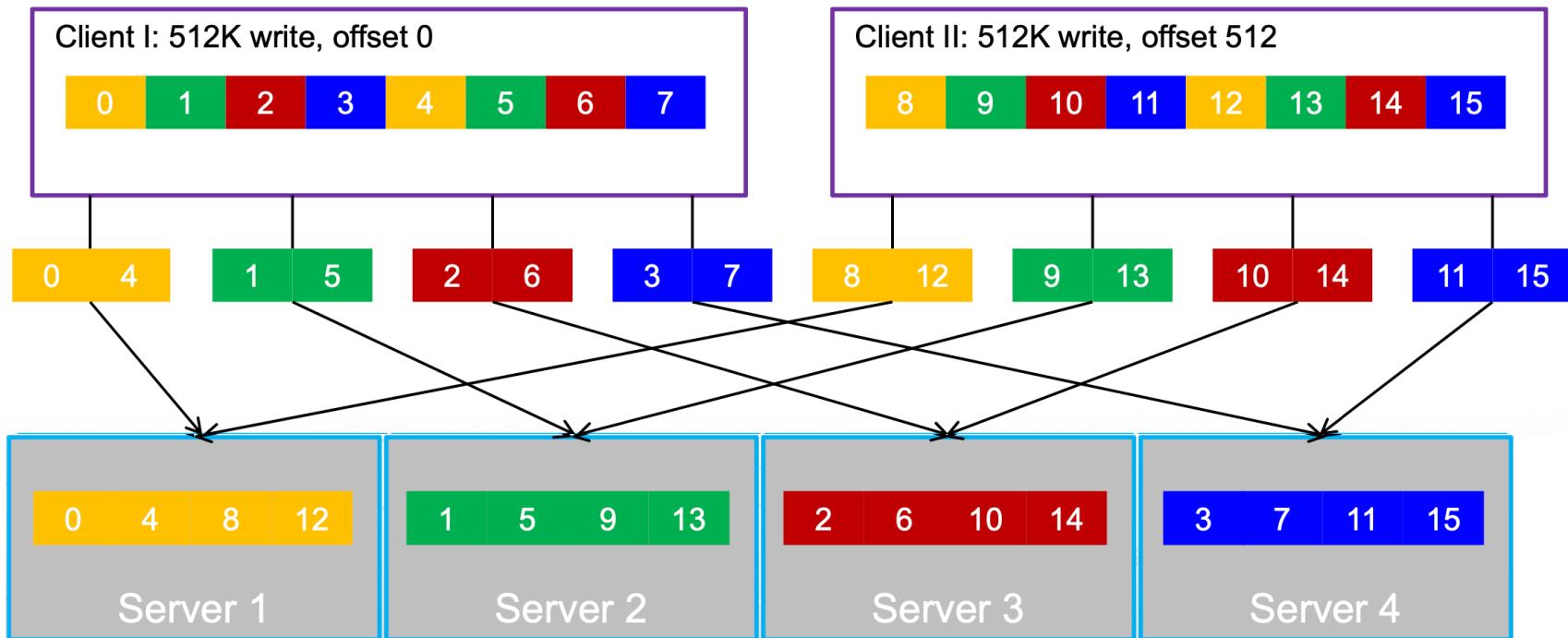
Striping Unit



循环分发 (2)



- 客户端在不同的区域执行文件的写入/读取



2D循环分发 (1)



- 当服务器数量很多时，读写数据需要访问大量服务器
 - ✓ 可以利用2D分发 (2D round-robin distribution)

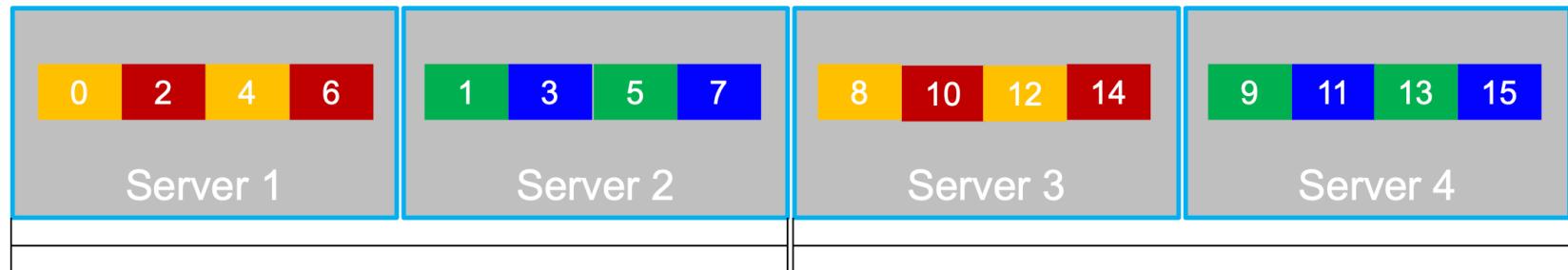
Logical File
→



Stripe Size



Striping Unit

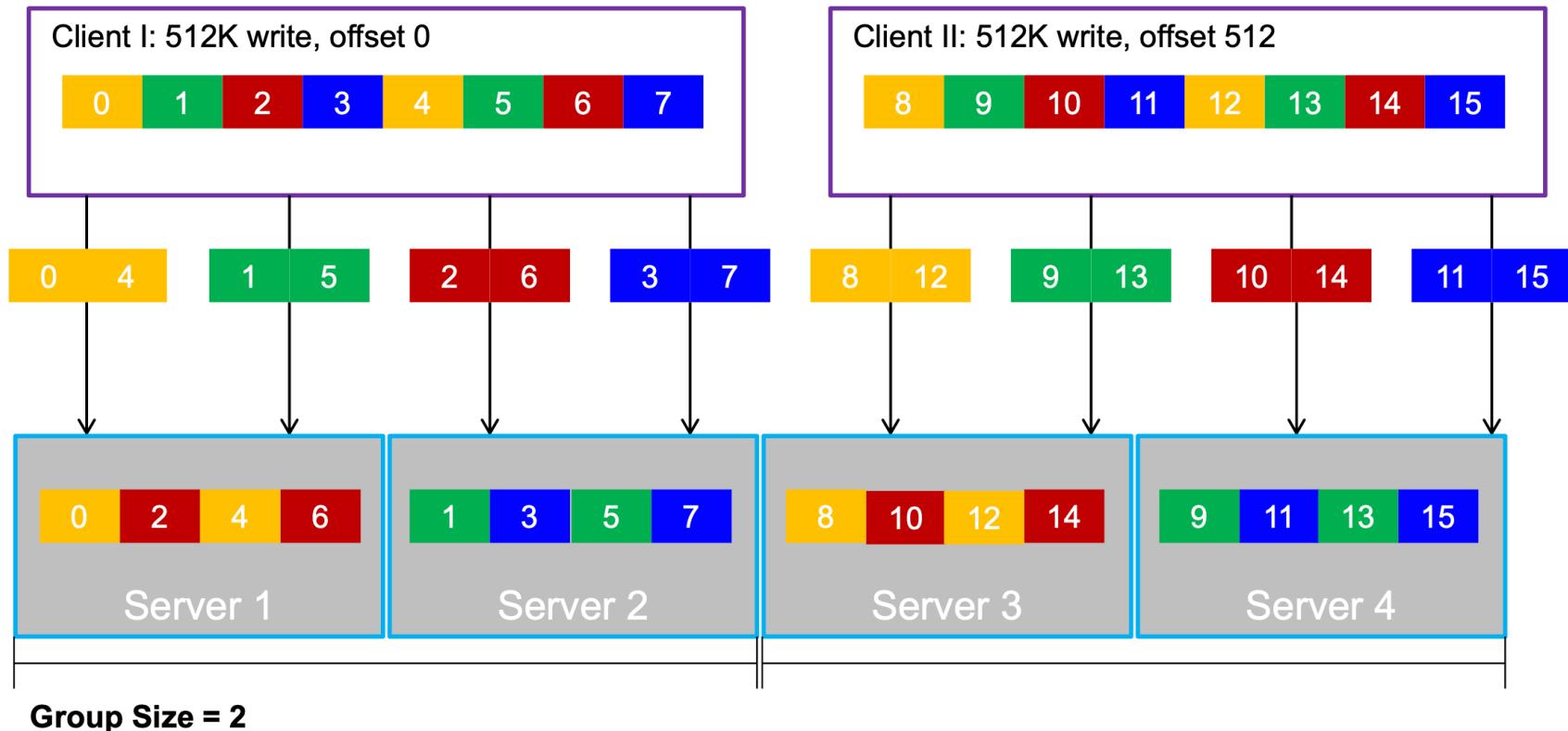


Group Size = 2

2D循环分发 (2)



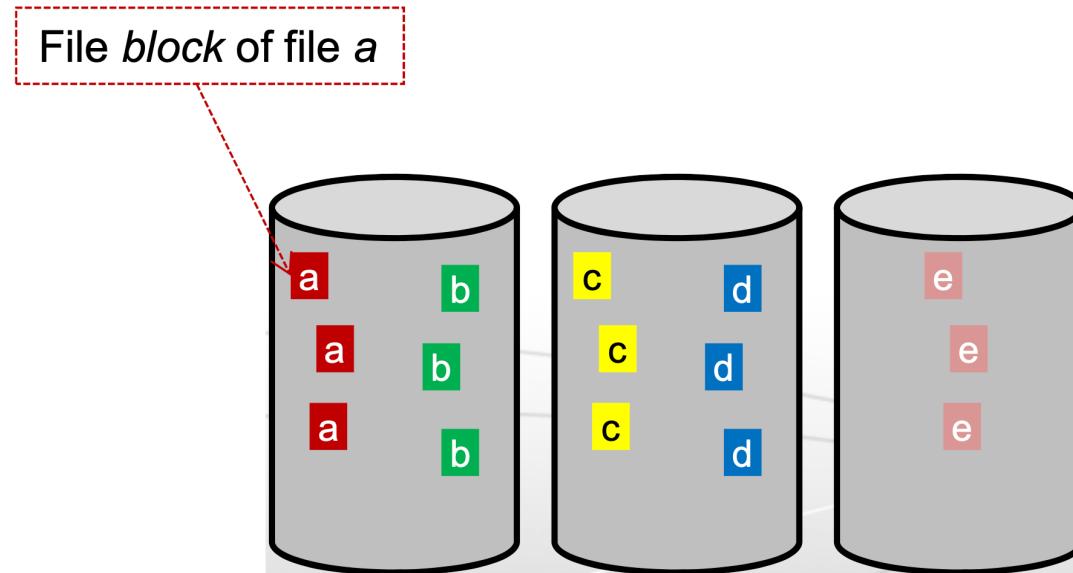
- 2D分发可以限制客户端需要访问的服务器数量



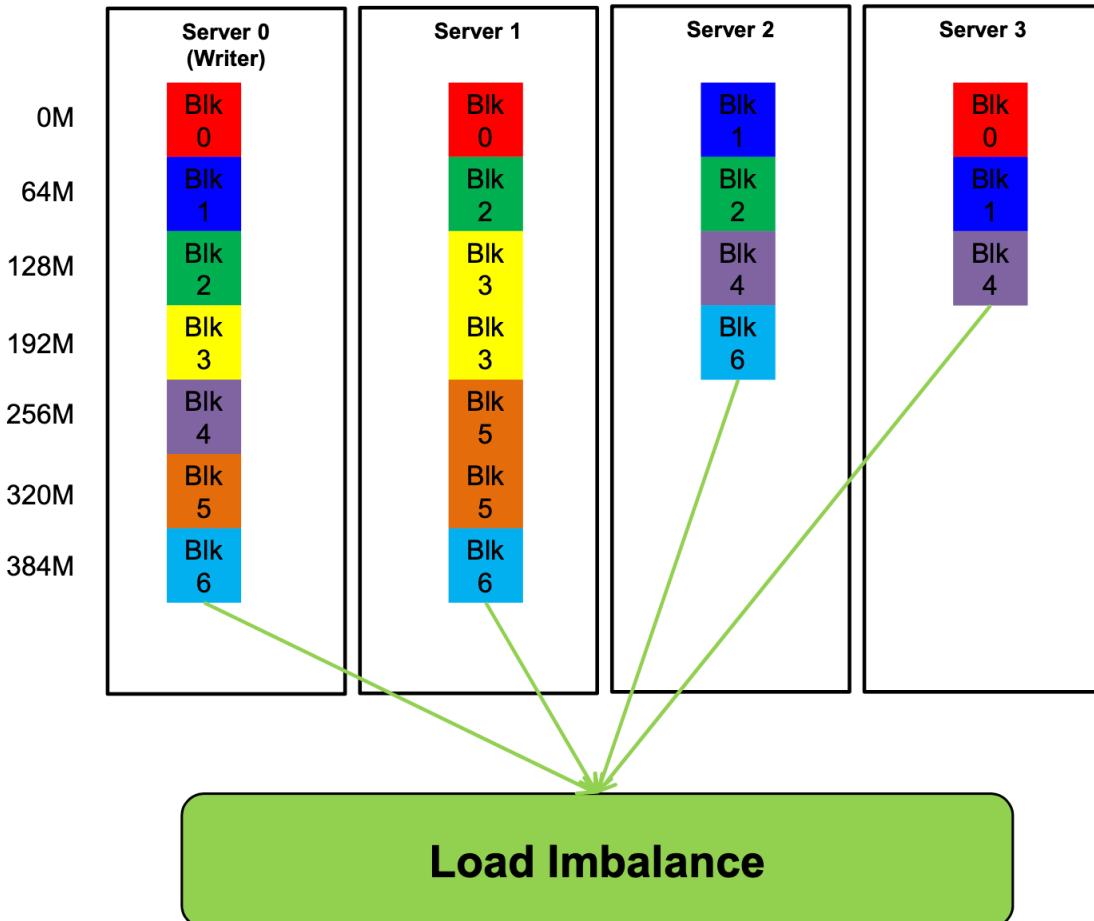
通用应用程序



- 对于**通用数据密集型应用**，或具有**不规则或许多不同类型数据的应用**，文件条带化算法可能无效（**为什么？**）
- 在这些情况下，将文件系统作为一个整体进行分区并简单地将文件存储在不同的服务器上通常更方便



随机分发



优点：平衡设备之间的负载

缺点：增加延迟和开销

- ✓ 需要访问多个服务器
- ✓ 访问模式不可预测，无法高效利用缓存
- ✓ 难以并行访问



DFS几个重要层面

- 架构 (Architecture)
 - ✓ DFS通常是如何组织的？
- 通信 (Communication)
 - ✓ DFS遵循何种通信模式？
 - ✓ DFS中的进程如何通信？
- 命名 (Naming)
 - ✓ DFS中通常如何处理命名？

通信 (Communication)



在分布式文件系统中，通信对于确保整个系统的**协调、数据访问和一致性**至关重要。

常用的通信范式：

- ❖ 远程过程调用 (Remote Procedure Call, RPC)
- ❖ 发布订阅模式 (Publish-Subscribe Pattern)
- ❖ 表征状态转移 (Representational State Transfer, REST)
- ❖ 消息传递 (Message Passing)



远程过程调用 (1)



```
blah, blah, blah
```

```
bar = add(i, j)
```

```
blah, blah, blah
```

```
int add(int x, int y) {  
    if (x>100)  
        return y-2;  
    else if (x>10)  
        return y-x;  
    else  
        return x+y;  
}
```

本地过程调用

```
blah, blah, blah  
bar = add(i, j)  
blah, blah, blah
```

Service 1

```
int add(int x, int y) {  
    if (x>100)  
        return y-2;  
    else if (x>10)  
        return y-x;  
    else  
        return x+y;  
}
```

Service 2

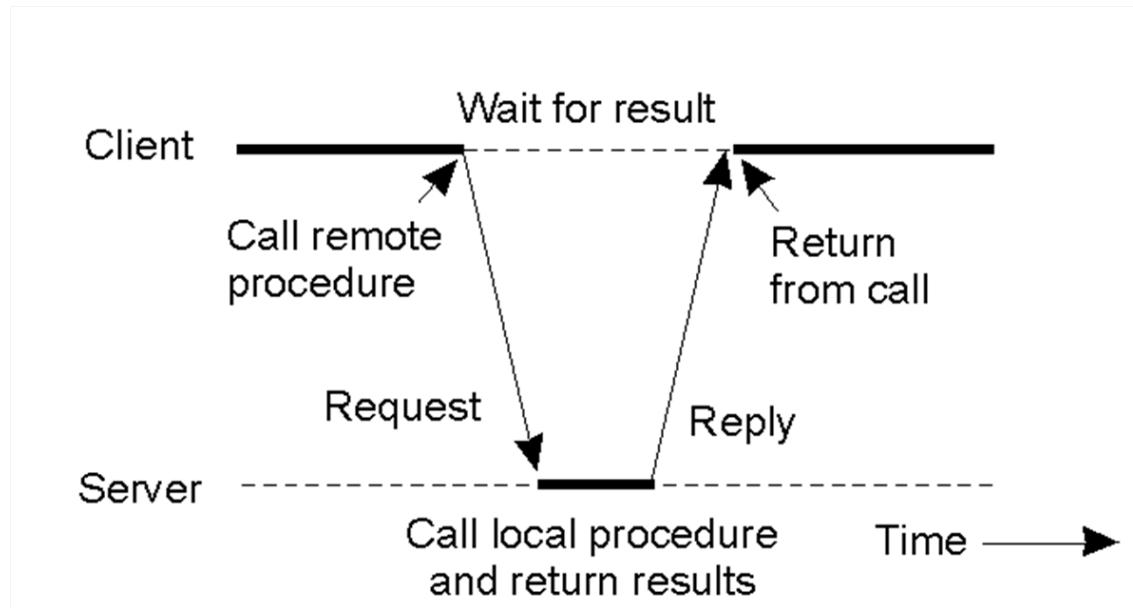
远程过程调用

为什么需要远程过程调用 ?

远程过程调用（1）



进程通过调用远程节点上的过程或函数进行通信，就好像它们是本地的一样



客户端和服务器之间的RPC通信过程

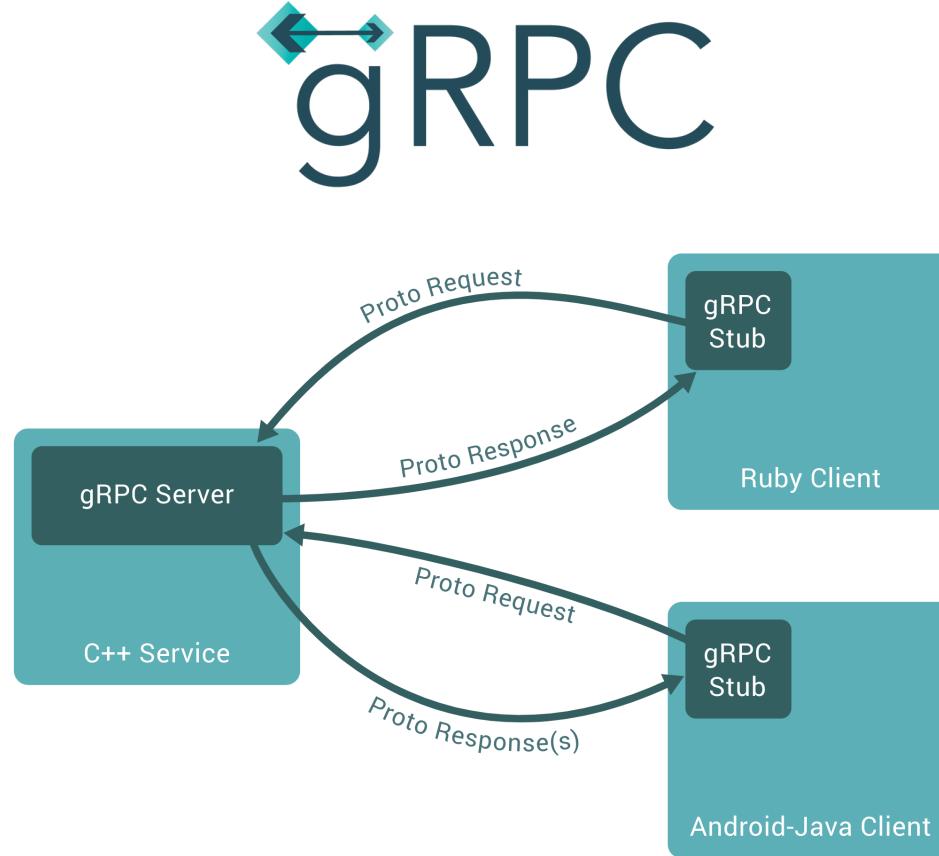
远程过程调用（1）



- 调用过程

1. 在客户端和服务器之间建立连接，通过TCP、UDP、HTTP等
2. 客户端应用告诉底层的RPC框架如何连接到服务器（如主机或IP地址）以及特定的端口，方法名称等
3. 客户端序列化参数成二进制的形式，称序列化（Serialize）或编组（marshal），通过寻址和传输将序列化的二进制发送给服务器
4. 服务器收到请求后，对参数进行反序列化，进行本地调用得到返回值
5. 返回值经过序列化发送给客户端，再反序列化后交给客户端上的应用

gRPC

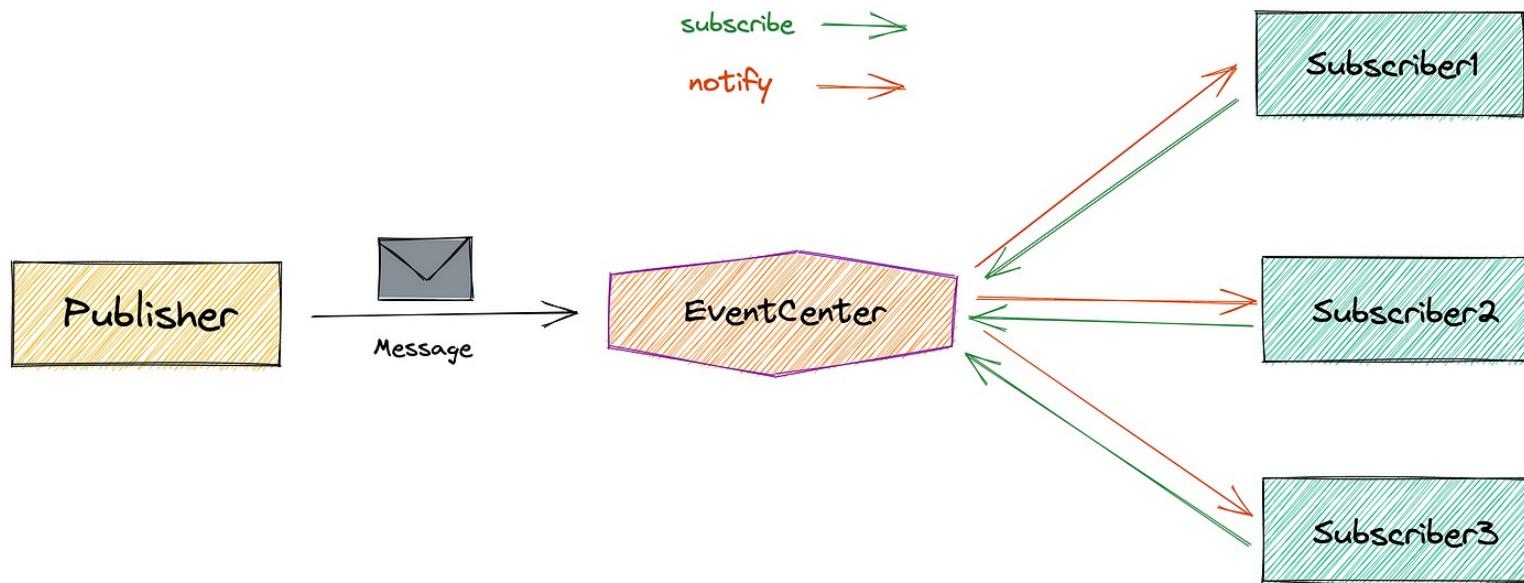


一个Hello实例：<https://grpc.io/docs/languages/python/quickstart/>

发布订阅模式 (1)



- 进程订阅特定主题或频道并在发布新信息时接收更新
- 对于需要通知多个节点系统中的更改（如文件更新或元数据更改）的情况很有用



发布订阅模式 (2)



- 典型的发布订阅系统



Apache Kafka



Rabbit MQ



Redis Pub/Sub

- 发布订阅模式典型应用



已关注 2628.7万



+ 关注



Subscribe

表征状态转移 (1)



- 资源与URL (Uniform Resource Identifier)

- ✓ 任何事物，只要有被引用到的必要，它就是一个资源

- 某用户的手机号码
 - 存储在云端的某个文件，如音乐、视频
 - 一个网站
 - 微服务之间的依赖关系
 - ...

- ✓ 资源需要有个唯一标识被识别，在Web中这个唯一标识就是URI

- <https://github.com/git>
 - https://zbchern.github.io/teaching/sse316/slides/sse316_lecture6.pdf

表征状态转移（2）



表征状态转移是一种**Web软件架构风格**，目的是便于不同软件/程序在网络（例如互联网）中互相传递信息。

REST设计的关键原则

- 无状态（stateless）：RESTful 服务应该是无状态的
- 客户端-服务器：REST 遵循客户端-服务器架构
- 可缓存：RESTful 服务应支持缓存，允许客户端存储响应以供重用
- 分层系统：RESTful 架构可以由多个层组成，每个层具有特定的职责
- 统一接口：RESTful 服务应遵循一致的统一接口

DFS几个重要层面

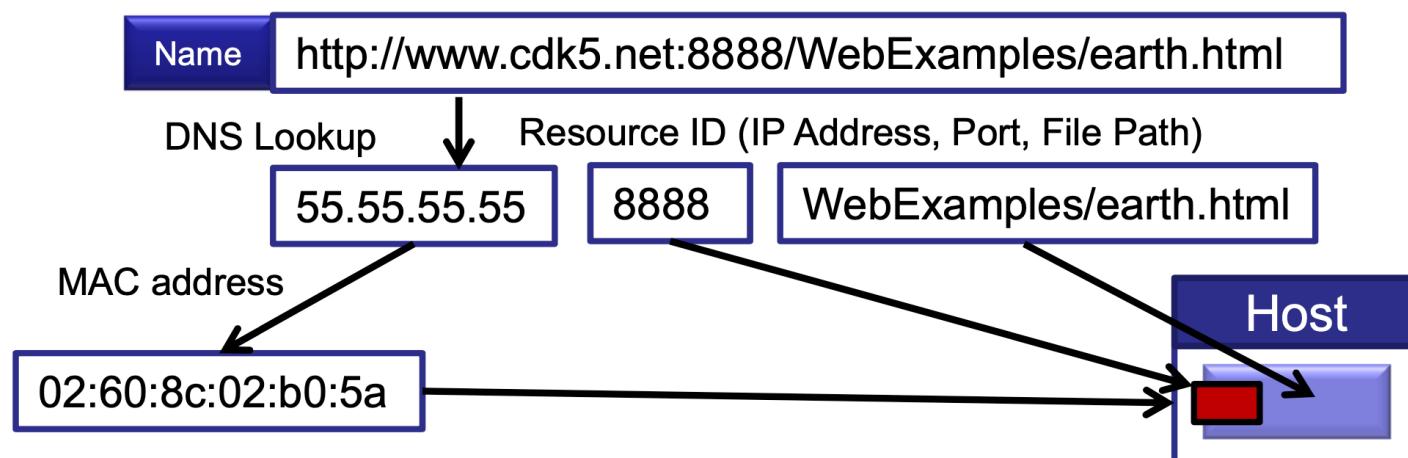


- 架构 (Architecture)
 - ✓ DFS通常是如何组织的？
- 通信 (Communication)
 - ✓ DFS遵循何种通信模式？
 - ✓ DFS中的进程如何通信？
- 命名 (Naming)
 - ✓ DFS中通常如何处理命名？

命名 (Naming)



- 名称用于唯一标识分布式系统中的实体
 - 实体可以是进程、远程对象、新闻组等
- 使用名称解析将名称映射到实体的位置
- 名称解析示例



分布式文件系统中的命名



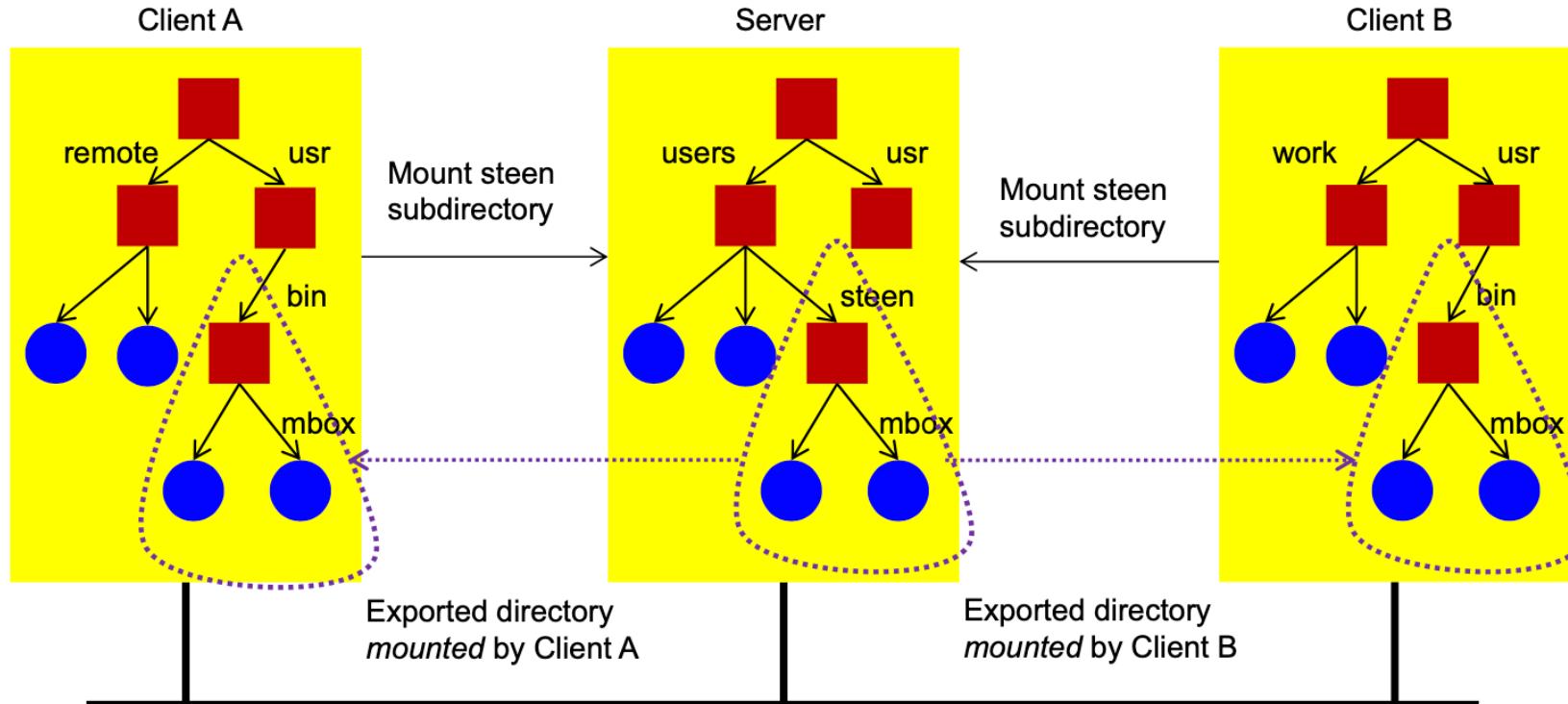
NFS是DFS中如何处理命名的代表

NFS中的命名 (Naming)



- NFS命名模型的基本思想是为客户端提供完全的透明
- NFS中的透明度是通过允许客户端将远程文件系统挂载到自己的本地文件系统中来实现的
- 但是，NFS不允许挂载整个文件系统，而是允许客户端仅装载文件系统的一部分

NFS中的挂载 (Mounting)



The file named `/usr/bin/mbox`
at Client A

Sharing files resolved

The file named `/usr/bin/mbox`
at Client B



全球最大搜索引擎、Google Maps、Google Earth、Gmail、YouTube等。这些应用的共性在于数据量巨大，且要面向全球用户提供实时服务。



The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Google*

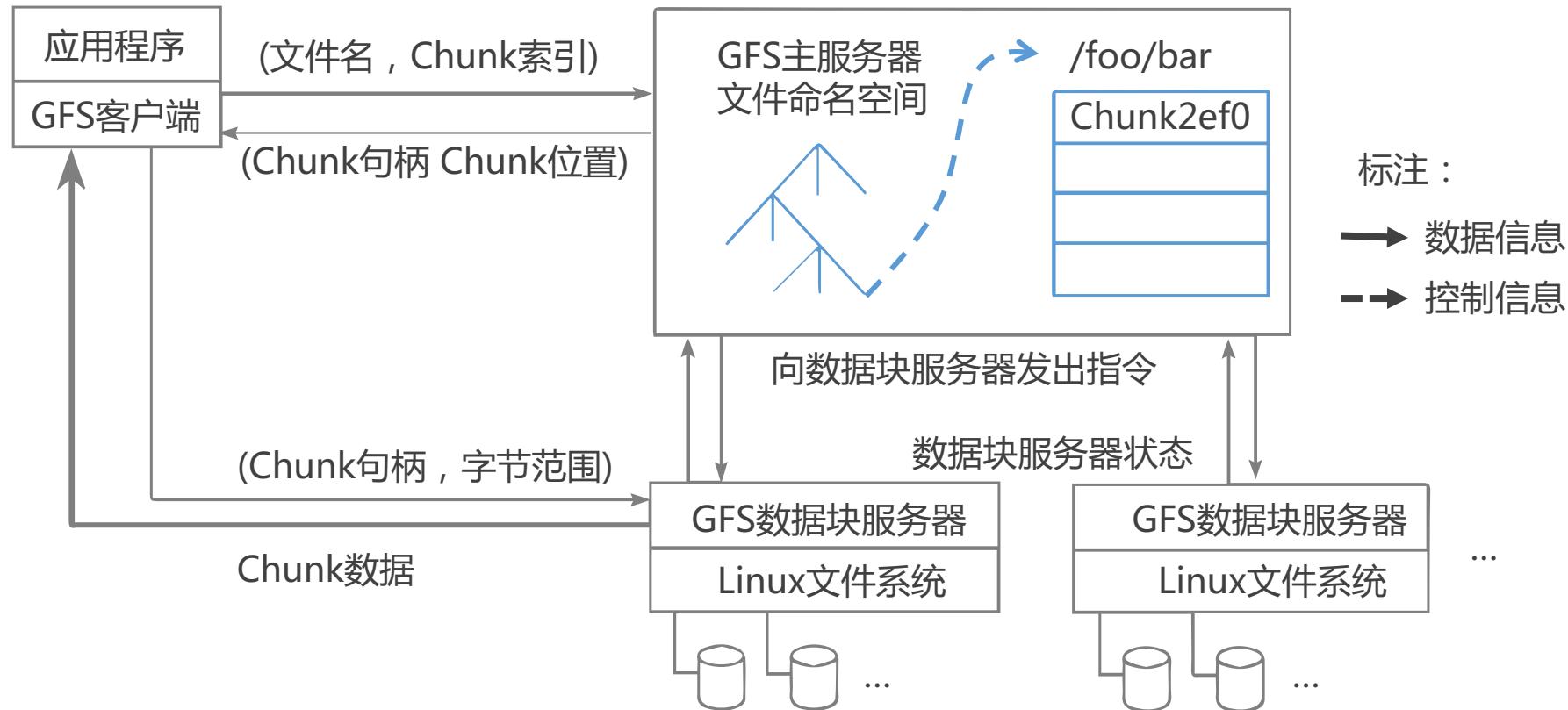
SOSP 2003

GFS数据分发策略



- 谷歌文件系统（GFS）是一种**基于云计算的可扩展DFS**，适用于大型分布式数据密集型应用程序
- GFS将大文件分为多个称为块或块的部分（默认为64MB），并将它们存储在不同的数据服务器上
- 每个GFS块具有唯一的64位标识符，并作为文件存储在数据服务器上的较低层本地文件系统中
- GFS使用**随机分发策略**在集群数据服务器之间分发数据块

GFS系统架构





GFS系统架构

- GFS将整个系统节点分成三类角色





GFS的实现机制

- 客户端首先访问Master节点，获取交互的Chunk Server信息，然后访问这些Chunk Server，完成数据存取工作。这种设计方法实现了控制流和数据流的分离。
- Client与Master之间只有控制流，而无数据流，极大地降低了Master的负载。
- Client与Chunk Server之间直接传输数据流，同时由于文件被分成多个Chunk进行分布式存储，Client可以同时访问多个Chunk Server，从而使得整个系统的I/O高度并行，系统整体性能得到提高。



1 采用中心服务器模式

- 可以方便地增加Chunk Server
- Master掌握系统内所有Chunk Server的情况，方便进行负载均衡
- 不存在元数据的一致性问题



2 不缓存数据

- 文件操作大部分是流式读写，不存在大量重复读写，使用 Cache 对性能提高不大
- Chunk Server 上数据存取使用本地文件系统从可行性看，Cache 与实际数据的一致性维护也极其复杂



3 在用户态下实现

- 利用POSIX编程接口存取数据降低了实现难度，提高通用性
- 用户态下有多种调试工具
- Master和Chunk Server都以进程方式运行，单个进程不影响整个操作系统
- GFS和操作系统运行在不同的空间，两者耦合性降低



云数据存储

- ❖ 背景介绍
- ❖ 云数据存储的基本概念
- ❖ 分布式文件系统
- ❖ 常见的云存储服务

常见的云存储服务



对象存储
Object Storage

块存储
Block Storage

文件存储
File Storage

对象存储



一种可扩展、高可用的存储方案，专为**存储非结构化数据**（如文档、图像和视频）而设计。在对象存储中，数据以具有唯一标识符和元数据的对象形式存储，以简化分布式系统中数据的管理和检索。



- ✓ 高度可扩展和分布式架构
- ✓ 适用于存储大量非结构化数据
- ✓ 可通过API或Web接口访问
- ✓ 存储和检索大量数据的成本效益高



Amazon S3



Google Cloud Storage

Microsoft Azure
Blob Storage



块存储



块存储旨在以**固定大小的块存储结构化数据**，通常用于数据库、虚拟机或其他需要低延迟和高性能存储的应用程序。块存储设备充当独立的磁盘驱动器，允许在块级别进行随机读写操作。



- ✓ 低延迟和高性能存储解决方案
- ✓ 适用于数据库、虚拟机和其他结构化数据存储
- ✓ 支持块级操作，如随机读写



Amazon **EBS** (Elastic Block Store)



GCP Persistent Disk

文件存储



文件存储服务提供了用于**存储和组织文件的分层结构**，类似于传统的文件系统。它专为需要多个客户端访问共享文件系统的应用程序设计，如内容管理系统、数据分析平台和备份系统。



- ✓ 熟悉的分层文件系统结构
- ✓ 对于需要共享访问的应用程序易于使用和管理
- ✓ 支持标准文件系统协议，如NFS



Amazon Elastic File System (Amazon EFS)



Google
Cloud Filestore



Azure Files



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

谢谢

陈壮彬
软件工程学院

<https://zbchern.github.io/sse316.html>