分布式存储系统可靠性：系统量化估算

vivo互联网技术 发布于 2021-08-02

English

一、引言

我们常常听到衡量分布式存储系统好坏的两个指标：可用性和可靠性指标。

可用性指的是系统服务的可用性。一般按全年可用时间除以全年时间来衡量可用性的好坏，平常我们说的 SLA指标就是可用性指标，这里就不展开细说。

可靠性指标指的是数据的可靠性。我们常说的数据可靠性11个9，在对象存储中就意味着存储一千亿个对象大概会有1个文件是不可读的。由此可见，数据可靠性指标给分布式存储系统带来的挑战不言而喻。

本文就重点来分析一下分布式系统的数据可靠性的量化模型。

二、背景

数据的可靠性是基础的基础，任何数据的丢失都会给企业造成无法计算的损失。

随着数据规模的日益增大，环境更加复杂，我们大体可以把威协数据可靠性的因素归为几大类：

- **硬件故障**：主要是磁盘故障、还有网络故障、服务器故障、IDC故障；
- **软件隐患**：内核BUG，软件设计上的BUG等；
- **运维故障**：人为误操作。

其中，第1类的硬件故障中又以磁盘故障最为频繁，坏盘对于从事分布式存储运维的同学来说再正常不过了。

因此，我们接下来从磁盘故障这个维度来尝试量化一下一个分布式系统的数据可靠性。

三、数据可靠性量化

为了提高数据的可靠性，数据副本技术和EC编码冗余技术是分布式系统可靠性最常用的手段了。以多副本为例，副本数越多，数据的可靠性肯定越高。

为了对分布式系统的数据可靠性作一个量化估算，进一步分析得到影响存储数据可靠性的因素主要有：

- **N**：分布式系统磁盘的总数，可以很直观理解，磁盘的数量是和可靠性强相关，N的大小与数据的打散程度有很大关系。
- **R**：副本数，副本数越高数据的可靠性肯定越高，但同时也会带来更大的存储成本。
- **T**：RecoveryTime出现坏盘情况下数据恢复的时间，这个也很好理解，恢复时间越短，数据的可靠性越高。
- **AFR**：Annualized Failure Rate磁盘的年度故障率，这个和磁盘本身的质量相关，质量越好，AFR越低，数据的可靠性越高。
- **S**：CopySet数量，一个盘上的数据的冗余在集群中的打散程度，打得越散，则有可能任意坏3块盘就刚好有数据的冗余数据都丢失。所以，仅从打散程度这个维度看，打散程度越小越好。

因此，我们可以用一个公式表示分布式系统的全年数据可靠性：

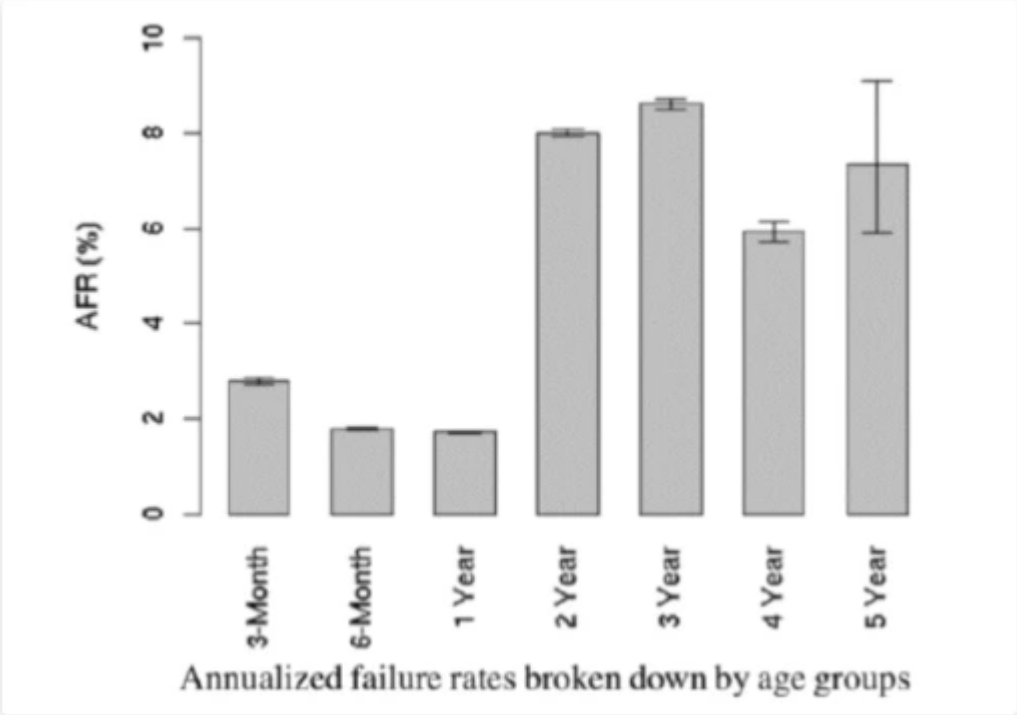
$$P = func(N, R, T, S, AFR)$$

3.1 磁盘年故障率：AFR

障的概率，可以很直观的理解，AFR越低，系统的可靠性越高，因为AFR与系统的数据可靠性强相关；而这个指标通常又是由另一个磁盘质量指标MTBF（Mean Time Before Failure）推算出来，而MTBF各大硬盘厂商都是有出厂指标的，比如说希捷的硬盘出厂的MTBF指标为120W个小时。以下为AFR的计算公式：

$$AFR = \frac{1}{MTBF/(24*365)} * 100$$

但是实际使用当中往往MTBF会低于硬盘出厂指标。Google就根据他们的线上集群的硬盘情况进行了统计计算AFR如下：

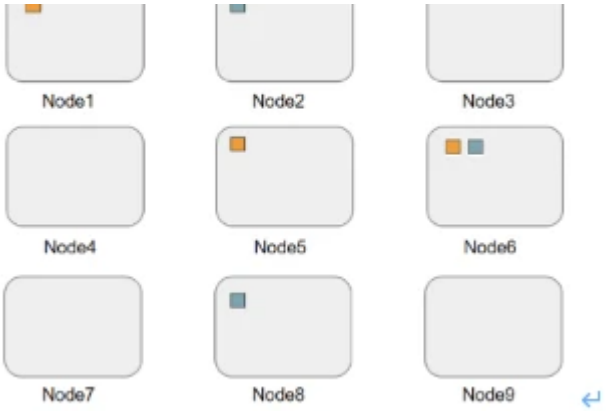


（5年内硬盘AFR统计情况）

（图片来自http://oceanbase.org.cn）

3.2 副本数据复制组：CopySet

副本数据复制组CopySet：用通俗的话说就是，包含一个数据的所有副本的节点，也就是一个copyset损坏的情况下，数据会丢失。



（单个数据随机复制分组示意图）

（图片来自https://www.dazhuanlan.com）

如图2所示，以9块盘为例，这9块盘的copyset就是：{1,5,6}，{2,6,8}，如果不做任何特殊处理，数据多了之后，数据的随机分布如下：

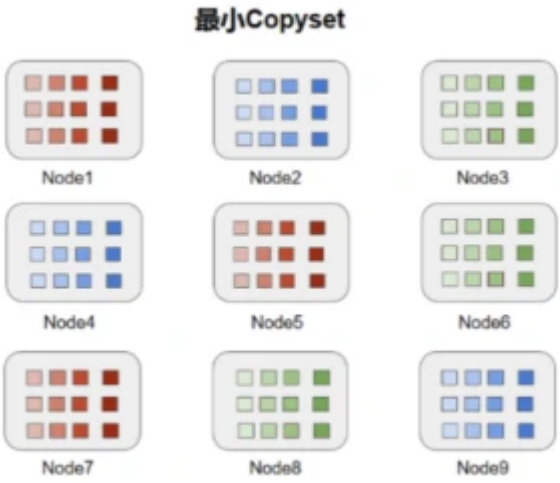
（海量数据随机分布示意图）

（图片来自https://www.dazhuanlan.com）

最大CopySet：如上图所示，12个数据的多副本随机打散到9块盘上，从上图中任决意挑3块盘都可以挑出包含某个数据的三个副本，就相当于从n个元素中取出k个元素的组合数量为：

$$n!$$

最大的CopySet配置下一旦有三块磁盘坏了，丢数据的概率是100%。另外一种情况，数据的分布是有规律的，比如一块盘上的数据只会在另外一块盘上备份，如下图所示，在这种情况下数据覆盖的CopySet只有（1，5，7）、（2，4，9）、（3，6，8）也就是说这种情况下CopySet为3。我们不难理解，9块盘的最小CopySet为3。也就是N/R。



（磁盘粒度冗余分布示意图）

因此，CopySet数量S符合以下：

$$N/R < S < C(N,R)$$

既然CopySet数据可以最小为N/R，能不能把CopySet数量调到最小，答案当然是不行的，因为，一方面如果CopySet调到最小，当有一个盘坏了后，只有其它2块盘进行这块盘的恢复操作，这样数据的恢复时间又变长了，恢复时间变长也会影响数据的可靠性；而且一旦命中了CopySet中的一个，则丢失的数据量规模非常大。因此，分布式系统中的CopySet的量和恢复速度RecoveryTime是一个均衡整个系统数据可靠性和集群可用性的参数。

文献【1】Copysets: Reducing the Frequency of Data Loss in Cloud Storage提供了一种分布式系统的CopySet Replication的选择策略，在分布式存储系统当中比如对象存储和文件存储当中，还有一种方

100G一个文件，8T盘上的最大文件存储数量也就是8T/100G = 80个文件，也就是说一个8T的盘的数据最多打散到了80块其它的盘上，对于集群盘远大于80的系统显然也能够很好的控制一个数据盘的数据打散程度。

因此，在磁盘上的分片是随机打散的情况下，CopySets数量可以量化为以下公式：

$$\min\left(C_R^N, \frac{P*80\%}{B}*\frac{N}{R}\right)$$

其中，P为磁盘的容量，B为分片大小，N为系统磁盘的数据，R为副本数。80%为使用率。

3.3 数据恢复时间：Recovery Time

数据恢复时间对数据可靠性影响很大，这个很好理解，因此缩短数据恢复时间可以有效降低数据丢失的风险。前面已经介绍数据恢复时间和磁盘上数据打散程度强相关，同时数据恢复时间也与服务本身的可用性相关。

比如磁盘带宽为200MB/s，假设留给恢复可用的带宽为20%就是40MB/s，磁盘容量为P，使用率为80%，B为BlockSize大小，则恢复速度可按以下方式计算：

$$\frac{P*80\%}{B}*40MB$$

四、可靠性模型推导

4.1 磁盘故障与泊松分布

泊松分布：泊松分布其实是二项分布的极限情况，泊松分布公式如下：

$$P_n(N(t)=n)=\frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

小时内出故障的磁盘平均数。

从3.1节我们已经介绍过了磁盘一年之内出故障的概率为AFR，那么单位时间1个小时的时间周期磁盘出故障的概率为FIT（Failures in Time）：

$$FIT = \frac{AFR}{24 * 365}$$

那么N块盘的集群在单位时间1小时内出故障的盘的数量为FIT*N，换句话说，也就是单位时间1小时内出故障的磁盘平均数。因此可以得到：

$$\lambda = FIT * N$$

4.2 系统全年可靠性计算推导

由4.1我们得到磁盘故障是符合泊松分布，N块盘的集群中在t小时内有n块盘故障的概率：

$$P_n(\lambda, t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!}$$

接下来我们以3副本为例，来推导一下全年集群没有数据丢失的概率的量化模型，3副本情况下，全年集群没有数据丢失的概率不太好量化，我们可以通过计算全年集群出现数据丢失的概率，然后全年集群没有数据丢失的概率就以计算出来：

全年集群出现数据丢失的概率：只有在t（1年）的时间内有第一块磁盘出现故障之后，然后系统进入数据恢复阶段，在数据恢复的时间tr内又有第二块磁盘出现故障，我们先不考虑数据恢复了多少，然后在tr内又有第三块磁盘出现故障，但是这三个磁盘不一定刚好命中了我们在3.2介绍的copyset复制组如果命中了copyset，那么集群在全年就真的有出现数据丢失了。因为全年集群出现数据丢失的概率和P1，P2，P3，以及Copyset命中概率Pc相关。

$$P = P_1 * P_2 * P_3 * P_C$$

1年时间t内有任意一块磁盘出现故障的概率为：

$n!$

上面这块磁盘出现问题后，需要马上恢复，在恢复时间tr内有另外一块盘出现故障概率：

$$P_2(any\ disk\ failure) = 1 - P_0(\lambda, tr) = 1 - P_0(FIT * (N-1), tr)$$
$$= 1 - \frac{[FIT * (N-1) * tr]^n e^{-FIT*(N-1)*tr}}{n!} = 1 - e^{-FIT*(N-1)*tr}$$

在恢复时间tr内有第三块任意盘出现故障的概率：

$$P_3(any\ disk\ failure) = 1 - P_0(\lambda, tr) = 1 - P_0(FIT * (N-2), tr)$$
$$= 1 - \frac{[FIT * (N-2) * tr]^n e^{-FIT*(N-2)*tr}}{n!} = 1 - e^{-FIT*(N-2)*tr}$$

而这三块出现故障的磁盘刚好命中集群的CopySets的概率为：

$$P_c = \frac{M}{C_R^N} = \frac{\min(C_R^N, \frac{Z*80\%}{B} * \frac{N}{R})}{C_R^N}$$

因此，不难得到全年集群出现数据丢失的概率P：

$$P = (1 - e^{-FIT*N*t}) * \left(1 - e^{-FIT*(N-1)*tr}\right) * \left(1 - e^{-FIT*(N-2)*tr}\right) * \frac{\min(C_R^N, \frac{Z*80\%}{B} * \frac{N}{R})}{C_R^N}$$

然后全年集群不出现数据丢失的概率1-P就可以计算得到了。

4.3 EC冗余全年可靠性计算推导

EC冗余机制相对于三副本机制是用额外的校验块来达到当有一些块出现故障的情况下数据不会丢，按（D,E）数据块进行EC编码，那么在计算EC冗余下的全年集群数据丢失概率的时候，EC模式下的恢复速度tr和三副本肯定是不一样的，另外，EC模式下的copysets是不一样的，EC模式是允许E个数

$$P = (1 - e^{-FIT*N*t}) * \left(1 - e^{-FIT*(N-1)*tr}\right) * \left(1 - e^{-FIT*(N-2)*tr}\right) * \left(1 - e^{-FIT*(N-3)*tr}\right) * \frac{\min\left(C_R^N, \frac{Z*80\%}{B} * \frac{N}{R} * C_E^{D+E}\right)}{C_R^N}$$

相对于三副本模式来说，EC模式的copyset需要考虑在D+E个块当中丢失其中任意E个块，则EC模式下的copyset数为：

$$\frac{\min\left(C_R^N, \frac{Z*80\%}{B} * \frac{N}{R} * C_E^{D+E}\right)}{C_R^N}$$

五、可靠性模型估算

5.1 量化模型影响因素

$$P = (1 - e^{-FIT*N*t}) * \left(1 - e^{-FIT*(N-1)*tr}\right) * \left(1 - e^{-FIT*(N-2)*tr}\right) * \frac{\min\left(C_R^N, \frac{Z*80\%}{B} * \frac{N}{R}\right)}{C_R^N}$$

以三副本为例，从以上量化的全集群出故障的概率计算公式可以得到影响的因素有：

- N：集群的盘的个数；
- FIT：就也是1小时磁盘的故障率,可以由AFR得到；
- t：这个是固定1年；
- tr：恢复时间，单位为小时，和恢复速度W和磁盘存储量、分片大小相关；
- R：副本数；
- Z：磁盘的存储总空间；
- B：分片或者Block的大小，小文件合并成大文件的最大Size。

5.2 可靠性量化计算

因素	N(磁盘数)	AFR	W(MB/s)	Z(GB)	B(GB)	R	P
1	48	0.43	100	8000	8	3	9.862308892280407e-12
2	360	0.43	100	8000	8	3	7.402016613468761e-13
3	804	0.43	100	8000	8	3	1.8437900014856886e-13
4	3600	0.43	100	8000	8	3	1.9037353614085208e-13
5	804	0.43	10	8000	8	3	1.8239161955909077e-08
6	3600	0.43	10	8000	8	3	9.393333484259648e-10
7	3600	1.2	100	8000	8	3	7.3041085675777976e-09
8	3600	7	100	8000	8	3	2.456394305287295e-07
9	100	0.43	100	8000	80	3	6.587036362894664e-13
10	100	0.43	100	8000	100	3	8.233773336596573e-13
11	10000	0.43	100	8000	1	3	2.1406966358279285e-11
12	10000	0.43	100	8000	8	3	2.6758707947849106e-12

表1：不同Case下参数影响可靠性指标结果

结合4.2的磁盘故障与可靠性的推导，通过表格中10个case的计算，可以看到：

Case 1,2,3通过扩展磁盘的数量从48块盘到804再到3600块盘，可靠性从11个9提高到接近13个9，然后804块盘到3600块盘还是维护在13个9，按理说，集群的规模增大，增3块盘的概率会提高，但是由于恢复速度也随着磁盘的增加而线性增加，因此，可靠性一直在提升，而从804到3600块盘，可靠性没有增加，是因为这时候恢复速度已经不随磁盘增加而线性增大，因为在磁盘量很大后，决定恢

Case 5,6比较好理解，恢复速度由100M/S变为10M/S，可靠性降低2个以上数量级；

Case 7,8也比较好理解，AFR由0.43提高到1.2再提高到7，可靠性降低了3个数量级；

Case 9,10比较绕，磁盘数在100的情况下，Block大小由80G一个提高到100G一个，可靠性降低了，这种情况下是因为恢复速度提高，CopySet也提高，但速度影响更大导致。

Case 11,12也比较绕，由于我们限定了恢复速度不能超过5分钟（模拟线上，因为系统检测坏盘，自动踢盘等操作也需要时间），这两个Case下的CopySet都超级大，所以恢复的并发度都非常高，但受限于5分钟限定，所以两个Case的恢复速度一样，所以PK CopySet的数量，Case12的CopySet比Case11的CopySet要小，所以更不容易丢失，所以可靠性更高。

六、总结

- 首先AFR越低越好，AFR是直接决定整个集群磁盘故障引起的数据丢失概率的最大因素；
- 其次是恢复速度：在不影响服务可用性指标的前提下，最大限度的提高磁盘故障的恢复带宽是提高集群数据可靠性的另一个重要因素；
- 如果在恢复速度受限的前提下，比如系统架构设计导致的相关发现坏盘到踢盘到进行数据恢复操作启动为5分钟，那么可以通过合理降低磁盘数据的分散程度降低CopySet，如果系统是按分片粒度或Block粒度，则相应的以提高Block粒度来降低数据分散程度的方式来提高数据的可靠性。

参考资料

1. https://zhuanlan.zhihu.com

2.《Copysets: Reducing the Frequency of Data Loss in Cloud Storage》

3. https://www.dazhuanlan.com

4.http://oceanbase.org.cn

> 作者：vivo互联网通用存储研发团队-Gong Bing

服务器 分布式 存储 可靠性

赞 收藏 分享

vivo 互联网技术

分享 vivo 互联网技术干货与沙龙活动，推荐最新行业动态与热门会议。

关注专栏

vivo互联网技术

2.6k 声望 **10k** 粉丝

关注作者

0 条评论

得票数 最新

撰写评论 ...

提交评论

继续阅读

手把手教你实现Android编译期注解

从早期令人惊艳的ButterKnife，到后来的以ARouter为首的各种路由框架，再到现在谷歌大力推行的Jetpack组件...

vivo互联网技术 赞 3 阅读 1.5k

vivo互联网技术 赞 1 阅读 2.1k

分布式存储系统可靠性如何估算？

常规情况下，我们一般使用多副本技术来提高存储系统的可靠性，无论是结构化数据库存储 (如典型的 mysql)、文...

网易云 赞 1 阅读 4.4k

Spark的分布式存储系统BlockManager全解析

摘要：BlockManager 是 spark 中至关重要的一个组件，在spark的运行过程中到处都有 BlockManager 的身影，只...

华为云开发者社区 阅读 853

盘点分布式文件存储系统

在项目的数据存储中，结构化数据通常采用关系型数据库，非结构化数据（文件）的存储就有很多种方式，服务...

三分恶 阅读 4.8k

Golang 分布式系统

比如某电商双 11 时，在 0:00 开始，会有千万到亿级的订单涌入，每秒要处理 10w+ 的订单。在将订单插入数据...

thepoy 阅读 911

分布式系统

分布式系统 分布式系统是若干独立计算机的集合，这些计算机对于用户来说就像是单个系统。 分布式系统的出现...

老污的猫 阅读 565

分布式系统

具体实现为，写本地事务，同步写一张事务消息表，发送事务消息，消息异步通知关联系统，处理失败则利用MQ...

东瓜 阅读 430