**摘要:** 2016年3月，我所就职的国内某某知名互联网公司组织研发了一套分布式支付平台，该支付平台主要满足公司快速发展和各业务线业务流量日益增加的支付需求，用于支撑各业务线的支付功能，我有幸被定为该平台的架构设计师，主要负责架构设计工作。本文以该支付平台为例，主要论述了软件容错技术和方法在该系统中的具体应用。通过采用以集群化的形式进行应用部署；通过主备形式的数据库部署进行软件容错；通过程序设计方面进行软件的容错与避错。事实证明，以上软件容错技术的应用对于系统的可用性、安全性和可扩展性方面起到了很好的效果，满足了该系统的性能需求，并且该平台从上线到目前一直稳定运行，得到了各业务线负责人和公司高层领导的认可和赞赏。

2016年3月，我所就职的国内莫某知名互联网公司组织研发了一套分布式支付平台，由于公司快速发展和各业务线业务流量的日益增加，各业务线迫切需要一个稳定健壮的支付平台，以下简称为平台，用于支撑各业务线的支付功能，公司的业务流量入口分为pc端、移动端、微信端和其他渠道等。我作为该平台的架构设计师，主要负责该平台的架构设计工作。  
平台采用的核心架构风格为微服务的架构风格，采用Java语言为核心开发语言，将平台服务划分为了三类服务，核心服务，平台web服务、平台保障服务。其中核心服务主要分为订单服务、账户服务、网关服务、清结算对账服务、一键支付聚合服务等等，平台web服务主要提供与用户对接的界面等，平台保障服务主要包括例如JOB框架，平台报警服务，MQ服务等等。对这些服务中的核心服务、平台web相关的服务必须要能满足7\*24小时的稳定运行，对软件的可靠性要求非常高，所以在该平台中的这些核心服务就必须具备一定的容错能力，在某个服务运行时出错的情况下不能影响到整个集群中的服务，这就要求在软件架构设计中必须考虑到软件容错技术的应用。  
提高软件系统可靠性技术主要分为容错技术和避错技术，容错技术的主要方式为冗余，冗余又分为结构冗余、时间冗余、信息冗余，冗余附加。结构冗余又分为静态冗余、动态冗余和混合冗余。软件容错技术主要有N版本程序设计，恢复块方法，和防卫式程序设计。结合互联网软件的性质，我主要采用了集群技术、数据库主从方式、和程序设计方面来进行软件的容错与避错处理。下面就从以上三方面详细讨论我所采用的容错技术和方法。  
1、通过集群技术来容错  
平台中的各服务如果在运行时部署在一台服务器上，那么当服务器发生故障，整个平台将不能再提供任何服务。所以一般非常小规模的应用才会采取这样的部署方式，像互联网应用这样的支付平台来说必须采用多机同时部署的方式，防止单台服务器宕机或者服务进程Crash导致整个平台不能提供服务的问题。通过多机同时部署，当一台服务出现问题时，可以很容易替换一台新服务器进行重新部署生效，通过服务客户端的软负载均衡功能，可动态剔除不可用服务机器，动态发现新加入集群的服务机器，使平台在出现故障时可平滑过渡，达到容错的目的。另外平台中各服务当首次获取到的不易变的静态数据会将其存入非本地缓存中，例如采用了redis cluster技术，可以很好的保证写入缓存中的数据获取的高可靠性，恰当使用缓存不但会提升平台性能，同时还可以起到容错的效果，例如当某个服务所依赖的后端存储发生了短时的故障或者网络抖动，在这个时候大量的并发请求发现存储获取失败直接从缓存中获取将数据返回给调用方，起到了很好的容错效果。  
2、通过数据库主从部署方式来容错  
对于该平台来说，所依赖的后端数据库存储的稳定性是非常非常重要的，所有的订单，交易、账户等等数据将直接存储到数据库中。如果数据库在运行过程中频繁宕机，那么带来的问题将是不能容忍的，因为会造成订单丢失，交易丢失，账户余额出错等等问题，所以在这里就要求数据库存储要具有非常高的可靠性，同时具有很强的容错性，在这里我主要采用了数据库的部署结构为主从式方式，要求部署在不同服务器上，在不出问题的情况下对于一些时效性要求不是很高的场合从库可以负责承担一部分读流量，当主库发生读写问题时，可快速由其他的从库升级为主库，继续服务，达到容错的效果。在这里我还采用了要求数据库宕机加报警的方式来防止宕机的主从数据库实例过多，导致在并发高的情况下没有可用的从库升级为主库提供服务，通过这样的方式也提高了整个平台的高可靠性。数据库的数据文件存储这里我也要求采用了RAID磁盘容错技术来防止单块磁盘损坏导致的数据文件丢失问题。  
3、通过程序设计方面进行软件的容错与避错。  
根据以往的架构经验，那么系统的不可靠大部分是由于程序内部的设计或者网络请求参数的配置或者连接池参数的配置不当所导致的。所以通过程序设计方面进行软件的容错是非常重要的。在程序设计方面的容错用的最普遍的就是防卫式程序设计，例如平台中的一键支付聚合服务，当在支付的过程中调用账户服务来进行账户金额扣减的时候，势必会调用账户服务传递请求对象来处理，如果说账户服务在被调用的这一刻网络抖动或者丢包的情况下，这个时候一键支付聚合服务必然会收到抛出的错误信息，如果没有通过恰当的容错处理，那么这次一键支付必然会给用户显示支付失败，不太友好，在这里我采用了TRY CATCH机制加3次重试的容错处理机制，就解决了该次支付因网络抖动导致的支付失败问题。平台采用的是微服务的架构风格，那么在服务之间的通讯过程中涉及到数据的传递，这里我采用了在数据传输协议的头部加CRC码来做到对错误数据处理的避错。  
通过采用了以上容错技术的方法和措施后，平台从上线运行到目前为止，各服务运行状态良好，通过日志分析来看，支付成功率很高，得到了公司高层领导和各业务线负责人的赞赏和认可。但在我看来还有如下两点不足。1.各服务间调用事物一致性问题的容错处理。针对该问题，目前只能保证事物的最终一致性，因为根据CAP理论，要解决该问题确实存在一定的难度，后面我准备研究下TCC事物处理方式尤其适合支付平台场景，争取在不损失性能的前提下最大限度解决分布式事物的一致性问题。2.目前所采用的最大努力推送型事物服务依赖MQ重复消息的问题。针对该问题我采用加了一张消息处理表的方式来解决，当收到消息的时候，先查询该条消息是否已经处理，如果没有处理直接进行处理并将其进行记录，防止重复处理导致支付数据出错。  
软件容错技术对软件的稳定性起着至关重要的作用，尤其是针对互联网性质的软件并发高存在流量峰值等问题，软件容错技术的应用的重要性就不言而喻了。经过这次我所采用的软件容错技术的方法和措施的实施效果后，使我也看到了自己身上的不足之处，我会在今后的架构设计过程中，不断更新自己的知识，不断完善自己的架构设计领域，设计出更好的软件架构，更好的支撑业务平台的运行，提高公司的竞争力，为公司为社会尽一份绵薄之力。