《大规模分布式存储系统》

图书基本信息

书名：《大规模分布式存储系统》

13位ISBN编号：9787111430520

10位ISBN编号：7111430522

出版时间：2013-9-1

出版社：机械工业出版社

作者：杨传辉

页数：293

《大规模分布式存储系统》

内容概要

《大规模分布式存储系统：原理解析与架构实战》是分布式系统领域的经典著作，由阿里巴巴高级技术专家“ 阿里日照 ”（OceanBase核心开发人员）撰写，阳振坤、章文嵩、杨卫华、汪源、余锋（褚霸）、赖春波等来自阿里、新浪、网易和百度的资深技术专家联袂推荐。理论方面，不仅讲解了大规模分布式存储系统的核心技术和基本原理，而且对谷歌、亚马逊、微软和阿里巴巴等国际型大互联网公司的大规模分布式存储系统进行了分析；实战方面，首先通过对阿里巴巴的分布式数据库OceanBase的实现细节的深入剖析完整地展示了大规模分布式存储系统的架构与设计过程，然后讲解了大规模分布式存储技术在云计算和大数据领域的实践与应用。

《大规模分布式存储系统：原理解析与架构实战》内容分为四个部分：基础篇——分布式存储系统的基础知识，包含单机存储系统的知识，如数据模型、事务与并发控制、故障恢复、存储引擎、压缩/解压缩等；分布式系统的数据分布、复制、一致性、容错、可扩展性等。范型篇——介绍谷歌、亚马逊、微软、阿里巴巴等著名互联网公司的大规模分布式存储系统架构，涉及分布式文件系统、分布式键值系统、分布式表格系统以及分布式数据库技术等。实践篇——以阿里巴巴的分布式数据库OceanBase为例，详细介绍分布式数据库内部实现，以及实践过程中的经验。专题篇——介绍分布式系统的主要应用：云存储和大数据，这些是近年来的热门领域，本书介绍了云存储平台、技术与安全，以及大数据的概念、流式计算、实时分析等。

《大规模分布式存储系统》

作者简介

杨传辉，阿里巴巴高级技术专家，花名日照，OceanBase核心开发人员，对分布式系统的理论和工程实践有深刻理解。曾在百度作为核心成员参与类MapReduce系统、类Bigtable系统和百度分布式消息队列等底层基础设施架构工作。热衷于分布式存储和计算系统设计，乐于分享，有技术博客NosqlNotes。

《大规模分布式存储系统》

书籍目录

《大规模分布式存储系统》

精彩短评

1、对几个主流的简单介绍，适合入门看下
2、第一篇章写的还是太过精简，要理解其中的理论需要从其他地方花好些功夫。至于其他篇章建议要把第一篇章内的知识学习完善。第三篇相当于把OceanBase的原理都讲了一遍。总的来说质量还是非常高的。
3、作者想表达的很多，篇幅又很短，导致什么也没说透，尤其是专题篇，真的很水...
4、前面5章感觉还算不错。后面感觉有很多重复，而且作为一个没什么分布式存储经验的人。。感觉有点看不懂。。。OceanBase前面介绍的还挺清晰，但到后面就不想看了。。。跳过了很多细节，就当作开阔技术视野吧。。
5、浏览了一遍
6、分布式资料不多，想这种原理产品结合一起讲解的书籍不多，作为入门资料很不错
7、讲的挺全，细节还是比较少。思路讲的不多。
8、2014年读的。扩展了下眼界，没有实践完全是空谈。
9、从宏观层面了解一下分布式存储系统，ps：参考资料挺棒的
10、粗略读了一遍，开阔一下视野。。
11、各方面都讲了一些，用来扫盲足够了。
12、看了众多评论和推荐后，冲着淘宝的名头去的，花了一周时间断断续续看完。感觉是一本分布式存储方面介绍性质的入门书籍，为更深入的学习提供一个指南。书中以OceanBase做为实践进行了详细介绍，展现了一个分布式数据库的基本原理及设计中的各种决策，为有志进行这方面开发的人员提供了参考，值得一读。
13、偏理论的一本书
14、还是太精炼了
15、应该说，作者本身还是有水平的，只是水平还不到写一本书的程度。本书涉及面广，而且作者本身实践水平和经验都很不错，是专家；但就本书而言，我觉得作为一个对分布式系统架构的概览，是很不错的，可惜每一章节，尤其是不是淘宝自己开发的技术时，写得就不太好，给人感觉是没有理解到位就写了这些章节。当然，我相信作者自身应该是对BIGTABLE这些原理比较了解的，但可能还没有到能写一本好书的程度。一句话：很多理论讲解得有歧义，或者模棱两可，没有论述清楚，感觉像是敷衍。
16、由于工作需要，平常看 OS 和存储、分布式方面的书不少；这本算是少有的国人作品，从 Google 老三篇开始讲，最大篇幅是阿里自己的 oceanbase，从架构说最后这个没啥新颖的，但是这样的大型系统最终实现，而且稳定运行并替换掉 oracle，可见工程 vs 学术是不同的，亚马逊的分布式键值对系统就非常学术化，让人仿佛回到了读书写论文的年代
17、很一般，泛泛而谈，都是讲个概念，这样都能出书，也是醉了。。。
18、师兄写的书，粗看了一遍还是有非常多干货的！
19、入个门
20、把分布式存储的实现和理论都讲了一遍，干货不少。
21、分布式系统导读，结合GFS/Bigtable/MapReduce论文看可以加深理解，很多萃取出经验可以借鉴，写的很不错了……
22、不够深入吧
23、分布式入门了解挺好的选择
24、在分布式书籍如此少的今天，一本不错的总结书！
25、不算是入门书籍，有一些相关经验后，再来阅读事半功倍哦
26、看的出来作者理解的挺透彻的，国人作者额外加分
27、内容比较系统
28、详细的说了OB，如果对OB的实现感兴趣，强烈推荐读。
29、对于我来说，后面的部分没有具体实践，理解起来有点儿费劲。
30、挺不错
31、作为门外汉的我还是学到些概念性东西。对于系统的具体实现，作者只是流于于表面，并没有太

多的细节分析。
对于收藏夹的问题，不是很明白为什么是跨表join，一次请求又不需要显示多个用户的收藏商品，不应该是条简单的 select 语句吗，传入用户ID即可得到收藏条目？
对于这种基准数据+ 增量的实现方式，真的时候于通用的sql场景吗？感觉效率很低下的说。
32、各大厂的系统大杂烩，一半篇幅在推OceanBase，甚至连code review和师兄带师弟也写，不知道怎么想的。话说16年双十一之后没见OceanBase出来吹啊，这是内部被淘汰了？
33、讲得比较通俗易懂，可以作为简单了解分布式存储的一些坑，深度还是不足
34、偏分布式存储导论性质的书
35、本书中关于分布式一些基础理论的表述不够准确，甚至有部分内容是错误的
36、2015.1.5读完 看完这本书才真正明白搞存储到底要搞什么 类似于分布式启蒙吧 要想真的搞通这块还得刷论文 码代码 另明白了 nosql并不是万能药 关系数据库理论还是存储的基础 自家的kv系统代码不到2000行 阿里的ob有70w行 不足之处就像有书评提到的 这本书就像是一半分布式存储理论收集 另一半是ob指南 拼凑感太强
37、虽然是有点混乱 长达两个月终于完成了新年看完一本专业书的愿望，奖励自己一朵小红花
38、走马观花的了解了一番各个厂商的分布式存储服务的架构、特色等信息，没有太具体的知识=。=还是可以看看的（逃
39、感觉总结的很好。
40、对新进入这个领域的人比较合适，整体把握「分布式存储领域」的知识结构
41、分布式，大数据这些技术以后应该会成为主流技术.后面云存储，大数据随便翻了翻
42、一本不太容易读懂的入门书籍，可以认识到分布式存储领域的边界，它们是什么；作者还总结了他们开发OceanBase过程中的体会到的敏捷开发、团队管理、公共平台建设经验。
43、书对各个著名的分布式系统都做了详细的介绍，开阔了眼界，非常适合分布式存储系统的入门。但是对具体如何实现一个系统没有做介绍。
44、需要承认作者很NB，但是说来说去都是理论的东西，各种理论的名词，完全不见真正的实现方案，看着看着就想睡觉的那种！！！
45、大学课本级别的。理论+案例。读起来有点累。。。坚持了几个月，实在无法追原理性的东西。。。打算放放再看。。
46、内存充实，结构系统，值得一读再读
47、浅出
48、一半理论一半实践，内容也比较充实。
49、核心原理清晰，对google的论文和实现分析的不错，分布式的核心还是工程能力，Google太强，抽象的层次也深。
50、20150807开始读。

《大规模分布式存储系统》

精彩书评

1、我两年前开始接触分布式相关的技术，但无奈分布式涵盖范围太广，分布式存储、分布式计算、CAP理论、什么GFS、Hadoop、Dynamo、hive等等，不下点功夫还真不能理顺它们之间的内在关系，特别是容易陷入到各种开源的框架中而无法自拔。本书相当清晰的给出了各个热门技术之间的关系，从单机存储系统降到分布式系统，这其中所涉及到的技术也都一一列出，然后讲述了各种范型，分布式文件系统由哪些，分布式键值系统有哪些，分布式表格系统有哪些，分布式数据库又包含哪些。不过在范型篇中，作者并没有下太多的功夫，对每个系统并没有系统的讲述，很多流行在网络中的描述在本书都能找到，可以理解，要完全搞懂这些并能讲述清楚，不吃不喝几年也是难以完成的。非常感谢这本书...
2、近期看的最慢的一本, 都是基础知识. 精彩摘要笔记：http://liguanglei.name/blogs/2014/03/08/large-scale-distributed-storage-system/
3、在这书里,作者刀枪剑戟斧钺钩叉随手舞来，天文地理吃喝嫖赌样样通透。。从书最开始对网络,存量,运行时间的估算就意思到,有严谨的态度才能做出合格的系统。离这样的架构师水平有好远的路要爬。分布式可大可小,但要做到像书里介绍的那些商业化长度,又有好远的路要走.该书即使不去配合动手,你也可以吸取很多营养,各种架构的设计,他们之间的无状态的协议设计思维.各种增大增粗提升战斗力的要素方向.看了就会对你的工作产生潜移默化的帮助哎呀哇，纸上得来终觉浅，绝知此事要躬行。相比同时期前阿里员工的一本算科普,这本算是深入..没怎么浅出啊.额，其实我一直对我没办法理解怎么解析sql语句而耿耿于怀.还有,能有篇幅介绍些常见设计的坑最好了,让书里多些刀光剑影，明枪暗箭,情仇怨念更精彩.再怨念一次,阿里c/c++系能变成主流吗
4、这本书是目前互联网分布式存储技术的全景图，其中有2点对我特别受启发。第1点，引用原文”Google的分布式存储系统一步步地从Bigtable到Megastore，再到Spanner，这也验证了分布式技术和传统关系数据库技术融合的必然性，即底层通过分布式技术实现可扩展性，上层通过关系数据库的模型和接口将系统的功能暴露给用户。“。或许，这是有划时代意义的讨论。第2点，引用原文”简单就是美。系统开发过程中，如果某个方案很复杂，一般是实践者没有想清楚。“。这一点其实，在很早以前就有听过，但作者通过开发复杂的分布式存储系统过程中得出这么宝贵经验，是”简单就是美“最好的注解。
5、看得出，作者水平还是非常强的，应该对很多开源的产品都深入研究过，也读过不少论文，就这一点，就可以推荐一下。很多原理性的东西，其实网上都有，大家更想看到的是他们在alibaba是如何应用的，有哪些优缺点，平时应用中遇到了哪些坑，呵呵。有一个建议，第一章提出的一些问题，建议在最后一章都给一些标准答案，呵呵，这样看完后可以检验一下是否自己理解了。现在稍大的公司都在做这个，或许不是分布式关系数据库，但分布式存储应用现在是越来越多了。
6、这本书有理论介绍也有实践经验，还算不错，同时支持下国内的原创作者，给4星。看得出作者有多年的分布式系统开发经验，对Google，FB，Amazon的各个分布式系统的特点娓娓道来。前半部分的基础+范型篇还是能学到不少，特别适合初学者。不过，这本书有一半的内容是介绍OceanBase的，感觉像是OceanBase的说明书。当然这和作者的工作有关系。虽然OB用不上，不过了解下思想也是好的。最喜欢的第11章：质量保证、运维及实践的11.4对于系统发展路径的介绍以及人员成长方面，都是干货啊。
7、北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐.北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐北京知名公司招聘分布式存储专家薪资50-100万。欢迎推荐自荐联系方式：18612377036 qq:2246684979

《大规模分布式存储系统》

章节试读

1、《大规模分布式存储系统》的笔记-第53页

为了进行选主或者维护系统中重要的全局信息，可以维护一套通过Paxos协议实现的分布式锁服务，比如Google Chubby或者它的开源实现Apache Zookeeper。

2、《大规模分布式存储系统》的笔记-第1页

前言
以前是纵向扩展，PC不行换小型机，小型机不行换中型机，现在是横向扩展，加入普通PC机器就够了。
云存储和大数据构建在分布式存储之上。

1 概述
分布式存储系统是大量普通PC服务器通过Internet互联，对外作为一个整体提供存储服务。
可扩展、低成本、高性能、易用。
数据分为三类：（1）非结构化数据，如文档、文本、图片、音视频；（2）结构化数据，譬如存储在关系型数据库中的数据，数据的模式跟内容是分开的；（3）半结构化数据，如HTML文档，跟结构化数据的不同是，模式跟内容是混合在一起的，不需要预先定义模式。
分布式存储系统分为四类：（1）分布式文件系统，如GFS；（2）分布式键值系统，如Amazon Dynamo；（3）分布式表格系统；（4）分布式数据库；

2 单机存储系统
单机存储引擎就是哈希表、B树等数据结构在机械磁盘、SSD等持久化介质上的实现。
数据库将一个或多个操作组成一个组，称作事务，事务必须满足原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability），简称为ACID特性。

哈希存储引擎

B树存储引擎
缓存区管理
LRU和它的改进版LIRS
LRU是Least Recent Used
LIRS是Low Inter-reference Recency Set，是为了避免像全表扫描这类操作，根据LRU算法会把热数据清理出内存。它将缓冲池分为两级，数据首先进入第一级，如果在较短时间内被访问过两次或以上，则进入第二级，每一级内部还是采用LRU替换算法。（InnoDB）

LSM树存储引擎
LSM树（Log Structured Merge Tree）的思想非常朴素，就是将对数据的修改增量保持在内存中，达到指定的大小限制后将这些修改操作批量写入磁盘，读取时需要合并磁盘中的历史数据和内存中最近的修改操作。（LevelDB）
LSM树的优势在于有效的规避了磁盘随机写入问题，但读取时可能需要访问较多的磁盘文件。

数据模型
如果说存储引擎相当于存储系统的发动机，那么，数据模型就是存储系统的外壳。存储系统的数据模型主要包括三类：文件、关系以及随着NoSQL技术流行起来的键值模型。

3、分布式系统
数据一致性：强一致性、弱一致性、最终一致性
数据分布：一致性hash

CAP理论

Paxos和2PC协议。
2PC协议保证多个数据分片上操作的原子性；Paxos协议实现同一个数据分片的多个副本之间的一致性。
（MongoDB就是使用Paxos协议在故障时来选举primary）

4、分布式文件系统
GFS采用数据流跟控制流分离的方法，从而能够基于网络拓扑结构更好地调度数据流的传输。
GFS主要是为追加而不是改写而设计的。一方面是因为改写的需求比较少，或者可以通过追加来实现，比如可以只使用GFS的追加功能构建分布式表格系统Bigtable；另一方面是因为追加的一致性模型相比改写要更加简单有效。

5、分布式键值系统
6、分布式表格系统
7、分布式数据库
8、OceanBase 架构初探
OceanBase可以划分为4个模块：主控服务器RootServer、更新服务器UpdateServer、基线数据服务器ChunkServer以及合并服务器MergeServer。
RootServer：集群管理、数据分布、副本管理；
MergeServer：协议解析、SQL解析、请求转发、结果合并、多表操作；
ChunkServer：存储多个子表，提供读取服务，执行定期合并以及数据分发；
UpdateServer：是集群中唯一能接受写入的模块，每个集群中只有一个主UpdateServer；

数据存储在ChunkServer中，用户的数据写入、更新首先都记录在UpdateServer的内存中，当达到一定大小之后可以持久化到SSD。数据读取需要ChunkServer数据跟UpdateServer数据结合才能返回给客户。
OceanBase集群会执行定期合并和数据分发，定期合并指的是将UpdateServer中的冻结内存表跟ChunkServer的SSTable合并形成新的SSTable。数据分发指的是将UpdateServer上冻结内存表的更新数据缓存到本地。
定期合并一般安排在业务低峰期，每日合并。
与定期合并不同的是，数据分发只是将UpdateServer冻结的数据缓存到ChunkServer，并不会生成新的SSTable文件。

书中很多个地方提到了多路归并，这个算法思路在大数据量计算上非常常见。

11、质量保证、运维及实践
这是全书中，最容易读懂的内容，所讲述的内容，我也有共鸣。
开发流程：Code Review、单元测试、压力测试、QA测试、灰度上线。
公共平台推广：生存、口碑、收集需求优化。
重视测试，相当于TDD；吃自己的狗粮。
12、云计算
13、大数据
一些概念的了解认识。

读完此书，对很多理论有了解，书中前部分介绍各种系统入门，中间部分介绍OceanBase，最后再介绍

云计算、大数据。那些实战经验分享相当不错。

3、《大规模分布式存储系统》的笔记-第54页

分布式存储系统中往往有一个总控节点用于维护数据分布信息，执行工作机管理，数据定位，故障检测和恢复，负载均衡等全局调度工作。通过引入总控节点，可以使得系统的设计更加简单，并且更加容易做到强一致性，对用户友好。那么，总控节点是否会成为性能瓶颈呢？

4、《大规模分布式存储系统》的笔记-第97页

Google Bigtable是分布式表格系统的始祖，它采用双层结构，底层采用GFS作为持久化存储层。GFS + Bigtable双层架构是一种里程碑式的架构，其他系统，包括Microsoft分布式存储系统Windows Azure Storage以及开源的Hadoop系统，均为其模仿者。Bigtable的问题在于对外接口不够丰富，因此，Google后续开发了两套系统，一套是Megastore，构建在Bigtable之上，提供更加丰富的使用接口；另外一套是Spanner，支持跨多个数据中心的数据库事务。

5、《大规模分布式存储系统》的笔记-第57页

由于拷贝数据的过程中存储节点再次发生故障的概率很高，所以这样的架构很难做到自动化，不适用大规模分布式存储系统。
大规模分布式存储系统要求具有线性可扩展性，即随时加入或者删除一个或者多个存储节点，系统的处理能力与存储节点的个数成线性关系。为了实现线性可扩展性，存储系统的存储节点之间是异构的。异构系统将数据划分为很多大小接近的分片，每个分片的多个副本可以分布到集群中的任何一个存储节点。如果某个节点发生故障，原有的服务将由整个集群而不是某几个固定的存储节点来恢复。

6、《大规模分布式存储系统》的笔记-第36页

服务器宕机：引发服务器宕机的原因可能是内存错误、服务器停电等。服务器宕机可能随时发生，当发生宕机时，节点无法正常工作，称为“不可用”（unavailable）。服务器重启后，节点将失去所有的内存信息。因此，设计存储系统时需要考虑如果通过读取持久化介质（如机械磁盘，固态硬盘）中的数据来恢复内存信息，从而恢复到宕机前的某个一致状态。进程运行过程中也可能随时因为core dump等原因退出，和服务器宕机一样，进程重启后也需要恢复内存信息。

7、《大规模分布式存储系统》的笔记-质量保证、运维及实践

通用分布式存储系统不是设计出来的，而是随着应用需求不断发展起来的。它来源于具体业务，又具有一定通用性，能够解决一大类问题。通用分布式存储平台的优势在于规模效应，等到平台的规模超过某个平衡点时，成本优势将会显现。

通用分布式存储平台主要有两种成长模式：

1. 公司高层制定战略大力发展通用平台。这种模式前期发展会比较顺利，但是往往会因为离业务太远而在中期暴露大量平台本身的问题。

2. 来源于具体业务并将业务需求通用化。这种模式会面临更大的技术挑战，但是团队成员反而能在这个过程中得到更多的锻炼。

8、《大规模分布式存储系统》的笔记-第93页

Dynamo采用无中心节点的P2P设计，增加了系统可扩展性，但同时带来了一致性问题，影响上层应用。
Dynamo及其开源实现Cassandra在实践中受到的关注逐步减少，无中心节点的设计短期之内难以成为主流，另一方面，Dynamo综合使用了各种分布式技术，在实践过程中可以选择性借鉴。

9、《大规模分布式存储系统》的笔记-第9页

由于同一个同一个接入层的服务器往往部署在一个机架内，因此，设计系统的时候需要考虑服务器是否在一个机架内，减少跨机架拷贝大量数据。例如，Hadoop HDFS默认存储三个副本，其中两个副本放在一个机架，就是这个原因。

10、《大规模分布式存储系统》的笔记-第67页

主控服务器中维护了系统的元数据，包括文件及chunk命名空间、文件到chunk之间的映射、chunk位置信息。它也负责整个系统的全局控制，如chunk租约管理、垃圾回收无用chunk、chunk复制等。主控服务器会定期与CS通过心跳的方式交换信息

11、《大规模分布式存储系统》的笔记-第27页

为什麼需要先写操作日志再修改内存中的数据呢？假如先修改内存中的数据，那么用户就能立刻读到修改后的结果，一旦在完成内存修改与写入日志之间发生故障，那么最近的修改操作无法恢复。然后，之前的用户可能已经读取了修改后的结果，这就会产生不一致的情况。

12、《大规模分布式存储系统》的笔记-第17页

如果说存储引擎相当于存储系统的发动机，那么，数据模型就是存储系统的外壳。存储系统的数据模型主要包括三类：文件、关系以及随着NoSQL技术流行起来的键值模型。

13、《大规模分布式存储系统》的笔记-第74页

ChunkServer是一个磁盘和网络IO密集型应用，为了最大限度地发挥机器性能，需要能够做到将磁盘和网络操作异步化，但这会增加代码实现的难度。

14、《大规模分布式存储系统》的笔记-第21页

由于互联网业务中读事务比例往往远高于写事务，为了提高读事务性能，可以采用写时复制（Copy-On-Write,COW）或者多版本并发控制（Multi-Version Concurrency Control，MVCC）技术来避免写事务阻塞读事务

15、《大规模分布式存储系统》的笔记-第98页

Bigtable构建在GFS之上，为文件系统增加一层分布式索引层。另外，Bigtable依赖Google的Chubby(即分布式锁服务)进行服务器选举及全局信息维护。

16、《大规模分布式存储系统》的笔记-第73页

Master定时检查，如果发现文件删除超过一段时间（默认为3天，可配置），那么它会把文件从内存元数据中删除，以后ChunkServer和Master元数据中已经不存在的chunk信息，这时，ChunkServer会释放这些chunk副本。为了减轻系统的负载，垃圾回收一般在服务低峰期执行，比如每天晚上凌晨1:00

开始。

17、《大规模分布式存储系统》的笔记-第6页

哈希存储引擎是哈希表的持久化实现，B树存储引擎是B树的持久化实现，而LSM树（Log Structure Merge Tree）存储引擎采用批量转储技术来避免磁盘随机写入。

18、《大规模分布式存储系统》的笔记-第66页

分布式文件系统的主要功能有两个：一个是存储文档、图像、视频之类的Blob类型数据；另外一个是作为分布式表格系统的持久化层。
GFS内部将大文件划分为大小约为64MB的数据块（chunk），并通过主控服务器（Master）实现元数据管理、副本管理、自动负载均衡等操作。

19、《大规模分布式存储系统》的笔记-第51页

容错处理的第一步是故障检测，心跳是一种很自然的想法。

20、《大规模分布式存储系统》的笔记-第93页

Tair作为一个分布式系统，是由一个中心控制节点和若干个服务节点组成。其中，中心控制节点称为Config Server，服务节点称为Data Server。Config Server负责管理所有的Data Server，维护其状态信息；Data Server对外提供各种数据服务，并以心跳的形式将自身状况汇报给Config Server。Config Server是控制节点，而且是单点，目前采用一主一备的形式来保证可靠性，所有的Data Server地位都是等价的。

21、《大规模分布式存储系统》的笔记-第1页

分布式存储涉及的技术主要来自两个领域：分布式系统以及数据库。

22、《大规模分布式存储系统》的笔记-第36页

分布式系统中有两个重要的协议，包括Paxos选举协议以及两阶段提交协议。Paxos协议用于多个节点之间达成一致，往往用于实现总控节点选择。两阶段提交协议用于保证跨多个节点操作的原子性，这些操作要么全部成功，要么全部失败。理解这两个分布式协议之后，学习其他分布式协议会变得相当容易。

23、《大规模分布式存储系统》的笔记-第25页

最后2行是不是矛盾了？最后一行应该仅仅说的是修改版本号吧

24、《大规模分布式存储系统》的笔记-第86页

Tair是淘宝网开发的分布式键值系统，它借鉴了Dynamo系统的一些设计思路并做了一些创新，其中最大的变化就是从P2P架构修改为带有中心节点的架构，笔者认为，这种思路在大方向上是正确的。

25、《大规模分布式存储系统》的笔记-第1页

《大规模分布式存储系统》

分布式存储系统的挑战主要在于数据、状态信息的持久化，要求在自动迁移、自动容错、并发读写的过程中保证数据的一致性。

26、《大规模分布式存储系统》的笔记-第1页

单机环境下的函数调用常常可以在微秒级内返回，所以除了少数访问外部设备（例如磁盘、网卡等）的函数采用异步方式调用外，大部分函数采用同步调用的方式，编译器和操作系统在调用前后自动保存与恢复程序的上下文；在分布式环境下，计算机之间的函数调用（远程调用，即RPC）的返回时间通常是毫秒或亚毫秒（0.1～1.0毫秒）级，差不多单机环境的100倍，使用同步方式远远不能发挥现代CPU处理器的性能，所以分布式环境下的RPC通常采用异步调用方式，程序需要自己保存和恢复调用前后的上下文，并需要处理更多的异常。

27、《大规模分布式存储系统》的笔记-第1页

分布式存储技术是互联网后端架构的‘九阳神功’，万变不离其宗，云存储的核心还是后端的大规模分布式存储系统。

28、《大规模分布式存储系统》的笔记-第42页

分布式系统区别于传统单机系统在于能够将数据分布到多个节点，并在多个节点之间实现负载均衡。数据分布的方式主要有两种：一种是哈希分布，如一致性哈希，代表系统有Amazon的Dynamo系统；另外一种方式是顺序分布，即每张表格上的数据按照主键整体有序，代表系统为Google的Bigtable系统。
将数据分散到多台机器后，需要尽量保证多台机器之间的负载是比较均衡的。
衡量机器负载涉及的因素很多，如机器Load值、CPU、内存，磁盘以及网络等资源使用情况，读写请求数及请求量，等等，分布式存系统需要能自动识别负载高的节点，当某台机器的负载较高时，将它服务的部分数据迁移到其他机器，实现自动负载均衡。
分布式存储系统的一个基本要求就是透明性，包括数据分布透明性、数据迁移透明性，数据复制透明性，故障处理透明性。

29、《大规模分布式存储系统》的笔记-第97页

Google Bigtable是分布式表格系统的始祖，它采用双层结构，底层采用GFS作为持久化存储层。GFS + Bigtable双层架构是一种里程碑式的架构，其他系统，包括Microsoft分布式存储系统Windows Azure Storage以及开源的Hadoop系统，均为其模仿者。Bigtable的问题在于对外接口不够丰富，因此，Google后续开发了两套系统，一套是Megastore，构建在Bigtable之上，提供更加丰富的使用接口；另外一套是Spanner，支持跨多个数据中心的数据库事务。

30、《大规模分布式存储系统》的笔记-第126页

有很多思路可以实现关系数据库的可扩展性。例如，在应用层划分数据，将不同的数据分片并划分到不同的关系数据库上，如MySQL Sharding；或者在关系数据库内部支持数据自动分片，如Microsoft SQL Azure，或者干脆从存储引擎开始重写愿意个全新的分布式数据库，如Google Spanner以及Alibaba OceanBase。

31、《大规模分布式存储系统》的笔记-第71页

Master容错与传统方法类似，通过操作日志加checkpoint的方式进行，并且有一台成为“Shadow

Master ”的实时热备。
Master上保存了三种元数据信息：
命名空间，也就是整个文件系统的目录结构以及chunk基本信息；
文件到chunk之间的映射；
chunk副本的位置信息，每个chunk通常有三个副本。
GFS Master的修改操作总是先记录操作日志，然后修改内存。

32、《大规模分布式存储系统》的笔记-第37页

磁盘故障是一种发生概率很高的异常。磁盘故障分为两种情况：磁盘损坏和磁盘数据错误。磁盘损坏时，将会丢失存储在上面的数据，因而，分布式存储系统需要考虑将数据存储到多台服务器，即使其中一台服务器磁盘出现故障，也能从其他服务器上恢复数据。

33、《大规模分布式存储系统》的笔记-第2页

分布式存储系统分为四类：分布式文件系统、分布式键值（Key-Value）系统、分布式表格系统和分布式数据库。

34、《大规模分布式存储系统》的笔记-第37页

引发网络异常的原因可能是消息丢失、消息乱序（如采用UDP方式通信）或者网络包数据错误。有一种特殊的网络异常称为“ 网络分区 ”，即集群的所有节点被划分为多个区域，每个区域内部可以正常通信，但是区域之间无法通信。例如，某分布式系统部署在两个数据中心，由于网络调整，导致数据中心之间无法通信，但是数据中心内部可以正常通信。
设计容错系统的一个基本原则是：网络永远是不可靠的，任何一个消息只有收到对方的回复后才可以认为发送成功，系统设计时总是假设网络将会出现异常并采取相应的处理措施。

35、《大规模分布式存储系统》的笔记-第55页

数据库可扩展性实现的手段包括：通过主从复制提高系统的读取能力，通过垂直拆分和水平拆分将数据分布到多个存储节点，通过主从复制将系统扩展到多个数据中心。当主节点出现故障时，可以将服务切换到从节点；另外，当数据库整体服务能力不足时，可以根据业务的特点重新拆分数据进行扩容。

36、《大规模分布式存储系统》的笔记-第75页

TFS架构设计时需要考虑如下两个问题：
1、Metadata信息存储。
由于图片数量巨大，单机存放不了所有的元数据信息，假设每个图片文件的元数据占用100字节，100亿图片的元数据占用的空间为10G x 0.1KB = 1TB，单机无法提供元数据服务。
2、减少图片读取的IO次数。
在普通的Linux文件系统中，读取一个文件包括三次磁盘IO：首先读取目录元数据到内存，其次把文件的inode节点装载到内存，最后读取实际的文件内容。由于小文件个数太多，无法将所有目录及文件的inode信息缓存到内存，因此磁盘IO次数很难达到每个图片读取只需要一次磁盘IO的理想状态。
TFS设计时采用的思路是：多个逻辑图片文件共享一个物理文件。

《大规模分布式存储系统》

版权说明