事务中间件 CICS 原理及应用开发

About this blog

IBM 交易中间件 CICS TS (Transaction Server) 产品具有超过 40 年的历史，是全球最有影响力的 IT 产品之一。世界 500 强中超过 90% 的公司在使用 CICS 处理他们的核心交易，全球每天通过 CICS 处理的交易

浅谈CICS性能分析 之 part1

CICSChina | Dec 4 2014 | Visits (12320)

CICS Transaction Server是IBM主机系统的核心交易中间件，目前已经有50年的历史。在50年的岁月中，CICS被证明是是极为稳定，高效的。套用一句经典台词，在交易处理领域，CICS一直被模仿，从未被超越…

回归正题，CICS目前在我国主要应用于银行系统，国内的大型银行和几个中小型银行每天都在使用CICS处理其核心业务，CICS的性能对于客户来说至关重要。刚刚过去的2014双十一，各家银行的交易量都再创新高，CICS以其优异的表现再次通过了这次年终大考。

在日常工作中，我有幸接触到很多主机领域的团队。很多同仁渴望了解更多CICS性能分析的知识，这也是我写这篇博客的初衷。在下面的篇幅里，我将介绍CICS的主要性能指标以及对CICS性能指标进行分析的方法。经验有限，知识浅薄，希望各位看官不吝斧正。

1.在一切开始之前

大家可能感觉奇怪，为什么单独设置这么一个小标题。我曾与一位资深的CICS英国专家闲聊，他提到了一个很值得深思的现象。在CICS中国实验室成立初期，他多次来中国参与客户的CICS调优和安全检查。他发现很多中国工程师特别喜欢“钻研”，但是过于钻研的结果就是因为要看到很小的一个点，却忽略了整个面。对于他的这次“挑衅”，我还是比较认同的，我们经常希望把很多细节搞清楚，结果不自觉地就陷到了细节里去了，可谓“不识庐山真面目，只缘身在此山中”。所以，有的时候看问题，退一步，也许反而容易发现问题的本质。

我接触的性能问题大体可以分为两类：“火烧眉毛型”和“悠然自得”型。“火烧眉毛型”代表着客户出现了严重的性能问题，例如核心交易响应时间很长，又或者CICS系统频繁触发MXT。这种情况客户往往很焦急。“悠然自得型”呢，就舒服多了，往往是客户希望定期对系统进行安全检查，或者希望对系统进行性能调优。这两种类型舒适度不同，但是性能分析的方式并没有大的差别，在后面的叙述中，我将侧重于第一种情况下的处理。

“火烧眉毛”的情况，很多同仁会很快进入细节分析，想尽快找到问题的原因。我的建议是在进行分析之前，先问自己几个问题：

1.系统出现的具体性能问题是什么？CPU使用增加？响应时间增加？

2.系统性能问题有没有一个模式？问题是发生在特定的LPAR，甚至是特定的CICS Region，还是整体都出现了问题？问题是发生在特定的时间点，时间段，还是随机的？问题是只发生了一次，还是频繁发生？

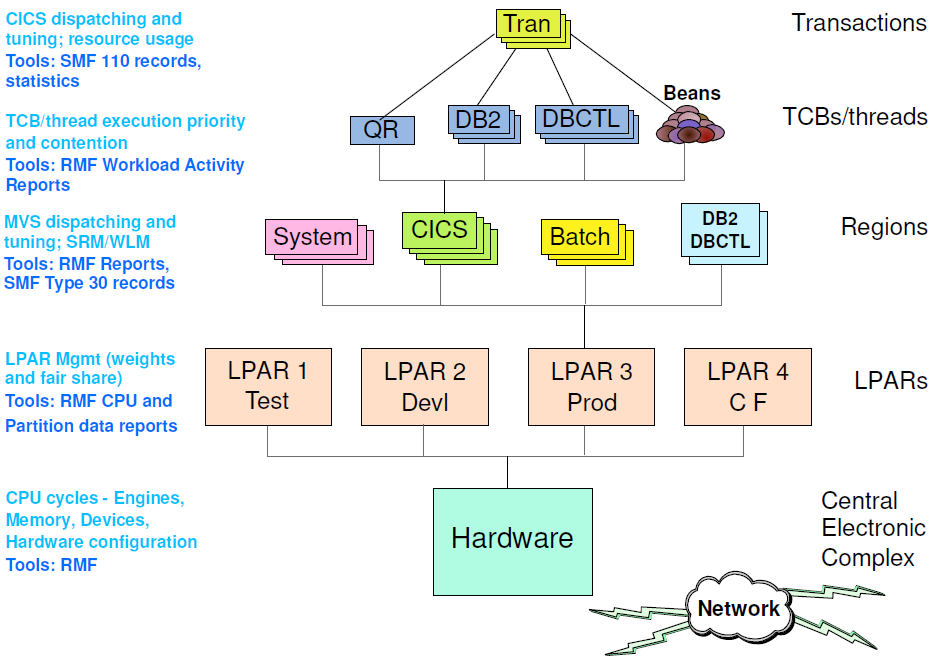
3.什么时候开始出现这样的问题？在问题发生之前，是否发生了应用变化，软件变化（包括PTF等运维变化）或者硬件变化？

先搞清上诉几个问题将为我们后面的性能分析提供一个更加明确的方向，有的放矢，才能事半功倍。

举个简单的例子，一个客户系统性能出现了问题，我们首先搞清楚客户响应时间增加了。通过进一步了解，我们发现客户的问题发生在特定的一个LPAR，特定的一个时间段，每天都会发生，影响范围是这个LPAR的所有的在线交易。到这里问题变得清晰起来，我们可以怀疑是不是这个LPAR有什么特殊的工作负载（批量应用？），在出问题的时间段，抢走了LPAR的资源（CPU？内存？），然后影响了所有在线交易。我们了解到果然这个LPAR在这个时间段会跑一个新上线的批量作业。问题的原因已经浮现出来，无需去分析CICS的SMF，RMF等数据了。这个例子很简单，但是我相信相当一部分性能问题是可以通过这样的思考发现并解决的。即使问题依然不清楚，在我们心中对于问题以及问题可能的原因也会有更好的判断。

1.如何获得主机系统性能数据

确定性能问题的方向之后，收集性能数据是非常重要。主机系统及其各个组件会统一将数据写入SMF（RMF数据也是记入SMF）。下面一幅主机系统的性能层次图，图中将主机系统的性能分为5个层次。每个层次的性能都依赖于下面层次提供的资源。在进行性能分析的时候，脑子中有这样的层次概念是很有帮助的。

图1： 主机系统的性能层次图

1.Transactions

交易这一层次是最为直观的层次。性能问题出现之后往往直观反应在交易性能上。这也是为什么CICS往往是主机系统性能分析的起点。本文章后面的章节也主要集中在交易分析这个级别。

在交易进行过程中，CICS可以记录每笔交易的性能数据，记录性能的组件被称作CICS Monitoring Facility，或者CICS Monitor。CICS Monitor会将交易过程中的性能数据记录到MVS的SMF 110（Type 1s）里，最为常见的性能数据包括交易在CICS里调度各个阶段的响应时间。CICS Monitor可以通过系统启动参数进行配置，也可以在CICS运行状态下通过CEMT，CEMN或者CICS SPI动态打开或者关闭。更多信息，可以参考IBM Knowledge Center。

http://www-01.ibm.com/support/knowledgecenter/SSGMCP\_5.2.0/com.ibm.cics.ts.doc/lpaths/workload\_lp\_cmf.html?lang=en

处理Monitor性能数据之外，CICS还会记录统计统计信息，也就是CICS Statistics。CICS Statistics被记录在MVS的SMF 110（Type 2s）。如果说CICS Monitor关注的是交易级别的性能数据的话，CICS Statistics则关注一段时间间隔内CICS整体的系统资源的使用及调度情况，系统资源包括交易，程序，TCB，内存，DB2线程，TDQ/TSQ，VSAM文件等。可以说，CICS Monitor关注整体，是系统健康状况的总结，也常常作为系统容量评估的依据。

CICS statistics主要分为两类，Interval statistics和End-of-day statistics。通CICS Monitor一样，CICS支持在运行状态下动态的设置并收集statistics。更多信息，可以参考IBM Knowledge Center。

http://www-01.ibm.com/support/knowledgecenter/SSGMCP\_5.2.0/com.ibm.cics.ts.performance.doc/topics/dfht3\_stats\_intro.html?lang=en

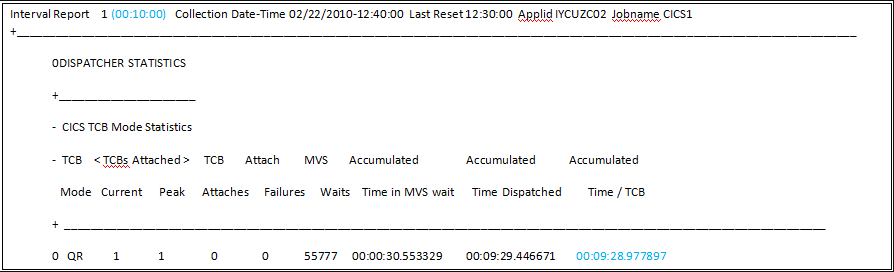
1.TCBs/Threads

TCBs或者Threads这一层是MVS的调度单元，MVS通过预先设置好的调度优先级为TCB或者Thread分配CPU。

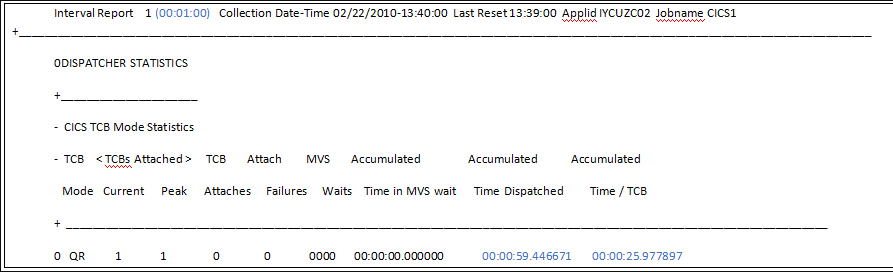
CICS最为重要的TCB就是QR TCB。在CICS大范围支持OTE（Open Transaction Environment）之前，大部分的应用逻辑都运行在QR TCB，由CICS的dispatcher组件进行子调度。即使CICS通过OTE支持将threadsafe的应用跑在Open TCB上（L8，L9，T8等），应用交易的发起和结束也需要通过QR进行处理。如果遇到一些non-threadsafe的API，应用也需要从Open TCB切换到QR TCB。可以说，QR TCB是CICS的生命线，QR TCB的健康状况会直接影响CICS的性能。

QR值得关注的QR的两项指标是QR的繁忙程度和QR CPU使用率。

比较常见评估QR繁忙程度的方法是在CICS Statistics里面找到Accumulated Time/TCB。这个时间是在CICS统计的Interval里面QR实际使用的CPU时间。然后使用这个时间除Interval的时间。例如下面的例子中，在CICS Statistics报告的统计Interval是10分钟。在Dispatcher部分，我们可以看到QR TCB的统计信息。其中QR CPU实际使用时间是9:28.977897，所以这个CICS region的CPU资源很充足，但是QR的繁忙程度达到95%，这意味着这个CICS基本饱和。



上面的例子中我们可以看到Accumulated Time/TCB和Accumulated Time Dispatched非常接近，这说明MVS在调度这个QR TCB的时候保证了其CPU。但是当系统繁忙时，LPAR之间，当LPAR里的地址空间之间，都会争抢有限的CPU资源，进而出现CPU contention的现象。具体反映到CICS里面，就是QR TCB虽然处于Dispatched状态，但是却只能在一部分时间有CPU使用。我们通过QR CPU使用率来衡量CPU contention的验证程度。例如下图中，我们可以看到在1分钟的interval里面，Accumulated Time/TCB和Accumulated Time Dispatched差别就比较大了，两者相除，我们可以得到QR的CPU/Disp比例也就是CPU使用率是43%。



提示：上面的Statistics报告出自比较早期的CICS版本，里面的域名称在新版本中有一定变化。具体请参考IBM Knowledge Center。

http://www-01.ibm.com/support/knowledgecenter/SSGMCP\_5.2.0/com.ibm.cics.ts.performance.doc/topics/dfht3\_stats\_disp\_tcbmode\_summary.html?lang=en

Open TCB在CICS被大量使用，尤其是在CICS对其提供的API/SPI进行大规模threadsafe改造之后，客户可用通过将threadsafe的应用更多的跑在Open TCB，释放对QR TCB的压力，从而提高吞吐量。Open TCB里，L8 TCB常用来运行DB2 TRUE，实现与DB2的交互。CICS提供的DB2CONN和DB2ENTRY定义中，提供PRIORITY选项来设置Pool thread TCB和Entry thread TCB较QR TCB的调度优先级，用户可根据实际情况进行设置。我个人的建议是优先保证QR TCB的CPU，如果QR TCB CPU充足的情况下，可以适当提高DB2 thread TCB的优先级，缩短与DB2交互交易的响应时间。每个用户的情况不同，如何设置最好通过压力测试来论证。

1.Regions

Regions这一层就是MVS的地址空间（address space）对应的层次。每个CICS Region对应一个地址空间，同样Batch, DB2和MQ也都拥有自己的地址空间。

一般情况下，CICS性能分析很少需要到这个层次，除非我们发现系统的性能问题来自于地址空间本身的资源不足，例如无法保证CPU，内存SoS。造成地址空间的资源不足往往和竞争有关，这里的竞争包括通过CICS之间的竞争，CICS与Batch的竞争，CICS与DB2的竞争等。所以在系统资源紧张的时候，合理的在z/OS WLM里设置各地址空间的调度优先级就非常重要了。关于z/OS地址空间优先级的设置见仁见智，下面的优先级排序仅供各位看官参考。

TCPIP & VTAM & IRLM > All TORs and MQCHINIT > (DB2MSTR, DB2DBM1, DB2DIST) = MQMSTR > AOR  > Batch

如果希望看到地址空间这一层次的资源调度情况，可以查看RMF包括，或者SMF Type 30记录。SMF Type 30记录了Region级别的统计信息。

1.LPARs

如果我们发现系统资源的竞争已经超越了本LPAR的地址空间，而是出现了跨LPAR的资源竞争。那么我们的性能分析就进入了LPARs这个层次。z/OS通过LPAR权重为各个LPAR分配硬件资源，如CPU。通过RMF的CPU报告和Partition data报告，我们可以清楚的看到各个LPAR的资源使用情况，是否已经到达了极限，如CP Fair Share。

1.Hardware和Network

Hardware和Network是最后一个层次。如果性能分析锁定了这个层次，用户可能需要考虑扩容了。RMF依然是分析这个层次问题的主要工具。

**3.如何理解CICS Monitor性能数据**

在上一部分，我们介绍了性能问题的层次结构以及在各个层次性能数据的记录位置。由于章节有限，在这部分我将主要针对Transaction这个层次，介绍分析CICS Monitor性能数据的方法。应该说，大部分我所接触到的性能问题都可以通过分析CICS Monitor性能数据得以解决。

若想理解CICS Monitor性能数据，首先需要理解CICS是如何调度task的。Task是交易在CICS内部的实体，是CICS进行调度的单位。MVS级别的调度是为不同的TCB分配CPU资源，而CICS级别的调度则是为CICS的task分配TCB资源，所以我们也管CICS的调度叫做子调度。

**CICS是如何在TCB上进行调度task的呢？**

用户的应用一般会运行在CICS的QR TCB和Open TCB上。

**首先介绍task在QR TCB上的调度。**图2显示一个task在QR TCB上运行状态。从CICS的角度，task的状态分为两部分，Dispatch状态和Suspend状态。

Dispatch状态始于CICS DSP动作，CICS在QR上调度并运行task。虽然从CICS角度，Dispatch状态的task一直在使用QR TCB，但是这并意味着task可以一直拿到CPU资源来实际运行指令。Task能够拿到多少CPU取决于MVS的调度。从图2中可以看到，当硬件中断发生的时候，Dispatch状态的task也会失去CPU。CICS Monitor Facility会记录这些信息在SMF里面，我在后面会更加仔细的介绍。所以进一步细分，**Dispatch状态可以分为运行CPU的时间和等待CPU的时间。**根据经验，QR的CPU时间除以Dispatch时间最好大于50%，70%以上代表着良好的CPU状态。

当task需要等待某些资源（包括等待CICS资源，调用系统组件进行IO，等待外部组件返回等）时，CICS 通过CICS WAIT将task挂起（Suspend），task进入Suspend。CICS的QR上面有一个task的队列，里面按照优先级排列需要被CICS调度的task。在挂起当前task之后，CICS会继续调度task队列里其他的task。所以对MVS来说，TCB并没有处于WAIT状态，CICS的Suspend机制是一种逻辑上的挂起，挂起的task，而不是TCB。当被挂起的task等到了资源，CICS会自动将其唤醒，并将该task按照优先级放置到task的调度队列里。所以task被唤醒之后，存在一定的间隔被再次调度，这段间隔的长短和QR的繁忙程度有关，而这段再次调度的等待时间也被计入到Suspend时间里。所以简单来讲，**Suspend状态也可以细化为等待资源时间和等待再次被调度的时间。**

**Task在Open TCB上的调度和**QR稍有不同。图3以L8/L9为例，显示了task在Open TCB上调度过程。对比图2和图3，主要区别就是CICS WAIT变成了MVS WAIT，MVS WAIT之后的CICS DSP变成了MVS DSP。Task没有suspend状态。造成这一系列变化的本质原因是Open TCB在同一时间段只调度一个task，也没有QR那样的task队列。当task需要等待资源的时候，CICS无需调用CICS WAIT对task做逻辑上的挂起（Suspend），而是调用MVS WAIT，所以唤醒task的方式也只能是MVS DSP。总结来说，Open TCB是可以容忍等待，task进入等待意味着TCB也随之进入等待，这也是其与QR TCB在task调度上的本质差别。如果是CICS通过精细化的管理让QR TCB尽可能的处于繁忙状态的话（避免MVS WAIT的发生），那么Open TCB的使用则显得“简单粗暴”。

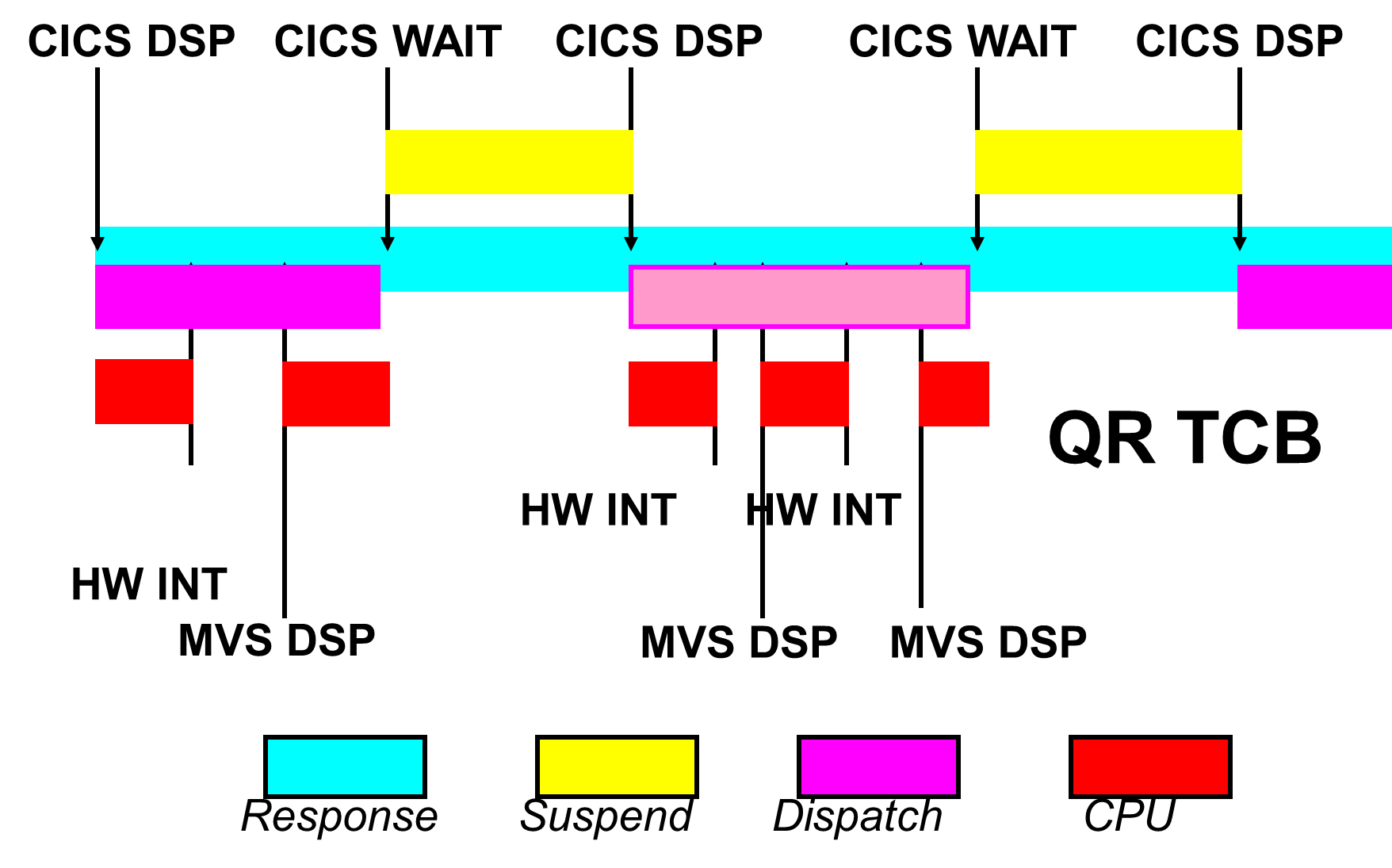
[](https://www.ibm.com/developerworks/community/blogs/cicschina/resource/BLOGS_UPLOADED_IMAGES/123454.png)

图2： Task在QR TCB的调度

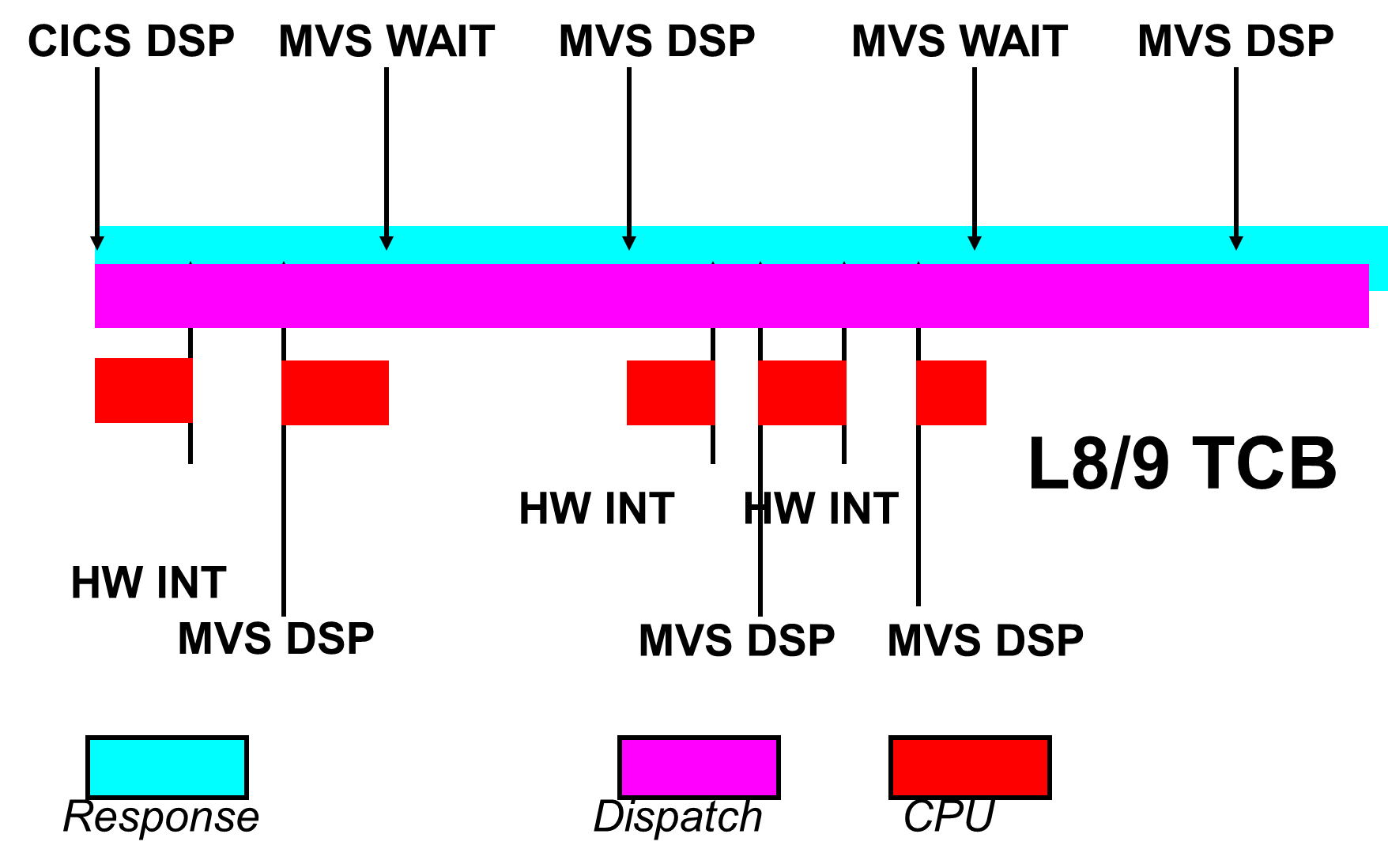
[](https://www.ibm.com/developerworks/community/blogs/cicschina/resource/BLOGS_UPLOADED_IMAGES/123455.png)

图3： Task在Open TCB的调度

**讲了task在不同TCB上的调度，就不能不提Change Mode。**Change Mode指一个task在生命周期里在不同TCB mode之间的切换。我前面提到所有的task的开始和介绍一定是需要在QR上运行的。当task需要访问DB2，MQ等外部资源管理器，或者我们希望通过使用OTE实现更大吞吐量的时候，task会在生命周期中跑到Open TCB上。在不同的TCB之间进行切换，task需要一段时间等待切换的完成并被调度在目标TCB上。CICS Monitor Facility会记录task在生命周期里Change Mode的次数以及总的等待时间。除了等待时间以外，因为每次TCB的切换还会额外消耗大概2000条的系统指令，所以应用调优的一个重要原则就是减少不必要的TCB切换。

**那么CICS Monitoring的性能指标有哪些？**

在具体介绍CICS性能指标之前，我先介绍一个CICS工具，CICS Performance Analyzer，简称CICS PA。CICS PA是分析CICS性能的标准分析工具。用户可以利用PA对CICS记录的SMF数据（甚至是DB2，CTG，MQ的数据）进行格式化处理，抽取并分析相关性能指标，并产生性能报告。据我所知，CICS PA已经是CICS客户的标配工具。想了解更多CICS PA信息，请参考IBM Knowledge Center：

<http://www-01.ibm.com/support/knowledgecenter/SSPPU4/welcome.html?lang=en>

那么下面我们看看CICS的性能指标有哪些。图4显示了CICS里task的响应时间的组成部分。

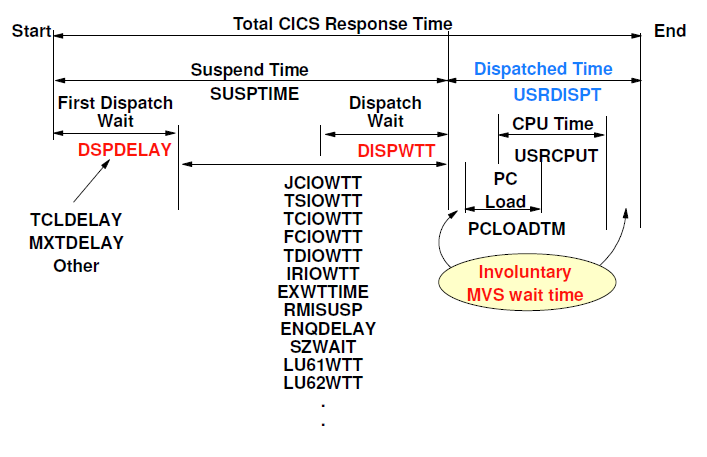
[](https://www.ibm.com/developerworks/community/blogs/cicschina/resource/BLOGS_UPLOADED_IMAGES/123456.png)

图4： CICS  task响应时间组成部分

Task从开始到结束的总的时间就是task的Response Time。Response time进一步可以细化为Suspend Time和Dispatched Time。我在前面的章节里面讲过了，QR mode下是存在Suspend时间的，但是Open TCB mode下所有的时间都是Dispatched Time。

Dispatched Time在CICS PA里对应着USRDISPT，包含着所有TCB mode下Dispatched 时间的总和。用户可以通过PA的QRDISPT，L8DISPT等指标查看不同TCB mode下的具体Dispatched时间。USRDISPT里面进一步包含task时间运行CPU的时间，对应PA的USRCPUT。同样USRCPUT也可以按照TCB mode进行区分。我常常用QRCPUT/QRDISPT得到task运行过程期间QR使用CPU的状态。如果QRCPUT/QRDISPT比较小，说明CPU资源很可能出现了争抢，QR无法获得足够的CPU。还有一种情况就是QR陷入了长于预期的MVS WAIT，丢掉了CPU资源。

Suspend Time在CICS PA里面对应着SUSPTIME。SUSPTIME可以进一步细化为First Dispatch Wait (DSPDELAY) 和各种资源等待的WAIT。

DSPDELAY是task从开始进入CICS到被调度在TCB上的总的时间。影响DSPDELAY的因素包括MXT，TCLASS以及CICS QR TCB上已有的task（更具体说就是正在调度和优先级更高的task）。

资源等待的WAIT包括等待Journal IO的时间（JCIOWTT），等待TSQ IO的时间（TSIOWTT），等待terminal键入的时间（TCIOWTT），等待文件IO的时间（FCIOWTT），等待TDQ IO的时间（TDIOWTT），等待MRO通信的时间（IRIOWTT），等待APPC通信的时间（LU62WTT）等等。我在这里就不一一列举了。关于这些指标的具体介绍，请参照CICS PA用户说明。CICS记录的资源等待时间是同类型的等待时间的加和，是累计值。

资源等待的WAIT时间里面主要包含两部分，一部分实际等待资源的时间，例如IO时间，通信返回时间等，还有一部分是容易被忽略的，就是重新被调度在TCB上的时间，DISPWTT。我前面讲过QR TCB上存在task队列的，如果当前task在被挂起之前在QR TCB上运行，那么其等待的IO或者通信返回之后，该task并不是马上可以被调度在QR TCB上，而是需要等待task队列里高优先级的task完成调度。当QR繁忙的时候，DISPWTT往往是资源等待WAIT的主要组成部分。如果当前task被挂起于Open TCB，DISPWTT基本没有，因为Open TCB在同一时间段只服务一个task。CICS记录的DISPWTT也是累加值，是所有因为资源等待被挂起后等待再次被调度TCB的时间的加和。

**如何锁定性能瓶颈？**

前面我介绍了CICS Monitoring性能数据有哪些指标，其含义是什么。那么一旦出现了性能问题，我们应该如何入手呢？

在文章的开始，我提到在分析问题前一定要极可能的搞清楚问题发生的背景，这包括**系统出现的具体性能问题是什么？系统性能问题有没有一个模式？什么时候开始出现这样的问题？**如果性能问题是响应时间的问题，就可以通过PA有目的性的分析CICS Monitoring数据。如果性能问题是CPU消耗的问题，那就需要借助类似于APA的工具找到CPU消耗的瓶颈。下面我们主要针对响应时间的问题。

响应时间问题的分析方法和具体问题有关，具体执行也是因人而异。我建议的一个原则是，从面到点，从粗到细。如果手头有CICS Statistics，不妨先了解下CICS的整体情况，然后分析CICS Monitoring数据的时候，先用PA打一些总体的报告，例如针对交易的WAIT分析报告，然后再去分析具体问题。有的时候通过PA可以锁定性能的瓶颈，具体的原因可能还需要借助更加细粒度的trace（CICS trace，system trace）。

**一点点结束语**

不知道读到这里各位看官是否有所收获。CICS的性能分析是一项复杂的任务，良好的大局观以及对CICS的基本运行原理的理解是这项任务的基础。如果您在CICS使用过程中遇到性能问题，不妨翻出这篇文章自己练习下。不断的实战是提升分析能力的必经之路。

（未完待续）

 作者： 周北春

 联系方式： beichunz@cn.ibm.com

Add a Comment

More Actions

Comments (0)

Add a Comment

More Actions

There are no comments to display

Previous Entry

Main

Next Entry

About

Help

Contact us

Submit content

Feeds

Newsletters

Follow

Like

Report abuse

Terms of use

Third party notice

IBM privacy

IBM accessibility

Faculty

Students

Business Partners

Select a language:

English

中文

日本語

Русский

Português (Brasil)

Español

Việt