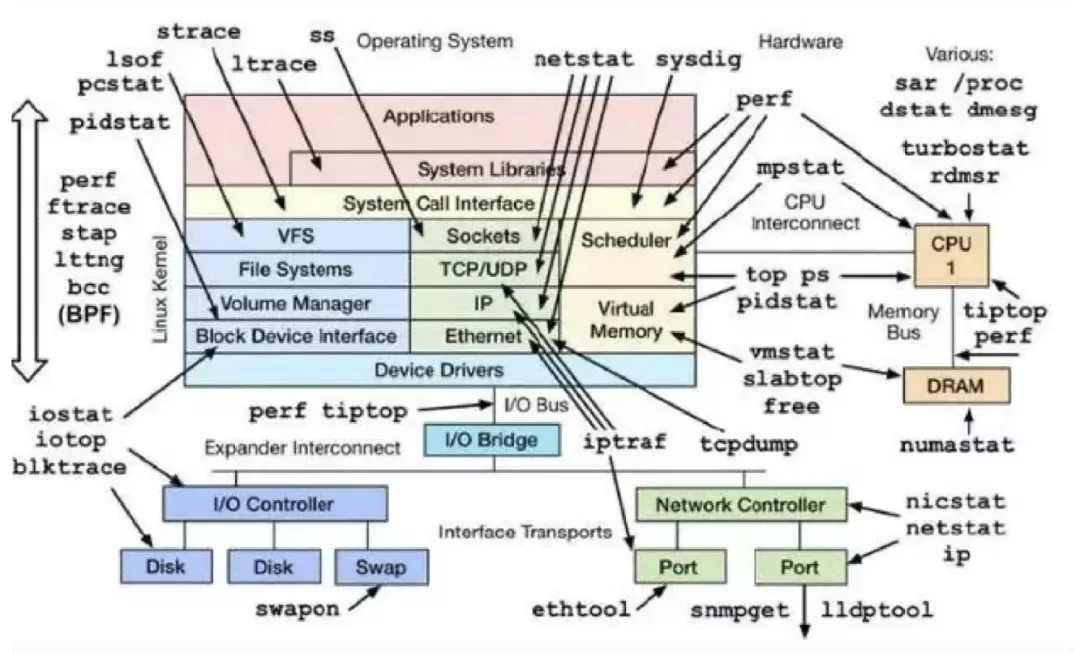
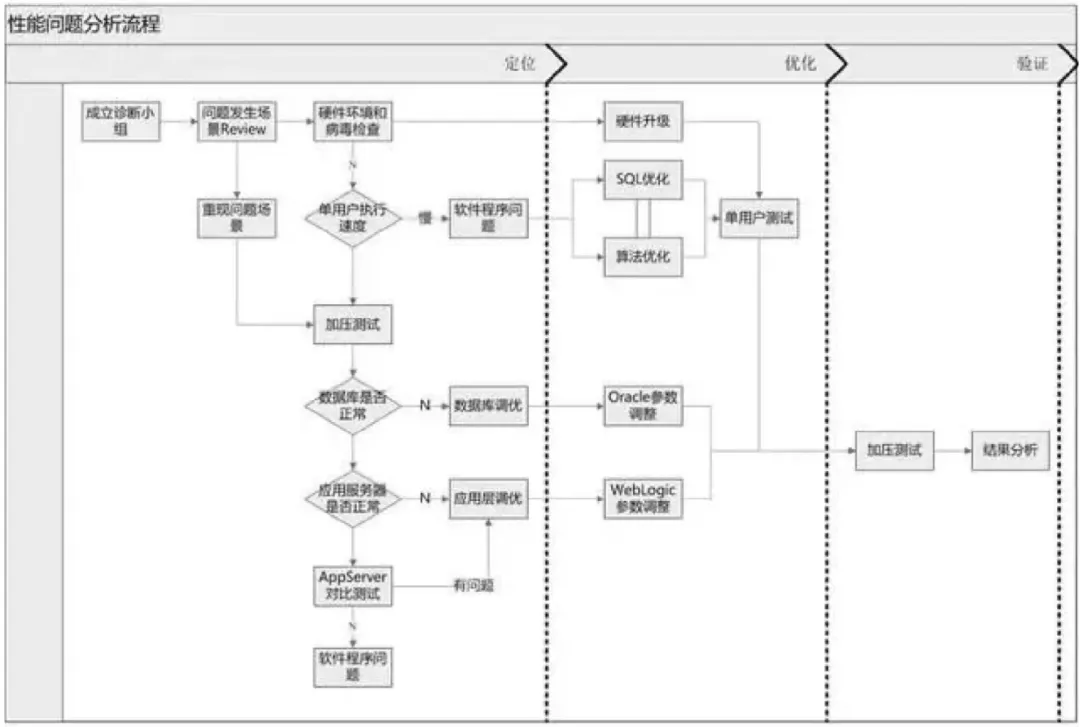
## 系统架构性能优化思路



[今天谈下业务系统性能问题分析诊断和性能优化方面的内容。这篇文章重点还是谈已经上线的业务系统后续出现性能问题后的问题诊断和优化重点。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)



[我们首先来分析下如果一个业务系统上线前没有性能问题，而在上线后出现了比较严重的性能问题，那么实际上潜在的场景主要来自于以下几个方面。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)1：[业务出现大并发的访问，导致出现性能瓶颈](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

2：[上线后的系统数据库数据日积月累，数据量增加后出现性能瓶颈](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

3：[其它关键环境改变，比如我们常说的网络带宽影响](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[正是由于这个原因，当我们发现性能问题的时候，首先就需要判断是单用户非并发状态下本身就有性能问题，还是说在并发状态才存在性能问题。对于单用户性能问题往往比较容易测试和验证，对于并发性能问题我们可以在测试环境进行加压测试和验证，以判断并发下的性能。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[如果是单用户本身就存在性能问题，那么大部分问题都出在程序代码和SQL需要进一步优化上面。如果是并发性能问题，我们就需要进一步分析数据库和中间件本身的状态，看是否需要对中间件进行性能调优。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

在加压测试过程中，我们还需要对CPU，内存和JVM进行监控，观察是否存在类似内存泄漏无法释放等情况，即并发下性能问题本身也可能是代码本身原因导致性能异常。

## **性能问题影响因素分析**



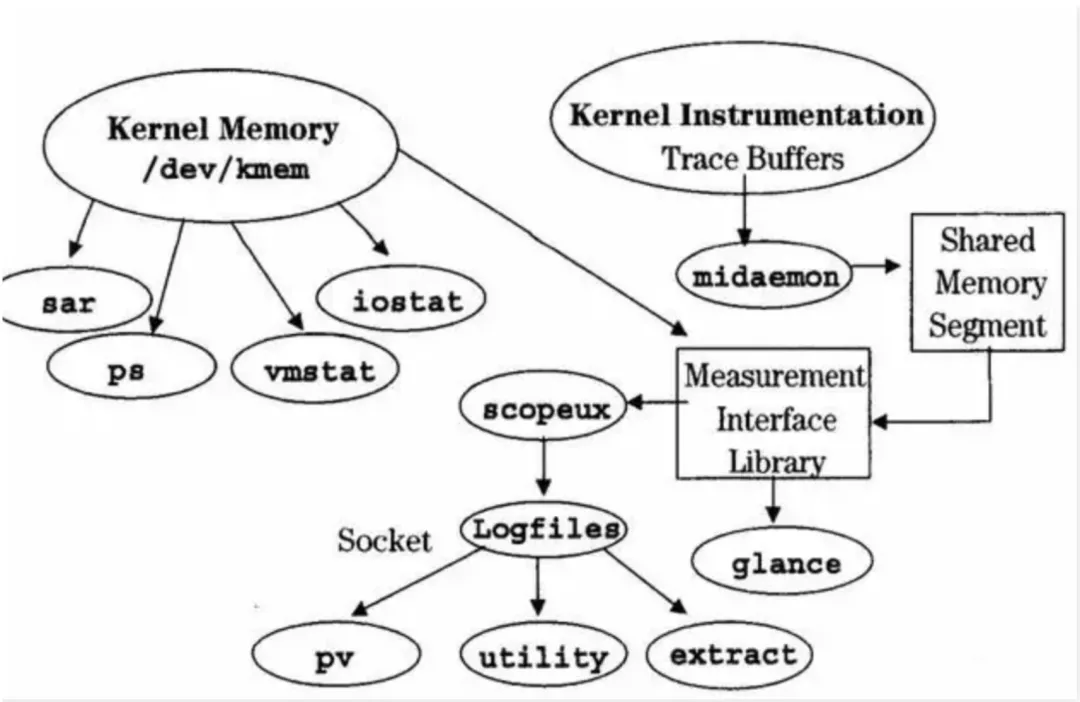
[对于性能问题影响因素，简单来说包括了硬件环境，软件运行环境和软件程序三个方面的主要内容。下面分别再展开说明下。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

###### **硬件环境**

[硬件环境就是我们常说的计算，存储和网络资源。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[对于服务器的计算能力，一般来说厂家都会提供TPMC参数作为一个参考数据，但是我们实际看到相同TPMC能力下的X86服务器能力仍然低于小型机的能力。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

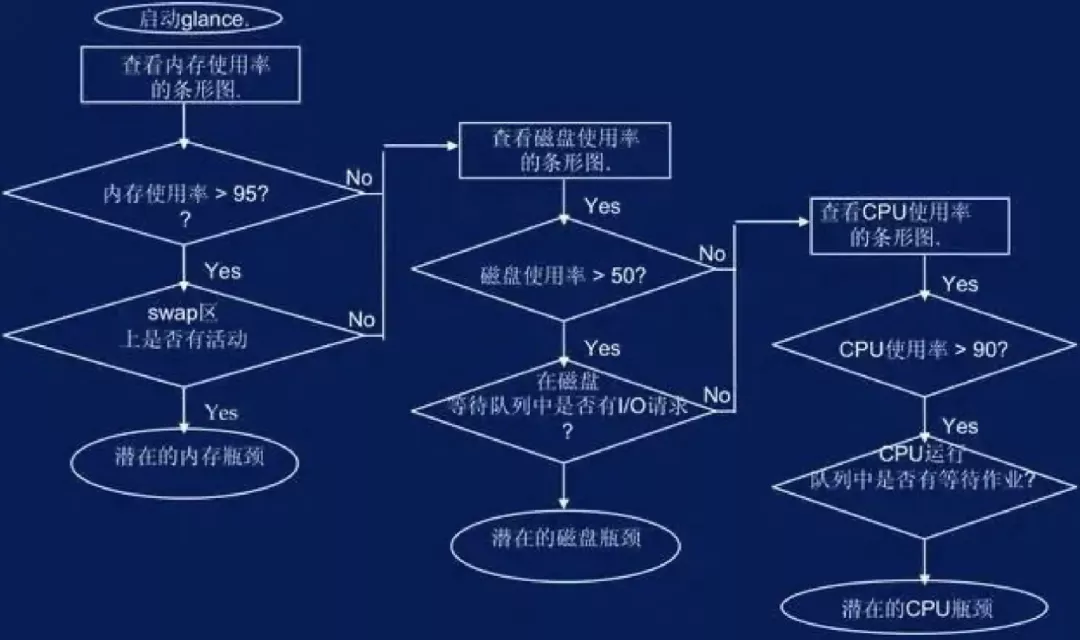
[除了服务器的计算能力参数，另外一个重点就是我们说的存储设备，影响到存储的重点又是IO读写性能问题。有时候我们监控发现CPU和内存居高不下，而真正的瓶颈通过分析反而发现是由于IO瓶颈导致，由于读写性能跟不上，导致大量数据无法快速持久化并释放内存资源。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)



[比如在Linux环境下，本身也提供了性能监控工具方便进行性能分析。比如常用的iostat,ps,sar,top,vmstat等，这些工具可以对CPU，内存，JVM，磁盘IO等进行性能监控和分析，以发现真正的性能问题在哪里。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[比如我们常说的内存使用率持续告警，你就必须发现是高并发调用导致，还是JVM内存泄漏导致，还是本身由于磁盘IO瓶颈导致。搜索公众号互联网架构师回复“2T”，送你一份惊喜礼包。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[对于CPU，内存，磁盘IO性能监控和分析的一个思路可以参考：](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

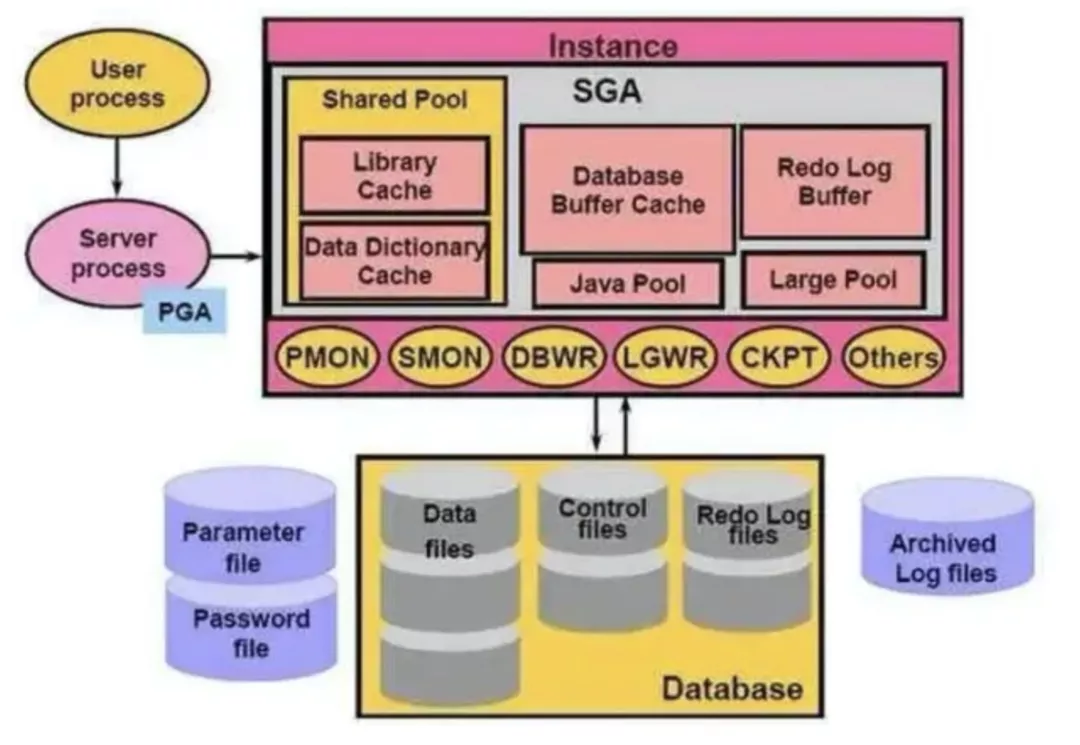


## **运行环境-数据库和应用中间件**

数据库和应用中间件性能调优是另外一个经常出现性能问题的地方。

###### **数据库性能调优**

[拿Oracle数据库来说，影响数据库性能的因素包括：系统、数据库、网络。数据库的优化包括：优化数据库磁盘I/O、优化回滚段、优化Rrdo日志、优化系统全局区、优化数据库对象。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)



要调整首先就需要对数据库性能进行监控

[我们可以在init.ora参数文件中设置TIMED\_STATISTICS=TRUE 和在你的会话层设置ALTER SESSION SET STATISTICS=TRUE 。运行svrmgrl 用 connect internal 注册，在你的应用系统正常活动期间，运行utlbstat.sql 开始统计系统活动，达到一定的时间后，执行utlestat.sql 停止统计。统计结果将产生在report.txt 文件中。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[数据库性能优化应该是一个持续性的工作，一个方面是本身的性能和参数巡检，另外一个方面就是DBA也会经常提取最占用内存的低效SQL语句给开发人员进一步分析，同时也会从数据库本身的以下告警KPI指标中发现问题。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

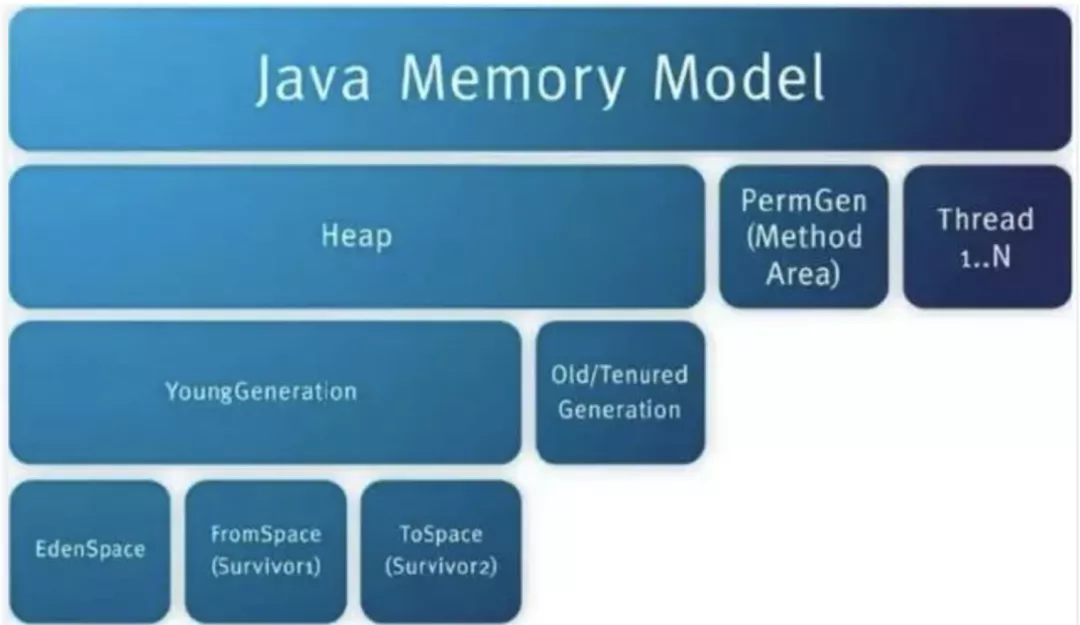
[比如我们可能会发现Oracle数据库出现内存使用率高的告警，而通过检查会发现是产生了大量的Redo日志导致，那么我们就需要从程序上进一步分析为何会产生如此多的回滚。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

###### **[应用中间件性能分析和调优](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)**

应用中间件容器即我们常说的Weblogic, Tomcat等应用中间件容器或Web容器。应用中间件调优一个方面是本身的配置参数优化设置，一个方面就是JVM内存启动参数调优。

[对于应用中间件本身的参数设置，主要包括了JVM启动参数设置，线程池设置，连接数的最小最大值设置等。如果是集群环境，还涉及到集群相关的配置调优。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

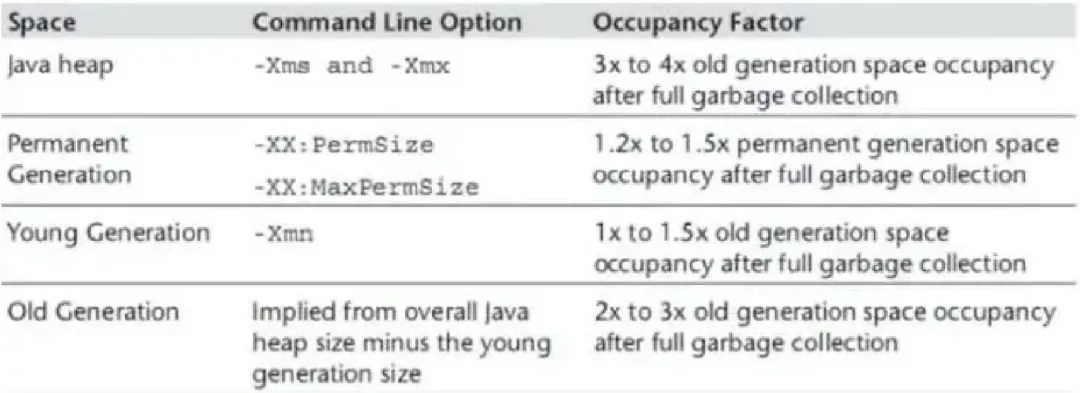
对于JVM启动参数调优，往往也是应用中间件调优的一个关键点，但是一般JVM参数调优会结合应用程序一起进行分析。



[比如我们常见的JVM堆内存溢出，如果程序代码没有内存泄漏问题的话，我就需要考虑调整JVM启动时候堆内存设置。在32位操作系统下只能够设置到4G，但是在64位操作系统下已经可以设置到8G甚至更大的值。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

其中JVM启动的主要控制参数说明如下:

-Xmx  #设置最大堆空间  
-Xms  #设置最小堆空间  
-XX:MaxNewSize #设置最大新生代空间  
-XX:NewSize    #设置最小新生代空间  
-XX:MaxPermSize  #设置最大永久代空间(注：新内存模型已经替换为Metaspace)  
-XX:PermSize     #设置最小永久代空间(注：新内存模型已经替换为Metaspace)  
-Xss   #设置每个线程的堆栈大小

[那么这些值究竟设置多大合适，具体来讲：](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

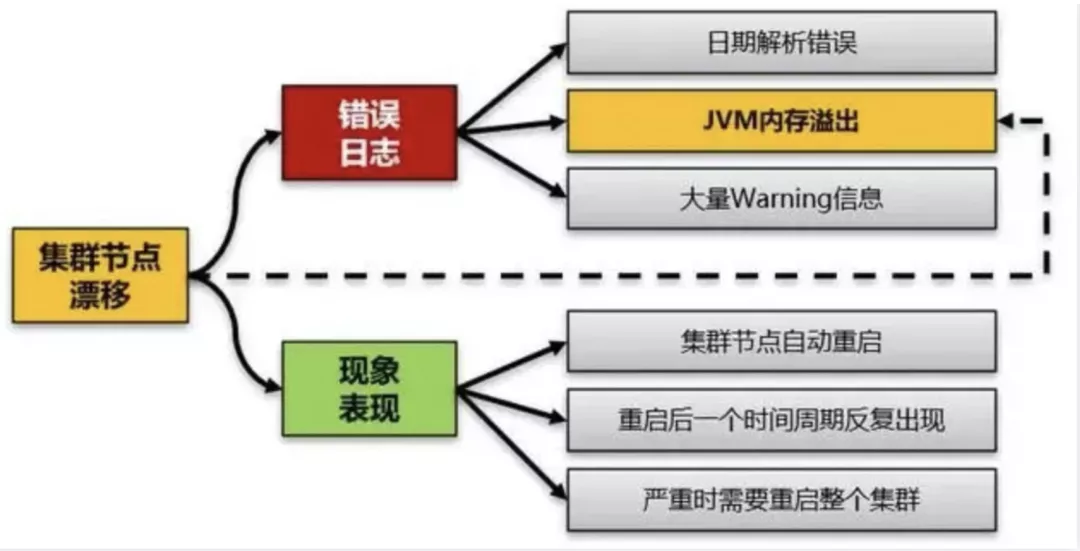
[Java整个堆大小设置，Xmx 和 Xms设置为老年代存活对象的3-4倍，即FullGC之后的老年代内存占用的3-4倍。永久代 PermSize和MaxPermSize设置为老年代存活对象的1.2-1.5倍。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[年轻代Xmn的设置为老年代存活对象的1-1.5倍。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[老年代的内存大小设置为老年代存活对象的2-3倍。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[注意在新的JVM内存模型下已经没有PermSize而是变化为Metaspace，因此需要考虑Heap内存和Metaspace大小的配比，同时还需要考虑相关的垃圾回收机制是采用哪种类型等。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[对于JVM内存溢出问题，我前面写过一篇专门的分析文章可以参考。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[从表象到根源-一个软件系统JVM内存溢出问题分析解决全过程](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

###### **软件程序性能问题分析**

[在这里首先要强调的一点就是，当我们发现性能问题后首先想到的就是扩展资源，但是大部分的性能问题本身并不是资源能力不够导致，而是我们程序实现上出现明显缺陷。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[比如我们经常看到的大量循环创建连接，资源使用了不释放，SQL语句低效执行等。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[为了解决这些性能问题，最好的方法仍然是在事前控制。其中包括了事前的代码静态检查工具的使用，也包括了开发团队对代码进行的Code Review来发现性能问题。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[所有已知的问题都必须形成开发团队的开发规范要求，避免重复再犯。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

## **业务系统性能问题扩展思考**

对于业务系统的性能优化，除了上面谈到的标准分析流程和分析要素外，再谈下其它一些性能问题引发的关键思考。

###### **上线前的性能测试是否有用？**

有时候大家可能觉得奇怪，为何我们系统上线前都做了性能测试，为何上线后还是会出现系统性能问题。那么我们可以考虑下实际上我们上线前性能测试可能存在的一些无法真实模拟生产环境的地方，具体为：

* [硬件能否完全模拟真实环境？最好的性能测试往往是直接在搭建完成的生产环境进行。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [数据量能否模拟实际场景？真实场景往往是多个业务表都已经存在大数据量的积累而非空表。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [并发能否模拟真实场景？一个是需要录制复合业务场景，一个是需要多台压测机。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

而实际上我们在做性能测试的时候以上几个点都很难真正做到，因此要想完全模拟出生产真实环境是相当困难的，这也导致了很多性能问题是在真正上线后才发现。

###### **系统本身水平弹性扩展是否完全解决性能问题？**

[第二个点也是我们经常谈的比较多的点，就是我们的业务系统在进行架构设计的时候，特别是面对非功能性需求，我们都会谈到系统本身的数据库，中间件都采用了集群技术，能够做到弹性水平扩展。那么这种弹性水平扩展能力是否又真正解决了性能问题？](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[实际上我们看到对于数据库往往很难真正做到无限的弹性水平扩展，即使对于Oracle RAC集群往往也是最多扩展到单点的2到3倍性能。对于应用集群往往可以做到弹性水平扩展，当前技术也比较成熟。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[当中间件能够做到完全弹性扩展的时候，实际上仍然可能存在性能问题，即随着我们系统的运行和业务数据量的不断积累增值。实际上你可以看到往往非并发状态下的单用户访问本身就很慢，而不是说并发上来后慢。因此也是我们常说的要给点，即：](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

* [单点访问性能正常的时候可以扩展集群来应对大并发状态下的同时访问](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [单点访问本身性能就有问题的时候，要优先优化单节点访问性能](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

###### **业务系统性能诊断的分类**

对于业务系统性能诊断，如果从静态角度我们可以考虑从以下三个方面进行分类

* [操作系统和存储层面](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [中间件层面（包括了数据库，应用服务器中间件）](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [软件层面（包括了数据库SQL和存储过程，逻辑层，前端展现层等）](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[那么一个业务系统应用功能出现问题了，我们当然也可以从动态层面来看实际一个应用请求从调用开始究竟经过了哪些代码和硬件基础设施，通过分段方法来定位和查询问题。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

比如我们常见的就是一个查询功能如果出现问题了，首先就是找到这个查询功能对应的SQL语句在后台查询是否很慢，如果这个SQL本身就慢，那么就要优化优化SQL语句。如果SQL本身快但是查询慢，那就要看下是否是前端性能问题或者集群问题等。

###### **软件代码的问题往往是最不能忽视的一个性能问题点**

对于业务系统性能问题，我们经常想到的就是要扩展数据库的硬件性能，比如扩展CPU和内存，扩展集群，但是实际上可以看到很多应用的性能问题并不是硬件性能导致的，而是由于软件代码性能引起的。对于软件代码常见的性能问题我在以往的博客文章里面也谈过到，比较典型的包括了。

* [循环中初始化大的结构对象，数据库连接等](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [资源不释放导致的内存泄露等](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [没有基于场景需求来适度通过缓存等方式提升性能](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [长周期事务处理耗费资源](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)
* [处理某一个业务场景或问题的时候，没有选择最优的数据结构或算法](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[以上都是常见的一些软件代码性能问题点，而这些往往需要通过我们进行Code Review或代码评审的方式才能够发现出来。因此如果要做全面的性能优化，对于软件代码的性能问题排查是必须的。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

通过IT资源监控或APM应用工具来发现性能问题



对于性能问题的发现一般有两条路径，一个就是通过我们IT资源的监控，APM的性能监控和预警来提前发现性能问题，一个是通过业务用户在使用过程中的反馈来发现性能问题。[搜索公众号互联网架构师回复“2T”，送你一份惊喜礼包。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

APM应用性能管理主要指对企业的关键业务应用进行监测、优化，提高企业应用的可靠性和质量，保证用户得到良好的服务，降低IT总拥有成本(TCO)。

###### **资源池-》应用层-》业务层**

这个可以理解为APM的一个关键点，原有的网管类监控软件更多的是资源和操作系统层面，包括计算和存储资源的使用和利用率情况，网络本身的性能情况等。但是当要分析所有的资源层问题如何对应到具体的应用，对应到具体的业务功能的时候很难。

[传统模式下，当出现CPU或内存满负荷的时候，如果要查找到具体是哪个应用，哪个进程或者具体哪个业务功能，哪个sql语句导致的往往并不是容易的事情。在实际的性能问题优化中往往也需要做大量的日志分析和问题定位，最终才可能找到问题点。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[比如在我们最近的项目实施中，结合APM和服务链监控，我们可以快速的发现究竟是哪个服务调用出现了性能问题，或者快速的定位出哪个SQL语句有验证的性能问题。这个都可以帮助我们快速的进行性能问题分析和诊断。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

[资源上承载的是应用，应用本身又包括了数据库和应用中间件容器，同时也包括了前端；在应用之上则是对应到具体的业务功能。因此APM一个核心就是要将资源-》应用-》功能之间进行整合分析和衔接。](http://mp.weixin.qq.com/s?__biz=MzI2MTIzMzY3Mw==&mid=2247487508&idx=1&sn=78cf235aa9ba5f988c6922ca98f8bfd6&chksm=ea5cdd72dd2b54647cf55b4a73dcafa69fc7228205ad39ecc98fe57b39cdecb21c238c6d6cb6&scene=21" \l "wechat_redirect" \t "https://mp.weixin.qq.com/_blank)

而随着DevOps和自动化运维的思路推进，我们更加希望是通过APM等工具主动监控来发现性能问题，对于APM工具最大的好处就是可以进行服务全链路的性能分析，方便我们发现性能问题究竟发生在哪里。比如我们提交一个表单很慢，通过APM分析我们很容易发现究竟是调用哪个业务服务慢，或者是处理哪个SQL语句慢。这样可以极大的提升我们性能问题分析诊断的效率。