Web 性能测试指标参考[1]

[1]Author：beiyu95 MSN: beiyu95@hotmail.com)

1 Web 应用性能度量

1.1 三个术语

1.1.1 负载

负载是网站所承受的压力总量。它始终会使我想到软水管，可能已经关闭，只能细细地流出水滴；也可能打开到了最大，让水流奔涌而出。对于网站，我们常常从并发用户的角度来讨论负载问题，这并不一定意味着每个用户都在完全相同的时刻请求某个网页，这其实是一种常见的误解。最好是在一定的时间范围内来考虑负载问题；例如，在特定的时间范围内访问该站点的用户数量，也许是以五分钟作为时间间隔，也许是每小时。

1.1.2 响应时间

响应时间是门户或者站点对请求做出响应所花费的时间。从浏览器的角度来看，这是真正的端到端的时间，并且通常不包括浏览器生成或显示页面所花费的时间。可以考虑到，随着站点负载的增加，响应时间通常将会发生变化（它可能会增加），最终可能增加到用户无法接受的情况。响应时间是一项受到普遍关注的度量，而您的最终目标是对门户进行优化，以便在预期的用户负载情况下，提供一致的响应时间范围。响应时间目标需要遵循相关的行业标准；例如，站点的目标可能是在五秒钟内响应 95% 的页面请求。

1.1.3 吞吐量

吞吐量是门户响应请求的速率。通常，我们将其视为系统的点击率或者页面速率，并且页面速率度量更加合适。吞吐量，再加上响应时间和用户的活动模型，能够帮助您确定您的系统在给定的时间范围内可以处理（负担）多少用户。通常，吞吐量的测量是相对于负载情况的，确定随着用户负载的继续增长所出现的系统边界。

1.2 系统度量指标

1.2.1 Running users

登入系统的每个用户拥有不同的状态，以各自的方式消耗着系统资源。系统能够支持的用户数是系统容量的重要标志。
系统在各种用户场景下的性能表现可以让我们更好的了解系统的性能特点，对系统做出准确的评估。

1.2.2 点击率 Hits/sec

每秒的请求数。也就是客户端在单位时间内向服务器发送的请求数。包括各种对象请求（包括图片、CSS 等）。

1.2.3 页面速率 Pages/sec

一个页面通常由大约几十次请求（点击）（因为浏览器需要逐个地下载图像）以及嵌入该页面的其他静态组件组成。最终用户并不真正关心这种方式的点击。获得一半的页面并没有什么意义，只有在获得了整个页面后，用户才会感到满意。测量页面速率可以帮助您摆脱混淆，并且使您更清楚地了解门户中所发生的情况。

示例：

图 1. 通用的每秒页面请求速率的公式

$$\frac{\text{\# users/hour} \times \text{\# pages/user}}{60\ \text{min/hour}} \div 60\ \text{sec/min} = \text{\# pages/sec}$$

要使用这个计算，您需要掌握一些关键的值。首先是在给定时期内访问该服务器的预期用户数量；这就是您对该门户所施加的负载。例如，假设您有 45 个用户（或者 vuser，视情况而定），这些用户可以在 5 分钟内运行脚本。那么可以得到下列信息：

在 5 分钟内有 45 个用户 = 45 * 12（一小时中有 12 个 5 分钟）= 一个小时内有 540 个用户

如果每个虚拟用户都通过脚本访问 8 个页面，那么您可以确定具体的页面速率：

图 2. 每秒页面速率的示例

$$\frac{540\ \text{users/hour} \times 8\ \text{pages/user}}{60\ \text{min/hour}} \div 60\ \text{sec/min} = 1.2\ \text{pages/sec}$$

1.2.4 每秒事务数 Trans/sec

单位时间内用户定义的 transaction 数量。表明单位时间内系统可以完成多少个定义的事务，在一定程度上反应了系统的处理能力。

1.2.5 Http response

http 请求应答情况，包括各种 http 应答的情况。如 200，302，500 等。

1.2.6 Errors

在进行测试的过程中，工具可能会给出一些错误信息。我们应该对这些提示信息给予足够的重视。来仔细查看错误具体属于哪种情况，是否暴露了性能问题或是功能缺陷。一般而言，错误信息可能会包含以下几种：

- ✧ 脚本相关的错误
  如脚本处理不当、环境配置等。
- ✧ 测试工具相关的错误
  测试工具本身的 bug 或是选项配置等。
- ✧ 系统出现了性能问题
  如资源瓶颈、程序效率低下导致超时等。
- ✧ 系统在压力下暴露出了一些功能性错误

1.3 指标值的统计方法

- 最大/小值、平均值
  （略）
- 标准偏差
  定义请参加相关统计学文档。基本上我们可以理解为这个值越大，结果的误差将越大。
- 90%线
  是指 90%的用户完成该操作的时间。

2 Linux 性能与度量

2.1 Linux 常用性能参数

2.1.1 Memory

- Swapd
  使用的虚拟内存的大小（KB）
- Free
  空闲内存的大小（KB）
- Buff
  用作 buffer 的内存大小（KB）
- Cache
  用作 cache 的内存的大小（KB）

注：
查看某个进程占用内存的比例，可以使用 ps aux 来查看，或者和 sed 等工具结合进行一段时间内的数据收集与处理。

2.1.2 CPU

- User time
  运行在非核心态模式的时间比
- Sys time

运行在核心模块是的时间比

- Idle
  处于空闲状态的 cpu 时间比（linux2.5.41 之前的内核版本，这个值包含了 IO 等待的时间）
- Wait
  系统等待 IO 的时间（linux2.5.41 之前的内核版本，这个值为 0）
- Loadaverage
  Load average 是处于 TASK_RUNNING 状态和 TASK_UNINTERRUPTIBLE 状态的进程队列阶段性的平均值。Uptime 和 top 命令给出的 loadaverage 在过去的 1 分钟、5 分钟、15 分钟间隔内的平均值。

2.1.3 IO

- Bi
  向块设备写数据的速率（blocks/s）
- Bo
  从块设备中读取数据的速率（blocks/s）

2.1.4 Swap

- Si
  从磁盘 swap in 内存的数据速率（KBps）
- So
  从内存 swap out 到磁盘的数据速率（KBps）

2.1.5 System

- In
  每秒中断数。
- Cs
  每秒的上下文切换。

注：big brother 监控图像中，bi 是读磁盘速率，单位是 KB，bo 是写磁盘速率，单位也是 KB。和 iostat –x 中的 rkB/s,wkB/s 数据相同。与手册中的解释不太一样。

2.2 Linux 相关设置

2.2.1 Loadrunner 监控 linux 的配置

Loadrunner 利用 linux 系统中的 xinetd 服务来取得 linux 系统的相关统计信息。因此，要想使用 loadrunner 来监控 linux，需要进行相关配置，详细做法请参考 loadrunner 手册。

2.2.2 其他服务与性能监控

除了系统命令的方式。其他工具也可以利用 linux 的相关服务来取得 linux 系统的参数。如 QALoad（美国 compuware 公司）就是利用了 snmp 协议来获取 linux 相关的系统性能信息。如果有需要，请参考相关工具文档即可。

3 Apache 性能与度量

3.1 性能参数

- Total Accesses 总的访问量
- Total KBs 总的吞吐量
- CPU Load CPU 负载百分数
- Uptime 启动以来的时间
- Requests Per Second 每秒请求数
- Bytes Per Second 每秒字节数
- Bytes Per Req 每个请求的 byte 数
- Busy Workers 处于 busy 状态的线程数
- Idle Workers 处于 idle 状态的线程数

通过查看 apache 的各个性能参数的指标值，我们可以获得 apache 负载情况。

3.2 相关设置

- **To Enable the Server Status**, follow the steps given below:

  1. In Apache's httpd.conf file, locate "Location /server-status" tag. If you are not able to locate the server-status tag, do the following

  2. Remove the comment in the Location/Server-status tag, to Enable SetHandler server-status

  3.Change the attribute "deny from all" to "Allow from all"

  4. Remove the comment in "LoadModule status_module modules/mod_status.so".

  5. Save the conf file and restart the Apache Server
- **To enable the Extended-status**, follow the steps given below:

1. Locate "ExtendedStatus" Attribute in httpd.conf file.
2. Remove the comment to enable the status.
3. Save the conf file and restart the Apache Server

- Example

  http://site:port/server-status

  信息输出：

  Current Time: Wednesday, 04-Mar-2009 20:47:30 CST

  Restart Time: Monday, 02-Mar-2009 08:30:46 CST

  Parent Server Generation: 0

  Server uptime: 2 days 12 hours 16 minutes 43 seconds

  Total accesses: 373158 - Total Traffic: 31.1 GB

  CPU Usage: u21.81 s10.79 cu0 cs0 - .015% CPU load

  1.72 requests/sec - 150.2 kB/second - 87.4 kB/request

  1 requests currently being processed, 127 idle workers

3.3 Apache 与 tomcat connector(mod_jk)

Apache 与 tomcat 的连接配置会影响到 apache 和 tomcat 的连接性能。
Mod_jk 使用的默认配置文件为：worker.properties。关于 mod jk 的工作原理和相关性能详情请参考：http://tomcat.apache.org/connectors-doc/reference/workers.html

4 Tomcat 性能与度量

4.1 Tomcat 性能参数

4.1.1 Response Time Details

- Average Response Time

Tomcat 处理请求的平均响应时间。

- Requests per Second

  Tomcat server 每秒收到的请求数。

- Average Bytes per Second

  每秒平均发送的字节数。

4.1.2 Memory Usage

- Total Memory

  Tomcat 使用的最大内存。

- Used Memory

Tomcat 当前使用的内存。

- Free Memory
  Tomcat 剩余的可用内存。

4.1.3 Thread Details

- Busy threads
  处于忙状态的 tomcat 线程数，也就是当前正在使用中的线程数。
- Current threads
  已经创建的 tomcat 线程数，也就是可用的线程数。

4.1.4 Response Summary

每种响应类型的数量。

4.1.5 Application Summary

通过对应用程序概要的查看，可以查看相关 jsp 页面的解析时间等性能参数。

4.1.6 Session Details

Tomcat session 的情况，包括 tomcat 的 session 数等。

Parameters	Description
Requests per Second	Specifies the number of requests received by the server in one second.
Average Bytes per Second	Refers to the average bytes per second.

4.2 Java 虚拟机性能

4.2.1 虚拟内存参数

Classes loaded	Number of classes loaded

Classes Unloaded	Number of classes unloaded
GC time	Time taken to perform garbage collection
Compile time	Time spent in just-in-time (JIT) compilation
Max file descriptor	Maximum permissible open file descriptor. Available only for UNIX.
Open file descriptor	Current count of open file descriptors. Available only for UNIX.

4.2.2 动态内存状态

Eden Space (Heap Memory)	Pool from which memory is initially allocated for most objects
Survivor Space (Heap Memory)	Pool containing objects that have survived GC of eden space.
Tenured Generation (Heap Memory)	Pool containing objects that have existed for some time in the survivor space.
Permanent Generation (Non-Heap)	Holds all the reflective data of the virtual machine itself, such as class and method objects. With JVMs that use class data sharing, this generation is divided into read-only and read-write areas.
Code Cache (Non-Heap)	Memory used for compilation and storage of native code.

4.2.3 线程状态

Live Threads	Number of live threads currently running
Daemon Threads	Number of daemon threads currently running
Runnable Threads	A thread executing in the Java virtual machine is in this state

Blocked Threads	A thread that is blocked waiting for a monitor lock is in this state
Waiting Threads	A thread that is waiting indefinitely for another thread to perform a particular action is in this state.
Timed waiting Threads	A thread that is waiting for another thread to perform an action for up to a specified waiting time is in this state

4.3 Tomcat 相关设置

4.3.1 Tomcat 连接数

修改 tomcat\conf\server.xml 文件中的如下部分

<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"port="8080" minProcessors="5" maxProcessors="75" enableLookups="true" redirectPort="8443" acceptCount="100" debug="0" connectionTimeout="20000" useURIValidationHack="false" disableUploadTimeout="true" />

其中 minProcessors 为最小连接数；maxProcessors 为最大连接数；acceptCount 为请求队列的最大长度；connectionTimeout 为网络连接超时时间毫秒数。

Connector 的配置属性说明：

Attribute	Description
acceptCount	The maximum queue length for incoming connection requests when all possible request processing threads are in use. Any requests received when the queue is full will be refused. The default value is 10.
allowChunking	If set to true, chunked output is allowed when processing HTTP/1.1 requests. This is set to true by default.
address	For servers with more than one IP address, this attribute specifies which address will be used for listening on the specified port. By default, this port will be used on all IP addresses associated with the server.
bufferSize	The size (in bytes) of the buffer to be provided for input streams created by this connector. By default, buffers of 2048 bytes will be provided.
connectionTimeout	The number of milliseconds this **Connector** will wait, after accepting

	a connection, for the request URI line to be presented. The default value is 60000 (i.e. 60 seconds).
debug	The debugging detail level of log messages generated by this component, with higher numbers creating more detailed output. If not specified, this attribute is set to zero (0).
maxProcessors	The maximum number of request processing threads to be created by this **Connector**, which therefore determines the maximum number of simultaneous requests that can be handled. If not specified, this attribute is set to 20.
minProcessors	The number of request processing threads that will be created when this **Connector** is first started. This attribute should be set to a value smaller than that set for maxProcessors. The default value is 5.
port	The TCP port number on which this **Connector** will create a server socket and await incoming connections. Your operating system will allow only one server application to listen to a particular port number on a particular IP address.
proxyName	If this **Connector** is being used in a proxy configuration, configure this attribute to specify the server name to be returned for calls to request.getServerName(). See Proxy Support for more information.
proxyPort	If this **Connector** is being used in a proxy configuration, configure this attribute to specify the server port to be returned for calls to request.getServerPort(). See Proxy Support for more information.
tcpNoDelay	If set to true, the TCP_NO_DELAY option will be set on the server socket, which improves performance under most circumstances. This is set to true by default.

4.3.2 虚拟机内存

修改 tomcat\bin\catalina.bat 文件，在 JAVA_OPTS 变量使用前加入

set JAVA_OPTS=-Xms128m -Xmx256m

其中 Xms 为最小内存，Xmx 为最大内存。

设定的最大内存可用如下命令测试：java -Xmx1048m -version

可以使用如下程序代码实现对内存的监控：

```
<%
Runtime lRuntime = Runtime.getRuntime();
out.println("Free Memory: "+lRuntime.freeMemory()+"<br>");
```

```
out.println("Max    Memory: "+lRuntime.maxMemory()+"<br>");
out.println("Total Memory: "+lRuntime.totalMemory()+"<br>");
%>
```

4.3.3 Tomcat 管理用户

修改 tomcat\conf\tomcat-users.xml。

```
<tomcat-users>
<role rolename="tomcat"/>
<role rolename="manager"/>
<role rolename="admin"/>
<user username="tomcat" password="tomcat" roles="tomcat,manager"/>
</tomcat-users>
```

4.3.4 JMX Remote 设置

Add the following parameters to your tomcat startup script:

```
set CATALINA_OPTS="-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.port=%my.jmx.port% \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.authenticate=false"
```

4.4 Tomcat 监控方法

我们可以使用 tomcat 自带的 manager 来对 tomcat 的状态进行监控，获得相关的性能信息。详情请参考：
http://tomcat.apache.org/tomcat-6.0-doc/manager-howto.html

5 MySQL 性能与度量

有许多商业或开源的工具对 mysql 的性能进行监控。我们也可以通过执行 show status 查询来获得 mysql 的性能快照。

5.1 Mysql 性能参数

5.1.1 Connection Time

Parameter	Description
Connection Time	Specifies the time taken to connect to the database
Request Rate	Number of request received in one second.

5.1.2 Requests Statistics

Parameter	Description
Request Rate	Number of request received in one second.
Bytes Received Rate	Number of bytes received in one second.
Bytes Sent Rate	Number of bytes sent in one second.

5.1.3 Connection Statistics

Parameter	Description
Open Connections	The number of connections opened at present in the MySql Server.
Aborted Connections	Number of tries to connect to the MySQL server that failed.
Aborted Clients	Number of clients aborted by MySQL server.

5.1.4 Thread Details

Parameter	Description
Threads Used	Number of threads processing the request.
Threads in Cache	Number of threads currently placed in the thread cache.

Thread Cache Size	Specifies the cache size in the MySQL server.

5.1.5 Database Details

Parameter	Description
Database Name	Name of the database instance.
Database Size	Size of the various databases in the MySQL server.

5.1.6 Table Lock Statistics

Parameter	Description
Immediate Locks	Number of times a table lock for the table is acquired immediately.
Locks Wait	Number of times a table lock could not be acquired after waiting.

5.1.7 Key Efficiency

Parameter	Description
Key Hitrate	Percentage of key read requests that resulted in actual key reads from the key buffer.
Key Buffer Used	Amount of allocated key buffer in use.
Key Buffer Size	Size of the buffer used for index blocks. Also known as the key cache.

5.1.8 Query Statistics

Parameter	Description
Queries Inserted/Min	No. of Insert Queries executed per minute
Queries Deleted/Min	No. of Delete Queries executed per minute
Queries Updated/Min	No. of Update Queries executed per minute

Queries Selected/Min	No. of Select Queries executed per minute

5.1.9 Query Cache Hitrate

Note: This performance data is not available for MySQL versions 3.23.x

Parameter	Description
Query Cache Hitrate	Ratio of queries that were cached and queries that were not cached.
Query Cache Size	Amount of memory allocated for caching query results.
Query Cache Limit	Maximum amount of memory for storing cache results.

5.1.10 Replication Details

Parameter	Description
Replication Status	The status of Slave process in MySQL Server
Slave IO Running	Status of the Slave IO Process in MySQL Server. Possible values are Yes/No
Slave SQL Running	Status of the Slate SQL Process in MySQL Server.Possible values are Yes/N.
Time Behind Master	This indicates of how “late” the slave is behind the Master

5.2 性能相关话题

5.2.1 Slow log

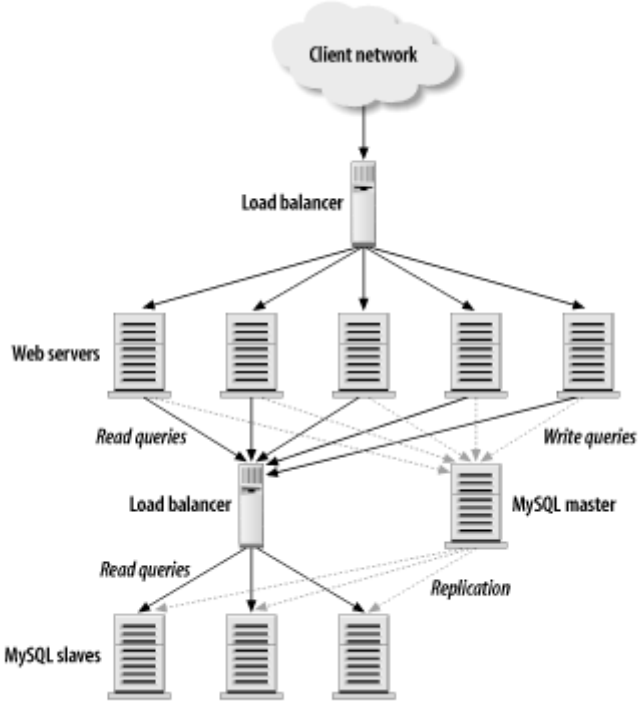
Slow log 是找出慢查询最常用也最有效的方式。通过在 mysql 的配置文件中配置 slow log 参数，可以将执行时间超长的查询鉴别出来。进一步调整和优化。

5.2.2 连接池

- C3P0
  C3P0 是在我们的 crm 系统中广泛使用的连接池模块。它是一个开源的应用模块，与性能相关的配置主要为连接池参数的配置。例如：
  C3P0.initialPoolSize=5
  C3P0.minPoolSize=5
  C3P0.maxPoolSize=25
  dbpool.numHelperThreads=5
  更多信息请参考 C3P0 开发者网站。
- Proxool
  请参考开发者网站。

5.2.3 负载均衡与读写分离

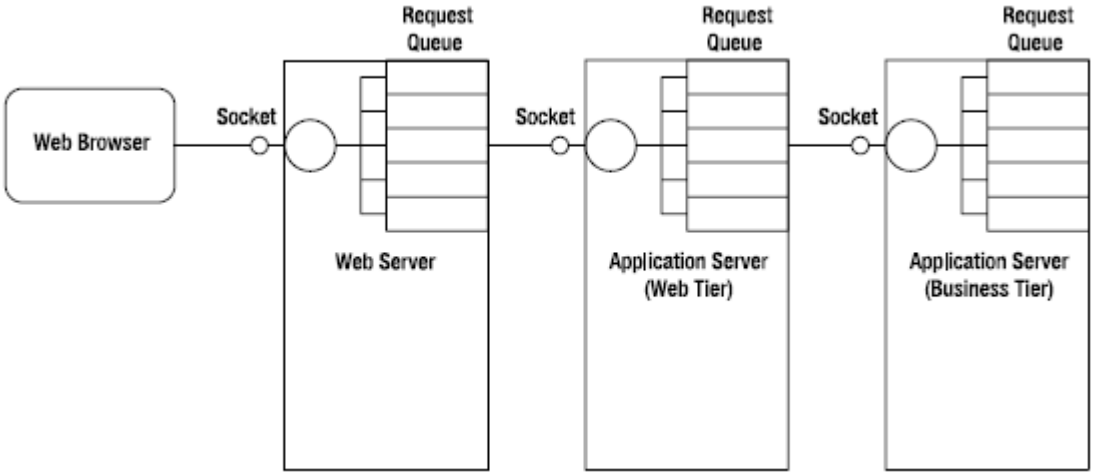
线上提供的很多服务都是进行了负载均衡和读写分离的。一般而言，写操作都是在主库进行的，读操作会采用一定的算法分离到从库上去，来减轻主库的负载，提高系统的总体性能。在进行性能测试时需要充分考虑到这一点。



以读为主的网站负载均衡示意图

6 J2EE 架构与性能管理

J2EE 的架构、性能管理与优化是一个专门的话题，此处仅作简单介绍，供了解之用。



Common path an application request follow through a java EE stack

6.1 Application 性能

衡量应用程序性能的指标有很多，以下的一些指标一定程度上可以用来作为对应用程序进行衡量的一个参考。

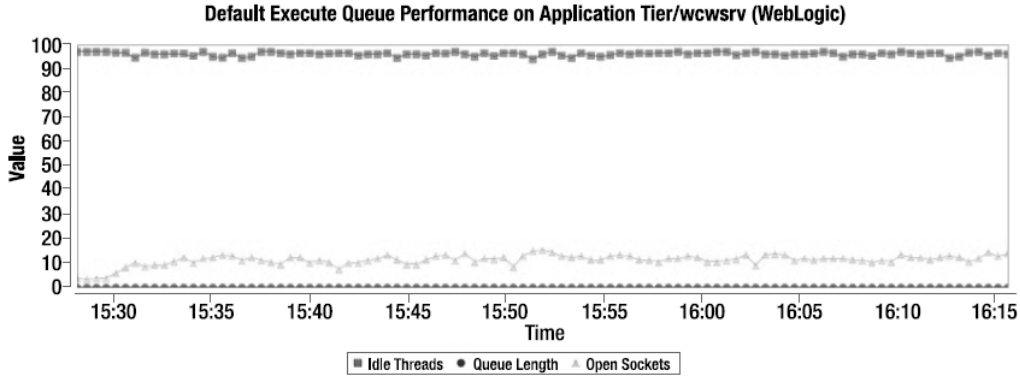
- Average response time
  The average response time for the request during the aggregated sample
- Maximum response time
  The maximum response time for the request during the aggregated sample
- Call count
  The number of times the request was executed during the aggregated sample
- Total response time
  The total time that this request spent executing during the sample (average response time multiplied by the call count)
- Exceptional exits
  For each request during the aggregated sample, how many times it ended in an exception
- Percent incomplete: For each request during the aggregated sample, how many times it failed to complete within the configured time-out value

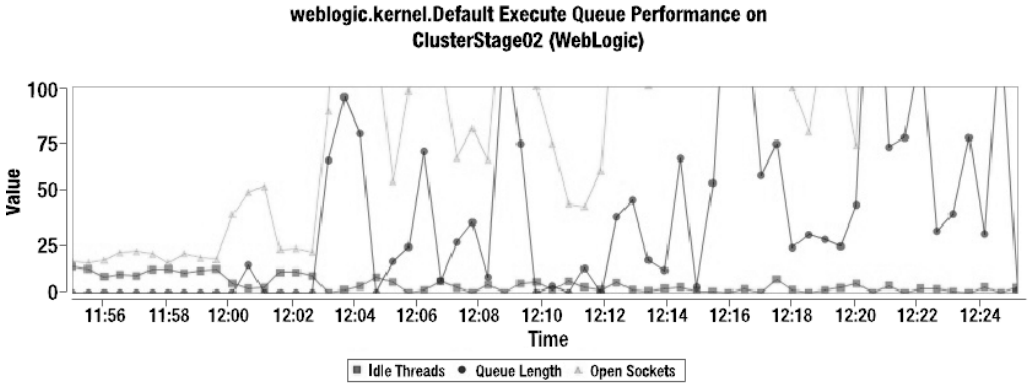
6.2 Application Server 性能参数

- Thread pools
  当应用服务器接收到请求时，会将它放到一个执行队列中，每个执行队列都维护了一个线程池。当有执行线程可用时，执行队列将请求取出，该执行线程处理请求。当没有空闲线程可用时，请求将处于等待状态。线程池的性能分析包含以下特征：

- ✧ Thread pool utilization
- ✧ Queue depth
- ✧ Request throughput

下面例子给出了一个状态良好的 thread pool 和一个效率较差的线程池，可作为参考：





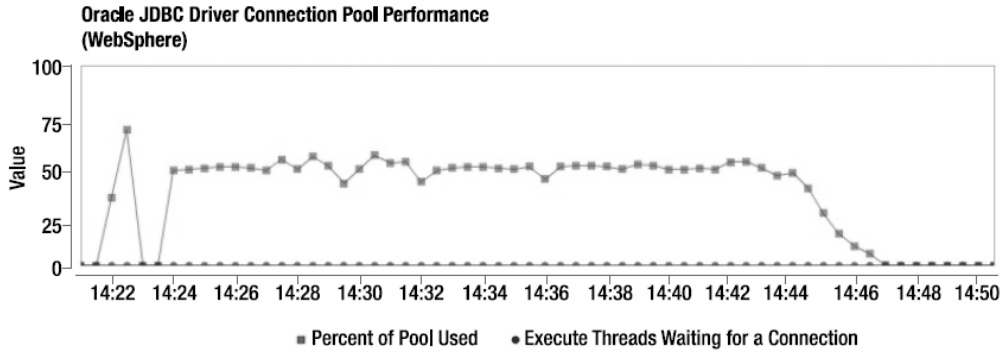
- Connection pools

在使用数据库和其他外部资源时，应用服务器要用到连接池。如 JDBC 的连接池。使用连接池技术大大提高了连接的效率。有两个非常重要的指标可以用来衡量连接池的性能情况：

- ✧ Connection pool utilization
- ✧ Execute threads waiting for a connection

连接池的利用率理想情况下应该在 50-70%之间，并且没有等待线程。如果低于 50%，说明应该减小池的大小，以释放不必要的占用的资源。如果高于 70%应该适当增加池的大小，因为任何用户行为模式的改变都可能对应用程序的性能造成较大的影响。

下面给出了一个健康的连接池的例子：



- Caches

- ✧ Hit count ： the number of requests satisfied by the cache
- ✧ Miss count ： the number of requests not satisfied by the cache

命中率(hit ratio)低于 70%时，说明它的效率较低。大于 90%时，说明它的效率基本可以接受。

- Component pools
  略。
- Message servers
  略。
- Transactions
  略。

6.3 Java 虚拟机性能

- Heap performance
  - ✧ Heap utilization
  - ✧ Heap growth pattern
  - ✧ Garbage collection behavior

- Process memory utilization
  观察进程内存利用情况时，下面两项指标非常重要：
  - ✧ Operating System memory constraints
  - ✧ Permanent memory anomalies

7 参考资料

以下列出了一些参考资料，供参考。

- IBM developerworks
- Linux performance tuning
- Tomcat documents
- Advent Net
- High performance mysql 2nd Edition(O'Reilly)
- www.apache.org