关于高可用的系统

2016年08月21日 (Http://coolshell.cn/articles/17459.html) 陈皓 (Http://coolshell.cn/articles/author/haoel) 44,918 人阅读

在《这多年来我一直在钻研的技术 (http://coolshell.cn/articles/17446.html)》这篇文章中，我讲述了一下，我这么多年来一直在关注的技术领域，其中我多次提到了工业级的软件，我还以为有很多人会问我怎么定义工业级？以及一个高可用性的软件系统应该要怎么干出来？这样我也可以顺理成章的写下这篇文章，但是没有人问，那么，我只好厚颜无耻的自己写下这篇文章了。哈哈。

另外，我在一些讨论高可用系统的地方看到大家只讨论各个公司的技术方案，其实，高可用的系统并不简单的是技术方案，一个高可用的系统其实还包括很多别的东西，所以，我觉得大家对高可用的系统了解的还不全面，为了让大家的认识更全面，所以，我写下这篇文章。

理解高可用系统

首先，我们需要理解什么是高可用，英文叫High Availability（Wikipedia词条 (https://en.wikipedia.org/wiki/High_availability)），基本上来说，就是要让我们的计算环境（包括软硬件）做到full-time的可用性。在设计上一般来说，需要做好如下的设计：

1. 对软硬件的冗余，以消除单点故障。任何系统都会有一个或多个冗余系统做standby
2. 对故障的检测和恢复。检测故障以及用备份的结点接管故障点。这也就是failover
3. 需要很可靠的交汇点（CrossOver）。这是一些不容易冗余的结点，比如域名解析，负载均衡器等。

听起似乎很简单吧，然而不是，细节之处全是魔鬼，冗余结点最大的难题就是对于有状态的结点的数据复制和数据一致性的保证（无状态结点的冗余相对比较简单）。冗余数据所带来的一致性问题是魔鬼中的魔鬼：

- 如果系统的数据镜像到冗余结点是异步的，那么在failover的时候就会出现数据差异的情况。
- 如果系统在数据镜像到冗余结点是同步的，那么就会导致冗余结点越多性能越慢。

所以，很多高可用系统都是在做各种取舍，这需要比对着业务的特点来的，比如银行账号的余额是一个状态型的数据，那么，冗余时就必需做到强一致性，再比如说，订单记录属于追加性的数据，那么在failover的时候，就可以到备机上进行追加，这样就比较简单了（阿里目前所谓的异地双活其实根本做不到状态型数据的双活）。

下面，总结一下高可用的设计原理：

- 要做到数据不丢，就必需要持久化
- 要做到服务高可用，就必需要有备用（复本），无论是应用结点还是数据结点
- 要做到复制，就会有数据一致性的问题。
- 我们不可能做到100%的高可用，也就是说，我们能做到几个9个的SLA。

高可用系统的技术解决方案

我在《分布式系统的事务处理 (http://coolshell.cn/articles/10910.html)》中引用过下面这个图：这个图来自来自：Google App Engine的co-founder Ryan Barrett在2009年的Google I/O上的演讲《Transaction Across DataCenter (http://snarfed.org/transactions_across_datacenters_io.html) 》（视频：http://www.youtube.com/watch?v=srOgpXECblk (http://www.youtube.com/watch?v=srOgpXECblk)）

这个图基本上来说是目前高可用系统中能看得到的所有的解决方案的基础了。M/S、MM实现起来不难，但是会有很多问题，2PC的问题就是性能不行，而Paxos的问题就是太复杂，实现难度太大。

总结一下各个高可用方案的的问题：

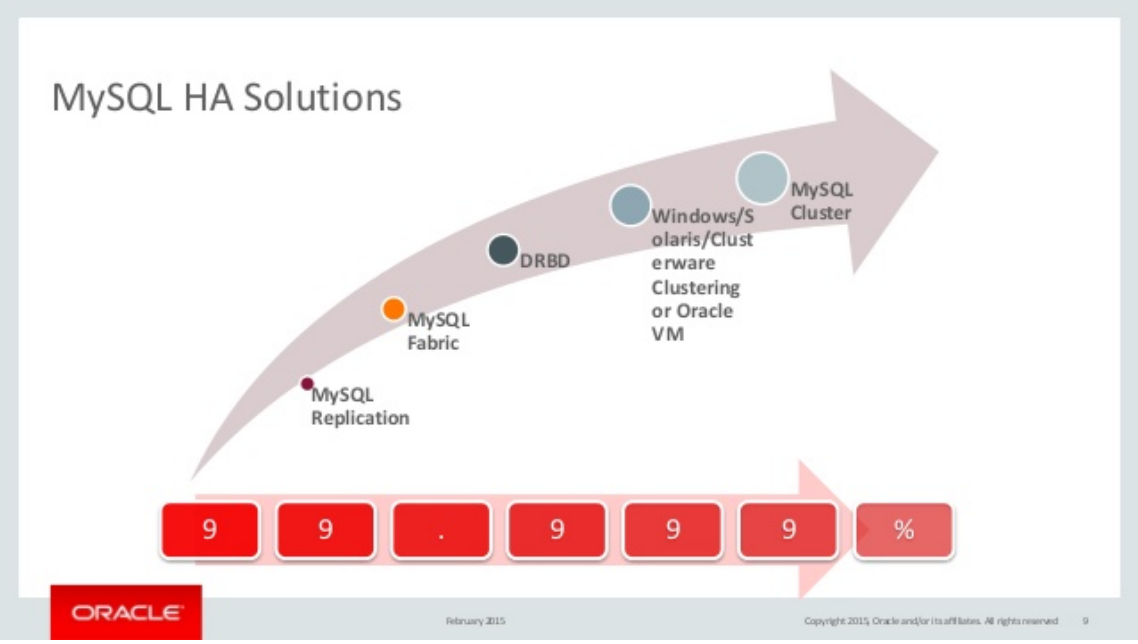
- 对于最终一致性来说，在宕机的情况下，会出现数据没有完全同步完成，会出现数据差异性。
- 对于强一致性来说，要么使用性能比较慢的XA系 (https://en.wikipedia.org/wiki/X/Open_XA)的两阶段提交的方案，要么使用性能比较好，但是实现比较复杂的Paxos协议。

注：这是软件方面的方案。当然，也可以使用造价比较高的硬件解决方案，不过本文不涉及硬件解决方案。

另外，现今开源软件中，很多缓存，消息中间件或数据库都有持久化和Replication的设计，从而也都有高可用解决方案，但是开源软件一般都没有比较高的SLA，所以，如果我们使用开源软件的话，需要注意这一点。

高可用技术方案的示例

下面，我们来看一下MySQL的高可用的方案的SLA（下图下面红色的标识表示了这个方案有几个9）：



(http://www.slideshare.net/andrewjamesmorgan/mysql-high-availability-solutions-feb-2015-webinar)

图片来源：MySQL High Availability Solutions (http://www.slideshare.net/andrewjamesmorgan/mysql-high-availability-solutions-feb-2015-webinar)

简单解释一下MySQL的这几个方案（主要是想表达一个越多的9就越复杂）

- MySQL Repleaction就是传统的异步数据同步或是半同步Semi-Sync（只要有一个slave收到更新就返回成功）这个方式本质上不到2个9。
- MySQL Fabric简单来说就是数据分片下的M/S的读写分离模式。这个方案的的可用性可以达到99%
- DRBD通过底层的磁盘同步技术来解决数据同步的问题，就是RAID 1——把两台以上的主机的硬盘镜像成一个。这个方案不到3个9
- Solaris Clustering/Oracle VM ，这个机制监控了包括硬件、操作系统、网络和数据库。这个方案一般会伴随着节点间的“心跳机制”，而且还会动用到SAN（Storage Area Network）或是本地的分布式存储系统，还会动用虚拟化技术来做虚拟机的迁移以降低宕机时间的概率。这个解决方案完全就是一个“全栈式的解决方案”。这个方

案接近4个9。
- MySQL Cluster是官方的一个开源方案，其把MySQL的集群分成SQL Node 和Data Node，Data Node是一个自动化sharing和复制的集群NDB，为了更高的可用性，MySQL Cluster采用了“完全同步”的数据复制的机制来冗余数据结点。这个方案接近5个9。

那么，这些2个9，3个9，4个9，5个9是什么意思呢？又是怎么来的呢？请往下看。

高可用性的SLA的定义

上面那些都不是本文的重点，本文的重点现在开始，如何测量系统的高可用性。当然是SLA，全称Service Level Agrement (https://en.wikipedia.org/wiki/Service-level_agreement)，也就是有几个9的高可用性。

工业界有两种方法来测量SLA，

- 一个是故障发生到恢复的时间
- 另一个是两次故障间的时间

但大多数都采用第一种方法，也就是故障发生到恢复的时间，也就是服务不可用的时间，如下表所示：

系统可用性%	宕机时间/年	宕机时间/月	宕机时间/周	宕机时间/天
90% (1个9)	36.5 天	72 小时	16.8 小时	2.4 小时
99% (2个9)	3.65 天	7.20 小时	1.68 小时	14.4 分
99.9% (3个9)	8.76 小时	43.8 分	10.1 分钟	1.44 分
99.99% (4个9)	52.56 分	4.38 分	1.01 分钟	8.66 秒
99.999% (5个9)	5.26 分	25.9 秒	6.05 秒	0.87 秒

比如，99.999%的可用性，一年只能有5分半钟的服务不可用。感觉很难做到吧。

就算是3个9的可用性，一个月的宕机时间也只有40多分钟，看看那些设计和编码不认真的团队，把所有的期望寄托在人肉处理故障的运维团队， 一个故障就能处理1个多小时甚至2-3个小时，连个自动化的工具都没有，还好意思在官网上声明自己的SLA是3个9或是5个9，这不是欺骗大众吗？。

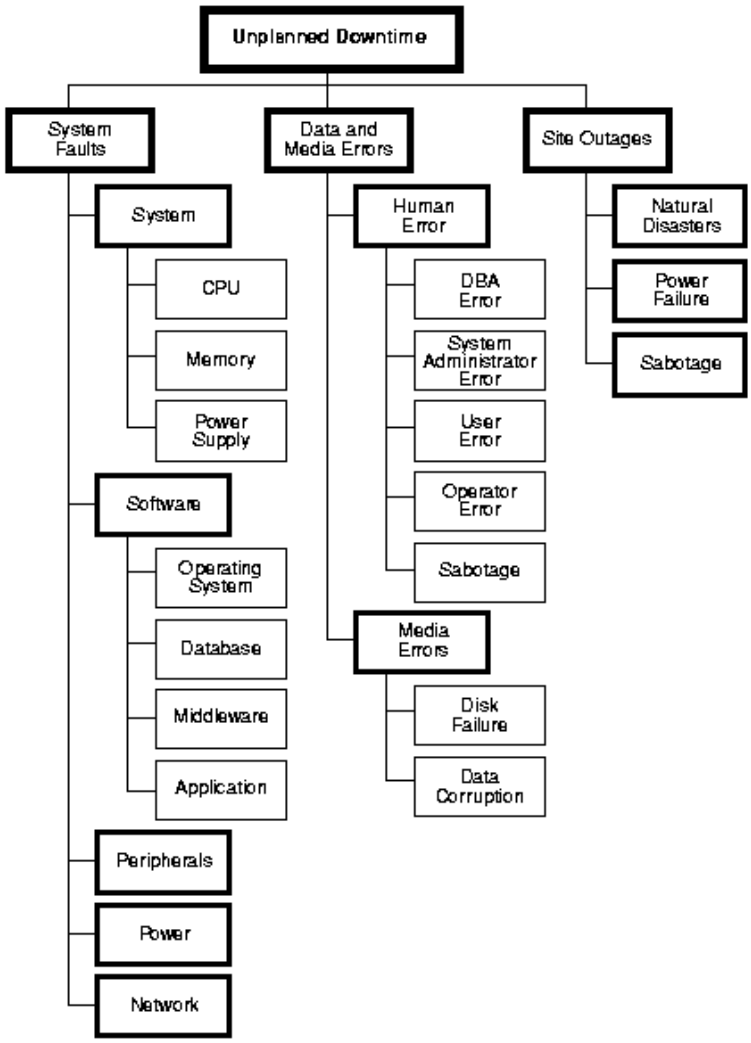
影响高可用的因素

老实说，我们很难计算我们设计的系统有多少的可用性，因为影响一个系统的因素实在是太多了，除了软件设计，还有硬件，还有每三方的服务（如电信联通的宽带SLA），当然包括“建筑施工队的挖掘机”。所以，正如SLA的定义，这不仅仅只是一个技术指标，而是一种服务提供商和用户之间的contract或契约。这种工业级的玩法，就像飞机一样，并不是把飞机造出来就好了，还有大量的无比专业的配套设施、工具、流程、管理和运营。

简而言之，SLA的几个9就是能持续提供可用服务的级别，不过，工业界中，会把服务不可用的因素分成两种：一种是有计划的，一种是无计划的。

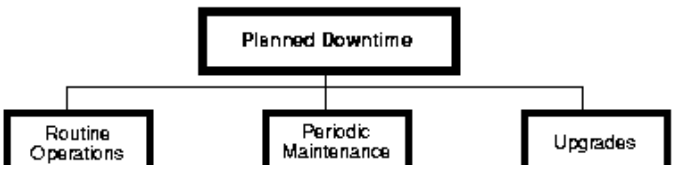
无计划的宕机原因

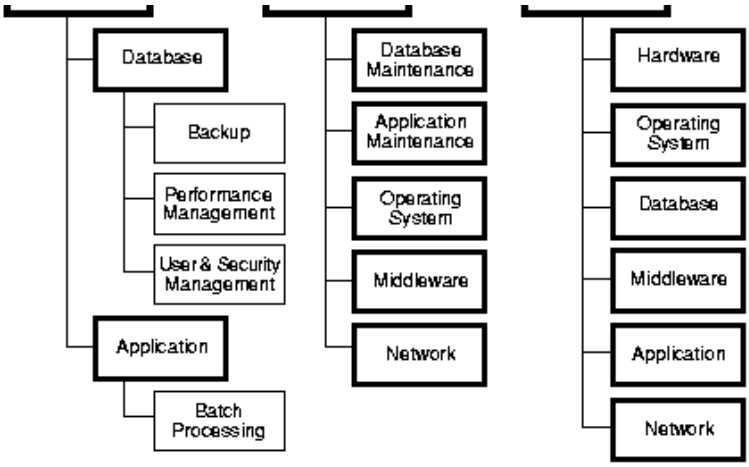
下图来自Oracle的 《High Availability Concepts and Best Practices (https://docs.oracle.com/cd/A91202_01/901_doc/rac.901/a89867/pshavdtl.htm)》



有计划的宕机原因

下图来自Oracle的 《High Availability Concepts and Best Practices (https://docs.oracle.com/cd/A91202_01/901_doc/rac.901/a89867/pshavdtl.htm)》





我们可以看到，上面的宕机原因包括如下：

无计划的

- 系统级的故障 – 包括主机、操作系统、中间件、数据库、网络、电源以及外围设备
- 数据和中介的故障 – 包括人员误操作、硬盘故障、数据乱了
- 还有：自然灾害、人为破坏、以及供电问题。

有计划的

- 日常任务：备份，容量规划，用户和安全管理，后台批处理应用
- 运维相关：数据库维护、应用维护、中间件维护、操作系统维护、网络维护
- 升级相关：数据库、应用、中间件、操作系统、网络、包括硬件升级

真正决定高可用系统的本质原因

从上面这些会影响高可用的SLA的因素，你看到了什么？如果你还是只看到了技术方面或是软件设计的东西，那么你只看到了冰山一角。我们再仔细想一想，那个5个9的SLA在一年内只能是5分钟的不可用时间，5分钟啊，如果按一年只出1次故障，你也得在五分钟内恢复故障，让我们想想，这意味着什么？

如果你没有一套科学的牛逼的软件工程的管理，没有牛逼先进的自动化的运维工具，没有技术能力很牛逼的工程师团队，怎么可能出现高可用的系统啊。

是的，要干出高可用的系统，这TMD就是一套严谨科学的工程管理，其中包括但不限于了：

- 软件的设计、编码、测试、上线和软件配置管理的水平
- 工程师的人员技能水平
- 运维的管理和技术水平
- 数据中心的运营管理水平
- 依赖于第三方服务的管理水平

深层交的东西则是——对工程这门科学的尊重：

- 对待技术的态度
- 一个公司的工程文化
- 领导者对工程的尊重

所以，以后有人在你面前提高可用，你要看的不是他的技术设计，而还要看看他们的工程能力，看看他们公司是否真正的尊重工程这门科学。

其它

有好些非技术甚至技术人员和我说过，做个APP做个网站，不就是找几个码农过来写写代码嘛。等系统不可用的时候，他们才会明白，要找技术能力比较强的人，但是，就算你和他们讲一万遍道理，他们也很难会明白写代码怎么就是一种工程了，而工程怎么就成了一门科学了。其实，很多做技术的人都不明白这个道理。

包括很多技术人员也永远不会理解，为什么要做好多像Code Review、自动化运维、自动化测试、持续集成之类这样很无聊的东西。就像我在《从Code Review 谈如何做技术 (http://coolshell.cn/articles/11432.html)》中提到的，阿里很多的工程师，架构师/专家，甚至资深架构师都没有这个意识，当然，这不怪他们，因为经历决定思维方式，他们的经历的是民用级的系统，做的都是堆功能的工作，的确不需要。

看完这些，最后让我们都扪心自问一下，你还敢说你的系统是高可用的了么？ ;-)

（全文完）

关注CoolShell微信公众账号可以在手机端搜索文章

——=== 访问 酷壳404页面 (http://coolshell.cn/404/) 寻找遗失儿童。 ===——

(31 人打了分，平均分： 4.97)

技术管理 (Http://coolshell.cn/category/%e6%8a%80%e6%9c%af%e7%ae%a1%e7%90%86), 程序设计 (Http://coolshell.cn/category/progdesign), 系统架构 (Http://coolshell.cn/category/%e7%b3%bb%e7%bb%9f%e6%9e%b6%e6%9e%84)

Design (Http://coolshell.cn/tag/design), High Availability (Http://coolshell.cn/tag/high-availability), Paxos (Http://coolshell.cn/tag/paxos), Programmer (Http://coolshell.cn/tag/programmer), 分布式 (Http://coolshell.cn/tag/%e5%88%86%e5%b8%83%e5%bc%8f), 程序员 (Http://coolshell.cn/tag/%e7%a8%8b%e5%ba%8f%e5%91%98)

相关文章

- 2014年01月20日 分布式系统的事务处理 (http://coolshell.cn/articles/10910.html)
- 2013年07月05日 IoC/DIP其实是一种管理思想 (http://coolshell.cn/articles/9949.html)
- 2011年10月25日 多些时间能少写些代码 (http://coolshell.cn/articles/5686.html)
- 2011年06月10日 软件真的好难做啊 (http://coolshell.cn/articles/4811.html)
- 2011年09月08日 千万不要把 bool 设计成函数参数 (http://coolshell.cn/articles/5444.html)
- 2012年03月09日 Bret Victor – Inventing on Principle (http://coolshell.cn/articles/6775.html)
- 2014年04月12日 从Code Review 谈如何做技术 (http://coolshell.cn/articles/11432.html)
- 2016年09月18日 什么是工程师文化？ (http://coolshell.cn/articles/17497.html)

《关于高可用的系统》的相关评论


石樱灯笼 (https://www.catscarlet.com/)说道：

2016年08月21日 13:16 (http://coolshell.cn/articles/17459.html#comment-1871738)
对于目前多数企业来讲：
HA=双机热备；
failover=半夜给开发打电话来加班；

光是机房托管主机的双备电源，都是接到一个插线板上的，更别说备用机了。
至于数据同步……缓存和数据库同步都不正常，更别说多数据库同步了。

一套严谨科学的工程管理，这个想有，但是没有。


孙华东说道：

2016年08月24日 16:21 (http://coolshell.cn/articles/17459.html#comment-1872730)
这个给赞


千橙 (http://qiancheng.me)说道：

2017年01月24日 16:52 (http://coolshell.cn/articles/17459.html#comment-1911635)
这个真相了（泪目


conry说道：

2016年08月21日 13:28 (http://coolshell.cn/articles/17459.html#comment-1871743)
就是支付宝也就90%可用性


conry说道：

2016年08月21日 13:31 (http://coolshell.cn/articles/17459.html#comment-1871745)

> *conry :*
> *就是支付宝也就90%可用性*

应该是99.9%

w说道：

2016年08月21日 13:41 (http://coolshell.cn/articles/17459.html#comment-1871749)
没钱，自己学，学不完，想哭。

haitao (http://haitao.appinn.me)说道：

2016年08月21日 17:06 (http://coolshell.cn/articles/17459.html#comment-1871819)
【如果按一年只出1次故障，你也得在五分钟内恢复故障】这种要求，只要做得起双机热备，应该都不怕了：5分钟足够切换备机顶上去就行了。具体故障分析解决，可以拿换下来的系统再慢慢研究

阿长说道：

2017年01月30日 10:23 (http://coolshell.cn/articles/17459.html#comment-1912058)
一个不小心，备机跟主机部署在同一个机架上，然后故障的原因是机架掉电/断网。或者不小心，在同一个机房，然后机房附近的工地发生了爆炸。。

或者说万一5分钟还没发现问题已经产生了咋办

vingc说道：

2016年08月21日 19:28 (http://coolshell.cn/articles/17459.html#comment-1871837)
好文，工业级和民用系统的区别真的很大；
了解的越多，自己越无知。

passinger说道：

2016年08月21日 20:58 (http://coolshell.cn/articles/17459.html#comment-1871848)
有多少公司敢把他们的几个9的SLA明确地宣传出来的？


jason说道：

2016年08月21日 22:56 (http://coolshell.cn/articles/17459.html#comment-1871868)
好文，求推荐分布式系统相关的中文/英文书籍或者文章


kinyou说道：

2016年08月21日 23:05 (http://coolshell.cn/articles/17459.html#comment-1871872)
貌似真正做到没有几家公司, 堆功能是99.9%,更直白的是没有钱买那多服务器


谢振业说道：

2016年08月21日 23:47 (http://coolshell.cn/articles/17459.html#comment-1871882)
mysql主从靠监控和人工切换，一个9肯定有，半自动的话两个9都能做到。fabric是有高可用功能的，这类做到3个9也是可以的


谢振业 (http://xiezhenye.com)说道：

2016年08月21日 23:52 (http://coolshell.cn/articles/17459.html#comment-1871884)
不过这种没算上整个机房挂掉的情况


nettm说道：

2016年08月22日 08:50 (http://coolshell.cn/articles/17459.html#comment-1871979)
国内有能达到皓哥说的这种工业级别的公司吗？


nkxiaochuan说道：

2016年08月22日 09:32 (http://coolshell.cn/articles/17459.html#comment-1871984)
应该是Service Level Agreement


mwang说道：

2016年08月22日 10:06 (http://coolshell.cn/articles/17459.html#comment-1871993)
5个9的稳定性是电信级要求，当年朗讯交换机号称可以达到6个9，不过真是太费钱了，行业垄断的时候才搞的起！现在通讯行业不景气，有时候感觉5个9的要求是不是over engineering，说实话有了5个9的要求，堆feature真的很难，成本也很高。互联网行业3个9的稳定性，大家用起来也挺高兴，反正免费的。A站宕机了换B站也没啥大不了。时代一直再变，玩法和以前也不一样了。不过对工程师来讲，SLA的概念是必须，否则就天天过自己挖坑自己埋的日子吧。


题叶 (http://tiye.me)说道：

2016年08月22日 10:13 (http://coolshell.cn/articles/17459.html#comment-1871997)
同样的概念, 放到前端和放到后端意义差别真是不一样啊, 毕竟是民用级别的.


asklxf (http://www.liaoxuefeng.com)说道：

2016年08月22日 10:46 (http://coolshell.cn/articles/17459.html#comment-1872004)
没有高可用运维搞高可用意义不大，普通网站搞个双机，数据库和web应用实时监控挂了就自动重启，数据库一年大概崩溃1-2次，node程序一个月大概崩溃一次，linux从来没崩溃过，这个无运维方案实测也能到99.9%


jade说道：

2016年08月22日 10:58 (http://coolshell.cn/articles/17459.html#comment-1872007)
「深层交的东西则是」，打错了？


刘sir说道：

2016年08月22日 11:18 (http://coolshell.cn/articles/17459.html#comment-1872009)
怒赞！


_www (http://yinxs2003@gmail.com)说道：

2016年08月22日 11:52 (http://coolshell.cn/articles/17459.html#comment-1872016)
6的飞起


Mars说道：

2016年08月22日 12:20 (http://coolshell.cn/articles/17459.html#comment-1872018)
这篇大赞特赞！


knull说道：

2016年08月22日 16:17 (http://coolshell.cn/articles/17459.html#comment-1872054)

> *haitao :*
> *【如果按一年只出1次故障，你也得在五分钟内恢复故障】这种要求，只要做得起双机热备，应该都不怕了：5分钟足够切换备机顶上去就行了。具体故障分析解决，可以拿换下来的系统再慢慢研究*

这仅仅是理想状态而已。之前做的运营商鉴权计费系统，运行环境是双机，数据库也是双机，会自动检测、发现故障会自动切换。但是，还是会发生启动失败，导致切换失败，来回切~~~


Jay (http://www.jayxu.com)说道：

2016年08月22日 16:52 (http://coolshell.cn/articles/17459.html#comment-1872061)
更正一个小细节：5个9一年宕机时间的“5.26分”，不是“5分26秒（5分半）”，换算一下约5分15秒


千橙 (http://qiancheng.me)说道：

2017年01月24日 17:00 (http://coolshell.cn/articles/17459.html#comment-1911638)
准确


这也是呢称？说道：

2016年08月22日 18:52 (http://coolshell.cn/articles/17459.html#comment-1872091)
通信行业出身，对于代码质量的认识深入骨髓了。


七戒说道：

2016年08月22日 19:36 (http://coolshell.cn/articles/17459.html#comment-1872102)
这篇写的太好了。谢谢。


黄土高坡说道：

2016年08月23日 09:28 (http://coolshell.cn/articles/17459.html#comment-1872225)
很长知识

Pingback： 关于高可用的系统 | 酷 壳 – CoolShell.cn - code (http://code.ownlike.com/%e5%85%b3%e4%ba%8e%e9%ab%98%e5%8f%af%e7%94%a8%e7%9a%84%e7%b3%bb%e7%bb%9f-%e9%85%b7-%e5%a3%b3-coolshell-cn.html)


运维派 (http://www.yunweipai.com)说道：

2016年08月23日 16:01 (http://coolshell.cn/articles/17459.html#comment-1872285)
从业务出发的高可用，需要从软件、系统、硬件、网络各个层面去检视去可靠性。


raindylong说道：

2016年08月23日 22:29 (http://coolshell.cn/articles/17459.html#comment-1872420)
写到心坎去了，工程能力工程能力工程能力，可惜没有多少老板&管理者看到这个层次的问题。


wombat说道：

2016年08月24日 16:14 (http://coolshell.cn/articles/17459.html#comment-1872728)
看的出来你很不喜欢阿里啊，不过我也不喜欢，程序用着么还行，那所谓的企业文化实在是。。。


Luke说道：

2016年08月24日 17:25 (http://coolshell.cn/articles/17459.html#comment-1872745)
@passinger
华为，华为会直接告诉客户，还有没达到客户按分钟罚款，合同里边有写。


烟台淘宝代运营 (http://www.xiaoshitou123.com/)说道：

2016年08月26日 17:09 (http://coolshell.cn/articles/17459.html#comment-1873641)
没有接触过，絮叨了新知识


枕边书说道：

2016年08月26日 21:19 (http://coolshell.cn/articles/17459.html#comment-1873733)
看出来是用五笔的了~哈哈，初学者还是要慢慢学。

tbag说道：

2016年08月30日 01:28 (http://coolshell.cn/articles/17459.html#comment-1874714)

这不仅是国内的现状，国外很多大公司也一样，客户不停提需求，有些feature就必须在内核或者驱动程序里实现，导致SLA越来越低

z-saks (http://gilarus.org)说道：

2016年08月30日 16:58 (http://coolshell.cn/articles/17459.html#comment-1874925)

10/9=90%

100/99=99%

1000/999=99.9%

10000/9999=99.99%

100000/99999=99.999%

...

关键第一步在于选择。。。。。100%的目标是不切合实际的，几个9，决定了投入多少。

D瓜哥 (http://www.diguage.com/)说道：

2016年09月05日 16:05 (http://coolshell.cn/articles/17459.html#comment-1877921)

> *z-saks :*
>
> *10/9=90%*
>
> *100/99=99%*
>
> *1000/999=99.9%*
>
> *10000/9999=99.99%*
>
> *100000/99999=99.999%*
>
> *...*
>
> *关键第一步在于选择。。。。。100%的目标是不切合实际的，几个9，决定了投入多少。*

10/9 = 1.1111 约等于 111%

100/99 = 1.01010101 约等于 101.01%

1000/999 = 1.001001001 约等于 100.10%

10000/9999 = 1.00010001 约等于 100.01%

100000/99999 = 1.00001 约等于 100.00%

专业砸场三十年！O(∩_∩)O~

z-saks (http://gilarus.org)说道：

2016年09月12日 17:33 (http://coolshell.cn/articles/17459.html#comment-1881165)

@D瓜哥

失误，:-(

you won!

monklof (http://monklof.com)说道：

2016年09月14日 15:34 (http://coolshell.cn/articles/17459.html#comment-1881850)

请教一下，这里的crossover是指什么呢？

shady说道：

2016年10月08日 13:43 (http://coolshell.cn/articles/17459.html#comment-1890959)

掐指一算，就现如今大天朝的工业制造能力，也许没有哪个行业可以算做是工业级的

RikuH说道：

2016年10月09日 18:20 (http://coolshell.cn/articles/17459.html#comment-1891170)

“当然是SLA，全称Service Level Agrement” 应为”Agreement”

echo说道：

2016年10月18日 17:29 (http://coolshell.cn/articles/17459.html#comment-1892565)

得90分容易，得99分难，根据实际业务需求吧，有的项目挂了一两天都没人知道。。。


阿里新秀|http://www.alixinxiu.com (http://www.alixinxiu.com/)说道：

2016年11月25日 12:03 (http://coolshell.cn/articles/17459.html#comment-1901515)
学习了,赞一个!


ｙａｎｇ说道：

2016年11月25日 22:19 (http://coolshell.cn/articles/17459.html#comment-1901634)
最后哪点评价 阿里的内容是本文的重点， 是作者最想表达的吧，哈哈~


小菜胖说道：

2016年12月01日 21:59 (http://coolshell.cn/articles/17459.html#comment-1903994)
向5个9努力


高到QVOD (http://1999yyy.com)说道：

2016年12月05日 11:38 (http://coolshell.cn/articles/17459.html#comment-1904766)
这篇写的太好了，很长见识

Pingback： 关于高可用的系统 | 酷 壳 – CoolShell.cn – Aoki自留地 (http://www.bugjc.com/2016/12/24/%e5%85%b3%e4%ba%8e%e9%ab%98%e5%8f%af%e7%94%a8%e7%9a%84%e7%b3%bb%e7%bb%9f-%e9%85%b7-%e5%a3%b3-coolshell-cn/)

Pingback： 从Gitlab误删除数据库想到的 | | 酷 壳 - CoolShell (http://coolshell.cn/articles/17680.html)


学习者说道：

2017年02月06日 17:15 (http://coolshell.cn/articles/17459.html#comment-1912597)
还有每三方的服务 tx 第三方