17 | 故障处理最佳实践：应对故障

2017-11-28 陈皓

或多或少我们都会经历线上的故障。在我的职业生涯中，就经历过很多的线上故障。老实说，线上故障是我们技术人员成长中必须要经历的事。从故障中我们可以吸取到很多教训，也能让我们学到很多书本上学不到的知识。坑踩多了，我们会变得越来越有经验，也就成为老司机了。

不过，我看到很多公司处理线上故障的方式并不科学，而且存在很多问题，所以，今天这篇文章就来分享一些我的经验。这些经验主要来自亚马逊和阿里这两家互联网公司，以及我个人的经验总结。希望这套方法能够对你有帮助。

故障发生时

在故障发生时，最重要的是快速恢复故障。而快速恢复故障的前提是快速定位故障源。因为在很多分布式系统中，一旦发生故障就会出现“多米诺骨牌效应”。也就是说，系统会随着一个故障开始一点一点地波及到其它系统，而且这个过程可能会很快。一旦很多系统都在报警，要想快速定位到故障源就不是一件简单的事了。

在亚马逊内部，每个开发团队至少都会有一位oncall的工程师。在oncall的时候，工程师要专心处理线上故障，轮换周期为每人一周。一旦发生比较大的故障，比如，S1全部不可用，或S2某功能不可用，而且找不到替代方案，那么这个故障就会被提交到一个工单系统里。几乎所有相关团队oncall的工程师都会被叫到线上处理问题。

工作流是这样的，工程师先线上签到，然后自查自己的服务，如果自己的服务没有问题，那么就可以在旁边待命（standby），以备在需要时进行配合。如果问题没有被及时解决，就会自动升级到高层，直到SVP级别。

大家都知道，在亚马逊，不是按技能分工，而是按职责分工，也就是一个团队不是按前端、后端、运维等来分工，而是按所负责的Service来分工。

所以，亚马逊的开发人员都是前端、后端、测试、运维全部都要干的。而亚马逊内部有很多的服务，一旦出现问题，为了避免一个工单在各个团队流转，需要所有团队上线处理，这样是最快的。

如果我们的系统架构是分布式服务化的，那么一个用户的请求可能会经过很多的服务，开发和运维起来是非常麻烦的。此时，跨团队跨部门的开发和运维就变得非常重要了。

就我的经历而言，在故障发生时，亚马逊的处理过程是比较有效和快速的，尤其是能够快速地定位故障源。对于被影响的其他团队也可以做一定的处理，比如做降级处理，这样可以控制故障的范围不被扩散。

故障源团队通常会有以下几种手段来恢复系统。

重启和限流。重启和限流主要解决的是可用性的问题，不是功能性的问题。重启还好说，但是限流这个事就需要相关的流控中间件了。

回滚操作。回滚操作一般来说是解决新代码的bug，把代码回滚到之前的版本是快速的方式。

降级操作。并不是所有的代码变更都是能够回滚的，如果无法回滚，就需要降级功能了。也就是说，需要挂一个停止服务的故障公告，主要是不要把事态扩大。

紧急更新。紧急更新是常用的手段，这个需要强大的自动化系统，尤其是自动化测试和自动化发布系统。假如你要紧急更新1000多台服务器，没有一个强大的自动化发布系统是很难做到的。

也就是说，出现故障时，**最重要的不是debug故障，而是尽可能地减少故障的影响范围，并尽可能快地修复问题**。

国内的很多公司，都是由专职的运维团队来处理线上问题的。然而，运维团队通常只能处理一些基础设施方面的问题，或是非功能性的问题。对于一些功能性的问题，运维团队是完全没有能力处理的，只能通过相应的联系人，把相关的开发人员叫到线上来看。

而可能这个开发人员看到的是别的系统有问题，又会叫上其它团队的人来。所以，一级一级地传递下去，会浪费很多时间。

故障前的准备工作

为了能够在面临故障时做得有条不紊，我们需要做一些前期的准备工作。这些准备工作做得越细，故障处理起来也就越有条理。我们知道，故障来临时，一切都会变得混乱。此时，对于需要处理故障的我们来说，事可以乱，但人不能乱。如果人跟着事一起乱，那就是真正的混乱了。

所以，我们需要做一些故障前的准备工作。在这里，我给出一些我的经验。

以用户功能为索引的服务和资源的全视图。首先，我们需要一个系统来记录前端用户操作界面和后端服务，以及服务使用到的硬件资源之间的关联关系。这个系统有点像CMDB（配置管理数据库），但是比CMDB要大得多，是以用户端的功能来做索引的。然后，把后端的服务、服务的调用关系，以及服务使用到的资源都关联起来做成一个视图。这个视图最好是由相应的自动化监控系统生成。有了这个资源图后，我们就可以很容易地找到处理故障的路径了。**这就好像一张地图，如果没有地图，我们只能像个无头苍蝇一样乱试了**。

为地图中的各个服务制定关键指标，以及一套运维流程和工具，包括应急方案。以用户功能为索引，为每个用户功能的服务都制定一个服务故障的检测、处理和恢复手册，以及相关的检测、查错或是恢复的运维工具。对于基础层和一些通用的中间件，也需要有相应的最佳实践的方法。

比如Redis，怎样检查其是否存在问题，怎样查看其健康和运行状态？哪些是关键指标，面对常见的故障应该怎么应对，服务不可用的服务方案是什么，服务需要回滚了应该怎么操作，等等。**这就好像一个导航仪，能够告诉你怎么做。而没有导航仪，就没有章法，会导致混乱**。

设定故障的等级。还要设定不同故障等级的处理方式。比如，亚马逊一般将故障分为4级：1级是全站不可用；2级是某功能不可用，且无替代方案；3级是某功能不可用，但有替代方案；4级是非功能性故障，或是用户不关心的故障。阿里内的分类更多样一些，有时会根据影响多少用户来定故障等级。

制定故障等级，主要是为了确定该故障要牵扯进多大规模的人员来处理。故障级别越高，牵扯进来的人就越多，参与进来的管理层级别也就越高。就像亚马逊的全员上线oncall一样。**这就好像是我们社会中常用的“红色警报”、“橙色警报”、“黄色警报”之类的，会触发不同的处理流程**。

故障演练。故障是需要演练的。因为故障并不会时常发生，但我们又需要不断提升处理故障的能力，所以需要经常演练。一些大公司，如Netflix，会有一个叫Chaos Monkey的东西，随机地在生产线上乱来。Facebook也会有一些故障演习，比如，随机关掉线上的一些服务器。总之，要提升故障处理水平，最好的方式就是实践。见得多了，处理得多了，才能驾轻就熟。**故障演练是一个非常好的实践**。

灰度发布系统。要减少线上故障的影响范围，通过灰度发布系统来发布是一个很不错的方式。毕竟，我们在测试环境中很难模拟出线上环境的所有情况，所以，在生产线上进行灰度发布或是A/B测试是一件很好的事。

在亚马逊，发布系统中有一个叫Weblab的系统，就是用来做灰度发布的。另外，亚马逊全球会有多个站点。一般来说，会先发中国区。如果中国区没什么问题了，就发日本区，然后发欧洲区，最后是美国区。而如果没有很多站点的话，那么你就需要一个流量分配系统来做这个事了。

好了。今天就分享这么多。我觉得，只要能做好上面的几点，你处理起故障来就一定会比较游刃有余了。

在这篇文章的末尾，我想发个邀请给你。请你来聊聊，你所经历过的线上故障，以及有哪些比较好的故障处理方法。

精选留言 25

左耳朵
1512465019

自动地图生成一般用APM式的系统。开源的可以看看zipkin

kimi
1512193120

2013 年，应该是 8 月吧，和耗子哥一起处理巨石塔上千台服务器宕机的故障，搞到凌晨三四点

ibrothergang
1511829964

“请你来聊聊，你所经历过的线上故障，以及有哪些比较好的故障处理方法。”

我是一名移动端开发的工程师。移动端的开发工作和前端(线上环境)开发还是有一点区别的。移动端的开发一般在上线前会做测试，严重的问题一般在测试过程就解决了，很少情况发版后出现大面积的奔溃情况。但是线上环境不一样，线上环境发版的周期会大大短于客户端，很多的活动都会频繁的上线和下线。影响的范围也大于移动端。

遇到过最严重的一次事故是由于服务端的修改引起了移动端的奔溃。而且这个奔溃发生在 app 启动的时候。也就是说用户点了应用图标，起来马上就又闪退了。当时的 app 设计是起来后会去请求服务端的相关配置信息，相信很多的 app 也是这么做的。造成这个故障的原因是由于 app 对异常的处理不够完备，服务器端又恰巧修改了配置数据，导致 app 端拿到了一个引起奔溃的数据结果。后来因为是上班时间，发现问题后大家都在，及时恢复了服务端数据，遏制了事态的进一步发展，但是已经出现奔溃的用户由于在重新请求服务端数据前就奔溃了，只能通过发布新版本解决这个问题。

一旦服务端和移动端相互影响(往往是服务端影响移动端)引起的奔溃，往往是比较严重的，很多时候不得不通过发布新版本才能解决问题。所以移动端一定要做好服务端的异常处理。

金胖子
1517532052

最典型的一次，项目组成员在测试版本中加了sleep来debug，结果上线的时候就把版本发布到生产，直接影响我第二天下午没能去看变形金刚

xpisme
1529941880

代码逻辑错误，导致查看分享的人能看到分享者所有信息，记录的上一个人的cookie.

Session存储在redis, flush db。所有用户重新登陆

小桥流水
1512095498

自动生成地图那是有什么工具推荐的吗？

Link
1525930135

楼上的，类似工具:鹰眼，watchMan,京东的CallGraph

paul.yang
1532139753

耗子叔，我是个自学转行做后端的程序员。最近在日活快接近2亿的一个后端团队里面犯了个错误导致某一个功能20分钟不可用，受到了打击，我微博给你留了言，希望能跟你交流下，寻求指导帮助。希望你能看到我的微博留言，呵呵的卫国杨

作者回复 微博回了

晏
1531010492

故障前的准备工作:
以用户功能为索引的服务和资源的全视图。
在地图中设置关键指标，以及运维流程和方案。
设定故障等级。
故障演练。

林子
1512129278

自动生成地图那是有什么工具推荐的吗？同问耗子哥

小沫
1511825464

之前有一次线上系统出现故障，导致工单无法处理。原因是北向接口服务出现故障，定位起来不太方便。因为接口为集群部署（使用F5）当时没有好的运维工具，只能模拟请求接口，经过一轮验证后才发现接口故障点。想问下耗子叔，对于你文章中说的自动生成地图那是有什么工具推荐的吗？

edisonhuang
1559004432

故障应对方法，分别包括故障发生时的恢复措施，自己故障发生前可做的准备。
故障发生时最重要的是限制故障影响的范围，尽最大可能保障服务的可用性，包括转发和限流，回滚，降级，服务重启，紧急更新，紧急发布等。
故障发生前应做好防范，需要以用户功能为索引建立全站服务和资源的地图，利用地图为各个服务生成关键性指标，并建立一套自动化运维的方案和工具。为故障设立等级，知道故障时我在哪，严重程度，进行必要的故障演练，做灰度发布等

西北偏北
1557451371

凡事预则立，故障诊断和处理不是依赖人员的瞎猜，盲查，而是要在故障前就想好对应的预案，基础系统的研发支持，日志埋点等等。毕竟线上不方便像本地一样debug

---

KaitoShy
1551534203

遇到的故障还是蛮多的，由于网站是PHP在请求过密的时候，出现502

Eleven
1550572565

故障发生时，我们公司一般按照顺序：重启和限流、紧急更新、版本回退、服务降级.

Geek_fb3db2
1542118338

咨询下耗子叔 文章提到的降级限流有没有有什么成熟的解决方案 目前项目中领导提到了 但是不知道如何做

永立
1536676757

技术不太够，这章很多内容看的不是很懂。

山哥
1532181436

大佬，CMDB的服务视图能发出来看下？

晏
1531010235

出现故障时，最重要的不是 debug 故障，而是尽可能地减少故障的影响范围，并尽可能快地修复问题。

KingPoker
1529206477

去年生产遇到不少问题，处理了几次，越来越有思路。
文章提到的各种工程化的管理，还需要很长的路

yun
1525878388

最关键的定位故障原因，怎么没有说呢？
我通过监控能定位具体出问题的服务，可是服务出问题的原因如果没有定位出来的话，是不大能知道如何恢复系统的吧？

Geek_03bf3e
1565327504

有一次做数据库某字段扩容升级，需求提给dba后，第二天大量新增用户无法注册，原因是dba更新数据库是，将自增索引更没了

prader
1563504829

现在再小公司，经常改一些代码的逻辑上的bug，谢谢分享。

依然didala
1518356634

还有一个处理，紧急切FO。

Frankzhang

1512211407

同求服务地图工具