火龙果•整理
uml.org.cn

101－5
GlusterFS系统原理剖析

火龙果•整理
uml.org.cn

培训提纲

- GlusterFS架构特点
- GlusterFS核心工作原理
- GlusterFS典型功能剖析

火龙果•整理
uml.org.cn

GlusterFS是什么？

集群
NAS

无中心
架构

分布式
文件系
统

标准
硬件

高性能

POSIX

自修复

Scale-
Out

自动
复制

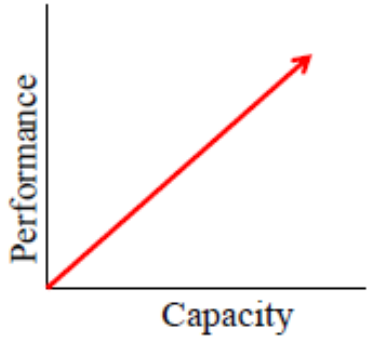
全局统
一命名
空间

Infin
iband

高可用

火龙果•整理
uml.org.cn

GlusterFS架构设计目标



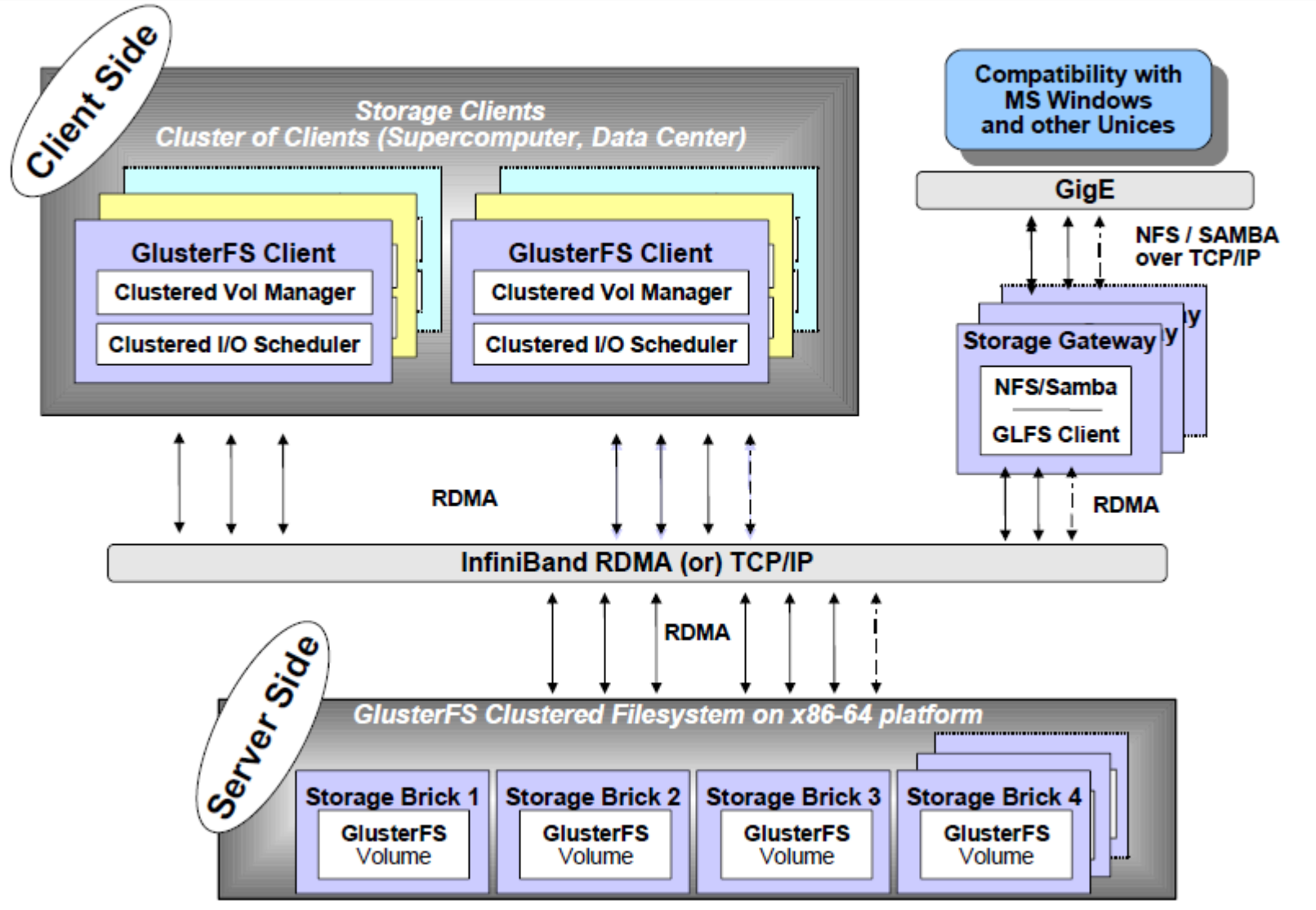
- **Elasticity**
  - Flexibility adapt to growth/reduction
  - Add, delete volumes & users
  - Without disruption
- **Scale linearly**
  - Multiple dimensions
    - Performance
    - Capacity
  - Aggregated resources
- **Eliminate metadata**
  - Improve file access speed
- **Simplicity**
  - Ease of management
  - No complex Kernel patches
  - Run in user space

火龙果•整理
uml.org.cn

GlusterFS架构特点

- 软件定义
- 无中心架构
- 全局命名空间
- 高性能
- 用户空间实现
- 堆栈式设计
- 弹性横向扩展
- 高速网络通信
- 数据自动修复

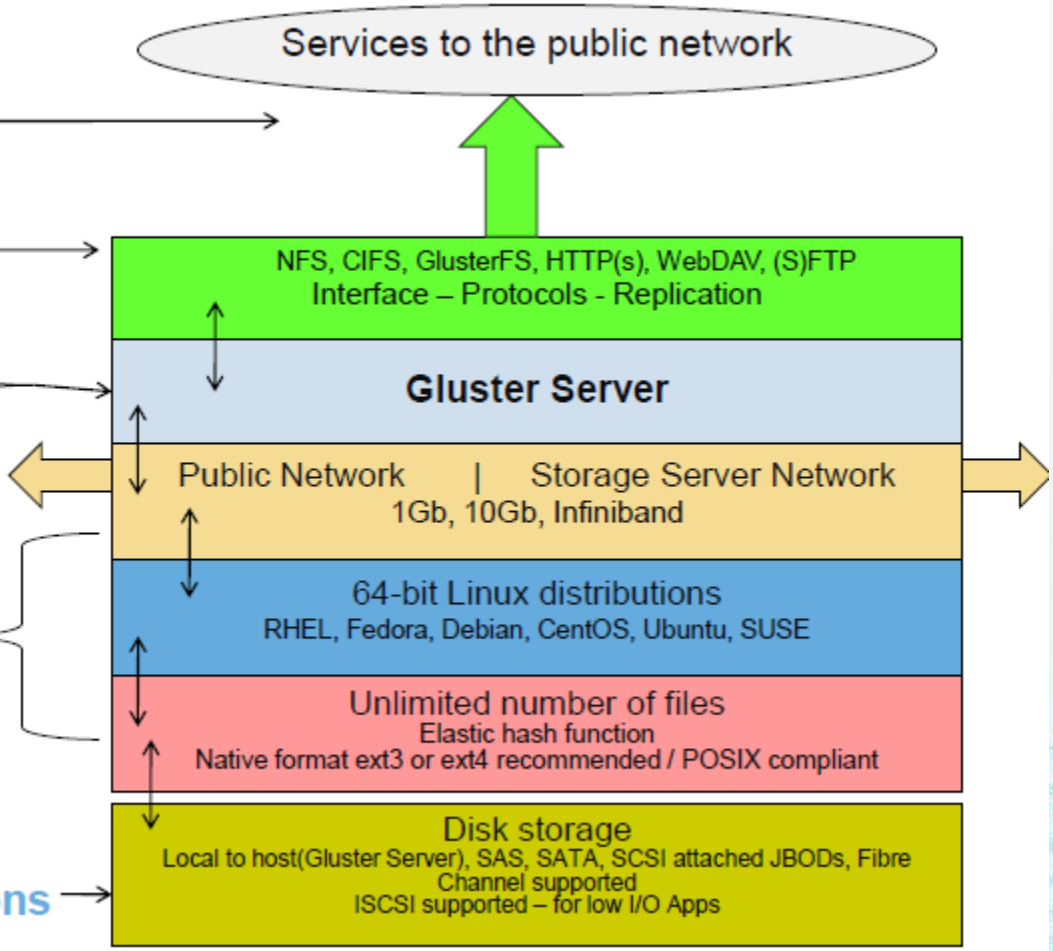
GlusterFS总体架构



火龙果•整理
uml.org.cn

模块化/堆栈式存储OS架构

- NAS protocols
- Replication
- Global namespace
- Clustered & replicated
- Built-in failover
- Self healing
- POSIX compliant
- No metadata server
- Elastic hash
- Thin provisioning
- Flexible storage configurations



火龙果•整理
uml.org.cn

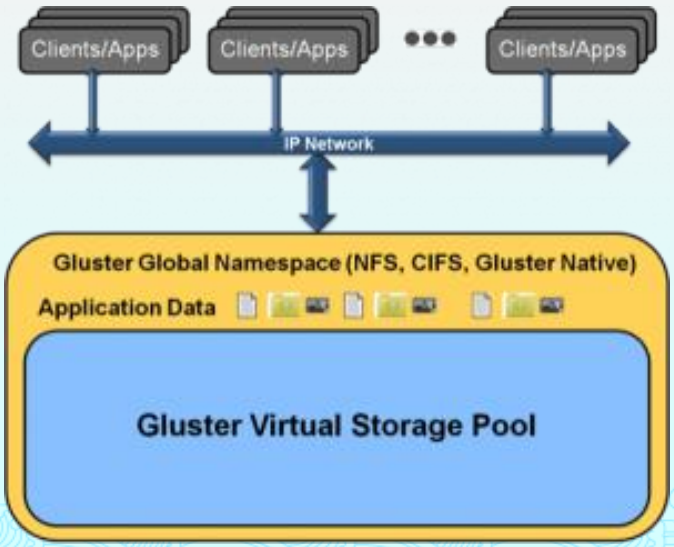
全局统一命名空间

通过分布式文件系统将物理分散的存储资源虚拟化成统一的存储池

文件系统

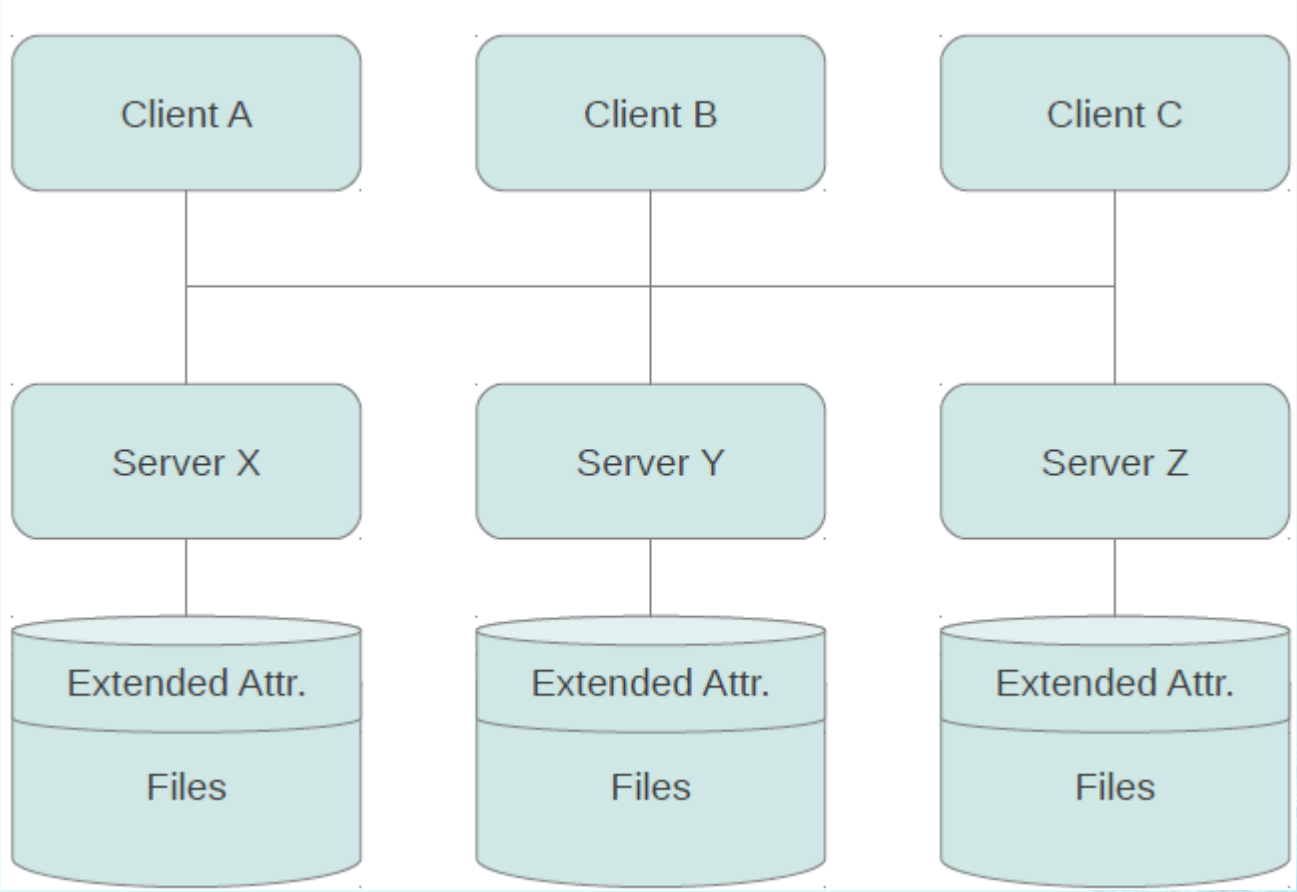
GlusterFS

RAID



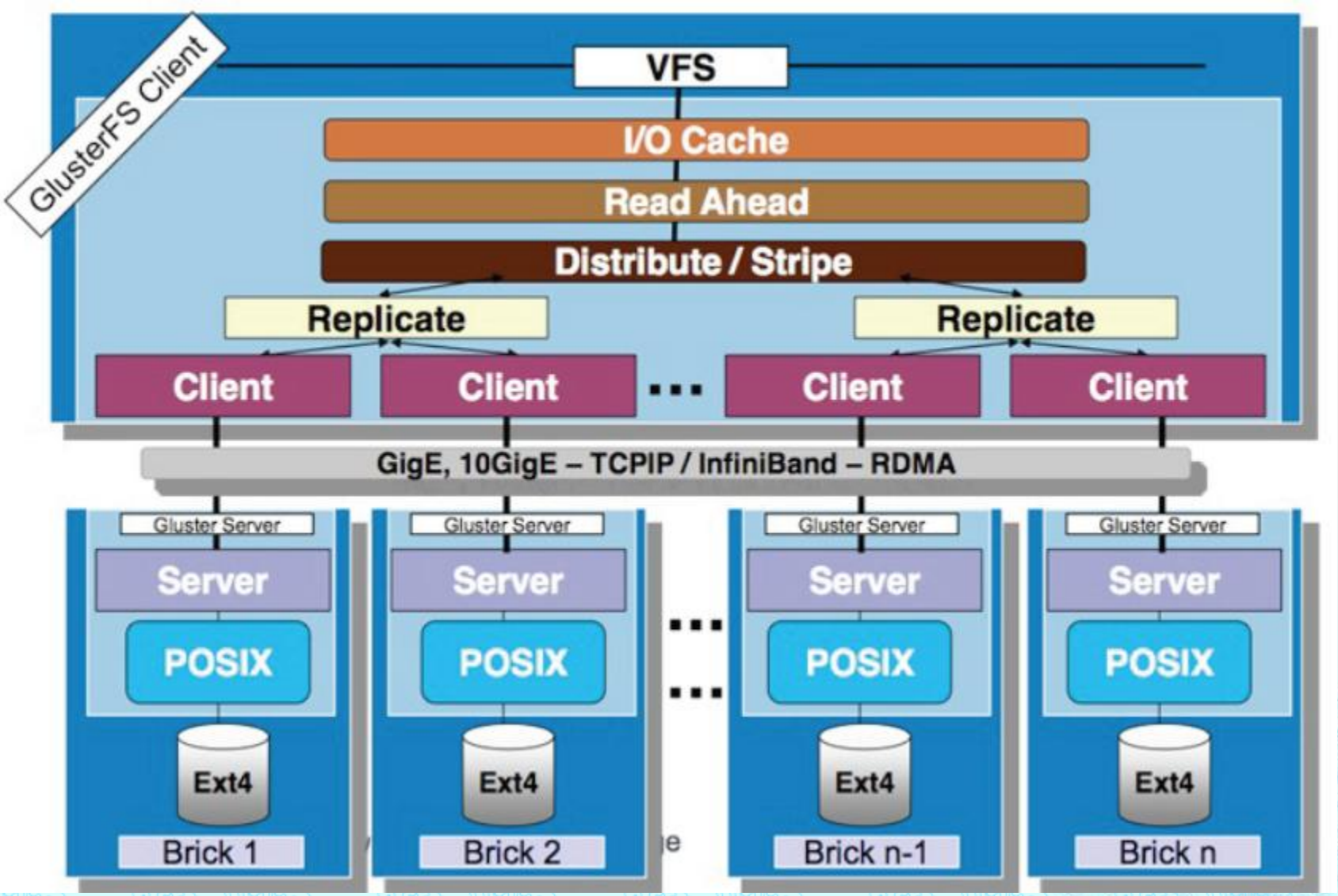
火龙果•整理
uml.org.cn

无集中元数据服务



GlusterFS堆栈式软件架构

火龙果•整理
uml.org.cn



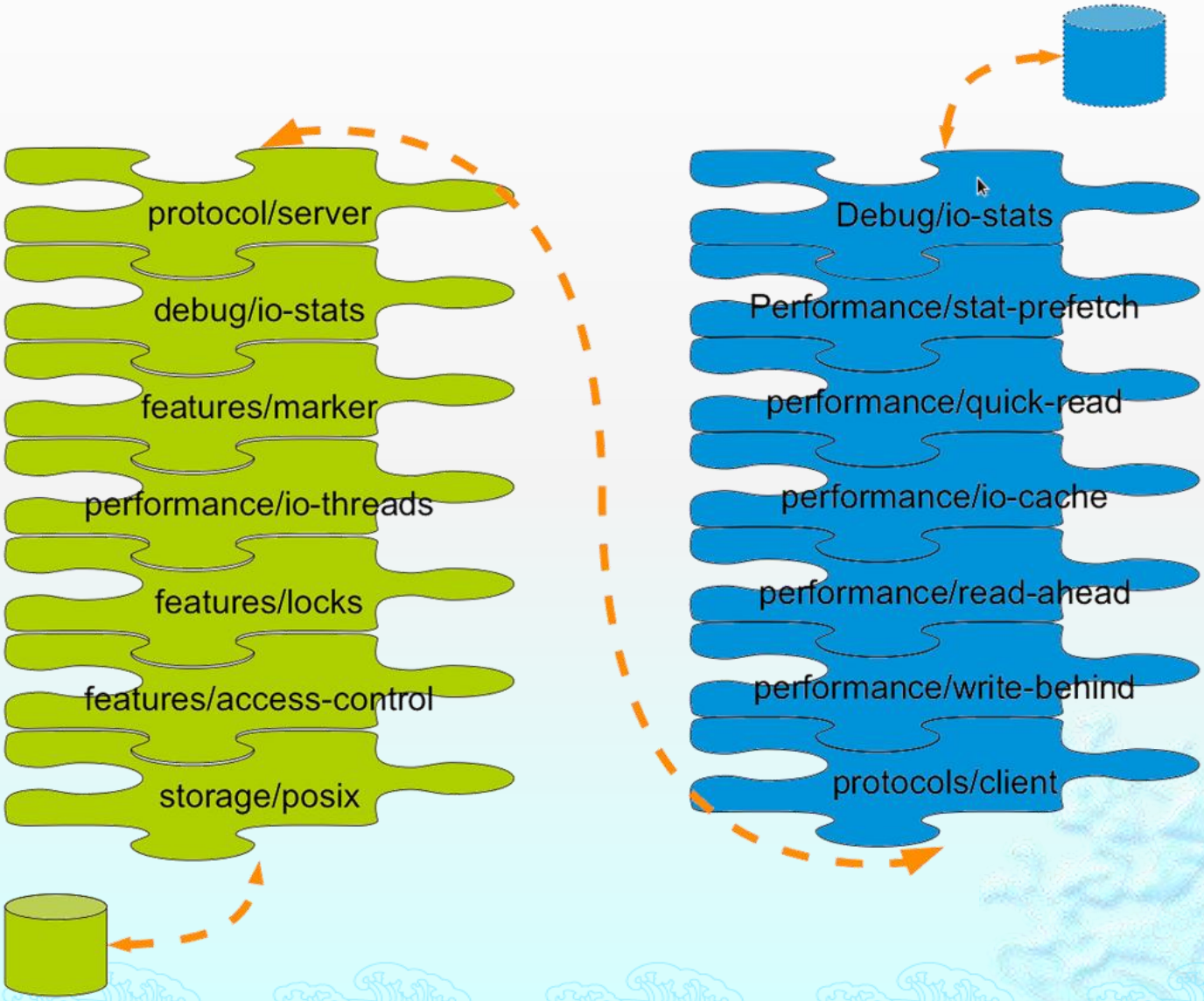
火龙果•整理
uml.org.cn

GlusterFS基本概念

- Brick
  - A filesystem mountpoint
  - A unit of storage used as a GlusterFS building block
- Translator
  - Logic between the bits and the Global Namespace
  - Layered to provide GlusterFS functionality
- Volume
  - Bricks combined and passed through translators
- Node / Peer
  - Server running the gluster daemon and sharing volumes

火龙果•整理
uml.org.cn

Translators

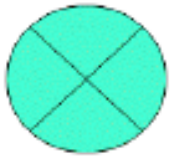


火龙果•整理
uml.org.cn

弹性hash算法

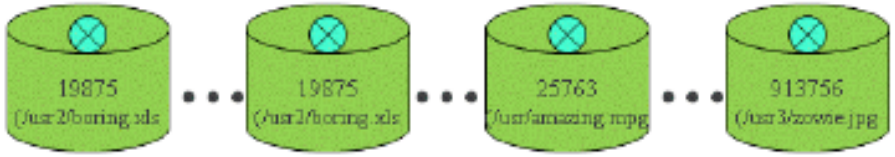
Elastic Hashing Algorithm

Goal: Systematically locate files based solely on their name



Algorithm

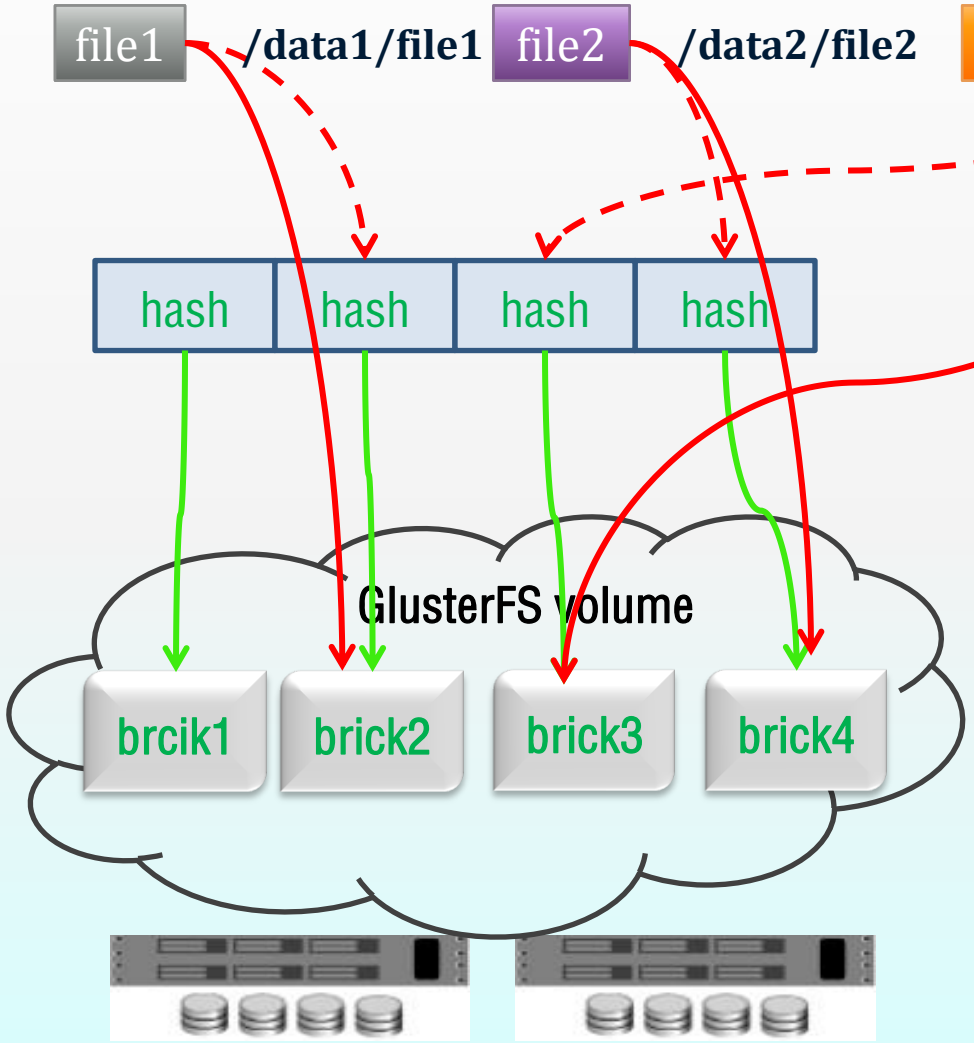
"1) Run path/filename through hash
2) Assign to logical disk based on numerical result
3) Separate logical storage from physical storage



Gluster Mgt Functions: Add, subtract, replicate, heal, recover, etc.

- Provision huge # of virtual disks
- Use hash algorithm to assign systematically & in distributed fashion to *virtual* storage locations
- Virtualization lets you deal flexibly with physical disks
  - ...add, subtract, deal with different disk performance or capacity parameters, etc.
- **WE HAVE MET OUR GOAL!**

弹性Hash算法流程



- 1、使用Davies-Meyer算法计算32位hash值，输入参数为文件名；
- 2、根据hash值在集群中选择子卷（存储服务器），进行文件定位；
- 3、对所选择的子卷进行数据访问。

Brick	Hash range
Brick1	00000000 ~ 3FFFFFFF
Brick2	4FFFFFFF ~ 7FFFFFFF
Brick3	8FFFFFFF ~ BFFFFFFF
Brick4	CFFFFFFF ~ FFFFFFFF

火龙果•整理
uml.org.cn

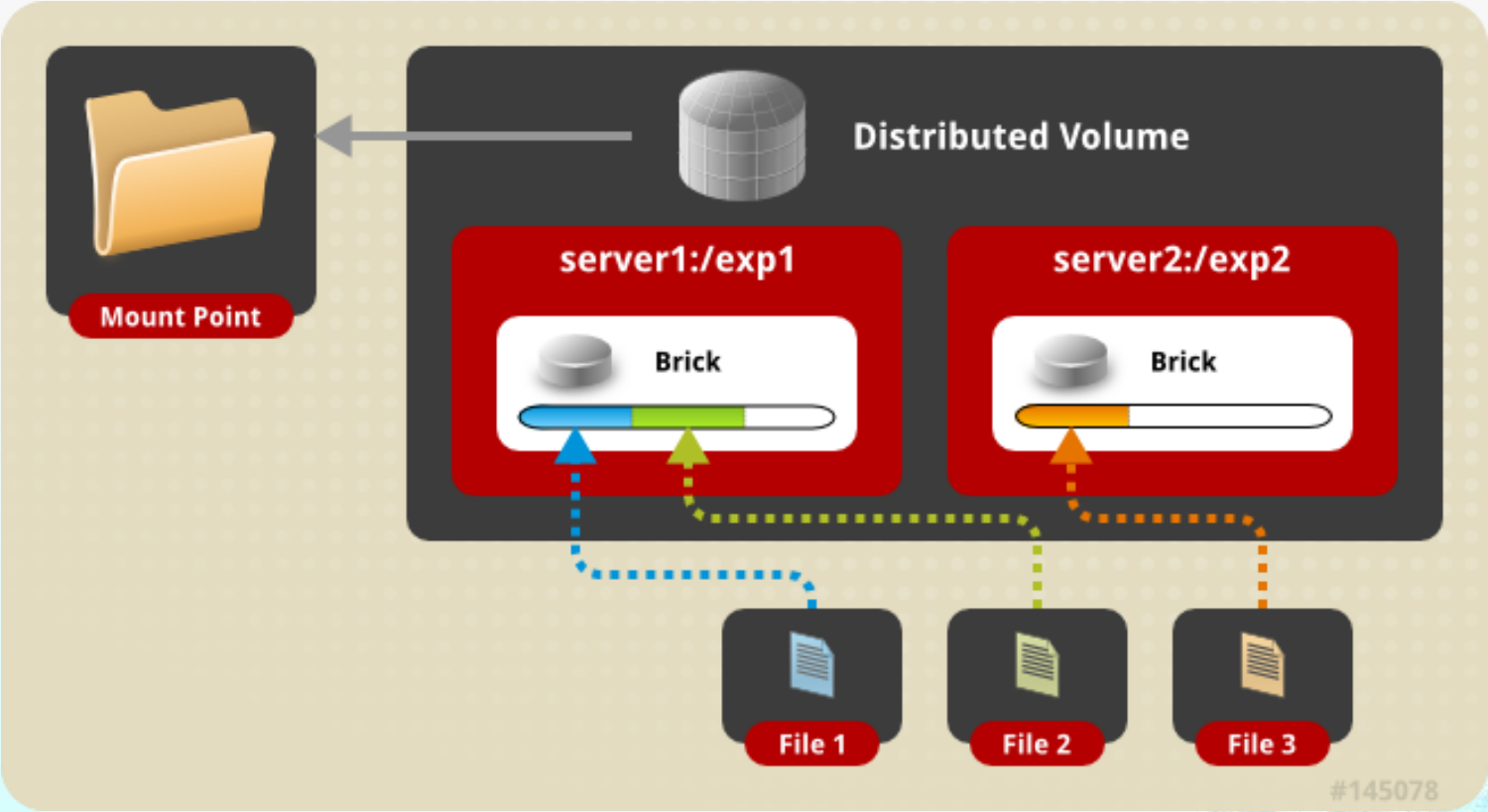
GlusterFS卷类型

- 基本卷
  - 哈希卷（Distributed Volume）
  - 复制卷（Replicated Volume）
  - 条带卷（Striped Volumes）
- 复合卷
  - 哈希复制卷(Distributed Replicated Volume)
  - 哈希条带卷（Distributed Striped Volume）
  - 复制条带卷（Replicated Striped Volume）
  - 哈希复制条带卷（Distributed Replicated Striped Volume）

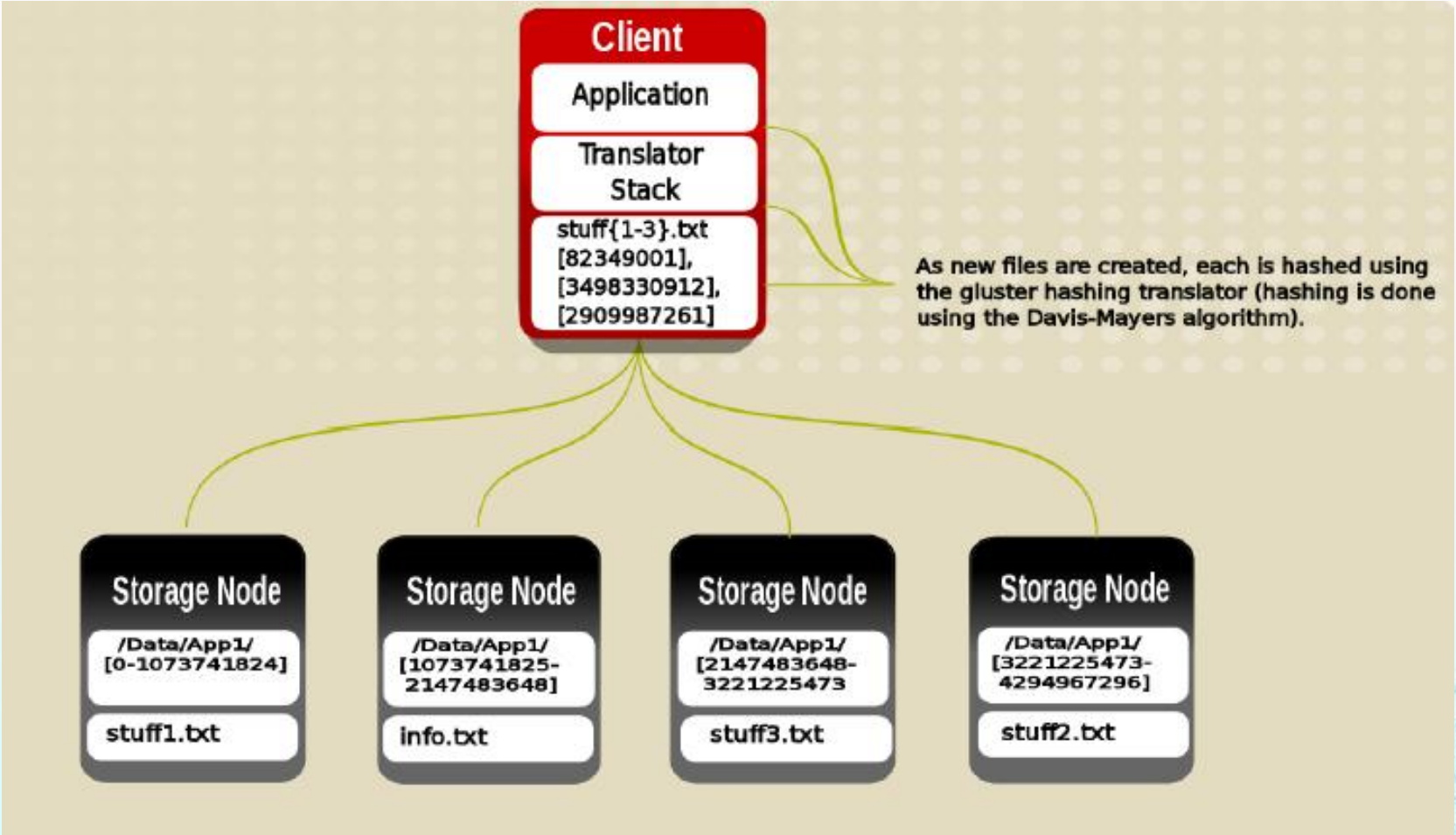
火龙果•整理
uml.org.cn

哈希卷（Distributed Volume）

- 文件通过hash算法在所有brick上分布
- 文件级RAID 0，不具有容错能力



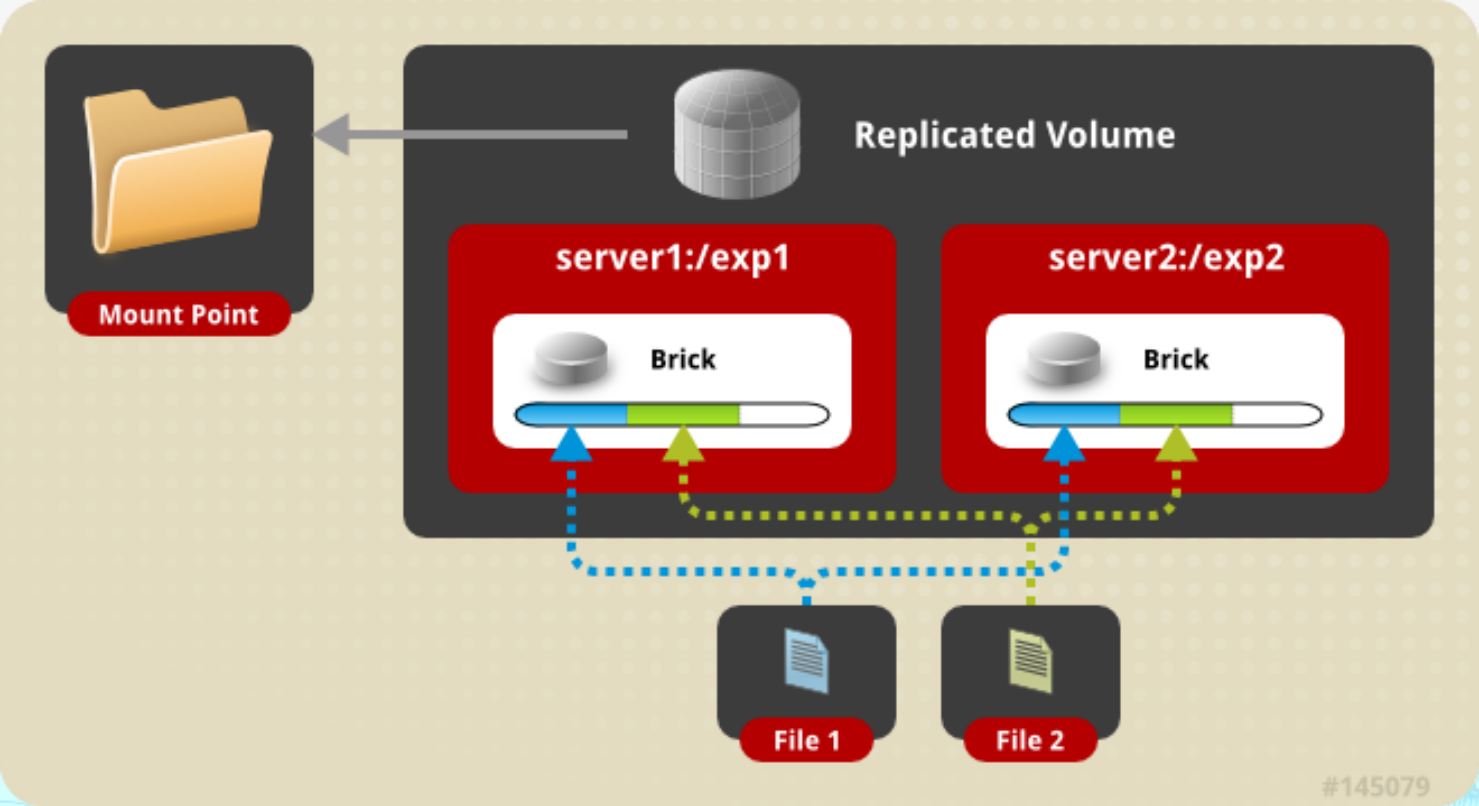
哈希卷工作原理


Client
Application
Translator Stack
stuff{1-3}.txt
[82349001],
[3498330912],
[2909987261]
As new files are created, each is hashed using the gluster hashing translator (hashing is done using the Davis-Mayers algorithm).
Storage Node
/Data/App1/
[0-1073741824]
stuff1.txt
Storage Node
/Data/App1/
[1073741825-
2147483648]
info.txt
Storage Node
/Data/App1/
[2147483648-
3221225473
stuff3.txt
Storage Node
/Data/App1/
[3221225473-
4294967296]
stuff2.txt

火龙果•整理
uml.org.cn

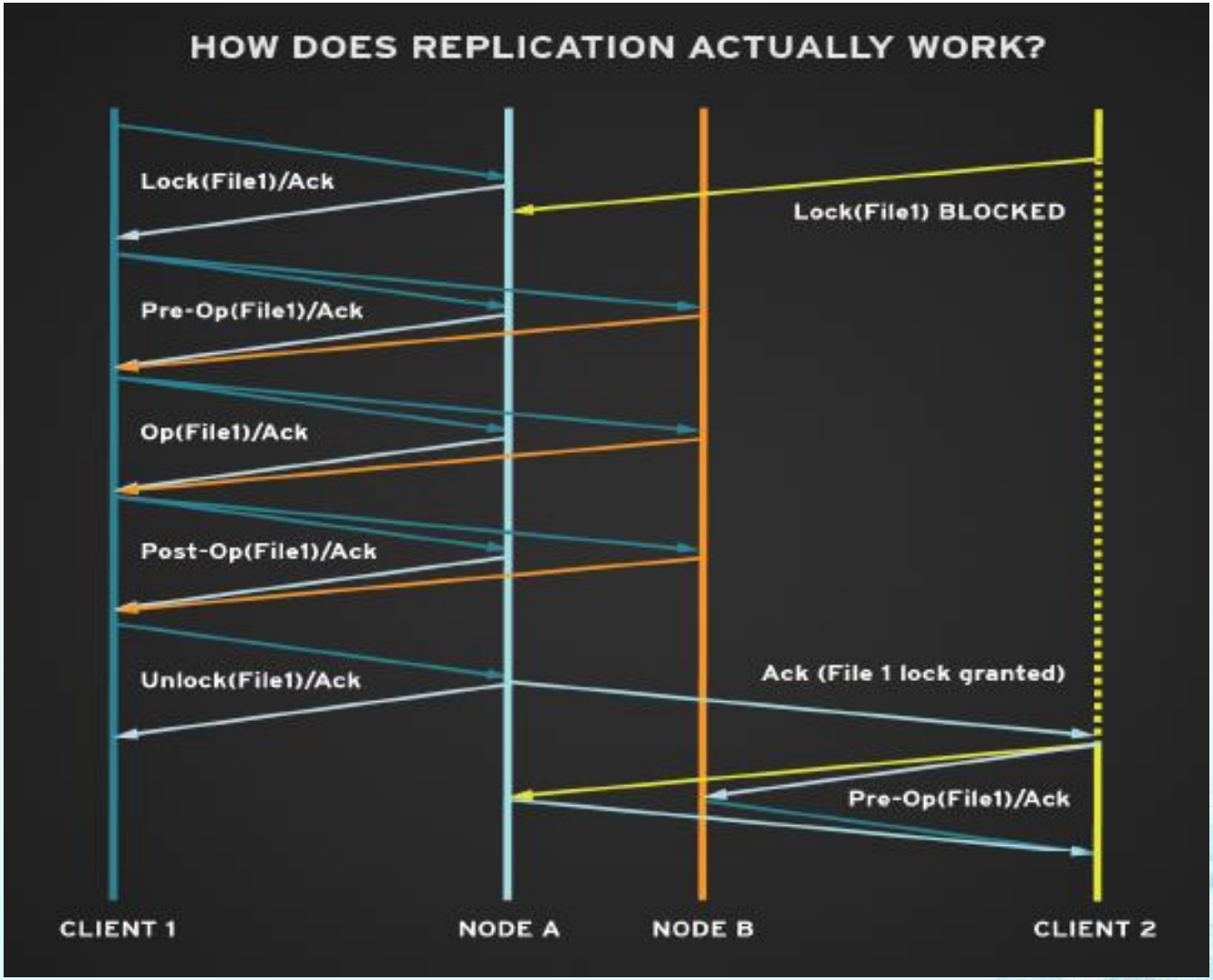
复制卷（Replicated Volume）

- 文件同步复制到多个brick上
- 文件级RAID 1，具有容错能力
- 写性能下降，读性能提升



火龙果•整理
uml.org.cn

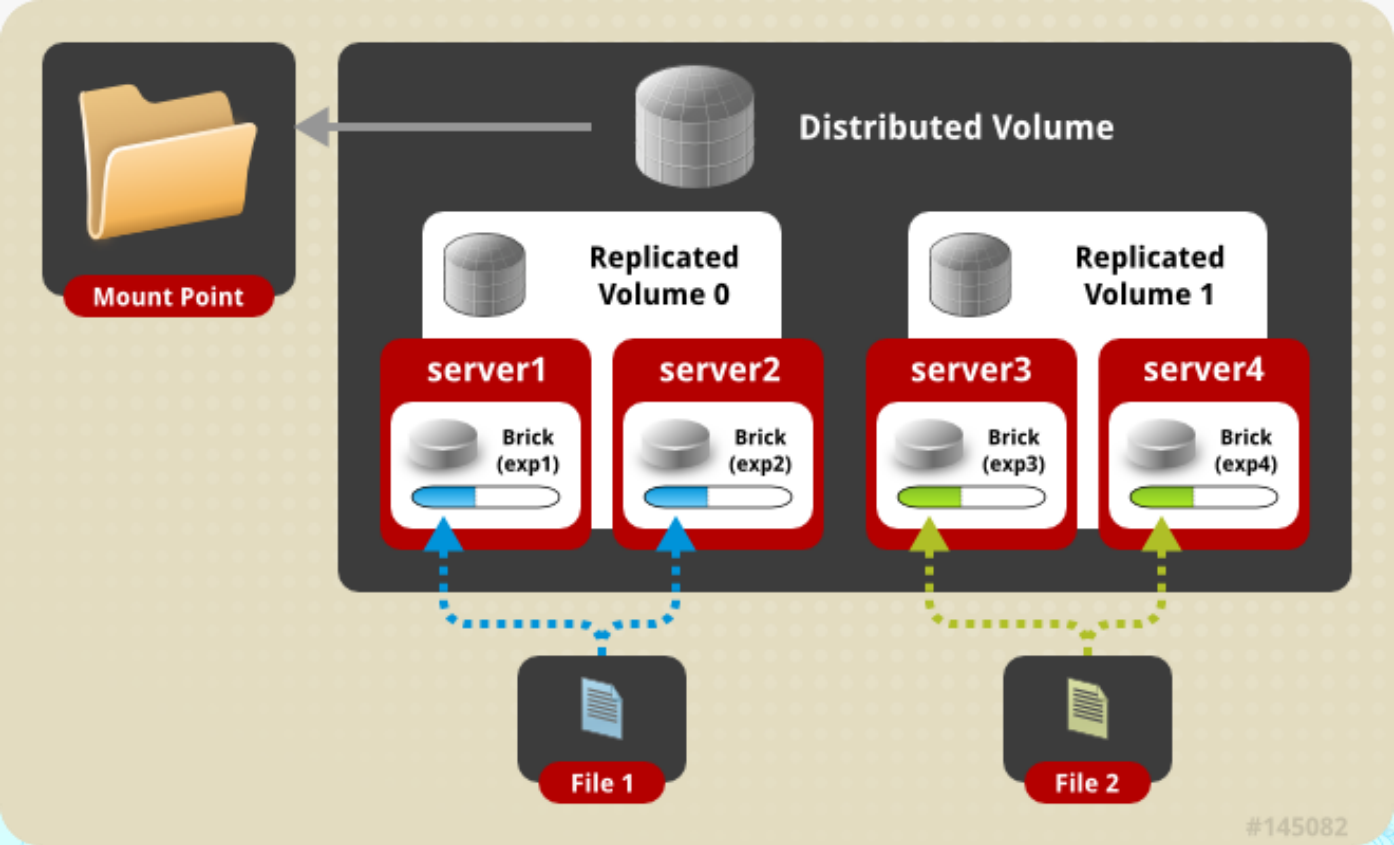
复制卷工作原理



火龙果•整理
uml.org.cn

复合卷：哈希 + 复制

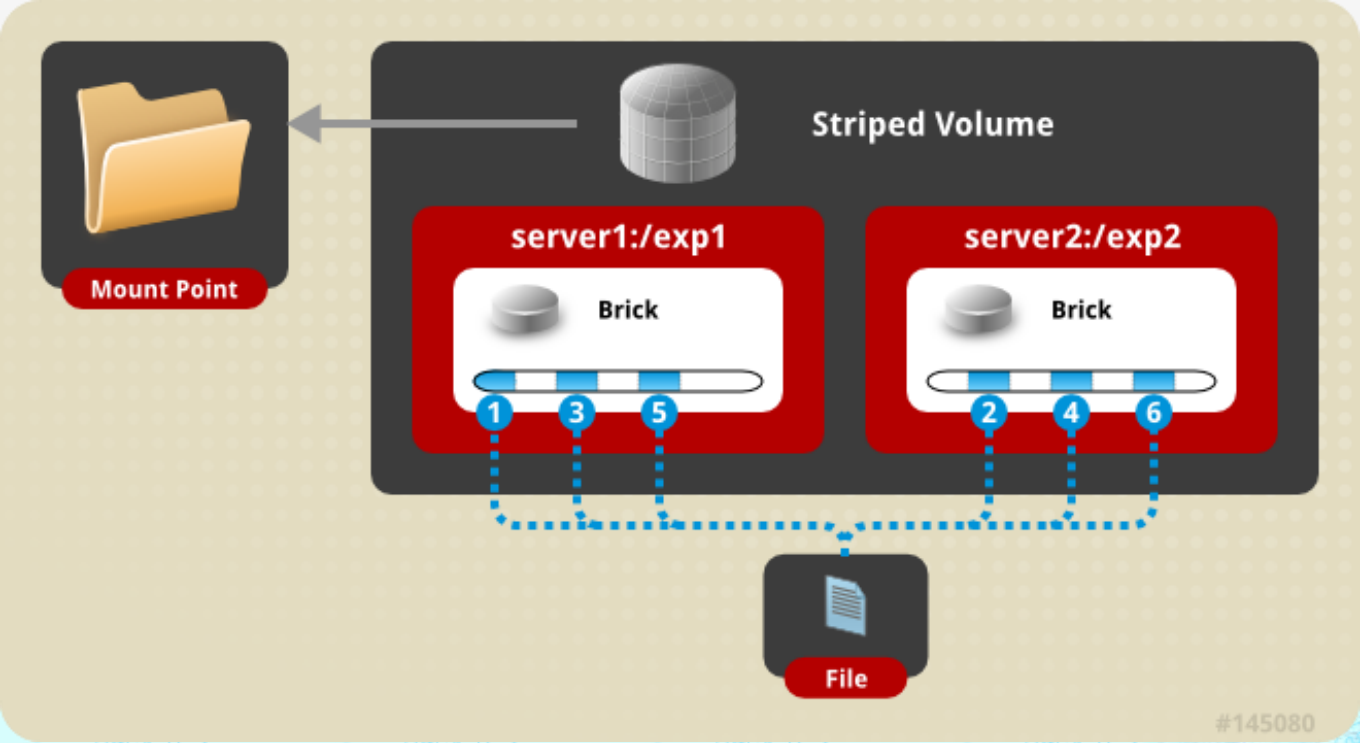
- 哈希卷和复制卷的复合方式
- 同时具有哈希卷和复制卷的特点



火龙果•整理
uml.org.cn

条带卷（Striped Volumes）

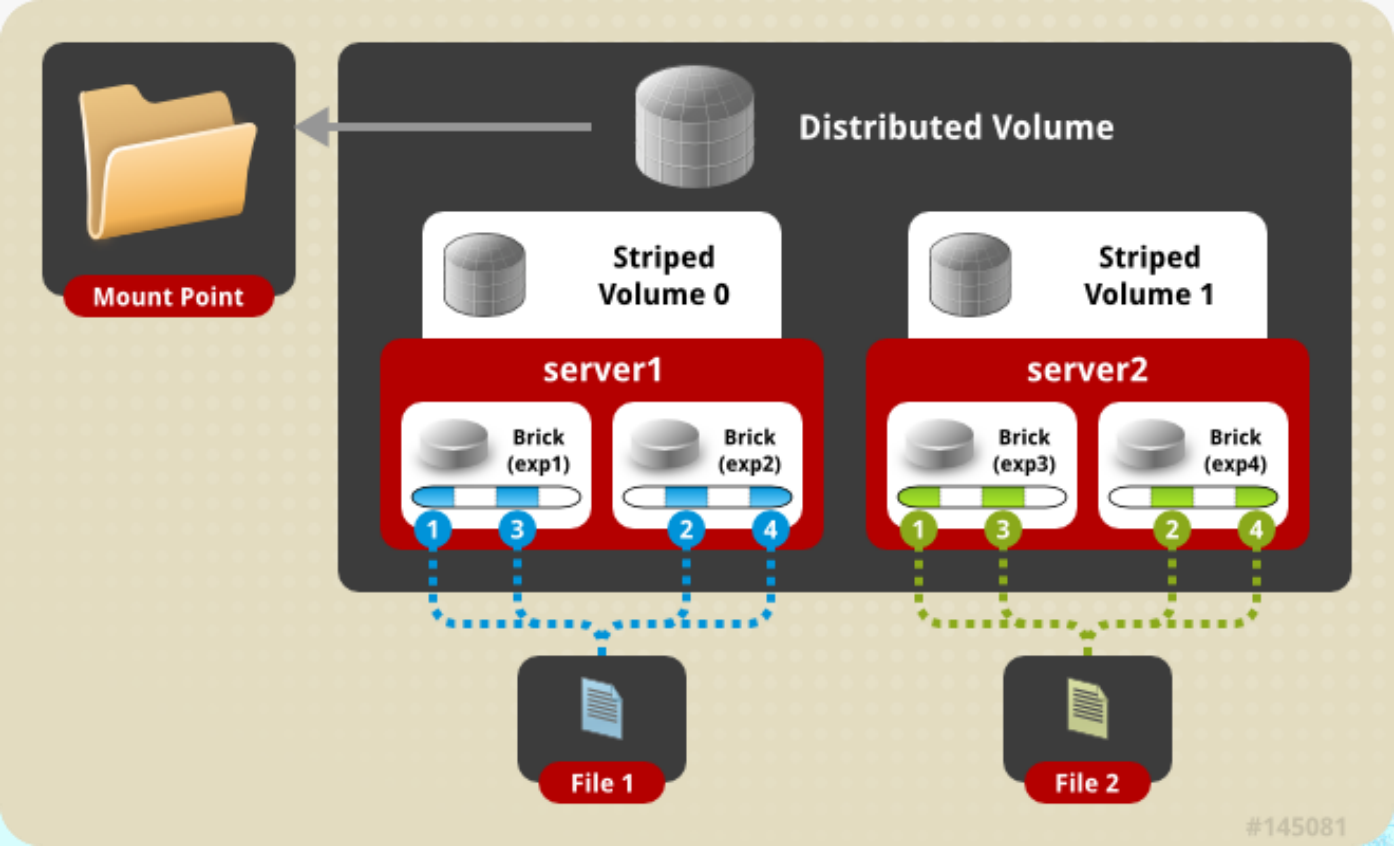
- 单个文件分布到多个brick上，支持超大文件
- 类似RAID 0，以Round-Robin方式
- 通常用于HPC中的超大文件高并发访问



火龙果•整理
uml.org.cn

复合卷：哈希 + 条带

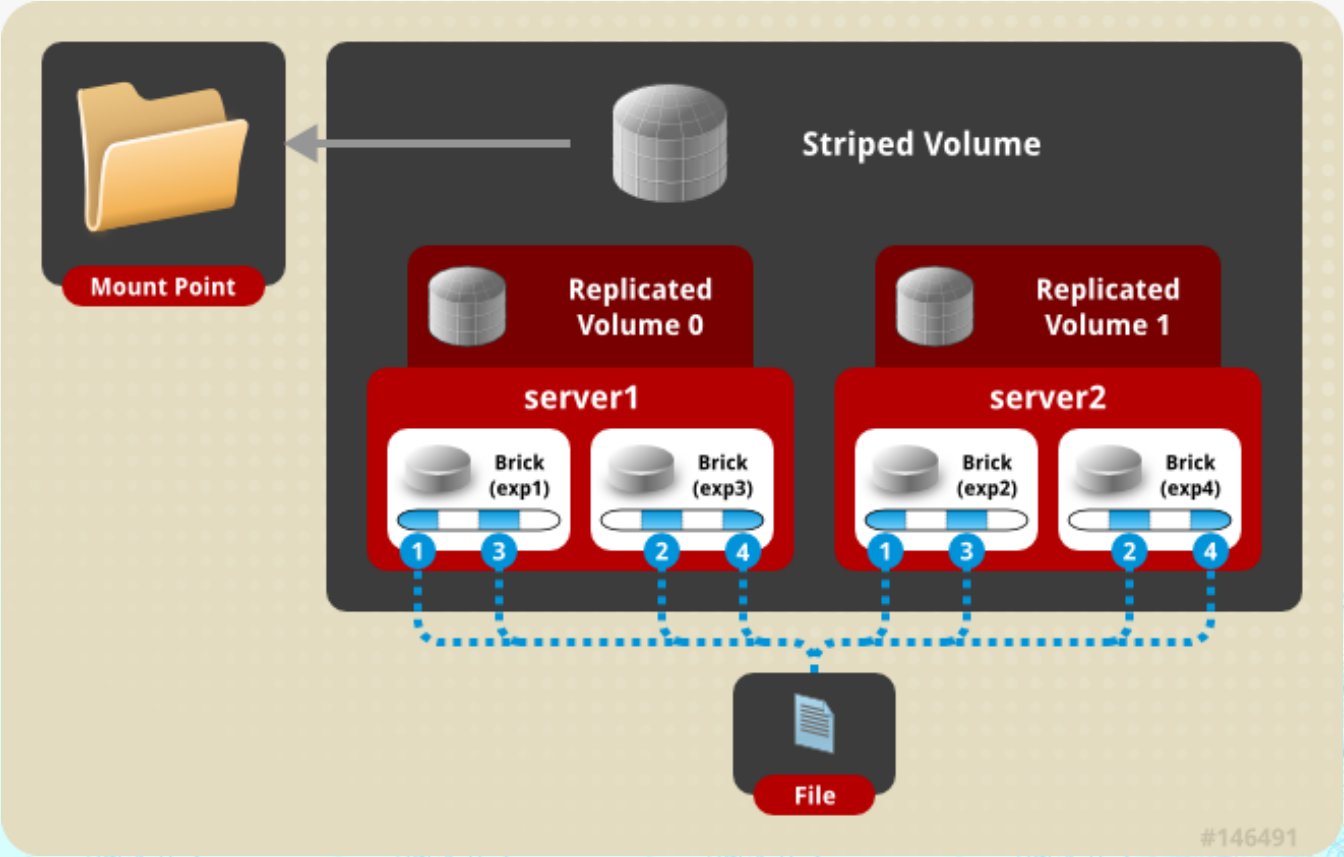
- 哈希卷和条带卷的复合方式
- 同时具有哈希卷和条带卷的特点



火龙果•整理
uml.org.cn

复合卷：条带 + 复制

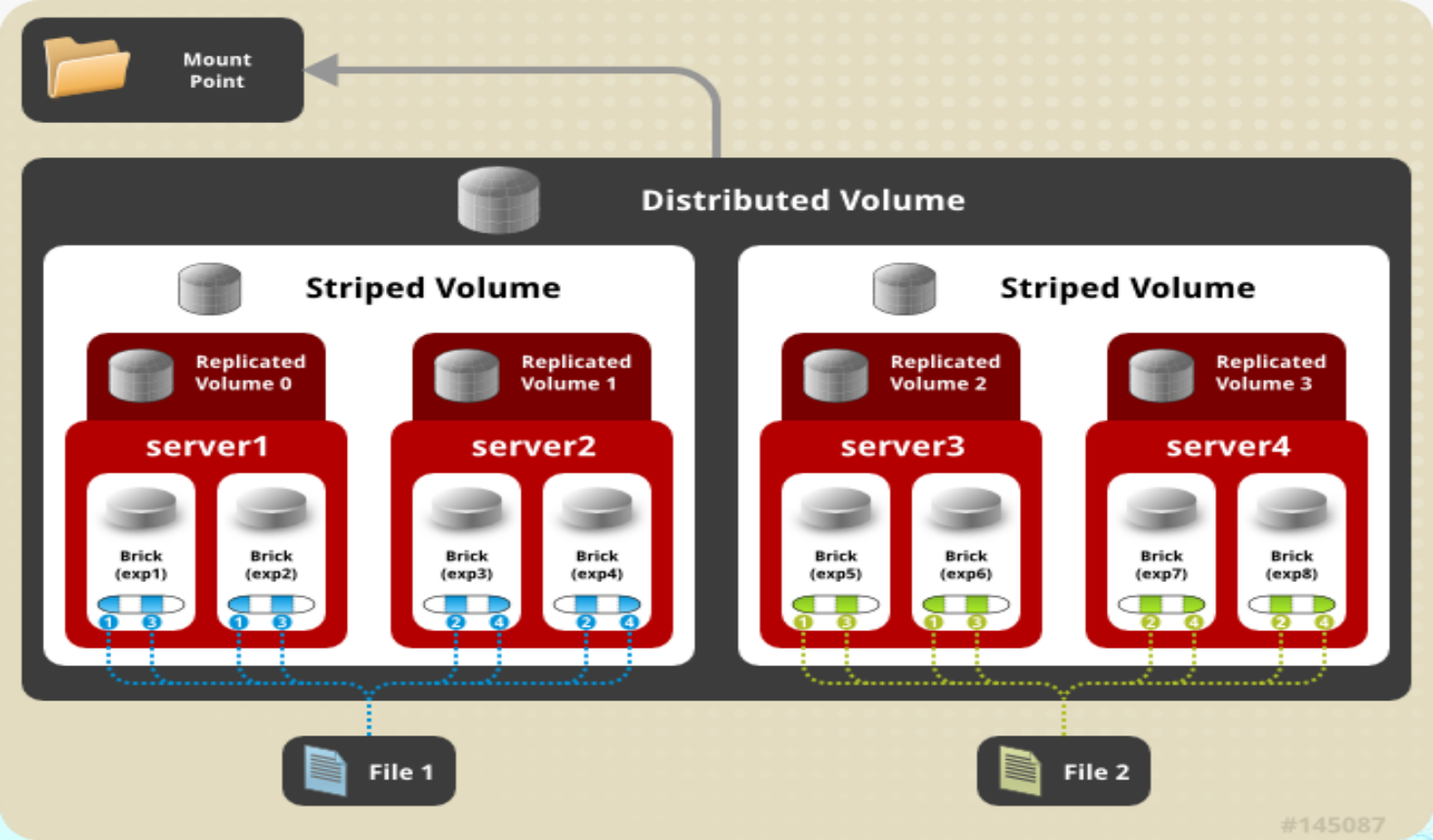
- 类似RAID 10
- 同时具有条带卷和复制卷的特点



火龙果•整理
uml.org.cn

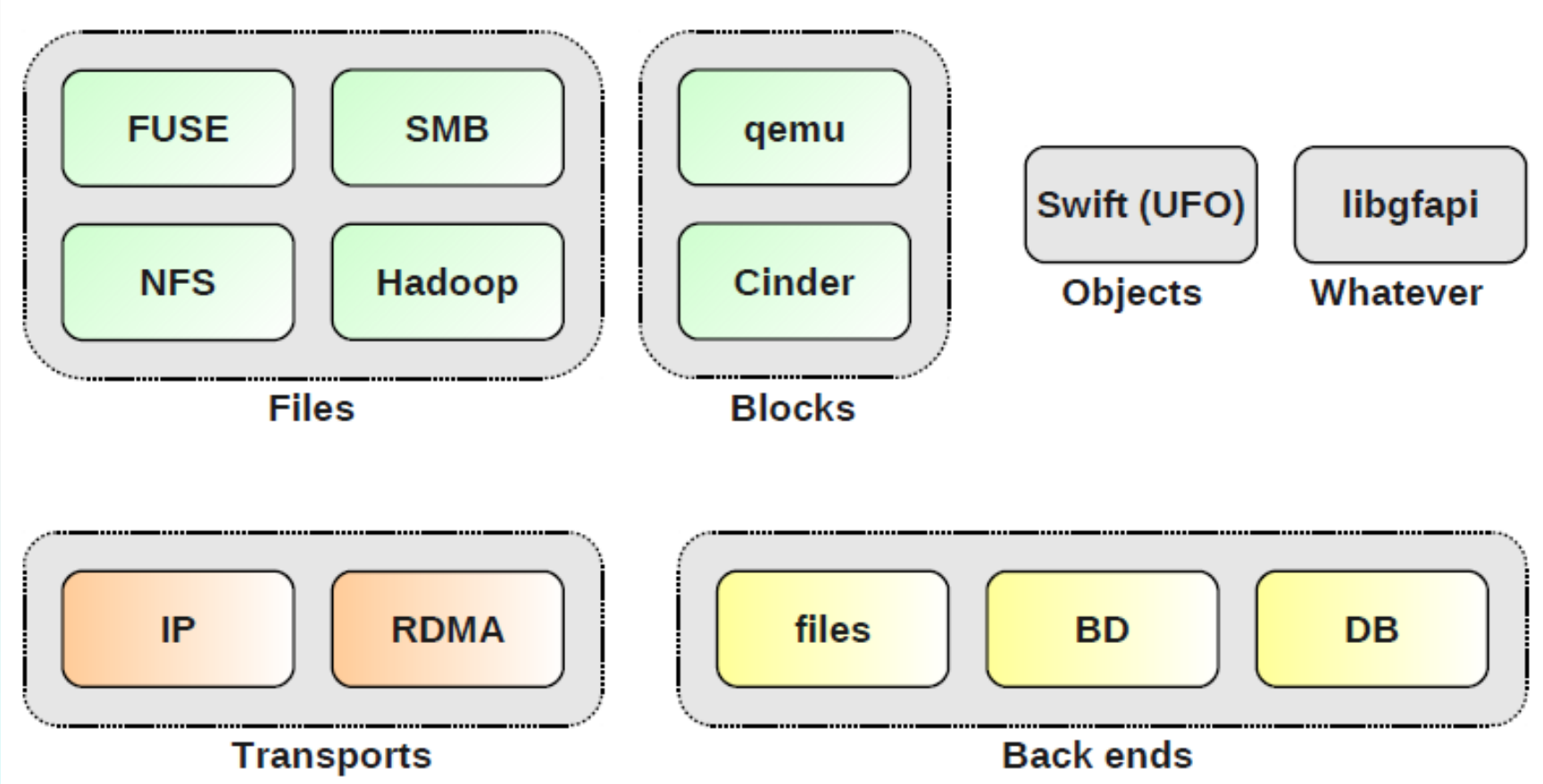
复合卷：哈希 + 条带 + 复制

- 三种基本卷的复合卷
- 通常用于类Map Reduce应用



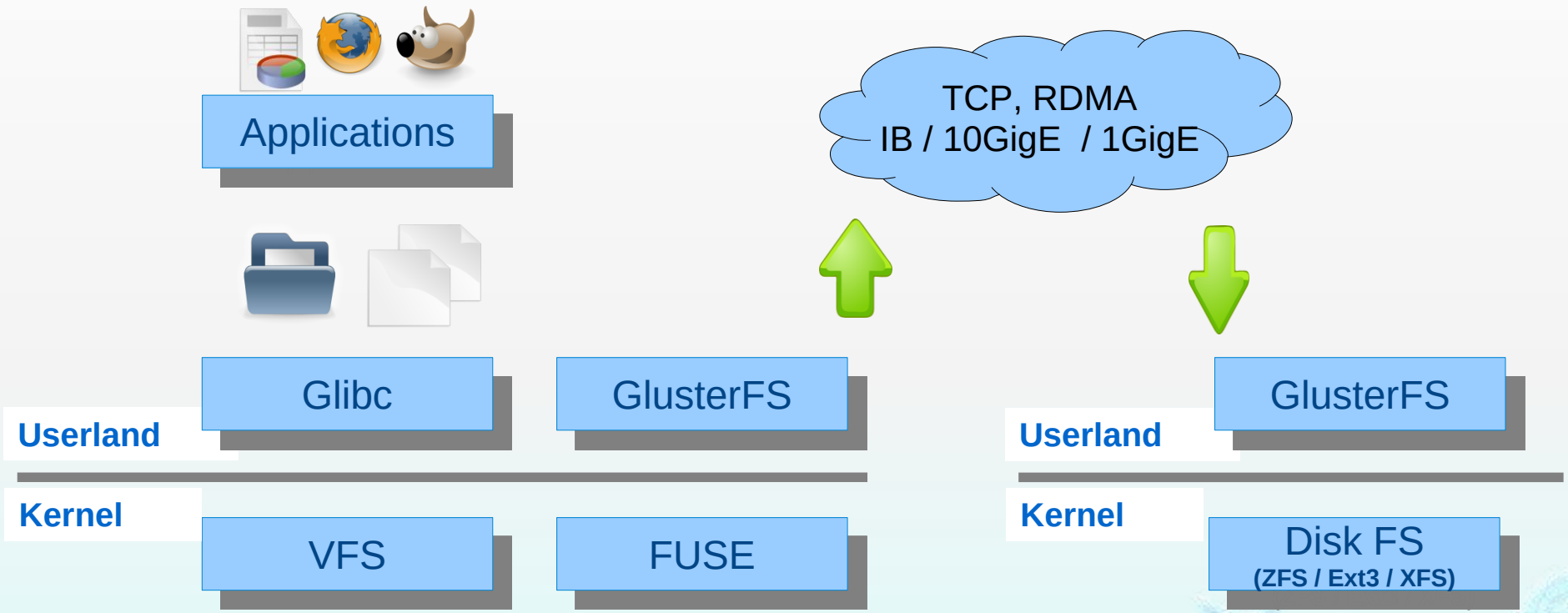
火龙果•整理
uml.org.cn

GlusterFS访问接口



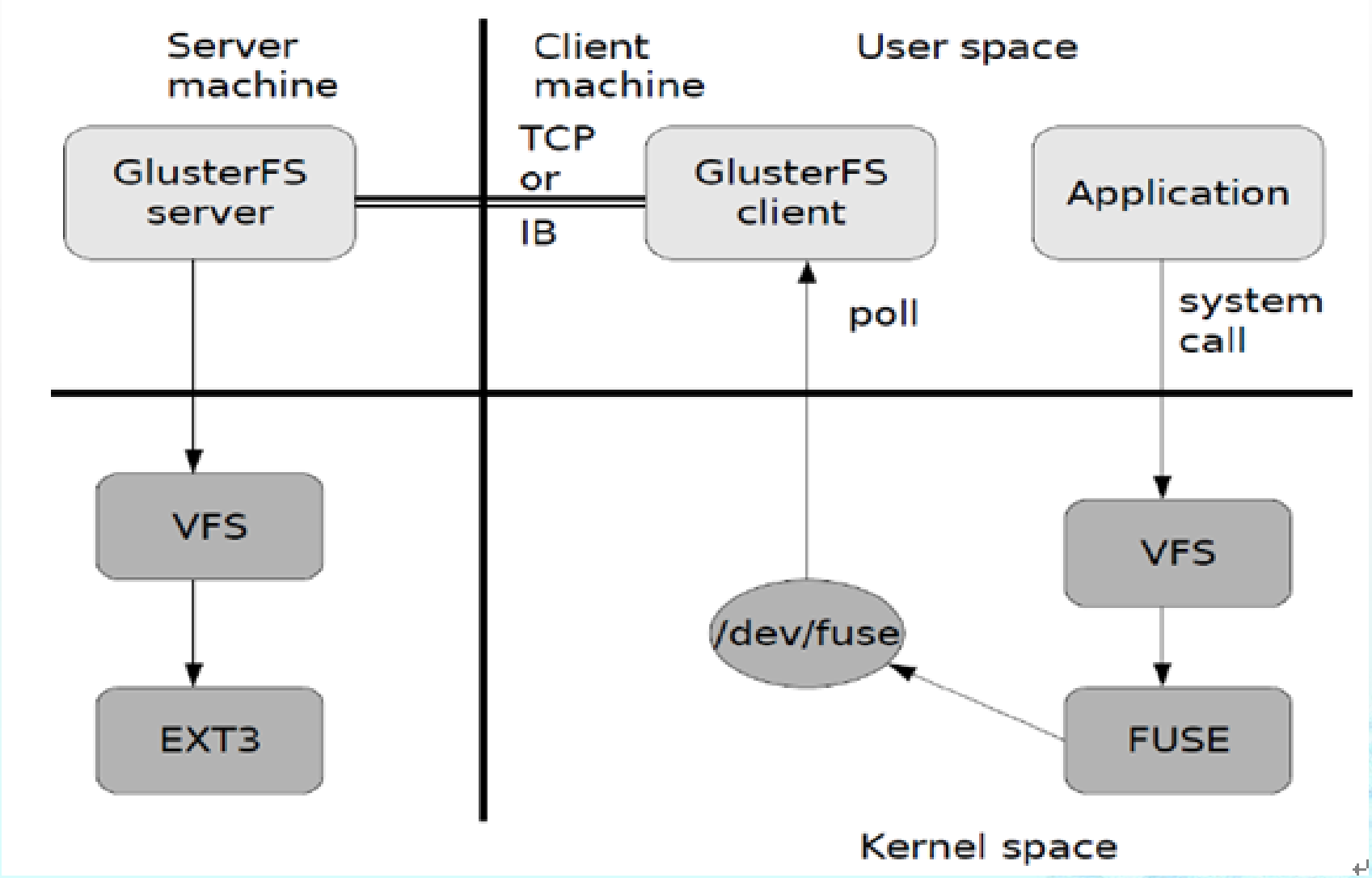
火龙果•整理
uml.org.cn

GlusterFS – FUSE Architecture



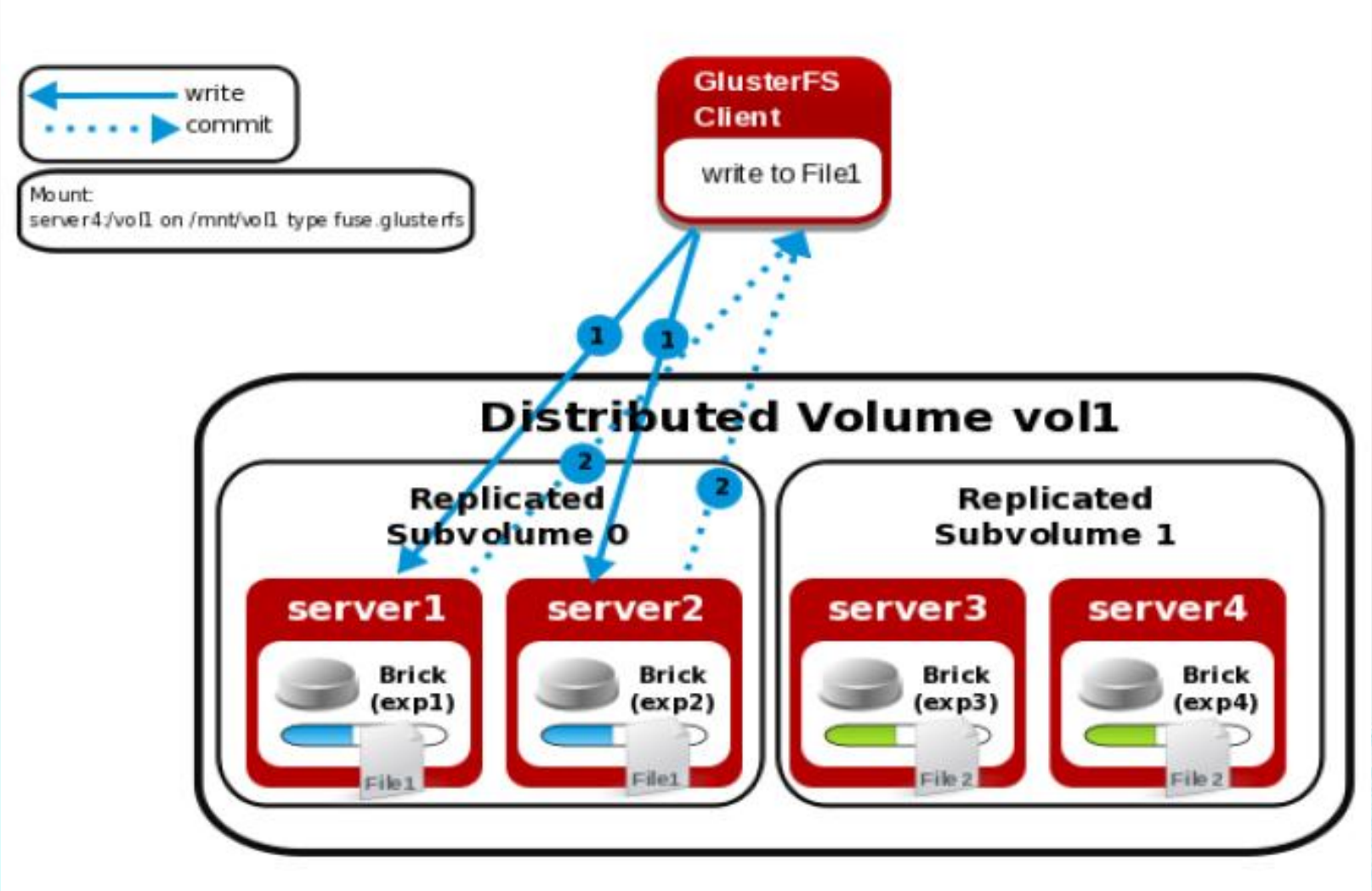
火龙果•整理
uml.org.cn

GlusterFS数据流



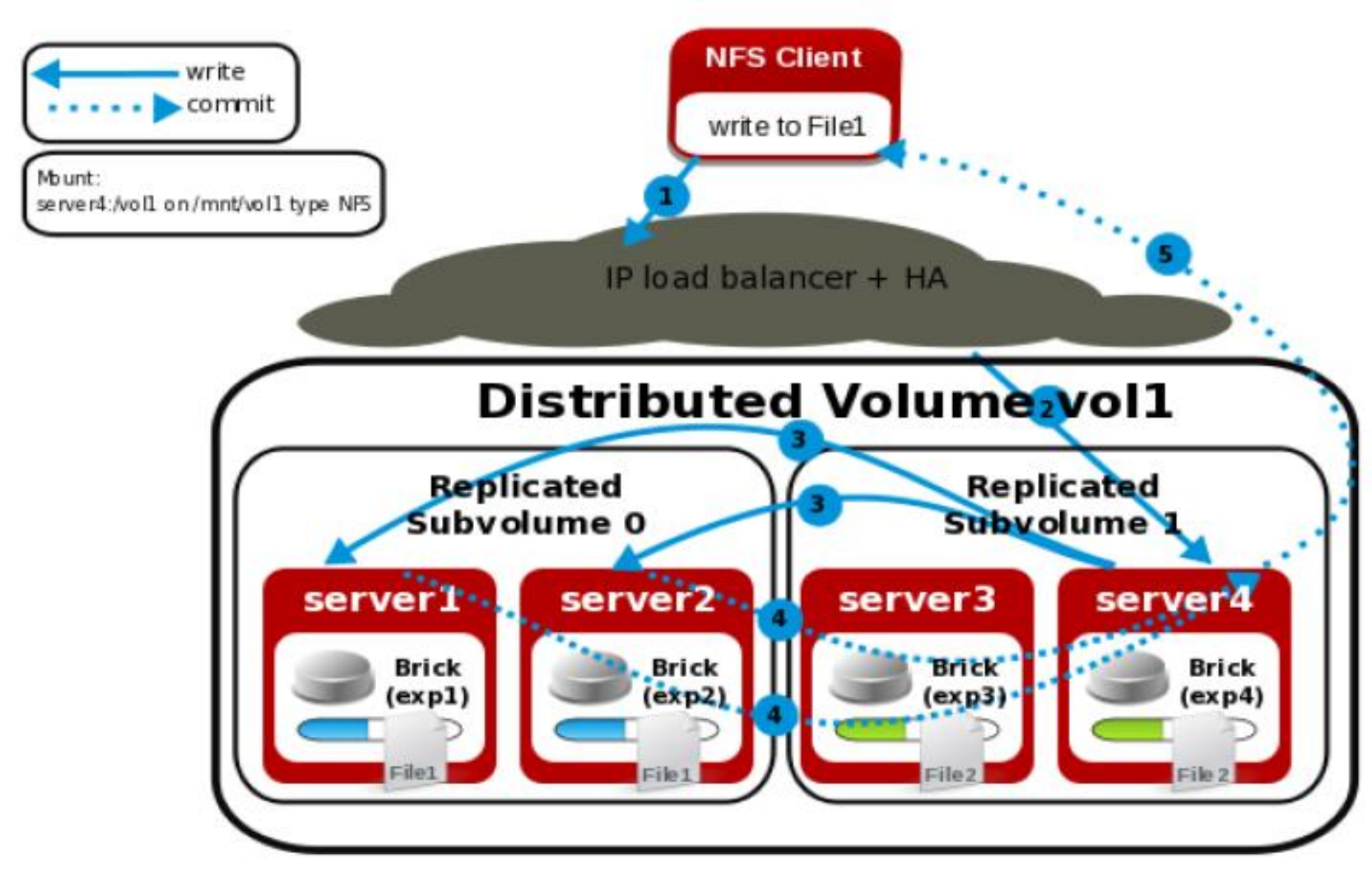
火龙果•整理
uml.org.cn

FUSE 访 问



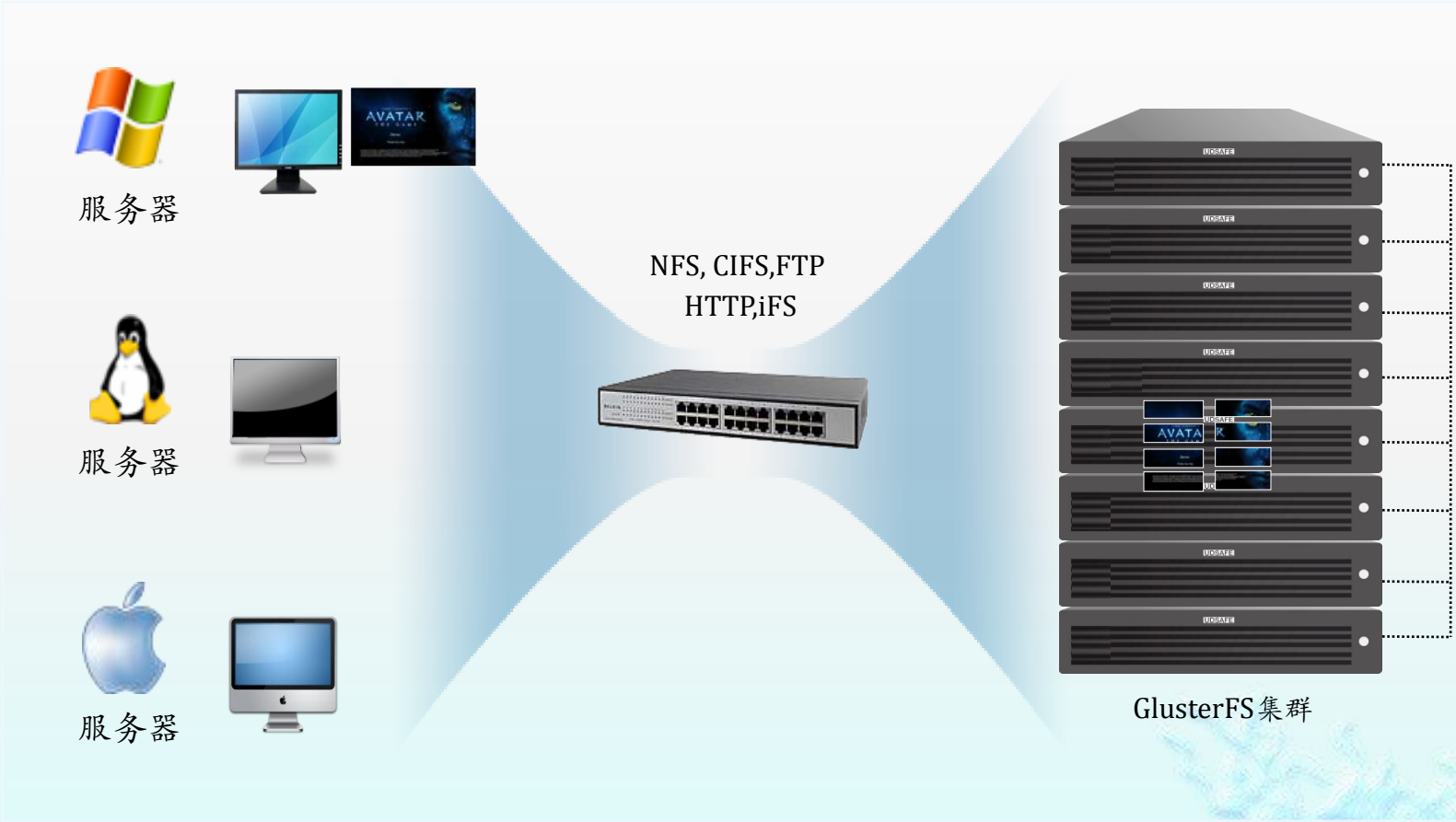
火龙果•整理
uml.org.cn

NFS/CIFS 访 问



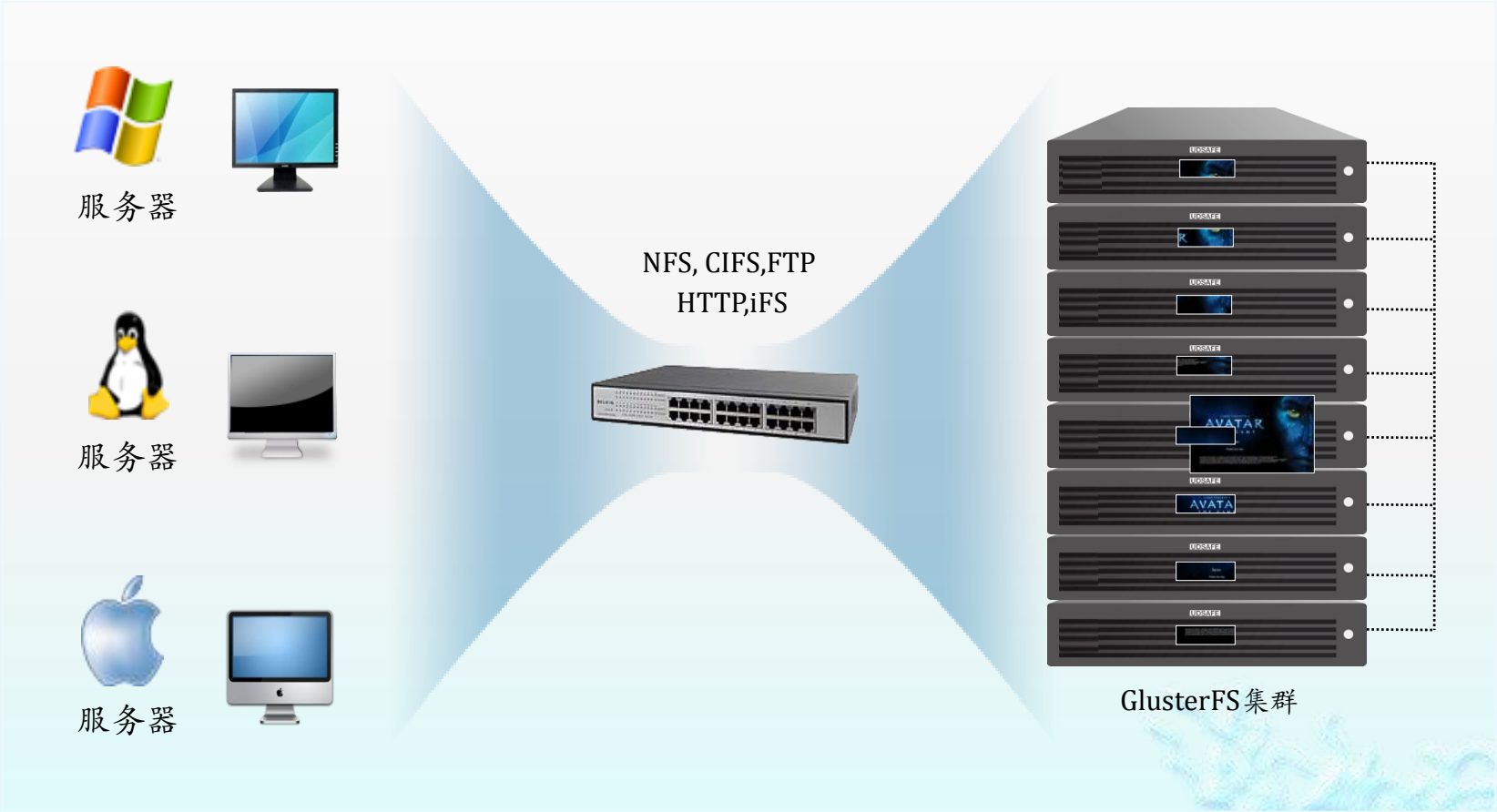
火龙果•整理
uml.org.cn

写入一个文件



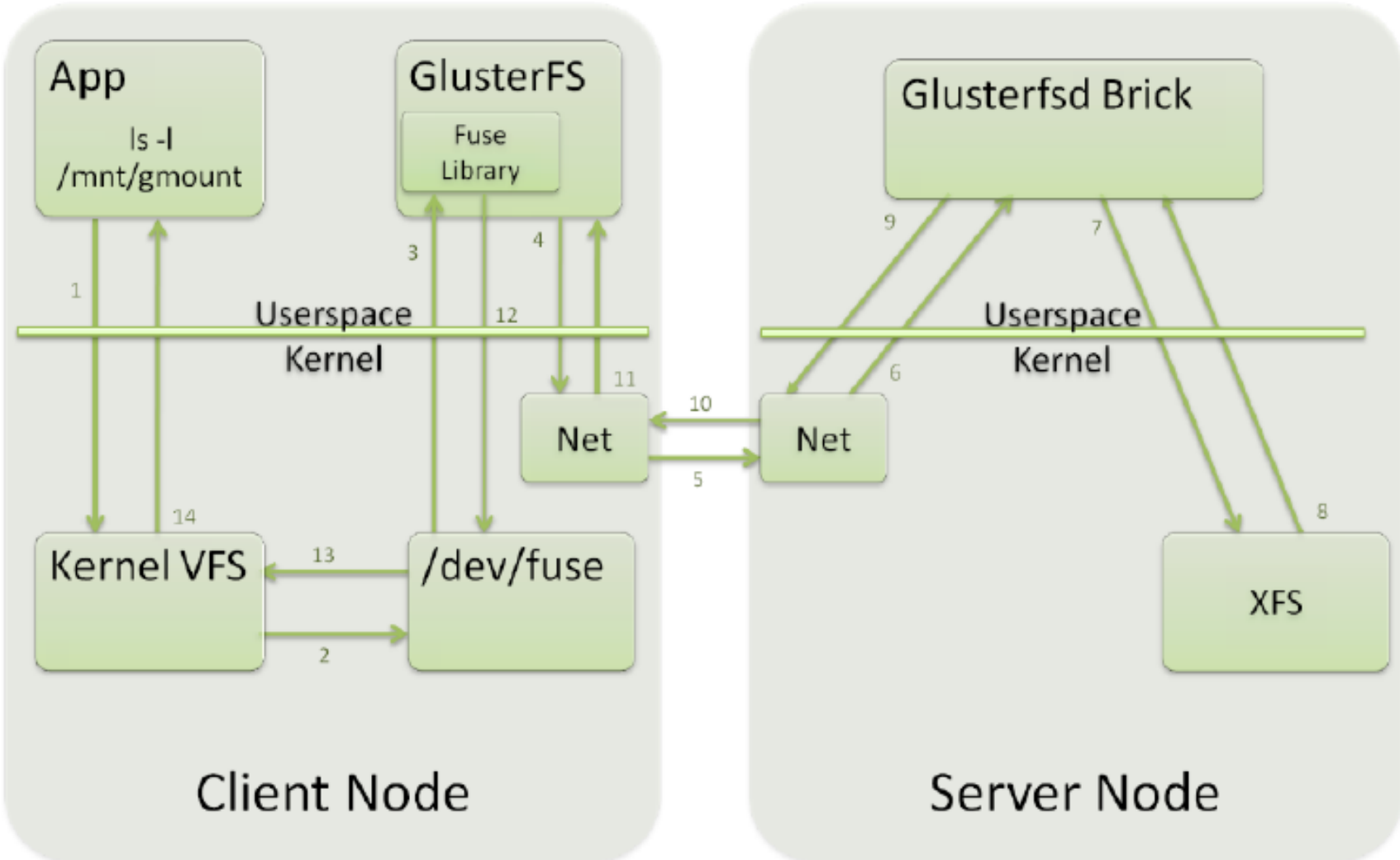
火龙果•整理
uml.org.cn

读取一个文件



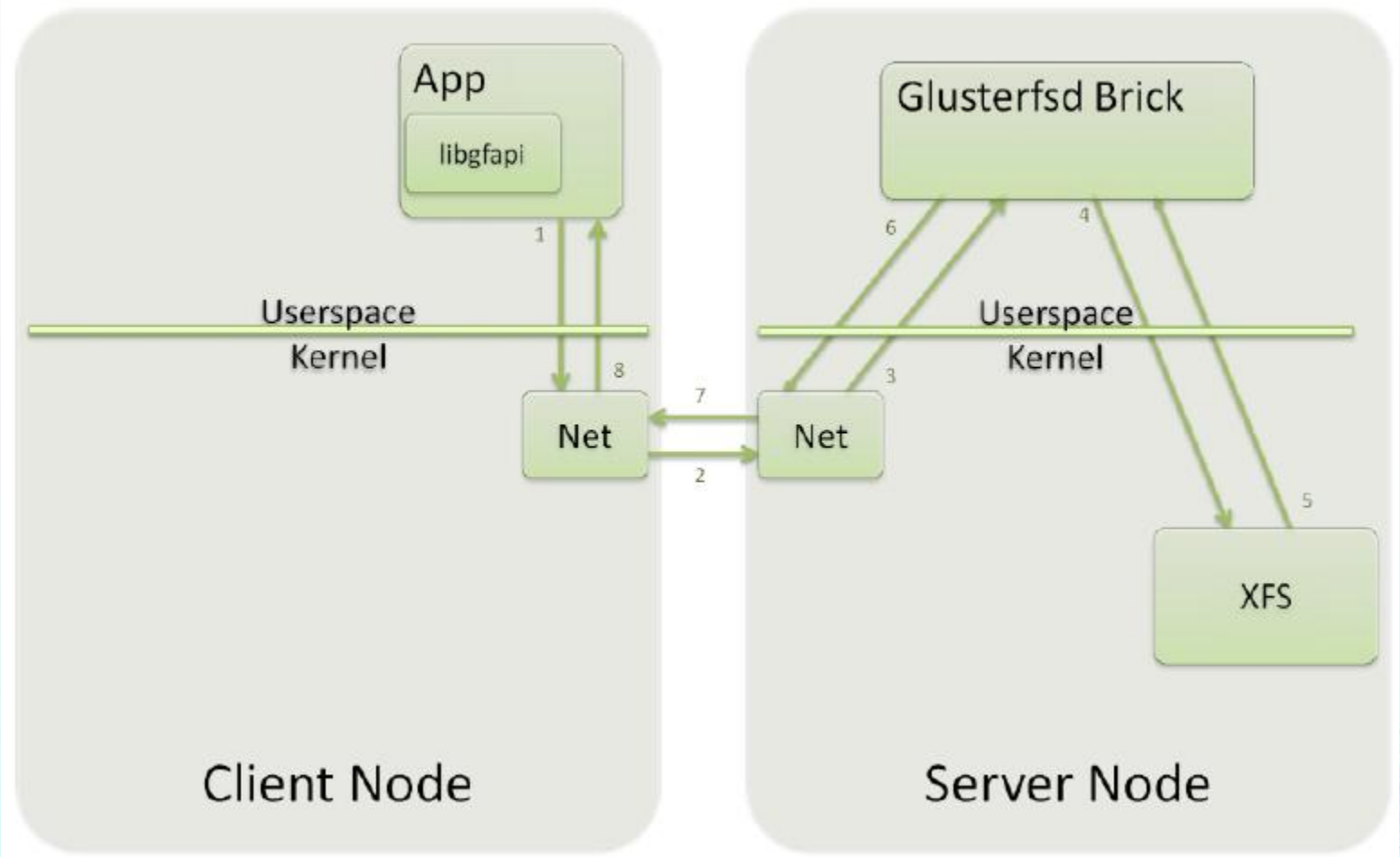
火龙果•整理
uml.org.cn

FUSE w/ Libgfapi 访问



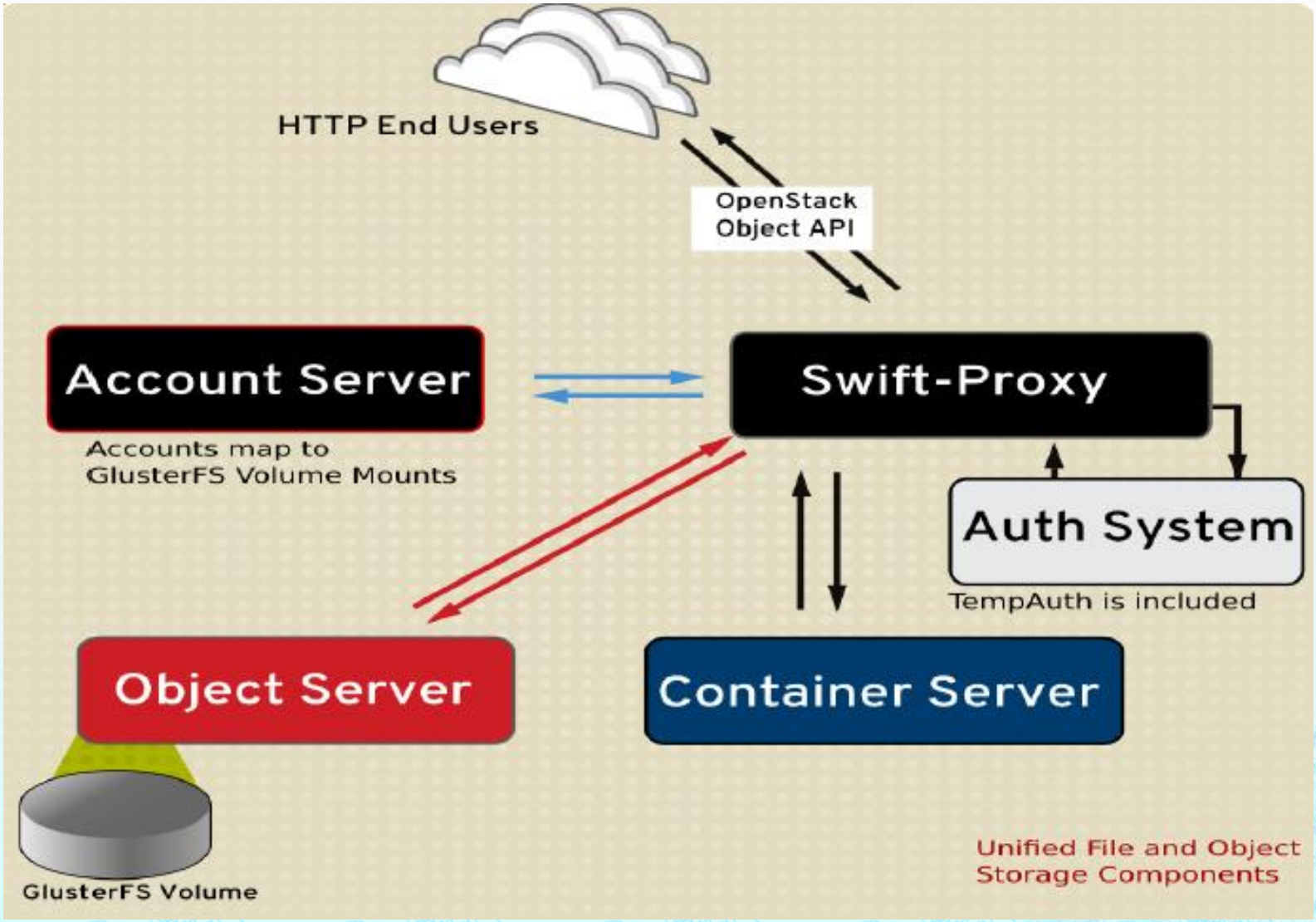
火龙果•整理
uml.org.cn

libgfapi 访 问



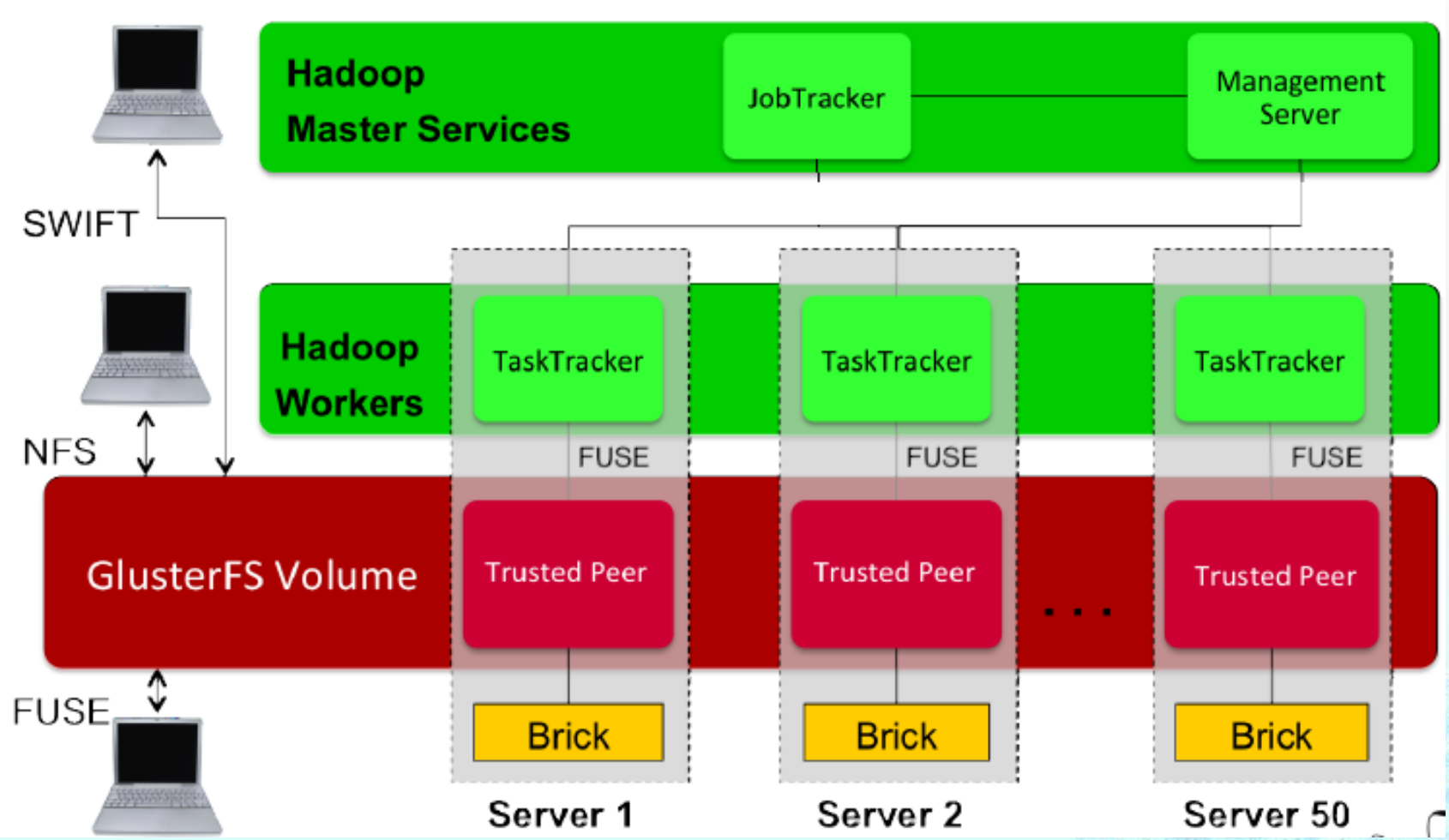
火龙果•整理
uml.org.cn

REST API 访 问



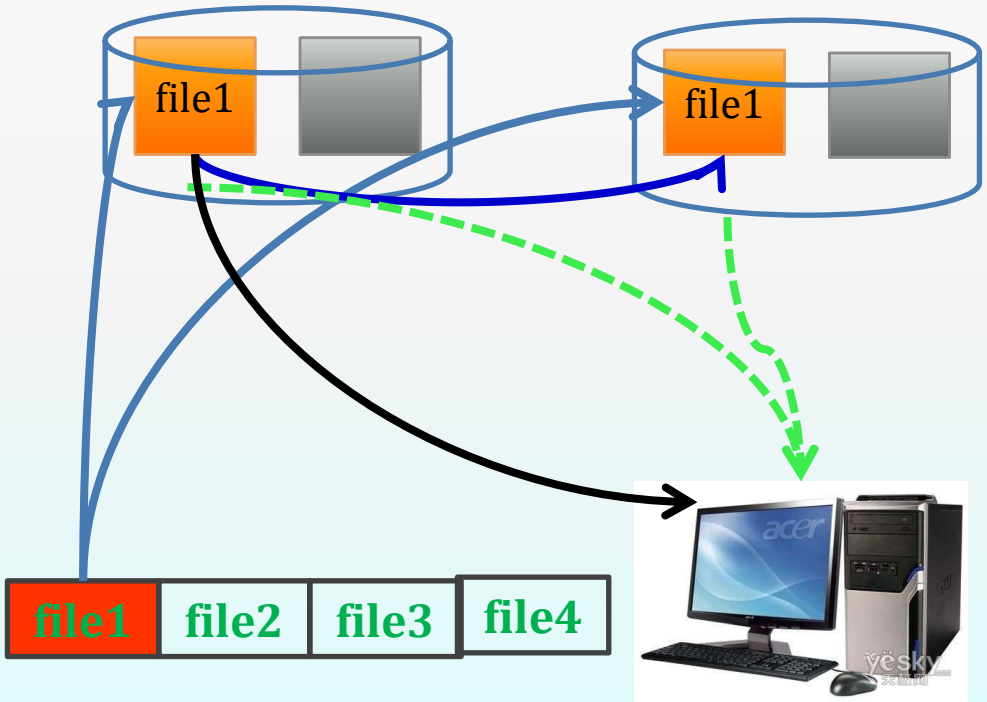
火龙果•整理
uml.org.cn

Hadoop 访 问



火龙果•整理
uml.org.cn

数据自修复 Self-heal



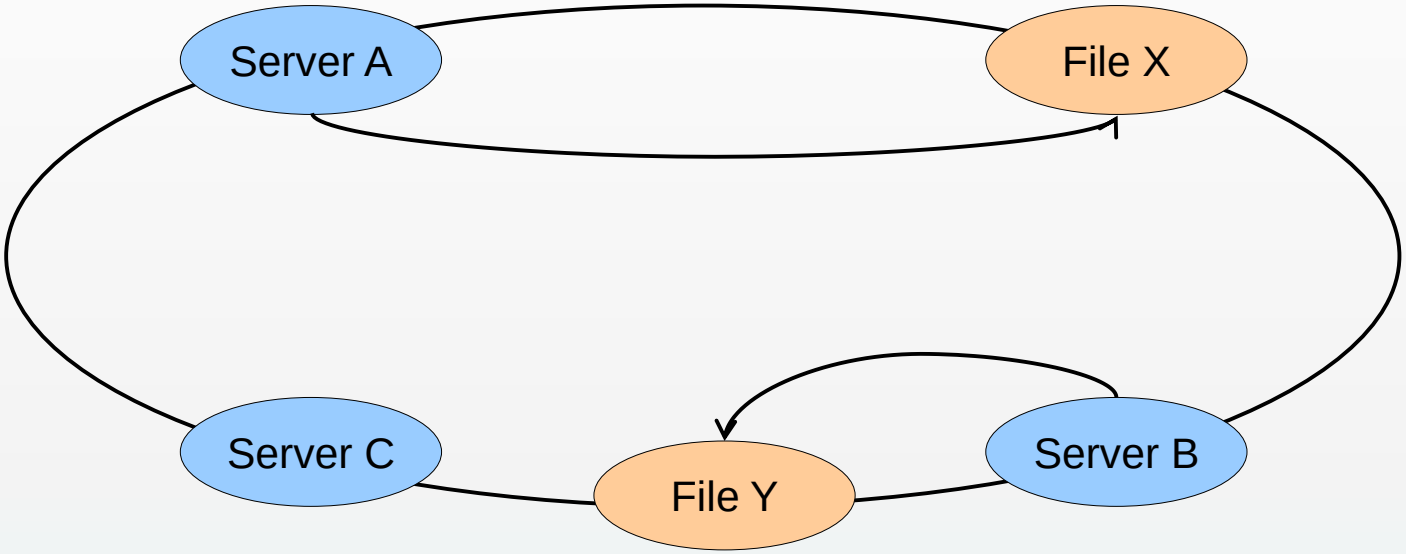
Self-heal发展

- 第一代：按需同步进行
- 第二代：完全人工扫描
- 第三代：并发自动修复(3.3)
- 第四代：基于日志

- 镜像卷文件副本保持一致性
- 触发时机：访问文件目录时
- 判断依据：扩展属性
- 脑裂问题：报错或按规则处理

火龙果•整理
uml.org.cn

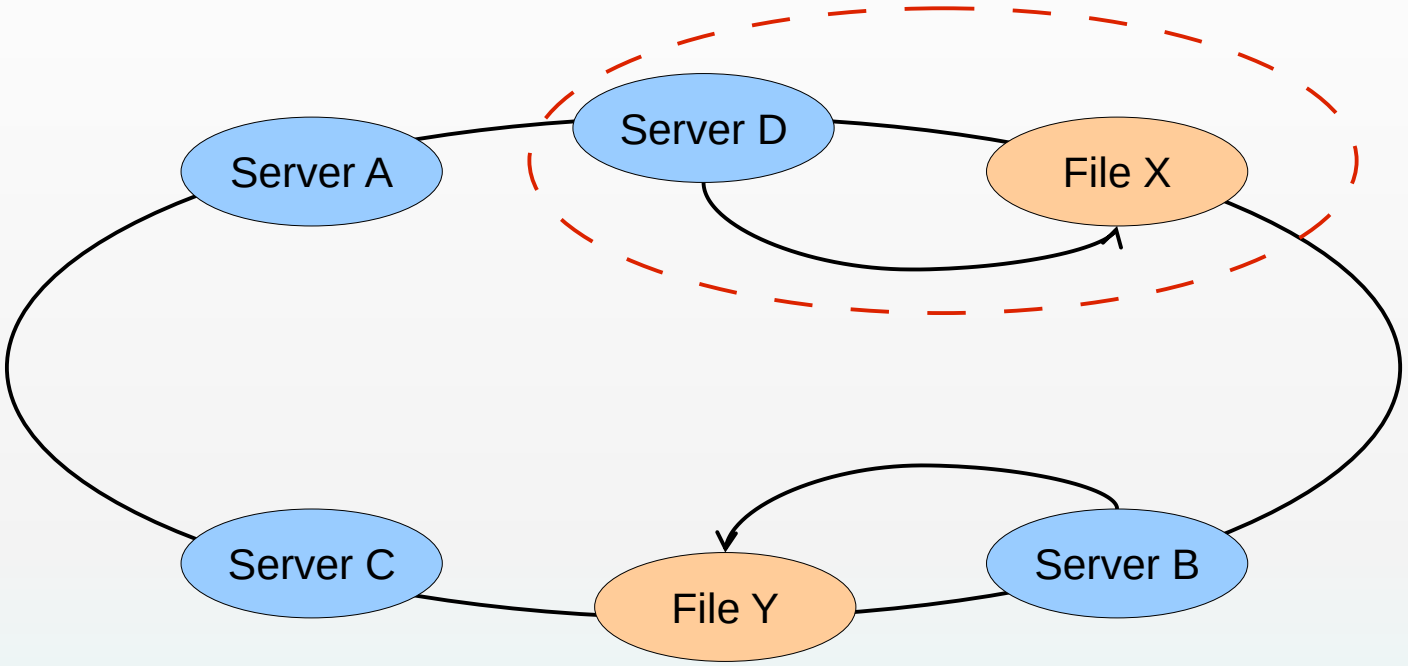
Distributed Hash Table (DHT)



- GlusterFS弹性扩展的基础
- 确定目标hash和brick之间的映射关系

火龙果•整理
uml.org.cn

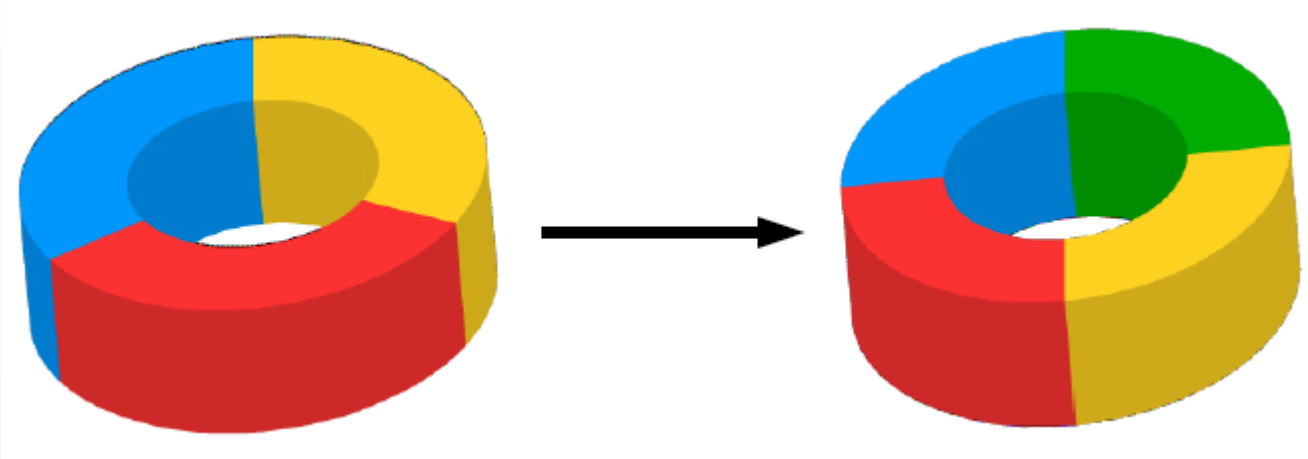
添加节点



- 添加新节点，最小化数据重新分配
- 老数据分布模式不变，新数据分布到所有节点上
- 执行rebalance，数据重新分布

火龙果•整理
uml.org.cn

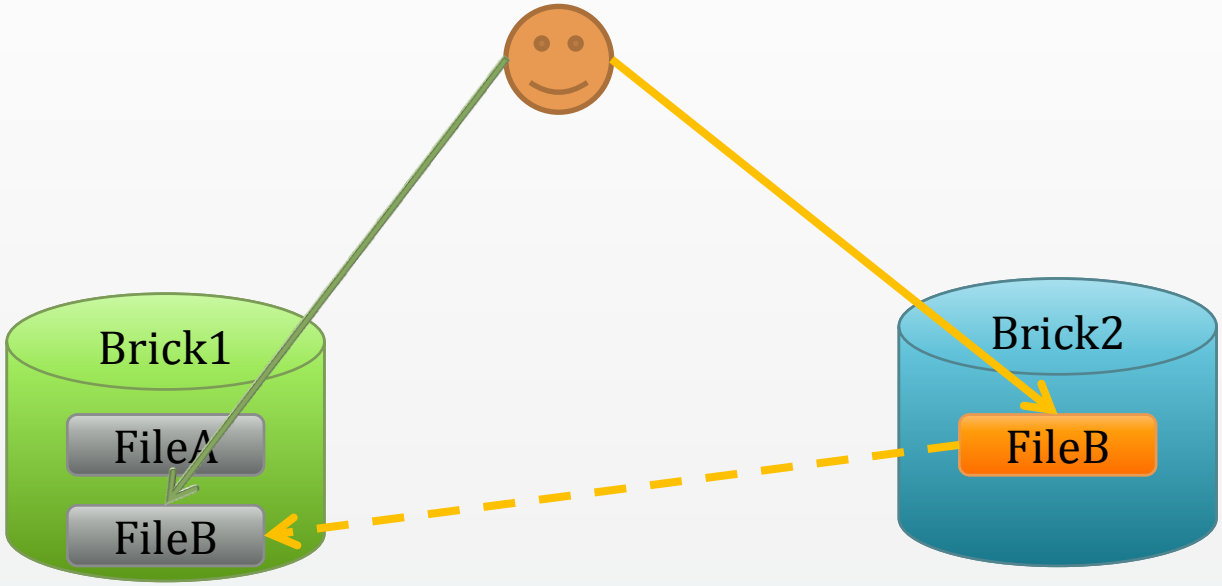
容量负载均衡



- Hash范围均衡分布，节点一变动全局
- 目标：优化数据分布，最小化数据迁移
- 数据迁移自动化、智能化、并行化

火龙果•整理
uml.org.cn

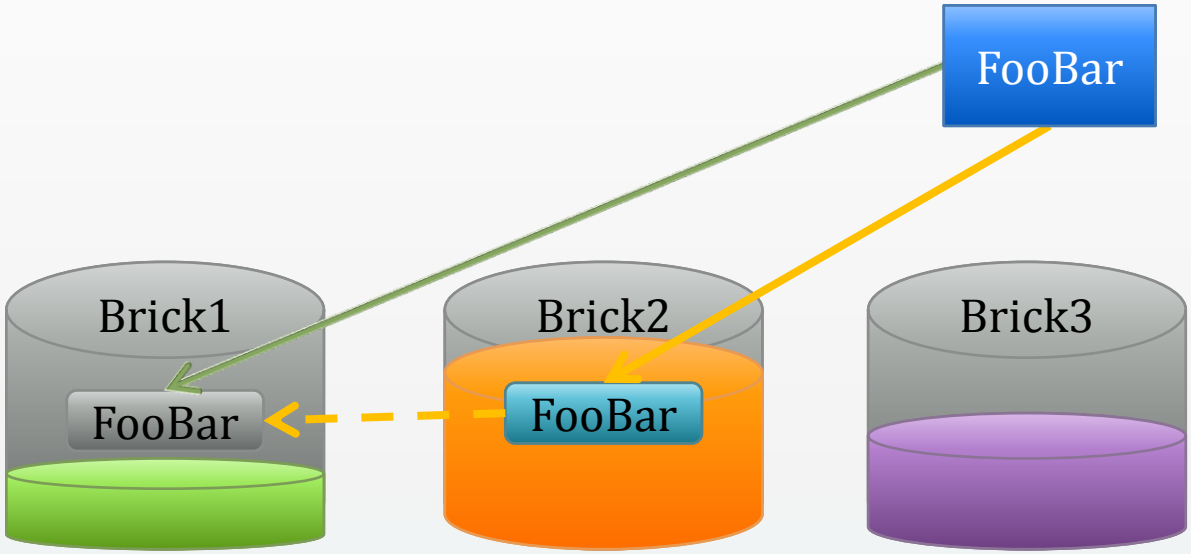
文件更名



- 文件更名：FileA → FileB
- 原先的hash映射关系失效，大文件难以实时迁移
- 采用文件符号链接，访问时解析重定向

火龙果•整理
uml.org.cn

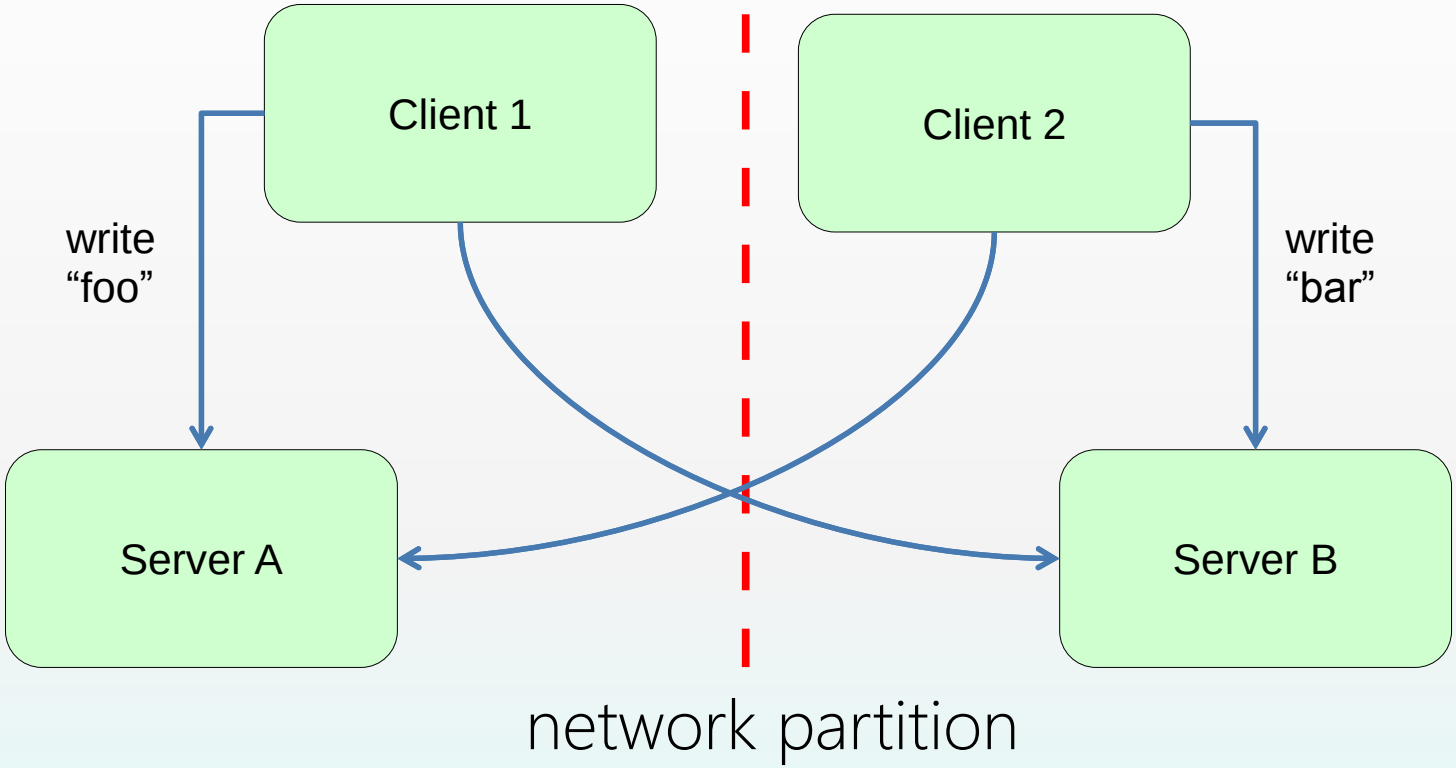
容量负载优先



- 设置容量阈值，优先选择可用容量充足brick
- Hash目标brick上创建文件符号链接
- 访问时解析重定向

火龙果•整理
uml.org.cn

Split Brain



- 裂脑如何产生的？
- 解决方法：1、报错处理；2、Quorum方法(N=2？)；3、仲裁机制

火龙果•整理
uml.org.cn

Q & A