SYBASE®
An SAP® Company

性能和调优系列：
利用统计分析改进性能

Adaptive Server® Enterprise

15.7

文档 ID：DC01067-01-1570-01

最后修订日期： 2011 年 9 月

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

目录

第 1 章

使用 *set statistics* 命令

set 命令包括 statistics 参数，后者包含显示性能统计信息的选项。所有统计信息选项的缺省设置均为 off。

set 命令语法

set statistics 的语法为：

```
set statistics {io, simulate, subquerycache, time, plancost} [on | off]
```

可以发出一个参数：

```
set statistics io on
```

可以通过使用逗号分隔命令将多个命令合并到一行：

```
set statistics io, time on
```

使用模拟统计信息

通过 optdiag 实用程序，您可以装载模拟统计信息并使用那些统计信息执行查询诊断。您甚至可以对空表装载模拟统计信息，这使您甚至可以在仅包含表和索引的小数据库中执行调优诊断。装载模拟统计信息时，它们不会覆盖现有统计信息。

装载模拟统计信息后，请通过输入以下内容来指示优化程序使用它们（而不是实际的统计信息）：

```
set statistics simulate on
```

请参见第 40 页的“使用模拟统计信息”。

检查子查询高速缓存性能

如果未展平或实现子查询，则会创建子查询高速缓存以存储之前执行子查询的结果。这会减少开销非常大的子查询执行次数。

查看编译和执行时间

set statistics time 显示有关分析和执行 Adaptive Server 命令所花时间的信息。

```
Parse and Compile Time 57.
Adaptive Server cpu time:5700 ms.

Execution Time 175.
Adaptive Server cpu time:17500 ms.  Adaptive Server elapsed time:70973 ms.
```

此输出的含义是：

- Parse and Compile Time — 用于分析、优化和编译查询的 CPU 时钟周期数。有关将时钟周期转换为毫秒的信息，请参见下文。
- Adaptive Server® cpu time — 以毫秒为单位的 CPU 时间。
- Execution Time — 用来执行查询的 CPU 时钟周期数。
- Adaptive Server cpu time — 用来执行查询的 CPU 时钟周期数，已转换为毫秒。

- Adaptive Server elapsed time — 从命令开始到当前时间（从操作系统时钟上得到）的时间差（以毫秒为单位）。

此输出显示查询是在 57 个时钟周期内分析和编译的。执行过程花费了 175 个时钟周期，或 17.5 秒的 CPU 时间。经历的总时间是 70.973 秒，表明 Adaptive Server 花费了一些时间处理其它任务，或等待磁盘或网络 I/O 结束。

注释　如果查询比较复杂，需要很长时间来运行，请检查是否由于次优计划而导致执行时间很长，或者由于优化性能而导致分析和编译时间很长。

将时钟周期转换成毫秒

若要将时钟周期转换为毫秒，请使用：

$$\text{Milliseconds} = \frac{\text{CPU\_ticks} * \text{clock\_rate}}{1000}$$

若要查看系统的 *clock_rate*，请执行：

```
sp_configure "sql server clock tick length"
```

请参见《系统管理指南，卷 1》中的第 5 章“设置配置参数”。

报告物理和逻辑 I/O 统计信息

set statistics io 报告有关物理和逻辑 I/O 的信息以及表的访问次数。set statistics io 输出位于查询结果的后面，并提供查询执行的实际 I/O。

对于查询中的每个表（包括工作表），statistics io 将报告带有查询所读取页的几个值的一行信息，以及报告写入总数的一行信息。如果系统管理员启用了资源限制，则 statistics io 还包括一行信息，报告查询的实际总 I/O 开销。下面的示例显示了查询的 statistics io 输出，其中已启用了资源限制：

```
select avg(total_sales)
from titles
Table:titles  scan count 1,  logical reads:(regular=656 apf=0 total=656),
physical reads:(regular=444 apf=212 total=656),  apf IOs used=212
Total actual I/O cost for this command:13120.
Total writes for this command:0
```

以下部分描述了 statistics io 输出的四个主要组成部分：

- 实际 I/O 开销
- 总写入数
- 读取统计信息
- 表名和“扫描计数”

总的实际 I/O 开销值

如果启用资源限制，则 statistics io 会输出“Total actual I/O cost”行。Adaptive Server 会将实际的总 I/O 报告为一个无单位的数值。确定查询开销的公式为：

开销 = 所有物理 IO X 25 + 所有逻辑 IO X 2 + CPU 开销 X 0.1

公式中将逻辑 I/O 的“开销”与逻辑 I/O 数相乘，物理 I/O 的“开销”与物理 I/O 数相乘。

例如：

```
Table:sysmessages scan count 1, logical
reads:(regular=454 apf=0 total=454), physical
reads:(regular=441 apf=0 total=441), apf IOs used=10
Total actual I/O cost for this command:11934
```

即：441*25 + 454*2 + 10*0.1 = 11934

表 1-1 介绍了“regular”和“apf”读取次数。

写入数的统计信息

statistics io 报告由命令写入的缓冲区总数。当只读查询使脏页移过高速缓存中的清洗标记以便在此页开始写入时，只读查询将报告写入。

更改数据的查询可能仅报告一种写入，即日志页的写入，因为已更改的页保留在数据高速缓存的 MRU 部分。

读取的统计信息

statistics io 报告包括在查询（包括工作表）中每个表和索引的逻辑读取数和物理读取数。表 I/O 中包含索引 I/O。

表 1-1 显示了 statistics io 所报告的逻辑和物理读取的值。

表 1-1：读取的 statistics io 输出

输出	说明
logical reads	
regular	在高速缓存内找到查询所需的某一页的次数；此处只对不是由异步预取 (APF) 引入的页进行计数
apf	在高速缓存内找到通过 APF 请求所引入请求的次数
total	*regular* 和 *apf* 逻辑读取次数之和
physical reads	
regular	通过常规异步 I/O 将缓冲区引入高速缓存的次数
apf	由 APF 将缓冲区引入到高速缓存的次数
total	*regular* 和 *apf* 物理读取次数之和
apf IOs used	由 APF 引入的缓冲区数，在查询过程中用到这些缓冲区中的一页或多页

有索引和没有索引的输出示例

使用 statistics io 对无索引的表执行查询及对有索引的表执行相同的查询将显示好的索引对查询和系统性能的重要性。以下是一个查询的示例：

```
select title
    from titles
where title_id = "T5652"
```

没有索引的 *statistics io*

在 title_id 上无索引时，statistics io 将使用 2K I/O 报告下列值：

```
Table:titles scan count 1, logical reads:(regular=624
apf=0 total=624), physical reads:(regular=230 apf=394
total=624), apf IOs used=394
Total actual I/O cost for this command:12480.
Total writes for this command:0
```

此输出显示了：

- 查询总共执行了 624 个逻辑 I/O，均为常规逻辑 I/O。
- 查询执行了 624 个物理读取。其中，230 个是常规异步读取，394 个是异步预取读取。
- 由 APF 读入的所有页均由查询使用。

有索引的 *statistics io*

在 title_id 上具有聚簇索引时，statistics io 对相同的查询将报告下列值，也使用 2K I/O：

```
Table:titles  scan count 1,  logical reads:(regular=3 apf=0 total=3),
physical reads:(regular=3 apf=0 total=3),  apf IOs used=0
Total actual I/O cost for this command:60.
Total writes for this command:0
```

此输出显示了：

- 查询执行了 3 个逻辑读取。
- 查询执行了 3 个物理读取：2 个读取针对索引页，1 个读取针对数据页。

游标的 *statistics io* 输出

对于使用游标的查询，statistics io 显示自游标打开时的累积 I/O：

```
                    1> open c
Table:titles scan count 0, logical reads:(regular=0 apf=0 total=0), physical
reads:(regular=0 apf=0 total=0), apf IOs used=0
Total actual I/O cost for this command:0.
Total writes for this command:0
                    1> fetch c


title_id type         price
 -------- ------------ -------------------------
 T24140   business                       201.95
Table:titles scan count 1, logical reads:(regular=3 apf=0 total=3), physical
reads:(regular=0 apf=0 total=0), apf IOs used=0
Total actual I/O cost for this command:6.
Total writes for this command:0
                    1> fetch c

title_id type         price
 -------- ------------ -------------------------
 T24226   business                       201.95
Table:titles scan count 1, logical reads:(regular=4 apf=0 total=4), physical
reads:(regular=0 apf=0 total=0), apf IOs used=0
Total actual I/O cost for this command:8.
Total writes for this command:0
```

扫描计数

statistics io 报告查询访问特定表的次数。“扫描”可以代表下列任何一种访问方法：

- 表扫描。
- 通过聚簇索引的访问。每次查询从索引的根页开始，然后是数据页的指针，将它计为一次扫描。
- 通过非聚簇索引的访问。每次查询从索引的根页开始，然后是指向索引的叶级的指针（对于覆盖的查询而言）或数据页的指针，将它计为一次扫描。
- 如果查询以并行方式运行，对表的每个工作进程访问作为一次扫描。

请参见 Performance and Tuning Series: Query Processing（《性能和调优系列：查询处理》）中的第 2 章“Using showplan”（使用 showplan）。

报告扫描计数为 1 的查询

返回扫描计数为 1 的查询示例如下：

- 点查询：

```
select title_id
    from titles
    where title_id = "T55522"
```

- 范围查询：

```
select au_lname, au_fname
    from authors
    where au_lname > "Smith"
    and au_lname < "Smythe"
```

如果对这些查询的 where 子句中的列编制索引，则这些查询可使用这些索引来扫描表；否则，它们会执行表扫描。在任何一种情况下，查询都只需要对表进行一次扫描，以返回所需的行。

报告扫描计数超过 1 的查询

返回较大扫描计数值的查询示例如下：

- 对每一工作进程都报告扫描的并行查询。
- 查询中含有由 or 连接的已被索引的 where 子句时，此查询将对每一个 or 子句报告一个扫描。如果查询使用特殊的 or 策略，则它会为每个值报告一次扫描。如果查询使用 or 策略，则它会为每个索引报告一次扫描，另外为 RID 列表访问报告一次扫描。

  如果 titles 表中有按 title_id 编制的索引和按 pub_id 编制的索引，则以下查询报告的扫描计数将为 2：

```
    select title_id
        from titles
        where title_id = "T55522"
        or pub_id  = "P988"
Table:titles scan count 2,logical reads:(regular=149 apf=0 total=149),
physical reads:(regular=63 apf=80 total=143), apf IOs used=80
Table:Worktable1  scan count 1, logical reads:(regular=172 apf=0
total=172), physical reads:(regular=0 apf=0 total=0), apf IOs
```

  还将报告工作表的 I/O。

- 对外部表中每个限定行扫描内部表一次的嵌套循环连接。在下面的示例中，外部表 publishers 有三个州为“NY”的 publishers，因此内部表 titles 报告的扫描计数为 3：

```
    select title_id
    from titles t, publishers p
    where t.pub_id = p.pub_id
        and p.state = "NY"
Table:titles scan count 3,logical reads:(regular=442 apf=0 total=442),
physical reads:(regular=53 apf=289 total=342), apf IOs used=289
Table:publishers scan count 1, logical reads:(regular=2 apf=0 total=2),
physical reads:(regular=2 apf=0 total=2), apf IOs used=0
```

  此查询对 publishers 执行表扫描，后者仅占用两个数据页，因此报告两个物理 I/O。publishers 中有三个匹配行，因此查询使用按 pub_id 编制的索引扫描 titles 三次。

- 外部表中具有重复值的合并连接为每一个重复值重新启动扫描，并报告每次的额外扫描计数。

报告扫描计数为 0 的查询

多步查询和某些其它类型查询可能报告一个为 0 的扫描计数。一些示例如：

- 执行延迟更新的查询
- select...into 查询
- 创建工作表的查询

物理和逻辑读取之间的关系

如果某页需要从磁盘读取，则同时将其计为物理读取和逻辑读取。逻辑 I/O 始终大于或等于物理 I/O。

逻辑 I/O 始终报告 2K 的数据页。将以缓冲区大小为单位报告物理读取和写入。在一个 I/O 操作中读取的多页将被视为一个单位：它们作为一个缓冲区读取、写入和移入高速缓存。

逻辑读取、物理读取和 2K I/O

使用 2K I/O，对于一个查询在高速缓存中查找一页的次数是逻辑读取数减去物理读取数。当表扫描的逻辑读取与物理读取的总数相同时，意味着在查询过程中每页仅从磁盘读取和访问一次。

当在高速缓存中找到查询页时，逻辑读取数大于物理读取数。这经常出现有较高索引级的页中，由于它们经常重复使用且倾向于驻留在高速缓存中。

物理读取和大 I/O

页中不报告物理读取，但缓冲区中报告，即 Adaptive Server 访问磁盘的实际次数。

- 如果查询使用 16K I/O（showplan 报告 I/O 大小），则一次物理读取会将 8 个数据页引入高速缓存。
- 如果查询报告 100 个 16K 物理读取，则它已经向高速缓存中读入 800 个数据页。
- 如果查询需要扫描每一个数据页，它将报告 800 个逻辑读取。
- 如果一个查询（如连接）因其它 I/O 已从高速缓存中刷新某页而必须多次读取该页，则会对每次物理读取计数。

工作表上的读取和写入

对于任何需要为查询创建的工作表，都将报告读取和写入。当查询创建多个工作表时，工作表将在 statistics io 输出中进行编号，从而与 showplan 输出中使用的工作表编号相对应。

高速缓存对读取数的影响

如果测试查询并检查该查询的 I/O 后，再一次执行相同的查询，则可能会得到令人惊讶的物理读取值，特别是当查询使用最近使用最少 (LRU) 替换策略时。

第一次执行报告了一个很大的物理读取数；第二次则报告物理读取数为零。

第一次执行查询时，所有数据页读入到高速缓存中并且驻留在其中，直到其它服务器进程从高速缓冲中将其刷新。这些页在缓存中可能保留较长的时间，也可能保留较短的时间，这取决于用于查询的高速缓存策略。

- 如果查询使用读取和放弃 (MRU) 高速缓存策略，则会将这些页读入到高速缓存的清洗标记处。

  在小的或非常活跃的高速缓存中，读入到高速缓存清洗标记处的页很快被刷新。

- 如果查询使用 LRU 高速缓存策略将页读入页链 MRU 端的顶部，则这些页在高速缓存中将保留较长的一段时间。

在对生产系统的实际应用过程中，期望查询能在高速缓存中查到一些以前其他用户访问留下的所需页，而其它页需要从磁盘中读取。特别是较高级别的索引，可能会被经常使用，因而更有可能保留在高速缓存中。

如果有一个表或索引被绑定到高速缓存，而该高速缓存足够容纳所有页，则一旦该对象被读入到高速缓存后，将不再发生物理 I/O。

但是，在对具有很少用户的开发系统进行查询调优时，您可能需要从高速缓存中清除查询使用的页，以便查看查询所需的全部物理 I/O。可用下列方法从高速缓存中清除一个对象页：

- 改变对象的高速缓存绑定：
  - 如果表或索引已被绑定到高速缓存，先解除绑定，然后再重新绑定。
  - 如果表或索引未被绑定到高速缓存，先将其绑定到任何可用的高速缓存，然后再解除绑定。

  必须至少有一个用户定义的高速缓存才能使用这一选项。

- 如果没有任何用户定义的高速缓存，则可对其它表执行足够数量的查询，以便所关注的对象从高速缓存中刷新。如果高速缓存非常大，这将耗费很多时间。
- 重新启动服务器。

有关测试和高速缓存性能的详细信息，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”。

statistics io 和合并连接

statistics io 输出不包括对合并连接的排序开销。如果启用 allow resource limits，则“Total estimated I/O cost”和“Total actual I/O cost”统计信息中不会报告排序开销。

使用 *set statistics plancost* 分析查询

set statistics plancost 可简化查询分析。它可以显示估计的逻辑 I/O、物理 I/O 和行计数值与每个运算符中计算的实际值的比较结果，以及 CPU 和排序缓冲区开销的报告。

请参见《迁移指南》第 6 章“确保稳定性和性能”中的“set statistics plancost”。

set statistics plancost 示例

set statistics plancost 可简化查询分析，方法是显示估计的逻辑 I/O、物理 I/O 和行计数值与每个运算符中计算的实际值的比较结果，以及 CPU 和排序缓冲区开销的报告。

```
set statistics plancost on
```

使用 plancost 比较查询计划的估计开销和实际开销。plancost 还可帮助您诊断查询性能问题。表 1-2 介绍了 plancost 的实际输出和估计输出。

表 1-2：估计开销和实际开销

对于每个运算符，plancost 显示	对于此操作
`el:` - estimated `l:` - actual	逻辑 I/O
`ep:` - estimated `p:` - actual	物理 I/O
`er:` - estimated `r:` - actual	行计数
`cpu:` - actual	CPU 计数

执行基于排序或基于散列的操作的执行运算符会报告用于该操作的专用缓冲区数（“`bufct:`”，不在下面的示例中显示）。plancost 可能会显示一小部分开销，因为并非所有这些数量都适用于所有运算符。使用子集开销检验优化程序的估计值对次优查询计划是否有效。

例如，当您对 authors、titleauthor 和 titles 表运行连接查询时，plancost 会生成以下输出：

```
select A.au_fname, A.au_lname, T.title
from authors A, titleauthor TA, titles T
where A.au_id = TA.au_id and T.title_id = TA.title_id
==================== Lava Operator Tree ====================
                                              Emit
                                              (VA = 6)
                                              r:25 er:342
                                              cpu:0
                                         /
                                    MergeJoin
                                    Inner Join
                                    (VA = 5)
                                    r:25 er:342
                                  /                          \
                             Sort                            IndexScan
                             (VA = 3)                        titles_indx (T)
                             r:25 er:25                      (VA = 4)
                             l:6 el:6                        r:18 er:18
                             p:0 ep:0                        l:2 el:3
                             cpu:0 bufct:24                p:0 ep:3
                         /
                    MergeJoin
                    Inner Join
                    (VA = 2)
                    r:25 er:25
               /                                   \
          IndexScan                                IndexScan
```

```
auidind (TA)                          authors_indx (A)
(VA = 0)                              (VA = 1)
r:25 er:25                            r:23 er:23
l:1 el:2                              l:1 el:2
p:0 ep:2                              p:0 ep:2
```

查询优化程序会生成估计数值；实际数值是查询执行的结果。

此输出显示左下方的 IndexScan 运算符（在 titleauthors 表中）的估计 (`er:`) 和实际 (`r:`) 行计数均为 25，这意味着优化程序的估计是正确的。但是，顶部 MergeJoin (`VA = 5`) 的行计数估计是错误的：查询处理器的估计值是 342，但实际行计数为 25。

通过使统计信息保持最新状态，或者通过增加直方图梯级数，或许可以更准确地控制查询处理器的估计值。使用 set option show_missing_stats on 可检验连接列是否有直方图。通过创建直方图（如果尚不存在），或许可以改进查询处理器的估计值。

如果估计的行计数为 25，但实际行计数为 30，并不一定表示查询处理器的估计是错误。比较估计值和实际值时，应注意“数量级差异”（如上面示例中的 25 与 342）。

查询处理器会显示 `IndexScan` 运算符节点的索引名称（而非表名称）。若要确定与运算符节点关联的表：

- 索引名称可唯一地标识表。
- 如果查询中包含相关名，则运算符节点输出中将会包括该相关名。例如，在上面的输出中，`IndexScan` 运算符中的“`(TA)`”指的是 SQL 查询中的“`titleauthor TA`”。

查询树和 showplan 输出包括“`(VA=`*n*`)`”，其中，*n* = 0、1、2 等，用于唯一地标识每个运算符节点。

第 2 章

统计表和使用 *optdiag* 显示统计信息

本章介绍如何存储和显示统计信息。

存储统计信息的系统表

systabstats 和 sysstatistics 表可存储所有表、索引和已为其显式创建统计信息的任何未建索引列的统计信息：

- systabstats 可将有关表或索引的信息存储为对象，并由查询处理、数据定义语言和 update statistics 命令进行更新
- sysstatistics 可将有关值的信息存储在特定列中，并由数据定义语言和 update statistics 命令进行更新

请参见第 45 页的“SQL 命令对统计信息的影响”。有关所有系统表（包括以上这些表）的完整信息，请参见《参考手册：表》。

systabstats 表

systabstats 表包含表和索引的基本统计信息，其中包括：

- 表的数据页数，或索引的叶级页数
- 表中的行数
- 索引的高度
- 数据行和叶行的平均长度

- 已转移和已删除的行数
- 空页数
- 用以提高 I/O 开销估计精确性的统计信息，包括集群比、与分布页共享扩充的页数、OAM 数以及用于此对象的分布页数
- reorg 命令的停止点，以便它可重新开始处理

systabstats 为每个聚簇索引、每个非聚簇索引、每个无聚簇索引的表以及每个分区包含一行。聚簇索引信息的存储取决于表的锁定方案：

- 对于 DOL 锁定表，systabstats 为聚簇索引另外存储一行。
- 对于所有页锁定表，数据页被视为索引的叶级，因此聚簇索引的 systabstats 条目存储在与表数据相同的行中。

  所有页锁定表上聚簇索引的 indid 列始终为 1。

sysstatistics 表

sysstatistics 表为用户表上的每个索引列存储一行或多行；它还为未建索引列存储统计信息：

- 每一列的第一行存储该列的基本统计信息，例如连接密度和搜索参数、某些运算符的选择性和存储在列的直方图中的梯级数。
- 如果索引有多个列，或者，如果在生成未建索引列的统计信息时指定了多个列，则有一行用于列的每个前缀子集。

有关前缀子集的详细信息，请参见第 24 页的“列统计信息”。

其它行存储前导列的直方图数据。如果在将任何数据插入表之前已创建索引，则直方图不存在。若要生成直方图，请在插入数据后运行 update statistics。

请参见第 28 页的“直方图显示的内容”。

使用 *optdiag* 实用程序查看统计信息

optdiag 实用工具可显示 systabstats 和 sysstatistics 表中的统计信息。还可使用 optdiag 更新 sysstatistics 信息。仅系统管理员可运行 optdiag。

optdiag 语法

optdiag 的语法为：

```
optdiag
[binary] [simulate] statistics {-i input_file |
database[.owner[.table[.partition [.column]]]]
[-o output_file]} [-U user_name] [-P password] [-I interfaces_file]
[-S server] [-v] [-h] [-s] [-T flag_value] [-z language]
[-J client_charset] [-a display_charset]
```

使用 optdiag 显示整个数据库、单个表及其索引和列或特定列的统计信息。

例如，若要显示 pubtune 数据库中所有用户表的统计信息，以便将输出放到 *pubtune.opt* 文件中，可使用：

```
optdiag statistics pubtune -Usa -Ppasswd -o pubtune.opt
```

若要显示 titles 表和该表上任何索引的统计信息，请使用：

```
optdiag statistics pubtune..titles -Usa -Ppasswd -o titles.opt
```

从分区表运行 optdiag

对于低于 15.0 的 Adaptive Server® 版本中分区表的 optdiag 输出，数据是根据目标表的模式装载的：

- 目标表是单分区表 — optdiag 使用标准方法装载该表，忽略“最大分区中的页数”字段。
- 目标表是多分区表，并且“最大分区中的页数”字段为空 — optdiag 将统计信息装载到第一个分区。
- “最大分区中的页数”字段不为空 — optdiag 使用此字段中的值将“数据页计数”和“数据行计数”字段的值按比例划分到所有分区。optdiag 始终将列级统计信息装载为列的全局统计信息。

有关 *optdiag* 的完整信息，请参见《实用程序指南》。以下各节提供了有关 optdiag 输出的信息。

optdiag 标题信息

输出版本信息后，optdiag 会输出服务器名称并总结用于显示统计信息的参数。

optdiag 报告的标题列出了该报告中描述的对象：

```
Server name:                            "test_server"

Specified database:                     "pubtune"
Specified table owner:                   not specified
Specified table:                        "titles"
Specified column:                         not specified
```

表 2-1 描述输出内容。

表 2-1：表和列信息

行标签	提供的信息
服务器名	服务器的名称，存储在 @@*servername* 变量中。必须使用 sp_addserver，然后重新启动服务器，才能使用变量中的服务器名称。
Specified database	optdiag 命令行中提供的数据库名称。
Specified table owner	optdiag 命令行中提供的表所有者。
Specified table	optdiag 命令行中提供的表名称。
Specified column	optdiag 命令行中提供的列名。

表的统计信息

下面是表统计信息的 optdiag 输出示例：

```
Table owner:                          "dbo"
Table name:                           "authors"

Statistics for table:                "authors"

     Partition count:                3

Statistics for partition:            "authors_1376004902"
     Data page count:                74
     Empty data page count:          0
     Data row count:1666.0000000000000000
     Forwarded row count:            0.0000000000000000
     Deleted row count:              0.0000000000000000
     Data page CR count:10.0000000000000000
     OAM + allocation page count:3
```

```
    First extent data pages:0
    Data row size:85.2623049219687914
    Parallel join degree:          0.0000000000000000
    Unused page count:             5
    OAM page count:                1

Derived statistics:
    Data page cluster ratio:1.0000000000000000
    Space utilization:             0.9521597490347491
    Large I/O efficiency:          1.0000000000000000
```

表 2-2：表的统计信息

行标签	提供的信息
Table owner	表所有者的名称。可以通过指定 *dbname..tablename* 在命令行中省略所有者名称。如果多个表具有相同的名称，但所有者不同，则 optdiag 会输出具有该名称的每个表的信息。
Table name	表名。
Statistics for table	输出其统计信息的表的名称。
Partition count	分区数。
Statistics for partition	显示其统计信息的分区的名称。
Data page count	表中的数据页数。
Empty data page count	只包含已删除行的页的计数。
Data row count	表中数据行的数目。
Forwarded row count	表中转移行的数目。对于所有页锁定表，此值始终为 0。
Deleted row count	已从表中删除的行数。这些是已提交的删除，其中的空间还未被某个用于清除已删除的行的函数回收。 对于所有页锁定表，此值始终为 0。
Data page CR count	用于得出数据页集群比的计数器。计算数据页集群比，以帮助确定表扫描和区域扫描的大 I/O 的效率。仅当运行 update statistics 时，才会更新此值
OAM + allocation page count	表的 OAM 页数，加上分配单元（表占据其空间）数。这些统计信息用于估计对 DOL 锁定表进行 OAM 扫描的开销。 只在 DOL 锁定表上维护此值。
First extent data pages	与分配页共享分配单元中第一个扩充的页数。这些页需要用 2K 的 I/O 而不是大型 I/O 读取。 只为 DOL 锁定表维护此信息。
Data row size	数据行的平均长度（以字节为单位）。大小包括行开销。 此值仅由 update statistics、create index 和 alter table...lock 进行更新。
Parallel join degree	指示用于嵌套循环连接的并行度的整数值。
Unused page count	扩充中的未使用页数。

行标签	提供的信息
OAM page count	OAM 页数。
Derived statistics	optdiag 为其派生信息的统计信息的组。
Data page cluster ratio	请参见下文的“数据页集群比”。
Space utilization	请参见下文的“Space utilization”。
Large I/O efficiency	请参见下文的“Large I/O efficiency”。

表级派生的统计信息

“派生的统计信息”报告有关“数据页集群比”、“空间利用率”和“大 I/O 效率”的统计信息，这些信息是从“数据页 CR 计数”和数据页计数中派生的。

数据页集群比

对于所有页锁定表，数据页集群比用于衡量在按页链顺序读取表时，在扩充上对页进行排列的效果。集群比为 1.0 时表示排列效果最好。集群比较小表示页链较分散。

对于 DOL 锁定表，数据页集群比用于衡量扩充上页的填充情况。集群比为 1.0 时表示完全填满扩充。数据页集群比较小表示扩充中包含分配给表的空页。

Space utilization

空间利用率使用平均大小和数目的行来计算数据页的预期最小数目，并将该预期最小数目与当前页数进行比较。如果空间利用率低，请在表上运行 reorg rebuild 或删除并重新创建聚簇索引，以减少数据页上的空闲空间量，以及扩充中分配给表的空白页数。

如果您使用空间管理属性（如 fillfactor 或 reservepagegap），则为带聚簇索引的表的数据页上的其它行保留的空闲空间和扩充中为该表保留的空白页数会影响空间利用率值。

如果最近没有更新统计信息，而且平均行大小已经发生了变化，或者行数和页数不准确，那么，空间利用就可能会报告大于 1.0 的值。

Large I/O efficiency

“大 I/O 效率”可估计由每个大 I/O 引入的有用页数。例如，如果“大 I/O 效率”的值为 .5，则 16K I/O 平均返回查询所需的 4 个 2K 页以及另外 4 页（空白页或因无法聚簇而需共享扩充的页）。低效率值表示在表上重新创建聚簇索引或运行 reorg rebuild 可改进 I/O 性能。

索引统计信息

下面是有关 DOL 锁定表上每个非聚簇索引、聚簇索引以及所有页锁定表的聚簇索引的输出示例。表统计信息报告中包含所有页锁定表上聚簇索引的相关信息。第 22 页的表 2-3 描述了该输出。

```
Statistics for index:                "title_id_ix" (nonclustered)
Index column list:                    "title_id"
     Leaf count:                       45
     Empty leaf page count:            0
     Data page CR count:               4952.0000000000000000
     Index page CR count:              6.0000000000000000
     Data row CR count:                4989.0000000000000000
     First extent leaf pages:          0
     Leaf row size:                    17.8905999999999992
     Index height:                     1


  Derived statistics:
     Data page cluster ratio:          0.0075819672131148
     Index page cluster ratio:         1.0000000000000000
     Data row cluster ratio:            0.0026634382566586
```

注释　仅对带聚簇索引的所有页锁定表显示未包括在此示例中的并行连接度、未使用页计数和 OAM 页计数。

表 2-3：索引统计信息

行标签	提供的信息
Statistics for index	索引名和类型。
Index column list	索引中列的列表。
Leaf count	索引中叶级页的数目。
Empty leaf page count	索引中空叶页的数目。
Data page CR count	一种计数器，用于计算数据页集群比，以便使用索引访问表。 请参见第 22 页的“索引级派生的统计信息”。
Index page CR count	用于计算索引页集群比的计数器。 请参见第 22 页的“索引级派生的统计信息”。
Data row CR count	用于计算数据行集群比的计数器。 请参见第 22 页的“索引级派生的统计信息”。
First extent leaf pages	索引中的叶页数，存储在分配单元的第一个扩充中。这些页需要用 2K 的 I/O 而不是大型 I/O 读取。 只为 DOL 锁定表上的索引维护此信息。
Leaf row size	索引中的叶级行的平均大小。此值仅由 update statistics、create index 和 alter table...lock 进行更新。
索引高度	索引的高度（叶级不计算在内）。只是对于所有页锁定表的聚簇索引，此行才包含在表级输出中。对于其它索引，索引高度出现在索引级输出中。 此值不适用于堆表。

索引级派生的统计信息

索引级部分的派生统计信息基于第 21 页的“索引统计信息”中所示的“CR 计数”值。

数据页集群比

当使用索引访问数据页时，数据页集群比用于衡量大 I/O 的效率。如果表经过最佳聚簇索引，则集群比为 1.0。数据页集群比可以在很大范围内变化。对一些索引来说，它们可能非常大，而对另一些来说，则可能很小。

索引页集群比

索引页集群比用于估计查询所用的大 I/O 的开销，这些查询需要从非聚簇索引或 DOL 锁定表的聚簇索引中读取大量叶级页。这些查询的某些示例是覆盖索引扫描和读取大量行的范围查询。

在新建的索引中，如果“索引页集群比”为 1.0 或非常接近于 1.0，表明这是扩充上索引叶页的最佳聚簇。索引页被拆分且新页从附加扩充中划分出来后，比率将会下降。很低的百分比可能表示，在其上删除和重建索引或运行 reorg rebuild 会改进性能，尤其是当许多查询都执行覆盖扫描时。

数据行集群比

数据行集群比用于估计当使用索引访问数据页时需要读取的页数。

索引的空间利用

空间利用使用平均行宽与行数来计算预期的最小叶级索引页大小，并将其与当前叶页数进行比较。

如果空间利用率低，则对索引运行 reorg rebuild，或删除并重新创建它可减少索引页上的空闲空间量，以及扩充中分配给索引的空白页数。

如果您正在使用空间管理属性（如 fillfactor 或 reservepagegap），则为叶页上其它行保留的空闲空间和在扩充中为索引保留的空白页数会影响空间利用率。

如果最近没有更新统计信息，而且平均行大小已经发生了变化，或者行数和页数不准确，那么，空间利用就可能会报告大于 1.0 的值。

索引的大 I/O 效率

大 I/O 效率可估计由每个大 I/O 引入的有用页数。例如，如果值为 .5，则 16K I/O 平均返回查询所需的 4 个 2K 页以及另外 4 页（空白页或因无法聚簇而需共享扩充的页）。

低效率值表示重新创建索引或运行 reorg rebuild 可改进 I/O 性能。

列统计信息

optdiag 列级统计信息包括：

- 给出列的密度和选择性的统计信息。如果索引包括多个列，optdiag 将输出如表 2-4 所示的有关索引键的每个前缀子集的信息。如果使用包含列名称列表的 update statistics 创建统计信息，则会在该列列表中存储每个前缀子集的密度统计信息。
- 一个直方图，如果在创建索引时表包含一行或多行数据，或如果运行 update statistics。则会有一个前导列的直方图用于：
  - 当前存在的每个索引（如果创建索引时列中至少有一个非空值）
  - 已创建和删除的任何索引（只要尚未运行 delete statistics）
  - 任何已运行 update statistics 的列列表

  还有一个直方图用于：
  - 索引中的每一列（如果使用了 update index statistics 命令）
  - 表中的每一列（如果使用了 update all statistics 命令）

optdiag 还会输出表中没有统计信息的列的列表。例如，authors 表中的以下列没有统计信息：

```
No statistics for column(s):        "address"
(default values used)               "au_fname"
                                    "phone"
                                    "state"
                                     "zipcode"
```

列统计信息的输出样本

以下示例显示 authors 表中 city 列的统计信息：

```
Statistics for column:                  "au_lname"
Derived statistics from local partitions.
Last update of column statistics:        Apr 8 2008 9:29:07:706PM

     Range cell density:                 0.0005507993510047
     Total density:                      0.0005507993510047
     Range selectivity:                  default used (0.33)
     In between selectivity:             default used (0.25)
     Unique range values:                0.0005215561314205
     Unique total values:                0.0005215561314205
     Average column width:                6.7988000000000000
```

表 2-4：列统计信息

行标签	提供的信息
Statistics for column	列名；如果此信息块提供组合索引或列的列表中前缀子集的信息，则行标签是“Statistics for column group”。
Last update of column statistics	创建索引的日期，上次运行 update statistics 的日期，或上次使用 optdiag 更改统计信息的日期。
Statistics originated from upgrade of distribution page	从低于 11.9 的 Adaptive Server 版本的分布页升级时产生的统计信息。如果已对表或索引运行 update statistics，或如果已删除索引并在升级后重新创建索引，则不输出此消息。 如果此消息显示在 optdiag 输出中，请运行 update statistics。
Statistics loaded from optdiag	optdiag 用于更改 sysstatistics 信息。create index 命令会输出警告消息，指示将覆盖已编辑的统计信息。 如果统计信息由 update statistics 或 create index 生成，则不显示此行。
Range cell density	列上等同性搜索参数的密度。 请参见第 26 页的“域单元与总密度值”。
Total density	列的连接密度。此值用于估计为此列上的连接返回的行数。 请参见第 26 页的“域单元与总密度值”。
Range selectivity	输出缺省值 .33，除非已使用 optdiag 输入模式更新该值。 如果不知道搜索参数，可将此值用于范围查询。 请参见下文的“域选择性和两者之间选择性的值”。
In between selectivity	输出缺省值 .25，除非已使用 optdiag 输入模式更新该值。 如果不知道搜索参数，可将此值用于范围查询。 请参见下文的“域选择性和两者之间选择性的值”。
Unique range values	列中唯一值的数目，不包括频率单元。
Unique total values	唯一值的总数。
Average column width	表中所有列宽的平均数。

域单元与总密度值

Adaptive Server为列密度值存储了两个值：

- “范围单元密度”仅测量范围单元的重复值。

  如果有任何列的频率单元，则从域单元密度的计算中消除它们。

  如果只有列的频率单元，而没有域单元，则域单元密度为 0。

  有关范围单元和频率单元的信息，请参见第 29 页的“了解直方图输出”。

- “总密度”测量所有列的重复值，包括由范围单元和频率单元表示的列。

使用两个单独的值可以提高优化程序对要返回行数的估计能力：

- 如果搜索参数与频率单元的值匹配，则会返回由频率单元的权值表示的行的系数。
- 如果搜索参数在域单元范围内, 则使用域单元密度和域单元权值来估计要返回的行数。

对于连接，优化程序根据每次扫描表时返回的平均行数进行估计，因此总密度（测量列中所有值的重复值的平均数）提供了最佳估计。如果搜索参数的值在优化查询时未知，总密度还可用于等同性参数。

请参见第 27 页的“域选择性和两者之间选择性的值”。

对于多个列上的索引，将为每个前缀子集存储域单元密度和总密度。在下面的输出示例中，对于 titles (pub_id, type, pubdate) 上的索引，每考虑增加一列，密度值会随之减少。

```
Statistics for column group:              "pub_id", "type"
Last update of column statistics:      Apr 8 2008 9:22:40:963AM

     Range cell density:0.0000000362887320
     Total density:0.0000000362887320
     Range selectivity:             default used (0.33)
     In between selectivity:        default used (0.25)
     Unique range values:           0.0000000160149449
     Unique total values:           0.0000000160149449
     Average column width:          default used (8.00)

Statistics for column group:       "pub_id", "type", "pubdate"
Last update of column statistics:   Apr 8 2008 9:22:40:963AM

     Range cell density:            0.0000000358937986
     Total density:                 0.0000000358937986
```

```
Range selectivity:              default used (0.33)
In between selectivity:         default used (0.25)
Unique range values:            0.0000000158004305
Unique total values:            0.0000000158004305
Average column width:           2.0000000000000000
```

对于有 5000 行的表，提高要返回的优化程序的行的估计精度将取决于查询中使用的搜索参数的数目：

- 仅对 pub_id 使用相同数目的搜索参数时，估计将返回 0.0335391029690461 * 5000 行，或 168 行。
- 对所有三列使用相同数目的搜索参数时，估计将返回 0.0002011791956201 * 5000 行（即，仅返回 1 行）。

随着被评估的搜索参数的增多而提高的精确性级别可显著提高许多查询的优化。

域选择性和两者之间选择性的值

optdiag 输出域和“两者之间”选择性的缺省值，或在早期的 optdiag 会话中已为这些选择性设置的值。在优化查询时如果搜索参数为未知，这些值将用于范围查询。

对于其值未知的等同性搜索参数，缺省情况下使用总密度。

搜索参数在以下各项进行优化时是未知的：

- 存储过程，用于设置过程内的变量
- 批处理中的查询，用于为批处理中的搜索参数设置变量

近似值可能导致生成次优查询计划，因为它们高估或低估了查询将返回的行数。

有关使用 optdiag 提供选择性值的信息，请参见第 37 页的“使用 optdiag 输入模式更新选择性”。

直方图显示的内容

直方图存储有关列中值分布的信息。表 2-5 显示用于创建和更新直方图的命令以及受影响的列。

表 2-5：创建直方图的命令

命令	直方图用于
create index	仅限前导列
update statistics	
table_name 或 *index_name*	仅限前导列
column_list	仅限前导列
update index statistics	所有索引列
update all statistics	所有列

直方图的输出样本

```
Histogram for column:              "city"
Column datatype:                    varchar(20)
Requested step count:               20
Actual step count:                  20
Sampling percent:                   0
Out of range histogram adjustment:   DEFAULT
```

optdiag 首先输出直方图的汇总数据，如表 2-6 所示。

表 2-6：直方图汇总统计信息

行标签	提供的信息
Histogram for column	列名。
Column datatype	列的数据类型，包括与该数据类型相应的长度、精度和标度。
Requested step count	为列请求的梯级数。
Actual step count	为列生成的梯级数。 如果列中不同值的数目小于请求的梯级数，则此数字可以小于请求的梯级数。
Sampling percent	为生成直方图而采集的表数据的百分比。值范围为 0 - 100。
Out of range histogram adjustment	指示是否调整该列的直方图，以便为列中的超范围搜索参数分配选择性值。值为 on，off 或 default 之一。仅在全局列统计信息中（而不在分区级别上）显示此行与直方图信息。

直方图输出按列显示，如表 2-7 中所示。

表 2-7：optdiag 直方图输出中的列

列	提供的信息
Step	梯级数。
Weight	梯级的权值。
（运算符）	<、<= 或 =，表明值的限制。根据单元是代表范围单元还是频率单元，运算符会有所不同。此列没有标题输出。
Value	由范围单元表示的值的上限，或者由频率计数表示的值。

了解直方图输出

直方图是一组单元，其中每个单元都分配有一个权值。每个单元有一个上限和一个下限，它们是不同的列值。单元的权值是介于 0 和 1 之间的浮点值，代表以下两者之一：

- 表中处于值范围内的行的系数（如果运算符为 <=），或者
- 如果运算符是 =，则为与梯级相匹配的值的数目。

优化程序使用域、权值以及密度值的组合来估计对于列的查询子句要返回的表中的行数。

Adaptive Server 使用等高直方图，其中每个单元表示的行数大约相等。例如，pubtune..authors 中 city 列的以下直方图有 20 个梯级；直方图中的每个梯级都大约表示该表的 5%：

```
Step     Weight                 Value

   1     0.00000000     <=      "APO Miami\377\377\377\377\377\377\377"
   2     0.05460000     <=      "Atlanta"
   3     0.05280000     <=      "Boston"
   4     0.05400000     <=      "Charlotte"
   5     0.05260000     <=      "Crown"
   6     0.05260000     <=      "Eddy"
   7     0.05260000     <=      "Fort Dodge"
   8     0.05260000     <=      "Groveton"
   9     0.05340000     <=      "Hyattsville"
  10     0.05260000     <=      "Kunkle"
  11     0.05260000     <=      "Luthersburg"
  12     0.05340000     <=      "Milwaukee"
  13     0.05260000     <=      "Newbern"
  14     0.05260000     <=      "Park Hill"
  15     0.05260000     <=      "Quicksburg"
```

```
16      0.05260000      <=      "Saint David"
17      0.05260000      <=      "Solana Beach"
18      0.05260000      <=      "Thornwood"
19      0.05260000      <=      "Washington"
20      0.04800000       <=      "Zumbrota"
```

直方图中第一个梯级表示表中空值的比例。由于 city 没有空值，因此权值为 0。表示空值的梯级值由小于最小列值的最大值表示。

对于字符串，第一个单元的值是小于最小列值（在上面的输出中为“ APO Miami ”）的可能最大的字符串值，使用服务器使用的字符集中的最长字符对其进行了填充，以达到定义的列长度。您在输出中所看到的实际内容取决于您用于查看 optdiag 输出文件的字符集、终端类型和软件。

在上面的直方图中，由每个单元表示的值都包括上限，但不包括下限。此直方图中的单元称为**范围单元**，因为每个单元都表示一系列值。

可按如下所示表示范围单元中包括的值的范围：

下限 <（单元的值）<= 上限

在 optdiag 输出中，下限是上一梯级的值，上限是当前梯级的值。

例如，在以上直方图中，第 4 梯级包括 Charlotte（上限），但不包括 Boston（下限）。该梯级的权值是 .0540，表示表的 5.4% 与查询子句相匹配：

```
where city > Boston and city <= "Charlotte"
```

optdiag 直方图输出中的运算符列显示 <= 运算符。不同的运算符被用于具有高重复值的直方图。

具有高重复值的列的直方图

具有高重复值的列的直方图与具有大量不连续值的列的直方图看上去差别很大。在具有高重复值的列的直方图中，有一个代表重复值的单元，称为**频率单元**。

频率单元的权值显示具有匹配值的列的百分比。

频率单元的直方图输出的不同，取决于列值代表以下哪一项：

- 密集频率计数，其中列中的值在域中是连续的。例如，1、2、3 是连续的整数值。
- 稀疏频率计数，其中，可能值的域包括不由表中一组离散值表示的值。
- 密集频率计数和稀疏频率计数的混合。

某些列的直方图输出包括频率单元和域单元的混合。

密集频率计数的直方图

以下输出显示具有 6 个不同整数值 (1 - 6) 和一些空值的列的直方图：

```
Step     Weight                 Value

   1     0.13043478     <=      1
   2     0.04347826     <=      1
   3     0.17391305     <=      2
   4     0.30434781     <=      3
   5     0.13043478     <=      4
   6     0.17391305     <=      5
   7     0.04347826      <=      6
```

以上的直方图显示密集频率计数，因为列中所有的值都是连续的整数。

第一个单元表示空值（在下面一节中介绍）。因为有空值，所以此单元的权值表示列中空值的百分比。

第一个梯级的“值”列显示表中的最小列值和 < 运算符。

稀疏频率计数的直方图

在表示具有稀疏频率计数的列的直方图中，高重复值由使用 = 运算符和单元的权值显示离散值的梯级表示。

在每个梯级之前，都有一个权值为 0.0、值相同且运算符为 < 的梯级，指示表的各行中没有介入值。对于具有空值的列，如果表中有空值，则第一个梯级将有一个非零权值。

以下直方图表示 titles 表中 type 列。因为只有 9 个不同的类型，所以它们由 18 个梯级表示。

```
Step      Weight                    Value

    1     0.00000000        <       "UNDECIDED   "
    2     0.11500000        =       "UNDECIDED   "
    3     0.00000000        <       "adventure   "
    4     0.11000000        =       "adventure   "
    5     0.00000000        <      "business    "
    6     0.11040000        =      "business    "
    7     0.00000000        <      "computer    "
    8     0.11640000        =      "computer    "
    9     0.00000000        <      "cooking     "
   10     0.11080000        =      "cooking     "
   11     0.00000000        <      "news        "
```

```
    12      0.10660000          =       "news        "
    13      0.00000000          <       "psychology  "
    14      0.11180000          =       "psychology  "
    15      0.00000000          <       "romance     "
    16      0.10800000          =       "romance     "
    17      0.00000000          <       "travel      "
    18      0.11100000           =       "travel      "
```

例如，类型列中 10.66% 的值是“news”，因此优化程序估计一个具有 5000 行的表将会有 533 行被返回。

具有稀疏值和密集值的列的直方图

对于一些值为高重复值而另一些值为分布式值的表，直方图的输出显示了运算符与频率单元和域单元混合的组合。

以下直方图表示的列中，具有很大比例的行的值分别为 30.0、50.0 和 100.0。

对于其中每个值，直方图中都有两个梯级：一个梯级表示高重复值具有 = 运算符和显示与该值匹配的列的百分比的权值。另一个梯级表示每个高重复值的梯级具有 <运算符和权值 0.0。此列的数据类型是 numeric(5,1)。

```
Step      Weight                Value
    1     0.00000000     <=      0.9
    2     0.04456094     <=     20.0
    3     0.00000000     <      30.0
    4     0.29488859     =      30.0
    5     0.05996068     <=     37.0
    6     0.04292267     <=     49.0
    7     0.00000000     <      50.0
    8     0.19659241     =      50.0
    9     0.06028834     <=     75.0
   10     0.05570118     <=     95.0
   11     0.01572739     <=     99.0
   12     0.00000000     <     100.0
   13     0.22935779      =     100.0
```

由于此列中的最小值为 1.0，则用 0.9 来代表空值的梯级。

选择高重复值的梯级数

本节中频率单元的直方图示例使用数量相对较少的高重复值，因此生成的直方图需要的梯级数少于 20，该数目是 create index 或 update statistics 的缺省梯级数。

如果表中某一列包含大量高重复值，且该列中键的分布不一致，则增加直方图中的梯级数可使优化程序在使用搜索参数查询该列时做出更精确的开销估计。

对于具有密集频率计数的列，梯级数应比值的数目至少多一个，以允许该单元的一个梯级表示空值。

对于具有稀疏频率计数的列，使用的梯级数至少应是不同值数目的两倍。这样就允许具有零权值的介入单元以及表示空值的单元。例如，如果 pubtune 数据库中的 titles 表有 30 个不同的价格，则此 update statistics 命令将用 60 个梯级创建直方图：

```
update statistics titles
using 60 values
```

此 create index 命令指定 60 个梯级：

```
create index price_ix on titles(price)
with statistics using 60 values
```

如果某一列包含的一些值只与极少的行匹配，这些值仍可能代表域单元，但最终产生的直方图梯级数将少于所请求的数。例如，为 state 列请求 100 个梯级，可能为那些由只占很小比例的行所表示的 state 生成一些域单元。

配置直方图调优因子

histogram tuning factor 控制使用 update statistics、update index statistics、update all statistics 和 create index 创建的每个直方图中 Adaptive Server 可分析的梯级数。

histogram tuning factor 可最大限度地减少直方图消耗的资源，并且仅当它可改善优化时才会增加资源使用。例如，当列中存在重复值或分布不一致的数据时。在这些情况下，最多可使用 400 个（最大值）直方图梯级。但是，在多数情况下，Adaptive Server 使用缺省值（在上例中为 20）。

对于 Adaptive Server 15.0.2 ESD #2 之前的版本，此配置参数的缺省值为 1（禁用）。对于 Adaptive Server 15.0.2 ESD #2 及更高版本，此配置参数的缺省值为 20（启用）。

请参见《系统管理指南，卷 1》的第 5 章“设置配置参数”中的 histogram tuning factor。

有多少个直方图梯级？

对于具有少量不同值（即“低基数”）的列或具有少量行的表，number of histogram steps 的缺省值 (20) 可能已足够。但是，对于包含大量行的表和其中列具有许多不同值的表，缺省值可能不够，尤其是在未将列的值平均分布到各行时。

如果您认为查询处理器会生成次优查询计划，请尝试通过增加直方图的梯级数来增加其粒度。在查询处理器生成查询计划前，增加其直方图梯级数可消耗更多资源（尤其是过程高速缓存使用率）并延长优化时间。

增加直方图梯级数时，optimization timeout limit 配置参数可导致查询处理器生成次优查询计划。

您必须确定更多数目的直方图梯级是否会生成更好的查询计划和更好的整体性能。您可能需要将 number of histogram steps 增加到 200；如果这样无法改进查询计划和性能，请尝试使用更大的数值，如 500。

或者，可针对每 10,000 个数据页使用一个直方图梯级。但是，通常情况下，即使将梯级数增加到 1000 - 2000 以上，您也不太可能看到任何改进。如果您在更改直方图梯级数后未看到查询计划或性能有所改进，则应在其它方面查找瓶颈。

在您执行 update index statistics 时，Adaptive Server 可确定直方图梯级数：

1 number of histogram steps 可为使用所有 create index 和 update index statistics 命令创建的新直方图定义缺省值。number of histogram steps 的缺省值为 20。

2 对于 update index statistics，可使用以下命令将直方图梯级数显式设置为 *nnn* 值：

```
update index statistics table_name
using nnn values
```

3 update index statistics 在创建新直方图时使用现有直方图中的当前梯级数。number of histogram steps 配置参数不适用于现有直方图。仅当您使用 `using` ***`nnn`*** `values` 显式指定梯级数时，update [index] statistics 才会覆盖现有直方图。

   此步骤完成后，number of histogram steps 配置参数的值就是直方图的目标梯级数。这在 optdiag 输出中显示为“Requested step count”。

4　Adaptive Server 将 *nnn* 值（由步骤 2 确定）与 histogram tuning factor 的值相乘，以生成内部中间直方图。例如，如果 *nnn* 为 100，并且 histogram tuning factor 为 20，则中间直方图可能最多有 2000 个梯级 (20 * 100 = 2000)。生成此内部直方图增加了 Adaptive Server 标识“频率单元”（其中包括重复数据值）的机会。如果 Adaptive Server 找不到频率单元，则它会将直方图恢复到原始梯级数。Adaptive Server 保留它找到的任何频率单元。

原始梯级数在 optdiag 输出中显示为“Requested step count”。optdiag 将其输出中的实际直方图梯级数显示为“Actual step count”。

5　在低于 15.0.1 ESD #1 的版本中，Adaptive Server 缺省情况下将 histogram tuning factor 设置为 1。对于 15.0.1 ESD#1 及更高的版本，Adaptive Server 对 histogram tuning factor 使用缺省值 20。Sybase® 建议，除非 Sybase 技术支持部门另行指导，否则请使用缺省值 20。

使用 *optdiag* 更改统计信息

系统管理员可使用 optdiag 更改列级统计信息。

警告！ 使用 optdiag 更改统计信息可以改进某些查询的性能。但是，optdiag 会覆盖系统表中的现有信息，从而影响对给定表的所有查询。

要非常谨慎使用并彻底测试在使用该表的所有查询上的所做的所有更改。如果可能，在将统计信息装载到生产服务器前，请在开发服务器上使用 optdiag simulate 测试这些更改。

如果未在模拟模式下装载统计信息，请根据需要准备恢复统计信息，方法是使用 optdiag 输出的未改动副本或重新运行 update statistics。

不要尝试通过运行 update、delete 或 insert 命令来更改任何统计信息。

可使用 32 位 Adaptive Server 中的 optdiag 输出更改另一 32 位 Adaptive Server 中的统计信息，但不能更改 64 位 Adaptive Server 中的统计信息。同样，不要将 64 位 Adaptive Server 的 optdiag 输出用作 32 位 Adaptive Server 的输入。

使用 optdiag 更改统计信息后，运行 create index 或 update statistics 将覆盖所作的更改。命令成功执行，但显示一则警告消息：

```
WARNING:Edited statistics are overwritten.表:'titles' (objectid 208003772),
column:'type'.
```

使用 *optdiag binary*

注释 输出以十六进制（而非二进制）形式显示。

因为使用浮点数时会失去精度，因此 optdiag 提供了一个 binary 选项。以下命令将显示可读统计信息和二进制统计信息：

```
optdiag binary statistics pubtune..titles.price
        -Usa -Ppasswd -o price.opt
```

在 binary 中，可使用 optdiag 编辑的任何统计信息都会输出两次，一次使用二进制值，一次使用浮点值。显示浮点值的行以 optdiag 注释字符（即井号 (#) 开头）。

下面的示例显示表示 authors 表中 city 列的直方图的前几行：

```
Step  Weight                Value

  1   0x3d2810ce    <=      0x41504f204d69616d68ffffffffffffffffffffff
# 1   0.04103165    <=      "APO Miami\377\377\377\377\377\377\377\377"
  2   0x3d5748ba    <=      0x41746c616e7461
# 2   0.05255959    <=      "Atlanta"
  3   0x3d5748ba    <=      0x426f79657273
# 3   0.05255959    <=      "Boyers"
  4   0x3d58e27d    <=      0x4368617474616e6f6f6761
# 4   0.05295037    <=       "Chattanooga"
```

当 optdiag 装载此文件时，会读取所有未注释的行，而忽略井号 (#) 后面的所有字符。若要编辑浮点值（而不是二进制值），请从显示浮点值的行中删除井号 (#)，然后在显示二进制值的相应行的开头插入井号 (#)。

何时使用 *binary*

optdiag 输出中的两个直方图梯级可能由于缺乏精度而显示相同的值，即使二进制值并不同。例如，在十进制中 1.999999999 和 2.000000000 可能都显示为 2.000000000，即使其二进制值是不同的。在此类情况下，应使用 binary 来进行输入。

如果不使用二进制模式，则 optdiag 会发出错误消息，指示不会增加梯级值，并建议您使用二进制模式。为避免在执行 sysstatistics 时失去精度，optdiag 会跳过装载出错的直方图。

使用 *optdiag* 输入模式更新选择性

如果在查询优化期间搜索参数的值未知，则优化程序会使用范围选择性值和两者之间选择性值。可使用 optdiag 自定义服务器范围内的缺省选择性值，以匹配应用程序中特定列的数据。

服务器范围的缺省值为：

- 范围选择性 — 0.33
- 两者之间选择性 — 0.25

下面的示例显示 optdiag 如何显示缺省值：

```
Statistics for column:              "city"
Last update of column statistics:   Feb  4 2008  8:42PM

     Range cell density:            0x3f634d23b702f715
#    Range cell density:            0.0023561189228464
     Total density:                 0x3f46fae98583763d
#    Total density:                 0.0007012977830773
     Range selectivity:             default used (0.33)
#    Range selectivity:             default used (0.33)
     In between selectivity:        default used (0.25)
#    In between selectivity:         default used (0.25)
```

若要编辑选择性值，请将整个“default used (0.33)”或“default used (0.25)”字符串替换为浮点值。以下示例使用十进制值将域选择性改为 .25，将两者之间选择性改为 .05：

```
     Range selectivity:            0.250000000
     In between selectivity:        0.050000000
```

编辑直方图

您可以编辑直方图，以便：

- 删除一个梯级，方法是先将其权值传送给相邻行，然后再删除该梯级。
- 增加一个或多个梯级，方法是先将某个单元的权值扩展到其它行，并且扩展任何新梯级所代表的列值的上限。

向直方图增加频率计数单元

编辑直方图的原因之一是，增加频率计数单元，而不大量增加梯级数。

使用密集频率计数编辑直方图

对直方图所做的更改会有所不同，具体取决于这些值是表示密集频率计数还是稀疏频率计数。若要为给定列值添加频率单元，请检查列值是否正好小于新单元的值。如果下一个更小的值与要添加的值很接近，则可确定是频率计数。

如果更改的下一个更小的列值与频率计数值很接近，则可轻松提取频率计数单元。

例如，如果某个列中至少包括一个 19 和多个 20，且直方图使用一个单元来表示所有大于 17 且小于等于 22 的值，则 optdiag 输出显示该单元的以下信息：

```
Step     Weight                Value
...
4        0.100000000    <=     17
5        0.400000000    <=     22
...
```

要变更此直方图以将值 20 放置在该值自己的梯级上，这需要增加两个梯级，如下所示：

```
...
4     0.100000000    <=     17
5     0.050000000    <=     19
6     0.300000000    <=     20
7     0.050000000    <=     22
...
```

在上面已变更的直方图上，梯级 5 代表所有大于 17 且小于等于 19 的值。在修改后的直方图中，梯级 5、6 和 7 的权值总和等于梯级 5 的原始权值。

使用稀疏频率计数编辑直方图

如果列中没有大于 17 且小于 20 的值，请使用稀疏频率计数表示形式。以下是初始的直方图梯级：

```
Step     Weight                Value
...
4        0.100000000    <=     17
5        0.400000000    <=     22
...
```

以下示例显示权值为零的梯级 5，它是稀疏频率计数所要求的：

```
...
4     0.100000000    <=    17
5     0.000000000    <     20
6     0.350000000    =     20
```

```
7     0.050000000    <=    22
...
```

梯级 5 的运算符必须为 <。梯级 6 必须为值 20 指定权值，而且其运算符必须是 =。

跳过对梯级编号的装载时间检验

缺省情况下，optdiag 输入模式会检查直方图中的梯级数是否是以 1 为增量进行编号。编辑直方图梯级数后，若要跳过此检查，请使用命令行标志 -T4：

```
optdiag statistics pubtune..titles -Usa -Ppassword -T4 -i titles.opt
```

直方图装载期间检查的规则

在直方图输入期间，会检查以下规则。冲突的规则会生成错误消息。

- 每个梯级编号都应大于上一梯级编号（除非打开 -T4）。对于每个梯级的列值，该值应大于或等于上一梯级的列值。
- 梯级的列值必须单调递增。
- 每个单元的权值必须介于 0.0 和 1.0 之间。
- 列的总权值必须接近 1.0。
- 第一个单元代表空值，而且它必须存在，即使是对于不允许空值的列也是如此。只能有一个单元代表空值。
- 相邻的两个单元不能同时使用 <运算符。

重新创建索引而不丢失统计信息更新

若要在更新直方图后删除和重新创建索引，并且您希望保留编辑的值，请在 create index 命令中将梯级数指定为 0。此命令会重新创建索引而不更改直方图：

```
create index title_id_ix on titles(title_id)
with statistics using 0 values
```

使用模拟统计信息

optdiag 可以生成统计信息，这些统计信息可用于模拟用户环境，而不需要复制表数据。这允许使用很小的数据库进行查询优化分析。例如，可使用模拟统计信息执行以下操作：

- 帮助技术支持部门复制优化程序问题
- 执行“假设”分析以计划配置更改
- 对开发服务器进行诊断

请参见第 42 页的“装载和使用模拟统计信息的要求”。

还可以将模拟统计信息装载到当初从中拷贝这些信息的数据库中。模拟统计信息会和可以将其与实际表数据区分开来的 ID 一起装载到系统表中。set statistics simulate on 命令可指示服务器使用模拟统计信息（而不是实际的统计信息）优化查询。

模拟统计信息的 *optdiag* 语法

若要显示 pubtune 数据库的模拟模式统计信息，请使用以下命令：

```
optdiag simulate statistics pubtune -o pubtune.sim
```

若要生成二进制模拟输出，请使用以下命令：

```
optdiag binary simulate statistics pubtune -o pubtune.sim
```

若要装载这些统计信息，请使用以下命令：

```
optdiag simulate statistics -i pubtune.sim
```

模拟统计信息输出

optdiag 的 simulate 选项的输出会为每行统计信息显示标有“模拟”的行，但直方图除外。可以修改和装载模拟值，同时将文件保留为实际值的记录。

- 如果指定 binary 模式，则输出以下三行：
  - 二进制“模拟”行
  - 十进制“模拟”行，已注释掉
  - 十进制“实际”行，已注释掉

- 如果未指定 binary 模式，则输出两行：
  - “模拟”行
  - “实际”行，已注释掉

下面是 pubtune 数据库中有关 authors 表的表级统计信息的示例：

```
Table owner:                          "dbo"
Table name:                             "authors"

Statistics for table:                   "authors"

    Partition count:                    3

Statistics for partition:               "authors_1376004902"
    Data page count:                    74
    Empty data page count:              0
    Data row count:                     1666.0000000000000000
    Forwarded row count:                0.0000000000000000
    Deleted row count:                  0.0000000000000000
    Data page CR count:                 10.0000000000000000
    OAM + allocation page count:        3
    First extent data pages:            0
    Data row size:                      85.2623049219687914
    Parallel join degree:               0.0000000000000000
    Unused page count:                  5
OAM page count:                         1

Derived statistics:
    Data page cluster ratio:            1.0000000000000000
    Space utilization:                  0.9521597490347491
    Large I/O efficiency:               1.0000000000000000
```

除表和索引统计信息外，simulate optdiag 输出还包括：

- 分区表的分区信息。如果是分区表，则会为该表的每个分区输出模拟信息和实际信息。“Pages in the largest partition”行不适用：

```
        Pages in largest partition:      390.0000000000000000 (simulated)
#       Pages in largest partition:       390.0000000000000000 (actual)
```

- 并行处理配置参数设置：

```
    Configuration Parameters:
        Number of worker processes:      20 (simulated)
#       Number of worker processes:      20 (actual)
        Max parallel degree:             10 (simulated)
#       Max parallel degree:             10 (actual)
        Max scan parallel degree:        3 (simulated)
```

```
#    Max scan parallel degree:          3 (actual)
```

- 关于缺省数据高速缓存与特定数据库或特定表及其索引使用的高速缓存的配置信息。如果 tempdb 绑定到一个高速缓存，也会包括该高速缓存的配置信息。以下是 pubtune 数据库所使用的高速缓存的输出样本：

```
Configuration for cache:              "pubtune_cache"

     Size of 2K pool in Kb:           15360 (simulated)
#    Size of 2K pool in Kb:           15360 (actual)
     Size of 4K pool in Kb:           0 (simulated)
#    Size of 4K pool in Kb:           0 (actual)
     Size of 8K pool in Kb:           0 (simulated)
#    Size of 8K pool in Kb:           0 (actual)
     Size of 16K pool in Kb:          0 (simulated)
#    Size of 16K pool in Kb:           0 (actual)
```

若要测试查询如何使用 16K 缓冲池，请更改要读取的上述模拟统计信息值：

```
Configuration for cache:              "pubtune_cache"

     Size of 2K pool in Kb:           10240 (simulated)
#    Size of 2K pool in Kb:           15360 (actual)
     Size of 4K pool in Kb:           0 (simulated)
#    Size of 4K pool in Kb:           0 (actual)
     Size of 8K pool in Kb:           0 (simulated)
#    Size of 8K pool in Kb:           0 (actual)
     Size of 16K pool in Kb:          5120 (simulated)
#    Size of 16K pool in Kb:           0 (actual)
```

装载和使用模拟统计信息的要求

若要使用模拟统计信息，请在运行查询前发出 set statistics simulate on。

精确地模拟查询：

- 对表使用同一锁定方案和分区方式
- 重新创建表上的所有触发器并使用同一参照完整性约束
- 设置任何非缺省高速缓存策略和任何非缺省并发优化值
- 将数据库和对象绑定到正在模拟的环境中使用的高速缓存
- 包括要测试的批处理中影响查询优化的任何 set 选项（如 set parallel_degree）

- 创建查询中所使用的任何视图
- 如果查询使用了游标，则使用这些游标
- 如果正在模拟过程执行，则使用存储过程

可将模拟统计信息装载到原始数据库中，或装载到为对查询执行“假设”分析而单独创建的数据库中。

在原始数据库中使用模拟统计信息

将统计信息装载到原始数据库中时，会将其放到系统表中的单独行中；它们不会覆盖现有非模拟统计信息。模拟统计信息仅用于 set statistics simulate 命令在其中有效的会话。

虽然模拟统计信息不用于优化其它会话的查询，但使用模拟统计信息执行任何查询可能会生成针对实际表和索引的次优查询计划，而且执行这些查询可能会对系统上的其它查询产生负面影响。

在另一个数据库中使用模拟统计信息

将统计信息装载到为对查询执行“假设”分析而单独创建的数据库中时，您必须确保：

- 在输入文件中命名的数据库存在；它最小可以是 2MB。由于数据库名称只在输入文件中出现一次，因此可更改数据库名称，例如，将 production 更改为 test_db。
- 输入文件中包含的所有表和索引存在。这些表不需要包含数据。
- 在输入文件中命名的所有高速缓存存在。它们可以为高速缓存大小的最小允许值 512K，且只有一个 2K 缓冲池。模拟统计信息提供有关缓冲池配置的信息。

删除模拟统计信息

装载模拟统计信息会将描述高速缓存配置的行添加到 master 数据库中的 sysstatistics 表中。若要删除这些统计信息，请使用 delete shared statistics。此命令对装载了模拟统计信息的数据库中的统计信息没有影响。

如果已将模拟统计信息装载到包含实际表和索引统计信息的数据库中，可使用以下方法之一删除模拟统计信息：

- 对表使用 delete statistics（这将删除所有统计信息），然后运行 update statistics，仅重新创建非模拟统计信息，或

- 使用 optdiag（无 simulate 模式）将统计信息复制出来；然后对表运行 delete statistics，并使用 optdiag（无 simulate 模式）将统计信息复制进去。

使用模拟统计信息运行查询

通过 set statistics simulate on 可使用模拟统计信息优化查询：

```
set statistics simulate on
```

在大多数情况下，您还需要使用 set showplan on。

如果已将模拟统计信息装载到生产数据库中，当使用模拟统计信息运行查询时可以使用 set noexec on，这样，就不会根据与实际表及索引不匹配的统计信息来执行查询。这使您能够检查 showplan 的输出，而不影响生产系统的性能。

模拟统计信息的 *showplan* 消息

启用 set statistics simulate 且有可用的模拟统计信息时，showplan 会输出：

```
Optimized using simulated statistics.
```

如果在其上执行模拟测试的服务器将并行查询选项的值设置为小于模拟值，则 showplan 输出会先显示使用模拟统计信息的计划，然后显示调整过的查询计划。如果打开 set noexec，则不显示调整过的计划。

包含双引号的字符数据

在表示字符和 datetime 列的直方图中，所有列数据都包含在双引号中。如果列本身包含双引号字符，optdiag 会显示两个引号。例如，如果列值为：

```
a quote "mark"
```

optdiag 将显示：

```
"a quote" "mark"
```

The only other special character in optdiag 输出中的唯一其它特殊字符是井号 (#)，它在输入模式中表示将忽略该行中井号 (#) 后面的所有字符，但当井号 (#) 作为列名或列值的一部分出现在引号中时除外。

SQL 命令对统计信息的影响

systabstats 和 sysstatistics 中存储的信息受数据定义语言 (DDL) 的影响。一些数据修改语言 (DML) 还影响 systabstats。表 2-8 总结了 DDL 对 systabstats 和 sysstatistics 表的影响。

表 2-8：DDL 对 systabstats 和 sysstatistics 的影响

命令	对 systabstats 的影响	对 sysstatistics 的影响
alter table...lock	更改这些值以反映对表以及索引结构和大小的更改。 将表和索引从所有页锁定更改为仅数据锁定时，会为表将聚簇索引的 indid 设置为 0，并为索引插入新行。	和 create index 一样，如果从所有页锁定更改为仅数据锁定（或反之），则不会对仅数据锁定方案之间的更改产生任何影响。
alter table，添加、删除或修改列定义	一些 alter table 参数（例如，使用 drop column、使用 add 添加非空列或使用 modify 减少可变长度列的长度）需要表的数据副本。其它 alter table 操作是通过更新系统目录信息来完成的。 如果此更改影响行的长度，并且 alter table 必须复制表，则 alter table 参数会更改这些值，以反映对表和索引结构和大小的更改。 如果 alter table 参数不执行数据复制，则不会执行任何更改。	如果更改需要数据复制，则 alter table 会重新构建所有索引，并与 create clustered 或 create non-clustered index 具有相同的效果。alter table 不影响非索引列上保留的统计信息行，但它确实会删除要删除的列上的统计信息行。 如果更改不执行数据复制，则不会执行任何更改。
create table	为表增加一行。如果约束创建索引，请参见下面的 create index 命令。	为表增加一行。如果约束创建索引，请参见下面的 create index 命令。
create clustered index	对于所有页锁定表，将 indid 更改为 1 并更新与索引有关的列；对于 DOL 锁定表，将添加一个新行。	为尚未包含的列增加行；为已包含的列更新行。
create nonclustered index	为非聚簇索引增加行。	为尚未包含的列增加行；为已包含的列更新行。
delete statistics	无影响。	删除表的所有行或只删除指定列的行。
drop index	删除非聚簇索引的行和 DOL 锁定表上聚簇索引的行。对于所有页锁定表上的聚簇索引，将 indid 设置为 0 并更新列值。	不删除索引列的实际统计信息。这样，优化程序就可以继续使用此信息。 删除非聚簇索引的模拟统计信息。对于所有页锁定表上的聚簇索引，更改包含模拟表数据的行中索引 ID 的值。
drop table	删除表的所有行。	删除表的所有行。

命令	对 systabstats 的影响	对 sysstatistics 的影响
reorg	如果用于时间限制，则更新重新启动点；如果页计数发生更改，则更新页数和集群比；根据所用的选项影响其它值，如空白页、转移或删除的行计数。	rebuild 选项将重新创建索引。
truncate table	重新设置值以反映空表。保留某些值（如行长度）。	无影响；这允许重新装载被截断的表，而无需重新运行 update statistics。
update statistics（对空表运行更新统计信息不会影响系统表。）		
table_name	更新表的值和指定表上的所有索引的值。	更新表上每个索引前导列的直方图；更新所有索引和索引前缀子集的密度。
index_name	更新指定索引的值。	更新指定索引的前导列的直方图；更新索引前缀子集的密度。
column_name(s)	无影响。	更新或创建列的直方图，以及更新或创建指定列的前缀子集的密度。
update index statistics		
table_name	更新表的值以及指定表上所有索引中的所有列的值。	更新表上每个索引的所有列的直方图；更新所有索引和索引前缀子集的密度。
index_name	更新指定索引的值	更新指定索引的所有列的直方图；更新索引的前缀子集的密度。
update all statistics		
table_name	更新表的值和指定表中所有列的值。	更新表上所有列的直方图；更新所有索引和索引前缀子集的密度。

查询处理如何影响 *systabstats*

数据修改可能会影响 systabstats 表中的许多值。为提高性能，管家杂事任务会定期在内存中更改这些值，并刷新到 systabstats。

若要直接查询 systabstats，可使用 sp_flushstats 将内存中的统计信息刷新到 systabstats。例如，若要刷新 titles 表和该表上任何索引的统计信息，可使用以下命令：

```
sp_flushstats titles
```

如果不提供表名，sp_flushstats 将刷新当前数据库中所有表的统计信息。

注释　某些统计信息，尤其是集群比，可能不太精确。因为在由数据修改查询进行更改期间，并未记录所有页的分配和解除分配。运行 update statistics 或 create index 可更正任何不一致的情况。

使用 *sp_showoptstats* 查看统计信息和直方图

Adaptive Server 包括 sp_showoptstats，其功能与 optdiag 独立实用程序类似，用于提取和在 XML 文档中显示系统表（如 systabstats 和 sysstatistics）中各种类型的数据对象的统计信息和直方图。语法为：

```
sp_showoptstats [dbname[.owner[.[table_name] ] ] ], [.column],
    [option]
```

请参见《Adaptive Server 参考手册：过程》中的 sp_showoptstats。

索引

符号

英文

C

D

F

G

H

J

K

L

M

P

Q

S

T

W

X

Y

Z