SYBASE®
An SAP® Company

性能和调优系列：使用 sp_sysmon 监控 Adaptive Server

Adaptive Server® Enterprise

15.7

文档 ID：DC01076-01-1570-01

最后修订日期：2011 年 9 月

Sybase, Inc., One Sybase Drive, Dublin, CA 94568。

目录

第 1 章

sp_sysmon 简介

本章讨论如何使用和激活 sp_sysmon。

使用 sp_sysmon

sp_sysmon 提供您系统上的活动的详细报告，并向您提供一些方法来指定要接收的信息的类型，收集该报告的数据的时间间隔，以及用于确定如何生成报告的其它选项。

sp_sysmon 报告包含大量独立的部分。您可以运行 sp_sysmon 来生成完整报告或仅生成某个独立的部分。您还可以指定报告应运行的时间间隔，或在所需时间段开始和结束时亲自执行存储过程。

sp_sysmon 仅报告采样周期内的监控数据。请确保根据有代表性的数据做出调优决策。例如，为了对螺旋锁进行调优，应根据峰值利用率报告中的数据做出决策。但是，应根据大量代表典型和峰值负载的样本来做出减少引擎数的决策。

报告的数据是从 Adaptive Server® 维护的一组监控计数器中收集的。其它应用程序也使用这些计数器。缺省情况下，在激活 sp_sysmon 报告时，它会清除这些计数器。清除计数器可能会干扰使用这些计数器的其它应用程序，从而导致它们报告的数据无效。

警告！ 若要控制 sp_sysmon 是否清除计数器，请指定 `noclear` 选项。指定 `noclear` 选项时， sp_sysmon 不清除计数器，从而允许 sp_sysmon 和其它 sp_sysmon 会话同时运行。缺省情况下，使用采样间隔运行 sp_sysmon 时，会启用 noclear，而在使用 begin_ *和* end_sample *参数运行* sp_sysmon 时，则会禁用它。有关详细信息，请参见第 6 页的 “使用 noclear 选项”。

在单 CPU 服务器上运行时，sp_sysmon 会占用约 5 - 7% 的开销，而在多处理器服务器上运行时，所占的比例更大 （具体百分比可能因您的网站而异）。随着 CPU 数的增加，开销量也会增加。 sp_sysmon `noclear` 使用相同的内部计数器。在不使用该选项运行时， sp_sysmon 会将这些计数器重置为 0。

sp_sysmon 和监控表之间存在共享的计数器。如果在 sp_sysmon 已经在运行时再次执行该命令，则会清除所有计数器，因此第一个 sp_sysmon 会话生成的报告不准确。

有关通过其它应用程序运行 sp_sysmon 的信息，请参见第 6 页的 “使用 noclear 选项”。

---

sp_sysmon 的性能调优提示基于您对 sp_sysmon 提供的采样间隔。在生产系统中纳入这些建议之前，请根据您的系统要求详细查看所有建议。Sybase® 强烈建议您使用数据设置一个测试区域，并在实施任何建议前测试所有更改。

由于 sp_sysmon 提供了系统的快照视图，因此您可能需要在负载更改时重新考虑这些建议。

注释 您不能通过缺省大小的 tempdb 在 Adaptive Server 上运行 sp_sysmon。将 tempdb 的大小增加至少 2MB，以便 Adaptive Server 不会用完临时数据库上的日志空间。

何时运行 sp_sysmon

您可以在调优 Adaptive Server 配置参数前后运行 sp_sysmon 以收集数据进行比较。该数据为性能调优奠定基础，并且您可使用它观察配置改变结果。

当系统出现您要调查的行为时，请使用 sp_sysmon。例如，若要了解系统在典型负载条件下的行为，可在正常情况下的典型负载时运行 sp_sysmon。例如，考虑从晚上 7:00 开始（批处理作业开始之前，当天的多数 OLTP 用户已离开站点之后）运行 sp_sysmon 10 分钟是否有意义。然而，应在正常 OLTP 装载和批处理作业期间运行 sp_sysmon。

在许多测试中，最好先启动应用程序，然后在高速缓存可能达到稳定状态时启动 sp_sysmon。如果要尝试测试容量，请确保服务器的给定工作量能够使其在测试期间处于繁忙状态。

如果服务器在采样间隔的部分时间内处于空闲状态，则许多统计信息（尤其是那些按秒测量的数据），其值将显得非常低。

一般说来，在以下情况下使用 sp_sysmon 时会产生有价值的信息：

- 高速缓存或内存池的配置改变前后
- 任何可能影响性能的 sp_configure 更改（例如，内存大小、高速缓存或与磁盘 I/O 相关的选项的更改）前后
- 在应用程序组合中添加新查询前后
- Adaptive Server 引擎数增减前后
- 添加新磁盘设备以及指派对象时
- 在高峰期间查找争用或瓶颈问题
- 在负载测试期间，估计 Adaptive Server 的最大预期应用程序负载配置
- 性能似乎降低或运行不正常时

您可能还会发现 sp_sysmon 在查询或应用程序开发期间很有用。例如，当您执行以下操作时：

- 使用索引和更新，以查看报告为 deferred_varcol 的某些更新是否会导致直接更新与延迟更新
- 检查特定查询或混合查询的高速缓存行为
- 为并行索引创建调优参数和高速缓存配置

激活 sp_sysmon

sp_sysmon 要求您将 enable monitoring 设置为 true （启用），并且用户具有 mon_role：

```
sp_configure 'enable monitoring', 1
grant role mon_role to sa
```

使用 sp_sysmon 时采用以下项：

- 固定时间间隔，以提供指定分钟数内的样本
- begin_sample 和 end_sample 参数，以启动和停止采样

定制输出以提供所需信息。您可以：

- 输出整个报告。
- 只输出报告的一部分，如 “高速缓存管理”或 “锁管理”。
- 包括指定应用程序 （如 isql、 bcp 或任何指定应用程序）和指定应用程序和用户名的组合的应用程序级详细报告。缺省情况下会省略此部分。

固定时间间隔

若要激活 sp_sysmon，请使用 isql 执行以下命令：

```
sp_sysmon interval [, section  [, applmon]]
```

interval 必须采用 “hh:mm:ss”的形式。例如，若要运行 sp_sysmon 10 分钟，请使用：

```
sp_sysmon "00:10:00"
```

有关 *applmon* 参数的信息，请参见第 6 页的 “指定应用程序详细信息参数”。

使用 *begin_sample* 和 *end_sample*

使用 begin_sample 和 end_sample 参数可在任何时间点激活 sp_sysmon 以启动采样、发出查询以及终止采样并输出结果。例如：

```
sp_sysmon begin_sample
execute proc1
execute proc2
select sum(total_sales) from titles
```

```
sp_sysmon end_sample
```

注释　在有很多 CPU 和大量活动的系统上，如果采样周期过长，计数器可能会溢出。

如果在 sp_sysmon 输出中看到负数，请缩短采样时间。

指定要输出的报告部分

若要输出报告的某一部分，请对 *section* 参数使用表 1-1 中列出的值之一。

表 1-1：sp_sysmon 报告的各个部分

报告部分	参数
应用程序管理	appmgmt
高速缓存向导	cache wizard
数据高速缓存管理	dcache
磁盘 I/O 管理	diskio
ESP 管理	esp
管家任务活动	housekeeper
索引管理	indexmgmt
内核利用率	kernel
锁管理	locks
内存管理	memory
元数据高速缓存管理	mdcache*
对执行 SQL 的监控访问	monaccess
网络 I/O 管理	netio
并行查询管理	parallel
过程高速缓存管理	pcache
恢复管理	recovery
RepAgent	repagent
任务管理	taskmgmt
事务管理	xactmgmt
事务配置文件	xactsum
工作进程管理	wpm

您还可以使用 sp_monitorconfig 获取 sp_sysmon mdcache 报告中提供的大多数信息。请参见 《参考手册：过程》。

指定应用程序详细信息参数

如果为 *sp_sysmon* 指定 applmon 参数，则报告将包含按应用程序或按应用程序和登录名排列的详细信息。仅当输出完整报告或为 section 参数指定 appmgmt 时，此参数才有效。如果指定应用程序详细信息参数并请求报告的任何其它部分，则会忽略应用程序详细信息参数。

第三个参数必须是以下值之一：

参数	报告的信息
appl_only	按应用程序名称排列的 CPU、 I/O、优先级改变和违反资源限制的信息。
appl_and_login	按应用程序名称和登录名排列的 CPU、 I/O、优先级更改和违反资源限制的信息。可用于所有部分。
no_appl	跳过报告的应用程序和登录部分。这是缺省值。

下面的示例将运行 sp_sysmon 5 分钟，并输出 “应用程序管理”部分，包括应用程序和登录详细报告：

```
sp_sysmon "00:05:00", appmgmt, appl_and_login
```

有关输出样本，请参见第 53 页的 “Per Application 或 Per Application And Login”。

使用 *noclear* 选项

缺省情况下， sp_sysmon 不清除用作报告源数据的监控计数器。如果清除计数器时有其它应用程序或 sp_sysmon 报告的实例正在运行，则清除计数器可能会导致它们报告的数据无效。

将 noclear 可选参数与 sp_sysmon 结合使用，以便不清除监控计数器。例如：

```
sp_sysmon “00:15:00”, noclear
```

此示例在 15 分钟的采样间隔内运行 sp_sysmon 报告，而不清除监控计数器。您可以将 `noclear` 值用于以下任意参数，这样您可以与其它参数值一起指定 noclear：

```
@section
@applmon
@filter
@dumpcounters
```

例如，若要在使用 noclear 选项时创建内核资源使用情况的报告，请发出：

```
sp_sysmon “00:15:00”, kernel, noclear
```

带 noclear 选项运行 sp_sysmon 时，您可能会在采样周期开始时检测到 I/O 活动量稍有增加。该报告会创建一个用于存储初始计数器值的临时表。除了多数处于抑制状态的系统外，此活动对 sp_sysmon 报告的数据的影响应该可以忽略不计。

您还可以使用 noclear 选项运行 sp_sysmon 报告的多个并发会话。考虑其它正在运行的系统监控应用程序是否使用监控计数器。如果它们使用监控计数器，请使用 noclear 选项。

从与您在其上执行 'begin_sample' 的不同会话执行 sp_sysmon 'end_sample' 时， sp_sysmon 不会减少监控计数器，并且 Adaptive Server 向客户端返回错误消息号 19374。这意味着，如果某个应用程序启用了监控计数器，但在注销前没有禁用它们，则使用计数继续反映与该应用程序注销前相同的用户数。

将输出重定向到文件

完整的 sp_sysmon 报告包含数百行输出内容。使用 isql 输入和输出重定向标志可将输出保存到文件。例如，如果创建名为 *sysmon_in* 的文件，其中包含的以下命令可运行 sp_sysmon 10 秒钟：

```
sp_sysmon '00:00:10'
go
```

使用 isql -o 参数将输出通过管道传递到名为 *sysmon_out* 的文件：

```
$SYBASE/$SYBASE_OCS/bin/isql -Usa -P -Sbig_server -isysmon_in -osysmon_out
```

有关 *isql* 的详细信息，请参见 《实用程序指南》。

第 2 章

使用 *sp_sysmon* 监控性能

本章介绍 sp_sysmon 的输出，其中包括为解释 sp_sysmon 的输出和推导可能的结论提供的一些建议。

当您了解 Adaptive Server 环境及其特定的应用时，sp_sysmon 的输出最有价值。

如何使用这些报告

sp_sysmon 可提供有关调优前后 Adaptive Server 系统行为的信息。应研究整个报告以了解所做更改将产生的全部影响。有时，解决一个性能瓶颈可能会显露出另一个性能瓶颈。类似地，调优工作可能会提高某一方面的性能，而实际上却导致另一个方面的性能降低。

除了指出需要进行调优的方面外， sp_sysmon 的输出还有助于确定在什么情况下进一步调优将不能获得更好的性能。懂得何时停止调优Adaptive Server，或懂得问题出现在其它什么地方与懂得调优什么内容是同样重要的。

sp_sysmon 的单次运行只呈现特定时间间隔内的资源利用率。确保您使用的时间间隔清楚地表示您要对其调优 Adaptive Server 的负载和情形。

其它信息有助于解释 sp_sysmon 的输出：

- sp_configure 或配置文件中所使用的配置参数的有关信息
- sp_cacheconfig 和 sp_helpcache 的高速缓存配置和高速缓存绑定的有关信息
- 磁盘设备、段及存储在其上的对象的有关信息

标头信息

sp_sysmon 的标头包括当前 sp_sysmon 运行的常规信息，其中包括 Adaptive Server 版本、运行 sp_sysmon 的日期、开始和结束时间、采样的持续时间以及您是否指定了服务器。标头指示运行 sp_sysmon 时所采用的模式（是否包含了 noclear 选项）以及上次清除计数器的时间 （有关计数器的信息，请参见第 1 页的 “使用 sp_sysmon”）。

标头还指示您是否需要设置任何配置参数以收集监控表信息，以及您是否需要向系统管理员授予 mon_role 角色。

读取输出

sp_sysmon 以格式一致的表形式显示性能统计信息。例如，在 SMP 环境中运行七个 Adaptive Server 引擎，输出结果类似于：

```
Engine Utilization (Tick %)        CPU Busy   I/O Busy         Idle
------------------------          --------   --------     --------
Engine 0                            68.7 %      2.5 %       28.8 %
Engine 1                            61.9 %      3.3 %       34.8 %
Engine 2                            67.0 %      2.4 %       30.6 %
Engine 3                            69.0 %      3.8 %       27.2 %
Engine 4                            60.2 %      2.7 %       37.2 %
Engine 5                            55.7 %      3.2 %       41.1 %
Engine 6                            53.8 %      3.2 %       43.0 %
------------------------          --------   --------     --------
Summary          Total             436.3 %     21.1 %      242.6 %
Average                             62.3 %      3.0 %       34.7 %
```

行

多数行表示特定类型的活动或事件，例如，获取锁或执行存储过程。当数据与 CPU 相关时，行可显示 SMP 环境中每个 Adaptive Server 引擎的性能信息。通常，有多组相关行时，最后一行为总计和平均值的摘要。

sp_sysmon 报告缩进一些行，以显示一个类别是另一类别的子类别。在下面的示例中，“Found in Wash”是“Cache Hits”的子类别，而“Cache Hits”又是“Cache Searches”的子类别：

```
Cache Searches                   per sec       per xact       count  % of total
                               ----------     ---------    -------   --------
   Cache Hits                   32731.6       153429.6     9819492     100.0 %
     Found in Wash                288.4         1351.7       86508       0.9 %
Cache Misses                       10.9           50.9        3257       0.0 %
---------------------        ------------     ---------   ---------
Total Cache Searches             32742.5       153480.5     9822749
```

“count”值为 0 时，将有许多行不会输出。

输出的列

除非本章中另外说明，否则本章的示例中的列表示以下性能统计信息：

- “per sec”— 采样间隔期间每秒的平均值。
- “per xact”— 采样间隔期间每个提交事务的平均值。
- “count”— 采样间隔期间的总数。
- “% of total”— 因环境而异，以每次出现时所做的说明为准。

解释数据

调优 Adaptive Server 时，衡量调优成功的基本标准是吞吐量的增加和应用程序响应时间的缩短。遗憾的是，调优 Adaptive Server 并不能简化为仅输出这两个值。

多数情况下，您的调优工作必须采用迭代方法，包括对 Adaptive Server 活动的全面观察、对查询和应用程序的仔细调优和分析，以及以逐个对象为基础监控锁定和访问。

每秒和每个事务数据

衡量每秒和每个事务数据对所测量的环境和类别的重要性。在基准检查或已定义好了工作负荷的测量环境中，每个事务数据通常更有意义。

您很可能会发现当比较测试数据时，每个事务数据比单独使用每秒数据更有意义，因为在基准测试环境中，通常已定义好了事务数，使比较更加简单直接。每个事务数据对于确定百分比结果的有效性也很有用。

当分析的负载涉及大量未在事务内执行的查询 （例如 select 语句）时，您可能会发现查看每秒平均值更为有用。每秒分析允许您对影响这些查询的调优进行比较。

总计的百分比和计数数据

“% of total”数据的含义会变化，具体取决于事件的环境和类别的总数。解释百分比时，请记住，它们通常有助于了解总趋势，但孤立地看待它们却容易受误导。例如，200 个事件的 50% 比 2 个事件的 50% 要有意义得多。

“count”数据是在采样间隔期间发生的事件的总数。您可以使用计数数据来确定百分比结果的有效性。

每个引擎数据

多数情况下，类别的每个引擎数据会显示出所有引擎上的活动非常均衡。有两种例外情况：

- 如果您的进程比 CPU 少，有些引擎会显示没有活动。
- 如果多数进程执行相当单调重复的活动 （例如，简单插入和简短选择），而某一个进程执行一些 I/O 密集型操作 （例如大批量复制），您将看到网络和磁盘 I/O 不均衡。

总计和摘要数据

摘要行通过报告总数和平均数来提供 Adaptive Server 引擎活动的概况。

解释平均值时要小心，因为当数据倾斜时会造成对正确结果的错误印象。例如，如果一个 Adaptive Server 引擎 98% 的时间在工作，而另一引擎 2% 的时间在工作，可能会误认为平均值为 49%。

高速缓存向导

“Cache Wizard”部分可以帮助您监控和配置高速缓存以获取最佳性能。

sp_sysmon cache wizard 要求您启用 object lockwait timing 配置参数。

使用 “高速缓存向导”，可以标识：

- 热对象 （经常访问的对象）。输出按命名高速缓存或缺省数据高速缓存中的逻辑读取数排列。
- 高速缓存上的螺旋锁争用。
- 高速缓存和缓冲池的使用。
- 高速缓存、缓冲池和对象级别的命中百分比。
- 大 I/O 的效力。
- 异步预取 (APF) 的效力。
- 各个对象的高速缓存使用情况。

仅当在 sp_sysmon 语法中指定了 cache wizard 时，sp_sysmon 输出中才会显示 “Cache Wizard”部分。您可以随 Cache Wizard 一起附带两个参数：topN 和 filter。当您在语法中包含 cache wizard 时，sp_sysmon 不包括其它部分 （“Kernel Utilization”、“Worker Process Management”等）。

sp_syntax 在其输出结果的末尾包含建议部分。

高速缓存向导的语法

若要获取高速缓存向导的输出，请使用：

```
sp_sysmon begin_sample
sp_sysmon { end_sample | interval }
[, 'cache wizard'  [, top_N [, filter] ] ]
```

参数

- top_N — varchar 数据类型，它根据指定间隔内的逻辑读取次数 （显示在 LR/sec 列中）的排列条件限制在 Object 部分报告的对象列表。

  根据指定的值是正整数还是负整数，排列顺序为升序或降序。可通过指定值 0 （零），获取在间隔结束时占用高速缓存的对象的整个列表。缺省值为 10。

- filter — varchar 数据类型，用来为报告中包括的高速缓存指定模式。

  例如，如果将 filter 指定为 default data cache，报告将只包含有关缺省数据高速缓存的信息。如果将 filter 值指定为 emp%，输出将包括有关其名称与此模式匹配的所有高速缓存的信息。

  若要显示所有高速缓存的输出，请将 filter 留空。首先显示 default data cache，然后按字母顺序显示其它高速缓存。

有关详细信息，请参见第 15 页的 “样本输出”。

示例

本例在 5 分钟间隔内针对缺省数据高速缓存运行高速缓存向导，以显示高速缓存中的前 15 个对象：

```
sp_sysmon '00:05:00','cache wizard','15','default data
cache'
```

准备运行高速缓存向导

sp_sysmon 从监控表和监控计数器检索高速缓存向导的信息。

有关详细信息，请参见 Performance and Tuning Series: Monitoring Tables （《性能和调优系列：监控表》）。

高速缓存向导报告要求启用以下配置参数。如果未启用这些配置参数，sp_sysmon 会自动在间隔开始时启用它们并在间隔结束时禁用它们 （所有这些选项都是动态的）：

- enable monitoring — 设置为 1
- per object statistics active — 设置为 1
- object lockwait timing — 设置为 1

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

样本输出

“Cache Wizard”部分的输出包含各个高速缓存的三个主要部分。之后是建议部分，报告最后是图例部分：

- 高速缓存部分 — 提供特定高速缓存的摘要统计信息：

```
default data cache
------------------
Run Size          :   100.00Mb  Usage%               :  2.86
LR/sec            :    41.10    PR/sec:             : 22.57  Hit%:45.9
Cache Partitions  :     4       Spinlock Contention% :  0.00
```

Usage% — 每次将一组页放入高速缓存中时，Adaptive Server 都会进行跟踪以查看是否引用 （使用）了该页。从高速缓存中删除该页后，此计数会降低。 Usage% 是高速缓存大小的百分比形式表示的高速缓存的当前使用情况。

LR/sec — （每秒的逻辑读取）逻辑读取是从高速缓存 （命中）或磁盘 （未命中）进行的读取。 LR/sec 是间隔期间高速缓存中的逻辑读取数除以采样间隔的时间长度。

PR/sec — （每秒的物理读取）物理读取是从磁盘进行的读取 （未命中）。 PR/sec 是间隔期间高速缓存中的物理读取数除以采样间隔。

Hit% — 命中次数与高速缓存读取总数的比率，如 (LR/sec - PR/sec) 与 LR/sec 的比率。

- 缓冲池部分 — 将高速缓存部分统计信息划分到高速缓存的各个缓冲池中。

```
Buffer Pool Information
-------------------------------------------------------------------------
IO Size Wash Size Run Size APF%   LR/sec   PR/sec   Hit%   APF-Eff% Usage%
------- ---------- ------- ------ ------- -------- ------ -------- ------
4 Kb     3276 Kb    16.00 Mb  10.00     0.47     0.13  71.43    n/a     0.20
2 Kb    17200 Kb    84.00 Mb  10.00    40.63    22.43  44.79    n/a     3.37
```

APF-Eff% — 异步预取使用的页数和异步预取引入的总页数。

Usage% 跟踪是否引用了引入缓冲池中的页，并提供缓冲池中引用的页与缓冲池的运行大小的比率。

- 对象部分 — 报告有关间隔结束时占用高速缓存的对象的统计信息。使用 topN 参数可限制这一部分的大小。对象按 PR/sec 的升序顺序显示。

```
Object Statistics
------------------------------------------------------------------------------
Object                       LR/sec  PR/sec   Hit%  Obj_Cached% Cache_Occp%
------------------------------ ------- ------- ------ ----------- -----
empdb.dbo.t1                   0.57    0.30  47.06      56.25       0.02
empdb.dbo.t2                   0.30    0.30   0.00      56.25       0.02
empdb.dbo.t3                   0.30    0.30   0.00      56.25       0.02

Object                               Obj Size  Size in Cache
------------------------------ -----------   ------------
empdb.dbo.t1                            32 Kb          18 Kb
empdb.dbo.t2                            32 Kb          18 Kb
empdb.dbo.t3                            32 Kb          18 Kb
```

- 建议部分 — 根据在采样间隔内收集的数据提供一组适用的建议：

```
The various recommendations are as follows:
Usage% for 'default data cache' is low (< 5%)
Usage% for  4k buffer pool in cache:default data cache is low (< 5%)
Consider using Named Caches or creating more cache partitions for
'default data cache' or both
Consider increasing the 'wash size' of the 2k pool for 'default data
cache'
Consider adding a large I/O pool for 'default data cache'
```

- 图例 — 解释输出中使用的各种术语。此处更为详细地介绍了输出中的一些术语。

高速缓存向导的输出样本

```
sp_sysmon '00:00:30', 'cache wizard'

===============================================================================
Cache Wizard
===============================================================================

------------------
default data cache
------------------
Run Size        :100.00 Mb   Usage%              :2.86
LR/sec          :41.10      PR/sec              :22.57   Hit%:45.09
Cache Partitions:4      Spinlock Contention%:0.00

Buffer Pool Information
-------------------------------------------------------------------------------
```

```
IO Size Wash Size  Run Size    APF%   LR/sec   PR/sec   Hit%   APF-Eff% Usage%
------- ---------- ----------- ------ -------- -------- ------ -------- ------
4 Kb     3276 Kb    16.00 Mb  10.00      0.47     0.13  71.43    n/a      0.20
2 Kb    17200 Kb    84.00 Mb  10.00     40.63    22.43  44.79    n/a      3.37

(1 row affected)

Object Statistics

-------------------------------------------------------------------------------
Object                          LR/sec  PR/sec  Hit%   Obj_Cached% Cache_Occp%
------------------------------- ------- ------- ------ ----------- -----------
tempdb.dbo.t1                     0.57    0.30  47.06       56.25        0.02
tempdb.dbo.t2                     0.30    0.30   0.00       56.25        0.02
tempdb.dbo.t3                     0.30    0.30   0.00       56.25        0.02
tempdb.dbo.t4                     0.30    0.30   0.00       56.25        0.02
tempdb.dbo.t5                     0.30    0.30   0.00       56.25        0.02
tempdb.dbo.t6                     0.30    0.30   0.00       56.25        0.02
tempdb.dbo.t8                     0.30    0.30   0.00       56.25        0.02
tempdb.dbo.t7                     0.57    0.20  64.71       62.50        0.02
tempdb.dbo.tempcachedobjstats     3.63    0.00 100.00       50.00        0.01
tempdb.dbo.tempobjstats           0.47    0.00 100.00       25.00        0.00

Object                          Obj Size    Size in Cache
------------------------------- ----------- -------------
tempdb.dbo.t1                         32 Kb        18 Kb
tempdb.dbo.t2                         32 Kb        18 Kb
tempdb.dbo.t3                         32 Kb        18 Kb
tempdb.dbo.t4                         32 Kb        18 Kb
tempdb.dbo.t5                         32 Kb        18 Kb
tempdb.dbo.t6                         32 Kb        18 Kb
tempdb.dbo.t8                         32 Kb        18 Kb
tempdb.dbo.t7                         32 Kb        20 Kb
tempdb.dbo.tempcachedobjstats         16 Kb         8 Kb
tempdb.dbo.tempobjstats               16 Kb         4 Kb

---------
company_cache
---------
Run Size        :1.00 Mb   Usage%              :0.39
LR/sec          :0.07      PR/sec              :0.07   Hit%:0.00
Cache Partitions:1      Spinlock Contention%:0.00

Buffer Pool Information
-------------------------------------------------------------------------------
IO Size Wash Size  Run Size    APF%   LR/sec   PR/sec   Hit%   APF-Eff% Usage%
```

```
------- ---------- ----------- ------ -------- -------- ------ -------- ------
2 Kb      204 Kb     1.00 Mb  10.00     0.07     0.07   0.00    n/a     0.39

Object Statistics
--------------------------------------------------------------------------------

Object               LR/sec  PR/sec  Hit%   Obj_Cached% Cache_Occp%
-------------------- ------- ------- ------ ----------- -----------
empdb.dbo.history      0.07    0.07   0.00       25.00        0.39

Object               Obj Size    Size in Cache
-------------------- ----------- -------------
empdb.dbo.history         16 Kb        4 Kb

-------------
companydb_cache
-------------
Run Size        :5.00 Mb   Usage%              :100.00
LR/sec          :380.97      PR/sec              :56.67   Hit%:85.13
Cache Partitions:1      Spinlock Contention%:0.00

Buffer Pool Information
--------------------------------------------------------------------------------
IO Size Wash Size  Run Size    APF%   LR/sec   PR/sec   Hit%   APF-Eff% Usage%
------- ---------- ----------- ------ -------- -------- ------ -------- ------
2 Kb     1024 Kb     5.00 Mb  10.00   380.97    56.67  85.13  98.42   100.00


Object Statistics
--------------------------------------------------------------------------------

Object                          LR/sec  PR/sec  Hit%   Obj_Cached% Cache_Occp%
------------------------------ ------- ------- ------ ----------- -----------
company_db.dbo.emp_projects      41.07   22.80  44.48       19.64        9.45
company_db.dbo.dept_det          93.03   20.67  77.79       99.08       54.53
company_db.dbo.emp_perf         116.70    2.63  97.74       97.77       34.18
company_db.dbo.dept_locs          0.43    0.17  61.54       50.00        0.16

Object                          Obj Size    Size in Cache
------------------------------ ----------- -------------
company_db.dbo.emp_projects       2464 Kb      484 Kb
company_db.dbo.dept_det           2818 Kb     2792 Kb
company_db.dbo.emp_perf           1790 Kb     1750 Kb
company_db.dbo.dept_locs            16 Kb        8 Kb

TUNING RECOMMENDATIONS
```

```
------------------------------------------------------------------------
Usage% for 'default data cache' is low (< 5%)
Usage% for  4k buffer pool in cache:default data cache is low (< 5%)
Usage% for  2k buffer pool in cache:default data cache is low (< 5%)

Usage% for 'company_cache' is low (< 5%)
Usage% for  2k buffer pool in cache:company_cache is low (< 5%)


 Consider adding a large I/O pool for 'companydb_cache'

LEGEND
------
LR/sec     - number of logical reads per second, i.e. sum of cache & disk reads
PR/sec     - number of physical reads per second i.e. disk reads
Run Size- size of cache or buffer pool in Kilobytes

Cache Partitions- number of cache partitions
Spinlock Contention%- Percentage spinlock contention for the cache
Hit%       - ratio of hits to total searches

Usage%     - ratio of pages referenced to Run Size

Wash Size- wash size of buffer pool in Kilobytes
APF%       - asynchronous prefetch % for this buffer pool
APF-Eff%- Ratio of buffers found in cache and brought in because
               of APF to the number of APF disk reads performed

Object     - combination of db, owner, object and index name
Obj Size- size of the object in Kilobytes
Size in Cache- size occupied in cache in Kilobytes at the end of sample
Obj_Cached%- Ratio of 'Size in Cache' to 'Obj Size'
Cache_Occp%- Ratio of 'Size in Cache' to 'Run Size' of cache
```

内核使用率

注释　“Kernel Utilization”部分报告的信息取决于您运行 Adaptive Server 时采用的模式：线程模式或进程模式（对于进程模式，请参见第 29 页的 “内核利用率 — 进程模式”）。

报告 Adaptive Server 活动。它将报告 CPU 可供 Adaptive Server 使用时 Adaptive Server 引擎的繁忙程度、 CPU 将控制权交给操作系统的频率、引擎检查网络和磁盘 I/O 的次数以及每次检查时所找到的处于等待状态的引擎的平均 I/O 数。

内核利用率 — 线程模式

样本输出

下面的样本显示了有一个线程池 syb_default_pool 和三个 Adaptive Server 引擎的环境中 “内核利用率” 的 sp_sysmon 输出。

内核使用率

```
------------------
 Engine Utilization (Tick %)   User Busy   System Busy    I/O Busy      Idle
 -----------------------  ------------  ------------  ----------  --------

 ThreadPool :syb_default_pool
    Engine 0                      8.7 %         0.8 %      31.1 %    59.5 %
    Engine 1                      8.0 %         0.1 %      33.2 %    58.7 %
    Engine 2                      8.8 %         0.3 %      32.7 %    58.2 %
    Engine 3                     15.0 %         0.3 %      32.6 %    52.0 %
   -------------------------  ------------  ------------  ----------  -------
   Server Summary      Total        40.4 %         1.6 %     129.6 %  228.4 %
                     Average        10.1 %         0.4 %      32.4 %   57.1 %

 Average Runnable Tasks               1 min        5 min     15 min  % of total
 -------------------------      ----------    ----------   -------   --------

 ThreadPool :syb_default_pool
    Global Queue                      0.0           0.0        0.0     0.0 %
    Engine 0                          0.3           0.2        0.1    61.7 %
    Engine 1                          0.0           0.0        0.0     0.7 %
    Engine 2                          0.1           0.1        0.0    17.3 %

   -------------------------  ------------  ------------  ---------
   Pool Summary        Total           0.5           0.3        0.1
                     Average           0.1           0.1        0.0

   -------------------------  ------------  ------------  ---------
   Server Summary      Total           0.5           0.3        0.1
                     Average           0.1           0.1        0.0

 CPU Yields by Engine              per sec      per xact      count % of total
```

```
 -------------------------        --------    ----------   --------  --------
ThreadPool :syb_default_pool
 Engine 0
    Full Sleeps                        20.4          13.8        2442     4.2 %
    Interrupted Sleeps                 81.7          55.4        9800    16.7 %
 Engine 1
    Full Sleeps                        22.8          15.5        2740     4.7 %
    Interrupted Sleeps                 89.1          60.4       10697    18.3 %
 Engine 2
    Full Sleeps                        19.9          13.5        2390     4.1 %
    Interrupted Sleeps                 93.4          63.3       11202    19.1 %
 Engine 3
    Full Sleeps                        17.8          12.1        2133     3.6 %
    Interrupted Sleeps                142.6          96.7       17113    29.2 %
 -------------------------   ------------  ------------  ----------
 Pool Summary                      487.6         330.6       58517

 -------------------------   ------------  ------------  ----------
 Total CPU Yields                  487.6         330.6       58517

Thread Utilization (OS %)       User Busy   System Busy        Idle
 -------------------------   ------------  ------------  ----------
 ThreadPool :syb_blocking_pool :no activity during sample

 ThreadPool :syb_default_pool
  Thread 6    (Engine 0)            0.0 %         0.0 %      100.0 %
  Thread 7    (Engine 1)            0.2 %         0.0 %       99.8 %
  Thread 8    (Engine 2)            0.0 %         0.0 %      100.0 %
 -------------------------   ------------  ------------  ----------
 Pool Summary        Total           0.2 %         0.0 %      299.9 %
                   Average           0.1 %         0.0 %       99.9 %

 ThreadPool :syb_system_pool
  Thread 10   (NetController)        0.1 %         0.0 %       99.9 %
  -------------------------   ------------  ------------  ----------
 Pool Summary        Total           1.1 %         0.0 %      400.0 %
                   Average           0.0 %         0.0 %      100.0 %

 -------------------------   ------------  ------------  ----------
 Server Summary      Total           0.2 %         0.0 %     1099.8 %
                   Average           0.0 %         0.0 %      100.0 %

Adaptive Server threads are consuming 0.0 CPU units.
Throughput (committed xacts per CPU unit) :275.0

 Page Faults at OS                 per sec      per xact       count % of total
```

```
 -------------------------       --------    ----------    --------  --------
  Minor Faults                        0.6           0.4          69    100.0 %
  Major Faults                        0.0           0.0           0      0.0 %
 -------------------------       --------    ----------    --------  --------
 Total Page Faults                    0.6           0.4          69    100.0 %

 Context Switches at OS            per sec      per xact       count % of total
 -------------------------       --------    ----------    --------  --------
  ThreadPool :syb_blocking_pool
    Voluntary                         0.0           0.0           0      0.0 %
    Non-Voluntary                     0.0           0.0           0      0.0 %
  ThreadPool :syb_default_pool
    Voluntary                        43.6         130.7        1307     56.3 %
    Non-Voluntary                     0.2           0.6           6      0.3 %
  ThreadPool :syb_system_pool
    Voluntary                        33.7         101.0        1010     43.5 %
    Non-Voluntary                     0.0           0.0           0      0.0 %
   -------------------------  -----------  ------------  ----------  ---------
 Total Context Switches              77.4         232.3        2323    100.0 %

 CtlibController Activity          per sec      per xact       count  % of total
 -------------------------       --------    ----------    --------  --------
  Polls                               1.0           0.7         120       n/a
  Polls Returning Events              0.0           0.0           0      0.0 %

DiskController Activity            per sec      per xact       count % of total
 -------------------------       --------    ----------    --------  --------
  Polls                            35019.1       23741.8     4202292       n/a
  Polls Returning Events             351.2         238.1       42145      1.0 %
  Polls Returning Max Events           0.0           0.0           0      0.0 %
  Total Events                       374.9         254.1       44984       n/a
  Events Per Poll                      n/a           n/a       0.011       n/a

NetController Activity             per sec      per xact       count % of total
 -------------------------       --------    ----------    --------  --------
  Polls                             7919.5        5369.1      950338       n/a
  Polls Returning Events            2537.8        1720.5      304536     32.0 %
  Polls Returning Max Events           0.0           0.0           0      0.0 %
  Total Events                      2537.8        1720.5      304536       n/a
  Events Per Poll                      n/a           n/a       0.320       n/a

 Blocking Call Activity            per sec      per xact       count % of total
 -------------------------       --------    ----------    --------  --------
  Total Requests                      0.0           0.0           0       n/a
```

Engine Utilization (Tick %)

使用内部测量方式 （“时钟周期”）报告 Adaptive Server 内核在每个 Adaptive Server 引擎上忙于执行任务的时间 （而非空闲时间）的百分比。输出根据线程池分组。“Server Summary”显示所有池的总时间。

“Engine Utilization (Tick %)”可帮助确定 Adaptive Server 引擎是过多还是过少。

“Engine Utilization (Tick %)”报告中显示的值可能与操作系统工具所报告的 CPU 使用率值不同。当 Adaptive Server 没有要处理的任务时，它将进入循环，在休眠之前查找工作。alter thread pool ... idle timeout 参数控制 Adaptive Server 引擎在放弃 CPU 之前查找可运行任务时在循环中花费的时间长度 （以毫秒为单位）。

若要缩短 Adaptive Server 检查可运行任务所花费的时间，请减小 idle timeout 线程池参数的值。不过，减小 idle timeout 的值可能会增加延迟，从而降低性能。

当 sp_sysmon 对计数器进行采样时 （缺省情况下，每 100 毫秒采样一次），每个引擎均指示当前正在执行的操作。 sp_sysmon 报告引擎的繁忙程度，以区分用户任务和系统任务：

- “User Busy”— 用户任务，例如用户连接。
- “System Busy”— 内部任务，例如管家。

例如，如果执行某项任务，引擎将报告 `"CPU Busy"`；如果空闲，引擎将报告 `"Idle"`。如果 Adaptive Server 有任何未完成的 I/O 而引擎处于空闲状态，则引擎将被视为 `"I/O Busy"`。如果有一个未完成的 I/O 和三个处于空闲状态的引擎，则每个引擎都将被视为 `"I/O Busy"`。

通过检查 sp_sysmon 输出中的问题以及进行调优以缓解争用，即使引擎报告的繁忙时间在 80-90% 范围内，响应时间仍旧非常快。如果此值一直很高 （大于 90%），则增加引擎可能会缩短响应时间和提高吞吐量。

“Engine Utilization (Tick %)”值是采样间隔期间的平均值，因此，如果平均值非常高，则表明该引擎在采样间隔的部分时间内可能处于 100% 繁忙状态。如果服务器的 CPU 使用率很高，请检查是否存在螺旋锁争用。内存表扫描也会增加 CPU 使用率，您可以通过相应地调整查询来降低 CPU 使用率。

当引擎利用率极其高时， housekeeper 清洗任务只将很少的页或不将任何页写入磁盘中 （因为它仅在 CPU 空闲周期内运行）。这意味着检查点可找到许多需要写入磁盘的页，而且检查点进程、大型批处理作业或数据库转储可能会在一段时间内将 CPU 使用率提高到 100%，从而导致响应时间的明显增加。

如果“Engine Utilization (Tick %)”百分比一直很高，而且您要通过添加 Adaptive Server 引擎缩短响应时间并增加吞吐量，则可在添加每个引擎后检查其它方面的资源争用是否增加。

在一个Adaptive Server为许多用户提供服务的环境中，性能通常会在各引擎间相当均匀地分布。然而，当引擎数大于任务数时，则可能会使某些引擎利用率百分比很高，而其它引擎却可能会空闲。

例如：

```
Engine Utilization (Tick %)      User Busy   System Busy    I/O Busy      Idle
   -------------------------  ------------  ------------  ----------  -------
   ThreadPool :Marketing_pool
    Engine 3                        78.0 %         4.5 %       3.4 %   18.0 %

   ThreadPool :syb_default_pool
    Engine 0                         8.7 %          .8 %       1.3 %   94.8 %
    Engine 1                        87.0 %         5.2 %       3.5 %   19.4 %
    Engine 2                        65.7 %         3.7 %       3.7 %   12.8 %
```

在 SMP 环境下，任务与引擎间具有软性密切连接。如果不存在可能会将任务放置在全局运行队列中的其它活动 （如锁争用），则此任务会继续在同一引擎上运行。

Average Runnable Tasks

使用 monSysLoad 监控表中的可运行任务平均值来提供 1 分钟、5 分钟和 15 分钟内的可运行任务平均值。引擎上所有正在运行 （和可运行）的任务都包含在该平均值中。分钟间隔是固定的，不会随着您配置的 sp_sysmon 采样时间变化。

通过可运行任务数，可以很好地衡量 Adaptive Server 的繁忙程度。例如，与平均“Engine Utilization”为 90% 且可运行任务数很低的服务器相比，平均 “Engine Utilization”为 90% 且可运行任务数很高的服务器更有可能从额外的引擎中获益。

比较 1 分钟、5 分钟和 15 分钟内的采样可帮助您确定服务器上的负载是在增加、保持稳定还是在减少。

CPU Yields by Engine

报告每个 Adaptive Server 引擎将控制权交给操作系统的次数。在引擎交出控制权后，它将进入休眠状态并持续短暂的一段时间。如果在唤醒后发现没有可运行的任务，它会再次进入休眠状态。输出内容根据线程池以及与线程池关联的引擎排列。对于每个引擎， sp_sysmon 都报告以下内容：

- Full Sleeps — 引擎在其完全休眠间隔内保持休眠状态 （即，在休眠时未接收可运行任务）。完全休眠很可能导致另一次休眠。不过，完全休眠不会增加延迟。
- Interrupted Sleeps — 引擎被可运行任务从休眠状态提前唤醒。中断式休眠很可能导致预定任务。中断式休眠可能会增加延迟。

“% of total”数据是一个引擎的放弃次数占所有引擎的组合放弃次数的百分比。

“Total CPU Yields”报告基于所有引擎的组合数据。

当引擎不忙时，它会在与 idle timeout 参数有关的一段时间后放弃 CPU。

- Engine Utilization 低/CPU Yields 低 — I/O Busy% 列的值大于 CPU Busy% 列的值表示存在大量未完成的 I/O。若要解决 I/O 处理过程中的瓶颈，请查看您可以在操作系统级别进行的更改。
- Engine Utilization 低/CPU Yields 高 — 引擎处于不活动状态。
- Engine Utilization 高/CPU Yields 低 — Adaptive Server 非常繁忙，并且有作业需要运行。添加引擎可能有助于提高性能。该值还应该与较高的 “Average Runnable Tasks”值相关。
- Engine Utilization 高/CPU Yields 高 — 这应该是有正常数量的用户连接在运行简单查询时出现的情况。添加引擎可能没有帮助，除非 “Engine Utilization”非常高或 “Average Runnable Tasks”数很高。
- 大多数引擎放弃应该属于 “完全休眠”。如果 “中断式休眠”占引擎总放弃数的 20% 以上，则总体延迟可能会增加。若要降低延迟，可考虑增加 idle timeout 的值。不过，这会提高 CPU 使用率。

请参见 《参考手册：命令》中的 alter thread pool。

Thread Utilization (OS %)

显示每个 Adaptive Server 线程在操作系统级别花费的时间。这部分时间是线程实际使用 CPU 的时间。“Pool Summary”表示在线程池中使用的线程的百分比。“Server Summary”描述在 Adaptive Server 中使用的线程的百分比。

“User time”报告在 sp_sysmon 采样间隔期间内线程在用户空间内执行代码 （即，在 Adaptive Server 中运行）的时间百分比。“System time”报告在 sp_sysmon 采样间隔期间内线程在系统空间内执行代码 （即，在操作系统内核中运行）的时间百分比。这种区别与 “Engine Utilization (Tick %)”部分中报告的 “User Busy”和 “System Busy”无关。

注释 由于 Adaptive Server 收集数据的方式所致， sp_sysmon 可能会显示某些线程的利用率超过 100%。

在检查 “Thread Utilization (OS%)”输出时，应考虑：

- “Thread Utilization (OS %)”值应高于“Engine Utilization (Tick %)”值。当 idle timeout 设置为较高的值，而负载不连续时，它们之间的差异最大。例如，100% 空闲但 idle timeout 为 -1 的引擎显示 0% 的引擎利用率但却显示 100% 的线程利用率。
- 大于 “Thread Utilization (OS %)”值的 “Engine Utilization (Tick %)”值可能表示主机上的 CPU 资源不足。这通常表示引擎没有收到所需的足够 CPU 时间，这可能导致性能严重下降。如果还涉及到螺旋锁争用，则性能可能会严重下降。
- 磁盘或网络任务的高 “Thread Utilization (OS %)”值可能表示系统需要一项额外的磁盘或网络任务。例如，如果网络任务线程的繁忙率为 80%，但引擎利用率很低，则达到饱和状态的网络任务很可能使引擎无工作可做。使用 sp_configure "number of network tasks" 添加网络任务可以减轻这种状况。

在 “Thread Utilization (OS %)”末尾， sp_sysmon 报告 Adaptive Server 线程使用的 CPU 单元数以及每个 CPU 的提交事务的吞吐量。例如：

```
Adaptive Server threads are consuming 5.6 CPU units.
Throughput is 8238.0 commited xacts per CPU unit.
```

可以将使用的 CPU 单元数与可供 Adaptive Server 使用的物理硬件进行比较。例如，如果系统有 8 个内核，每个内核 2 个线程，则总共有 16 个可用 CPU 单元。如果 Adaptive Server 使用了 6 个 CPU 单元，则主机就可以执行其它工作。但是，如果 Adaptive Server 使用了 14 个 CPU 单元，剩下的容量就非常少，子内核线程就会被过度使用。

Page Faults at OS

报告主要和次要页面错误数以及所有页面错误的摘要。仅当报告了页面错误时， sp_sysmon 才会显示该部分。

注释　对于 Windows 系统，不会出现 “Page Faults at OS”。

- Minor Faults — 当 Adaptive Server 需要的内存页在内存管理单元中未标记为可用但在物理内存中却可用时，会出现此错误。
- Major Faults — 当 Adaptive Server 需要的内存页在物理内存中不可用，必须从磁盘检索该页时，会出现此错误。

次要和主要页面错误均表示物理内存可能不足。主要页面错误对性能的影响远大于次要页面错误。主要页面错误表示对于主机而言，您将 Adaptive Server 内存配置设置得过高，或者其它进程 （如文件系统高速缓存）正在使用 Adaptive Server 所需的物理页。

Context Switches at OS

报告 Adaptive Server 线程在操作系统级别执行的自愿和非自愿环境切换数。仅当报告了环境切换，并且操作系统支持收集数据时， sp_sysmon 才会显示该部分。

注释　对于 Windows、 Red Hat 5 或 SLES 11 系统，不会出现 “Page Faults at OS”。

Adaptive Server 可能出于多种原因执行 “自愿” 环境切换，例如由于 I/O 轮询或引擎休眠。环境切换的数量取决于负载。当操作系统在 Adaptive Server 线程未请求删除的情况下从 CPU 中删除该线程时 （例如，当操作系统抢占该线程时），即发生 “非自愿” 环境切换。“非自愿” 环境切换可能表示主机的 CPU 资源不足，以及 Adaptive Server 性能可能受到严重影响。出现少量非自愿环境切换是正常的，这并不表示存在问题。

下面的样本输出来自一个未过度配置的系统。“非自愿” 环境切换数非常低：

```
Context Switches at OS          per sec      per xact      count   % of total
 -------------------------  -----------  ------------  ---------  ----------
  Voluntary                       278.7         278.7       2787       99.7 %
  Non-Voluntary                     0.9           0.9          9        0.3 %
 -------------------------  -----------  ------------  ---------  ----------
Total Context Switches            279.6         279.6       2796      100.0 %
```

下面的样本输出来自一台过载的计算机，“非自愿”环境切换级别非常高：

```
Context Switches at OS          per sec      per xact      count   % of total
 -------------------------  -----------  ------------  ---------   ----------
  Voluntary                     2683.7       16370.4     163704       18.4 %
  Non-Voluntary                11893.0       72547.4     725474       81.6 %
 -------------------------  -----------  ------------  ---------   ----------
Total Context Switches          14576.7       88917.8     889178      100.0 %
```

为了改善这种情况，可减少引擎数，从主机中删除非 Adaptive Server 负载，添加更多 CPU，或者迁移到具有更大处理能力的主机。

CtlibController Activity、 DiskController Activity 和 NetController Activity

“CtlibController Activity”描述 Adaptive Server 检查 Client Library (CTLIB) 控制器事件的频率（在 `Polls` 行中指示）以及 Client Library 返回事件的次数 （在 `Polls Returning Events` 行中指示）。

“DiskController Activity”描述 Adaptive Server 检查磁盘控制器事件的频率（在 `Polls` 行中指示）以及磁盘返回事件的次数（在 `Polls Returning Events` 行中指示）。

“NetController Activity”描述 Adaptive Server 检查网络控制器事件的频率（在 `Polls` 行中指示）以及网络返回事件的次数（在 `Polls Returning Events` 行中指示）。

这些部分包含以下行：

- Polls — 此控制器轮询操作系统以了解完成情况的次数。
- Polls Returning Events — 此控制器轮询操作系统并至少收到一个事件的次数。
- Polls Returning Max Events — 操作系统轮询返回最大事件数的次数。
- Total Events — 返回的 I/O 事件总数。
- Events Per Poll — 每个池中返回的平均事件数。

如果配置多个磁盘或网络任务，则显示的数据将是该控制器的所有任务的集合。

在以下情况下，可考虑添加磁盘或网络任务：

- “Polls Returning Max Events”大于零。
- “Events per Poll”的值大于三。

但是，在添加其它磁盘或网络任务之前，应检查控制器的 “Thread Utilization”值和整体系统负载。

Blocking Call Activity

报告阻塞调用任务 （即，对 syb_blocking_pool 的请求）。如果存在很大比例的排队请求或需要等待很长时间，可考虑调整 syb_blocking_pool 的大小。

内核利用率 — 进程模式

报告 Adaptive Server 活动。它将报告 CPU 可供 Adaptive Server 使用时 Adaptive Server 引擎的繁忙程度、 CPU 将控制权交给操作系统的频率、引擎检查网络和磁盘 I/O 的次数以及每次检查时所找到的处于等待状态的引擎的平均 I/O 数。

样本输出

下面的样本显示了有八个 Adaptive Server 引擎的环境中 “内核利用率”的 sp_sysmon 输出。

```
Kernel Utilization
------------------

Your Runnable Process Search Count is set to 2000
and I/O Polling Process Count is set to 10

Engine Busy Utilization       CPU Busy    I/O Busy      Idle
-------------------------     --------    --------  --------
Engine 0                         3.3 %      13.7 %     83.0 %
Engine 1                         2.4 %       9.2 %     88.4 %
Engine 2                         4.9 %      19.9 %     75.2 %
Engine 3                          .8 %      19.1 %     76.1 %
Engine 4                         2.8 %       7.0 %     90.2 %
Engine 5                         1.9 %       7.3 %     90.8 %
Engine 6                         2.5 %       7.6 %     89.9 %
Engine 7                         2.0 %       8.1 %     89.9 %
-------------------------     --------    --------   --------
Summary              Total      24.6 %      91.9 %    683.5 %
Average                          3.1 %      11.5 %     85.4 %

CPU Yields by Engine           per sec      per xact       count   % of total
-------------------------  -----------   -----------   ---------   ---------
```

```
Engine 0                        31.5          0.0        2802      11.6 %
Engine 1                        38.1          0.0        3388      14.0 %
Engine 2                        21.8          0.0        1936       8.0 %
Engine 3                        30.2          0.0        2689      11.1 %
Engine 4                        37.9          0.0        3372      13.9 %
Engine 5                        38.4          0.0        3421      14.1 %
Engine 6                        36.1          0.0        3217      13.3 %
Engine 7                        38.4          0.0        3420      14.1 %
-------------------------   ----------   -----------    --------
Total CPU Yields               272.4          0.2       24245

Network Checks
Non-Blocking                434722.4        366.8    38690292      99.9 %
Blocking                       272.4          0.2       24247       0.1 %
-------------------------   ----------   -----------    --------
Total Network I/O Checks    434994.8        367.1    38714539
Avg Net I/Os per Check           n/a          n/a     0.00003         n/a

Disk I/O Checks
Total Disk I/O Checks       435855.6        367.8    38791149         n/a
Checks Returning I/O        125358.1        105.8    11156875      28.8 %
Avg Disk I/Os Returned           n/a          n/a     0.00313         n/a
```

Engine Busy Utilization

报告 Adaptive Server 内核在每个 Adaptive Server 引擎上忙于执行任务的时间（而非空闲时间）的百分比。摘要行给出所有组合引擎的总活动时间和平均活动时间。

“Engine Busy Utilization”报告中显示的值可能与操作系统工具所报告的 CPU 使用率值不同。当 Adaptive Server 没有要处理的任务时，它将进入循环，定期检查是否有网络 I/O、是否有已完成的磁盘 I/O 以及运行队列中是否有任务。

无法在 Adaptive Server 中进行的测量是 Adaptive Server 控制 CPU 的时间相对于操作系统使用 CPU 的时间这两者的百分比。操作系统向 Adaptive Server 提供 CPU 时间。 sp_sysmon 报告 Adaptive Server 如何使用操作系统提供的时间。应始终将 sp_sysmon 与操作系统的监控工具结合使用以获取这两个系统的读数。

查看操作系统文档，以了解正确的命令。

若要缩短 Adaptive Server 在空闲时检查 I/O 所花费的时间，可以降低 sp_configure 参数 runnable process search count。该参数指定 Adaptive Server 引擎在交出 CPU 控制权之前其循环操作查找可运行任务的次数。不过，减小 runnable process search count 的值可能会增加延迟，从而降低性能。

“Engine Busy Utilization”衡量 Adaptive Server 引擎在给定 CPU 时间内的繁忙程度。如果在 10 分钟采样间隔内，Adaptive Server 可以利用引擎的时间为 80%，且 “Engine Busy Utilization”为 90%，则意味着 Adaptive Server 的繁忙时间为 7 分 12 秒，而空闲时间为 48 秒。

当 sp_sysmon 对计数器进行采样时 （缺省情况下，每 100 毫秒采样一次），每个引擎均指示当前正在执行的操作。例如，如果正在执行任务，它会报告 "CPU Busy"；如果处于空闲状态，它会报告 "Idle"；如果处于空闲状态，并且至少有一个未完成的异步磁盘 IO，它会报告 "IO Busy"。

“Engine Busy Utilization”可帮助确定 Adaptive Server 引擎是过多还是过少。 Adaptive Server 的高可伸缩性归功于可避免资源争用的调优机制。

通过检查 sp_sysmon 输出中的问题以及进行调优以缓解争用，即使 “Engine Busy”值在 80 - 90% 范围内，响应时间仍旧非常快。如果此值一直很高 （大于 90%），则增加引擎可能会缩短响应时间和提高吞吐量。

“Engine Busy Utilization”值是采样间隔期间的平均值，因此，如果平均值非常高，则表明该引擎在采样间隔的部分时间内可能处于 80% 繁忙的状态。螺旋锁争用会增加 CPU 使用率，如果服务器的 CPU 使用率很高，则应检查是否存在螺旋锁争用。内存表扫描也会增加 CPU 使用率，您可以通过相应地调整查询来降低 CPU 使用率。

当引擎利用率极其高时， housekeeper 清洗任务将很少的页或不将任何页写入磁盘中 （因为它仅在 CPU 空闲周期内运行）。这表明检查点可找到许多需要写入磁盘的页，而且检查点进程、大型批处理作业或数据库转储可能会在一段时间内将 CPU 使用率提高到 100%，从而导致响应时间的明显增加。

如果 “Engine Busy Utilization”百分比一直很高，而且您要通过添加 Adaptive Server 引擎缩短响应时间并增加吞吐量，则可在添加每个引擎后检查其它方面的资源争用是否增加。

在一个Adaptive Server为许多用户提供服务的环境中，性能通常会在各引擎间相当均匀地分布。然而，当引擎数大于任务数时，则可能会使某些引擎利用率百分比很高，而其它引擎却可能会空闲。例如：

```
Engine Busy Utilization          CPU Busy       I/O Busy        Idle
-------------------------        ---------       -------     --------
Engine 0                            68.7 %          2.5 %      28.8 %
Engine 1                            61.9 %          3.3 %      34.8 %
Engine 2                            67.0 %          2.4 %      30.6 %
Engine 3                            69.0 %          3.8 %      27.2 %
Engine 4                            60.2 %          2.7 %      37.2 %
Engine 5                            55.7 %          3.2 %      41.1 %
Engine 6                            53.8 %          3.2 %      43.0 %
-----------------------          --------       --------    --------
```

```
Summary                  Total       436.3 %          21.1 %     242.6 %
Average                                62.3 %           3.0 %      34.7 %
```

在 SMP 环境下，任务与引擎间具有软性密切连接。如果不存在可将任务放置在全局运行队列中的其它活动 （如锁争用），则此任务会继续在同一引擎上运行。

CPU Yields by Engine

每个 Adaptive Server 引擎将控制权交给操作系统的次数。“% of total”数据是一个引擎的放弃次数占所有引擎的组合放弃次数的百分比。

“Total CPU Yields”报告基于所有引擎的组合数据。但是，具有一个或多个未完成异步磁盘 I/O 的引擎不会交出 CPU 控制权，即使 runnable process search count 已经用完。

如果“Engine Busy Utilization”数据表明引擎利用率低，请使用“CPU Yields by Engine”确定 “Engine Busy Utilization”数据是否反映了真正不活动的引擎或其 CPU 经常被操作系统夺走的引擎。

引擎不忙时，它会在与 runnable process search count 参数有关的一段时间后放弃 CPU。“CPU Yields by Engine”的值较大时表明引擎主动放弃。

- Engine Busy 低/CPU Yields 低 — I/O Busy% 列的值高于 CPU Busy% 列的值表示存在大量未完成的 I/O。请查看您可以在操作系统级别进行的更改，以解决 I/O 处理过程中的瓶颈。如果 Adaptive Server 系统 CPU 时间过高（基于操作系统输出），请减小 runnable process search count 的值。
- Engine Busy 低/CPU Yields 高 — 引擎处于不活动状态。
- Engine Busy 高/CPU Yields 低 — Adaptive Server 非常繁忙，并且有作业需要运行。添加引擎可能有助于提高性能。
- Engine Busy 高/CPU Yields 低 — 这应该是有正常数量的用户连接运行简单查询时出现的情况。降低 runnable process search count 可能几乎没有什么效果。添加引擎也不太可能有帮助，除非 “Engine Busy Utilization”非常高。

请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

```
CPU Yields by Engine            per sec      per xact       count   % of total
------------------------   ------------   -----------   --------    ---------
Engine 0                          73.5           1.8      44087      26.2 %
Engine 1                          40.4           1.0      24224      14.4 %
Engine 2                          23.7           0.6      14208       8.4 %
Engine 3                          36.2           0.9      21746      12.9 %
Engine 4                          38.7           1.0      23207      13.8 %
```

```
Engine 5                          41.3           1.0       24760       14.7 %
Engine 6                          26.8           0.7       16108        9.6 %
----------------------     ------------    ----------   --------
Total CPU Yields                  280.6           6.9      168340
```

Network Checks

报告以下相关信息：阻塞和非阻塞网络 I/O 检查、间隔期间进行的 I/O 检查的总数以及每次网络检查的网络 I/O 的平均数。

Adaptive Server 具有两种检查网络 I/O 的方法：阻塞模式和非阻塞模式。

```
Network Checks
Non-Blocking                166371.5        4098.3    99822899       99.8 %
Blocking                       279.0           6.9      167388        0.2 %
-------------------------   ---------      --------   ---------
Total Network I/O Checks    166650.5        4105.2    99990287
Avg Net I/Os per Check           n/a           n/a     0.00476         n/a
```

Non-Blocking

Adaptive Server 执行非阻塞网络检查的次数。通过非阻塞网络 I/O 检查，引擎可检查网络 I/O 并继续进行处理，而无论是否发现 I/O 处于等待状态 （这是引擎检查网络 I/O 的通常方式）。

Blocking

Adaptive Server 执行阻塞网络检查的次数。这是引擎将 CPU 控制权交给操作系统的方式，应等于放弃次数 （由于计时问题，这可能会有所不同）。

引擎完成队列中的所有可运行任务后，将会循环短暂的一段时间，以轮询网络并检查下一客户端请求。如果在特定的循环次数（由 sp_configure 参数 runnable process search count 确定）后，引擎没有找到可运行任务，并且不存在未完成的异步磁盘 I/O，则引擎将执行阻塞网络 I/O 轮询，将其 CPU 控制权交给操作系统。此时，引擎就称为进入休眠状态。

休眠引擎每个时钟周期唤醒一次以执行例行管家任务和检查可运行任务。如果没有可运行任务，引擎再执行一次阻塞网络检查，然后重新进入休眠状态。

如果某个不同的引擎完成了某个网络 I/O，Adaptive Server 可能会在下一时钟周期之前唤醒休眠引擎。唤醒后，引擎将执行处理后 I/O 工作，这通常会导致任务成为可运行任务 （在 I/O 工作处理完毕之前，该任务不太可能再次交出 CPU 控制权）。

Total Network I/O Checks

引擎对进入包和发出包的轮询次数。此类别与 “CPU Yields by Engine” 结合使用时很有用处。

引擎处于空闲状态时，会进行循环，同时检查网络包。如果 “Network Checks”很低而 “CPU Yields by Engine”很高，则引擎可能过于频繁地放弃 CPU 而不能以足够的频率检查网络。如果系统能够承受此开销，或许可以接受减少放弃。

引擎循环您使用 runnable process search count 定义的次数后，才交出 CPU 控制权 （除非引擎至少有一个未完成的磁盘 I/O）。因此，引擎过于频繁地交出 CPU 控制权可能表示因为 runnable process search count 设置得过低而导致性能受到影响。

Average Network I/Os per Check

对于采样间隔期间进行的所有 Adaptive Server 引擎检查而言，每次检查的网络 I/O （发送和接收）平均数。

sp_configure 参数 i/o polling process count 指定 Adaptive Server 在调度程序检查磁盘和网络 I/O 是否完成之前运行的最大进程数。调整 i/o polling process count 既会影响响应时间，又会影响 Adaptive Server 的吞吐量。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

如果 Adaptive Server 引擎频繁进行检查，但不对网络 I/O 进行频繁检索，则可以尝试降低网络 I/O 检查的频率。

磁盘 I/O 检查次数

报告磁盘 I/O 检查的总数，以及返回 I/O 的检查数量。

```
磁盘 I/O 检查次数
Total Disk I/O Checks        171839.7        4233.0    103103833       n/a
Checks Returning I/O          31783.1         782.9     19069887    18.5 %
Avg Disk I/Os Returned            n/a           n/a       0.01722       n/a
```

- `Checks Returning I/O` — 实际检查次数。
- `Avg Disk I/Os Returned` — 产生一个或多个完成的 I/O 的 `Checks Returning I/O` 的百分比。

Total Disk I/O Checks

引擎进入例程以检查是否存在磁盘 I/O 的次数。io polling process count 配置参数控制此项检查的频率。

当某个任务需要执行 I/O 时，运行该任务的 Adaptive Server 引擎立即发出 I/O 请求并使任务休眠以等待 I/O 结束。此引擎可以处理其它任务，但继续检查已完成的 I/O。引擎发现已完成的 I/O，会将此任务从休眠队列移动到运行队列中。

Checks Returning I/O

在进入磁盘检查例程时 I/O 未完成的次数。

如果在服务器的一个时钟周期内或者在 "idle loops" 期间已经运行了 "io polling process count" 值所指定的任务数，则 Adaptive Server 引擎会轮询网络 I/O。但是，除非存在至少一个未完成的磁盘 IO，否则引擎不会轮询磁盘 IO。 "Checks Returning IO" 表示引擎试图检查磁盘 IO 的次数以及 Adaptive Server 试图检查并且至少存在一个未完成磁盘 IO 的次数。

例如，如果引擎检查预期的 I/O 100,000 次，则该平均值表示 I/O 实际挂起时间的百分比。在这 100,000 次检查中，如果 I/O 完成 10,000 次，则检查的 10% 有效，而另外 90% 为开销。

但是，仍应核对每次检查所返回的 I/O 平均数，以及采样间隔期间引擎的繁忙程度。如果采样包括空闲时间，或者 I/O 通信量时有时无，则繁忙期间可能会有很大比例的检查返回 I/O。

如果此类别的结果似乎较低或较高，则可配置 i/o polling process count 来增加或降低检查的频率。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Average Disk I/Os Returned

产生一个或多个完成的 I/O 的 “Checks Returning I/O”的百分比。

增加两次检查之间 Adaptive Server 引擎的等待时间可带来更理想的吞吐量，因为如果 Adaptive Server 引擎用较少的时间检查 I/O，就有更多的时间来进行处理。但是，这需要在所使用的环境中加以证实。如果 I/O Busy% 值很高，则 Adaptive Server 应该更频繁地执行检查，以便在 I/O 完成后立即选取它们。i/o polling process count 配置参数控制 Adaptive Server 执行此项检查的频率。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

工作进程管理

报告工作进程的使用情况，包括准许和拒绝的工作进程请求数，以及工作进程内存请求的成功和失败情况。

结合第 38 页的 “并行查询管理”的输出分析此输出。

样本输出

```
Worker Process Management
-------------------------
                                    per sec      per xact     count    %
of total
                                ------------  ------------  --------    -
--------
Worker Process Requests
Requests
Granted                         0.1           0.4         23       100.0 %
Requests
Denied                          0.0           0.0          0         0.0 %
------------------------        ------------  ------------    ------
Total Requests                           0.1           0.4         23

Requests
Terminated                      0.0           0.0          0         0.0 %

Worker Process Usage
Total
Used                               0.2           1.1         69         n/a
Max Ever Used During
Sample          0.0            0.1          9         n/a

Memory Requests for Worker Processes
Succeeded                                2.2          10.1        646
 100.0 %
Failed                                   0.0           0.0          0
   0.0 %
------------------------       -----------  ------------   -------
Total Requests                           2.2          10.1        646

Avg Mem Ever Used by a WP (in
bytes)  n/a           n/a      1208.9         n/a
```

Worker Process Requests

报告工作进程请求和工作进程内存请求。并行查询可能进行多次工作进程请求。例如，需要排序的并行查询可能进行一次访问数据请求和另一次并行排序请求。

“Requests Granted”和 “Requests Denied”行显示准许的请求数和由于执行时缺少可用的工作进程而拒绝的请求数。

若要查看对工作进程数进行调整的次数，请参见第 39 页的 “Parallel Query Usage”。

“Requests Terminated”报告通过用户操作终止请求的次数，如按 Ctrl+c 取消查询。

Worker Process Usage

报告在采样间隔期间所使用的工作进程总数。“Max Ever Used During Sample”报告 sp_sysmon 的采样期间的任意时刻所使用的最高值。您可以使用 “Max Ever Used During Sample”设置配置参数 number of worker processes。

Memory Requests for Worker Processes

为工作进程分配内存所进行的请求数，以及其中成功与失败的请求数。工作进程内存从使用参数 memory per worker process 配置的内存池进行分配。

如果 “Failed”为非零值，则可能需要增加 memory per worker process 的值。

Avg Mem Ever Used By a WP

报告在采样间隔期间的任一时刻所有活动工作进程所使用的最大平均内存。每个工作进程都需要内存，主要用于交换协调消息。此内存由 Adaptive Server 从全局内存池中进行分配。

池的大小由 number of worker processes 和 memory per worker process 两个配置参数的乘积确定。

如果将 number of worker processes 设置为 50，memory per worker process 设置为缺省值 1024 字节，则池中的可用值为 50K。将 memory for worker process 增加到 2048 字节将需要 50K 的额外内存。

启动时，将为每个工作进程创建静态结构。工作进程在使用中时，将根据需要从池中分配额外内存，并在不需要时解除分配。所输出的平均值是为所有工作进程分配的静态和动态内存平均值，是通过除以采样间隔期间实际使用中的工作进程数得出的。

如果配置了大量工作进程，而在采样间隔期间仅使用极少数，则会由于为不使用的进程平均分配静态内存而使输出值不真实。

如果“Avg Mem”接近 memory per worker process 设置的值，并且“Max Ever Used During Sample”中的工作进程数接近配置的数目，则可能需要增加此参数的值。

如果工作进程需要从缓冲池获得内存，然而无可用内存，此进程则输出错误消息并退出。

注释 对于大多数并行查询处理，缺省值 1024 已经足够了。

dbcc checkstorage 是一个例外，如果只配置了一个工作进程，它最多可使用 1792 字节。如果使用 dbcc checkstorage，并且 number of worker processes 设置为 1，则可能需要增加 memory per worker process。

并行查询管理

报告并行查询的执行情况。它报告并行查询的总数、运行时调整工作进程数的次数，以及合并和排序过程中授予的锁数。

样本输出

```
Parallel Query Management
-------------------------

Parallel Query Usage            per sec     per xact      count   % of total
------------------------   ------------   ----------  ---------   ----------
Total Parallel Queries              0.1          0.3         19          n/a
WP Adjustments Made
Due to WP Limit                     0.0          0.0          0        0.0 %
Due to No WPs                       0.0          0.0          0        0.0 %

Merge Lock Requests             per sec     per xact      count   % of total
------------------------   ------------   ----------  ---------    ----------
```

```
Network Buffer Merge Locks
Granted with no wait                1.1          5.0         320        21.3 %
Granted after wait                  3.9         18.4        1179        78.7 %

Result Buffer Merge Locks
Granted with no wait                0.0          0.0           0         0.0 %
Granted after wait                  0.0          0.0           0         0.0 %

Work Table Merge Locks
    Granted with no wait            0.0          0.0           0         0.0 %
    Granted after wait              0.0          0.0           0         0.0 %
------------------------   ------------   ----------  ---------    ----------
Total # of Requests                 5.0         23.4        1499

Sort Buffer Waits                 per sec     per xact      count   % of total
------------------------   ------------   ----------  ---------    ----------
Producer Waits                      0.0          0.1           8       100.0 %
Consumer Waits                      0.0          0.0           0         0.0 %
------------------------   ------------   ----------  ---------    ----------
Total # of Waits                    0.0          0.1           8
```

Parallel Query Usage

报告符合并行运行条件的查询总数。由优化程序确定最佳计划，决定以串行方式还是以并行方式运行查询以及决定用于并行查询的工作进程数。

“WP Adjustments Made”报告优化程序建议在运行时必须调整工作进程数的次数。会报告两种可能原因：

- “Due to WP Limit”— 指示由于 set parallel_degree 或 set scan_parallel_degree 设置的会话级限制，而对高速缓存查询计划的工作进程数进行调整的次数。

  如果 “Due to WP Limit”为非零值，则会查找设置了会话级限制的应用程序。

- “Due to No WPs”— 指示由于缺少可用工作进程而请求减少工作进程数的次数。这些查询可能以串行方式运行，或者用少于优化程序建议的工作进程以并行方式运行。这可能意味着使用未很好优化的计划运行查询。

  如果 “Due to No WPs”为非零值，并在系统处于典型负载状态时采样，则可能需要增大 number of worker processes 配置参数或为某些查询设置会话级限制。

使用调整后的计划对登录的 fid （系列 ID）运行 sp_showplan 将只显示高速缓存的计划，而不显示调整后的计划。

如果该登录运行调整后的计划，则 sp_who 显示的 fid 的工作进程数与 sp_showplan 结果所指示的不同。

Merge Lock Requests

已提出的并行合并锁请求数、已立即准许的请求数，以及必须等待每类合并的请求数。三种合并类型为：

- “Network Buffer Merge Locks”— 报告将结果返回给客户端的网络缓冲区的争用。
- “Result Buffer Merge Locks”— 报告用于处理未分组集合结果和未排序、非集合变量赋值结果的结果缓冲区争用。
- “Work Table Merge Locks”— 报告合并工作表中的结果时的锁争用。

“Total # of Requests”输出三种类型的合并请求的总数。

Sort Buffer Waits

报告用于并行排序的排序缓冲区争用。并行排序缓冲区由下列项目使用：

- 生产者 — 从并行扫描返回行的工作进程。
- 消费者 — 执行并行排序的工作进程。

如果等待的数目很多，您可以将 number of sort buffers 配置为较高的值。

有关准则，请参见 《性能和调优：查询处理和抽象计划》中的第 10 章“利用统计信息来提高性能”。

任务管理

提供有关已打开的连接、按引擎列出的任务环境切换以及按原因列出的任务环境切换的信息。`Task Management` 标识任务停止运行的原因（即，报告突出显示资源短缺和争用）。

样本输出

Task Management	per sec	per xact	count	% of total
Connections Opened	0.2	0.0	154	n/a
Task Context Switches by Engine				
Engine 0	269.9	3.4	240477	7.3 %
Engine 1	209.4	2.7	186574	5.7 %
Engine 2	198.4	2.5	176730	5.4 %
Engine 3	198.5	2.5	176859	5.4 %
Engine 4	278.4	3.5	248054	7.5 %
Engine 5	187.1	2.4	166697	5.1 %
Engine 6	202.7	2.6	180640	5.5 %
Engine 7	312.2	4.0	278129	8.5 %
Engine 8	215.7	2.7	192198	5.8 %
Engine 9	260.3	3.3	231940	7.1 %
Engine 10	235.1	3.0	209447	6.4 %
Engine 11	409.2	5.2	364604	11.1 %
Engine 12	225.8	2.9	201166	6.1 %
Engine 13	484.9	6.1	432020	13.1 %
Total Task Switches:	3687.5	46.7	3285535	
Task Context Switches Due To:				
Voluntary Yields	860.8	10.9	766929	23.3 %
Cache Search Misses	852.0	10.8	759122	23.1 %
System Disk Writes	12.7	0.2	11353	0.3 %
Exceeding I/O batch size	184.7	2.3	164550	5.0 %
Logical Lock Contention	0.3	0.0	258	0.0 %
Address Lock Contention	0.0	0.0	12	0.0 %
Latch Contention	0.7	0.0	587	0.0 %
Log Semaphore Contention	31.1	0.4	27737	0.8 %
PLC Lock Contention	0.6	0.0	577	0.0 %
Group Commit Sleeps	6.9	0.1	6122	0.2 %
Last Log Page Writes	71.4	0.9	63619	1.9 %
Modify Conflicts	19.9	0.3	17728	0.5 %
I/O Device Contention	0.0	0.0	0	0.0 %
Network Packet Received	152.1	1.9	135483	4.1 %
Network Packet Sent	643.7	8.1	573528	17.5 %
Network services	127.8	2.3	7564	8.9 %
Other Causes	850.7	10.8	757930	23.1%

Connections Opened

Adaptive Server 的已打开连接数。它包括任何类型的连接，如客户端连接和远程过程调用。它只将在采样间隔期间开始的连接计算在内；采样间隔开始前所建立的连接不包括在内，虽然它们可能处于活动状态并使用资源。

“Connections Opened”提供对 Adaptive Server 环境和间隔期间的负载的一般性了解。该数据对于了解应用程序的行为也十分有用，它可以帮助确定应用程序是否重复打开和关闭连接或者是否对每个连接执行多个事务。

有关提交的事务的信息，请参见第 59 页的 “事务配置文件”。

Task Context Switches by Engine

Adaptive Server 上运行的每个进程都有它自己的 “上下文”（即环境），其中包含有关当前数据库、客户端字符集、当前运行的查询等内容的信息。当 Adaptive Server 正在运行一个客户端或系统进程，但必须切换到另一个客户端或系统进程时，即发生 “环境切换”，这种切换涉及到更改引擎在其中运行的环境。

“Task Context Switches by Engine”报告每个 Adaptive Server 引擎将环境从一个用户任务切换到另一个用户任务的次数。“% of total”报告每个 Adaptive Server 引擎的引擎任务切换占所有 Adaptive Server 引擎加起来的任务切换总数的百分比。

“Total Task Switches”汇总了 SMP 服务器上所有引擎的任务切换活动。您可以使用 “Total Task Switches”来观察重新配置后的结果。如果任务似乎在高速缓存搜索未命中时发生阻塞并经常被切断，则可能需要重新配置高速缓存或增加内存。然后检查数据，以查看任务往往会更多还是更少地被切断。

Task Context Switches Due To

Adaptive Server 因许多常见原因切换环境的次数。“% of total”报告由于特定原因而发生的环境切换的次数占所有 Adaptive Server 引擎加起来进行的任务环境切换总数的百分比。

“Task Context Switches Due To”概述了出现任务关闭引擎这类情况的原因。可以通过检查其它 sp_sysmon 输出来研究本部分中所示的可能的性能问题，如描述这些原因的部分中所述。

例如，如果多数任务切换是由物理 I/O 引起的，请尝试通过添加更多内存或重新配置高速缓存来最大限度减少物理 I/O。然而，如果任务切换多数由于锁争用所导致，则应检查报告的锁定部分。

有关详细信息，请参见第 88 页的 “Lock Management”。

Voluntary Yields

任务完成但未放弃的次数。`Network Packet Sent` 列中表示完成后放弃的任务。自动放弃数值高表明极少存在争用。

配置参数 time slice 设置进程可运行的时间长度。不会因其它原因而断开的占用大量 CPU 的任务将会在代码的特定 “屈服点”放弃 CPU，从而使得其它进程可以使用 CPU。请参见第 53 页的 “Allotted Slices Exhausted”确定任务是否正在用尽其时间配额。

有关详细信息，请参见 《性能和调优系列：基础知识》中的 “使用引擎和 CPU”。

Cache Search Misses

因为所需的页不在高速缓存中且不得不从磁盘读取而断开任务的次数。对于数据和索引页，执行物理读取时将断开任务。

有关 sp_sysmon 输出中与高速缓存相关的部分的详细信息，请参见第 98 页的 “数据高速缓存管理”。

Exceeding I/O Batch Size

因为 Adaptive Server 超出 I/O 批处理限制而导致 I/O 密集型任务从引擎断开的次数。但是，“Exceeding I/O batch size”只报告从非日志 I/O 批处理进行的环境切换。 Adaptive Server 将调整磁盘写入速度，以防止在需要执行大量 I/O 的某些操作过程中将磁盘 I/O 子系统淹没。例如，检查点任务会将大量数据页写入磁盘。 Adaptive Server 会断开检查点任务，因此在批写完成之前，它一直处于休眠状态。然后检查点会唤醒并进行另一批处理。

使用 i/o batch size 参数配置非日志 I/O 和日志 I/O 的批处理大小。

缺省情况下，每个批处理的写入数设置为 100。下列情况下可能需要增加每个批处理的写入数：

- 您使用一个具有大量数据高速缓存的高吞吐量、事务密集型环境。
- 非 I/O 绑定的系统。

System Disk Writes

任务由于以下原因而断开的次数：

- 超出日志 I/O 批处理大小 （10 页）
- 需要执行磁盘写入
- 需要访问正由其它进程 （如管家或检查点进程）写入的页

多数 Adaptive Server 写入均为异步操作，但在页面拆分写入、恢复和 OAM 页面写入过程中，进程会进入休眠状态。

如果 “System Disk Writes”看上去较高，请检查页面拆分的值以查看问题是否起因于数据页和索引页拆分。

请参见第 79 页的 “Page Splits”。

如果系统磁盘写入值较高不是页面拆分引起的，则不能通过调优 Adaptive Server 来影响该值。

Logical Lock Contention

因为表、数据页或数据行上的锁争用而断开任务的次数。

通过检查事务明细和报告的锁管理部分研究锁争用问题。

- 请参见第 62 页的 “Transaction Detail”和第 88 页的 “Lock Management”。
- 检查查询是否在执行延迟更新和昂贵的直接更新，它们可能导致其它索引锁。请参见第 64 页的 “Updates”。
- 使用 sp_object_stats 以每个对象为基础报告信息。

  有关详细信息，请参见 Performance and Tuning Series: Locking （《性能和调优系列：锁定》）中的第 3 章 “Locking Reports”（锁定报告）。

有关锁和锁争用的其它帮助，请参见：

- Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的 “Types of Locks”（锁类型），其中提供有关可在服务器或查询级别使用的锁类型的信息。
- Performance and Tuning Series: Locking （《性能和调优系列：锁定》）中的第 2 章 “Locking Configuration and Tuning”（锁定配置和调优），其中提供有关减少锁争用的信息。

- Performance and Tuning Series: Locking （《性能和调优系列：锁定》）中的第 1 章 “Introduction to Locking”（锁定简介），其中提供有关索引和查询调优的信息。特别是，使用索引以确保更新和删除时未导致表扫描和排它表锁。

Address Lock Contention

因为地址锁而断开任务的次数。 Adaptive Server 在所有页锁定表的索引页上获取地址锁。这些锁会阻止访问数据页。例如，索引页上的排它地址锁会阻止另一系统进程 ID (spid) 读取该页并按该索引找到数据。但是，任务仍然可以使用其它访问方法访问数据。此外，当叶级索引页被以排它方式锁定时，将无法对锁定页上索引行所指向的任何数据进行任何修改，因为索引行可能也需要修改。

Latch Contention

因为需要等待闩锁而断开任务的次数。

如果用户表仅使用所有页锁定，则会在 DOL 锁定系统表或分配页上出现闩锁争用。

如果应用程序使用仅数据锁定，则此处报告的争用包括所有闩锁等待，包括索引页上和 OAM 页以及分配页上的闩锁等待。

在页分配过程中减少争用

在 SMP 环境中，插入和扩展更新非常多，以至于页分配频繁出现，分配页的锁存争用可能会降低性能。通常， Adaptive Server 在分配单元上为对象分配新页，此分配单元已由此对象使用并已知具有可用空间。

对于每个对象， Adaptive Server 会跟踪该分配页号，页号将作为提示，用于需要为该对象分配页的任何任务。当同一分配单元上有多个需要同时分配页的任务时，第二个和随后的任务将在分配页的闩锁上阻塞。

您可以指定 “贪心分配”方案，以便 Adaptive Server 保留对表进行页分配的八个分配提示的列表。

下面的命令为数据库 6 中的 salesdetail 表启用贪心分配：

```
dbcc tune(des_greedyalloc, 6, salesdetail, "on")
```

若要将其关闭，请使用：

```
dbcc tune(des_greedyalloc, 6, salesdetail, "off")
```

dbcc tune(des_greedyalloc) 的作用不是持久性的，所以重新启动 Adaptive Server 后必须重新发出相关命令。

只在具备以下所有条件时使用 dbcc tune(des_greedyalloc)：

- 有多个引擎。少于四个引擎时该命令几乎没有作用。
- 为此对象分配了大量的页。您可以使用 sp_spaceused 或 optdiag 跟踪页数。
- 闩锁争用计数器显示争用情况。

将表指定给它们自己的段时贪心分配的作用更大。如果在同一段上对几个表启用贪心分配，则同一分配提示可用于一个以上的表。

不允许在 master 和 tempdb 数据库中使用贪心分配，也不允许在系统表中使用。贪心页分配不适用于已分区的表。

提示是可能具有可用空间的分配页。保留的最大提示数为 16。

物理、逻辑和对象锁转换

- 物理锁转换 — 标识与获取物理锁相关的环境切换。
- 逻辑锁转换 — 标识与获取集群逻辑锁相关的环境切换
- 对象锁转换 — 标识与获取对象锁相关的环境切换

对于这三行，`% total` 表示由集群消息传送导致的环境切换的总数。此环境较高的 `% total` 值表示存在大量集群消息活动。将该数值与 `Number of Cluster Lock Requests` 进行比较可确定每个锁请求的集群消息数。有关环境切换的详细信息，请查询 monSysWaits 监控表。

如果锁请求导致太多环境切换，请评估您的应用程序分区。较高的环境切换数表示需要更多集群消息缓冲区。如果 CIPC 消息缓冲区不足，monCIPC 可能具有较高的 ReceiveCount 和 TransmitCount 值。增加 CIPC 消息缓冲区大小可解决此问题。

Log Semaphore Contention

信号是一种简单的内部锁定机制，可防止另一个任务访问当前正在使用的数据结构。 Adaptive Server 使用信号来保护用户日志高速缓存，因为有多个进程可以访问 ULC 的记录并强制刷新。

“Log Semaphore Contention”报告因为需要获取其它任务持有的事务日志信号而断开任务的次数。它仅适用于 SMP 系统。

如果日志信号争用很严重，请参见第 66 页的 “事务管理”。

检查事务日志所使用磁盘上的磁盘队列。请参见第 120 页的 “磁盘 I/O 管理”。

另请参见第 23 页的 “Engine Utilization (Tick %)”。如果引擎利用率报告的值较低，且响应时间在可接受的限制范围内，则可考虑减少引擎数。通过减少试图同时访问日志的任务数，这样可运行较少的引擎以减少争用。

严重的日志信号争用通常表示日志记录或 I/O 子系统存在问题。但是，在具有高吞吐量的 OLTP 环境中，即使经过良好调优的系统也可能具有严重的日志信号争用。

PLC Lock Contention

报告用户日志高速缓存中的锁争用。

Group Commit Sleeps

某个任务执行事务提交并在另一任务将日志写入磁盘之前进入休眠状态的次数。

提交某事务时，其日志记录从它的用户日志高速缓存刷新到高速缓存中事务日志的当前页中。如果该日志页 （或者多个日志页，如果配置了大的日志 I/O 大小）不完整，则会将任务断开并将其置于运行队列的末尾。在以下情况下执行日志页写入：

- 另一进程填充日志页，并刷新日志。这显示为 Group Commit Sleep 或 I/O Pacing。
- 任务到达运行队列的顶部，且其它进程未刷新日志页。这显示为 Last Log Page Write。

在吞吐量较大的环境中，大的日志 I/O 大小有助于防止日志设备上磁盘队列中出现问题。组提交休眠的百分比例较高时不应看作是问题。

Last Log Page Writes

在写入最后一个日志页时任务由于进入休眠状态而断开的次数。

任务之所以会断开，是因为它负责写入最后一个日志页，而不是在等待其它某个任务写入日志页时进入休眠状态 （如第 47 页的 “Group Commit Sleeps”中所述）。

如果该值较高，请检查第 75 页的 “Avg # Writes per Log Page”以确定 Adaptive Server 是否将相同的最后一页重复写入日志中。如果日志 I/O 大小大于 2K,，则减小日志 I/O 大小可减少不必要的日志写入次数。

请参见“Optimizing Transaction Performance in Adaptive Server Enterprise 12.5”（在 Adaptive Server Enterprise 12.5 中优化事务性能）白皮书位于 http://www.sybase.com（在 Sybase 主页上的搜索字段中输入“Optimizing Transaction Performance”）。

Modify Conflicts

在特殊的轻量保护机制下，某任务试图独占访问由另一任务持有的页的次数。对于某些操作， Adaptive Server 采用轻量保护机制，从而可独占访问某页而不使用实际页锁 （例如，访问某些系统表和脏读）。这些进程需要独占访问此页，即使它们不对此页进行修改。

I/O Device Contention

任务在等待特定设备的信号时进入休眠状态的次数。

当某个任务需要执行物理 I/O 时，Adaptive Server 填充 I/O 结构并将其链接到每个引擎的 I/O 队列。如果两个Adaptive Server引擎同时请求同一设备上的 I/O 结构，则其中一个引擎在等待信号时进入休眠状态。

仅当您启用了磁盘镜像 （缺省值为 “disabled”）或者 I/O 被延迟时，才会出现设备争用。有关延迟 I/O 的详细信息，请参见第 120 页的 “磁盘 I/O 管理”。

如果严重争用 I/O 设备信号，请尝试通过重新分布设备间的表或通过增加设备并将表和索引移动到设备上来减少争用。

Network Packet Received

“Network Packet Received”报告任务切换时，任务切换起因于以下原因之一：

- 某任务接收到多包批处理的一部分，并断开以等待客户端发送批处理的下一个包，或者
- 任务彻底处理完当前批处理，并在等待从客户端接收下一个命令或包时被置于接收休眠状态。这是由于 “Network Packet Received”而导致的任务环境切换的最常见原因。

如果 “Network Packet Received”很高，请参见第 126 页的 “网络 I/O 管理”。您可以为所有连接配置网络包大小，或允许某些连接使用较大的包大小登录。

有关网络包大小的详细信息，请参见 《性能和调优系统：基础知识》中的第 1 章 “基础知识简介”。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Network Packet Sent

任务在等待网络将每个包发送到客户端时进入发送休眠状态的次数。网络模型确定在任意一个时间点，每个连接只能有一个未完成的包。任务可以在不向网络 IO 任务发布发送请求的情况下发送网络包，从而避免任务环境切换。

如果要发送很多数据，并且该任务要发送许多小包 （每个包 512 字节），则此任务可能会多次结束休眠。数据包大小是可配置的，而且不同的客户端可请求不同的数据包大小。查询完成后，该操作将包含在 `Network Packet Send` 列中。

有关网络包的详细信息，请参见 《性能和调优系统：基础知识》中的第 2 章 “网络和性能”

如果 “Network Packet Sent”是导致任务切换的主要原因，请参见第 126 页的 “网络 I/O 管理”。

Network Services

测量由于在网络操作 （不是 send 和 receive 操作）上休眠而导致的环境切换次数。

Other Causes

任务由于上述原因以外的任何原因而断开的次数。由 “Network Services” 引起的等待数会减少该值。在经过良好调优的服务器中，该值会随任务切换的可调优资源的减少而增大。

应用程序管理

报告用户任务的执行统计信息。如果使用资源限制，或者通过设置执行属性和指派引擎密切连接来计划对应用程序进行调优，则本部分十分有用。对应用程序、登录或存储过程进行任何调整前，请在典型负载期间运行 sp_sysmon，并熟悉本部分的统计信息。

有关相关背景信息，请参见 《性能和调优系列：基础知识》。

请求详细的应用程序信息

如果用第三个 sp_sysmon 参数 (*applmon*) 请求有关特定任务的信息，sp_sysmon 输出除了给出摘要信息以外，还会分别给出每个应用程序的特定统计信息。您可以选择使用以下两种方式之一显示详细的应用程序信息：

- 应用程序和登录信息 （使用 sp_sysmon 参数 appl_and_login) — sp_sysmon 输出每个登录和所执行的应用程序的单独部分。
- 仅应用程序信息 （使用 sp_sysmon 参数， appl_only）：sp_sysmon 输出每个应用程序的一部分，该部分组合了执行该应用程序的所有登录的数据。

例如，如果有 10 名用户使用 isql 登录，5 名用户使用名为 sales_reports 的应用程序登录，则请求 “应用程序和登录”信息会输出 15 个详细信息部分。请求 “仅应用程序”信息会显示两个详细信息部分，一个概述所有 isql 用户的活动，另一个概述 sales_reports 用户的活动。

appl_and_login 可用于 sp_sysmon 的所有部分。下面是一个语法示例：

```
sp_sysmon “00:05:00”, @applmon=appl_and_login
```

请参见第 6 页的 “指定应用程序详细信息参数”。

样本输出

```
Application Management
----------------------

Application Statistics Summary (All Applications)
-------------------------------------------------
Priority Changes                per sec      per xact      count   % of total
--------------------------  ------------  ----------  --------  ----------
To High Priority                    3.4           0.1      2058       18.6 %
To Medium Priority                  9.6           0.2      5774       52.3 %
To Low Priority                     5.4           0.1      3217       29.1 %
------------------------   ------------  ----------  --------
Total Priority Changes             18.4           0.5     11049

Allotted Slices Exhausted       per sec       per xact      count   % of total
--------------------------  -----------  ----------  -------  ----------
High Priority                       0.0           0.0        0        0.0 %
Medium Priority                    13.8           0.3     8251      100.0 %
Low Priority                        0.0           0.0        0        0.0 %
--------------------------  -----------  ----------  --------  ----------
```

```
Total Slices Exhausted              13.8            0.3        8251

Skipped Tasks By Engine          per sec       per xact      count  % of total
-------------------------  -----------  ----------  --------  ----------
Total Engine Skips                  0.0           0.0         0         n/a

Engine Scope Changes                0.0           0.0         0         n/a
```

以下示例显示应用程序和登录的输出；其中仅包括一个应用程序和登录的信息。第一行标识应用程序名（箭头前面）和登录名（箭头后面）。

```
---------------------------------------------------------------------------
  Application->Login:ctisql->adonis

  Application Activity          per sec      per xact      count   % of total
--------------------------  -----------  ----------  --------  ----------
CPU Busy                            0.1           0.0        27       2.8 %
    I/O Busy                        1.3           0.1       461      47.3 %
    Idle                            1.4           0.2       486      49.9 %

Number of Times Scheduled           1.7           0.2       597         n/a

 Application Priority Changes     per sec      per xact      count   % of total
--------------------------  -----------  ----------  --------  ----------
To High Priority                    0.2           0.0        72      50.0 %
    To Medium Priority              0.2           0.0        72      50.0 %
    To Low Priority                 0.0           0.0         0       0.0 %
--------------------------  -----------  ----------  --------  ----------
Total Priority Changes              0.4           0.0       144

  Application I/Os Completed     per sec      per xact      count   % of total
--------------------------  -----------  ----------  --------  ----------
Disk I/Os Completed                 0.6           0.1       220      53.9 %
    Network I/Os Completed          0.5           0.1       188      46.1 %
--------------------------  -----------  ----------  --------
    Total I/Os Completed            1.1           0.1       408

 Resource Limits Violated        per sec      per xact      count   % of total
--------------------------  -----------  ----------  --------  ----------
IO Limit Violations
    Estimated                       0.0           0.0         0       0.0 %
    Actual                          0.1           4.0         4      50.0 %
  Time Limit Violations
    Batch                           0.0           0.0         0       0.0 %
    Xact                            0.0           0.0         0       0.0 %
  RowCount Limit Violations         0.1           4.0         4      50.0 %
```

```
-------------------------    -----------        -------     --------
Total Limits Violated                 0.1            8.0            8
```

Application Statistics Summary (All Applications)

摘要部分的 sp_sysmon 统计信息可帮助确定资源利用中是否存在异常情况。如果存在，可使用详细的报告进行进一步的研究。

应用程序统计信息显示：

- 任务是否在不同的优先级间来回切换
- 指派给任务允许其运行的时间是否适当
- 已指派低优先级的任务的 CPU 时间是否不足
- 有关负载平衡的引擎绑定是否正确

“Application Statistics Summary”包括系统任务及用户任务的数据。如果摘要报告指出资源问题，而您未在应用程序或应用程序和登录信息中看到支持证据，请研究报告的 sp_sysmon 内核部分 （第 19 页的 “内核使用率”）。

Priority Changes

报告在采样间隔期间每个优先级运行队列中所有用户任务发生的优先级变化。与系统相关的活动会引起的某些优先级切换，看到这种情况是正常的。例如，在以下情况时就会发生这种优先级切换：

- 任务在等待锁时进入休眠状态 — Adaptive Server 临时提升任务的优先级。
- 管家任务休眠 — Adaptive Server 在管家清洗和管家杂事任务唤醒时将优先级提升为中级，并在管家清洗和管家杂事任务重新休眠时将其重新更改为低优先级。
- 任务执行存储过程 — 该任务采用存储过程的优先级，并在执行此过程后恢复其先前的优先级别。

如果使用逻辑进程管理并且与稳定状态值相比，进行了大量优先级更改，则可能表明某个应用程序，或与该应用程序相关的用户任务的优先级更改频繁。请检查各个应用程序的优先级更改数据。确认应用程序和登录是否按照您预期的方式运行。

如果确定高优先级更改率不是由某个应用程序或相关任务引起的，则可能是由系统活动引起的。

Total Priority Changes

报告在采样期间优先级更改的总次数。本部分提供了一种快捷的方法，可用来确定是否发生了大量的运行队列优先级更改。

Allotted Slices Exhausted

每个运行队列中的用户任务超过分配的执行时间的次数。用户任务获得对引擎的访问后，允许该任务在给定的一段时间内执行。如果时间用尽前任务未放弃引擎， Adaptive Server会请求它尽快放弃而不要占有关键资源。放弃后，任务会重新回到运行队列。

“Allocated Slices Exhausted”可帮助您确定是否存在应该对其执行属性或引擎关联进行调优的 CPU 密集型应用程序。如果这些数值高，则表明应用程序是 CPU 密集型的。应用程序级信息可帮助您确定要调优的应用程序。某些任务，尤其是那些执行大量排序操作的任务是 CPU 密集型的。

使用“Alloted slices exhausted”评估新硬件要求。速度更快的 CPU 通常更有优势，您遇到的用尽的时间片越多，升级到速度更快的 CPU 后，您的网站获得的好处就越多。

Skipped Tasks By Engine

引擎在运行队列顶部跳过用户任务的次数。这种情况出现在运行队列顶部的任务与引擎组密切连接，并且非引擎组中的引擎在队列中将其绕过的时候。

此值受配置引擎组和引擎组绑定影响。如果低优先级的任务被绕过而执行更多关键的任务，则该类别的较高值是可接受的。可能存在这种情况：引擎组绑定以至准备运行任务无法找到兼容的引擎。这种情况下，引擎处于空闲状态时，任务可能需要等待执行。请研究引擎组及其绑定方式，并检查负载平衡。

Engine Scope Changes

用户在采样间隔期间更改任何用户任务的引擎组绑定的次数。

Per Application 或 Per Application And Login

本部分提供有关由特定应用程序和登录任务，或每个应用程序的所有用户使用的系统资源的详细信息。

Application Activity

帮助您确定应用程序是 I/O 密集型还是 CPU 密集型。它报告在应用程序中所有用户任务在执行、进行 I/O 或处于空闲状态时所花费的时间。它还报告安排和选择运行任务的次数。

CPU 繁忙

采样间隔期间执行用户任务的时钟周期数。如果该类别中的数值较高，则表明是 CPU 绑定的应用程序。如果这会导致出现问题，您可能需要考虑引擎绑定。

I/O Busy

采样间隔期间用户任务执行 I/O 的时钟周期数。如果该类别中的数值高，则表明是 I/O 密集型的进程。如果空闲时间值也高，则此应用程序可能是 I/O 绑定。

如果为应用程序指派较高的优先级、将其绑定到轻载引擎或引擎组，或将应用程序的数据分隔到多个设备上，则可能会实现较好的吞吐量。

Idle

采样间隔期间用户任务空闲的时钟周期数。

Number of Times Scheduled

安排和选择用户任务在引擎上运行的次数。该数据可帮助确定应用程序是否有充足的资源。对于通常需要占用大量 CPU 时间的任务，如果该值低，则可能表明资源不充足。请考虑在带有足够的引擎资源的已装载系统中更改优先级。

Application Priority Changes

该应用程序在采样期间更改其优先级的次数。

当“Application Management”类别指明某个问题时，可使用这一部分来找出问题的起因。

Application I/Os Completed

报告在采样间隔期间该应用程序完成的磁盘和网络 I/O 数。

该类别表明完成的磁盘和网络 I/O 总数。

如果怀疑 I/O 的完成情况存在问题，请参见第 120 页的“磁盘 I/O 管理”和第 126 页的 “网络 I/O 管理”。

Resource Limits Violated

针对以下各项的违反次数和类型：

- 违反 I/O 限制 — 估计的和实际的
- 时间限制 — 批处理和事务
- 违反 RowCount 限制
- Total Limits Violated

如果在采样期间未超过任何限制，则仅输出总计行。

ESP 管理

本部分报告扩展存储过程的使用情况。

样本输出

```
===============================================================================

ESP Management                  per sec      per xact       count   % of total
--------------------------  ------------  ------------  ----------  ----------
ESP Requests                        0.0           0.0           0         n/a
```

ESP Requests

采样间隔期间扩展存储过程调用的次数。

Avg. Time to Execute an ESP

报告在采样间隔期间执行所有扩展存储过程的平均时间。

Housekeeper Task Activity

报告管家任务。如果将配置参数 housekeeper free write percent 设置为 0，则管家任务 HK WASH 不运行。将 enable housekeeper GC 设置为 1 到 5 之间的值可启用空间回收。将 enable housekeeper GC 设置为 0 可禁用空间回收。

有关详细信息，请参见：

- 《性能和调优：基础知识》中的第 3 章 “使用引擎和 CPU”
- 《系统管理指南，卷 1》中的 “诊断系统问题”

样本输出

```
===============================================================================

Housekeeper Task Activity
-------------------------
                         per sec    per xact     count        % of total
                         ---------  --------  --------        -----------
Buffer Cache Washes
   Clean                     228.0       5.6    136828            100.0 %
   Dirty                       0.0       0.0        12              0.0 %
                         ---------  --------  --------
Total Washes                 228.1       5.6    136840
Garbage Collections            1.3       0.0       791               n/a
Pages Processed in GC          0.0       0.0         0               n/a
Statistics Updates            12.0       0.3      7196               n/a
```

Buffer Cache Washes

本部分报告：

- 由管家清洗任务检查的缓冲区数
- 查找到的干净缓冲区数
- 查找到的脏缓冲区数

当缓冲区代表的数据高速缓存中的数据页包含 Adaptive Server 尚未写入磁盘的更改时，该缓冲区被视为 “脏”缓冲区。当缓冲区代表的数据页是磁盘上的数据副本时，该缓冲区被视为 “干净”缓冲区。

脏缓冲区数包括因在清洗标记处开始写入，而已经位于 I/O 中的那些缓冲区。

sp_sysmon 的“Recovery Management”部分报告管家清洗任务能够为某个数据库的所有脏缓冲区进行写入的次数。请参见第 117 页的 “恢复管理”。

Garbage Collections

管家碎片收集任务进行检查以确定是否有提交的删除操作 （表示数据页上有可以回收的空间）的次数。

Pages Processed in GC

管家碎片收集任务在其中成功回收 DOL 锁定表页上未使用空间的页数。

Statistics Updates

管家杂事任务查看是否需要写入统计信息的次数。

对执行 SQL 的监控访问

报告：

- sp_showplan 访问查询计划时出现的争用。

  注释　查询监控表 （例如 monProcessSQLText 和 monSysSQLText）不会更新这些计数器

- 在采样间隔期间，SQL 批处理文本缓冲区中的溢出数，以及发送的 SQL 批处理文本的最大大小

样本输出

```
Monitor Access to Executing SQL
-------------------------------
                                  per sec    per xact      count    % of total
                                -----------  ----------   -------    ----------
Waits on Execution Plans              0.0         0.0         0         n/a
Number of SQL Text Overflows          8.6          .2      5138         n/a
Maximum SQL Text Requested            n/a         n/a    588307         n/a
(since beginning of sample)
```

Waits on Execution Plans

某个试图使用 sp_showplan 的进程为获得查询计划的读取权限而必须等待的次数。完成编译计划前或查询计划执行完成后，如果运行 sp_showplan，则查询计划可能不可用。这些情况下， Adaptive Server 会尝试访问此计划三次，然后向用户返回消息。

Number of SQL Text Overflows

SQL 批处理文本超过文本缓冲区大小的次数。

Maximum SQL Text Requested

自采样间隔开始起 SQL 批处理文本的最大大小。您可以使用该值设置配置参数 max SQL text monitored。

请参见 《系统管理指南，卷 1》中的 “诊断系统问题”。

事务配置文件

按命令类型和表锁定方案报告数据修改。

样本输出

```
===============================================================================
Transaction Profile
-------------------

  Transaction Summary           per sec    per xact     count        % of total
------------------------      ---------  ---------  ---------  ----------
    Committed Xacts               40.6        n/a      24357          n/a

  Transaction Detail            per sec    per xact     count        % of total
  ------------------------      ---------  ---------  ---------  ----------
  Inserts
    Fully Logged
      APL Heap Table               0.0        0.0          0          0.0 %
      APL Clustered Table          0.0        0.0          0          0.0 %
      Data Only Lock Table      1337.3      906.7     160479        100.0 %
      Fast Bulk Insert             0.0        0.0          0          0.0 %
    Minimally Logged
      APL Heap Table               0.0        0.0          0          0.0 %
      APL Clustered Table          0.0        0.0          0          0.0 %
      Data Only Lock Table         0.0        0.0          0          0.0 %
  ----------------------  -----------  ---------  ---------  ----------
  Total Rows Inserted           1337.3      906.7     160479        100.0 %

  Updates
    Total Rows Updated             0.0        0.0          0          n/a
  ----------------------  -----------  ---------  ---------  ----------
  Total Rows Updated               0.0        0.0          0          0.0 %

  Data Only Locked Updates
    Total Rows Updated             0.0        0.0          0          n/a
  ----------------------  -----------  ---------  ---------  ----------
  Total DOL Rows Updated           0.0        0.0          0          0.0 %

  Deletes

    Fully Logged
      APL Deferred                 0.0        0.0          0          0.0 %
      APL Direct                   0.0        0.0          0          0.0 %
```

```
    DOL                          0.3          0.2         36           100.0 %
  Minimally Logged
    APL Direct                   0.0          0.0          0             0.0 %
    DOL                          0.0          0.0          0             0.0 %
----------------------  ------------  ----------  -------  ----------
Total Rows Deleted               0.3          0.2         36             0.0 %

======================  ============  ==========  ========
Total Rows Affected           1337.6        906.9     160515
======================  ============  ==========  ========
```

Transaction Summary

报告已提交的事务。“Committed Xacts”报告采样间隔期间提交的事务数。

计入的事务包括隐式和显式事务、链式 （ANSI 样式）或非链式事务：

- 隐式事务执行数据修改命令，如 insert、update 或 delete。如果不指定 begin transaction 语句，Adaptive Server将每个操作作为单独的事务进行解释；无需显式 commit transaction 语句。例如，以下操作计为三个事务：

  ```
  1> insert…
  2> go
  1> insert…
  2> go
  1> insert…
  2> go
  ```

- 显式事务将数据修改命令包括在 begin transaction 和 commit transaction 语句中，并按提交的语句数计算事务数。例如，以下语句组计为一个事务：

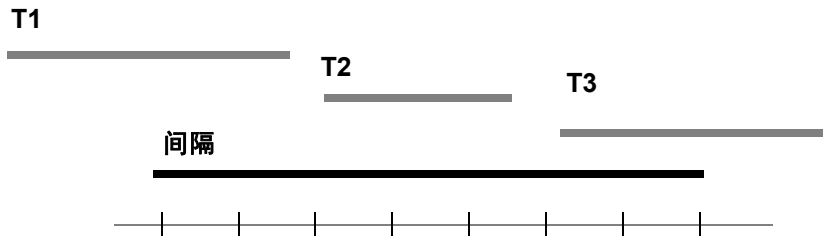
  ```
  1> begin transaction
  2> insert…
  3> insert…
  4> insert…
  5> commit transaction
  6> go
  ```

- 在 ANSI 事务模式中，任何 select 或数据修改命令均启动一个事务，但 commit transaction 语句必须完成此事务。 sp_sysmon 按 commit transaction 语句数来对事务进行计数。例如，以下语句组计为一个事务：

```
1> insert…
2> insert…
3> insert…
4> commit transaction
5> go
```

如果事务在采样间隔开始前开始，并在间隔过程中完成，则 “Committed Xacts” 报告的事务数大于在采样间隔期间开始和完成的事务数。如果事务在间隔期间未完成，则 “Total # of Xacts” 不包括它们。在图 2-1 中，对 T1 和 T2 都进行了计数，但未对 T3 计数。

图 2-1：事务的计数方法



多数据库事务的计数方法

多数据库事务也进行计数。例如，修改三个数据库的事务被计为三个事务。

多数据库事务比单数据库事务的开销大：它们需要更多日志记录和更多次用户日志高速缓存 (ULC) 刷新，并且它们涉及在数据库之间进行包括两个阶段的提交。

在可能的情况下，可通过减少多数据库事务的数量来提高性能。

Transaction Detail

提供与按类型进行的数据修改操作相关的详细统计信息。由回退事务执行的工作包括在下面的输出中，虽然此事务不计在事务数量中。

在插入、更新和删除的 “Total Rows”中，“% of total”列报告该事务类型占所有事务的百分比。

有关延迟的和直接的插入、更新和删除的详细信息，请参见 《性能和调优系列：查询处理和抽象计划》中的 “如何执行更新操作”。

在本部分的输出中， APL 表示锁定所有页的表， DOL 表示只锁定数据的表。

Inserts

提供与以下方面相关的详细信息：在堆表 （包括分区的堆表）和聚簇表上进行的插入操作的类型，以及所有插入操作占所有 insert、 update 和 delete 操作的百分比。“Inserts”显示对以下项执行插入操作的次数：

- 所有页锁定堆表
- 有聚簇索引的所有页锁定表
- DOL 锁定表

插入统计信息不包括快速批量复制插入数，因为它们被直接写入数据页和磁盘而无需使用常规插入和记录机制。

完整日志记录和最少日志记录

报告 “完整日志记录”（对于完整日志记录操作）和 “最少日志记录”（对于最少日志记录操作）。有时，项目在 “完整日志记录”子部分中但不在 “最少日志记录”中，因为项目不适合最少日志记录。

APL Heap Tables

在所有页锁定堆表 （不带聚簇索引的所有表）上进行的行插入操作的次数。这包括：

- 分区的堆表
- 未分区的堆表
- 插入到堆表中的慢速批量复制

- select into 命令
- 插入到工作表。除非应用程序大量使用不带聚簇索引的 APL 表，否则这是最常见的情形。

“% of total”列显示在堆表中进行的行插入次数占插入总数的百分比。

如果对堆表进行了大量的插入操作，则确定这些插入是否会产生争用。

检查 sp_sysmon 报告，以查看第 92 页的 “Lock Detail”中的堆上最后一页的锁上的数据。如果看起来存在争用问题，您可以查询监控表以确定所涉及的表。

许多情况下，创建使插入活动随机化的聚簇索引，可解决堆的性能问题。在其它情况下，可能需要在未分区的表上建立分区或增加分区表上的分区数。

APL Clustered Table

向带聚簇索引的所有页锁定表执行的行插入的次数。“% of total”列显示在带有聚簇索引的表中插入的行数占插入行总数的百分数。

对所有页锁定聚簇表的插入操作可导致分页。

Adaptive Server 可以使用带聚簇索引的工作表来执行矢量集合 （例如 group by），不过，对于 Adaptive Server 15.0 和更高版本，基于散列的算法更常见。

请参见第 78 页的 “RID Upd from Clust Split”和第 79 页的 “Page Splits”。

Data Only Lock Table

所有 DOL 锁定表的插入次数。“% of total”列显示对 DOL 锁定表进行的插入数占所有插入数的百分比。

Fast Bulk Insert

部分日志记录的快速 bcp 执行的插入次数。

Total Rows Inserted

报告对所有表执行的所有行插入。它给出每秒所有插入数的平均值、每个事务所有插入操作的平均次数和插入操作的总数。“% of total”显示插入的行数占数据修改操作影响到的总行数的百分比。

“Updates”和更新细节部分

“Updates”有两部分组成：“Updates”和 “Data Only Locked Updates”。

包括 “完整日志记录”（对于完整日志记录操作）和 “最少日志记录”（对于最少日志记录操作）报告。有时，项目在 “完整日志记录”子部分中但不在 “最少日志记录”中，因为项目不适合最少日志记录。

Updates

延迟和直接的行更新次数。“% of total”列报告每种类型的更新数占行更新总数的百分比。 sp_sysmon 报告以下类型的更新：

- APL 延迟
- APL 直接现场
- APL 直接廉价
- APL 直接昂贵
- DOL 延迟
- DOL 直接

“完整日志记录”部分的 “APL 直接廉价”表示完整日志记录的所有页锁定直接更新的数量。“最少日志记录”部分的 “APL 直接廉价”表示发生的最小日志记录的所有页锁定直接更新的数量。

和延迟更新相比，直接更新能限制日志扫描的数量、减少锁定、减少 B 树索引的浏览 （减少锁争用），所以其所需开销更少，而且通常速度更快。另外，还因为 Adaptive Server 不必预先读取页以执行基于日志记录的修改，所以也能节省 I/O。

有关更新类型的说明，请参见 《性能和调优系列：查询处理和抽象计划》中的第 1 章 “了解查询处理”。

Total Rows Updated

报告组合的所有延迟的和直接的更新。“% of total”列显示更新的行占所有已修改行的百分比。

DOL 锁定表更新

报告有关 DOL 锁定表更新的更多详细信息：

- DOL 替换 — 更新未改变行宽；行的一部分或全部进行了更改，但仍保持相同行宽。
- DOL 收缩 — 更新使行缩短，从而使页上非连续的空闲空间在空间回收时被收集起来。
- DOL 廉价扩展 — 行宽增加；由于它是页上的最后一行，因此扩展该行的宽度不需要移动页上的其它行。
- DOL 昂贵扩展 — 行宽增加并要求移动页上的其它行。
- DOL 扩展和转移 — 行宽增加，已经超出此页范围。该行被转移到一个新位置。
- DOL 转移行返回 — 该更新影响转移的行；此行未超出原页范围并被返回到该页。

包括 “完整日志记录”（对于完整日志记录操作）和 “最少日志记录”（对于最少日志记录操作）报告。有时，项目在 “完整日志记录”子部分中但不在 “最少日志记录”中，因为项目不适合最少日志记录。

在 “Total DOL Rows Updated”中报告的总数并不包括在本部分最后的 “Total Rows Affected”总和中，因为该组中的更新提供与已在 “DOL Deferred”和 “DOL Direct”中报告的更新不同的细目分类。

Deletes

完整日志记录和最少日志记录

报告 “完整日志记录”（对于完整日志记录操作）和 “最少日志记录”（对于最少日志记录操作）。有时，项目在 “完整日志记录”子部分中但不在 “最少日志记录”中，因为项目不适合最少日志记录。

Total Rows Deleted

报告组合的所有延迟的和直接的删除。“% of total”列报告删除的行占所有已插入、更新或删除的行的百分比。

事务管理

报告事务管理活动，包括用户日志高速缓存 (ULC) 刷新到事务日志、ULC 日志记录、 ULC 信号请求、日志信号请求、事务日志写入和事务日志分配。

样本输出

```
Transaction Management
----------------------

    ULC Flushes to Xact Log     per sec      per xact      count  % of total
    ----------------------  ------------  ------------  ----------  ----------
    Any Logging Mode DMLs
      by End Transaction             1.6           1.1         189       1.0 %
      by Change of Database          0.0           0.0           4       0.0 %
      by Unpin                       0.0           0.0           2       0.0 %
      by Log markers                 0.0           0.0           0       0.0 %

    Fully Logged DMLs
      by Full ULC                  145.5          98.6       17458      95.9 %
      by Single Log Record           4.5           3.1         544       3.0 %

    Minimally Logged DMLs
      by Full ULC                    0.0           0.0           0       0.0 %
      by Single Log Record           0.0           0.0           0       0.0 %
      by Start of Sub-Command        0.0           0.0           0       0.0 %
      by End of Sub-Command          0.0           0.0           0       0.0 %
    ----------------------  ------------  ------------  ----------
    Total ULC Flushes            151.6         102.8       18197

    ULC Flushes Skipped          per sec      per xact       count   % of total
    ----------------------  ------------  ------------  ----------  ----------
    Fully Logged DMLs
      by ULC Discards                0.2           0.1          21        100 %
    Minimally Logged DMLs
      by ULC Discards                0.0           0.0           0       0.0 %
    ----------------------  ------------  ------------  ----------
    Total ULC Flushes Skips          0.2           0.1          21

    ULC Log Records             per sec       per xact       count   % of total
    ----------------------  ------------  ------------  ----------  ----------
    Fully Logged DMLs            6662.3        4516.8      799476       100.0
    Minimally Logged DMLs           0.0           0.0           0         0.0
```

```
    ---------------------  ------------  ------------  ----------
    Total ULC Log Records        6662.3        4516.8      799476

    Max ULC Size During Sample
    --------------------------
    Fully Logged DMLs               n/a           n/a       16384         n/a
    Minimally Logged DMLs           n/a           n/a           0         n/a

    ML-DMLs Sub-Command Scans   per sec      per xact       count  % of total
    ---------------------  ------------  ------------  ----------  ----------
---
    Total Sub-Command Scans         0.0           0.0           0         n/a

    ML-DMLs ULC Efficiency      per sec      per xact       count  % of total
    ---------------------  ------------  ------------  ----------  ----------
    Total ML-DML Sub-Commands       0.0           0.0           0         n/a

    ULC Semaphore Requests
      Granted                   13727.0        9306.4     1647239     100.0 %
      Waited                        0.0           0.0           0       0.0 %
    ---------------------  ------------  ------------  ----------
    Total ULC Semaphore Req     13727.0        9306.4     1647239

    Log Semaphore Requests
      Granted                     184.0         124.7       22076     100.0 %
      Local Waited                  0.0           0.0           0       0.0 %
      Global Waited                 0.0           0.0           0       0.0 %
    ---------------------  ------------  ------------  ----------
    Total Log Semaphore Req       184.0         124.7       22076

    Transaction Log Writes        167.5         113.6       20102         n/a
    Transaction Log Alloc         167.4         113.5       20092         n/a
    Avg # Writes per Log Page       n/a           n/a     1.00050         n/a

    Tuning Recommendations for Transaction Management
    -------------------------------------------------
    - Consider increasing the 'user log cache size'
      configuration parameter.
```

ULC Flushes to Xact Log

报告用户日志高速缓存 (ULC) 刷新到事务日志的总次数。“% of total”列报告每个类别的某种刷新的发生次数占 ULC 刷新总次数的百分比。此类别可帮助您识别引起 ULC 刷新问题的应用程序中的区域。

每个已配置用户连接都有一个 ULC。Adaptive Server 使用 ULC 缓冲事务日志记录。在 SMP 和单处理器系统上，这有助于减少事务日志 I/O。对于 SMP 系统，它可以减少对当前事务日志页的争用。

可以使用配置参数 user log cache size 配置 ULC 的大小。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章“设置配置参数”。

ULC 刷新由下列活动引起：

- “by Full ULC”— 一个进程的 ULC 已满。
- “by End Transaction”— 事务结束（rollback 或 commit，隐式或显式）。
- “by Change of Database”— 事务修改了不同数据库中的对象（多数据库事务）。
- “by System Log Record”— 在用户事务内发生的系统事务（如 OAM 页分配）。
- “by Unpin”— 两个事务尝试更新 DOL 表的同一页。
- “by Start of Sub-Command”或 “by End of Sub-Command”— ULC 需要在最少日志记录 DML 的子命令开始或结束时刷新。
- “by Other”— 任何其它原因，包括需要写入到磁盘。

当这些活动之一引起 ULC 刷新时，Adaptive Server 从用户日志高速缓存将所有日志记录复制到数据库事务日志。

“Total ULC Flushes”报告在采样间隔过程中发生的所有 ULC 刷新的总数。

Any Logging Mode DMLs

by End Transaction

如果 “by End Transaction”的值较高，则表示大量简短事务。

by Change of Database

每次数据库修改都会刷新 ULC。如果该值很高，且大于 2K.，则应考虑减小 ULC 的大小。

by Unpin

当两个事务试图更新同一个数据页，从而导致第二个事务刷新第一个事务的用户日志高速缓存时，将发生 “解锁”，以将该数据页从第一个事务的 ULC 中解锁。

因为两个事务都尝试更新同一页，为了提高并发性， Adaptive Server 通常使用行级锁定。

以下情况下会发生解锁：

1　事务 T1 更新行 R1。

2　Adaptive Server 将该更新记录到 P1 的 ULC 中，以确保将该数据页锁定到 T1 的 ULC。

3　当 T2 尝试更新 P1 页上的不同行时，它发现该页已锁定到 T1 的 ULC。

4　为了更新该页， T2 将刷新 T1 的 ULC，以解锁该数据页。

若要减少解锁次数，请执行以下操作：

- 如果可能，重新设计应用程序以便它直接请求不同的数据页。
- 如果行锁定是不可避免的，请使用 sp_chgattribute max_rows_per_page 将数据分布到更多数据页。

by Log markers

较高的 “by Log markers”值表示由于存在永久性日志标记扫描 （永久性日志标记扫描表示 syslogs 包括日志记录） Adaptive Server 正在刷新 ULC。 Adaptive Server 使用日志记录执行如下操作：触发、回退、中止等等。当 Adaptive Server 需要永久性日志标记但却没有该标记时，它会刷新 ULC 以创建新的日志标记。

当 “by Log markers”显示比例较大的 ULC 刷新时，您可能需要减少不必要或多余的触发、回退或中止操作的数量。

by System Log Record and By Other

如果这两个值中有一个略大于 20%，且您的 ULC 的大小超过 2048，则应考虑减小 ULC 的大小。

检查与日志活动相关的 sp_sysmon 报告的各部分：

- 用户日志高速缓存中的信号争用 （仅限于 SMP）；请参见第 74 页的 “ULC Semaphore Requests”
- 日志信号的争用。（仅限于 SMP）；请参见第 75 页的 “Log Semaphore Requests”
- 事务日志写入的次数；请参见第 75 页的 “Transaction Log Writes”

Fully Logged DMLs

by Full ULC

如果 “by full ULC”值较高，则表示对于每个事务， Adaptive Server 多次刷新 ULC，这样将抵消用户日志高速缓存的某些性能优点。如果 “by Gull ULC”的 “% of total”值大于 20%，请考虑增加 user log cache size 参数的大小。

增加 ULC 大小可为每个用户连接增加内存量，这样您就不用配置 ULC 的大小以适应大事务的小百分比。

by Single Log Record

单个日志记录无法在 ULC 中生存，因此它肯定会导致 ULC 刷新。

最少日志记录 DML

如果 “Minimally Logged DMLs”的任何 “ULC Flushes to Xact Log”值很高，则 ULC 不能充分地包含构成最少日志记录 DML 的每个子命令的日志记录。这种情况下，因为不能放弃日志记录，所以 Adaptive Server 性能不能充分地享受最少日志记录带来的好处。请增加 user log cache size 配置参数的大小，然后重新运行 sp_sysmon 以查看 “ULC Flushes to Xact Log”的值是否已减小。如果该值已减小，则表明 ULC 不再需要刷新到 syslogs。

by Full ULC

表示最少日志记录 DML 的单个子命令记录的日志记录超出 ULC 可以容纳的数量，必须刷新到 syslogs 中。仅当单个子命令的所有日志记录可以记录在单个 ULC 中时，最少日志记录才会生效。当子命令完成并且其所有日志记录都填入 ULC 时，日志记录将被放弃而不会刷新到 syslogs 中。

by Single Log Record

单个日志记录无法在 ULC 中生存，因此它肯定会导致 ULC 刷新。

by Start of Sub-Command

来自与最少日志记录 DML 在同一事务中执行的完整日志记录命令的日志记录必须从 ULC 刷新到 syslogs，然后最少日志记录 DML 才能启动。

by End of Sub-Command

因为之前的完整 ULC、单个日志记录等原因，必须记录最少日志记录 DML 的单个子命令。 ULC 需要在子命令结束时进行刷新以保留该子命令的所有日志记录。

Total ULC Flushes

报告在采样间隔期间 ULC 刷新的总次数。

ULC Flushes Skipped

Adaptive Server 跳过的用户日志高速缓存刷新次数。

Fully Logged DMLs

如果在您执行完整日志记录 DML 时，事务的所有日志记录都填入 ULC 中，则 Adaptive Server 可以放弃所有日志记录。

“Fully Logged DMLs”值越高表示性能越好。

最少日志记录 DML

由于最少日志记录 DML 的子命令的所有日志记录都填入 ULC 而可以放弃 ULC 中的日志记录的次数。

“Minimally Logged DMLs”值越高表示性能越好。

Total ULC Flushes Skips

报告所有组合的用户日志高速缓存刷新。总数是按列显示的。

ULC Log Records

此行提供每个事务的日志记录平均数。在基准测试或受控开发环境中确定每个事务写入到 ULC 的日志记录数是很有用的。

许多事务 （例如，影响几个索引或延迟的更新或删除的那些事务）在修改单个数据时需要进行几个日志记录。修改大批行的查询对于每行都使用一个或多个记录。

如果此数据有异常，则应研究下一节 （Max ULC Size During Sample）中的数据，并查看应用程序中长期运行的事务和修改大量行的事务。

Fully Logged DMLs

表示完整日志记录 DML 的每个事务的日志记录数量。

最少日志记录 DML

表示最少日志记录 DML 的每个事务的日志记录数量。

Total ULC Log Records

报告所有组合的用户日志高速缓存日志记录。总数是按列显示的。

Max ULC Size During Sample

“count”列的值是在任何 ULC（遍及所有 ULC）中使用的最大字节数。此数据可帮助您确定 ULC 的大小是否已正确配置。

由于 Adaptive Server 在事务完成时刷新 ULC，因此分配给 ULC 的任何未使用的内存将会浪费掉。如果“count”列中的值始终小于为 user log cache size 配置参数定义的值，则可将 user log cache size 减小到“count”列中的值（但不小于 2048 字节）。

当“Max ULC Size During Sample”等于用户日志高速缓存的大小时，请检查由于填充用户日志高速缓存的事务引起的刷新次数（请参见第 70 页的“by Full ULC”）。如果由于 ULC 已满引起的日志刷新次数超过 20%，请考虑增加 user log cache size 配置参数。

有关配置参数的详细信息，请参见《系统管理指南，卷 1》中的第 5 章“设置配置参数”。

Fully Logged DMLs

用于完整日志记录 DML 的任何用户日志高速缓存的最大字节数。

最少日志记录 DML

用于最少日志记录 DML 的任何用户日志高速缓存的最大字节数。

ML-DMLs Sub-Command Scans

在执行某些最少日志记录命令时， Adaptive Server 必须扫描子命令的日志记录。“ML-DMLs Sub-Command Scans”的值是指 Adaptive Server 执行的扫描次数。

ULC 扫描

该扫描可以完全在 ULC 内执行，因为它包含要扫描的所有日志记录。

syslogs 扫描

必须使用 syslogs 执行该扫描，因为 ULC 不再包含要扫描的日志记录。

Total Sub-Command Scans

ULC 和 syslogs 扫描的总和。

ML-DMLs ULC Efficiency

Discarded Sub-Commands

Adaptive Server 可以放弃最少日志记录 DML 的子命令的日志记录（即，日志记录没有从 ULC 刷新到 syslogs）的次数。

“Discarded Sub-Commands”数值越高，Adaptive Server 执行最少日志记录 DML 的效率就越高，因为已经最大限度地减少日志记录数量。

Logged Sub-Commands

Adaptive Server 无法放弃最少日志记录 DML 的子命令的日志记录（即，ULC 中的日志记录需要刷新到 syslogs）的次数。

Total ML-DML Sub-Commands

“Discarded Sub-Commands”和 “Logged Sub-Commands”的总和。

ULC Semaphore Requests

用户任务被立即授予信号的次数，或必须等待信号的次数。“% of total”显示授予信号的任务和等待信号的任务占 ULC 信号请求总数的百分比。这仅适用于 SMP 环境。

“ULC Semaphore Requests”提供以下信息：

- Granted — 一经请求立即对任务授予 ULC 信号的次数。不存在 ULC 争用。
- Local Waited — 任务尝试写入到 ULC 而遇到信号争用的次数。
- Total ULC Semaphore Requests — 在间隔期间发生的 ULC 信号请求的总数，包括已授予或必须等待的请求。

Log Semaphore Requests

报告在高速缓存中保护事务日志当前页的日志信号争用情况。此数据只对 SMP 环境有意义。

本类别提供下列信息：

- Granted — 提出请求后立即对任务授予日志信号的次数。“% of total”报告立即授权的请求占日志信号请求总次数的百分比。
- Local Waited — 由于日志信号已由另一任务获取，尝试刷新其 ULC 的任务必须等待日志信号的次数。“% of total”报告被迫等待日志信号的任务占日志信号请求总数的百分比。
- Global Waited — 由于另一任务已在该节点或其它节点上获取日志信号，共享磁盘集群中尝试刷新其 ULC 的任务必须等待日志信号的次数。
- Total Log Semaphore Requests — 任务请求日志信号的总次数，包括立即授予的请求和任务被迫等待的请求。

Transaction Log Writes

报告 Adaptive Server 将事务日志页写入磁盘的总次数。在事务提交（等待组提交休眠后）或当前日志页已满时，将事务日志页写入磁盘。

Transaction Log Allocations

将其它页分配给事务日志的次数。此数据对于比较此节中的其它数据和跟踪事务日志增长率很有用。

Avg # Writes per Log Page

报告将每个日志页写入磁盘的平均次数。“count”列中报告了该值。

在高吞吐量应用程序中，此数值应尽可能低。如果事务日志使用 2K 的 I/O，则可能的最低值是 1；如果使用 4K 的日志 I/O，则可能的最低值是 0.5，因为一个日志 I/O 可写 2 个日志页。

在低吞吐量的应用程序中，此数值将极高。在非常低的吞吐量环境中，它可能与每个完成的事务的一次写入一样高。

Tuning Recommendations for Transaction Management

sp_sysmon 介绍它基于 “Transaction Management”部分的结果所推荐的任何调优。

索引管理

报告索引管理活动，包括非聚簇维护、页面拆分和索引收缩。

样本输出

```
Index Management
----------------

  Nonclustered Maintenance      per sec      per xact      count   % of total
  ------------------------  ------------  ------------  ----------  ----------
  Ins/Upd Requiring Maint        1996.3          25.3     1778674         n/a
    # of NC Ndx Maint              694.9           8.8      619122         n/a
    Avg NC Ndx Maint / Op            n/a           n/a     0.34808         n/a

  Deletes Requiring Maint        1922.5          24.3     1712946         n/a
    # of NC Ndx Maint              621.1           7.9      553387         n/a
Avg NC Ndx Maint / Op                n/a           n/a     0.32306         n/a

  RID Upd from Clust Split           0.0           0.0           0         n/a
    # of NC Ndx Maint                0.0           0.0           0         n/a

  Upd/Del DOL Req Maint              9.0           0.1        7989         n/a
    # of DOL Ndx Maint               9.2           0.1        8238         n/a
    Avg DOL Ndx Maint / Op           n/a           n/a     1.03117         n/a

  Page Splits                       11.7           0.1       10452         n/a
    Retries                          0.0           0.0           0       0.0 %
    Deadlocks                        0.0           0.0           0       0.0 %
    Add Index Level                  0.0           0.0           0       0.0 %

Page Shrinks                         0.1           0.0         124         n/a
Deadlocks                              %
Deadlock Retries Exceeded            0.0           0.0           0       0.0 %

  Index Scans                   per sec      per xact       count   % of total
```

```
----------------------  ------------  ------------  --------  -----------
  Ascending Scans            69751.6         883.1  62148709       29.3 %
  DOL Ascending Scans       168653.5        2135.3 150270303       70.7 %
  Descending Scans               8.8           0.1      7884        0.0 %
  DOL Descending Scans          23.9           0.3     21271        0.0 %
                           ----------    ----------  ---------
Total Scans                238437.9        3018.8  212448167
```

Nonclustered Maintenance

本类别报告对一个或多个索引需要的或可能需要的维护操作次数；也就是说，它报告 Adaptive Server 必须检查以确定是否有必要更新索引的操作数。输出也给出已更新的索引数和每次操作维护的平均索引数。

在带聚簇索引和一个或多个非聚簇索引的表中，所有插入、所有删除、某些更新操作和任何数据页面拆分都需要对非聚簇索引进行更改。高索引维护值表明您应该评估维护索引对 Adaptive Server 性能的影响。在索引加速数据检索的同时，维护索引也会减慢数据修改的速度。维护时需要索引页的其它进程、其它 I/O 和其它锁定。

与评估非聚簇索引相关的其它 sp_sysmon 输出有：

- 有关更新、插入和删除的总次数的信息，以及页面拆分的数量和类型的信息。请参见第 62 页的 “Transaction Detail”和第 79 页的 “Page Splits”。
- 有关锁争用的信息。请参见第 92 页的 “Lock Detail”。
- 有关地址锁争用的信息。请参见第 45 页的 “Address Lock Contention”和第 93 页的 “Exclusive Address 和 Shared Address”。

例如，您可将发生的插入次数与导致的维护操作次数进行比较。如果发生的维护操作、页面拆分和重试的次数较高，则应考虑应用程序中索引是否有用。

有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 5 章 “Indexes”（索引）。

Ins/Upd Requiring Maint

本部分中的数据提供了有关插入和更新操作如何影响所有页锁定表中的索引的信息。例如，在拥有三个非聚簇索引的表中，一个插入要求更新所有三个索引，因此导致对非聚簇索引进行维护的平均操作次数是三次。

但是，对同一表进行更新可能只要求进行一次维护操作 — 对其键值已经更改的索引进行。

- Ins/Upd Requiring Maint — 报告对具有索引的表进行插入和更新操作的次数，这些操作可能会要求对一个或多个索引进行修改。
- # of NC Ndx Maint — 报告因插入和更新操作而需要维护的非聚簇索引的数目。
- Avg NC Ndx Maint/Op — 报告要求维护的每个插入或更新操作的非聚簇索引的平均数。

对于 DOL 锁定表，在 “Ins/Upd Requiring Maint”中报告插入数，在 “Upd/Del DOL Req Maint”中报告删除数和插入数。

Deletes Requiring Maintenance

本部分中的数据提供了有关删除操作如何影响所有页锁定表的索引的信息：

- Deletes Requiring Maint — 报告可能要求对一个或多个索引进行修改的删除操作数。
- # of NC Ndx Maint — 报告因删除操作而要求维护的非聚簇索引数。
- Avg NC Ndx Maint/Op — 报告每个删除操作的要求维护的非聚簇索引的平均数。

请参见第 65 页的 “Deletes”。

RID Upd from Clust Split

报告在带聚簇索引的所有页锁定表中，由页面拆分引起的索引维护活动。这些拆分要求为移动到新数据页的所有行更新非聚簇索引。

- RID Upd from Clust Split — 报告需要维护非聚簇索引的页面拆分总数。
- # of NC Ndx Maint — 报告因行 ID 更新操作而要求维护的非聚簇行数。
- Avg NC Ndx Maint/Op — 报告为每个页面拆分更新的非聚簇索引条目的平均数。

Upd/Del DOL Req Maint

本部分中的数据提供有关更新和删除操作如何影响 DOL 锁定表的索引的信息：

- “Upd/Del DOL Req Maint”报告可能要求对一个或多个索引进行修改的更新和删除操作的数目。
- “# of DOL Ndx Main”报告因更新或删除操作而要求维护的索引数。
- “Avg DOL Ndx Maint/Op”报告每个更新或删除操作的要求维护的索引平均数。

Page Splits

数据页、聚簇索引页或非聚簇索引页因为空间不足容纳不下新行而进行的页面拆分数。

当一个数据行插入到拥有聚簇索引的所有页锁定表中时，此行必须根据键值按物理顺序放置。索引行也必须按物理顺序放置在此页中。如果在一页中没有足够的空间容纳一个新行，则Adaptive Server拆分此页，分配一个新页并将某些行移动到新页中。页面拆分会产生开销，因为它会更新父索引页和相邻页上的页指针以及添加锁争用。对于聚簇索引，页面拆分还会更新指向新页上的行的所有非聚簇索引。

有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 5 章 “Indexes”（索引）。

减少页面拆分以进行升序键插入

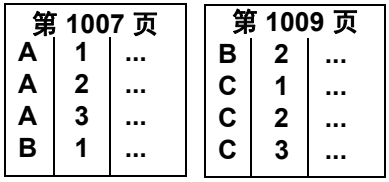
如果 “Page Splits”较高而您的应用程序正在将值插入到在复合键上带有聚簇索引的所有页锁定表中，则或许可以通过特殊优化操作 （更改这些索引的页面拆分点）来减少页面拆分数。

特殊优化操作旨在减少页面拆分并导致更完整地填充数据页。它只影响带有复合键的聚簇索引，这里第一个键已经在表中使用，第二列基于增加的值。

缺省数据页面拆分

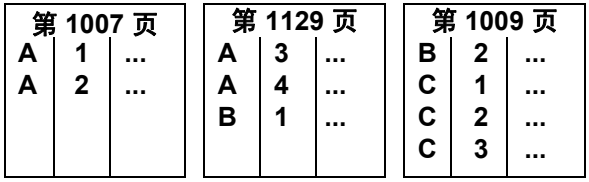
表 sales 中的 store_id 和 customer_id 上具有聚簇索引。有三个商店（A、B 和 C）。每个商店都按数值的升序添加客户记录。表包含键值为 A,1； A,2； A,3； B,1； B,2； C,1； C,2 和 C,3 的行，且每页都有四行，如图 2-2 所示。

图 2-2：插入前的聚簇表



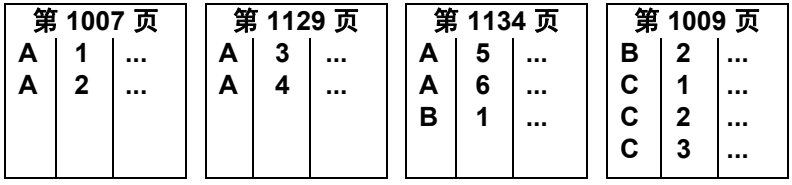
使用常规页面拆分机制时，如果插入 “A,4”，将分配新页，并且会将一半的行移动到新页，再将新行插入到相应位置，如图 2-3 所示。

图 2-3：插入操作引起页面拆分



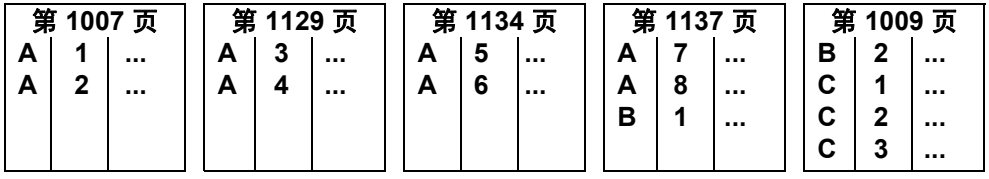
插入 “A,5” 时，不需要拆分，但插入 “A,6” 时会进行另一次拆分，如图 2-4 所示。

图 2-4：另一次插入引起另一次页面拆分



添加 “A,7” 和 “A,8” 会导致另一次页面拆分，如图 2-5 所示。

图 2-5：页面拆分继续

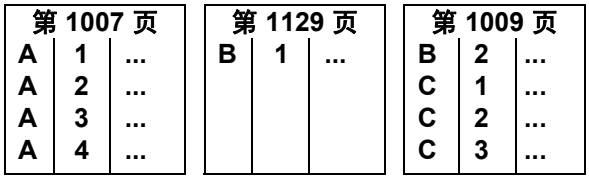


有关数据页面拆分的详细信息，请参见 《性能和调优：基础知识》中的第 12 章 “索引的工作原理”。

升序插入的影响

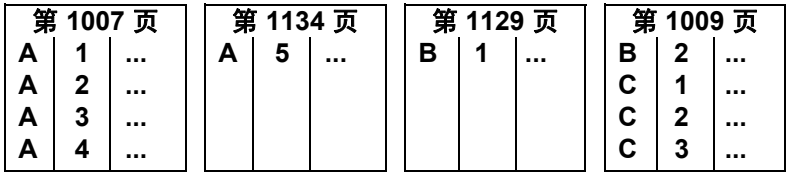
您可以对表设置升序插入模式，以便页面可在插入行处拆分，而不是在页面中间拆分。从第 80 页的图 2-2 中显示的原始表开始，插入 “A,4” 会导致在插入点处拆分，且此页中的其余行会移动到新分配的页中，如图 2-6 所示。

图 2-6：采用升序插入模式的第一次插入



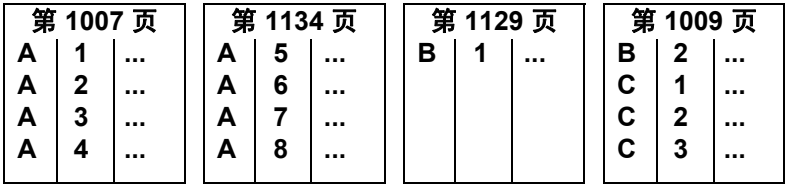
插入 “A,5” 会导致分配新页，如图 2-7 所示。

图 2-7：其它升序插入引起页面分配



添加 “A,6”、“A,7” 和 “A,8” 会填充新页，如图 2-8 所示。

图 2-8：其它插入填充新页



设置表的升序插入模式

若要为 sales 表打开升序插入模式，请使用：

```
dbcc tune (ascinserts, 1, "sales")
```

若要关闭升序插入模式，请使用：

```
dbcc tune (ascinserts, 0, "sales")
```

这些命令会更新 sysindexes 的 status2 位。

如果表有时会遇到随机插入且批处理作业期间的顺序插入较多，则应只在批处理作业运行期间启用 dbcc tune (ascinserts)。

Retries 和 Deadlocks

导致死锁的索引页面拆分及收缩的数目。 Adaptive Server 具有名为 deadlock retries 的机制，该机制可尝试避免由索引页死锁引起的事务回退。“Retries”报告 Adaptive Server 使用此机制的次数。

当两个事务中的每一个事务需要获得由另一个事务持有的锁时，会发生索引页的死锁。在数据页中，死锁会导致选择一个进程 （CPU 累计时间最少的进程）作为死锁对象并回退此进程。

截止到发生索引死锁时，事务已经更新数据页并持有数据页锁，因此回退此事务将会导致开销。

在由页面拆分和收缩引起的绝大多数索引死锁中，两个事务都可通过删除一组索引锁并重新启动索引扫描而获得成功。用于其中一个进程的索引锁被释放 （仍持有数据页中的锁）， Adaptive Server再次尝试索引扫描，遍历索引根页中的索引。

通常情况下，在扫描到达需要拆分的索引页时，其它事务已经完成，将不会发生任何死锁。缺省情况下，在认为事务已死锁并回退前，任何因页面拆分或收缩引起的索引死锁最多重试五次。

有关使用 sp_configure “deadlock retries” 更改死锁重试次数缺省值的信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

死锁重试机制使持有数据页锁的时间比常规情况下略长，并增加锁定和开销。但是，该机制可减少由于死锁造成回退的事务数量。缺省设置对下列两种情况提供合理的折衷：持有数据页锁时间较长的开销和回退必须重新发出事务的开销。

大量的索引死锁和死锁重试说明在索引 B 树的小区域内争用很激烈。

如果您的应用程序遇到大量的死锁重试，请在重新创建此索引时使用 fillfactor 减少页面拆分。

有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 5 章 “Indexes”（索引）。

Add Index Level

添加新索引级的次数。这种情况并不频繁发生，因此大多数情况下都应该看到结果值为 0。如果您的样本包括对一个空表或带有索引的小表的插入操作，计数值可能为 1 或 2。

Page Shrinks

删除索引行引起索引收缩页的次数。收缩需要开销，因为需要在索引中锁定，同时还需要在相邻页中更新指针。大于 0 的重复 “count”值表示索引中可能有许多页，但由于删除和更新操作导致每页中的行数相当少。如果存在大量的收缩，请考虑重建索引。

Index Scans

报告按锁定方案进行的正向和反向扫描：

- Ascending Scans — 报告对所有页锁定表的正向扫描数。
- DOL Ascending Scans — 报告对 DOL 锁定表进行的正向扫描数。
- Descending Scans — 报告对所有页锁定表进行的反向扫描数。
- DOL Descending Scans — 报告对 DOL 锁定表进行的反向扫描数。

有关正向和反向扫描的详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的 “Indexes”（索引）。

元数据高速缓存管理

报告元数据高速缓存的使用情况，元数据高速缓存储三种类型元数据的相关信息：对象、索引和数据库。本部分也报告在采样间隔期间处于活动状态的对象、索引和数据库描述符的数量，以及服务器最近一次启动后所使用的描述符的最大数量。它也报告对象和索引元数据高速缓存的螺旋锁争用情况。

样本输出

```
Metadata Cache Management
-------------------------
  Metadata Cache Summary            per sec   per xact     count    % of total
  --------------------------       --------- --------  --------    ---------
  Open Object Usage
    Active                              n/a       n/a      4604           n/a
    Max Ever Used Since Boot            n/a       n/a      4612           n/a
    Free                                n/a       n/a     25396           n/a
    Reuse Requests
      Succeeded                         n/a       n/a         0           n/a
      Failed                            n/a       n/a         0           n/a

  Open Index Usage
    Active                              n/a       n/a      2747           n/a
    Max Ever Used Since Boot            n/a       n/a      2753           n/a
    Free                                n/a       n/a      2253           n/a
    Reuse Requests
      Succeeded                         n/a       n/a         0           n/a
      Failed                            n/a       n/a         0           n/a

  Open Partition Usage
    Active                              n/a       n/a      2747           n/a
    Max Ever Used Since Boot            n/a       n/a      2753           n/a
    Free                                n/a       n/a      2253           n/a
    Reuse Requests
      Succeeded                         n/a       n/a         0           n/a
      Failed                            n/a       n/a         0           n/a

  Open Database Usage
    Active                              n/a       n/a        32           n/a
    Max Ever Used Since Boot            n/a       n/a        32           n/a
    Free                                n/a       n/a        18           n/a
    Reuse Requests
      Succeeded                         n/a       n/a         0           n/a
      Failed                            n/a       n/a         0           n/a

  Object Manager Spinlock Contention    n/a       n/a       n/a          1.3 %

  Object Spinlock Contention            n/a       n/a       n/a           0.0
  Index Spinlock Contention             n/a       n/a       n/a          0.0 %

  Index Hash Spinlock Contention        n/a       n/a       n/a           0.0
  Partition Spinlock Contention         n/a       n/a       n/a          0.0 %

  Partition Hash Spinlock Contention    n/a       n/a       n/a          0.0 %
```

Open Object Usage、 Open Index Usage、 Open Partition Usage 和 Open Database Usage

这些部分中每一部分都包含这三种类型元数据高速缓存的相同的信息。输出提供下列信息：

- “Active”报告在采样间隔期间处于活动状态的对象、索引或数据库的数量。
- “Max Ever Used Since Boot”报告自上一次重新启动 Adaptive Server 以来使用的描述符的最大数量。
- “Free”报告高速缓存中可用描述符的数量。
- “Reuse Requests”报告必须搜索高速缓存以查找可重用的描述符的次数：
  - “Failed”表示高速缓存中的所有描述符都在使用且发出请求的客户端收到一条错误消息。
  - “Succeeded”表示此请求在高速缓存中找到可重用的描述符。虽然 “Succeeded”表示客户端没有收到错误消息，但 Adaptive Server 还要做一些额外的工作来查找高速缓存中可重用的描述符，以及读取磁盘中的元数据信息。

您可以使用此信息设置以下配置参数：number of open indexes、number of open objects 和 number of open databases，如表 2-1 所示。

表 2-1：根据元数据高速缓存使用情况统计信息采取的操作

***sp_sysmon* 输出**	**操作**
大量的 “Free”描述符	设置较低的参数
非常少的 “Free”描述符	设置较高的参数
“Reuse Requests Succeeded”非零	设置较高的参数
“Reuse Requests Failed”非零	设置较高的参数

Descriptors immediately discarded

Adaptive Server 立即放弃的所有描述符数量。有许多配置参数可用于确定 Adaptive Server 使用的描述符数量（例如 o/s file descriptors 和 txn to pss ratio）。请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Object Manager Spinlock Contention

这是一个全服务器范围的螺旋锁，用于管理对象描述符的内部状态。

如果对此螺旋锁的争用大于 10%，则可以使用 dbcc tune(des_bind) 降低争用并提高服务器的可扩展性。请参见 《Adaptive Server Enterprise 参考手册》中的 dbcc tune(des_bind)。

Object Spinlock Contention 和 Index Spinlock Contention

报告对象描述符和索引描述符高速缓存上的螺旋锁争用情况。可使用此信息调整以下配置参数：open object spinlock ratio 和 open index spinlock ratio。如果报告的争用超过 3%，请减少相应参数的值以减少受单个螺旋锁保护的对象或索引的数量。

Index Hash Spinlock Contention

报告索引元数据高速缓存散列表上的螺旋锁争用情况。您可使用此信息对 open index hash spinlock ratio 配置参数进行调优。如果已报告的争用大于 3%，请减小此参数的值。

使用 *sp_monitorconfig* 查找元数据高速缓存使用情况的统计信息

sp_monitorconfig 显示某些共享服务器资源上的元数据高速缓存使用情况统计信息，包括：

- 可在任何时刻打开的数据库、对象及索引数
- 参照完整性查询使用的辅助扫描描述符的数量
- 可用的描述符和活动的描述符数量
- 活动的描述符的百分比
- 服务器最近启动以来使用的描述符的最大数量
- 过程高速缓存的当前大小和实际使用量

例如，假设 number of open indexes 配置参数为 500。在高峰期，您可以运行 sp_monitorconfig 获得索引描述符的元数据高速缓存实际使用情况的准确读数：

```
1> sp_monitorconfig "number of open indexes"

Usage information at date and time:Apr 22 2002  2:49PM.
Name                        Num_free    Num_active  Pct_act  Max_Used
Reuse_cnt   Instance_Name
------------------------   ----------  ----------  -------  -----------
----------  --------------------
number of open                     217         283    56.60          300
         0                  NULL
```

在该报告中，尽管 Adaptive Server 配置为 500，但服务器自上次启动以来使用的打开索引的最大数量是 300。因此，可以将 number of open indexes 配置参数重新设置为 330，以容纳使用的最大索引描述符数目 300，外加 10% 的额外空间。

还可以使用 sp_monitorconfig 'procedure cache size' 确定过程高速缓存的当前大小，该参数描述过程高速缓存中的空间量以及实际使用的最大空间量。例如，以下服务器中的过程高速缓存配置为 20,000 页：

```
1> sp_configure "procedure cache size"

option_name                    config_value run_value
------------------------------ ------------ ---------
procedure cache size                  20000      3271
```

但是，当您运行 sp_montorconfig “procedure cache size” 时，您会发现所使用的大多数过程高速缓存都是 14241 页，这意味着可以通过降低过程高速缓存的运行值来节省内存：

```
1> sp_monitorconfig "procedure cache size"

Usage information at date and time:Apr 22 2002  2:49PM.
Name                        Num_free    Num_active  Pct_act  Max_Used
Reuse_cnt   Instance_Name
------------------------   ----------  ----------  -------  -----------
----------  --------------------
procedure cache                  5878       14122    70.61        14241
          0                    NULL
```

Partition Spinlock Contention

指示分区的螺旋锁争用的数量。通过将高速缓存划分为更多分区可减少分区的螺旋锁争用。请参见 《系统管理指南，卷 2》中的第 5 章 “内存使用和性能”以及 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 2 章 “Locking Configuration and Tuning”（锁定配置和调优）。

Partition Hash Spinlock Contention

指示分区的散列螺旋锁争用的数量。请参见 《系统管理指南，卷 2》中的第 5 章“内存使用和性能”以及 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 2 章 “Locking Configuration and Tuning”（锁定配置和调优）。

Lock Management

报告锁、死锁、锁升级和锁争用。

样本输出

```
Lock Management
---------------

Lock Summary                per sec      per xact       count   % of total
------------------------  ------------  -----------  ---------  ---------
Total Lock Requests          279236.5        6878.6  167541893        n/a
Avg Lock Contention               1.2           0.0        691       0.0 %
Cluster Locks Retained            0.0           0.0          0       0.0 %
Deadlock Percentage               0.0           0.0          0       0.0 %

Lock Detail                 per sec      per xact       count   % of total
------------------------  ------------  -----------  ---------  ---------

Table Lock Hashtable
  Lookups                    9571.4         235.8     5742818        n/a
  Avg Chain Length              n/a           n/a     0.00981        n/a
  Spinlock Contention           n/a           n/a         n/a       0.2 %
```

```
排它表
  Granted                      14.3           0.4         8580       100.0 %
  Waited                        0.0           0.0            0         0.0 %
------------------------  ------------  ------------  ----------  ----------
Total EX-Table Requests        14.3           0.4         8580         0.0 %

共享表
  Total SH-Table Requests       0.0           0.0            0          n/a

排它意图
  Granted                     139.7           3.4        83793       100.0 %
  Waited                        0.0           0.0            0         0.% 
------------------------  ------------  ------------  ----------  ----------
Total EX-Intent Requests      139.7           3.4        83793         0.1 %

共享意图
  Granted                    9412.9         231.9      5647759       100.0 %
  Waited                        0.0           0.0            0         0.0 %
------------------------  ------------  ------------  ----------  ----------
Total SH-Intent Requests     9412.9         231.9      5647759         3.4 %

Page & Row Lock HashTable
  Lookups                  173637.2        4277.3    104182335          n/a
  Avg Chain Length              n/a           n/a      0.01387          n/a
  Spinlock Contention           n/a           n/a          n/a         0.4 %

Exclusive Page
  Granted                     306.5           7.6       183911       100.0 %
  Waited                        0.0           0.0            2         0.0 %
------------------------  ------------  ------------  ----------  ----------
Total EX-Page Requests        306.5           7.6       183913         0.1 %

Update Page
  Granted                   19052.6         469.3     11431583       100.0 %
  Waited                        0.1           0.0           60         0.0 %
------------------------  ------------  ------------  ----------  ----------
Total UP-Page Requests      19052.7         469.3     11431643         6.8 %

Shared Page
  Granted                   63587.2        1566.4     38152347       100.0 %
  Waited                        0.0           0.0            0         0.0 %
------------------------  ------------  ------------  ----------  ----------
Total SH-Page Requests      63587.2        1566.4     38152347        22.8 %

Exclusive Row
  Granted                     33.9           0.8        20326       100.0 %
```

```
  Waited                          0.0          0.0          0        0.0 %
------------------------  ------------  ------------  ----------  ----------
Total EX-Row Requests            33.9          0.8       20326        0.0 %

Update Row
  Granted                        10.1          0.2        6030       98.2 %
  Waited                          0.2          0.0         108        1.8 %
------------------------  ------------  ------------  ----------  ----------
Total UP-Row Requests            10.2          0.3        6138        0.0 %

Shared Row
  Granted                     88431.3       2178.4    53058781      100.0 %
  Waited                          0.0          0.0           0        0.0 %
------------------------  ------------  ------------  ----------  ----------
Total SH-Row Requests        88431.3       2178.4    53058781       31.7 %

Next-Key
  Total Next-Key Requests         0.0          0.0           0         n/a

Address Lock Hashtable
  Lookups                     98247.7       2420.2    58948613         n/a
  Avg Chain Length                n/a          n/a     0.00019         n/a
  Spinlock Contention             n/a          n/a         n/a        0.5 %

Exclusive Address
  Granted                      2758.9         68.0     1655359      100.0 %
  Waited                          0.6          0.0         380        0.0 %
------------------------  ------------  ------------  ----------  ----------
Total EX-Address Requests     2759.6         68.0     1655739        1.0 %

Shared Address
  Granted                     95487.9       2352.2    57292733      100.0 %
  Waited                          0.2          0.0         141        0.0 %
------------------------  ------------  ------------  ----------  ----------
Total SH-Address Requests    95488.1       2352.2    57292874       34.2 %

Last Page Locks on Heaps
  Granted                       799.4         19.7      479637      100.0 %
  Waited                          0.0          0.0           0        0.0 %
------------------------  ------------  ------------  ----------  ----------
Total Last Pg Locks              799.4         19.7      479637      100.0 %

Deadlocks by Lock Type          per sec      per xact       count   % of total
------------------------  ------------  ------------  ----------  ----------
Total Deadlocks                 0.0          0.0           0         n/a
```

```
Deadlock Detection
  Deadlock Searches                 0.0           0.0          19          n/a
  Searches Skipped                  0.0           0.0           0        0.0 %
  Avg Deadlocks per Search          n/a           n/a     0.00000          n/a

Lock Promotions
  Total Lock Promotions             0.0           0.0           0          n/a
Lock Timeouts by Lock Type       per sec      per xact       count   % of total
------------------------   ------------   -----------   ---------   ---------
Total Timeouts                    0.0           0.0           0          n/a


Cluster Lock Summary             per sec      per xact       count   % of total
------------------------   ------------   -----------   ---------   ---------

Physical Locks Summary           per sec      per xact       count   % of total
------------------------   ------------   -----------   ---------   ---------
No physical locks are acquired

Logical Locks Summary            per sec      per xact       count   % of total
------------------------   ------------   -----------   ---------   ---------

Object Locks Summary             per sec      per xact       count   % of total
------------------------   ------------   -----------   ---------   ---------
```

即使采样间隔期间没有出现明细行，也会报告明细行；例如上述样本输出中的 “Deadlocks by Lock Type”。

Lock Summary

概述采样间隔期间发生的锁活动的统计信息。

- “Total Lock Requests”报告锁请求的总数。
- “Avg Lock Contention”报告锁争用的平均次数，以占锁请求总数的百分比形式表示。

该值不足以确定锁争用的严重程度。若要确定锁争用的严重程度，请查询 monSysWaits、monProcessWaits 和 monLocks 监控表，或者使用 sp_object_stats。有关详细信息，请参见 Performance and Tuning Series: Monitoring Tables （《性能和调优系列：监控表》）。有关 *sp_object_stats* 的详细信息，请参见 《Adaptive Server 参考手册：过程》。有关调优锁定行为的详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）。

- “Cluster Locks Retained”表示保留的集群锁的数量。如果该数值很高，则表示应用程序经过很好的分区，并且大多数集群锁已经授予当前实例。如果该数值较低，可能表示您应该重新评估对象的模式和负载模式，以减少多个实例之间的数据访问冲突。

  monCLMobjectActivity 包含每个对象的集群级锁活动的值。这些值可帮助在对象级别做出锁定决策。

- “Deadlock Percentage”报告死锁占锁请求总数的百分比。如果该值很高，请参见第 94 页的 “Deadlocks by Lock Type”。

Lock Detail

提供可帮助确定应用程序是否会引起锁争用或与死锁相关的问题的信息。

锁争用对 Adaptive Server 的性能可能有较大影响。表锁产生的锁争用比页或行锁要多，因为没有其它的任务可访问已具有排它表锁的表，且如果任务需要排它表锁，它必须等待直到所有共享锁释放。如果锁争用很严重，请运行 sp_object_stats 来帮助找出涉及到的表。

“Lock Detail”按类型报告锁：

- Granted — 立即向应用程序授予每个锁类型的次数。
- Waited — 每个锁类型等待特定锁类型的次数。
- Lookups —
- Avg Hash Chain Length — 采样间隔期间每个散列桶的平均锁数。
- Spinlock Contention — 引擎遇到螺旋锁争用的次数。

“Lock Detail”报告以下锁类型的 `Granted` 和 `Waited` 值：

- Exclusive Table
- Exclusive Intent
- Shared Intent

- Exclusive Page
- Exclusive Row
- Update Row
- Shared Row
- Shared Address
- Last Page Locks on Heaps

“Lock Detail”报告以下锁类型的 Avg Hash Chain Length、 Lookups 和 Spinlock Contention：

- Shared Table
- Page & Row Lock HashTable
- Update Page
- Shared Page
- Next-Key
- Address Lock Hashtable

您可以使用 lock hashtable size 配置锁散列表的大小。如果每个散列链的锁平均数大于四，则应考虑增加散列表的大小。

“% of total”列显示授予的或被迫等待的特定锁类型占锁请求总数的百分比。

采用批量复制的大批量插入操作是此指导原则的一种例外情况。锁散列链长度可能在大批量复制期间变长。

请参见 Performance and Tuning Series: Locking and Concurrency Control（《性能和调优系列：锁定和并发控制》）中的第 2 章“Locking Configuration and Tuning”（锁定配置和调优）以及《系统管理指南，卷 1》中的第 5 章“设置配置参数”。

Exclusive Address 和 Shared Address

立即授予地址锁的次数或任务必须等待锁的次数。地址锁存放在所有页锁定表的索引页上。地址锁可能会造成严重的影响，因为索引页上的锁会阻止对索引页指向的所有数据页的访问。

堆上的最后页锁

报告分区堆表或未分区堆表的最后一页中的锁定尝试。它只报告所有页锁定表。

此信息可表明系统中是否存在这样的表：它们能够从使用仅数据锁定、分区或增加的分区数中受益。添加在数据页中随机分配插入的聚簇索引也会有所帮助。如果您知道一个或多个表已出现争用最后页的问题，则监控表可帮助确定遇到该问题的表。请参见 Performance and Tuning Series: Monitoring Tables （《性能和调优系列：监控表》）。

有关分区如何帮助解决未分区堆表中最后页锁定问题的信息，请参见 《性能和调优系列：物理数据库调优》中的第 1 章 “控制物理数据放置”。

Table Lock Hashtable

在锁散列表中搜索页、行或表上的锁的次数。“Table Lock Hashtable”报告：

- Lookups
- Avg Chain Length
- 螺旋锁争用

使用配置参数 lock hashtable size 配置锁散列表的大小。如果每个散列链的锁平均数大于四，则应考虑增加散列表的大小。

有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 2 章 “Locking Configuration and Tuning”（锁定配置和调优）。

Deadlocks by Lock Type

特定类型的死锁数量。“% of total”给出每种死锁的数量，以占死锁总数的百分比形式表示。

同一数据库中许多事务同时执行的情况下可能发生死锁。随着事务之间的锁争用的增加，死锁就会更加经常发生。

本类别报告下列死锁类型的数据：

- 排它表
- 共享表
- 排它意图

- 共享意图
- 排它页
- 更新页
- 共享页
- 排它行
- 更新行
- 共享行
- 共享下一键
- 排它地址
- 共享地址
- 其它

“Total Deadlocks”汇总了所有锁类型的数据。

正如这一部分的示例所示，如果没有任何死锁，sp_sysmon 不显示任何详细信息，而是只输出带零值的 “Total Deadlocks”行。

若要找出死锁发生的位置，请执行以下一种或两种操作：

- 使用 sp_object_stats。

  有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 2 章 “Locking Configuration and Tuning”（锁定配置和调优）。

- 启用将详细死锁信息输出到日志功能。

  有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 3 章 “Locking Reports”（锁定报告）。

Deadlock Detection

在采样间隔期间找到死锁的死锁搜索次数，以及跳过的死锁搜索的次数。

有关详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 3 章 “Locking Reports”（锁定报告）。

Deadlock Searches

Adaptive Server 在采样间隔期间启动死锁搜索的次数。对于未发生死锁或死锁级别很低的应用程序来说，死锁检查是很费时的开销。可将此数据与第 96 页的 “Average Deadlocks per Search”结合使用以确定 Adaptive Server 对死锁的检查是否过于频繁。

Searches Skipped

某个任务开始执行死锁检查但发现死锁检查正在进行而跳过此检查的次数。“% of total”报告跳过的死锁搜索占搜索总数的百分比。

当锁争用阻止某进程时，该进程会等待一个时间间隔 （由配置参数 deadlock checking period 设置）。此阶段过去后，该进程开始死锁检查。如果搜索已在进行中，则该进程将跳过该搜索。

如果发现有一些搜索被跳过，但某些搜索正在查找死锁，则应略微增加参数值。如果发现许多搜索被跳过，而且没有任何死锁或很少有死锁，则可以大幅度增加参数值。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Average Deadlocks per Search

报告每次搜索发现的死锁平均数。

本类别衡量 Adaptive Server 是否过于频繁地检查死锁。如果应用程序很少死锁，您可以通过增加 deadlock checking period 配置参数的值来调整任务搜索死锁的频率。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Lock Promotions

发生下列升级的次数：

- “Ex-Page to Ex-Table” — 排它页到排它表。
- “Sh-Page to Sh-Table” — 共享页到共享表。
- “Ex-Row to Ex-Table” — 排它行到排它表。
- “Sh-Row to Sh-Table” — 共享行到共享表。

- “Sh-Next-Key to Sh-Table” — 共享下一键到共享表。

“Total Lock Promotions”报告每秒和每个事务的组合锁升级类型的平均数。

如果在采样间隔期间没有发生任何锁升级，则只输出行总数。

如果没有锁升级，则 sp_sysmon 不显示详细信息。

“Lock Promotions”数据可以：

- 帮助检测应用程序中的锁升级是否是锁争用和死锁的原因。
- 在调优锁升级变量前后用于确定值的效力。

查看 “Granted”数据和 “Waited”数据以查找争用迹象。如果锁争用多且锁升级发生频繁，可考虑更改所涉及表的锁升级阈值。

您可以在服务器范围内或对单个表配置锁升级阈值。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Lock Time-out Information

某个任务等待获取锁且由于会话级或服务器级锁超时而回退事务的次数。只有在采样期间发生锁超时时，才输出显示锁类型的详细行。如果没有发生任何锁超时，则显示 “Total Lock Time-outs”行以及所有等于 0 的值。

有关锁超时的详细信息，请参见 Performance and Tuning Series: Locking and Concurrency Control （《性能和调优系列：锁定和并发控制》）中的第 4 章 “Using Locking Commands”（使用锁定命令）。

Cluster Lock Summary

sp_sysmon 报告集群锁的以下活动：

- Lock Garbage Collection — 锁碎片收集的运行次数。如果每秒钟有大量碎片收集，则表示集群的锁数量不足。
- Targeted Collection Success — 锁碎片收集能够成功回收目标锁数的次数。 Cluster Edition 在内部确定目标锁数，它不是一个可配置的参数。如果每秒钟有少量碎片收集，则表示集群的锁数量不足或可用的 CIPC 消息空间量不足。

- Cluster Lock Requests — 由于出现争用而不能授予并且需要集群通信的用户物理、逻辑或对象锁请求的数量。较高的值表示需要大量集群通信量。
- Local Master — 拥有本地所有者的物理、逻辑或对象锁的数量。较高的值表示理想锁，这可能表示在满足集群锁请求时只需要较少的开销。

  如果 Cluster Edition 找不到本地所有者：

  - 您可能需要重新评估您的应用程序分区。
  - 单个实例有太多负载，这会导致在所有者和远程实例之间进行负载平衡。考虑更改负载模式 （例如对数据进行分区）。
- Lock Granted — 等待远程实例的集群锁请求的数量，该远程实例拥有不存在冲突的任务所有权的集群锁。较高的 `Lock Granted` 值表示集群中的多个实例之间的锁请求出现冲突。请重新评估您的应用程序分区。
- Lock Waited — 等待远程实例的集群锁请求的数量，该远程实例拥有存在冲突的任务所有权的集群锁。较高的 `Lock Waited` 值表示锁级别争用：不同实例上的两个任务尝试获取同一锁，因此出现冲突。为了纠正此问题，请重新评估您的应用程序分区。
- Downgrade Req Recv — 显示远程实例在请求由本地实例拥有的锁时发送的降级请求的数量。

数据高速缓存管理

sp_sysmon 报告所有高速缓存的摘要统计信息，以及每个命名高速缓存的统计信息。

sp_sysmon 报告缺省数据高速缓存和每个命名高速缓存的下列活动：

- 螺旋锁争用
- 使用率
- 高速缓存搜索，包括命中和未命中次数
- 所有配置的池的池周转率
- 缓冲区清洗行为，包括清理干净的缓冲区、已在 I/O 中的缓冲区和清洗脏了的缓冲区。

- 执行的和拒绝的预取请求
- 脏读页请求

使用 sp_cacheconfig 和 sp_helpcache 输出帮助分析报告中这一部分的数据。sp_cacheconfig 提供有关高速缓存和池的信息，而 sp_helpcache 提供有关绑定到高速缓存的对象的信息。

有关如何使用这些系统过程的信息，请参见 《系统管理指南，卷 2》中的第 4 章 “配置数据高速缓存”。

有关性能问题和命名高速缓存的详细信息，请参见 《性能和调优系列：基础知识》中的第 5 章 “内存使用和性能”。

样本输出

以下样本显示 “数据高速缓存管理”类别的 sp_sysmon 输出。第一个数据块，“Cache Statistics Summary”包括所有高速缓存的信息。sp_sysmon 报告每个高速缓存的单独数据块。这些块根据高速缓存名称来识别。虽然在间隔期间配置了更多的高速缓存，但这里显示的输出样本只包括缺省数据高速缓存。

```
数据高速缓存管理
----------------------

Cache Statistics Summary (All Caches)
-------------------------------------
                            per sec       per xact       count   % of total
                          ----------     -----------    -------    --------
Cache Search Summary
  Total Cache Hits           39472.3        185026.5   11841696     99.9 %
  Total Cache Misses            33.8           158.4      10139      0.1 %
----------------------   ------------    ------------   --------
Total Cache Searches         39506.1        185184.9   11851835

Cache Turnover
  Buffers Grabbed              177.0           829.6      53097        n/a
  Buffers Grabbed Dirty          0.0             0.0          0      0.0 %

Cache Strategy Summary
  Cached (LRU) Buffers       35931.1        168427.1   10779332     98.6 %
  Discarded (MRU) Buffers      511.7          2398.4     153500      1.4 %

Large I/O Usage
  Large I/Os Performed          46.9           219.9      14071     94.5 %
```

```
Large I/Os Denied due to
  Pool < Prefetch Size              2.5            11.6        745      5.0 %
  Pages Requested
  Reside in Another
  Buffer Pool                       0.3             1.3         80      0.5 %
----------------------   ------------    ------------   --------
Total Large I/O Requests           49.7           232.8      14896

Large I/O Effectiveness
  Pages by Lrg I/O Cached         171.3           803.1      51398       n/a
  Pages by Lrg I/O Used           567.2          2658.7     170157    331.1 %

Asynchronous Prefetch Activity
  APFs Issued                       9.6            45.0       2878      1.2 %
  APFs Denied Due To
    APF I/O Overloads               0.0             0.0          0      0.0 %
    APF Limit Overloads             0.0             0.0          0      0.0 %
    APF Reused Overloads            0.0             0.0          0      0.0 %
  APF Buffers Found in Cache
      With Spinlock Hel             1.7             7.8        501      0.2 %
      W/o Spinlock Held           787.1          3689.5     236128     98.6 %
----------------------   ------------    ------------   --------
Total APFs Requested              798.4          3742.3     239507

  Other Asynchronous Prefetch Statistics
    APFs Used                       9.4            43.9       2808       n/a
    APF Waits for I/O               6.8            31.8       2034       n/a
    APF Discards                    0.0             0.0          0       n/a

Dirty Read Behavior
    Page Requests                   0.0             0.0          0       n/a

------------------------------------------------------------------------------
Cache:default data cache
                                per sec        per xact      count   % of total
                           ----------     -----------    -------    --------
  Spinlock Contention              n/a             n/a        n/a       2.8 %

  Utilization                      n/a             n/a        n/a      82.9 %

  Cache Searches
    Cache Hits                 32731.6        153429.6    9819492    100.0 %
      Found in Wash              288.4          1351.7      86508      0.9 %
    Cache Misses                  10.9            50.9       3257      0.0 %
----------------------   ------------    ------------   --------
Total Cache Searches           32742.5        153480.5    9822749
```

```
Pool Turnover
  2  Kb Pool
    LRU Buffer Grab                123.0            576.7          36908     92.5 %
      Grabbed Dirty                  0.0              0.0              0      0.0 %
  16 Kb Pool
    LRU Buffer Grab                 10.0             47.1           3012      7.5 %
      Grabbed Dirty                  0.0              0.0              0      0.0 %
----------------------   ------------   ------------   ----------
Total Cache Turnover               133.1            623.8          39920

Buffer Wash Behavior
  Statistics Not Available - No Buffers Entered Wash Section Yet

Cache Strategy
  Cached (LRU) Buffers           30012.1         140681.9        9003640     98.3 %
  Discarded (MRU) Buffers          511.7           2398.4         153500      1.7 %

Large I/O Usage
  Large I/Os Performed              10.0             47.1           3012     80.2 %

  Large I/Os Denied due to
    Pool < Prefetch Size             0.8              3.9            247      6.6 %
    Pages Requested
    Reside in Another
    Buffer Pool                      1.7              7.8            498     13.3 %
----------------------   ------------   ------------   ----------
Total Large I/O Requests           12.5             58.7           3757

Large I/O Detail
  4   Kb Pool
    Pages Cached                    0.0              0.0              0       n/a
    Pages Used                      0.0              0.0              0       n/a
  16  Kb Pool
    Pages Cached                   80.3            376.5          24096       n/a
    Pages Used                     76.1            356.5          22817     94.7 %




Dirty Read Behavior
      Page Requests                 0.0              0.0              0       n/a


-------------------------------------------------------------------------------
Cache:pub_log_cache
                           per sec        per xact      count  % of total
                         ----------     -----------    ----------  -------
```

```
  Spinlock Contention                n/a             n/a           n/a    0.2 %

  Utilization                        n/a             n/a           n/a    9.1 %

  Cache Searches
    Cache Hits                    3591.9         16837.1       1077574  100.0 %
      Found in Wash                  0.0             0.0             0    0.0 %
    Cache Misses                     0.0             0.0             0    0.0 %
-----------------------   ------------    ------------    ----------
Total Cache Searches              3591.9         16837.1       1077574

Pool Turnover
  2  Kb Pool
    LRU Buffer Grab                0.3             1.3            80    0.8 %
      Grabbed Dirty                0.0             0.0             0     0.0 %
  4  Kb Pool
    LRU Buffer Grab               33.9           158.9         10171   99.2 %
      Grabbed Dirty                0.0             0.0             0    0.0 %
-----------------------   ------------    ------------    ----------
Total Cache Turnover              34.2           160.2         10251

Buffer Wash Behavior
  Statistics Not Available - No Buffers Entered Wash Section Yet

Cache Strategy
  Cached (LRU) Buffers          2872.7         13465.9        861817  100.0 %
  Discarded (MRU) Buffers          0.0             0.0             0    0.0 %

Large I/O Usage
Large I/Os Performed              33.9           158.9         10171   99.2 %

Large I/Os Denied due to
  Pool < Prefetch Size             0.0             0.0             0    0.0 %
  Pages Requested
  Reside in Another
  Buffer Pool                      0.3             1.3            80    0.8 %
-----------------------   ------------    ------------    ----------
Total Large I/O Requests          34.2           160.2         10251

Large I/O Detail
  4   Kb Pool
    Pages Cached                  67.8           317.8         20342       n/a
    Pages Used                    67.8           317.8         20342   100.0 %

Dirty Read Behavior
    Page Requests                  0.0             0.0             0      n/a
```

```
------------------------------------------------------------------------------
Cache:tempdb_data_cache
                              per sec        per xact      count  % of total
                           ----------     -----------    ----------  -------
  Spinlock Contention             n/a             n/a          n/a      0.0 %

  Utilization                     n/a             n/a          n/a      8.0 %

Cache Searches
  Cache Hits                   3148.8         14759.8       944630     99.3 %
    Found in Wash                 7.9            37.0         2365      0.3 %
Cache Misses                    22.9           107.5         6882      0.7 %
-----------------------    ------------    ------------   ----------
Total Cache Searches          3171.7         14867.4       951512

Pool Turnover
  2  Kb Pool
    LRU Buffer Grab              6.8            31.8         2038     69.7 %
      Grabbed Dirty              0.0             0.0            0      0.0 %
  4  Kb Pool
    LRU Buffer Grab              0.1             0.4           24      0.8 %
      Grabbed Dirty              0.0             0.0            0      0.0 %
  16 Kb Pool
    LRU Buffer Grab              2.9            13.5          864     29.5 %
      Grabbed Dirty              0.0             0.0            0       0.0%
-----------------------    ------------    ------------   ----------
Total Cache Turnover             9.8            45.7         2926

Buffer Wash Behavior
  Statistics Not Available - No Buffers Entered Wash Section Yet


Cache Strategy
  Cached (LRU) Buffers        3046.3         14279.3       913875    100.0 %
  Discarded (MRU) Buffers        0.0             0.0            0      0.0 %

Large I/O Usage
  Large I/Os Performed            3.0            13.9          888    100.0 %

  Large I/Os Denied due to
    Pool < Prefetch Size          0.0             0.0            0      0.0 %
    Pages Requested
    Reside in Another
    Buffer Pool                   0.0             0.0            0      0.0 %
-----------------------    ------------    ------------   ----------
Total Large I/O Requests          3.0            13.9          888
```

```
  Large I/O Detail
    4   Kb Pool
      Pages Cached                0.2             0.8              48         n/a
      Pages Used                  0.2             0.7              47      97.9 %
    16  Kb Pool
      Pages Cached               23.0           108.0            6912         n/a
      Pages Used                  2.9            13.5             864      12.5 %

Dirty Read Behavior
      Page Requests               0.0             0.0               0         n/a
```

Cache Statistics Summary (All Caches)

这部分总结了缺省数据高速缓存和所有组合的命名数据高速缓存的行为。输出每个数据高速缓存的相应信息。请参见第 108 页的 “Cache Management by Cache”。

Cache Search Summary

提供有关高速缓存命中和未命中的摘要信息。使用此数据可快速查看高速缓存设计是否有效。大量的高速缓存未命中次数说明您应考察每个高速缓存的统计信息。

- “Total Cache Hits”报告在任何高速缓存中找到所需页的次数。“% of total”报告高速缓存命中次数占高速缓存搜索总次数的百分比。
- “Total Cache Misses”报告在一个高速缓存中未找到需要的页而必需从磁盘读取的次数。“% of total”报告在高速缓存中未找到缓冲区的次数占所有高速缓存搜索次数的百分比。
- “Total Cache Searches”报告高速缓存搜索的总次数，包括所有组合的高速缓存命中次数和未命中次数。

Cache Turnover

提供高速缓存周转的摘要：

- “Buffers Grabbed”报告在所有高速缓存中替换的缓冲区数。“count”列代表 Adaptive Server 从高速缓存的 LRU 端读取缓冲区代替数据库页的次数。如果服务器最近重新启动过，缓冲区就会是空的，将页读入空缓冲区将不会统计在这里。

- “Buffers Grabbed Dirty”报告读取缓冲区时在高速缓存的 LRU 端找到脏页而在缓冲区写至磁盘时被迫等待的次数。如果该值不为零，则应确定哪些高速缓存受到影响，以及磁盘 IO 的性能是否足够并增加池清洗大小。这代表严重的性能问题。

Cache Strategy Summary

提供使用的高速缓存策略的摘要。

- “Cached (LRU) Buffers”报告所有高速缓存中位于最近使用最多的 (MRU) 和最近使用最少的 (LRU) 链顶部的缓冲区的总数。
- “Discarded (MRU) Buffers”报告所有高速缓存中遵循读取放弃策略的缓冲区 （清洗标记处的缓冲区）的总数。

Large I/O Usage

提供有关所有高速缓存中的大 I/O 请求的摘要信息。如果 “Large I/Os Denied”较高，则应考察单个高速缓存以确定原因。

- “Large I/Os Performed”衡量执行请求的大 I/O 的次数。“% of total”是执行的大 I/O 请求占已进行的 I/O 请求总数的百分比。
- “Large I/Os Denied”报告大 I/O 不能执行的次数。“% of total”报告拒绝的大 I/O 请求占已进行的请求总数的百分数。
- “Total Large I/O Requests”报告所有高速缓存的所有大 I/O 请求 （授予的和拒绝的）数。

Large I/O Effectiveness

帮助确定大 I/O 的性能优势。它将通过大 I/O 引入高速缓存的页数与在高速缓存中时被实际引用的页数进行比较。如果 “Pages by Lrg I/O Used”的百分比很低，则意味着引入高速缓存的页中只有很少的页正被查询访问。请研究各个高速缓存，以确定问题的原因。请使用 optdiag 检查每个表和索引的 “Large I/O Efficiency”值。

- “Pages by Lrg I/O Cached”报告由采样间隔期间发生的所有大 I/O 操作引入到所有高速缓存中的页数。较低的百分比可能表示存在以下情况之一：
  - 表存储中的分配分段
  - 高速缓存策略不适当

- “Pages by Lrg I/O Used”会报告在由大 I/O 引入高速缓存后使用的页的总数。如果没有 “Pages by Lrg I/O Cached”， sp_sysmon 不会显示此类别的输出结果。

Asynchronous Prefetch Activity 报告

报告所有高速缓存的异步预取活动。

有关每个数据库设备的异步预取的信息，请参见第 120 页的 “磁盘 I/O 管理”。有关异步预取的信息，请参见 《性能和调优系列：基础知识》中的第 6 章 “调优异步预取”。

“Total APFs Requested”报告符合预取条件的页的总数，即，在采样间隔期间发出的所有查询的预先设置大小的总和。“Asynchronous Prefetch Activity”中的其它行提供以下类别的详细信息：

- 已经预取的页 （“APFs Issued”）
- 拒绝预取的原因
- 在高速缓存中找到的页

APFs Issued

系统在采样间隔期间发出的异步预取请求的数量。

APFs Denied Due To

报告未发出 APF 的原因：

- “APF I/O Overloads”报告由于缺少磁盘 I/O 结构或由于磁盘信号争用而使 APF 使用被拒绝的次数。

  如果此数值很高，则应检查 “Disk I/O Management”中的以下信息：

  - disk i/o structures 配置参数的值。请参见第 122 页的 “Disk I/O Structures”。
  - 每个数据库设备的设备信号争用的值，用于确定问题原因。请参见第 125 页的 “Mirror Semaphore Granted 和 Mirror Semaphore Waited”。

  如果问题是由于缺乏磁盘 I/O 结构引起的，则应将配置参数设置提高，然后重复测试。如果问题是由于高磁盘信号争用引起的，则检查高 I/O 发生处的对象物理放置。

- “APF Limit Overloads”指示已超出可用于异步预取的缓冲池的百分比。此限制是由 global async prefetch limit 配置参数为服务器整体设置的。使用 sp_poolconfig 分别对每个池进行调优。
- “APF Reused Overloads”表示由于页链扭结，或因为在能够访问 APF 引入的缓冲区前这些缓冲区被换出而使 APF 的使用被拒绝。

  理想情况下，连续页包含连续行。但是，如果页链从其当前位置跳到较远磁盘位置处的几页，然后再跳回其原来的位置，则该页将被视为“扭结”。例如，使用页 1, 2, 396, 397, 3, 4 的页链即为扭结页链。

APF Buffers Found in Cache

在采样间隔期间，在数据高速缓存中找到的来自 APF **预见性设置**的缓冲区数量。异步预取使用快速扫描在数据高速缓存中尝试查找要读取的页，而不持有高速缓存螺旋锁。如果此操作没有成功，则 APF 在持有螺旋锁的情况下执行全面扫描。

Other Asynchronous Prefetch Statistics

报告另外三个异步预取统计信息：

- “APFs Used”报告由异步预取引入到高速缓存并在采样间隔期间使用的页数。为此报告统计的页数可能已经在采样间隔期间引入到高速缓存中，或在采样间隔开始之前由发出的异步预取请求引入到高速缓存中。
- “APF Waits for I/O”报告进程被迫等待异步预取完成的次数。这表示预取未能及时发出，从而使页未能在查询需要前位于高速缓存中。出现一定百分比的 “APF Waits”是合理的。以下一些原因可能导致任务必须等待：
  - 查询的第一个异步预取请求通常包括在 “APF Waits”中。
  - 每次顺序扫描移动到新的分配单元并发出预取请求时，查询必须等待直到第一个 I/O 完成。
  - 每次非聚簇索引扫描找到一组限定行并发出页预取请求时，它必须等待第一页返回。

  其它可影响 “APF Waits for I/O”的因素包括需要在每页上完成的处理量和 I/O 子系统的速度。
- “APF Discards”表示由异步预取读入并在使用前放弃的页数。“APFs Discards”的值较大可能表示增加缓冲池的大小有助于提高性能，也可能表示 APF 正在将页引入到查询不需要的高速缓存中。

Dirty Read Behavior

提供可帮助分析脏读 （隔离级别 0 读取）如何影响系统的信息。

Page Requests

报告在隔离级别 0 上请求的平均页数。“% of total”列报告脏读占页读取总数的百分比。

如果脏读页请求导致重新启动了许多脏读，则它们会造成很大的开销。

Dirty Read Re-starts

发生的脏读重新启动的次数。本类别仅针对整个服务器进行报告，而不针对各个高速缓存报告。如果没有 “Dirty Read Page Requests”，则 sp_sysmon 不会显示此类别的输出结果。

当在某页中脏读处于活动状态，而另一进程对此页进行了更改引起该页释放，则会发生脏读重新启动。对级别 0 的扫描必须重新启动。

“% of total”输出是在隔离级别 0 中进行的脏读重新启动占页读取总数的百分比。

如果这些值很高，您可能需要采取一些步骤，通过修改应用程序减小这些值，因为与脏读相关的开销和导致的重新启动要付出昂贵的代价。大多数应用程序都应避免重新启动，因为由此导致的开销很高。

Cache Management by Cache

报告服务器中每个活动高速缓存的利用率。样本输出显示缺省数据高速缓存的结果。

高速缓存螺旋锁争用

引擎在高速缓存中遇到螺旋锁争用而被迫等待的次数，以该次数占该高速缓存中螺旋锁请求总数的百分比表示。这只在 SMP 环境中有意义。

当用户任务对高速缓存进行任何更改时，则在进行更改过程中螺旋锁拒绝所有其它任务访问此高速缓存。虽然螺旋锁的持有时间极短，也会减慢具有高事务处理速率的多处理器系统的性能。如果螺旋锁争用超过 10%，则应考虑使用命名高速缓存或添加高速缓存分区。

有关添加高速缓存的信息，请参见 《性能和调优：基础知识》中的第 5 章 “内存使用和性能”。

Utilization

报告使用此高速缓存的搜索占跨所有高速缓存的搜索的百分比。您可对每个高速缓存的此值进行比较，以确定是否存在过度利用或未充分利用的高速缓存。如果确定了某个高速缓存没有很好地利用，您可以：

- 更改高速缓存绑定以平衡利用率。有关详细信息，请参见 《性能和调优指南：基础知识》中的第 5 章 “内存使用和性能”。
- 调整高速缓存的大小使其与本身的利用情况更加适应。

  有关详细信息，请参见 《系统管理指南，卷 2》中的第 4 章 “配置数据高速缓存”。

高速缓存搜索、命中数和未命中数信息

此高速缓存的命中数和未命中数以及搜索总数。高速缓存命中数基本上等于 statistics io 报告的逻辑读取值；高速缓存未命中数基本上等于物理读取值。 sp_sysmon 报告的值总是高于 statistics io 显示的值，因为 sp_sysmon 还报告系统表的 I/O、日志页、 OAM 页和其它系统开销。

要解释高速缓存命中数据，您需要了解应用程序如何使用每个高速缓存。在为了持有特定对象 （如索引或查询表）而创建的高速缓存中，高速缓存命中率可能会达到 100%。在用于大表随机点查询的高速缓存中，高速缓存命中率可能相当低，但高速缓存使用的效率不减。

高速缓存命中数和未命中数还可以帮助您确定添加更多内存是否能够改善性能。例如，如果 “Cache Hits”很高，添加内存可能不会有太大帮助。

Cache Hits

在数据高速缓存中找到所需页的次数。“% of total”报告高速缓存命中次数占高速缓存搜索总次数的百分比。

Found in Wash

在高速缓存的清洗部分中找到所需页的次数。“% of total”报告在清洗区找到缓冲区的次数占命中总次数的百分比。如果数据显示在清洗部分高速缓存命中百分比很高，这可能意味着清洗区过大。不过，对于只读或写操作次数较低的高速缓存来说，较大的命中数并不是问题。

较大的清洗部分可能会导致物理 I/O 增加，因为在它们通过清洗标记时，Adaptive Server 会对所有脏页启动写操作。如果将清洗区中的页写入到磁盘，然后进行第二次更新，则会浪费 I/O。请检查确认在清洗标记处是否有大量缓冲区正在进行写操作。

有关详细信息，请参见第 112 页的 “Buffer Wash Behavior”。

如果对高速缓存中表的查询将 “读取和放弃”策略用于非 APF I/O，则页的第一次高速缓存命中可在清洗区中找到它。缓冲区移动到链的 MRU 端，因此紧跟第一个高速缓存命中的第二个高速缓存命中将会找到仍在清洗区外的缓冲区。

有关信息，请参见第 112 页的 “Cache Strategy”。

您可以更改清洗大小。如果减小清洗大小，可在完整负载的条件下再次运行 sp_sysmon 并检查值大于 0 的 “Grabbed Dirty”输出

请参见第 104 页的 “Cache Turnover”。

Cache Misses

在高速缓存中未找到需要的页而必须从磁盘读取的次数。“% of total”报告在高速缓存中未找到缓冲区的次数占搜索总次数的百分比。

Total Cache Searches

概述高速缓存搜索活动。“Found in Wash”数据是 “Cache Hits”数值的子类别，在摘要计算中不会使用它。

Pool Turnover

从高速缓存中的每个池替换缓冲区的次数。每个高速缓存可最多有 4 个池， I/O 的大小分别为 2K、 4K、 8K 和 16K。如果存在任何 “Pool Turnover”，则 sp_sysmon 输出每个已配置的池的 “LRU Buffer Grab”和 “Grabbed Dirty”信息和整个高速缓存的总周转数。如果没有任何 “Pool Turnover”，则 sp_sysmon 只为 “Total Cache Turnover”输出零值。

此信息可帮助您确定池和高速缓存的大小是否正确。

LRU Buffer Grab

“LRU Buffer Grab”仅在一页代替另一页时递增。如果最近重新启动过 Adaptive Server，或者刚刚解除对象或数据库与高速缓存的绑定然后又重新建立了绑定，则在计算周转率时不会考虑将页读入到空缓冲区的操作。

如果内存池对于吞吐量来说太小，则可能会得到这样的结果：池中的周转率很高、高速缓存命中率降低以及 I/O 率提高。如果某些池中周转率很高而在其它池中却很低，您也许希望将活动性低的池中的空间移动到活动性高的池中，尤其是在这样可能提高高速缓存命中率的情况下。

如果池有 1000 个缓冲区，而 Adaptive Server 每秒替换 100 个缓冲区，则每秒有 10% 的缓冲区在周转。这可能表示缓冲区在高速缓存中停留的时间不足以使对象有机会使用此高速缓存。

Grabbed Dirty

提供在写入磁盘前到达 LRU 的脏缓冲区数量的统计信息。当 Adaptive Server 需要从高速缓存的 LRU 端争夺一个缓冲区以便从磁盘读取一页，却找到一个脏缓冲区而不是干净缓冲区时，它必须等待脏缓冲区上的 I/O 完成。“% of total”报告脏缓冲区争夺占缓冲区争夺总次数的百分比。

如果 “Grabbed Dirty”是非零值，则表示对于池中的吞吐量而言池的清洗区太小。补救措施取决于池的配置和使用情况：

- 如果池非常小且周转率很高，则应考虑增加池和清洗区的大小。
- 如果池很大，且用于大量的数据更改操作，则应增加清洗区的大小。
- 如果有几个对象使用此高速缓存，则将其中的一些对象移动到其它高速缓存中可能会有所帮助。
- 如果高速缓存正在由 create index 使用，则高 I/O 率可引起脏缓冲区争夺，尤其在较小的 (16K) 池中。在这些情况下，可将池的清洗区大小设置得尽可能高，可达到池中缓冲区的 80%。
- 如果高速缓存已分区，则应减少分区数。
- 检查查询计划和对象的 I/O 统计信息，此对象使用高速缓存进行能够在池中执行许多物理 I/O 的查询。如有可能，通过添加索引对查询进行调优。

请在第 112 页的 “Buffer Wash Behavior”中检查 “Buffers Already in I/O”和 “Buffers Washed Dirty”的 “per second”值。清洗区应足够大以便允许 I/O 在到达 LRU 前能够在脏缓冲区中完成。完成 I/O 需要的时间取决于由磁盘驱动器完成的每秒实际物理写操作次数。

还要检查第 120 页的 “磁盘 I/O 管理”以查看 I/O 争用是否会减慢磁盘写操作。

另外，增加 housekeeper free write percent 配置参数的值也会有所帮助。请参见 《系统管理指南，卷 1》中的第 5 章。

Total Cache Turnover

提供在高速缓存的所有池中发生的缓冲区争夺的总次数。

Cluster Cache Behavior

sp_sysmon 描述为集群环境的缓冲区获取的任何锁。

Buffer Wash Behavior

报告有关缓冲区在到达池的清洗标记时的状态的信息。当缓冲区到达清洗标记时，它处于以下三种状态中的一种：

- “Buffers Passed Clean”报告通过清洗标记时清理干净缓冲区的数量。缓冲区在高速缓存中时不进行修改，或进行了修改，而且已经由管家或检查点写入到磁盘。“% of total”报告清理干净的缓冲区占通过清洗标记的缓冲区总数的百分比。
- “Buffers Already in I/O”报告 I/O 在进入清洗区时在缓冲区中已经处于活动状态的次数。页面在位于高速缓存中时被修改。管家或检查点已经启动该页中的 I/O，但 I/O 还没有完成。“% of total”报告已经在 I/O 中的缓冲区占进入清洗区的缓冲区总数的百分比。
- “Buffers Washed Dirty”报告缓冲区进入已脏的清洗区且尚未在 I/O 中的次数。缓冲区在高速缓存中时已更改且没有写入到磁盘。异步 I/O 通过清洗标记时已在页中启动。“% of total”报告清洗脏了的缓冲区占进入清洗区的缓冲区总数的百分比。

如果在采样间隔期间没有缓冲区通过清洗标记，则 sp_sysmon 输出：

```
Statistics Not Available - No Buffers Entered Wash Section Yet!
```

Cache Strategy

在遵循“读取和放弃”(MRU) 或常规 (LRU) 高速缓存策略的情况下，放置在高速缓存中的缓冲区数量：

- Cached (LRU) Buffers — 报告使用普通高速缓存策略并被放置在高速缓存的 MRU 端的缓冲区的数量。这包括直接从磁盘读取并放置在 MRU 端的所有缓冲区，以及在高速缓存中找到的所有缓冲区。逻辑 I/O 完成时，将缓冲区放置在高速缓存的 MRU 端。
- Discarded (MRU) Buffers — 报告在使用 “读取和放弃”策略的情况下被放置在清洗标记处的缓冲区数量。

  如果希望将整个表进行高速缓存，但发现 “Discarded Buffers”的值很高，可使用 showplan 查看优化程序是否在应该使用常规高速缓存策略时生成 “读取和放弃”策略。

  有关详细信息，请参见 《性能和调优系列：查询处理和抽象计划》中的第 2 章 “使用 showplan”。

Large I/O Usage

提供有关大 I/O 的 Adaptive Server 预取请求的数据。它报告已执行或已拒绝的大 I/O 请求的数量统计信息。

Large I/Os Performed

衡量执行请求的大 I/O 的次数。“% of total”报告执行的大 I/O 请求占已进行的请求总数的百分比。

仅当 Adaptive Server 在高速缓存中执行大 I/O 时， sp_sysmon 才显示该部分。

Large I/Os Denied

大 I/O 无法执行的次数。“% of total”报告拒绝的大 I/O 请求占已进行的请求总数的百分数。

在以下情况下， Adaptive Server 无法执行大 I/O：

- 如果缓冲区中的任何页已经驻留在其它池中。
- 当请求的池中没有任何缓冲区可用时。
- 在分配单元的第一个扩充上，因为它包含分配页，该页始终被读入到 2K 池中。

如果大 I/O 被拒绝的比例较高，则说明较大池的使用情况没有获得预期的效果。如果高速缓存包含大 I/O 池，且查询对相同的对象执行 2K 和 16K 的 I/O，则总会有一定比例的大 I/O 不能执行，因为页在 2K 池中。

如果拒绝的大 I/O 超过半数，且您在使用 16K 的 I/O，则应尝试将所有的空间从 16K 的池中移动到 8K 的池中。重新运行测试以查看 I/O 总数是否已减少。当 16K 的 I/O 被拒绝时，Adaptive Server 并不检查 8K 或 4K 的池，而是使用 2K 的池。

可以使用本类别和 “Pool Turnover”中的信息帮助判断正确的池大小。

仅当 Adaptive Server 在高速缓存中执行大 I/O 时， sp_sysmon 才显示该部分。

Total large I/O Requests

提供执行的和拒绝的大 I/O 的摘要统计信息。

Large I/O Detail

分别提供每个池的摘要信息。它包含在高速缓存中配置的每个 4K、 8K 或 16K 的池的信息块。它输出为已配置的每个 I/O 大小引入的页（“Pages Cached”）和引用的页 （“Pages Used”）。

例如，如果查询执行一个 16K 的 I/O 并读取单个的数据页，则 “Pages Cached”值为 8，“Pages Used”的值为 1。

- “Pages by Lrg I/O Cached”输出读取到高速缓存中的页的总数。
- “Pages by Lrg I/O Used”报告在高速缓存中时由查询使用的页数。

Dirty Read Behavior

报告在隔离级别 0 时请求的平均页数。

“Dirty Read Page Requests”的“% of total”输出显示脏读次数相对于页读取总次数的百分比。

过程高速缓存管理

显示：

- 重新编译存储过程的次数。
- 触发重新编译的阶段：执行时间、重新编译，等等。
- 重新编译的原因：表丢失、权限更改，等等。

样本输出

```
Procedure Cache Management       per sec      per xact       count   % of total
-------------------------  ------------   -----------  ---------   ---------
  Procedure Requests              14.2           0.3       8499         n/a
  Procedure Reads from Disk        0.0           0.0         27        0.6%
  Procedure Writes to Disk         0.0           0.0          0        0.0%
  Procedure Removals               0.0           0.0          0         n/a
  Procedure Recompilations         0.0           0.0          0         n/a

SQL Statement Cache:
  Statements Cached                0.6           2.7        171         n/a
  Statements Found in Cache        2.0           9.5       1529         n/a
```

```
Statements Not Found                0.6             2.7          171          n/a
Statements Dropped                  0.2             0.8           52          n/a
Statements Restored                 0.0             0.0            0          n/a
Statements Not Cached               0.0             0.0            0          n/a
```

过程高速缓存包含所请求的过程的大部分执行计划。在此示例中，过程高速缓存只读取磁盘中 0.6% 的内容。

该系统的语句高速缓存是有效的，因为 “Statements Found in Cache”的值为 611。该值远高于 “Statements Not Found”和 “Statements Cached”的值，因此语句的重用率很高。

Procedure Requests

执行存储过程的次数。

在执行过程时，会发生以下情况之一：

- 内存中有查询计划的空闲副本，因此可进行复制和使用；或者
- 内存中没有过程的副本，或内存中计划的所有副本都在使用，因此必须从磁盘读取过程。

Procedure Reads from Disk

空闲副本不可用且 Adaptive Server 必须从磁盘读取过程（即， Adaptive Server 无法复制已经位于高速缓存中的内容）的次数。

“% of total”报告从磁盘读取的过程占过程请求总数的百分比。如果该百分比相对较高，可能表示过程高速缓存太小。

Procedure Writes to Disk

间隔期间创建的过程数量。如果应用程序生成存储过程，则该值会很大。

Procedure Removals

过程在高速缓存中老化的次数。

Procedure Recompilations

重新编译存储过程的次数。

SQL Statement Cache

在间隔期间发生以下情况的语句数量：被高速缓存、在高速缓存中找到、未找到、删除、恢复以及未高速缓存。

样本输出

```
SQL Statement Cache:
Statements Cached                0.6            2.7          171       n/a
Statements Found in Cache        2.0            9.5          611       n/a
Statements Not Found             0.6            2.7          171       n/a
Statements Dropped               0.2            0.8           52       n/a
Statements Restored              0.0            0.0            0       n/a
Statements Not Cached            0.0            0.0            0       n/a
```

- Statements Cached — 在采样间隔期间添加到语句高速缓存中的语句的数量。
- Statements Found in Cache — 在语句高速缓存中找到的语句的数量。该值表示 Adaptive Server 使用语句高速缓存以避免重新编译语句的次数。
- Statements Not Found — Adaptive Server 在语句高速缓存中查找语句但未找到的次数。
- Statements Dropped — Adaptive Server 删除已经位于语句高速缓存中的语句以便为要高速缓存的新语句腾出空间的次数。
- Statements Restored — 由于更改语句所引用的对象的模式而使现有计划失效后恢复的查询计划的数量。
- Statements Not Cached — Adaptive Server 已编译但未进入语句高速缓存的语句的数量。如果将 statement cache size 设置为 0（禁用语句高速缓存），“Statements Not Cached”等于 Adaptive Server 本应高速缓存的语句的数量。

内存管理

采样间隔期间分配和取消分配的页数。

样本输出

```
Memory Management              per sec       per xact       count   % of total
-------------------------   ------------   -----------    --------    --------
Pages Allocated                 2565.7          63.2     1539423         n/a
Pages Released                  2565.7          63.2     1539423         n/a
```

Pages Allocated

在内存中分配新页的次数。

Pages Released

释放某页的次数。

恢复管理

此数据显示普通检查点进程引起的检查点数量、由管家任务启动的检查点数量和每种类型的平均时间长度。使用此信息正确地设置恢复参数和管家参数。

注释　如果使用的是 Adaptive Server 12.5.3 或更高版本，内部基准会指出检查点任务的执行速度最多可比以前快 200%，从而大大提高了数据库恢复速度。速度的这种提高不要求您执行任何操作。

但是，性能是否提高取决于 I/O 子系统的效力。在 IO 带宽可用之前，您可能看不到这些好处。

样本输出

```
================================================================================

Recovery Management
-------------------
Checkpoints                    per sec     per xact    count    % of total
-----------------------    -----------   ----------  -------      --------
# of Normal Checkpoints            0.2          0.0      106        100.0 %
# of Free Checkpoints              0.0          0.0        0          0.0 %
-----------------------  ------------   ----------  -------
Total Checkpoints                  0.2          0.0      106
Average Time per Normal Chkpt
```

Checkpoints

检查点将脏页 （已在内存中修改但还没有写入到磁盘中的页）写入到数据库设备中。 Adaptive Server 自动 （常规）检查点机制会发挥作用以保持最小恢复间隔。通过跟踪自执行最后检查点起的事务日志中的日志记录数，该机制可以估计出恢复事务需要的时间是否超出了恢复间隔。如果是这样，则检查点进程扫描所有数据高速缓存并写出所有已更改的数据页。

当 Adaptive Server 没有要处理的用户任务时，管家清洗任务会开始将脏缓冲区写至磁盘。因为这些写入在服务器空闲周期内进行，所以称为 “自由写入”。它们可提高 CPU 利用率并降低事务处理过程中冲洗缓冲区的需求。

of Normal Checkpoints

由常规检查点进程执行的检查点的数目。

如果常规检查点执行大多数工作，尤其是如果检查点需要的时间很长，则增加由管家清洗任务执行的写操作的次数可能会有所帮助。

有关配置参数的详细信息，请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

of Free Checkpoints

管家清洗任务执行的检查点数。管家清洗任务仅在从所有配置的高速缓存中清除了所有脏页后才执行检查点。

如果管家清洗任务完成了将所有高速缓存中的所有脏页写至磁盘的操作，则它会检查事务日志中自上一个检查点以后的行数。如果日志记录超过 100 行，则发出一个检查点。这称为 “自由检查点”，因为它需要的开销非常少。另外，它减少了常规检查点的未来开销。

您可以使用 housekeeper free write percent 参数配置管家清洗任务可将数据库写操作增加的最大百分比。请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Total Checkpoints

报告在采样间隔期间发生的常规和自由检查点的总数量。

Average Time per Normal Checkpoint

报告常规检查点持续的平均时间。

Average Time per Free Checkpoint

报告自由 （或管家）检查点持续的平均时间。

Increasing the Housekeeper Batch Limit

管家清洗任务具有内置批处理限制，可避免单个设备的磁盘 I/O 过载。缺省情况下，管家写操作的批处理大小设置为 3。管家检测到已经向单个设备发出 3 个 I/O 后立即停止当前缓冲池中的处理操作，并开始检查其它池中的脏页。如果来自下一个池的写操作被分配到同一设备，则它会移动到另一个池。管家检查所有池后，它开始等待直到由它发出的最后一个 I/O 完成，然后再次开始该循环。

缺省批处理限制是为速度较慢的磁盘提供良好的设备 I/O 特性而设计的。通过增加快速磁盘驱动器的批量大小，您会获得较好的性能。可以为服务器上的所有设备全局设置此限制，也可以为拥有不同速度的磁盘设置不同的值。每次 Adaptive Server 重新启动时，都必须重新设置此限制。

此命令通过使用 sysdevices 的虚拟设备号将单个设备的批量大小设置为 10：

```
dbcc tune(deviochar, 8, "10")
```

若要查看设备号，请使用 sp_helpdevice 或：

```
select name, vdevno
from sysdevices
```

若要更改服务器上所有设备的管家批量大小，请使用 -1 代替设备号：

```
dbcc tune(deviochar, -1, "5")
```

对于非常快速的设备，如果将批量大小设置为 50 那么高，则测试期间会提高性能。

在下列情况下，您可能希望尝试将批量大小设置得更高：

- 常规检查点的平均时间很长。
- 超出 I/O 配置限制或出现对设备信号的争用不会出现任何问题。

磁盘 I/O 管理

报告磁盘 I/O。它概述整个服务器的磁盘 I/O 活动，并报告每个逻辑设备的读、写和信号争用。

样本输出

```
Disk I/O Management
-------------------

Max Outstanding I/Os            per sec      per xact       count   % of total
------------------------    -----------   -----------   ---------   ---------
  Server                             n/a           n/a         527         n/a
  Engine 0                           n/a           n/a         429         n/a
  Engine 1                           n/a           n/a         448         n/a
  Engine 2                           n/a           n/a         457         n/a
  Engine 3                           n/a           n/a         508         n/a
  Engine 4                           n/a           n/a         526         n/a
  Engine 5                           n/a           n/a         416         n/a
  Engine 6                           n/a           n/a         425         n/a


I/Os Delayed by
  Disk I/O   Structures              n/a           n/a           0         n/a
  Server Config   Limit              n/a           n/a           0         n/a
  Engine Config Limit                n/a           n/a           0         n/a
```

```
  Operating System Limit              n/a           n/a           0          n/a


Total Requested Disk I/Os           547.2          13.5      328339

Completed Disk I/O's
Asynchronous I/O's
  Engine 0                          135.0           3.3       81017       24.7 %
  Engine 1                           70.3           1.7       42172       12.8 %
  Engine 2                           50.1           1.2       30083        9.2 %
  Engine 3                           68.0           1.7       40789       12.4 %
  Engine 4                           72.5           1.8       43473       13.2 %
  Engine 5                           81.2           2.0       48749       14.8 %
  Engine 6                           70.1           1.7       42070       12.8 %
Synchronous I/O's
  Total Completed I/Os                0.0           0.0           0          n/a
------------------------   ------------   -----------   ---------
Total Completed I/Os                547.3          13.5      328353


Device Activity Detail
----------------------

Device:
  /dev/sybase/c1t0d0s0r
  master                             per sec      per xact       count   % of total
------------------------    -----------   -----------   ---------   ---------
  Reads
    APF                               0.0           0.0           2        0.2 %
    Non-APF                           0.3           0.0         154       17.5 %
  Writes                              1.2           0.0         723       82.3 %
-------------------------    -----------   -----------   ---------   ---------
Total I/Os                            1.5           0.0         879        0.3 %


--------------------------------------------------------------------------------

Device:
  /dev/sybase/c1t0d0s1r
  dev01                              per sec      per xact       count   % of total
------------------------    -----------   -----------   ---------   ---------
Reads
  APF                                 1.0           0.0         584        0.5 %
    Non-APF                           8.5           0.2        5075        4.3 %
    Writes                          185.6           4.6      111355       95.2 %
------------------------   -----------   -----------   ---------   ---------
```

```
Total I/Os                          195.0           4.8       117014       35.6 %

------------------------------------------------------------------------------

Device:
  /dev/sybase/c1t0d0s3r
  dev02                           per sec      per xact        count   % of total
------------------------    -----------   -----------   ---------   ---------
  Reads
    APF                             0.0           0.0           1        0.5 %
    Non-APF                         0.0           0.0           5        2.4 %
  Writes                            0.3           0.0         204       97.1 %
-------------------------   -----------   -----------   ---------   ---------
Total I/Os                          0.4           0.0         210        0.1%
```

Maximum Outstanding I/Os

整个 Adaptive Server 待执行的 I/O 最大数量（第一行），以及每个 Adaptive Server 引擎在采样间隔期间的任一时刻待执行 I/O 的最大数量。

如果任何 “I/Os Delayed By” 都非零，则此信息可帮助配置服务器或操作系统级的 I/O 参数。

I/Os Delayed by

当系统出现 I/O 延迟问题时，很可能是因为有一个或多个 Adaptive Server 或操作系统限制阻塞了 I/O。

大多数的操作系统都有限制可以发生的异步 I/O 数量的内核参数。

Disk I/O Structures

因达到磁盘 I/O 结构的限制而延迟的 I/O 数量。当 Adaptive Server 超出可用磁盘 I/O 控制块的数量时，I/O 会被延迟，因为 Adaptive Server 要求任务在启动 I/O 请求前获得磁盘 I/O 控制块。

如果结果是非零值，可尝试通过增加配置参数 disk i/o structures 来增加可用的磁盘 I/O 控制块的数量。请参见《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Server Configuration Limit

Adaptive Server 可超出异步磁盘 I/O 请求数量的限制，而这些请求对于整个 Adaptive Server 来说可能同时处于未完成状态。您可以使用 max async i/os per server 配置参数提高此限制。请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Engine Configuration Limit

引擎可以超出未完成异步磁盘 I/O 请求的限制。您可以使用 max async i/os per engine 配置参数更改此限制。请参见 《系统管理指南，卷 1》中的第 5 章 “设置配置参数”。

Operating System Limit

采样间隔期间超出操作系统对未完成的异步 I/O 的限制的次数。操作系统内核可以限制进程或整个系统可以在任何一个时间具有待执行状态的异步 I/O 最大数量。请参见 《系统管理指南》；还请参见您的操作系统文档。

请求和完成的磁盘 I/O

此数据显示请求的磁盘 I/O 总数，以及每个 Adaptive Server 引擎完成的 I/O 数量和百分比。

“Total Requested Disk I/Os”和 “Total Completed I/Os”应相同或非常接近。如果已请求的 I/O 由于饱和而没有完成，则这些值差别会很大。

请求的 I/O 值包括在采样间隔期间启动的所有请求，而且其中可能会有一些请求是在采样间隔结束后完成的。这些 I/O 不包括在 “Total Completed I/Os”中，并且会在没有饱和问题时使百分比低于 100。

反过来也一样。如果 I/O 请求在采样间隔开始前提出并在此期间完成，则您会发现 “Total Completed I/Os”值的 “% of total”大于 100%。

如果数据显示存在大量的已请求的磁盘 I/O，而已完成的磁盘 I/O 数量相对较小，则可能是操作系统中的瓶颈延迟了 I/O。

Total Requested Disk I/Os

Adaptive Server 请求磁盘 I/O 的次数。

Completed Disk I/Os

每个引擎完成 I/O 的次数。“% of total”报告每个引擎完成 I/O 的次数占所有 Adaptive Server 引擎完成 I/O 的总次数的百分比。

在线程模式中，Adaptive Server 不以每个引擎为基础输出完成的磁盘 I/O，而是输出全服务器范围内的摘要：

```
Disk I/O Management
-------------------

 Max Outstanding I/Os             per sec      per xact       count  % of total
 -------------------------  ------------  ------------  ----------  ---------
  Server                            n/a           n/a          17        n/a
  Engine 0                          n/a           n/a           5        n/a
  Engine 1                          n/a           n/a           5        n/a
  Engine 2                          n/a           n/a          17        n/a

 I/Os Delayed by
  Disk I/O Structures               n/a           n/a           0        n/a
  Server Config Limit               n/a           n/a           0        n/a
  Engine Config Limit               n/a           n/a           0        n/a
  Operating System Limit            n/a           n/a          20        n/a

 Total Requested Disk I/Os         12.6         104.3         417

 Completed Disk I/O's
  Asynchronous I/O's
   Total Completed I/Os            12.6         104.3          41      100.0 %
  Synchronous I/O's
   Total Completed I/Os             0.0           0.0           0        n/a
 -------------------------  ------------  ------------  ----------
 Total Completed I/Os              12.6          104.3         417
```

也可使用此信息确定操作系统是否能够与所有引擎发出的磁盘 I/O 请求保持同步。

Device Activity Detail

报告每台逻辑设备上的活动。它有助于检查数据库设备间的 I/O 是否保持良好的平衡，以及帮助查找可能会延迟 I/O 的设备。例如，如果“Task Context Switches Due To”数据指出设备争用的数量极大，则可以使用“Device Activity Detail”找出引起问题的设备。

此部分输出有关服务器上每个数据设备的 I/O 的以下信息：

- 逻辑和物理设备名称
- 读和写操作的次数和 I/O 总数
- 设备上立即授予的设备信号请求数，以及进程被迫等待设备信号的次数

读和写操作

在设备上进行读或写操作的次数。“Reads”报告异步预取读取的页数和其它 I/O 活动引入高速缓存的页数。“% of total”列报告读或写操作占设备 I/O 总数的百分比。

Total I/Os

报告对设备的读和写的总数。“% of total”列是每个命名设备的读和写的总数占所有设备的读和写的总数的百分比。

您可以使用此信息检查磁盘上 I/O 分布模式，以及决定能够帮助平衡所有设备中的磁盘 I/O 的对象放置情况。例如，此数据可能指示某些磁盘比其它磁盘使用得更频繁。如果发现特定的命名设备的所有 I/O 占的比例较大，您可以考察驻留在该设备中的表，然后确定解决此问题的办法。

请参见 《性能和调优系列：物理数据库调优》中的第 1 章 “控制物理数据放置”。

Mirror Semaphore Granted 和 Mirror Semaphore Waited

立即授予镜像信号请求的次数以及信号正忙而任务被迫等待信号释放的次数。“% of total”列是将信号授予设备 （或任务被迫等待）的次数占镜像信号请求的总次数的百分比。此数据只在 SMP 环境中有意义。

仅当启用磁盘镜像或者 I/O 被延迟时， Adaptive Server 才使用信号。更改该配置参数的缺省值 “disabled”后，即可启用磁盘镜像。

处于等待状态的 I/O 请求的比例较大说明存在信号争用问题。您可能需要尝试在物理设备上重新分布数据。

网络 I/O 管理

报告每个 Adaptive Server 引擎的以下网络活动：

- 已请求的网络 I/O 总数
- Network I/Os delayed
- 接收和发送的 Tabular Data Stream (TDS) 包和字节总数。
- 接收和发送的信息包的平均大小

此数据按引擎进行划分，因为每个引擎都处理它自己的网络 I/O。不平衡的现象通常由下列情况之一引起：

- 引擎多于任务，因此没有工作可执行的引擎报告无 I/O，或
- 大多数任务发送和接收短信息包，但另一任务执行工作量大的 I/O，例如，批量复制。

样本输出

```
Network I/O Management
----------------------

  Network I/O Requests             per sec      per xact       count % of total
  -------------------------  ------------   -----------    --------  --------
  Total Network I/O Requests       1384.2         938.4      166099       n/a
  Network I/Os Delayed                0.0           0.0           0     0.0 %

  Network Receive Activity        per sec      per xact       count
  -------------------------  ------------  ------------  ----------
  Total TDS Packets Rec'd          1380.7         936.1      165684
  Total Bytes Rec'd             2267752.3     1537459.1   272130277
  Avg Bytes Rec'd per Packet          n/a           n/a        1642

  Network Send Activity           per sec      per xact       count
  -------------------------  ------------   -----------  ----------
  Total TDS Packets Sent           1339.5         908.2      160743
  Total Bytes Sent                56194.2       38097.8     6743306
  Avg Bytes Sent per Packet           n/a           n/a          41
```

Total Network I/Os Requests

报告已发送和接收的包的总数。

如果知道网络每秒可处理的包数，就可以确定网络带宽是否制约了 Adaptive Server。

无论 I/O 属于入站还是出站，此问题都是一样的。如果 Adaptive Server 接收到一个大于包大小的命令，则它将等待，直到它接收到完整的命令才开始处理。因此，需要多个信息包的命令执行速度较慢，且占用的 I/O 资源也较多。

如果每个信息包的平均字节数接近为服务器配置的缺省信息包大小，则可能需要为某些连接配置较大的信息包大小。您可以为所有连接配置网络包大小，或允许某些连接使用较大的包大小登录。

请参见 《性能和调优系列：基础知识》中的第 2 章 “网络和性能”。

Network I/Os Delayed

I/O 被延迟的次数。如果该数值始终是非零值，则应向网络管理员咨询。

Total TDS Packets Received

每个引擎接收到的 TDS 包数。“Total TDS Packets Rec'd”报告在采样间隔期间接收到的信息包数。

Total Bytes Received

报告每个引擎接收到的字节数。“Total Bytes Rec'd”报告在采样间隔期间接收到的字节总数。

Average Bytes Received per Packet

在采样间隔期间接收到的所有包的平均字节数。

Total TDS Packets Sent

每个引擎发送的包数，以及整个服务器发送的包的总数。

Total Bytes Sent

在采样间隔期间由每个 Adaptive Server 引擎发送的字节数，以及整个服务器发送的字节数。

Average Bytes Sent per Packet

报告在采样间隔期间发送的所有包的平均字节数。

Replication Agent

Replication Agent 组显示配置为使用 Replication Agent 的所有数据库的 Replication Agent 特定的活动。

样本输出

```
Replication Agent

---------------

Replication Agent:primdb

Replication Server:MRP_REPSRVR1
                                per sec      per xact       count % of total
                             ------------   -----------    --------  --------

 Log Scan Summary
   Log Records Scanned               n/a            n/a      710273       n/a
   Log Records Processed             n/a            n/a      127098       n/a
   Number of Log Scans               n/a            n/a           0       n/a
   Amount of Time for Log Scans (ms) n/a            n/a       19868       n/a
   Longest Time for Log Scan (ms)    n/a            n/a       19868       n/a
   Average Time per Log Scan (ms)    n/a            n/a         0.0       n/a
```

```
Log Scan Activity
  Updates                          n/a             n/a               0        n/a
  Inserts                          n/a             n/a           66380        n/a
  Deletes                          n/a             n/a               0        n/a
  Store Procedures                 n/a             n/a               2        n/a
  DDL Log Records                  n/a             n/a               0        n/a
  Writetext Log Records            n/a             n/a               0        n/a
  Text/Image Log Records           n/a             n/a           60420        n/a
  CLRs                             n/a             n/a               0        n/a
  Checkpoints Processed            n/a             n/a               4        n/a
  SQL Statements Processed         n/a             n/a               0        n/a

Transaction Activity
  Opened                           n/a             n/a             140        n/a
  Commited                         n/a             n/a             141        n/a
  Aborted                          n/a             n/a               1        n/a
  Prepared                         n/a             n/a               1        n/a
  Delayed Commit                   n/a             n/a               0        n/a
  Maintenance User                 n/a             n/a               4        n/a

Log Extension Wait
  Count                            n/a             n/a               2        n/a
  Amount of time (ms)              n/a             n/a           14346        n/a
  Longest Wait (ms)                n/a             n/a           14346        n/a
  Average Time (ms)                n/a             n/a          7173.0        n/a

Schema Cache
 Usage
  Max Ever Used                    n/a             n/a               4        n/a
  Schemas reused                   n/a             n/a               0        n/a
 Forward Schema Lookups
  Count                            n/a             n/a               0        n/a
  Total Wait (ms)                  n/a             n/a               0        n/a
  Longest Wait (ms)                n/a             n/a               0        n/a
  Average Time (ms)                n/a             n/a             0.0        n/a

Backward Schema Lookups
  Count                            n/a             n/a               0        n/a
  Total Wait (ms)                  n/a             n/a               0        n/a
  Longest Wait (ms)                n/a             n/a               0        n/a
  Average Time (ms)                n/a             n/a             0.0        n/a

Truncation Point Movement
 Moved                             n/a             n/a               0        n/a
 Gotten from RS                    n/a             n/a               0        n/a

Connections to Replication Server
```

```
  Success                          n/a            n/a             0        n/a
  Failed                           n/a            n/a             0        n/a

Network Packet Information
    Packets Sent                   n/a            n/a         39429        n/a
    Full Packets Sent              n/a            n/a          4381        n/a
    Largest Packet                 n/a            n/a             0        n/a
    Amount of Bytes Sent           n/a            n/a      73240036        n/a
    Average Packet                 n/a            n/a        1857.5        n/a

I/O Wait from RS
    Count                          n/a            n/a         48191        n/a
    Amount of Time (ms             n/a            n/a         58995        n/a
    Longest Wait (ms)              n/a            n/a            17        n/a
    Average Wait (ms)              n/a            n/a           1.2        n/a
```

Log Scan Summary

Replication Server 的日志活动的摘要。Replication Agent 会扫描它在其中运行的数据库的事务日志，然后将扫描的信息发送给 Replication Server。

主数据库的事务日志包含 Replication Agent 不处理的修改，因为这些修改属于未标记为要复制的对象。

Log Records Scanned

报告 Replication Agent 扫描的日志记录总数。

Log Records Processed

报告因为包含标记为要复制的对象而被发送到 Replication Server 的日志记录总数。

Number of Log Scans

Rep Agent 扫描日志的总次数。

Amount of Time for Log Scans

Rep Agent 在日志扫描上花费的时间 （以毫秒为单位）。

Longest Time for Log Scan

Rep Agent 在日志扫描上花费的最长时间 （以毫秒为单位）。

Average Time per Log Scan

Rep Agent 在日志扫描上花费的平均时间 （以毫秒为单位）。

Log Scan Activity

报告日志中的以下活动：

- Updates — 处理的更新数量。
- Inserts — 处理的插入数量。
- Deletes — 处理的删除数量。
- Store Procedures — 标记为要复制的已执行存储过程的数量。
- DDL Log Records — 包含要复制的数据定义语言 (DDL) 的日志记录的数量。该值可能与主数据库中执行的 DDL 命令数量不对应，因为在主数据库上执行的单个 DDL 命令可能产生多个 DDL 命令日志记录。
- Writetext Log Records — 由 writetext 生成且已经处理的日志记录的数量。
- Text/Image Log Records — 具有标记为要复制的 text 或 image 数据的表的数据操纵语言 (DML) 日志记录的数量。
- CLRs — 已处理的补偿日志记录 (CLR) 的数量。
- Checkpoints Processed — 执行的检查点数量。
- SQL Statements Processed — 已处理的 SQL 语句的数量。

Transaction Activity

报告以下活动：

- Opened — 在主日志中找到的 begin transaction 命令的数量。
- Committed — 已提交事务的数量。
- Aborted — 已中止事务的数量。
- Prepared — 处于 “准备” 状态的事务数量。

- Delayed Commit — delayed commit 事务的数量
- Maintenance User — 维护用户打开的事务数量。

Log Extension Wait

在正常处理过程中，当 Replication Agent 到达事务日志末尾时，它必须等待 Adaptive Server 扩展日志 （以便主数据库能够继续执行活动）。

“Log Extension Wait”报告以下活动：

- Count — Replication Agent 等待 Adaptive Server 扩展日志的次数。
- Amount of time — Replication Agent 等待 Adaptive Server 扩展日志所花费的时间 （以毫秒为单位）。
- Longest Wait — Replication Agent 等待 Adaptive Server 扩展日志所花费的最长时间 （以毫秒为单位）。
- Average Time — Replication Agent 等待 Adaptive Server 扩展日志所花费的平均时间 （以毫秒为单位）。

Schema Cache

- Max Ever Used — 模式高速缓存中的最大模式数。
- Schemas Reused — 处理过程中重用的模式数。

“Schema Cache”报告以下活动：

- Forward Schema — 有时， Replication Agent 必须对主事务日志执行正向扫描以查找对象模式更改。
- Count — Replication Agent 执行正向扫描的次数。
- Total Wait — Replication Agent 执行正向扫描所花费的时间 （以毫秒为单位）。
- Longest Wait — Replication Agent 执行正向扫描所花费的最长时间 （以毫秒为单位）。
- Average Time — Replication Agent 执行正向扫描所花费的平均时间 （以毫秒为单位）。

Backward Schema

当 Adaptive Server 在事务内执行 DDL， Replication Agent 必须在主事务日志中执行反向扫描。“Backward Schema”报告以下活动：

- Count — Replication Agent 执行反向扫描的次数。
- Total Wait — Replication Agent 执行反向扫描所花费的时间 （以毫秒为单位）。
- Longest Wait — Replication Agent 执行反向扫描所花费的最长时间 （以毫秒为单位）。
- Average Time — Replication Agent 执行反向扫描所花费的平均时间 （以毫秒为单位）。

Truncation Point Movement

报告以下活动：

- Moved — Replication Agent 在主数据库上移动辅助截断点的次数。
- Gotten from RS — Replication Agent 查询 Replication Server 以获取新的辅助截断点的次数。

Connections to Replication Server

报告以下活动：

- Success — 与 Replication Server 的成功连接数。
- Failed — 与 Replication Server 的失败连接数。

Network Packet Information

报告以下活动：

- Packets Sent — 发送到 Replication Server 的包数。
- Full Packets Sent — 发送到 Replication Server 的满包数。
- Largest Packet — 发送到 Replication Server 的最大包。
- Amount of Bytes Sent — 发送到 Replication Server 的字节数。
- Average Packet — 平均包大小。

I/O Wait from RS

报告：

- Count — Replication Agent 向 Replication Server 发布 ct_sendpassthru 的次数。
- Amount of Time — Replication Agent 处理来自 Replication Server 的结果所花费的时间 （以毫秒为单位）。
- Longest Wait — Replication Agent 处理结果所花费的最长时间 （以毫秒为单位）。
- Average Wait — Replication Agent 处理结果所花费的平均时间 （以毫秒为单位）

索引

英文

B

C

D

F

G

H

J

K

L

N

P

R

S

T

W

X

Y

Z