

The SACC logo is rendered in a large, bold, white sans-serif font. It features a blue glow effect and a blue swoosh underline. The background of the entire image is a dark blue with a complex, glowing wireframe pattern of geometric shapes and lines, creating a sense of depth and technology.

SACC

2021 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2021

数字转型 架构重塑

IT168.com

ChinaUnix

ITPUB

云上会议 网络直播 | 2021.5.20-2021.5.22

高性能分布式文件系统FastCFS

——可以跑数据库的文件系统

余庆

2021-05-10

个人简介

- 余庆是分布式文件系统 FastDFS 作者
- 曾任职于新浪、雅虎中国和阿里巴巴
- 目前正在进行下一代分布式文件系统 FastCFS 的研发和推广工作

FastCFS和FastDFS是什么关系？

- FastDFS：对象存储系统
- FastCFS：通用文件系统

数据库后端存储之关键指标

- 高可靠
- 高性能
- 高可用

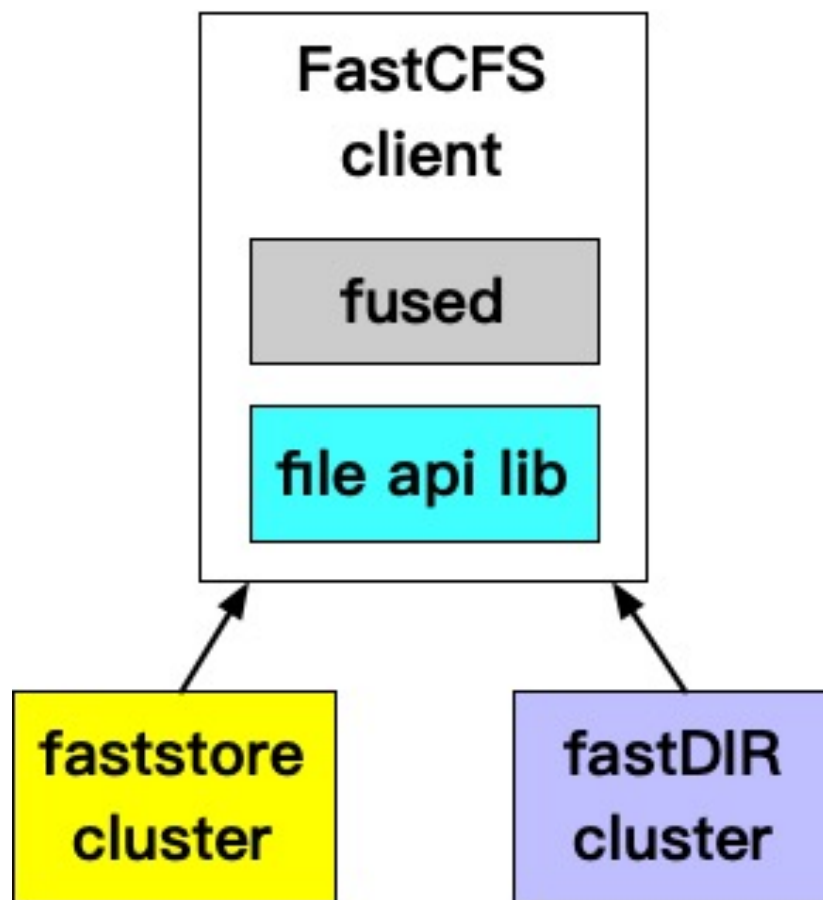
分布式文件系统FastCFS简介

- 基于块存储的通用分布式文件系统
- 可以作为数据库、K8s和KVM等虚拟机的后端存储
- 2020年12月份推出V1.0，2021年4月份推出V2.0
- C语言原生实现，代码行数约12万行

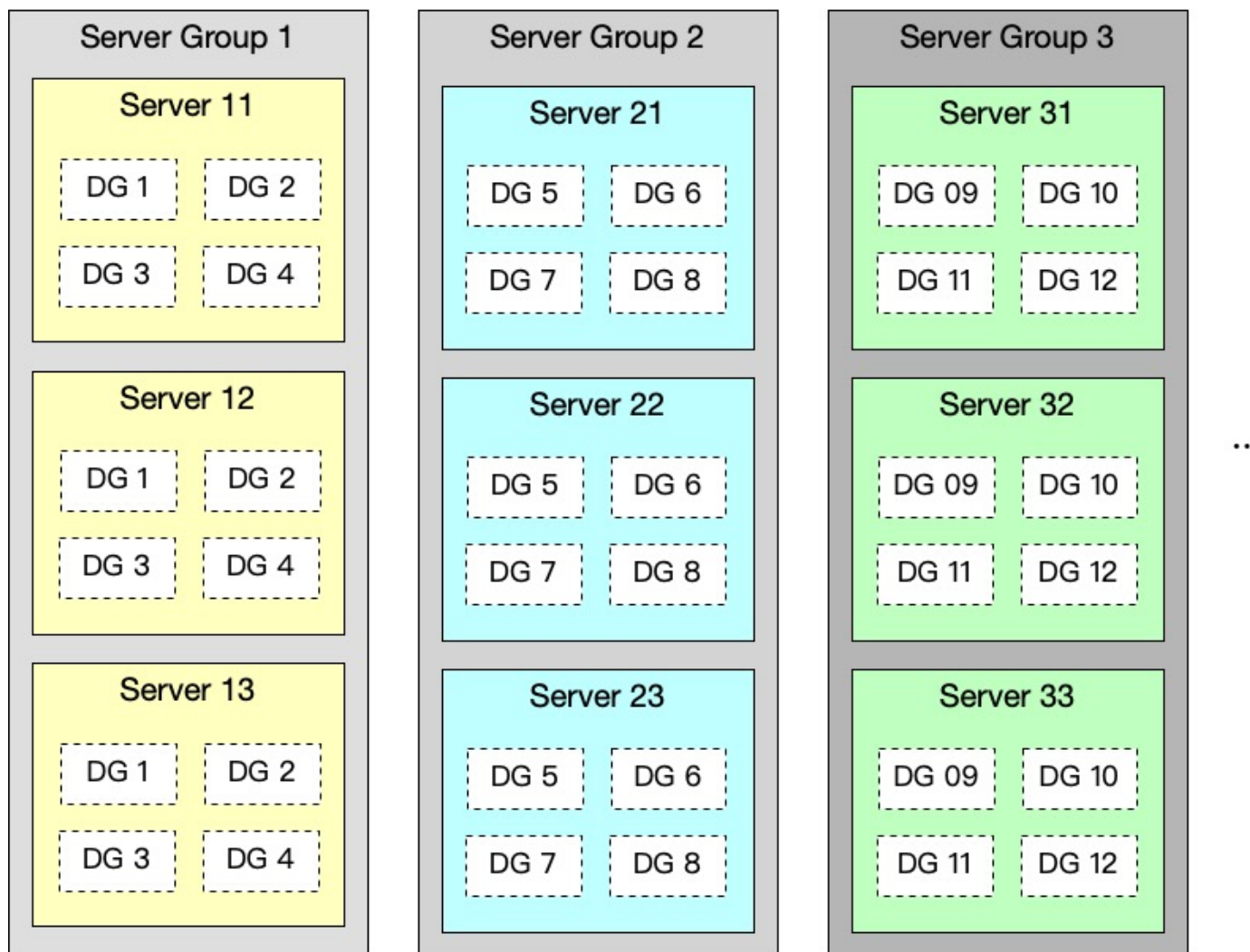
FastCFS核心组件简介

- **FastDIR**：分布式目录服务
- **FastStore**：数据存储服务
- **Auth**：认证中心
- **FUSE Client**：文件系统挂载

FastCFS核心组件关系图



FastStore架构图



FastCFS架构特点

- 数据存储无中心设计
- 分组方式
 - 服务器分组
 - 数据分组

FastCFS如何做到数据一致性

- 一写多读模型
- 通过binlog实现元数据持久化
- 远程调用幂等机制保证

FastCFS采用简单高效的同步复制模式，保证数据强一致

FastCFS数据同步复制做法

- 维持精准的集群状态，ACTIVE方可提供在线服务
- slave为ACTIVE状态，master同步调用slave
- slave掉线后，只有历史数据追加完成后，方可切换为ACTIVE状态

binlog机制

- 数据版本号
- 对账机制
 - ✓ 多个副本binlog对账机制
 - ✓ 单机两套binlog事务保证
- 落盘时机

FastCFS与Ceph性能对比测试—单节点

读写方式	并发数	IOPS (4KB Block)		比值
		FastCFS	Ceph	
顺序写	4	39,680	10,496	378%
	8	48,896	14,028	349%
	16	38,400	16,537	232%
随机写	4	16,281	7,782	209%
	8	26,624	7,884	338%
	16	34,816	8,345	417%
顺序读	4	19,430	14,387	135%
	8	31,744	17,792	178%
	16	40,704	21,017	194%
随机读	4	12,211	10,905	112%
	8	18,329	13,107	140%
	16	26,368	15,923	166%

FastCFS与Ceph性能对比测试—3节点

读写方式	并发数	IOPS (4KB Block)		比值
		FastCFS	Ceph	
顺序写	4	32,256	5,120	630%
	8	55,296	8,371	661%
	16	76,800	11,571	664%
随机写	4	6,374	4,454	143%
	8	11,264	6,400	176%
	16	16,870	7,091	238%
顺序读	4	14,771	14,848	99%
	8	28,672	24,883	115%
	16	50,944	38,912	131%
随机读	4	11,110	12,160	91%
	8	20,966	22,220	94%
	16	35,072	36,608	96%

FastCFS和Ceph机制差异

- 数据存储方式
 - FastCFS : EXT4
 - Ceph : 裸设备
- 文件挂载方式
 - FastCFS : FUSE
 - Ceph : VFS

FastCFS如何做到高性能

- 架构
- 实现

FastCFS软件本身不会成为性能瓶颈！

简单高效的架构方案

- 文件分块存储，采用无中心架构
- 比RAFT更简单高效的数据复制方案
- 顺序写盘方式，确保高效的写入性能，解决了写放大问题

简洁高效的原生实现

- 异步IO模型
- 简洁高效的代码实现，不依赖第三方组件：简单高效的数据结构、无锁编程、线程模型、对象池、连接池、线程池、智能指针。。。

FastCFS极致性能TODO

- 支持裸设备
- 采用io_uring, 优化本机IO
- 支持RDMA, 优化网络IO
- VFS接口
- 碎片整理
- 分级存储

请你思考

FastCFS是如何保证数据一致性前提下做到极高性能的？
请用一个关键字概括。

FastCFS微信公众号



A futuristic, blue-toned wireframe cityscape. The scene is composed of various rectangular blocks and structures of different heights, creating a sense of depth and perspective. A prominent diagonal line cuts across the upper left portion of the image. In the center, the word "THANKS" is displayed in large, white, sans-serif capital letters. A bright, horizontal lens flare effect passes through the middle of the text. In the background, a building-like structure features the letters "SACC" in a stylized font. The overall aesthetic is high-tech and digital.

THANKS