



性能和调优指南

Sybase IQ 15.3

文档 ID: DC00953-01-1530-01

最后修订日期: 2011 年 5 月

版权所有 © 2011 Sybase, Inc. 保留所有权利。

除非新版本或技术声明中另有说明, 否则本出版物适用于 Sybase 软件及所有后续版本。本文档中的信息如有更改, 恕不另行通知。本出版物中描述的软件按许可证协议提供, 其使用或复制必须符合协议条款。

要订购其它文档, 美国和加拿大的客户请拨打客户服务部门电话 (800) 685-8225 或发传真至 (617) 229-9845。

持有美国许可证协议的其它国家/地区的客户可通过上述传真号码与客户服务部门联系。所有其它国际客户请与 Sybase 子公司或当地分销商联系。仅在软件的定期发布日期提供升级内容。未经 Sybase, Inc. 的事先书面许可, 不得以任何形式、任何手段 (电子的、机械的、手工的、光学的或其它手段) 复制、传播或翻译本出版物的任何部分。

可在 <http://www.sybase.com/detail?id=1011207> 上的 Sybase 商标页中查看 Sybase 商标。Sybase 和列出的标记均是 Sybase, Inc. 的商标。® 表示已在美国注册。

SAP 和此处提及的其它 SAP 产品与服务及其各自的徽标是 SAP AG 在德国和世界各地其它几个国家/地区的商标或注册商标。

Java 和基于 Java 的所有标记都是 Sun Microsystems, Inc. 在美国和其它国家/地区的商标或注册商标。

Unicode 和 Unicode 徽标是 Unicode, Inc. 的注册商标。

本书中提到的所有其它公司和产品名均可能是与之相关的相应公司的商标。

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

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

目录

管理系统资源	1
性能注意事项	1
优化内存使用	1
分页增加可用内存	2
监控交换的实用程序	2
服务器内存	3
管理缓冲区高速缓存	3
确定缓冲区高速缓存的大小	4
设置缓冲区高速缓存大小	6
指定页大小	7
为大量用户进行优化	8
特定于平台的内存选项	10
进程线程模型	12
平衡 I/O	13
原始 I/O (在 UNIX 操作系统上)	13
使用磁盘条带化	14
内部条带化	15
使用多个文件	16
策略文件位置	16
用于插入、删除和同步的工作空间	18
设置保留空间选项	19
用于调优资源使用的选项	19
限制并发查询	19
设置可用 CPU 数	20
限制查询使用的临时数据库空间	20
限制按行返回的查询	21
强制游标为非滚动游标	22
限制游标数	22
限制语句数	23
预取高速缓存页	23
针对典型使用进行优化	24

控制预取行数	24
改进资源使用的其它方式	25
管理 Multiplex 数据库中的磁盘空间	25
使用逻辑服务器管理 Multiplex 资源	25
查询服务器间的负载平衡	25
限制数据库访问	26
磁盘高速缓存	26
索引提示	27
管理数据库大小和结构	28
非规范化以提高性能	29
使用 UNION ALL 视图以便更快装载	29
优化引用 UNION ALL 视图的查询	30
管理 UNION ALL 视图性能	31
网络性能	31
监控和调优性能	35
查看 Sybase IQ 环境	35
使用存储过程获取信息	35
分析数据库过程	36
监控性能统计信息	40
在服务器级监控性能	40
内存使用统计信息	41
高速缓存统计信息	41
CPU 使用率统计信息	43
线程统计信息	43
连接统计信息	44
请求统计信息	45
事务统计信息	45
存储 I/O 统计信息	46
数据库空间使用情况统计信息	47
网络统计信息	47
监控缓冲区高速缓存	48
启动缓冲区高速缓存监控器	49
输出选项	50
在监控器运行时检查结果	59

停止缓冲区高速缓存监控器	60
检查并保存监控器结果	60
缓冲区高速缓存结构	61
避免缓冲区管理器抖动	62
监控 Windows 系统上的分页	63
监控 UNIX 系统上的分页	63
缓冲区高速缓存监控清单	64
监控 CPU 使用的系统实用程序	67
优化查询和删除	69
实现结构化查询的提示	69
GROUP BY 与 UNION ALL 配合使用时对查询性能的影响	69
增强 ORDER BY 查询性能	71
提高了子查询性能	71
使用高速缓存方法	72
计划查询	72
查询评估选项	72
查询树	74
使用查询计划	74
控制查询处理	75
设置查询时间限制	75
设置查询优先级	76
设置查询优化选项	76
设置用户提供的条件提示	77
监控负载	78
优化删除操作	79
HG 删除操作	79
WD 删除操作	80
TEXT 删除操作	81
在 32 位 Windows 系统上调优服务器	83
一般性能准则	83
文件系统	83
最大化网络应用程序数据吞吐量	83
监控性能	84

目录

虚拟地址空间	84
NTFS 高速缓存	85
插入和查询	86
备份操作	86
索引	89

管理系统资源

对硬件和软件配置进行调优可以提供更好的性能和更快的查询。

性能注意事项

性能通常用响应时间和吞吐量来衡量。良好的设计和索引策略可以实现最大的性能增益。

响应时间

响应时间是完成单个任务所用的时间。多个因素会影响到响应时间：

- 减少争用和等待时间，特别是磁盘 I/O 等待时间
- 使用更快的组件
- 减少需要资源的时间（提高并发性）

吞吐量

吞吐量是指在固定时间段内完成的工作量。吞吐量通常以每秒的事务数 (tps) 进行度量，也可以按每分钟、每小时、每天等进行度量。

设计注意事项

最大的性能增益可以通过建立良好的设计和选择正确的索引策略来实现。

其它考查事项（如硬件和网络分析）可确定安装中的瓶颈。

另请参见

- 优化内存使用（第 1 页）
- 进程线程模型（第 12 页）
- 平衡 I/O（第 13 页）
- 用于调优资源使用的选项（第 19 页）
- 改进资源使用的其它方式（第 25 页）
- 索引提示（第 27 页）
- 管理数据库大小和结构（第 28 页）
- 使用 UNION ALL 视图以便更快装载（第 29 页）
- 网络性能（第 31 页）

优化内存使用

了解 Sybase® IQ 分配内存的方式可以帮助您让系统达到最佳性能。

分页增加可用内存

虽然分页可以增加可用内存量，但应避免页面交换或最大程度地减少页面交换以实现良好的内存管理。

当系统没有足够内存时，性能会严重降低。如果发生这种情况，则需要找到某种方式，使更多内存可用。分配给 Sybase IQ 的内存越多越好。

但是，系统的内存量始终存在固定限制，因此有时操作系统只能将一部分数据放在内存中，而将其余部分放在磁盘上。如果操作系统为了满足一次内存请求而必须转向磁盘并检索任何数据，则这种情况称为分页或交换。好的内存管理的主要目标就是避免分页或交换，或者使其最大程度减少分页或交换。

最频繁使用的操作系统文件是交换文件。当内存用完时，操作系统就会将内存页交换到磁盘，以便为新数据腾出空间。当再次调用被交换的页时，就会交换其它页，并返回所需的内存页。对于有很高磁盘使用率的用户，这会非常耗时。通常，应当尝试组织内存，以避免交换，从而最大程度减少对操作系统文件的使用。

为了最大程度利用物理内存，Sybase IQ 对所有数据库读取和写入均使用缓冲区高速缓存。

注意： 您的磁盘上的交换空间必须至少有足够的空间，才能容纳所有物理内存。

另请参见

- 监控交换的实用程序（第 2 页）
- 服务器内存（第 3 页）
- 管理缓冲区高速缓存（第 3 页）
- 确定缓冲区高速缓存的大小（第 4 页）
- 设置缓冲区高速缓存大小（第 6 页）
- 指定页大小（第 7 页）
- 为大量用户进行优化（第 8 页）
- 特定于平台的内存选项（第 10 页）

监控交换的实用程序

使用操作系统上的实用程序来了解您的系统是否分页过多。

使用 UNIX **vmstat** 命令、UNIX **sar** 命令或 Windows 任务管理器可以获取正在运行的进程数以及页出和交换次数的统计信息。使用此信息可以了解系统是否分页过多，然后进行任何必要的调整。可能需要将交换文件放在特殊的快速磁盘上。

另请参见

- 分页增加可用内存（第 2 页）
- 服务器内存（第 3 页）
- 管理缓冲区高速缓存（第 3 页）

- 确定缓冲区高速缓存的大小 (第 4 页)
- 设置缓冲区高速缓存大小 (第 6 页)
- 指定页大小 (第 7 页)
- 为大量用户进行优化 (第 8 页)
- 特定于平台的内存选项 (第 10 页)

服务器内存

Sybase IQ 为缓冲区、事务、数据库和服务器分配堆内存。也许还会使用共享内存，但数量上少得多。

在操作系统级别，Sybase IQ 服务器内存由堆内存组成。在大多数情况下，您不需要关注 Sybase IQ 所使用的内存是堆内存还是共享内存。所有内存分配都会自动处理。但是，在运行 Sybase IQ 之前，可能需要确保正确配置操作系统内核以使用共享内存。

管理 *Multiplex* 内存

Multiplex 中的每个服务器都可以在其自己的主机上，也可以与其它服务器共享主机。相同系统上的两个或更多服务器不会比处理相同负载的单个组合服务器消耗更多 CPU 时间，但分开的服务器可能比单个组合服务器需要更多物理内存，因为每个服务器都不会与其它任何服务器共享所使用的内存。

警告！ 在 UNIX 系统上终止进程可能导致留下信号或共享内存而不是将其自动清理掉。在 UNIX 上关闭 Sybase IQ 服务器的正确方式是使用 **stop_iq** 实用程序。

另请参见

- 分页增加可用内存 (第 2 页)
- 监控交换的实用程序 (第 2 页)
- 管理缓冲区高速缓存 (第 3 页)
- 确定缓冲区高速缓存的大小 (第 4 页)
- 设置缓冲区高速缓存大小 (第 6 页)
- 指定页大小 (第 7 页)
- 为大量用户进行优化 (第 8 页)
- 特定于平台的内存选项 (第 10 页)

管理缓冲区高速缓存

对于大多数数据库来说，缺省高速缓存大小（主高速缓存为 16MB，临时高速缓存为 12MB）太低。请尽可能多地为 IQ 主缓冲区高速缓存和临时缓冲区高速缓存分配内存。

Sybase IQ 对用于缓冲区高速缓存的内存量超出了用于任何其它用途的内存量。Sybase IQ 有两个缓冲区高速缓存，一个用于 IQ 存储，另一个用于临时存储。它将这两个缓冲区高速缓存用于所有数据库 I/O 操作一分页、插入数据库以及备份和还原。只

要数据在内存中，它就会存储在这两个高速缓存的某一个之中。所有用户连接都共享这些缓冲区高速缓存。Sybase IQ 会跟踪与每个连接关联的数据。

另请参见

- 分页增加可用内存 (第 2 页)
- 监控交换的实用程序 (第 2 页)
- 服务器内存 (第 3 页)
- 确定缓冲区高速缓存的大小 (第 4 页)
- 设置缓冲区高速缓存大小 (第 6 页)
- 指定页大小 (第 7 页)
- 为大量用户进行优化 (第 8 页)
- 特定于平台的内存选项 (第 10 页)

确定缓冲区高速缓存的大小

根据多个因素，计算正确的缓冲区高速缓存大小。

- 系统上的物理内存总量
- Sybase IQ、操作系统和其它应用程序执行其任务时需要此内存的大小是多少
- 是否正在执行装载、查询或同时执行这两个操作
- schema 配置和查询负载

操作系统及其它应用程序

大多数操作系统会将大部分可用内存用于文件系统缓冲。了解适用于您的操作系统的缓冲策略以避免过度分配内存。

有关与 Sybase IQ 一起运行的应用程序的内存要求，请参见应用程序和操作系统文档。

另请参见

- 内存开销 (第 4 页)
- 主缓冲区高速缓存和临时缓冲区高速缓存 (第 6 页)

内存开销

在确定操作系统和其它应用程序需要多少物理内存之后，请计算 Sybase IQ 需要多少剩余内存。

原始分区与文件系统

对于 UNIX 系统，使用文件系统而不是原始分区的数据库可能需要剩余内存的另外 30% 来处理操作系统的文件缓冲。在 Windows 上，应当通过将

OS_FILE_CACHE_BUFFERING 设置为 “OFF” (对于新数据库，这是缺省值) 来禁用文件系统高速缓存。

多用户数据库访问

对于数据库的多用户查询，Sybase IQ 需要对每个“活动”用户使用大约 10MB。活动用户被定义为同时访问或查询数据库的用户。例如，30 个用户可能连接到 Sybase IQ，但仅 10 个左右用户可能正在任何时间同时频繁使用数据库。

用于线程堆栈的内存

处理线程需要少量的内存。您使用的 Sybase IQ 处理线程越多，需要的内存就越多。

-iqmt 服务器开关控制 Sybase IQ 的线程数。 **-iqtss** 和 **-gss** 服务器开关控制为每个线程分配的堆栈内存量。为 IQ 堆栈分配的总内存量大致等于： $(-\text{gn} * (-\text{gss} + -\text{iqtss})) + (-\text{iqmt} * -\text{iqtss})$ 。

如果拥有大量用户，则处理线程所需的内存将增加。 **-gn** 开关控制数据库服务器可以并发执行的任务（用户和系统请求）数。 **-gss** 开关在一定程度上控制执行这些任务的服务器执行线程的堆栈大小。 IQ 使用以下公式计算这些工作线程的堆栈大小： $(-\text{gss} + -\text{iqtss})$ 。

线程总数（**-iqmt** 加 **-gn**）不得超过平台允许的数目。

其它内存使用

所有命令和事务都会使用一些内存。以下操作是除了前面讨论的内存使用以外最重要的内存使用：

- 备份。用于备份的虚拟内存量是在创建数据库时所指定的 **IQ PAGE SIZE** 函数。它大约是 $2 * \text{CPU 数} * 20 * (\text{IQ PAGE SIZE}/16)$ 。在某些平台上，通过在 **BACKUP** 命令中调整 **BLOCK FACTOR**，可能能够提高备份性能，但增加 **BLOCK FACTOR** 也会增加所使用的内存数量。
- 数据库检验和修复。在检查整个数据库时，**sp_iqcheckdb** 过程将在启动任何处理之前打开所有 Sybase IQ 表及其各自的字段和索引。根据 Sybase IQ 表的数量以及这些表中的累积列数和索引数，**sp_iqcheckdb** 可能需要非常少或非常多的虚拟内存。若要限制需要的内存量，请使用 **sp_iqcheckdb** 选项，以检查或修复单个索引或表。
- 删除泄漏块。删除漏洞操作也需要打开所有 Sybase IQ 表、文件和索引，因此它会使用像 **p_iqcheckdb** 在检查整个数据库时所使用的那么多虚拟内存。它使用 Sybase IQ 临时缓冲区高速缓存来跟踪所使用的块。

另请参见

- 操作系统及其它应用程序（第 4 页）
- 主缓冲区高速缓存和临时缓冲区高速缓存（第 6 页）

主缓冲区高速缓存和临时缓冲区高速缓存

高速缓存大小的一般准则是 40% 用于主缓冲区高速缓存，而 60% 用于临时缓冲区高速缓存。请基于此准则监控服务器性能，然后根据需要调整高速缓存大小。

缓冲区高速缓存和物理内存

用于 Sybase IQ 主缓冲区高速缓存和临时缓冲区高速缓存的内存总量，加上 Sybase IQ 内存开销，以及用于操作系统和其它应用程序的内存，不得超过系统的物理内存。

为了获得最佳性能，请尽可能多地为 IQ 的主缓冲区高速缓存和临时缓冲区高速缓存分配内存。例如，如果计算机上有 4GB 的物理内存可供 Sybase IQ 使用，则可拆分该内存量供主缓冲区高速缓存和临时缓冲区高速缓存使用。

其它注意事项

缓冲区高速缓存大小要求取决于使用情况。若要获得最高性能，请在插入、查询数据库和混合使用之间更改设置。但是，在混合使用环境中，为了重置缓冲区高速缓存选项而要求所有用户退出数据库并不始终可行。在这些情况下，可能需要照顾到装载或查询性能。

注意：

- 这些准则假定系统上每次有一个活动数据库。如果有多个活动数据库，则需要在期望使用的数据库中进一步拆分剩余的内存。
 - 在某些 UNIX 平台上，可能需要设置其它服务器开关，以便使更多内存可用作缓冲区高速缓存。
-

另请参见

- 操作系统及其它应用程序（第 4 页）
- 内存开销（第 4 页）

设置缓冲区高速缓存大小

Sybase IQ 最初分别将主缓冲区高速缓存和临时缓冲区高速缓存的大小设置为 16MB 和 12MB。请更改主缓冲区高速缓存和临时缓冲区高速缓存的缺省大小，以适应您的应用程序。

缓冲区高速缓存大小设置

使用这些选项可以设置缓冲区高速缓存大小。

表 1. 更改缓冲区高速缓存大小的设置

方法	何时使用它	设置一直有效的时间
-iqmc 和 -iqtc 服务器开关	推荐的方法。当数据库和服务器未运行时设置高速缓存大小。允许高速缓存大小 >4GB。 对于 64 位平台，或在将高速缓存大小数据库选项设置为超过系统容量的情况下，尤其有用。	从服务器启动时到服务器停止 重新启动服务器以更改缓冲区高速缓存大小。 -iqmc 和 -iqtc 服务器启动选项仅在服务器正在运行时保持有效，因此每次重新启动服务器时都需要包括它们。

指定页大小

页大小和缓冲区高速缓存大小会影响数据库的内存使用和磁盘 I/O 吞吐量。

注意： 页大小无法更改，并且决定某些数据库对象的大小上限以及是否可以使用 LOB 功能。

页大小

Sybase IQ 以页为单位与内存双向交换数据。创建数据库时，应为目录存储和 IQ 存储指定单独的页大小。临时存储的页大小与 IQ 存储相同。

目录存储的页大小不会对性能产生真正的影响。缺省值 4096 字节应当足够。IQ 页大小确定其它两个性能因素：缺省 I/O 传输块大小和数据库最大数据压缩。

块大小

所有 I/O 均以块为单位发生。在创建 Sybase IQ 数据库时，系统将设置这些块的大小；除非重新创建数据库，否则不能更改它。缺省情况下，IQ 页大小确定 I/O 传输块大小。例如，128KB 的缺省 IQ 页大小导致 8192 字节的缺省块大小。通常，Sybase IQ 使用此缺省块大小与页大小的比率，但它还会考虑其它因素。

缺省块大小应当导致大多数系统的 I/O 传输率和磁盘空间使用情况达到最佳平衡。但是，与性能相比，它更有益于节省空间。如果缺省块大小表现不佳，则可以按照每页中不能有少于 2 个和多于 16 个块的约束，将它设置为 4096 和 32,768 之间的 2 的任何次幂。在某些情况下，可能需要显式设置块大小：

- 对于使用磁盘阵列的原始磁盘安装，更大的块可能获得更好的磁盘空间使用性能。
- 对于文件系统安装，若要通过磁盘空间优化性能，IQ 块大小应当大于或等于操作系统的本机块大小（如果有）。如果 IQ 块大小与文件系统的块大小匹配，则可能获得更好的 I/O 率。

数据压缩

Sybase IQ 压缩所有数据。压缩量基于 IQ 页大小确定。

节省内存

如果计算机没有足够的内存，请减小缓冲区高速缓存大小。但是，如果缓冲区高速缓存减少得太多，则由于缓冲区不足，可能使数据装载或查询效率降低或无法完成。

另请参见

- 分页增加可用内存（第 2 页）
- 监控交换的实用程序（第 2 页）
- 服务器内存（第 3 页）
- 管理缓冲区高速缓存（第 3 页）
- 确定缓冲区高速缓存的大小（第 4 页）
- 设置缓冲区高速缓存大小（第 6 页）
- 为大量用户进行优化（第 8 页）
- 特定于平台的内存选项（第 10 页）

为大量用户进行优化

为了支持最大数量的用户，您可能需要增加临时数据库空间、调整操作系统参数以及更改启动参数。

新连接和现有连接的相对优先级

如果 Sybase IQ 正忙于处理已连接的用户，则响应新连接请求的速度可能会变慢。在极端情况（例如，测试脚本在服务器忙于执行插入时触发数百个循环连接）下，新连接可能会出现连接请求超时的情况。在此情况下，服务器可能看似已停机，而实际上它只是非常繁忙。遇到此行为的用户应尝试再次连接并应考虑增加连接超时参数。

启动选项

对于有大量用户的操作，请使用以下启动选项。

-gm

设置缺省连接数。

-gm #_connections_to_support

尽管此选项表示服务器将支持的连接总数，但并非所有连接都将在任一时间处于活动状态。

-iqgovern

为在 ***-iqgovern*** 队列中等待的每个查询分配优先级。

-iqgovern #_ACTIVE_queries_to_support

-iqgovern 限制同时执行的最大查询数。如果已提交查询的用户数超过了 ***-iqgovern*** 限制，则新查询将排队，直到某个活动查询完成。

-iqgovern 的最佳值取决于查询的性质、CPU 数和 Sybase IQ 缓冲区高速缓存的大小。缺省值为 $2 * numCPU + 100$ 。在有很多连接用户的情况下，您可能发现将此选项设置为 $2 * numCPU + 4$ 可以提供更好的吞吐量。

-gn

设置多个用户同时运行时用于目录存储和连接的执行线程数。

-gn *number of tasks (both user and system requests) that the database server can execute concurrently*

-gn 的正确值取决于 **-gm** 的值。**start_iq** 实用程序将计算 **-gn** 并进行相应设置。将 **-gn** 设置得太低会阻止服务器正确操作。建议不要将 **-gn** 设置为超过 480。

-c

设置目录存储高速缓存大小。

-c catalog_store_cache_size

目录存储缓冲区高速缓存还是目录存储的常规内存池。若要以 MB 为单位指定，请使用 **-c nM** 格式，例如 **-c 64M**。Sybase 建议使用下列值：

表 2. 目录缓冲区高速缓存设置

针对此数目的用户	在这些平台上	将 -c 设置为此最小值或更高
最高 1000	仅 64 位	64MB
最高 200	64 位	48MB (64 位系统的 start_iq 缺省值)；大量用户可能从 64MB 中受益
最高 200	32 位	32MB (32 位系统的 start_iq 缺省值)

-cl 和 -ch

设置目录存储高速缓存大小的上限 (**-ch**) 和下限 (**-cl**)。

-cl *minimum cache size* **-ch** *maximum cache size*

如果标准目录高速缓存大小太小，请设置 **-cl** 和 **-ch** 参数。在 32 位平台上，请尝试以下设置：

-cl 128M -ch 256M

不得将 **-c** 和 **-ch** 或 **-cl** 用于同一配置文件或命令行中。有关相关信息，请参见 **-ch cache-size** 选项。

警告！ 若要显式控制目录存储高速缓存大小，必须在配置文件 (.cfg) 中或在用于启动服务器的 UNIX 命令行上执行以下操作之一（但不要同时执行这两项操作）：

- 设置 **-c** 参数

- 使用 **-ch** 和 **-cl** 参数设置目录存储高速缓存大小的特定上限和下限。如不按上述说明指定参数组合，则可能产生意外结果。

-iqmt

设置处理线程数。

如果对于 **-gm** 设置而言，**-iqmt** 设置得太低，则可能会导致线程匮乏。

特定于平台的内存选项

在所有 64 位平台上，可用内存总量仅受限于系统的虚拟内存。在 32 位平台上，存在一些限制。

32 位系统上的可用内存

在 32 位平台上，请参见下表。

表 3. 32 位平台上的总计可用内存

平台	总计可用内存
RedHat Linux 2.1	大约有 1.7GB 可供 Sybase IQ 使用
RedHat Linux 3.0	大约有 2.7GB 可供 Sybase IQ 使用
Windows 2000/2003/XP ^a	有 2.75GB 可供 Sybase IQ 使用

^a 您需要 Windows 2000 Advanced Server 或 Datacenter Server、Windows Server 2003 Standard、Enterprise 或 Datacenter Edition 或者 Windows XP Professional，才能获得这么多内存，并且必须设置 /3GB 开关。如果没有开关，则限制为 2GB。此数量是可用于进程的总计内存。在 Windows 服务器上，即使有 /3GB 设置，缓冲区高速缓存的总计大小也不得超过 2GB。有关详细信息，请参见《适用于 Windows 的安装和配置指南》。

由于在 Sybase IQ 服务器上使用了虚拟内存模式，虚拟内存碎片可能会导致 Windows 平台上的处理工作大幅增加。为了减少发生此情况的可能性，Sybase IQ 支持在 Windows XP 和 Windows Server 2003 上使用 Microsoft 的低碎片堆 (LFH)。

连系内存池

在 HP 和 Sun 平台上，可以指派指定数量的内存作为连系内存。连系内存是锁定到物理内存中的共享内存。内核不能从物理内存中对该部分内存进行分页。

当其它应用程序正同时运行于相同计算机上时，连系内存可能提高 Sybase IQ 性能。但是，为 Sybase IQ 分配专用连系内存会使它对该计算机上的其它应用程序不可用。

若要仅在这些 UNIX 平台上创建连系内存池，请指定 **-iqwmem** 命令行开关，以指示连系内存的 MB 数。（除在 Sun 上以外，您必须是 **root** 用户，才能设置 **-iqwmem**。）在 64 位平台上，**-iqwmem** 的唯一上限是计算机的物理内存。

例如，在有 14GB 内存的计算机上，您可能能够留出 10GB 连系内存。为此，请指定：

```
-iqwmem 10000
```

注意：仅当您有足够的内存可专用于为此目的而指定的数量时，才使用 **-iqwmem**。否则，可能导致性能大幅下降。

- 在 Sun Solaris 上，**-iqwmem** 始终提供连系内存。
- 在 HP 上，如果您作为 root 启动服务器，则 **-iqwmem** 提供连系内存。如果启动服务器时您不是 root，则它提供非连系内存。此行为可能在未来版本中有更改。

其它应用程序和数据库的影响

服务器内存来自于所有应用程序和数据库使用的内存池。如果尝试同时在相同计算机上运行多个服务器或多个数据库，或者有其它正在运行的应用程序，则可能需要减少服务器所请求的内存数量。

还可以发出 UNIX 命令 **ipcs -mb** 以查看实际段数。

对 HP 内存问题进行故障排除

在 HP-UX 上，检查 **maxdsiz_64bit** 内核参数的值。此参数限制在 64 位 HP 处理器上可用于 Sybase IQ 的虚拟内存量。有关建议值，请参见《安装和配置指南》。

控制文件系统缓冲

在某些文件系统上，您可以打开或关闭文件系统缓冲。关闭文件系统缓冲通常会减少分页并提高性能。

若要对现有数据库的 IQ 主数据库空间禁用文件系统缓冲，请发出以下语句：

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING = OFF
```

若要对现有数据库的 IQ 临时 dbspace 禁用文件系统缓冲，请发出以下语句：

```
SET OPTION "PUBLIC".OS_FILE_CACHE_BUFFERING_TEMPDB = OFF
```

只能为 PUBLIC 组设置此选项。请关闭数据库并重新启动，以使更改生效。

此直接 I/O 性能选项仅在 Sun Solaris UFS、Linux、Linux IBM、AIX 和 Windows 文件系统上可用。此选项对于 HP-UX 和 HP-UXi 不起作用，并且不影响原始磁盘上的数据库。在 Linux，直接 I/O 在内核版本 2.6.x 中受支持。

要在 Linux 内核版本 2.6 和 AIX 上启用直接 I/O，还要将环境变量 **IQ_USE_DIRECTIO** 设置为 1。缺省情况下，直接 I/O 在 Linux 内核版本 2.6 和 AIX 上处于禁用状态。**IQ_USE_DIRECTIO** 对 Sun Solaris 和 Windows 不起作用。

注意：

- 在 Linux 内核版本 2.4 上，Sybase IQ 不支持直接 I/O。如果在 Linux 内核版本 2.4 上设置了 **IQ_USE_DIRECTIO** 环境变量，则 Sybase IQ 服务器不启动。系统会报告错误 “Error: Invalid Block I/O argument, maybe <pathname> is a directory, or it exceeds maximum file size limit for”

the platform, or trying to use Direct IO on unsupported OS”。

- Solaris 没有内核参数来约束其文件系统缓冲区高速缓存的大小。一段时间以后，文件系统缓冲区高速缓存将增长并转移 IQ 缓冲区高速缓存页，从而导致过度的操作系统分页活动，并使 Sybase IQ 性能降低。
 - Windows 可以通过耗用文件系统来调整分页算法的偏差，以使应用程序受益。为了提高 Sybase IQ 性能，建议使用此偏差。
-

另请参见

- 启用 Java 的数据库的选项（第 12 页）

启用 Java 的数据库的选项

设置 **JAVA_HEAP_SIZE** 选项可以防止失控的 Java 应用程序使用过多内存。

SET OPTION 命令的 **JAVA_HEAP_SIZE** 选项可以设置基于每个连接而分配给 Java 应用程序的那部分内存的最大大小（字节）。通常，基于每个连接的内存分配由所分配的 Java 变量和 Java 应用程序堆栈空间的用户工作集组成。Java 应用程序在连接上执行时，每个连接分配会来自数据库服务器的固定高速缓存，所以，防止无法控制的 Java 应用程序占用过多内存是很重要的。

另请参见

- 控制文件系统缓冲（第 11 页）

进程线程模型

Sybase IQ 使用操作系统内核线程，以获得最佳性能。

轻型进程是内核所支持的基础控制线程。操作系统决定哪些轻型进程 (LWP) 应当运行在哪个处理器上，以及何时运行。它不知道用户线程是什么，但如果用户线程正在等待或能够运行，则它会知道。

操作系统内核将 LWP 调度到 CPU 资源上。它使用它们的调度类和优先级。每个 LWP 均由内核单独进行调度，执行独立的系统调用，调用独立的页错误并在多处理器系统上并行运行。

单个、高度线程化的进程服务于所有 Sybase IQ 用户。Sybase IQ 基于该连接所完成的处理类型、可用线程总数和各种选项设置，将不同数目的内核线程分配给每个用户连接。

线程不足错误

如果查询的线程不足，Sybase IQ 将产生以下错误：

Not enough server threads available for this query

这种情况可能是临时性的。当其它某些查询完成时，线程将变为可用，在您下一次发出查询时，查询可能成功。如果此情况仍然存在，则可能需要重新启动服务器并指定更多 Sybase IQ 线程。还有可能是 **-iqmt** 为连接数设置的值过低。

用于管理线程使用的 **Sybase IQ** 选项

- 使用服务器启动选项 **-iqmt** 可以设置最大线程数。缺省值是从连接数和 CPU 数计算得到的，通常已足够。
- 使用服务器启动选项 **-iqtss** 可以设置内部执行线程的堆栈大小。缺省值通常已足够，但如果复杂查询返回一条指示堆栈的深度已超过此限制的错误，则可能会增加。
- 使用 **SET OPTION MAX_IQ_THREADS_PER_CONNECTION** 命令可以为单个用户设置最大线程数。**SET OPTION MAX_IQ_THREADS_PER_TEAM** 可以设置可用于一组线程的线程数。

使用这些选项可以控制特定操作消耗的资源数量。例如，DBA 可在发出 **INSERT**、**LOAD**、**BACKUP** 或 **RESTORE** 命令之前设置此选项。

另请参见

- 性能注意事项（第 1 页）
- 优化内存使用（第 1 页）
- 平衡 I/O（第 13 页）
- 用于调优资源使用的选项（第 19 页）
- 改进资源使用的其它方式（第 25 页）
- 索引提示（第 27 页）
- 管理数据库大小和结构（第 28 页）
- 使用 UNION ALL 视图以便更快装载（第 29 页）
- 网络性能（第 31 页）

平衡 I/O

在系统上平衡 I/O。它解释如何在单独的磁盘上使用磁盘分条以及如何在单独的磁盘上找到文件，以获得更好的性能。还会讨论对消息日志文件大小的控制。

原始 I/O (在 UNIX 操作系统上)

您可在原始设备或文件系统文件上创建数据库或数据库空间。

大多数 UNIX 文件系统将磁盘划分成固定大小分区。磁盘分区的访问模式通常有两种：文件系统模式（例如，通过 UFS 文件系统）或原始模式。原始模式执行无缓冲的 I/O，通常使用每个读取或写入系统调用与设备进行双向数据传输。UFS 是缺省的 UNIX 文件系统，也是缓冲 I/O 系统，它每次在缓冲区中收集数据，直到它可以传输整个缓冲区。

您可在原始设备或文件系统文件上创建数据库或数据库空间。Sybase IQ 自动按照您指定的路径名确定它是原始分区还是文件系统文件。原始分区可以是任意大小。

有关详细信息，请参见《系统管理指南第一卷》中的“使用数据库对象”。

另请参见

- 使用磁盘条带化（第 14 页）
- 内部条带化（第 15 页）
- 使用多个文件（第 16 页）
- 策略文件位置（第 16 页）
- 用于插入、删除和同步的工作空间（第 18 页）
- 设置保留空间选项（第 19 页）

使用磁盘条带化

磁盘条带化是一种将单个文件的数据分散到多个磁盘驱动器的常规方法。与单个磁盘相比，条带化磁盘大幅提高了性能。

磁盘条带化将一个或多个物理磁盘（或磁盘分区）组合成单个逻辑磁盘。分条磁盘跨组件物理设备对 I/O 传输进行拆分，从而并行执行这些传输。

磁盘分条允许将数据块放在不同磁盘上。第一个块位于第一个驱动器上。第二个块位于第二个驱动器上，依此类推。所有驱动器均已使用后，此过程再次循环，并使用驱动器上的其它块。磁盘分条的实际效果是数据随机分布在多个磁盘驱动器上。针对存储于分条磁盘上的文件的随机操作趋向于使在分条集中的所有驱动器保持同等繁忙状态，从而使每秒的磁盘操作总数最大化。这是数据库环境中非常有效的技术。

在 *UNIX* 上设置磁盘条带化

提供分条磁盘的 *UNIX* 系统提供了相应的实用程序，用于将物理磁盘配置成分条设备。有关详细信息，请参见 *UNIX* 或存储管理系统文档。

在 *Windows* 上设置磁盘条带化

在 *Windows* 上，可通过适当的 SCSI-2 磁盘控制器来使用硬件磁盘条带化。如果计算机不支持硬件分条，但您有可用于数据库的多个磁盘，则可以使用 *Windows* 分条将磁盘 I/O 分散到多个磁盘上。请使用磁盘管理工具设置 *Windows* 分条。

磁盘条带化的建议

- 将条带化文件系统中的各个磁盘分散在几个磁盘控制器上。不要用太多磁盘使磁盘控制器饱和。有关详细信息，请参见硬件文档。
- 不要将磁盘与较慢的设备（例如磁带驱动器或 CD-ROM）放在相同控制器上。这会减慢磁盘控制器的速度。
- 在带区中每个服务器 CPU 分配 4 个磁盘。

- 单个磁盘必须是相同的设备。也就是说，它们必须有相同大小、相同格式，并且通常是相同品牌。如果布局不同，则通常使用最小那个磁盘的大小，并会浪费其它磁盘空间。而且，通常会使用最慢磁盘的速度。
- 避免将文件条带化磁盘用于任何其它用途。例如，不要将文件分条磁盘用作交换分区。
- 永远不要将包含根文件系统的磁盘用作分条设备的一部分。
- 使用原始分区以达到最高性能。

注意：为了在装载数据时获得最佳结果，请将数据转储到位于条带化磁盘上的平面文件，然后用 **LOAD TABLE** 命令将数据读取到 Sybase IQ 中。

另请参见

- 原始 I/O (在 UNIX 操作系统上) (第 13 页)
- 内部条带化 (第 15 页)
- 使用多个文件 (第 16 页)
- 策略文件位置 (第 16 页)
- 用于插入、删除和同步的工作空间 (第 18 页)
- 设置保留空间选项 (第 19 页)

内部条带化

使用磁盘条带化，您可以一次利用多个磁盘主轴，并提供并行磁盘写入的速度。

Sybase IQ 提供磁盘条带化选项，而不使用第三方软件。如果您已经有通过第三方软件和硬件实现的磁盘分条解决方案，则应当使用该方法。可以通过为 **CREATE DBSPACE** 命令指定 **STRIPING ON** 选项启用磁盘分条。

打开或关闭磁盘条带化

在创建数据库空间时更改缺省条带化：

```
SET OPTION "PUBLIC".DEFAULT_DISK_STRIPIING = { ON | OFF }
```

对于所有平台，**DEFAULT_DISK_STRIPIING** 选项的缺省值均为 **ON**。当磁盘条带化为 **ON** 时，传入数据将分散到有可用空间的所有数据库空间。当磁盘条带化为 **OFF** 时，数据库空间 (磁盘段) 在逻辑文件上从前面填充，每次填充一个磁盘段。

如果更改 **DEFAULT_DISK_STRIPIING** 的值，将影响不指定分条首选项的所有后续 **CREATE DBSPACE** 操作。

磁盘条带化处于打开状态时，可以使用 **ALTER DBSPACE DROP** 命令从数据库空间中删除文件。但是，在删除数据库空间之前，必须使用 **sp_iqemptyfile** 存储过程在该数据库空间中重新定位所有数据。由于磁盘条带化使数据分散在多个文件中，因此 **sp_iqemptyfile** 进程可能需要重新定位多个表和索引。请使用 **sp_iqdbspaceinfo** 和 **sp_iqdbspace** 存储过程来确定哪些表和索引驻留在数据库空间上。

另请参见

- 原始 I/O (在 UNIX 操作系统上) (第 13 页)
- 使用磁盘条带化 (第 14 页)
- 使用多个文件 (第 16 页)
- 策略文件位置 (第 16 页)
- 用于插入、删除和同步的工作空间 (第 18 页)
- 设置保留空间选项 (第 19 页)

使用多个文件

在数据库空间中使用多个文件允许跨多个操作系统文件或分区分布 Sybase IQ 和临时数据。多个文件可以提高吞吐量并缩短数据库空间的平均延迟时间。

何时添加文件

如果可以，请在创建 `dbspace` 时分配所有文件以确保数据分布均匀。

如果在今后添加文件，请使用 **ALTER DBSPACE** 命令将其它文件添加到数据库空间中。Sybase IQ 将跨新旧数据库空间对新数据进行条带化。分条可能是均匀的，也可能是不均匀的，具体取决于更新的类型。由于版本控制而“翻转”的页数会对是否重新均衡分条产生重要影响。

另请参见

- 原始 I/O (在 UNIX 操作系统上) (第 13 页)
- 使用磁盘条带化 (第 14 页)
- 内部条带化 (第 15 页)
- 策略文件位置 (第 16 页)
- 用于插入、删除和同步的工作空间 (第 18 页)
- 设置保留空间选项 (第 19 页)

策略文件位置

提供其它存储资源以改进随机和顺序文件磁盘 I/O。

通过增加专门用于存储随机访问的文件的磁盘驱动器数，并由此增加每秒对这些文件执行的操作数，可以提高与这些文件相关的性能。随机文件包括用于 IQ 存储、临时存储、目录存储的文件，以及程序（包括 Sybase IQ 可执行文件、用户及存储过程和应用程序）和操作系统文件。

相反，通过将这些文件放在专用磁盘驱动器上，由此消除与其它进程的争用，可以提高与顺序访问的文件相关的性能。顺序文件包括事务日志文件和消息日志文件。

若要避免磁盘瓶颈，请遵循以下建议：

- 将随机磁盘 I/O 和顺序磁盘 I/O 分开。此外，为了达到最佳性能，每个数据库空间应仅使用物理设备（磁盘或硬件 RAID 集）的一个分区。

- 将 Sybase IQ 数据库 I/O 与其它数据库（例如 Adaptive Server® Enterprise）中的代理表的 I/O 隔离。
- 将事务日志和消息日志放在单独的磁盘上，与 IQ 存储、目录存储和临时存储分离，并与任何代理数据库（例如 Adaptive Server Enterprise）分离。
- 将数据库文件、临时数据库空间和事务日志文件放在与数据库服务器相同的物理计算机上。

事务日志

事务日志文件包含允许 Sybase IQ 从系统故障恢复的信息。还需要对事务日志进行审计。此文件的缺省文件扩展名为 .log。

若要移动或重命名事务日志文件，请使用事务日志实用程序 (**dblog**)。请参见《实用程序指南》>“数据库管理实用程序”>“事务日志实用程序 (dblog)”。

警告！ Sybase IQ 事务日志文件与大多数关系数据库事务日志文件不同。如果由于某个原因而丢失了数据库文件，则会失去数据库（除非丢失的是日志文件）。但是，如果有合适的备份，则可以重新装载数据库。

截断事务日志

Sybase IQ 在事务日志中记录从系统故障恢复所需的信息。尽管所记录的信息对于每个提交的事务很小，但事务日志的大小却持续增长。如果系统中有大量会更改数据的事务，则在一段时间后，日志可能增长到非常大。

何时截断日志应当由支持 Sybase IQ 系统的 DBA 负责，并取决于日志文件的增长情况以及现场的操作过程。

下表显示了截断 Sybase IQ 事务日志的方法。

表 4. 截断事务日志

如果您的数据库……	请使用以下方法……	有关详细信息，请参见……
已停止	-m 开关，该开关导致事务日志在所有数据库的每个检查点之后被截断	“截断停止的数据库的事务日志”
正在运行	带有 -xo 开关或 -r 开关的 dbbackup 命令行实用程序	《实用程序指南》>“数据库管理实用程序”>“备份实用程序 (dbbackup)”

另请参见

- 消息日志（第 18 页）
- 截断停止的数据库的事务日志（第 18 页）

截断停止的数据库的事务日志

使用 **-m** 服务器启动开关可以截断数据库事务日志。建议不要将 **-m** 服务器启动开关永久保持已设置状态。此开关应当仅用于启动 Sybase IQ 以执行事务日志截断。如何执行该操作由 DBA 负责，但以下过程提供了一个建议。

1. 以一个将文件标识为日志截断配置设置的名称，创建服务器开关 .cfg 文件的副本，并编辑此文件副本以添加 **-m** 开关。
2. 使用包含 **-m** 选项的配置文件启动 Sybase IQ。请注意，这时不允许有任何用户访问或事务发生。
3. 关闭 Sybase IQ，并使用未设置 **-m** 选项的配置文件重新启动。

另请参见

- 截断事务日志（第 17 页）
- 消息日志（第 18 页）

消息日志

每个数据库都有消息日志文件。此文件的缺省名为 `dbname.iqmsg`。消息日志文件实际是在创建数据库后首次启动该数据库时创建的。

缺省情况下，Sybase IQ 会在消息日志文件中记录所有消息，包括错误、状态和插入通知消息。可以在 **LOAD** 和 **INSERT** 语句中使用参数关闭通知消息。

在某些站点上，由于插入次数、**LOAD** 选项和 **NOTIFY_MODULUS** 数据库选项设置或某些其它条件，消息日志文件常常会迅速增长。Sybase IQ 允许您回卷消息日志或设置最大文件大小以及在活动 IQ 消息日志已满时存档日志文件，从而限制此文件的大小。

有关设置最大日志文件大小、存档消息日志文件以及启用消息日志回卷的信息，请参见《系统管理指南第一卷》>“Sybase IQ 系统管理概述”中的“消息记录”。

另请参见

- 截断事务日志（第 17 页）
- 截断停止的数据库的事务日志（第 18 页）

用于插入、删除和同步的工作空间

插入或删除数据以及同步连接索引时，Sybase IQ 在 IQ 存储中需要一些工作空间。在需要此空间的事务提交后，将收回此空间以用于其它目的。

通常，只要在 IQ 存储中维护合理百分比的可用空间，就会有足够的可用空间。但是，对于某些删除（具体取决于数据的大小及其在数据库页中的分布情况），您可能需要大量的工作空间。在即将删除数据库的主要部分且数据稀疏分散在很多页中的情况下，可以将数据库的大小临时加倍。

长事务（如在一个事务中对大表逐行执行更新）可能会消耗大量主数据库空间。系统将保存每次更新的版本控制信息，直到提交事务。

另请参见

- 原始 I/O（在 UNIX 操作系统上）（第 13 页）
- 使用磁盘条带化（第 14 页）
- 内部条带化（第 15 页）
- 使用多个文件（第 16 页）
- 策略文件位置（第 16 页）
- 设置保留空间选项（第 19 页）

设置保留空间选项

两个数据库选项 **MAIN_RESERVED_DBSPACE_MB** 和 **TEMP_RESERVED_DBSPACE_MB** 控制 Sybase IQ 为某些操作而保留的空间数量。

有关详细信息，请参见《系统管理指南第一卷》>“使用数据库对象”中的“IQ 主存储区和 IQ 临时存储空间管理”。

另请参见

- 原始 I/O（在 UNIX 操作系统上）（第 13 页）
- 使用磁盘条带化（第 14 页）
- 内部条带化（第 15 页）
- 使用多个文件（第 16 页）
- 策略文件位置（第 16 页）
- 用于插入、删除和同步的工作空间（第 18 页）

用于调优资源使用的选项

调优资源以创建更快的查询

限制并发查询

设置 **-iqgovern** 开关可以指定特定服务器上的并发查询数量。这与连接数不同，连接数是由许可证控制的。

通过指定 **-iqgovern** 开关，您可以帮助 IQ 优化将缓冲区数据分页调度到磁盘上，避免过量使用内存。**-iqgovern** 的缺省值是 $(2 \times \text{CPU 数}) + 10$ 。您可能需要试验确定理想值。对于使用大量活动连接的站点，请尝试将 **-iqgovern** 设置得略低些。

另请参见

- 设置可用 CPU 数（第 20 页）
- 限制查询使用的临时数据库空间（第 20 页）

- 限制按行返回的查询 (第 21 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

设置可用 CPU 数

设置 **-iqnumbercpus** 启动开关可以指定可用 CPU 数。此参数会出于资源规划目的覆盖实际的 CPU 数目。

建议仅在以下情况下使用 **-iqnumbercpus** 开关：

- 在配有 Intel® CPU 且已启用超线程的计算机上
- 在已使用操作系统实用程序将 Sybase IQ 限制为仅可使用其内 CPU 子集的计算机上

请参见《系统管理指南第一卷》>“运行 Sybase IQ”中的“设置 CPU 数”。

另请参见

- 限制并发查询 (第 19 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

限制查询使用的临时数据库空间

设置 **QUERY_TEMP_SPACE_LIMIT** 可以指定拒绝查询前临时空间的最大估计量。

如果查询的估计临时空间使用量超出指定大小，则 **QUERY_TEMP_SPACE_LIMIT** 选项将导致查询被拒绝。缺省情况下，对查询使用的临时存储没有限制。

Sybase IQ 会估计完成该查询所需的临时空间。如果估计值超过当前 **QUERY_TEMP_SPACE_LIMIT** 设置，Sybase IQ 将返回错误：

```
Query rejected because it exceeds total space resource limit
```

如果此选项设置为 0 (缺省值)，则表示没有限制，并且没有任何查询基于其临时空间需求被拒绝。

若要限制每个连接的实际临时存储使用情况, 请为所有 DML 语句设置 `MAX_TEMP_SPACE_PER_CONNECTION` 选项, 包括查询。`MAX_TEMP_SPACE_PER_CONNECTION` 监控和限制语句的实际运行时临时存储使用情况。如果连接超过 `MAX_TEMP_SPACE_PER_CONNECTION` 选项设置的限额, 则系统会返回错误, 当前语句将回退。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制按行返回的查询 (第 21 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

限制按行返回的查询

设置 `QUERY_ROWS_RETURNED_LIMIT` 选项的值可以防止优化程序拒绝具有大结果集的查询。

`QUERY_ROWS_RETURNED_LIMIT` 选项告诉查询优化程序拒绝可能消耗太多资源的查询。如果查询优化程序估计来自查询的结果集将超过此选项的值, 则它会拒绝该查询, 并返回消息:

```
Query rejected because it exceed resource: Query_Rows_Returned_Limit
```

如果使用此选项, 则设置它, 以便它仅拒绝消耗大量资源的查询。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

强制游标为非滚动游标

消除返回很大结果集的查询中的临时存储节点以提高性能。

使用没有声明宿主变量的滚动游标时, **Sybase IQ** 将创建临时存储节点, 用于缓冲查询结果。此存储与临时存储缓冲区高速缓存分隔开。通过临时存储节点, 可以在应用程序搜索结果集时有效向前和向后滚动。

然而, 如果查询返回很多 (数百万) 行输出, 并且如果您的应用程序主要执行向前滚动操作, 则临时存储节点的内存要求可能会降低查询性能。若要提高性能, 请通过发出以下命令消除临时存储节点:

SET TEMPORARY OPTION FORCE_NO_SCROLL_CURSORS = 'ON'

注意: 如果您的应用程序通常执行向后滚动, 则将 **FORCE_NO_SCROLL_CURSORS** 选项设置为 **ON** 实际上可能会降低查询性能, 因为如果不存在临时高速缓存, 系统将强制 **Sybase IQ** 对每个向后滚动重新执行查询。

如果您的应用程序很少执行向后滚动, 请将 **FORCE_NO_SCROLL_CURSORS = 'ON'** 作为永久 **PUBLIC** 选项。它将使用更少内存, 并提高查询性能。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

限制游标数

设置 **MAX_CURSOR_COUNT** 选项可以防止单个连接占用过多的可用内存或 CPU 资源。

MAX_CURSOR_COUNT 选项限制连接一次可以使用的最大游标数。缺省值为 50。如果将此选项设置为 0, 则允许无限数目的游标。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)

- 强制游标为非滚动游标 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

限制语句数

设置 **MAX_STATEMENT_COUNT** 选项可以限制连接可以使用的预准备语句数。

MAX_STATEMENT_COUNT 选项限制连接一次可以使用的最大预准备语句数。如果服务器在任一时间对任一连接需要支持的预准备语句数超过缺省值 (50) , 则您可以将 **MAX_STATEMENT_COUNT** 选项设置为更高的值。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

预取高速缓存页

设置 **PREFETCH_BUFFER_LIMIT** 和 **BT_PREFETCH_MAX_MISS** 选项可以控制预取内存行为。

PREFETCH_BUFFER_LIMIT 选项定义 Sybase IQ 可用于执行预取 (对数据库页的提前读取) 的高速缓存页数。此选项的缺省值为 0。仅当 Sybase 技术支持部门建议这样做时, 才应设置此选项。

BT_PREFETCH_MAX_MISS 选项可以确定是否继续为给定查询预取页。如果使用 HG 索引的查询的运行速度比预期慢很多, 请尝试逐步增大此选项的值。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)

- 限制语句数 (第 23 页)
- 针对典型使用进行优化 (第 24 页)
- 控制预取行数 (第 24 页)

针对典型使用进行优化

设置 **USER_RESOURCE_RESERVATION** 选项可以调整用于当前用户数的内存使用量。

Sybase IQ 可跟踪打开的游标数，并相应地分配内存。在某些环境中，可以设置 **USER_RESOURCE_RESERVATION** 选项，以调整 Sybase IQ 认为当前正在使用该产品的当前最少游标数，并因此更节省地从临时高速缓存分配内存。

应当仅在仔细的分析显示实际需要它之后，才设置此选项。如果需要设置此选项，请与 Sybase 技术支持部门联系，以了解详细信息。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)
- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 控制预取行数 (第 24 页)

控制预取行数

设置 **PrefetchRows** 和 **PrefetchBuffer** 参数可以在特定条件下提高游标性能。

预取可以提高仅提取 **relative 1** 或 **relative 0** 的游标的性能。两个连接参数允许您更改游标预取缺省值。**PrefetchRows (PROWS)** 可以设置预取行数；**PrefetchBuffer (PBUF)** 可以设置此连接可用来存储预取行的内存。在某些情况下，增加预取的行数有可能提高性能：

- 应用程序较少使用绝对读取来读取许多行 (几百或更多)。
- 应用程序以高速率提取行，客户端和服务器位于同一台计算机上或通过快速网络连接。
- 客户端/服务器通信是通过速度较慢的网络 (如拨号链接或广域网) 进行的。

另请参见

- 限制并发查询 (第 19 页)
- 设置可用 CPU 数 (第 20 页)
- 限制查询使用的临时数据库空间 (第 20 页)
- 限制按行返回的查询 (第 21 页)

- 强制游标为非滚动游标 (第 22 页)
- 限制游标数 (第 22 页)
- 限制语句数 (第 23 页)
- 预取高速缓存页 (第 23 页)
- 针对典型使用进行优化 (第 24 页)

改进资源使用的其它方式

可以通过多种方式调整系统以获得最高性能或更好利用磁盘空间。

管理 Multiplex 数据库中的磁盘空间

让用户定期提交其当前事务，并允许写入服务器删除旧的表版本以释放磁盘块。

由于任何服务器上的任何用户都可能处在可能需要旧版表的事务中，因此 Sybase IQ 不能删除这些旧版表。在 Multiplex 数据库中同时发生表更新和查询时，Sybase IQ 可能因此消耗非常多的磁盘空间。所消耗的空间数量取决于数据和索引的性质以及更新率。

通过允许写入服务器删除查询不再需要的过时版本，可以释放磁盘块。所有服务器上的所有用户都应当定期提交其当前事务，以允许恢复旧版表。服务器可能始终正常工作，并完全可用。sp_iqversionuse 存储过程可用于显示远程服务器的版本使用情况。

另请参见

- 查询服务器间的负载平衡 (第 25 页)
- 限制数据库访问 (第 26 页)
- 磁盘高速缓存 (第 26 页)

使用逻辑服务器管理 Multiplex 资源

逻辑服务器让您能够最有效地管理 Multiplex 资源的使用。使用逻辑服务器可将几组不同的 Multiplex 服务器分配到不同应用程序，以满足它们各自的性能要求。

在 Multiplex 中，每个连接都在单个逻辑服务器环境下运行。当您将查询提交到 Multiplex 服务器时，其执行可能分布到一个或多个 Multiplex 服务器，具体取决于连接的逻辑服务器的配置。若要动态调整分配给逻辑服务器的资源，请从逻辑服务器添加或删除 Multiplex 服务器，以满足它所服务的应用程序的不断变化的需求。

查询服务器间的负载平衡

要使用 IQ 网络客户端在 Multiplex 查询服务器间平衡查询负载，需要一个能够将客户端连接调度到池内计算机的中间系统。

若要使用此方法，请通过使用此中间负载平衡系统的 IP 地址和端口号、常规服务器名称以及设置为 NO 的 VerifyServerName 连接参数，在客户端系统上创建特殊 ODBC

DSN。当客户端使用此 DSN 进行连接时，负载平衡器将与它确认的有最低负载的计算机建立连接。

有关如何定义在查询服务器负载平衡中使用的 ODBC DSN 的详细信息，请参见《系统管理指南第一卷》>“连接和通信参数”中的“VerifyServerName 参数 [Verify]”。

另请参见

- 管理 Multiplex 数据库中的磁盘空间（第 25 页）
- 限制数据库访问（第 26 页）
- 磁盘高速缓存（第 26 页）

限制数据库访问

安排在需求较低时进行更新以提高查询性能。

为了获得更好的查询性能，如果可能，请将数据库设置为只读，或使其在低峰使用时间进行重要更新。当您正在表中进行插入或删除时，Sybase IQ 允许多个查询用户从该表读取数据。但是，在数据库并发更新期间，性能会降低。

另请参见

- 管理 Multiplex 数据库中的磁盘空间（第 25 页）
- 查询服务器间的负载平衡（第 25 页）
- 磁盘高速缓存（第 26 页）

磁盘高速缓存

保持大量内存处于活动备用状态，从而使对真实内存的需求与对磁盘数据的需求保持平衡。

磁盘高速缓存是操作系统临时用于存储磁盘块副本的内存。所有基于磁盘读取和写入的文件系统通常都会通过磁盘高速缓存执行操作。从应用程序的角度来看，涉及磁盘高速缓存的所有读取和写入等同于实际的磁盘操作。

操作系统使用两种不同方法向磁盘高速缓存分配内存：固定和动态。在固定分配中使用预置数量的内存；通常会留出 10-15% 的内存分配量。操作系统通常使用 LRU（最近使用最少的）算法来管理此工作空间。对于动态分配，操作系统将在运行时确定磁盘高速缓存分配量。目标是保持尽可能多的内存处于活动备用状态，从而使对磁盘数据的需求与对真实内存的需求保持平衡。

另请参见

- 管理 Multiplex 数据库中的磁盘空间（第 25 页）
- 查询服务器间的负载平衡（第 25 页）
- 限制数据库访问（第 26 页）

索引提示

选择正确的列索引类型，以加快查询运行速度。

Sybase IQ 自动提供某些索引（在所有列上用于优化投影的索引，以及用于 **UNIQUE**、**PRIMARY KEYS** 和 **FOREIGN KEYS** 的 **HG** 索引）。虽然这些索引对某些用途来说非常有用，但您可能需要其它索引以尽快处理特定查询。

索引顾问

索引顾问可以在查询中生成优化器何时会从一个或多个列的其它索引中受益的消息。

若要激活索引顾问，请将 **INDEX_ADVISOR** 选项设置为 **ON**。如果不启用查询计划，则消息作为查询计划的一部分或作为单独消息显示在消息日志 (.iqmsg) 中，并且输出采用 **OWNER.TABLE.COLUMN** 格式。有关详细信息，请参见《参考：语句和选项》的“数据库选项”中的“**INDEX_ADVISOR** 选项”。

LF 或 HG 索引

如果列未使用枚举 FP 存储，请考虑对连接查询中的 **WHERE** 子句引用的分组列创建一个 **LF** 或 **HG** 索引。Sybase IQ 优化程序可能需要枚举 FP 或 HG/LF 索引中的元数据以生成最佳查询计划。**HAVING** 子句中引用的非集合列也可能从 **LF** 或 **HG** 索引中受益以帮助优化查询。例如：

```
SELECT c.name, SUM(l.price * (1 - l.discount)) FROM customer c,
orders o, lineitem l WHERE c.custkey = o.custkey      AND o.orderkey =
l.orderkey      AND o.orderdate >= "1994-01-01"      AND o.orderdate <
"1995-01-01" GROUP by c.name HAVING c.name NOT LIKE "I%"      AND
SUM(l.price * (1 - l.discount)) > 0.50 ORDER BY 2 desc
```

添加索引会增加存储要求和装载时间。只有在对查询性能有益的情况下，才应添加索引。

使用连接索引

用户经常需要一次查看多个表中的数据。此数据可以在查询时连接，也可以提前通过创建连接索引进行连接。有时，可以通过为以一致方式连接的列创建连接索引，提高查询性能。

因为连接索引需要大量时间和空间进行装载，所以应当只为经常需要的连接创建连接索引。Sybase IQ 连接索引支持一对多和一对一连接关系。

为删除留出足够磁盘空间

删除数据行时，Sybase IQ 将为包含任何被删除数据的每个数据库页创建一个版本页。这些版本会一直保留，直到删除事务提交。因此，在删除数据时，可能需要添加磁盘空间。有关详细信息，请参见“重叠版本和删除”。

另请参见

- 性能注意事项（第 1 页）

- 优化内存使用 (第 1 页)
- 进程线程模型 (第 12 页)
- 平衡 I/O (第 13 页)
- 用于调优资源使用的选项 (第 19 页)
- 改进资源使用的其它方式 (第 25 页)
- 管理数据库大小和结构 (第 28 页)
- 使用 UNION ALL 视图以便更快装载 (第 29 页)
- 网络性能 (第 31 页)

管理数据库大小和结构

数据库大小在很大程度上取决于索引和数据量。创建索引有利于更快进行查询。删除不需要的对象可以释放磁盘空间并缩短装载时间。

数据量

若要控制在给定表中存储的数据量, 请消除不再需要的数据行。如果数据库包含最初来自 SQL Anywhere 数据库的数据, 则通过简单重播 Anywhere 删除操作, 可能能够删除不需要的数据; 命令语法是兼容的。可以对 Adaptive Server Enterprise 数据库中的数据执行相同操作, 因为 Sybase IQ 提供了 Transact-SQL 兼容性。

索引碎片

- 当索引页未被使用到最大量时, 就会出现内部索引碎片。
- 当删除行时, 会出现行碎片。删除行的整个页可以释放该页, 但如果页上的某些行是未使用的, 则未使用的空间会留在磁盘上。
- 对表执行的 DML 操作 (**INSERT**、**UPDATE**、**DELETE**) 可能导致索引碎片。

运行以下存储过程可以了解有关碎片问题的信息:

- **sp_iqrowdensity** 报告缺省索引级别的行碎片。请参见《参考: 构件块、表和过程》的“系统过程”中的“**sp_iqrowdensity** 过程”。
- **sp_iqindexfragmentation** 报告在补充索引中的内部碎片。请参见《参考: 构件块、表和过程》>“系统过程”中的“**sp_iqindexfragmentation** 过程”。

查看输出并决定是要重新创建索引、重组索引还是要重新生成索引。可以创建其它索引, 以补充缺省列索引。但是, 与从表中删除行时所需要的空间相比, 这些索引可能会使用更多空间。

最大程度地减少目录文件增长

目录文件的增长是正常情况, 它根据应用程序和目录内容不同而各不相同。.db 文件的大小对性能没有影响, 并且系统会根据需要重用 .db 文件中的可用页。

最大程度地减少目录文件增长:

- 避免对 **CREATE TABLE** 语句使用 **IN SYSTEM**
- 运行系统存储过程之后发出 **COMMIT** 语句
- 在长时间运行的事务期间发出 **COMMIT** 语句

非规范化以提高性能

尽管非规范化数据库可以提高性能，但也存在一些风险和缺点。

风险

非规范化只能在您全面了解应用程序时才能成功执行，并且仅当性能问题指示需要这样做时，才应当执行它。请考虑使数据保持最新所需的工作量。

这是决策支持应用程序（它们经常需要大量数据的摘要）与事务处理需求（它们执行不连续的数据修改）的差异的良好示例。非规范化通常以其它处理的开销来支持某些处理。

非规范化有潜在的数据完整性问题，必须仔细记录它们，并在应用程序设计中加以解决。

决定非规范化

请分析环境中应用程序的数据访问需求及其实际性能特征，包括：

- 关键查询是什么，以及期望的响应时间是多少？
- 它们使用什么表或列？每个访问有多少行？
- 常用排序顺序是什么？
- 并发预期是什么？
- 最常访问的表有多大？
- 任何进程都计算汇总吗？
- 是否应当创建连接索引以提高性能？

使用 **UNION ALL** 视图以便更快装载

UNION ALL 视图可以在为表中的所有行保留次级索引代价过高的情况下提高装载性能。

Sybase IQ 允许您将数据拆分至多个单独的基表中（例如，按日期）。数据将装载到这些较小的表中。然后，通过 **UNION ALL** 视图，将表重新连接在一起，形成一个逻辑整体，之后可以对其执行查询。

此策略可以提高装载性能，但是可能会对一些类型的查询的性能产生负面影响。针对单个基表或较小基表的 **UNION ALL** 视图执行的大多数类型的查询都具有大致相似的性能，只要视图定义满足所有约束。然而，与针对单个大型基表执行相比，针对 **UNION ALL** 视图执行时，一些类型的查询的执行速度可能会显著降低，特别是那些涉及 **DISTINCT** 或涉及多个连接列的连接的查询。在选择使用此策略之前，请确定装载性能的提高是否值得以降低应用程序的查询性能为代价。

UNION ALL 视图对管理员来说非常高效。例如，如果数据按月分区，则通过删除表和正确更新 **UNION ALL** 视图定义，就可以删除整个月份的数据值。您可以有年份、季度等的许多视图定义，而不必添加额外的日期范围谓词。

若要创建 **UNION ALL** 视图，请选择可以将基表划分到多个单独物理表中的逻辑方式。最常见的划分方式是按月划分。例如，若要创建包括第一季度的所有月份的视图，请输入：

```
CREATE VIEW
  SELECT * JANUARY
  UNION ALL
  SELECT * FEBRUARY
  UNION ALL
  SELECT * MARCH
  UNION ALL
```

在每个月，可以将数据装载到单个基表中（在此示例中为 **JANUARY**、**FEBRUARY** 或 **MARCH**）。在下一个月，则将数据装载到有相同列和相同索引类型的新表中。

有关语法的详细信息，请参见《参考：语句和选项》中的 **UNION** 操作。

注意： 不能在 **UNION ALL** 视图中执行 **INSERT...SELECT**。**UNION ALL** 运算符在此版本中不是完全并行的。使用这些运算符可能会限制查询并行度。

优化引用 **UNION ALL** 视图的查询

若要调整引用 **UNION ALL** 视图的查询的性能，请设置 **JOIN_PREFERENCE** 选项，该选项影响 **UNION ALL** 视图之间的连接。

UNION ALL 视图中的所有分区都必须有一组为工作优化定义的完整索引。有 **DISTINCT** 查询在使用 **UNION ALL** 视图时通常运行速度比基表更慢。

Sybase IQ 包括针对 **UNION ALL** 视图的专利优化技术，包括：

- 在 **UNION ALL** 视图上拆分 **GROUP BY**
- 将连接下推到 **UNION ALL** 视图中

仅当 **UNION** 满足以下所有约束时，才能将它视为分区表：

- 它只包含一个或多个 **UNION ALL**。
- **UNION** 的每个分支在其 **FROM** 子句中只有一个表，并且该表是物理基表。
- **UNION** 的任何分支都没有 **DISTINCT**、**RANK**、集合函数或 **GROUP BY** 子句。
- **UNION** 的每个分支中的 **SELECT** 子句中的每一项都是一列。
- 在第一个 **UNION** 分支的 **SELECT** 列表中的列的数据类型序列与 **UNION** 的每个随后分支中的序列相同。

另请参见

- 管理 **UNION ALL** 视图性能（第 31 页）

管理 UNION ALL 视图性能

结构化查询，以便首先评估 **DISTINCT** 运算符，然后再执行 **ORDER BY**，排序顺序为 **ASC**。

当 **ORDER BY** 为 **DESC** 时，系统不会应用某些特定优化（例如将 **DISTINCT** 运算符推入 **UNION ALL** 视图），因为评估 **UNION** 下的 **DISTINCT** 的优化不适用于 **DESC** 顺序。例如，下面的查询将对性能产生负面影响：

```
SELECT DISTINCT state FROM testVU ORDER BY state DESC;
```

要解决此性能问题，查询应先评估 **DISTINCT** 运算符，然后才能执行 **ORDER BY**，在这种情况下，排序顺序为 **ASC**，并且可应用优化：

```
SELECT c.state FROM (SELECT DISTINCT state
  FROM testVUA) c
ORDER BY c.state DESC;
```

另请参见

- 优化引用 UNION ALL 视图的查询（第 30 页）

网络性能

环境中的微小更改可以解决某些网络性能问题。

改进大型数据传输

减少总体吞吐量并增加平均响应时间：

- 如果可能，在下班时段执行大型传输。
- 在大型传输期间限制并发查询数。
- 不要在大型传输期间并发运行查询和插入。
- 使用存储过程以减少总通信量。
- 使用行缓冲，以便通过网络移动大批数据。
- 如果经常执行大型传输，请考虑安装适合这类传输的更好的网络硬件。例如：
 - 令牌环 - 在高负荷利用时段比以太网硬件有更好的响应能力。
 - 光纤 - 提供非常宽的带宽，但它通常特别昂贵，难以在整个网络中使用。
 - 单独的网络 - 可以用来处理最高容量工作站与服务器之间的网络通信量。

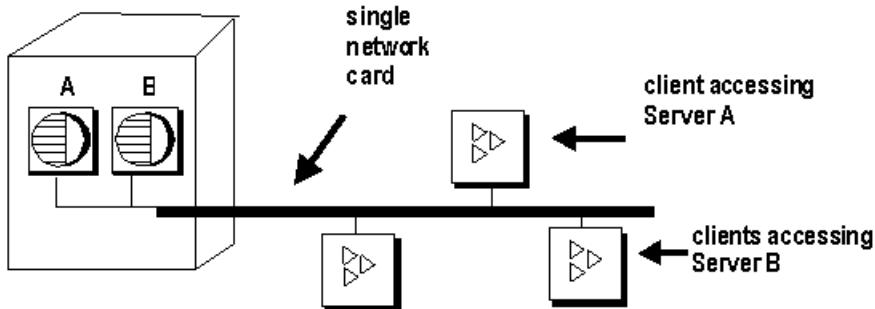
隔离重网络负荷用户

在图 12-4 的示例 A 中，访问两个不同数据库服务器的客户端使用一个网卡。也就是，访问服务器 A 和 B 的客户端必须争用网络并通过该网卡。在示例 B 中，访问服务器 A 的客户端使用的网卡与访问服务器 B 的客户端使用的网卡不同。

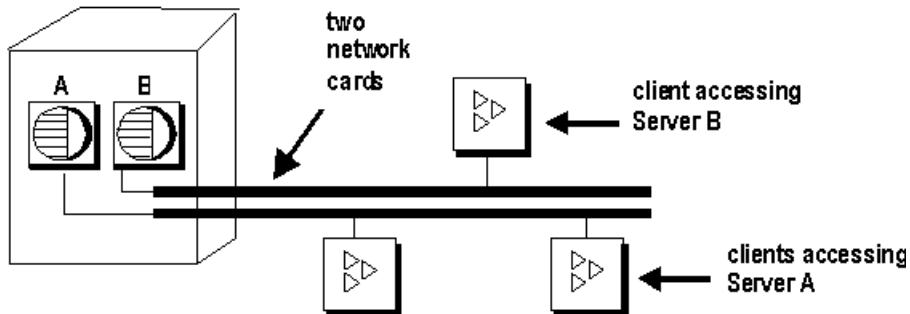
将数据库服务器放在不同计算机上甚至会更好。还可以将不同数据库的高负荷用户放在不同计算机上。

图 1：隔离重网络负载用户

Case A



Case B



将少量数据放在较小包中

如果通过网络发送少量数据，请保持缺省网络包大小较小（缺省值为 512 字）。-p 服务器启动选项允许您指定最大包大小。客户端应用程序还可能允许您设置包大小。

将大量数据放在较大包中

如果大多数应用程序发送和接收大量数据，请增加缺省网络包大小。这将导致更少（但更大）的传输。

在服务器级处理

在服务器级过滤尽可能多的数据。

另请参见

- 性能注意事项（第 1 页）
- 优化内存使用（第 1 页）
- 进程线程模型（第 12 页）
- 平衡 I/O（第 13 页）
- 用于调优资源使用的选项（第 19 页）

- 改进资源使用的其它方式 (第 25 页)
- 索引提示 (第 27 页)
- 管理数据库大小和结构 (第 28 页)
- 使用 UNION ALL 视图以便更快装载 (第 29 页)

监控和调优性能

介绍用于确定系统是否正在很好地利用可用资源的工具。

查看 Sybase IQ 环境

调优 Sybase IQ 性能的第一步是查看环境。

使用存储过程获取信息

几个存储过程可以显示数据库信息。

表 5. 名称统计信息

名称	描述
sp_iqconnection	显示有关连接和版本的信息，包括哪些用户使用临时数据库空间、哪些用户使用这些版本、连接在 Sybase IQ 中做什么、连接状态、数据库版本状态等等。 请参见《参考：构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqconnection 过程”
sp_iqcontext	按照连接跟踪和显示有关当前执行的语句的信息。 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcontext 过程”
sp_iqcheckdb	检查当前数据库的有效性。（可选）更正 dbspace 或数据库的分配问题。 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcheckdb 过程”
sp_iqdbstatistics	报告最近一次执行 sp_iqcheckdb 的结果。 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqdbstatistics 过程”
sp_iqdbsize	显示当前数据库的大小。 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqdbsize 过程”
sp_iqspaceinfo	显示数据库中每个对象的空间使用情况 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqspaceinfo 过程”

名称	描述
sp_iqstatus	显示有关数据库的杂项状态信息。 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqstatus 过程”
sp_iqtablesize	显示当前数据库中每个对象使用的块数和对象所在的 dbspace 的名称。 请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqtablesize 过程”

有关所有 Sybase IQ 存储过程的语法详细信息和示例，请参见《参考：构件块、表和过程》。

分析数据库过程

过程分析可以跟踪过程、触发器和其它系统事件的执行时间。使用 Sybase Central 中的分析可以确定数据库或数据库对象的性能问题。

查看过程分析统计信息

设置 Sybase Central 中的数据库分析选项可以监控存储过程、函数、事件和触发器的执行时间。

数据库分析信息属性

表 6. 数据库分析信息属性。

属性名称	描述
名称	列出对象的名称。
所有者	列出对象的所有者。
表	列出触发器属于哪个表（此列仅出现在数据库“分析”选项卡上）。
事件	显示系统触发器的触发器类型。可以是 Update 或 Delete。
类型	列出对象的类型，例如，过程。
执行次数	列出每个对象已被调用的次数。
毫秒数	列出各个对象的总执行时间。

数据库对象分析信息

数据库对象包括存储过程、函数、事件和触发器。 数据库对象分析信息属性逐行显示并汇总执行时间。

表 7. 对象分析信息属性。

属性名称	描述
调用次数	列出对象已被调用的次数。
毫秒	列出各个对象的总执行时间。
行号	列出每个过程行旁边的行号。
源	逐行显示 SQL 过程。

在 Sybase Central 中设置数据库分析属性

在 Sybase Central 中设置数据库分析属性需要 DBA 权限。 您的服务器必须正在运行，而且您必须连接到数据库。

1. 在 Sybase Central 中，右键单击数据库，然后选择“属性”。
2. 单击“分析设置”选项卡。
3. 有关其它分析选项，请参见联机帮助。

查看某一类数据库对象的分析信息

若要在 Sybase Central 中显示有关某一类数据库对象的分析信息，请单击父文件夹，然后查看对象的分析信息。

1. 打开对象文件夹：
 - 过程和函数
 - 事件
 - 触发器
 - 系统触发器
2. 单击右窗格中的“分析”选项卡。

有关该对象的分析信息随即出现在右窗格中的“分析”选项卡上。

查看特定数据库对象的分析信息

若要在 Sybase Central 中显示有关特定数据库对象的分析信息，请选择对象，然后查看对象的分析信息。

1. 打开对象文件夹：
 - 过程和函数
 - 事件

- 触发器
 - 系统触发器
2. 单击父文件夹中的对象。
3. 单击右窗格中的“分析”选项卡。

有关该对象的分析信息随即出现在右窗格中的“分析”选项卡上。

过程分析统计信息

设置数据库分析选项，然后使用分析选项来返回存储过程、函数、事件和触发器的性能统计信息。

sa_procedure_profile_summary

sa_procedure_profile_summary 是一个系统过程，可以报告已经在数据库中执行的所有过程、函数、事件或触发器的执行时间的摘要信息。此过程为这些对象提供的信息与 Sybase Central 中“分析”选项卡上的信息相同。

表 8. sa_procedure_profile_summary 统计信息

列名	描述
object_type	确定对象类型： <ul style="list-style-type: none"> • P (存储过程) • F (函数) • T (触发器) • E (事件) • S (系统触发器)
object_name	列出对象的名称。
executions	列出每个对象已被调用的次数。
owner_name	列出对象的所有者。
table_name	指定要分析触发器的表。
executions	列出对象已被调用的次数。
Milliseconds	确定该行的执行时间 (以毫秒为单位)。
foreign_owner	确定拥有系统触发器外表的数据库用户。
foreign_table	确定系统触发器外表的名称。

过程分析信息

sa_procedure_profile 报告数据库中执行的过程、函数、事件或触发器中每一行的执行时间信息。

表 9. sa_procedure_profile 统计信息

列名	描述
object_type	确定对象类型： <ul style="list-style-type: none">• P (存储过程)• F (函数)• T (触发器)• E (事件)• S (系统触发器)
object_name	列出对象的名称。
owner_name	列出对象的所有者。
table_name	确定与触发器关联的表 (对于其它对象类型, 该值为空值)。
Line_number	确定过程中的行号。
executions	列出对象已被调用的次数。
Milliseconds	列出对象的执行时间。
percentage	确定特定行所需要的总执行时间的百分比。
foreign_owner	确定拥有系统触发器外表的数据库用户。
foreign_table	确定系统触发器外表的名称。

使用 *Interactive SQL* 设置数据库分析选项

使用 **sa_server_option** 可以设置 *Interactive SQL* 中的数据库分析选项。您的服务器必须正在运行, 而且您必须具有 DBA 权限并连接到数据库。

在 *Interactive SQL* 中, 运行 **sa_server_option**, 然后设置 **procedure_profiling** 选项。

例如:

```
CALL sa_server_option ( 'procedure_profiling', 'ON' )
```

有关其它选项, 请参见 “SQL Anywhere Server - SQL 参考” > “系统过程” > “sa_server_option 系统过程”。

使用 *Interactive SQL* 生成分析信息

sa_procedure_profile 和 **sa_procedure_profile_summary** 可以生成过程、函数、事件和触发器的执行统计信息。

在 *Interactive SQL* 中，运行 **sa_procedure_profile** 或 **sa_procedure_profile_summary**。例如：

```
CALL sa_server_option ( 'procedure_profiling', 'ON' )
```

有关其它选项，请参见“SQL Anywhere Server - SQL 参考”。

监控性能统计信息

使用 *Sybase Central* 中的性能监视器可以显示 Simplex 和 Multiplex 服务器的统计信息。统计信息实时显示在动态图表中。

注意：本节中的主题仅涵盖 Simplex 服务器。有关 Multiplex 服务器，请参见《使用 *Sybase IQ Multiplex*》。

在服务器级监控性能

使用 *Sybase Central* 中的性能监视器可以监控 Simplex 或 Multiplex 服务器的统计信息。

1. 若要启动性能监视器，请在 *Sybase Central* 树视图中单击服务器名。
2. 在右窗格中，单击“性能监视器”选项卡。

有关详细信息和选项，请参见 *Sybase IQ* 帮助中的“服务器”>“监控性能”。

另请参见

- 内存使用统计信息（第 41 页）
- 高速缓存统计信息（第 41 页）
- CPU 使用率统计信息（第 43 页）
- 线程统计信息（第 43 页）
- 连接统计信息（第 44 页）
- 请求统计信息（第 45 页）
- 事务统计信息（第 45 页）
- 存储 I/O 统计信息（第 46 页）
- 数据库空间使用情况统计信息（第 47 页）
- 网络统计信息（第 47 页）

内存使用统计信息

内存使用统计信息显示服务器内存统计信息。

表 10. 内存使用

名称	描述	缺省情况下是否监控?
分配的内存	由 IQ 服务器分配的内存 (以兆字节计)	是
分配的最大内存	由 IQ 服务器分配的最大内存 (以兆字节计)	否

另请参见

- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

高速缓存统计信息

高速缓存统计信息描述了高速缓存的使用情况。

表 11. 高速缓存统计信息

名称	描述	缺省情况下是否监控?
目录高速缓存命中数	每秒目录高速缓存命中数。	否
临时高速缓存命中数	每秒临时高速缓存命中数。	否
主高速缓存命中数	每秒主高速缓存命中数。	否
目录高速缓存读取数	每秒查找目录高速缓存页的次数。	是
临时高速缓存读取数	每秒查找临时高速缓存页的次数。	否

名称	描述	缺省情况下是否监控?
主高速缓存读取数	每秒的主高速缓存页查找次数。	否
目录高速缓存当前大小	当前目录高速缓存大小 (以兆字节计)。	否
临时高速缓存当前大小	当前临时高速缓存大小 (以兆字节计)。	否
主高速缓存当前大小	当前主高速缓存大小 (以兆字节计)。	否
正在使用的目录高速缓存百分比	正在使用的目录高速缓存百分比。	否
正在使用的临时高速缓存百分比	正在使用的临时高速缓存百分比。	否
正在使用的主高速缓存百分比	正在使用的主高速缓存百分比。	否
已固定的目录高速缓存	已固定的目录高速缓存页数。	否
已固定的临时高速缓存	已固定的临时高速缓存页数。	否
已固定的主高速缓存	已固定的主高速缓存页数。	否
已固定的目录高速缓存百分比	已固定的目录高速缓存的百分比。	否
已固定的临时高速缓存百分比	已固定的临时高速缓存的百分比。	否
已固定的主高速缓存百分比	已固定的主高速缓存百分比。	否
目录高速缓存脏页百分比	目录高速缓存脏页的百分比。	否
临时高速缓存脏页百分比	临时高速缓存脏页的百分比。	否
主高速缓存脏页百分比	主高速缓存脏页的百分比。	否

另请参见

- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

CPU 使用率统计信息

CPU 使用率统计信息显示占用的 CPU 资源的百分比。

表 12. CPU 使用率

名称	描述	缺省情况下是否监控?
CPU 使用率	IQ 进程 CPU 使用百分比, 包括系统和用户使用率。	是
CPU 系统使用率	IQ 进程 CPU 系统使用百分比。	否
CPU 用户使用率	IQ 进程 CPU 用户使用百分比。	否

另请参见

- 内存使用统计信息 (第 41 页)
- 高速缓存统计信息 (第 41 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

线程统计信息

线程统计信息描述了线程的使用情况。

表 13. 线程统计信息

名称	描述	缺省情况下是否监控?
正在使用的 IQ 线程	IQ 服务器使用的线程数	否
可用 IQ 线程	IQ 服务器中可用的线程数	否
正在使用的 SA 线程数	SQL Anywhere 引擎使用的线程数。	否

另请参见

- 内存使用统计信息 (第 41 页)

- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

连接统计信息

连接统计信息显示连接活动。

表 14. 连接统计信息

名称	描述	缺省情况下是否监控?
连接总计	连接总数包括用户和 INC 连接。	是
用户连接数	用户连接数。	否
INC 进来的连接数	INC 进来的连接数	否
INC 外发的连接数	INC 外发的连接数	否
每分钟的用户连接数	每分钟的用户连接数	否
每分钟断开的用户连接数	每分钟断开的用户连接数	否

另请参见

- 内存使用统计信息 (第 41 页)
- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

请求统计信息

请求统计信息描述了专用于响应来自客户端应用程序请求的活动。

表 15. 请求统计信息

名称	描述	缺省情况下是否监控?
请求	为使服务器能够处理新请求或继续处理现有的请求每秒进入服务器的次数。	否
未调度的请求	当前排队等待可用服务器线程的请求的数量。	否
IQ 正在等待操作	正在等待资源调控器的 IQ 操作数	否
IQ 活动的操作	活动的 IQ 操作数	否

另请参见

- 内存使用统计信息 (第 41 页)
- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

事务统计信息

事务统计信息显示事务活动。

表 16. 事务统计信息

名称	描述	缺省情况下是否监控?
事务总数	包括用户事务和 INC 事务在内的活动事务的总数。	否
用户事务计数	活动的用户事务数	否
INC 事务计数	活动的 INC 事务数	否

名称	描述	缺省情况下是否监控?
活动 Load Table 语句	活动 LOAD TABLE 语句数	否

另请参见

- 内存使用统计信息 (第 41 页)
- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 请求统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

存储 I/O 统计信息

存储 I/O 统计信息描述了磁盘的读取和写入情况。

表 17. 存储 I/O 统计信息

名称	描述	缺省情况下是否监控?
目录存储磁盘读取数	每秒从目录存储读取的千字节数。	否
临时存储磁盘读取数	每秒从临时存储读取的千字节数。	否
主存储磁盘读取数	每秒从主存储读取的千字节数。	否
目录存储磁盘写入数	每秒写入目录存储的千字节数。	否
临时存储磁盘写入数	每秒写入临时存储的千字节数。	否
主存储磁盘写入数	每秒写入主存储的千字节数。	否

另请参见

- 内存使用统计信息 (第 41 页)
- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)

- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 数据库空间使用情况统计信息 (第 47 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

数据库空间使用情况统计信息

数据库空间使用情况统计信息可以确定数据库空间的可用性。

表 18. DBSpace 使用情况

名称	描述	缺省情况下是否监控?
正在使用的 DBSpace 文件大小	正在使用的 DBSpace 大小。每个 <code>dbspace</code> 均有一项此类统计信息。	否
可用的 DBSpace 大小的百分比	可用于每个 <code>dbspace</code> 文件的可用空间百分比。每个文件的每个 <code>dbspace</code> 均有一项此类统计信息。	否

另请参见

- 内存使用统计信息 (第 41 页)
- 高速缓存统计信息 (第 41 页)
- CPU 使用率统计信息 (第 43 页)
- 线程统计信息 (第 43 页)
- 连接统计信息 (第 44 页)
- 请求统计信息 (第 45 页)
- 事务统计信息 (第 45 页)
- 存储 I/O 统计信息 (第 46 页)
- 网络统计信息 (第 47 页)
- 在服务器级监控性能 (第 40 页)

网络统计信息

网络统计信息显示网络活动。

表 19. 网络统计信息

名称	描述	缺省情况下是否监控?
收到的字节	客户端/服务器通信过程中每秒收到的字节数。	是

名称	描述	缺省情况下是否监控?
收到的未压缩字节	禁用压缩时，在客户端/服务器通信过程中每秒收到的字节数。	否
发送的字节	客户端/服务器通信过程中每秒发送的字节数。	是
发送的未压缩字节	禁用压缩时，在客户端/服务器通信过程中每秒发送的字节数。	否
可用通信缓冲区数	可用网络通信缓冲区的数量。	否
通信缓冲区总数	网络通信缓冲区的总数。	否

另请参见

- 内存使用统计信息（第 41 页）
- 高速缓存统计信息（第 41 页）
- CPU 使用率统计信息（第 43 页）
- 线程统计信息（第 43 页）
- 连接统计信息（第 44 页）
- 请求统计信息（第 45 页）
- 事务统计信息（第 45 页）
- 存储 I/O 统计信息（第 46 页）
- 数据库空间使用情况统计信息（第 47 页）
- 在服务器级监控性能（第 40 页）

监控缓冲区高速缓存

缓冲区高速缓存性能是总体性能中的关键因素。缓冲区高速缓存监控器可以记录缓冲区高速缓存、内存和 I/O 统计信息。

使用缓冲区高速缓存监控器可以调优主缓冲区高速缓存和临时缓冲区高速缓存的内存分配。如果一个高速缓存执行明显比另一个高速缓存更多的 I/O，则重新分配少量内存，例如以迭代方式分配百分之十的高速缓存。在重新分配之后，请重新运行负载，并监控性能变化。

另请参见

- 查看 Sybase IQ 环境（第 35 页）
- 监控性能统计信息（第 40 页）
- 缓冲区高速缓存结构（第 61 页）
- 避免缓冲区管理器抖动（第 62 页）
- 监控 CPU 使用的系统实用程序（第 67 页）

- 缓冲区高速缓存监控清单（第 64 页）

启动缓冲区高速缓存监控器

从 Interactive SQL 运行缓冲区高速缓存监控器。在每次启动监控器时，它都会在 Sybase IQ 中作为单独内核线程运行。

使用以下语法启动监控器：

```
IQ UTILITIES { MAIN | PRIVATE } INTO
dummy_table_name
START MONITOR 'monitor_options [ ... ]'
```

MAIN 启动对主缓冲区高速缓存的监控，监控针对您所连接的数据库的 IQ 存储中的所有表进行。

PRIVATE 启动对临时缓冲区高速缓存的监控，监控针对您所连接的数据库的临时存储区中的所有表进行。

您需要发出单独的命令，以监控每个缓冲区高速缓存。在监控器收集结果时，必须使每个会话保持打开状态；关闭其连接时，监控器的运行会停止。一个连接可以运行最多两个监控器运行，一个用于主缓冲区高速缓存，另一个用于临时缓冲区高速缓存。

dummy_table_name 可以是任何 Sybase IQ 基表或临时表。为了与其它 **IQ UTILITIES** 命令语法兼容，必须提供表名。最好使用仅用于监控的表。

若要控制监控器输出文件的目录位置，请设置 **MONITOR_OUTPUT_DIRECTORY** 选项。如果不设置此选项，则监控器会将输出发送到数据库所在的相同目录。所有监控器输出文件均用于监控器运行期间。在监控器运行已停止之后，这些文件会保留下来。

在创建任何 **Multiplex** 查询服务器之前，要么声明在监控中使用的临时表，要么在创建新数据库时创建永久性伪表。这些解决方案避免了 **DDL** 更改，以便在生产运行期间数据在查询服务器上一直保持工作状态。

注意： 若要简化监控器的使用，请创建存储过程以声明伪表，并指定其输出位置，然后启动监控器。

注意： 间隔（有两个例外）应用于每行输出，而不是每页。例外是 **-cache_by_type** 和 **-debug**，这时，每个显示内容以新页开始。

另请参见

- 输出选项（第 50 页）
- 在监控器运行时检查结果（第 59 页）
- 停止缓冲区高速缓存监控器（第 60 页）
- 检查并保存监控器结果（第 60 页）

输出选项

缓冲区高速缓存监控器输出取决于 *monitor_options* 参数包括的开关。

-summary

显示主缓冲区高速缓存和临时缓冲区高速缓存的摘要信息。如果不指定任何监控器选项，则会收到摘要报告。

用法

```
monitor_options -summary
```

输出

表 20. -summary 输出字段

输出字段	描述
<i>Users</i>	连接到缓冲区高速缓存的用户数
<i>IO</i>	由缓冲区高速缓存执行的组合物理读取数和写入数

所显示的字段是为其它选项描述的字段，外加以下字段：

另请参见

- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-cache

详细显示主缓冲区高速缓存或临时缓冲区高速缓存的活动。关键字段是 *Finds*、*HR%* 和 *BWaits*。

用法

```
monitor_options -cache
```

输出

表 21. -cache 输出字段

输出字段	描述
<i>Finds</i>	对缓冲区高速缓存的查找请求数。如果 <i>Finds</i> 值突然降到零并保持不变，则服务器被死锁。当服务器有任何活动时， <i>Finds</i> 值应当是非零值。
<i>Creates</i>	在数据库中创建页的请求数
<i>Dests</i>	在数据库中销毁页的请求数
<i>Dirty</i>	缓冲区脏化（修改）的次数
<i>HR%</i>	命中率，在不请求任何 I/O 的情况下，缓冲区高速缓存满足查询的百分比。命中率越高越好，如果高速缓存设置得足够大，通常是 90% - 100%。对于大型查询，命中率可能一开始很低，但预取开始工作时则会增加。
<i>BWaits</i>	强制等待繁忙页（页帧争用）的查找请求数。通常它很低，但在某些特殊情况下它可能很高。例如，如果同时启动相同查询，则都需要相同页，因此第二个请求必须等待第一个请求从磁盘获取页。
<i>ReReads</i>	在相同事务中存储区的相同部分需要重读到高速缓存中的大约次数。应当始终很低，但对 Sybase IQ 12.4.2 和以上版本来说，高数字也没关系。
<i>FMiss</i>	假丢失数，缓冲区高速缓存需要多次查找才能在内存中找到页的次数。此数字应当是 0 或非常小。如果该值很高，则可能发生了回退，并且需要重复某些操作。
<i>Cloned</i>	Sybase IQ 为了给写入程序建立新版本而需要的缓冲区数，而它必须为并发读取程序保留早期版本。仅当其它用户正在查看某页时，才复制该页。
<i>Reads/Writes</i>	由缓冲区高速缓存执行的物理读取和写入
<i>PF/PFRead</i>	预取请求数和为预取完成的读取数。
<i>GDirty</i>	LRU 缓冲区被脏抢夺并且 Sybase IQ 在使用它之前必须将其写出的次数。此值不应当长时间大于 0。如果是，则可能需要增加清理程序线程数，或移动清洗标记。
<i>Pin%</i>	缓冲区高速缓存中正在使用和锁定的页的百分比。

输出字段	描述
<i>Dirty%</i>	被修改的缓冲块的百分比。不要尝试让此值超过 85-90%；否则， <i>GDirty</i> 将变为大于 0。

另请参见

- [-summary \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-cache_by_type

按照 IQ 页类型细分 **-cache** 结果。（例外是 **Bwaits** 列，此列仅显示总计。）当需要向 Sybase 技术支持部门提供信息时，此格式最有用。

用法

```
monitor_options -cache_by_type
```

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-file_suffix

创建名为 <dbname>.<connid>-<main_or_temp>-<suffix> 的监控器输出文件。如果您没有指定可选文件扩展名，则文件扩展名缺省为 iqmon。

用法

```
monitor_options -file_suffix {extension}
```

另请参见

- [-summary](#) (第 50 页)
- [-cache](#) (第 50 页)
- [-cache_by_type](#) (第 52 页)
- [-io](#) (第 53 页)
- [-bufalloc](#) (第 54 页)
- [-contention](#) (第 55 页)
- [-threads](#) (第 57 页)
- [-interval](#) (第 57 页)
- [-append | - truncate](#) (第 58 页)
- [-debug](#) (第 59 页)

-io

显示在指定时段内主缓冲区高速缓存或临时 (专用) 缓冲区高速缓存 I/O 率和压缩率。这些计数器表示服务器的所有活动；这些信息不会被设备断开。

用法

```
monitor_options -io
```

输出

表 22. -io 输出字段

输出字段	描述
Reads	由缓冲区高速缓存执行的物理读取数
Lrd(KB)	读入的逻辑 KB 数 (页大小乘以请求数)
Prd(KB)	读入的物理 KB 数
Rratio	逻辑对物理读入数据的压缩率，用于衡量磁盘读取的压缩效率
Writes	缓冲区高速缓存执行的物理写入数

输出字段	描述
Lwrt(KB)	逻辑写入 KB 数
Pwrt(KB)	物理写入 KB 数
Wratio	逻辑与物理写入数据的压缩率

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-bufalloc

显示有关主缓冲区分配器或临时缓冲区分配器的信息，该分配器可在缓冲区高速缓存中保留空间以供排序、散列和位图等对象使用。

用法

```
monitor_options -bufalloc
```

输出

表 23. -bufalloc 输出字段

输出字段	描述
<i>OU</i>	User_Resource_Reservation 选项设置 (以前是 Optimize_For_This_Many_Users)
<i>AU</i>	当前活动用户数
<i>MaxBuf</i>	缓冲区分配器控制下的缓冲区数
<i>Avail</i>	pin 配额分配的当前可用缓冲区数
<i>AvPF</i>	预取配额分配的当前可用缓冲区数
<i>Slots</i>	使用缓冲区高速缓存配额的当前注册对象数

输出字段	描述
<i>PinUser</i>	使用 pin 配额的对象数（例如散列、排序和 B 树对象）
<i>PFUsr</i>	使用预取配额的对象数
<i>Posted</i>	预先计划配额用户的对象数
<i>UnPost</i>	作为特别配额用户的对象数
<i>Locks</i>	在缓冲区分配器上执行的互斥锁定数
<i>Waits</i>	线程必须等待锁定的次数

另请参见

- `-summary` (第 50 页)
- `-cache` (第 50 页)
- `-cache_by_type` (第 52 页)
- `-file_suffix` (第 53 页)
- `-io` (第 53 页)
- `-contention` (第 55 页)
- `-threads` (第 57 页)
- `-interval` (第 57 页)
- `-append | -truncate` (第 58 页)
- `-debug` (第 59 页)

-contention

显示很多关键的缓冲区高速缓存和内存管理器锁定。这些锁定和互斥计数器显示在缓冲区高速缓存和堆内存中的活动，以及解除这些锁定的速度。如果超时数超过 20%，则表示有问题。

用法

```
monitor_options -contention
```

输出

表 24. -contention 输出字段

输出字段	描述
<i>AU</i>	当前活动用户数
<i>LRULks</i>	LRU 的锁定次数（对临时高速缓存重复）
<i>woTO</i>	授予锁定而没有超时的次数（对临时高速缓存重复）

输出字段	描述
<i>Loops</i>	在授予锁定之前 Sybase IQ 重试的次数 (对临时高速缓存重复)
<i>TOs</i>	Sybase IQ 超时并且必须等待锁定的次数 (对临时高速缓存重复)
<i>BWaits</i>	高速缓存中的缓冲区“忙等待”数 (对临时高速缓存重复)
<i>IOLock</i>	Sybase IQ 锁定压缩 I/O 池的次数 (对临时高速缓存重复) ; 可以被忽略
<i>IOWait</i>	Sybase IQ 必须等待对压缩 I/O 池的锁定的次数 (对临时高速缓存重复) ; 可以被忽略
<i>HTLock</i>	Sybase IQ 锁定块映射散列表的次数 (对临时高速缓存重复)
<i>HTWait</i>	Sybase IQ 必须等待块映射散列表的次数 (对临时高速缓存重复) ; HTLock 和 HTWait 指示您正在使用多少块映射
<i>FLock</i>	Sybase IQ 必须锁定空闲列表的次数 (对临时高速缓存重复)
<i>FLWait</i>	Sybase IQ 必须等待对空闲列表的锁定的次数 (对临时高速缓存重复)
<i>MemLks</i>	Sybase IQ 取得内存管理器 (堆) 锁定的次数
<i>MemWts</i>	Sybase IQ 必须等待内存管理器锁定的次数

注意: 由于操作系统的改进, Sybase IQ 不再使用旋转锁定。因此, 很少使用 woTO、Loops 和 TOs 统计信息。

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-threads

显示处理线程管理器计数。值是全服务器范围的（即无论对主或专用选择此选项都没关系）。它们表示在报告的最后一页后出现的新事件。

用法

```
monitor_options -threads
```

输出

表 25. -thread 输出字段

输出字段	描述
<i>cpus</i>	Sybase IQ 正在使用的 CPU 数；此数可能小于系统上的数字
<i>Limit</i>	Sybase IQ 可以使用的最大线程数
<i>NTeams</i>	当前在使用中的线程组数
<i>MaxTms</i>	曾经在使用中的最大组数
<i>NThrds</i>	当前的现有线程数
<i>Resrvd</i>	为系统（连接）使用而保留的线程数
<i>Free</i>	可供分配的线程数。请监控此值，如果它非常低，则指示线程匮乏
<i>Locks</i>	对线程管理器使用的锁定数
<i>Waits</i>	Sybase IQ 必须等待对线程管理器的锁定的次数

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | -truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-interval

指定报告间隔（以秒为单位）。缺省值为每 60 秒。最小值为每 2 秒。通过在查询期间或有性能问题的时间按缺省间隔运行监控器，通常可以获取有用的结果。较短

的间隔可能不会给出有意义的结果。间隔应当与作业时间成比例；通常，一分钟就足够了。

用法

```
monitor_options -interval
```

输出

第一个显示内容显示自服务器启动以来的计数器。后续显示则显示与前一显示不同的地方。

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)
- [-debug \(第 59 页\)](#)

-append | - truncate

将输出追加到现有输出文件或截断现有输出文件。Truncate 是缺省值。

用法

```
monitor_options -append | -truncate
```

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-debug \(第 59 页\)](#)

-debug

显示性能监视器可用的所有信息，无论是否有覆盖相同信息的标准显示模式。**-debug** 主要用于向 Sybase 技术支持部门提供信息。

用法

```
monitor_options -debug
```

输出

页首是按磁盘块类型分类的统计信息阵列。其后是其它缓冲区高速缓存统计信息、内存管理器统计信息、线程管理器统计信息、空闲列表统计信息、CPU 利用率和最终缓冲区分配器统计信息。

然后，缓冲区分配器统计信息按客户端类型（散列、排序等）分类，并显示最近分配的缓冲区柱状图。内存分配指示有多少是在报告的最后一页后分配的。

另请参见

- [-summary \(第 50 页\)](#)
- [-cache \(第 50 页\)](#)
- [-cache_by_type \(第 52 页\)](#)
- [-file_suffix \(第 53 页\)](#)
- [-io \(第 53 页\)](#)
- [-bufalloc \(第 54 页\)](#)
- [-contention \(第 55 页\)](#)
- [-threads \(第 57 页\)](#)
- [-interval \(第 57 页\)](#)
- [-append | - truncate \(第 58 页\)](#)

在监控器运行时检查结果

在 UNIX 系统上，可以在查询正在运行时监视监控器输出。

例如，可以使用以下命令启动监控器：

```
iq utilities main into monitor_tab start monitor "-cache -interval
2 -file_suffix iqmon"
```

此命令将输出发送到名为 `dbname.conn#[main|temp]-iqmon` 的 ASCII 文件。因此，对于数据库 `iqdemo`，结果会发送到 `iqdemo.2-main-iqmon`。

若要监视结果，请在系统提示下发出以下命令：

```
$ tail -f iqdemo.2-main-iqmon
```

另请参见

- [启动缓冲区高速缓存监控器 \(第 49 页\)](#)
- [输出选项 \(第 50 页\)](#)

- 停止缓冲区高速缓存监控器 (第 60 页)
- 检查并保存监控器结果 (第 60 页)

停止缓冲区高速缓存监控器

用于停止监控器运行的命令与用于启动它的命令相似，只是不需要指定任何选项。

使用以下语法可以停止 Sybase IQ 缓冲区高速缓存监控器：

```
IQ UTILITIES { MAIN | PRIVATE } INTO
dummy_table_name
STOP MONITOR
```

注意：为了让某些选项设置生效，必须重新启动数据库。如果监控器正在运行，则需要关闭它，以便数据库可以重新启动。

另请参见

- 启动缓冲区高速缓存监控器 (第 49 页)
- 输出选项 (第 50 页)
- 在监控器运行时检查结果 (第 59 页)
- 检查并保存监控器结果 (第 60 页)

检查并保存监控器结果

缓冲区高速缓存监控器记录每次运行的结果。

日志的缺省名：

- `dbname.connection#-main-iqmon` 用于主缓冲区高速缓存结果
- `dbname.connection#-temp-iqmon` 用于临时缓冲区高速缓存结果

前缀 `dbname.connection#` 表示数据库名称和连接号。如果看见多个连接号，并且不能确定哪个是您的，则可以运行目录存储过程 `sa_conn_info`。此过程将显示与数据库的每个活动连接的连接号、用户 ID 和其它信息。

可以在 **IQ UTILITIES** 命令中使用 **-file_suffix** 参数，将后缀 `iqmon` 更改为您选择的后缀。

若要查看监控器运行的结果，请使用您通常用于显示或打印文件的文本编辑器或任何其它方法。

当再次从相同数据库和连接号运行监控器时，缺省情况下它将覆盖以前的结果。如果需要保存监控器运行的结果，请在从相同数据库再次启动监控器之前将文件复制到另一个位置，或使用 **-append** 选项。

另请参见

- 启动缓冲区高速缓存监控器 (第 49 页)

- 输出选项 (第 50 页)
- 在监控器运行时检查结果 (第 59 页)
- 停止缓冲区高速缓存监控器 (第 60 页)

缓冲区高速缓存结构

更改 `CACHE_PARTITIONS` 值可以提高多 CPU 配置中的装载或查询性能。

Sybase IQ 按照系统的 CPU 数自动计算缓冲区高速缓存的高速缓存分区数。如果多 CPU 配置中的装载或查询性能比期望值慢, 可能能够通过更改 `CACHE_PARTITIONS` 数据库选项的值来提高它。有关详细信息, 请参见《参考: 语句和选项》中的 `CACHE_PARTITIONS` 选项。

当缓冲区接近高速缓存的最近使用最少 (LRU) 末尾时, 它们将越过清洗标记。Sybase IQ 会将最旧页 (那些越过清洗标记的页) 写出到磁盘中, 以便可以重用它们占据的高速缓存空间。一组 Sybase IQ 处理线程 (名为清理程序线程) 将清除 (写入) 最旧的缓冲区。

当 Sybase IQ 需要将一页数据读取到高速缓存中时, 它会抢夺 LRU 缓冲区。如果缓冲区仍然是“脏的” (已修改), 则它必须首先写入到磁盘中。监控器 `-cache` 报告中的 `Gdirty` 列会显示 LRU 缓冲区被脏抢夺并且 Sybase IQ 在使用它之前必须先将它写出到磁盘的次数。

通常, Sybase IQ 能够使 `Gdirty` 值保持为 0。如果此值在较长时段内大于 0, 则可能需要调整某个用于控制清理程序线程数和清洗标记的数据库选项。请参见《参考: 语句和选项》中的“`SWEEPER_THREADS_PERCENT` 选项”或“`WASH_AREA_BUFFERS_PERCENT` 选项”。

另请参见

- 查看 Sybase IQ 环境 (第 35 页)
- 监控性能统计信息 (第 40 页)
- 监控缓冲区高速缓存 (第 48 页)
- 避免缓冲区管理器抖动 (第 62 页)
- 监控 CPU 使用的系统实用程序 (第 67 页)
- 缓冲区高速缓存监控清单 (第 64 页)

避免缓冲区管理器抖动

系统必须先写入脏页然后才能读取请求页的情况下会发生抖动，这会显著降低系统速度。为获得最佳性能，请始终分配足够可用的内存，从而使页写入程序满足可用空间需求。

缓冲区高速缓存抖动

缓冲区高速缓存抖动与系统抖动类似，且在没有足够可用于读取的清洁缓冲区的情况下发生。这导致在高速缓存中出现相同类型的“先写入然后读取”延迟，且可能会在没有足够大的缓冲区高速缓存来容纳查询中引用的所有对象的情况下发生。

若要消除缓冲区高速缓存抖动，必须为缓冲区高速缓存分配更多内存。请不要过多分配缓冲区高速缓存。分配过多内存可能会导致为数据库缓冲区高速缓存分配内存时发生系统抖动。在极端情况下，没有解决缓冲区高速缓存抖动问题时分配过多内存可能会引起多级别的抖动。

在多用户环境中，或者查询复杂性导致的曲解或不确定性使优化程序在某情况下（在该情况中需要用明显多于查询可用的高速缓存所容纳的值数目生成 HASH 对象）选择 HASH 算法时可能会出现其它不易发现形式的缓冲区高速缓存抖动。

设置缓冲区大小

设置缓冲区大小时，请记住以下事项：

- 如果 Sybase IQ 缓冲区高速缓存太大，则在 Sybase IQ 尝试使用所有这部分内存时，操作系统将强制分页。
- 如果 Sybase IQ 缓冲区高速缓存太小，则 Sybase IQ 会发生抖动，因为它不能将足够的查询数据填到高速缓存中。

如果正在遇到剧烈的性能问题，则应当监视分页，以确定是否是抖动问题。如果是，请重置缓冲区大小。

查询和散列算法

如果您监控分页并确定存在抖动问题，您还可以限制在执行包含涉及散列算法的查询的语句过程中的抖动量。通过调整 HASH_THRASHING_PERCENT 数据库选项，可以控制在语句回退且返回错误之前允许的硬盘 I/O 百分比。

HASH_THRASHING_PERCENT 的缺省值为 10%。增大 HASH_THRASHING_PERCENT 将允许在回退之前对磁盘进行更多分页，而减小 HASH_THRASHING_PERCENT 将减少在回退之前允许的分页。

如果查询涉及在 Sybase IQ 的早期版本中执行的散列算法，则这些查询现在可能在达到缺省的 HASH_THRASHING_PERCENT 限制时发生回退。Sybase IQ 报告错误 Hash insert thrashing detected 或 Hash find thrashing detected。执行以下一项或多项操作可为查询提供执行所需的资源：

- 通过增大 `HASH_THRASHING_PERCENT` 的值来放宽分页限制。
- 增加临时高速缓存的大小（仅限 DBA）。请切记，增加临时高速缓存的大小要求在主高速缓存分配中减少相等大小以防止系统抖动的可能性。
- 对于此语句，尝试确定 `Sybase IQ` 错误估计一个或多个散列大小的原因并缓解该错误估计所产生的后果。例如，请检查需要 `LF` 或 `HG` 索引的所有列都有一个。还要考虑多列索引是否合适。
- 减少数据库选项 `HASH_PINNABLE_CACHE_PERCENT` 的值。

有关这些数据库选项的详细信息，请参见《参考：语句和选项》中的“`HASH_THRASHING_PERCENT` 选项”和“`HASH_PINNABLE_CACHE_PERCENT` 选项”。

若要发现查询中可能存在的问题，请用临时数据库选项 `QUERY_PLAN = 'ON'` 和 `QUERY_DETAIL = 'ON'`，运行查询，以生成查询计划，然后检查查询计划中的估计值。生成的查询计划存在于消息日志文件中。

另请参见

- 管理缓冲区高速缓存（第 3 页）

监控 Windows 系统上的分页

使用 Windows 性能工具可以监控分页和对象内存。

若要访问系统监控器，请选择对象“Logical Disk”（包含文件 `PAGEFILE.SYS` 的磁盘实例）和计数器 `Disk Transfers/Sec`。将 Windows 页面文件放在不同于数据库的数据库空间设备的磁盘上。还可以监控对象“Memory”和计数器 `Pages/Sec`。但是，该值是包括软故障和硬故障的所有内存故障的总和。

另请参见

- 监控 UNIX 系统上的分页（第 63 页）

监控 UNIX 系统上的分页

使用 `vmstat` 可以监控诸如分页之类的系统活动。

`vmstat` 的简短命令语法是：

```
vmstat interval
               count
```

`interval` 是输出行之间的时间，`count` 是输出行的显示次数。有关 `vmstat`（包括其选项和字段说明）的详细信息，请参见操作系统的文档。

另请参见

- 监控 Windows 系统上的分页（第 63 页）

缓冲区高速缓存监控清单

请查看此清单以调整超出正常范围的高速缓存行为。

表 26. 缓冲区高速缓存监控清单

统计信息	正常行为	需要调整的行为	建议操作
HR% (高速缓存命中率)	<p>高于 90%。</p> <p>对于单个内部数据结构，比如 garray、barray、位图 (bm)、散列对象、排序对象、变长 btree (btreev)、定长 btree (btreef)、位向量 (bv)、dbext、dbid、vdo、存储、检查点块 (ckpt)，在查询运行时命中率应当是超过 90%。它可能首先低于 90%。一旦预取开始工作 (PF 或 PrefetchReqs > 0)，则命中率应当逐渐增长到超过 90%。</p>	<p>在预取工作之后，命中率低于 90%。</p> <p>注意：某些对象不执行预取，因此其命中率在正常情况下可能很低。</p>	<p>尝试通过调整 -iqmc 和 -iqtc 重新平衡主高速缓存大小和临时高速缓存大小。</p> <p>还应通过调整 PREFETCH_THREADS_PERCENT 选项，尝试增加预取线程数。</p>
Gdirty (脏抢夺)	在有中等高速缓存大小 (< 10GB) 的系统中为 0。	<p>GDirty > 0</p> <p>注意：仅当脏页的数目达到清洗区的某个百分比时，系统才会激活清理程序线程。如果 GDirty/GrabbedDirty 大于 0 并且 I/O 率 (写入) 很低，则系统可能直接被轻装载，并且不需要执行操作。</p>	<p>调整 SWEeper_THREADS_PERCENT 选项 (缺省值 10%) 或 WASH_AREA_BUFFERS_PERCENT 选项 (缺省值 20%)，以增加清洗区的大小。</p>
BWaits (缓冲区忙等待)	0	持久 > 0，表示多个作业正在争夺同一个缓冲区。	<p>如果 I/O 率 (写入) 很高，则忙等待可能是高速缓存抖动导致的。请检查高速缓存报告中的命中率，以确定是否需要重新平衡主高速缓存与临时高速缓存。</p> <p>如果批量作业正在同时启动很多几乎相同的查询，请尝试错开启动时间。</p>

统计信息	正常行为	需要调整的行为	建议操作
LRU 等待 (debug 报告中的 LRUNum TimeOuts 百 分比)	20% 或更少	> 20%， 指示发生严重争用问题。	检查操作系统修补程序级别和其它环境设置。此问题常常是操作系统问题。
IOWait (IONumWait s)	10% 或更低	> 10%	检查是否有磁盘错误或 I/O 重试
FLWait (FLMutexWa its)	20% 或更低	> 20%	检查 dbspace 配置： 数据库空间即将不足? DISK_STRIPE 是否为 ON? sp_iqcheckdb 是否报告大于 15% 的碎片?
HTWait (BmapHTNu mWaits) MemWts (MemNtimes Waited) (PFMgrCond VarWaits)	10% 或更低	> 10%	与 Sybase 技术支持部门联系。

统计信息	正常行为	需要调整的行为	建议操作
CPU 时间 (调试报告中的 CPU Sys Seconds, CPU Total Seconds)	CPU 系统时间 (秒) < 20%	CPU 系统时间 (秒) > 20% 如果 CPU Total Seconds 还报告低利用率，并有让系统忙的足够作业，则高速缓存可能正在抖动，或可能丢失平行性。	调整 -iqgovern 以减少允许的并发查询总数。 检查高速缓存报告中的命中率和 I/O 率，确定是否有高速缓存抖动。还应查看 cache_by_type (或 debug) 报告中散列对象的命中率，以确定散列对象是否正在抖动：在 I/O 率 (写入) 很高时，它是否 <90% ? 检查查询计划中所尝试的平行性。是否有足够线程可用？ 系统是否有非常庞大的 CPU 数？可能需要诸如 Multiplex 配置这样的策略。
InUse% (使用中的缓冲区)	除了启动期间以外等于或接近 100%	小于大约 100%	缓冲区高速缓存可能太大。 尝试通过调整 -iqmc 和 -iqtc 重新平衡主高速缓存大小和临时高速缓存大小。
Pin% (已占用的缓冲区)	< 90%	> 90 到 95%，指示系统正在危险地接近缓冲区不足条件，而这会导致事务回退	尝试重新平衡主高速缓存与临时高速缓存的大小。 如果不可能重新平衡缓冲区高速缓存大小，请尝试减少 -iqgovern ，以限制并发运行的作业数。

统计信息	正常行为	需要调整的行为	建议操作
可用线程 (ThrNumFree)	可用线程数 > 保留线程数	如果可用线程数降低至保留计数，则系统可能发生线程空闲。	<p>尝试下列操作之一：</p> <p>通过设置 -iqmt，增加线程数。</p> <p>减少与线程相关的选项：</p> <p>MAX_IQ_THREADS_PER_CONNECTION, MAX_IQ_THREADS_PER_TEAM.</p> <p>通过设置 USER_RESOURCE_RESERVATION，限制查询引擎资源分配。</p> <p>通过设置 -iqgovern，限制作业数。</p>
FlOutOfSpace (仅调试)	0，指示此存储区的空闲列表不完整；未分配的页可用	1，指示已完全分配此存储（主存储或临时存储）	将更多 dbspace 添加到该存储

另请参见

- 查看 Sybase IQ 环境（第 35 页）
- 监控性能统计信息（第 40 页）
- 监控缓冲区高速缓存（第 48 页）
- 缓冲区高速缓存结构（第 61 页）
- 避免缓冲区管理器抖动（第 62 页）
- 监控 CPU 使用的系统实用程序（第 67 页）

监控 CPU 使用的系统实用程序

有一些特定于操作系统的实用程序可用于监控 CPU 使用率

OS	实用程序	描述
UNIX	top (Sun、Linux、HP-UX) 、 topas (IBM-AIX)	实时地持续查看处理器活动。
	ps	报告进程状态。

OS	实用程序	描述
	vmstat	显示有关系统进程、内存、分页、块 IQ、捕获和 CPU 活动的信息。
	iostat -x	显示磁盘子系统信息。
Windows	系统监控器任务管理器	提供有关计算机性能和正在运行的应用程序、进程、CPU 使用率及其它系统服务的详细信息。

另请参见

- 查看 Sybase IQ 环境 (第 35 页)
- 监控性能统计信息 (第 40 页)
- 监控缓冲区高速缓存 (第 48 页)
- 缓冲区高速缓存结构 (第 61 页)
- 避免缓冲区管理器抖动 (第 62 页)
- 缓冲区高速缓存监控清单 (第 64 页)

优化查询和删除

帮助您计划、结构化和控制查询的建议。

实现结构化查询的提示

改进查询结构可以加快查询运行速度。

- 有些情况下，包含子查询的命令语句也可表示为连接，从而有可能加快运行速度。
- 如果对 **GROUP BY** 子句中的多列进行分组，请按唯一值的数目降序列出这些列（如果可以）。这将提供最佳的查询性能。
- 与即席连接相比，连接索引通常能使连接查询的执行速度更快，不过其代价是要使用更大的磁盘空间且显著增加装载时间。但是，当连接查询不引用多表连接索引中的最大表时，或较小和较大表之间的行计数差异很大时，即席连接的性能通常优于连接索引。
- 您可以通过使用附加列提高性能，以便存储频繁计算的结果。

注意：涉及包含大量空值的列的查询比先前版本中的运行速度快。但是，在将大量空值插入到表的情况下，在该表中插入或更新数据这一过程可能会花费较长时间（同先前版本相比）。

GROUP BY 与 UNION ALL 配合使用时对查询性能的影响

在某些情况下，使用拆分 **GROUP BY** 方法可以减少查询处理时间。

为了提高装载性能，非常大的表有时被分割为多个小表，并通过在视图中使用 **UNION ALL** 进行访问。对于一些使用此类视图和 **GROUP BY** 的非常特殊的查询，Sybase IQ 优化程序能够通过将一些 **GROUP BY** 操作复制到此类 **UNION ALL** 的各个分支，以并行执行操作，然后合并结果，从而增强性能。此方法称为拆分 **GROUP BY**，可以减少由顶层 **GROUP BY** 处理的数据量，从而减少查询处理时间。

只有将 **GROUP BY** 与 **UNION ALL** 配合使用的某些查询才能提高性能。例如，下面的简单查询可以从拆分 **GROUP BY** 中受益：

```
CREATE VIEW vtable (v1 int, v2 char(4)) AS SELECT a1, a2 FROM tableA
UNION ALL SELECT b1, b2 FROM tableB;  SELECT COUNT(*), SUM(v1) FROM
vtable GROUP BY v2;
```

让我们来分析此查询，优化程序首先对 **tableA** 执行 **COUNT(*) GROUP BY**，对 **tableB** 执行 **COUNT(*) GROUP BY**，然后将结果传递至顶层 **GROUP BY**。顶层 **GROUP BY** 对这两个 **COUNT(*)** 结果执行 **SUM**，以产生最终查询结果。请注意，顶层 **GROUP BY** 的角色会更改：顶层 **GROUP BY** 使用的集合为 **SUM**，而不是 **COUNT**。

另请参见

- 增强 **ORDER BY** 查询性能（第 71 页）

- 提高了子查询性能 (第 71 页)
- 使用高速缓存方法 (第 72 页)

拆分 GROUP BY 的示例

请参见这些示例，以了解拆分 **GROUP BY** 如何提高性能。

在此示例中，名为 tableA 的大表被分割为四个小表：tabA1、tabA2、tabA3 和 tabA4。系统使用四个小表和 **UNION ALL** 创建视图 unionTab：

```
CREATE VIEW unionTab (v1 int, v2 int, v3 int, v4 int) AS SELECT a, b, c, d FROM tabA1 UNION ALL SELECT a, b, c, d FROM tabA2 UNION ALL SELECT a, b, c, d FROM tabA3 UNION ALL SELECT a, b, c, d FROM tabA4;
```

Sybase IQ 优化程序将 **GROUP BY** 操作拆分到下列查询中，提高了查询性能：

```
SELECT v1, v2, SUM(v3), COUNT(*) FROM unionTab GROUP BY v1, v2;  
SELECT v3, SUM(v1*v2) FROM unionTab GROUP BY v3;
```

另请参见

- 针对拆分 **GROUP BY** 的限制 (第 70 页)

针对拆分 GROUP BY 的限制

从拆分 **GROUP BY** 中受益的情况和查询存在一些限制。

- 如果查询使用 **UNION ALL** 而不是 **UNION**，则可以从拆分 **GROUP BY** 中受益。下面的查询结合使用了 **GROUP BY** 和 **UNION**，因此它不能利用拆分 **GROUP BY**：

```
CREATE VIEW viewA (va1 int, va2 int, va3 int, va4 int) AS SELECT b1, b2, b3, b4 FROM tableB UNION SELECT c1, c2, c3, c4 FROM tableC; SELECT SUM(va1) FROM viewA GROUP BY va3;
```

- 如果查询中的集合不包含 **DISTINCT**，则查询可以从拆分 **GROUP BY** 中受益。下面的查询使用了 **SUM DISTINCT**，因此它不能利用拆分 **GROUP BY**：

```
CREATE VIEW viewA (va1 int, va2 int, va3 int, va4 int) AS SELECT b1, b2, b3, b4 FROM tableB UNION ALL SELECT c1, c2, c3, c4 FROM tableC; SELECT SUM(DISTINCT va1) FROM viewA GROUP BY va3;
```

- 要使查询能够从拆分 **GROUP BY** 中受益，您的临时共享缓冲区高速缓存需要有足够的内存，以便存储用于处理附加 **GROUP BY** 运算符的集合信息和数据。

```
CREATE VIEW viewA (va1 int, va2 int, va3 int, va4 int) AS SELECT b1, b2, b3, b4 FROM tableB UNION ALL SELECT c1, c2, c3, c4 FROM tableC UNION ALL SELECT d1, d2, d3, d4 FROM tableD UNION ALL SELECT e1, e2, e3, e4 FROM tableE UNION ALL SELECT f1, f2, f3, f4 FROM tableF UNION ALL SELECT g1, g2, g3, g4 FROM tableG; SELECT SUM(va1) FROM viewA GROUP BY va3;
```

在此示例中，Sybase IQ 优化程序拆分 **GROUP BY** 并将六个 **GROUP BY** 运算符插入到查询计划中。因此，查询需要更大的临时高速缓存来存储集合信息和数据。如果系统不能分配足够的高速缓存，则优化程序将不会拆分 **GROUP BY**。如果有可用内存，您可以使用 **TEMP_CACHE_MEMORY_MB** 数据库选项来增加临时高速缓存大小。

- 要使查询能够从拆分 GROUP BY 中受益，AGGREGATION_PREFERENCE 数据库选项应设置为其缺省值 0。使用此值，Sybase IQ 优化程序可确定要应用到 GROUP BY 的最佳算法。如果 AGGREGATION_PREFERENCE 的值迫使 Sybase IQ 优化程序选择排序算法处理 GROUP BY，则查询不能从拆分 GROUP BY 中受益。可使用 AGGREGATION_PREFERENCE 选项覆盖优化程序为处理 GROUP BY 所做的算法选择，此时不应将其设置为 1 或 2。

另请参见

- 拆分 GROUP BY 的示例（第 70 页）
- 确定缓冲区高速缓存的大小（第 4 页）

增强 ORDER BY 查询性能

使用多列 HG 索引可以增强 ORDER BY 查询的性能。

通过对单表查询中多列的引用，使用多列 HG 索引增强 ORDER BY 查询的性能。这一更改对用户是透明的，但是可提高查询性能。

使用多列 HG 索引，ORDER BY 子句中具有多列的查询可以加快运行速度。例如，如果在表 T 上用户具有多列索引 HG(x,y,z)，则此索引用于有序投影：

```
SELECT abs (x) FROM T
ORDER BY x, y
```

在以上示例中，HG 索引按照排序顺序垂直投射 x 和 y。

如果 ROWID() 函数位于 SELECT 列表表达式中，则系统还会使用多列 HG 索引。例如：

```
SELECT rowid() +x, z FROM T
ORDER BY x,y,z
```

如果 ROWID() 位于 ORDER BY 列表的末尾，而且该列表的列（ROWID() 除外）位于该索引内，且排序键与前导 HG 列依序匹配，则系统会在查询中使用多列索引。例如：

```
SELECT z,y FROM T ORDER BY x,y,z,ROWID()
```

另请参见

- 提高了子查询性能（第 71 页）
- 使用高速缓存方法（第 72 页）
- GROUP BY 与 UNION ALL 配合使用时对查询性能的影响（第 69 页）

提高了子查询性能

使用 SUBQUERY_FLATTENING_PREFERENCE 和 SUBQUERY_FLATTENING_PERCENT 控制子查询展平。

子查询展平是一种优化技术，使用此技术，优化程序会将包含子查询的查询重新写入使用连接的查询中。Sybase IQ 会展平多个子查询但不是所有子查询。当优化程序选

优化查询和删除

择使用此优化方式时，请使用 `SUBQUERY_FLATTENING_PREFERENCE` 和 `SUBQUERY_FLATTENING_PERCENT` 来进行控制。

在 Sybase IQ 15.0 中，不建议使用 `FLATTEN_SUBQUERIES` 选项。

另请参见

- `GROUP BY` 与 `UNION ALL` 配合使用时对查询性能的影响（第 69 页）
- 使用高速缓存方法（第 72 页）
- 增强 `ORDER BY` 查询性能（第 71 页）

使用高速缓存方法

设置 `SUBQUERY_CACHING_PREFERENCE` 选项，为相关子查询选择高速缓存方法。

相关子查询包含对子查询范围外的一个或多个表的引用，且相关子查询在每次被引用列中的值更改时都会重新执行。使用 `SUBQUERY_CACHING_PREFERENCE` 选项，为执行相关子查询选择高速缓存方法。

另请参见

- 提高了子查询性能（第 71 页）
- 增强 `ORDER BY` 查询性能（第 71 页）
- `GROUP BY` 与 `UNION ALL` 配合使用时对查询性能的影响（第 69 页）

计划查询

生成查询计划可帮助您运行更高效的查询。

如果您已创建正确的索引，Sybase IQ 查询优化程序通常能够以最有效的方式执行查询，有时即使您并未使用最有效的语法也可以做到这一点。但是适当的查询设计依然很重要。在您仔细计划查询时，您所做的计划可能会对速度和结果的正确性产生重大影响。

在执行任何查询之前，Sybase IQ 查询优化程序会创建查询计划。Sybase IQ 使您能够通过下面几节介绍的选项来检查和影响查询计划，以帮助您评估查询。有关如何指定这些选项的详细信息，请参见《参考：语句和选项》。

注意：对于所有接受整数值的数据库选项，Sybase IQ 会将任何小数形式的 *option-value* 设置截断为整数值。例如，值 3.8 将被截断为 3。

查询评估选项

设置相应选项可帮助您评估查询计划。

- **INDEX_ADVISOR** - 当设置为 **ON** 时，索引顾问会将索引建议作为 Sybase IQ 查询计划中的一部分或作为 Sybase IQ 消息日志文件中单独的消息（如果未启用查询计划）进行打印。这些消息以字符串“Index Advisor:”开头，并且您可以使用该字

字符串在 Sybase IQ 消息文件中对其进行搜索和过滤。此选项以 OWNER . TABLE . COLUMN 格式输出消息，缺省情况下为 **OFF**。

另请参见《参考：构件块、表和过程》的“系统过程”中的“sp_iqindexadvice 过程”。

- **INDEX_ADVISOR_MAX_ROWS** - 用于限制由索引顾问所存储的消息数。一旦达到指定限制，**INDEX_ADVISOR** 将不会存储新的建议。但是它将继续为现有设备更新计数和时间戳。
- **NOEXEC** - 当设置为 **ON** 时，Sybase IQ 会生成查询计划，但不会执行整个查询。当 **EARLY_PREDICATE_EXECUTION** 选项为 **ON** 时，系统仍会执行查询的某些部分。如果 **EARLY_PREDICATE_EXECUTION** 为 **OFF**，则查询计划可能与正常运行查询时具有很大区别，因此不建议将它设置为 **OFF**。
- **QUERY_DETAIL** - 当此选项以及 **QUERY_PLAN** 或 **QUERY_PLAN_AS_HTML** 均为 **ON** 时，Sybase IQ 将在生成查询计划时显示有关查询的其它信息。当 **QUERY_PLAN** 和 **QUERY_PLAN_AS_HTML** 为 **OFF** 时，系统将忽略此选项。
- **QUERY_PLAN** - 当设置为 **ON** (缺省值) 时，Sybase IQ 会生成有关查询的消息。其中包括针对查询的有关使用连接索引、连接顺序及连接算法的消息。
- **QUERY_PLAN_TEXT_ACCESS** - 当将此选项设置为 **ON** 时，您可以从 Interactive SQL 客户端查看、保存和打印 IQ 查询计划。当 **QUERY_PLAN_ACCESS_FROM_CLIENT** 设置为 **OFF** 时，系统将不会对查询计划进行高速缓存，并且与查询计划相关的其它数据库选项对 Interactive SQL 客户端的查询计划显示不起任何作用。缺省情况下，此选项为 **OFF**。
请参见《参考：构件块、表和过程》中的“**GRAPHICAL_PLAN** 函数 [字符串]”和“**HTML_PLAN** 函数 [字符串]”。
- **QUERY_PLAN_AFTER_RUN** - 当设置为 **ON** 时，系统会在查询结束运行之后打印查询计划。它允许查询计划包括其它信息，如从查询中各节点传递的实际行数。要使此选项起作用，**QUERY_PLAN** 必须为 **ON**。缺省情况下，此选项为 **OFF**。
- **QUERY_PLAN_AS_HTML** - 生成 HTML 格式的图形化查询计划，以便在 Web 浏览器中查看。节点之间的超链接使得 .iqmsg 文件的 HTML 格式比文本格式更易于使用。使用 **QUERY_NAME** 选项，以便将查询名称包含在查询计划文件名中。缺省情况下，此选项为 **OFF**。
- **QUERY_PLAN_AS_HTML_DIRECTORY** - 当 **QUERY_PLAN_AS_HTML** 为 **ON** 并用 **QUERY_PLAN_AS_HTML_DIRECTORY** 指定一个目录时，Sybase IQ 会将 HTML 查询计划写入该指定目录中。
- **QUERY_PLAN_TEXT_CACHING** - 为用户提供了一种机制，用于控制对计划进行高速缓存的资源。在此选项为 **OFF** (缺省值) 的情况下，不会针对该用户连接对查询计划进行高速缓存。
如果针对某个用户关闭 **QUERY_PLAN_TEXT_ACCESS** 选项，则对于来自该用户的连接，无论 **QUERY_PLAN_TEXT_CACHING** 如何设置都不会对查询计划进行高速缓存。

另请参见《参考：构件块、表和过程》中的“**GRAPHICAL_PLAN** 函数 [字符串]”和“**HTML_PLAN** 函数 [字符串]”。

- **QUERY_TIMING** - 控制对查询引擎中有关子查询及其它某些重复函数的计时统计信息的收集活动。通常该选项应为 **OFF** (缺省值)，这是因为对于非常短的相关子查询来说，就性能而言每个子查询执行的计时成本可能很高。

注意：查询计划可向您的 `.iqmsg` 文件添加大量文本。当 **QUERY_PLAN** 为 **ON** 时，尤其是 **QUERY_DETAIL** 为 **ON** 时，可能需要启用消息日志回卷或消息日志存档，以避免填满消息日志文件。有关详细信息，请参见《系统管理指南第一卷》的“Sybase IQ 系统管理概述”中的“消息日志回卷”。

另请参见

- 查询树 (第 74 页)
- 使用查询计划 (第 74 页)

查询树

查询树表示查询的数据流。

查询树由节点组成。每个节点代表一个工作阶段。树上最低的节点是叶节点。每个叶节点代表查询中的一个表或预连接索引组。

在查询计划的顶部是运算符树的根。信息流从表开始向上流动，流经任何表示连接、排序、过滤、存储、集合和子查询的运算符。

另请参见

- 查询评估选项 (第 72 页)
- 使用查询计划 (第 74 页)

使用查询计划

设置 **QUERY_PLAN_AS_HTML** 选项可以生成 HTML 版本的查询计划，您可以在 Web 浏览器中查看此文件。

在 HTML 查询计划中，树中的每个节点都是连接至详细信息的超链接。每个框都被链接至上部的树。您可以单击任何节点，以便快速浏览整个计划。

授权用户现在可在 **Interactive SQL** 计划窗口中显示查询计划。用户还可以从 **Interactive SQL** 保存和打印查询计划，而不用访问服务器上的 `.iqmsg` 文件或查询计划文件。

SQL 函数 **GRAPHICAL_PLAN** 和 **HTML_PLAN** 分别返回 XML 和 HTML 格式的 IQ 查询计划，作为字符串结果集。数据库选项 **QUERY_PLAN_TEXT_ACCESS** 和 **QUERY_PLAN_TEXT_CACHING** 控制新函数的行为。

采用以下方式从 **Interactive SQL** 计划窗口中查看查询计划：

- 执行查询并打开计划窗口。根据从“计划”选项（“工具”>“选项”>“计划”）中选择的计划类型，计划窗口显示相应计划。

仅当选择 **GRAPHICAL_PLAN** 选项时，该窗口才会显示 IQ 查询计划。其它计划会返回错误消息“计划类型不受支持”。

- 在 SQL 语句窗口中输入查询，并从菜单“SQL”>“获取计划”中进行选择。根据从“计划”选项（“工具”>“选项”>“计划”）中选择的计划类型，计划窗口显示相应计划。
仅当选择 **GRAPHICAL_PLAN** 选项时，该窗口才会显示 IQ 查询计划。其它计划会返回错误消息“计划类型不受支持”。
- 使用 SQL 函数 **GRAPHICAL_PLAN** 和 **HTML_PLAN** 返回查询计划，作为字符串结果。

若要访问查询计划，请对下列查询使用 SQL 函数 **GRAPHICAL_PLAN** 和 **HTML_PLAN**：
SELECT、**UPDATE**、**DELETE**、**INSERT SELECT** 和 **SELECT INTO**。

若要从 **Interactive SQL** 保存查询计划，请使用 **GRAPHICAL_PLAN** 或 **HTML_PLAN** 检索查询计划并使用 **OUTPUT** 语句将输出保存到文件中。

若要查看保存的计划，请从 **Interactive SQL** 客户端菜单中选择“文件”>“打开”，导航到保存计划的目录。还可以通过选择“文件”>“打印”来打印计划窗口中显示的计划。

有关详细信息，请参见《参考：构件块、表和过程》中的“**GRAPHICAL_PLAN** 函数 [字符串]”和“**HTML_PLAN** 函数 [字符串]”。有关支持这些查询计划函数的选项，请参见《参考：语句和选项》中的“**QUERY_PLAN_TEXT_ACCESS** 选项”和“**QUERY_PLAN_TEXT_CACHING** 选项”。

另请参见

- 查询评估选项（第 72 页）
- 查询树（第 74 页）

控制查询处理

任何用户都可以对用在处理特定查询上的时间量设置限制。具有 **DBA** 特权的用户能够给予某些用户高于其他用户的查询优先级，或者更改处理算法以影响查询处理速度。

设置查询时间限制

设置 **MAX_QUERY_TIME** 选项可以限制查询能够运行的时间。如果查询的执行时间长于 **MAX_QUERY_TIME**，Sybase IQ 会停止查询并显示相应的错误。

注意： Sybase IQ 会将所有小数形式的 *option-value* 设置截断为整数值。例如，值 3.8 将被截断为 3。

另请参见

- 设置查询优先级（第 76 页）

- 设置查询优化选项（第 76 页）
- 设置用户提供的条件提示（第 77 页）
- 监控负载（第 78 页）

设置查询优先级

设置查询优先级选项可按用户分配查询处理优先级。

队列中等待处理的查询首先按照提交查询的用户优先级顺序排队以等待运行，其次是按照查询提交的顺序进行排队。在高优先级查询全部执行完毕之前，不会执行低优先级查询。

下列选项可按用户为查询分配处理优先级。

- **IQGOVERN_PRIORITY** - 为队列中等待处理的查询分配数字优先级（1、2 或 3，1 表示优先级最高）。
- **IQGOVERN_MAX_PRIORITY** - 允许 DBA 为用户或组设置 **IQGOVERN_PRIORITY** 的上限。
- **IQ_GOVERN_PRIORITY_TIME** - 如果高优先级（优先级为 1）查询在 -iqgovern 队列中等候的时间超过了指定时间，则允许高优先级用户开始执行查询。

若要查看查询的优先级，请查看 **sp_iqcontext** 存储过程返回的 **IQGovernPriority** 属性。

另请参见

- 设置查询时间限制（第 75 页）
- 设置查询优化选项（第 76 页）
- 设置用户提供的条件提示（第 77 页）
- 监控负载（第 78 页）

设置查询优化选项

优化选项影响查询处理速度。

- **AGGREGATION_PREFERENCE** - 控制用于处理集合（**GROUP BY**、**DISTINCT**、**SET** 函数）的算法的选择。此选项主要供内部使用；除非您是经验丰富的数据库管理员，否则请不要使用它。
- **DEFAULT_HAVING_SELECTIVITY_PPM** - 设置查询中所有 **HAVING** 谓词的选择性，以覆盖优化程序对 **HAVING** 子句要过滤的行数所做的估计。
- **DEFAULT_LIKE_MATCH_SELECTIVITY_PPM** - 为一般 **LIKE** 谓词设置缺省选择性，例如 **LIKE 'string%string'**，其中 % 是通配符。如果未提供其它选择性信息，并且匹配字符串不是以一组常量字符后跟单个通配符开头，则优化程序会使用此选项。

- **DEFAULT_LIKE_RANGE_SELECTIVITY_PPM** - 设置前导为常量字符的 LIKE 谓词的缺省选择性，形式如 `LIKE 'string%'`，其中匹配字符串是一组常量字符，后接单个通配符 (%)。如果未提供其它选择性信息，优化程序会使用此选项。
- **EARLY_PREDICATE_EXECUTION** - 控制在执行连接优化前是否执行简单本地谓词。在大多数情况下，不应对对其进行更改。
- **IN_SUBQUERY_PREFERENCE** - 控制处理 IN 子查询的算法选择。此选项主要供内部使用；除非您是经验丰富的数据库管理员，否则请不要使用它。
- **INDEX_PREFERENCE** - 为查询处理设置要使用的索引。Sybase IQ 优化程序一般会选择最佳可用索引来处理本地 **WHERE** 子句谓词和其它能够在 IQ 索引内部完成的操作。此选项用于取代供测试使用的优化程序选择；在大多数情况下，不应对对其进行更改。
- **JOIN_PREFERENCE** - 控制处理连接时的算法选择。此选项主要供内部使用；除非您是经验丰富的数据库管理员，否则请不要使用它。
- **JOIN_SIMPLIFICATION_THRESHOLD** - 控制在应用任何连接优化程序简化机制之前被连接在一起的最小表数。通常，不需要更改此值。
- **MAX_HASH_ROWS** - 设置查询优化程序将考虑执行散列算法的最大预计行数。缺省值为 2,500,000 行。例如，如果两表之间有连接，而这两个表中参与该连接的估计行数超过该选项值，则优化程序将不会考虑散列连接。在 **TEMP_CACHE_MEMORY_MB** 值超过每用户 50MB 的系统上，需要为该选项考虑更高的值。
- **MAX_JOIN_ENUMERATION** - 设置在应用优化程序简化之后要进行连接顺序优化的表的最大数量。通常无需设置此选项。

另请参见

- 设置查询时间限制（第 75 页）
- 设置查询优先级（第 76 页）
- 设置用户提供的条件提示（第 77 页）
- 监控负载（第 78 页）

设置用户提供的条件提示

选择性提示可以帮助优化程序选择适当的查询策略。

Sybase IQ 查询优化程序使用来自可用索引的信息选择用于执行查询的适当策略。对于查询中的每个条件，优化程序将决定是否可以使用索引执行该条件，如果可以，优化程序会选择使用的索引以及相对于该表中其它条件的执行顺序。在这些决策中最重要的因素是条件的选择性；即，表的行中满足该条件的部分。

优化程序通常在没有用户干预的情况下做出决定，并且一般可提供最佳决策。但在某些情况下，优化程序可能无法在条件执行之前准确确定其选择性。通常只有以下两种条件之一成立时，才会出现这些情况：一种是条件所处的列没有适当的可用索引，另一种是条件涉及某些算术或函数表达式，因而过于复杂，使优化程序无法准确估计。

有关语法、参数和示例，请参见《参考：构件块、表和过程》的“SQL 语言元素”中的“用户提供的条件提示”。

另请参见

- 设置查询时间限制（第 75 页）
- 设置查询优先级（第 76 页）
- 设置查询优化选项（第 76 页）
- 监控负载（第 78 页）

监控负载

使用存储过程来监控表、列和索引的使用情况，从而提高查询性能。

通常，创建索引的目的是为了提供优化元数据以及强制唯一性和主/外键关系。但是，创建索引之后，DBA 面临对索引带来的优势进行定量的挑战。

通常，表是在 IQ 主存储区中创建的，用于临时存储必须由多个连接访问或在较长时间内访问的数据。您可能会遗忘这些表，然而它们却继续使用宝贵的磁盘空间。此外，数据仓库中的表的数目过大且负载非常复杂，从而无法手动分析使用情况。

因此，未使用的索引和表会浪费磁盘空间，增加备份时间，并降低 DML 性能。

Sybase IQ 提供了用于收集和分析已定义负载的统计信息的工具。DBA 可快速确定查询正在引用并应该由此而保留的数据库对象。可删除未使用的表/列/索引，以便减少浪费的空间，提高 DML 性能，并缩短备份时间。

负载监控是使用存储过程实现的，这些存储过程控制对表、列和索引信息的详细使用情况的收集和报告。这些过程是对 **INDEX_ADVISOR** 功能的补充，该功能生成相应消息，这些消息建议增加列索引以提高一个或多个查询的性能。添加推荐的索引之后，即可跟踪其使用情况，以确定这些索引是否值得保留。

有关负载监控过程的详细信息，请参见《参考：构件块、表和过程》中的“`sp_iqcolumuse` 过程”、“`sp_iqindexadvice` 过程”、“`sp_iqindexuse` 过程”、“`sp_iqtableuse` 过程”、“`sp_iqunusedcolumn` 过程”、“`sp_iqunusedindex` 过程”、“`sp_iqunusedtable` 过程”和“`sp_iqworkmon` 过程”。

另请参见《参考：语句和选项》中的“**INDEX_ADVISOR** 选项”。

另请参见

- 设置查询时间限制（第 75 页）
- 设置查询优先级（第 76 页）
- 设置查询优化选项（第 76 页）
- 设置用户提供的条件提示（第 77 页）

优化删除操作

Sybase IQ 选择最佳算法来处理带有 **HG** 和 **WD** 索引的列的删除操作。

HG 删操作

Sybase IQ 在三种算法中选择任意一种算法来处理带有 **HG** (High_Group) 索引的列的删除操作。

- 当从非常少的组中删除行时，少量删除会表现出最佳性能。当只删除 1 行或删除操作对带有 **HG** 索引的列使用相等谓词时，通常会选择它。少量删除算法可以随机访问 **HG**。最坏情况下，I/O 也会与访问的组数成正比。
- 当从若干组中删除行，而组数又足够少使得所访问的 **HG** 页面不多时，中型删除会表现出最佳性能。中型删除算法可提供对 **HG** 的有序访问。最坏情况下，I/O 也不会超过索引页面的数量。中型删除会因为对要删除的记录进行排序而产生新的开销。
- 从大量组中删除行时，大型删除会表现出最佳性能。大型删除按顺序扫描 **HG**，直至删除所有行为止。最坏情况下，I/O 也不会超过索引页面的数量。大型删除也为并行操作，但其并行操作受索引内部结构和待删除记录所属组的分布的约束。**HG** 列上的范围谓词可用于缩小大型删除的扫描范围。

HG 删的开销

在 Sybase IQ 12.6 之前，**HG** 删开销模型只考虑了最坏情况下的 I/O 性能，因此大多数情况下首选的是大型删除。当前的开销模型将考虑多种因素，包括 I/O 开销、CPU 开销、可用资源、索引元数据、并行处理和查询中可用的谓词。

在带 **HG** 索引的列上指定谓词，会大大改善开销。若要让 **HG** 开销选择大型删除以外的算法，必须能够确定受删除影响的不同的值（组）的数量。离散值个数初始假定为索引组数和被删除行数中的较小者。谓词可以提供更完善、更精确的离散值个数估计值。

当前开销不考虑范围谓词对大型删除的影响。这会导致在采用大型删除将更快的情况下选择中型删除。在这种情况下，您可以根据需要按照下一节所述方法强制使用大型删除算法。

使用 **HG** 删性能选项

您可以使用 **HG_DELETE_METHOD** 选项来控制 **HG** 删的性能。

通过 **HG_DELETE_METHOD** 选项指定的参数值会强制使用指定的删除算法，如下所示：

- 1 = 少量删除
- 2 = 大型删除
- 3 = 中型删除

- **DML_OPTIONS5 = 4** (禁用推式删除谓词)。缺省值 0 - 禁止将范围谓词推入 **HG** 大型删除。

有关 **HG_DELETE_METHOD** 数据库选项的详细信息, 请参见《参考: 语句和选项》中的“**HG_DELETE_METHOD** 选项”。

另请参见

- **WD** 删除操作 (第 80 页)
- **TEXT** 删除操作 (第 81 页)

WD 删除操作

Sybase IQ 在三种算法中选择任意一种算法来处理带有 **WD** (Word) 索引的列的删除操作。

- 当被删除行中几乎不包含任何非重复字时, 并不需要访问很多 **WD** 页面, 因此少量删除会表现出最佳性能。 **WD** 少量删除算法执行 **WD** 有序访问。最坏情况下, I/O 也不会超过索引页面的数量。 少量删除包含对要删除记录中的文字和记录 ID 进行排序的开销。
- **WD** 中型删除是 **WD** 少量删除的变换形式, 适用于与少量删除相同的条件, 即, 当被删除行几乎不包含任何非重复字时。 **WD** 中型删除仅对要删除记录中的文字进行排序。排序为并行操作, 其并行操作受文字数和 CPU 可用线程数的约束。对于文字索引, 中型删除方法通常比少量删除更快。
- 当被删除行包含大量非重复文字时, 需要访问索引中的大量“组”, 此时大型删除会表现出最佳性能。大型删除按顺序扫描 **WD**, 直至删除所有行为止。最坏情况下, I/O 也不会超过索引页面的数量。大型删除也为并行操作, 但其并行操作受索引内部结构和待删除记录所属组的分布的约束。

WD 删除的开销

WD 删除的开销模型会考虑很多因素, 包括 I/O 开销、CPU 开销、可用资源、索引元数据和并行。

您可以使用 **WD_DELETE_METHOD** 数据库选项来控制 **WD** 删除的性能。

使用 **WD** 删除性能选项

通过 **WD_DELETE_METHOD** 选项指定的参数值会强制使用指定的删除算法, 如下所示:

- 0 = 开销模型选定的中型删除或大型删除
- 1 = 少量删除
- 2 = 大型删除
- 3 = 中型删除

有关 **WD_DELETE_METHOD** 数据库选项的详细信息, 请参见《参考: 语句和选项》的“**数据库选项**”中的“**WD_DELETE_METHOD** 选项”。

另请参见

- HG 删除操作 (第 79 页)
- TEXT 删除操作 (第 81 页)

TEXT 删除操作

Sybase IQ 在两种算法中选择任意一种算法来处理带有 **TEXT** 索引的列的删除操作。

- 当被删除行中几乎不包含任何非重复字时，并不需要访问很多 **TEXT** 页面，因此少量删除会表现出最佳性能。**TEXT** 少量删除算法执行 **TEXT** 有序访问。最坏情况下，I/O 也不会超过索引页面的数量。少量删除包含对要删除记录中的文字和记录 ID 进行排序的开销。
- 当被删除行包含大量非重复文字时，需要访问索引中的大量“组”，此时大型删除会表现出最佳性能。大型删除按顺序扫描 **TEXT**，直至删除所有行为止。最坏情况下，I/O 也不会超过索引页面的数量。大型删除也为并行操作，但其并行操作受索引内部结构和待删除记录所属组的分布的约束。

TEXT 删除的开销

TEXT 删除的开销模型会考虑很多因素，包括 I/O 开销、CPU 开销、可用资源、索引元数据和并行。

您可以使用 **TEXT_DELETE_METHOD** 数据库选项来控制 **TEXT** 删除的性能。

使用 **TEXT** 删除性能选项

通过 **TEXT_DELETE_METHOD** 选项指定的参数值会强制使用指定的删除算法，如下所示：

- 0 = 开销模型选定的中型删除或大型删除
- 1 = 少量删除
- 2 = 大型删除
- 3 = 中型删除

有关 **TEXT_DELETE_METHOD** 数据库选项的详细信息，请参见《Sybase IQ 中的非结构化数据分析》的“**TEXT** 索引和文本配置对象”中的“**TEXT_DELETE_METHOD** 选项”。

另请参见

- HG 删除操作 (第 79 页)
- WD 删除操作 (第 80 页)

在 32 位 Windows 系统上调优服务器

请参见调优准则以优化 Windows 32 位系统上的性能。

一般性能准则

将这些一般准则应用于装载和查询数据。

最低内存要求

在 Windows 下运行 Sybase IQ 的推荐最低内存 (RAM) 为 512MB。对于大多数应用程序，建议采用 4GB 以获得最佳性能。在 boot.ini 中设置 /3GB 和 /PAE 开关，以允许 Sybase IQ 在 Windows 32 位系统上寻址尽可能多的内存。

防止内存过度分配

系统页面错误过多是由于计算机上物理内存 (RAM) 的过度分配造成的。过多的页面故障可大幅降低 Sybase IQ 的性能。可通过认真分配 Sybase IQ 缓冲区、监控 Sybase IQ 进程的虚拟地址空间以及监控可用的物理内存，防止内存的过度分配。此节介绍监控 Sybase IQ 对计算机物理内存的使用情况的准则。

监控物理内存

可供应用程序 (Sybase IQ) 使用的物理内存量显示在“物理内存(K)”下方。如果“可用数”值始终低于 5000，则计算机的物理内存可能存在过度分配。这是因为，在达到 5000(K) 标志时，Windows 会产生页面错误以维持最小 5MB 可用内存。

若要监控物理内存，请从“任务管理器”中选择“性能”选项卡。

文件系统

Windows 文件系统支持在文件、目录和卷级别进行压缩。检查压缩选项，并对存储 Sybase IQ 数据库的所有磁盘和卷禁用 Windows 文件系统压缩。

Sybase IQ 提供内置压缩功能。Windows 文件系统压缩不但不能进一步减少数据库大小，而且可能会在执行读取或写入时增加 CPU 开销。

另请参见

- 最大化网络应用程序数据吞吐量 (第 83 页)

最大化网络应用程序数据吞吐量

启用网络服务服务器选项“最大化网络应用程序数据吞吐量”可以最大化吞吐量。

1. 在“控制面板”上，双击“网络连接”。

2. 右键单击“本地连接”，然后选择“属性”。
3. 选择“Microsoft 网络的文件和打印机共享”，然后单击“属性”。
4. 在“优化”下，选择“最大化网络应用程序数据吞吐量”。

注意：在某些 Windows 版本上，可能需要安装 Microsoft Internet Information Services (IIS) 才能将服务器属性设置为最大化网络应用程序数据吞吐量。

另请参见

- 文件系统 (第 83 页)

监控性能

使用 Sybase IQ 性能监视器、Windows 任务管理器和 Windows 性能工具可以监控系统性能。

虚拟地址空间

为了避免出现过多的系统页面故障，虚拟地址空间应小于计算机的物理内存。

进程的虚拟地址空间为该进程的总大小。进程的工作集为当前分配给该进程的物理内存大小。为了避免出现过多的系统页面故障，Sybase IQ 进程的虚拟地址空间应小于计算机的物理内存。

由于在 Sybase IQ 服务器上使用了虚拟内存模式，虚拟内存碎片可能会导致 Windows 平台上的处理工作大幅增加。为了减少发生此情况的可能性，Sybase IQ 支持在 Windows XP 和 Windows Server 2003 上使用 Microsoft 的低碎片堆 (LFH)。

监控虚拟地址空间和内存使用情况

更改任务管理器中的选项，以监控虚拟地址空间和内存使用情况。

1. 右键单击任务栏上的空白空间。
2. 选择“任务管理器”，然后单击“进程”选项卡。
3. 单击“查看”，然后选择“选择列”。
4. 在“选择列”对话框上，选择下列项：
 - 内存使用
 - 内存使用增量
 - 内存使用高峰值
 - 页面错误
 - 页面错误增量
 - 虚拟内存大小

另请参见

- 监控页面故障 (第 85 页)

监控页面故障

使用 Windows 性能监视器可以跟踪硬页面故障的数量，这些故障可能表明计算机的物理内存已过度分配。

1. 在“控制面板”上，单击“管理工具”，然后选择“性能”。
 2. 选择 Sybase IQ 进程。
 3. 选择计数器“**Page Faults/sec**”。
- 此计数器的结果中包括软页面故障和硬页面故障。硬页面故障是因磁盘 I/O 导致的页面故障。软页面故障一般不是性能问题。
4. 若要确定硬页面故障的数量，请选择“**LogicalDisk**”对象和文件 `pagefile.sys` 所在位置的实例（该实例应该与 Sybase IQ 数据库不在同一个卷上）。
 5. 选择“**Disk Transfers/sec**”计数器。

将此值与“**Page Faults/sec**”值比较，即可了解硬页面故障在整个页面故障中所占的百分比。理想情况下，应该仅有极少或者甚至没有任何针对页面文件的 I/O 活动。不过，在低内存配置下，可能会发生分页。

如果硬页面错误率持续高于每秒 20，则表明计算机的物理内存已过度分配。

注意：有关使用性能工具的详细信息，请在“性能”中单击“帮助”，然后选择“帮助主题”。

另请参见

- 监控虚拟地址空间和内存使用情况（第 84 页）

NTFS 高速缓存

使用 NTFS 高速缓存可以提高插入和查询的性能。

NTFS 在使用同等大小物理内存的情况下可存储比 Sybase IQ 缓冲区高速缓存多很多的数据。通过降低主高速缓存的大小，将此内存提供给可以更有效地使用它的 NTFS，可以最有效地利用这一优点。在 IQ 15.x 中，在装载和查询处理中将使用更多临时高速缓存。因此，使用 NTFS 高速缓存对于主高速缓存仍是一项非常有用的技术，但不会提供临时高速缓存的优点。

Sybase IQ 缓冲区高速缓存以未压缩的形式来存储 Sybase IQ 数据（页面）。因此，100MB 的 Sybase IQ 缓冲区高速缓存可存储相当于 100MB 的数据。与之相反的是，NTFS 高速缓存以压缩后的形式管理 Sybase IQ 数据。因此，如果压缩比率为 2:1，则 100MB 的 NTFS 高速缓存可能会存储 200MB 的 Sybase IQ 数据。因此，NTFS 高速缓存有可能保持更高的高速缓存命中率，从而导致 I/O 减少。节省的 I/O 超过将数据从 NTFS 高速缓存移动到 Sybase IQ 缓冲区高速缓存时对数据解压缩所需的计算开销。

另请参见

- 一般性能准则 (第 83 页)
- 监控性能 (第 84 页)
- 插入和查询 (第 86 页)
- 备份操作 (第 86 页)

插入和查询

经过合理调优的 Sybase IQ 插入操作会展现某些特征。您可以从 Windows 任务管理器和 Windows 性能工具中查看这些特征。

I/O 操作

- 插入操作通常为 CPU 密集型。系统内的所有 CPU 在运行时使用率都应接近 100%，其中 CPU 的 95% 或更高使用率处于用户模式。通过单击 Windows 任务管理器的“性能”选项卡并设置“查看” - “显示内核时间”选项，便可轻松查看到此信息。
- 物理内存不应过度分配，特别是，Sybase IQ 进程的虚拟地址空间应小于计算机的物理内存 (RAM)。
- 硬页面故障 (针对 `pagefile.sys` 所在卷的 I/O) 应该很低，理想情况下应接近于 0 (零)。
- 针对 IQ 存储的 I/O 操作应保持稳定，并处于磁盘子系统的 I/O 能力之内。

装载性能

Sybase IQ 使用 Windows CreateFile 选项 (用于创建和打开文件) 来指定要读取以便后续访问的文件。此选项用于在 **LOAD TABLE** 命令中指定的文件。这样，由于可提前读取并降低了对 NTFS 高速缓存内存的利用，装载的性能得到了提高。

可通过将 Sybase IQ 的主缓冲区高速缓存大小设置为远小于计算推荐值，来进一步提高装载性能。您可将 Sybase IQ 主缓冲区高速缓存设置为比计算推荐值小 50%。

对查询进行调优

减少主缓冲区高速缓存大小以提高查询性能。

另请参见

- 主缓冲区高速缓存和临时缓冲区高速缓存 (第 6 页)
- NTFS 高速缓存 (第 85 页)

备份操作

将 **BLOCK FACTOR** 设置为接近于最大块大小，以优化备份操作。

Windows 仅支持固定大小的 I/O 设备。这就是说，对磁带的每次读取或写入的大小必须与其前后的读取或写入的大小相同。如有任何读取/写入操作超出了硬件设备的

容量，则相应操作将失败。这意味着，对于备份和恢复操作，只有所有写入（或读取）都以相应硬件所配置的大小进行，备份（或恢复）才可能成功。

Sybase IQ 缺省值设计用于在每一平台上都实现尽可能高效的读取和写入操作。不过，如果您在创建 Sybase IQ 数据库时覆盖了缺省的块大小，则需要在备份该数据库时调整块因子。

对于任何备份或恢复：

$$\text{block size} \times \text{block factor} \approx \text{I/O size}$$

若要在 Windows 系统上调整块因子，必须知道可以通过磁带设备处理的最大物理块大小。驱动器制造商通常未针对此信息提供文字说明。若要确定此值（通常为 64KB），您需要使用 WIN32 API 调用编写一个小程序。然后，必须使用该数据库的块大小和 **BACKUP** 命令的 **BLOCK FACTOR** 选项，才能优化备份性能。有关完整的语法和用法，请参见《参考：语句和选项》。

使每一 I/O 操作的大小越接近于块大小的最大值，则备份性能越高。需使用与块大小相乘时乘积与相应驱动器的块大小最接近的 **BLOCK FACTOR**（整数）。

请记住，Sybase IQ 在每个块被写入时会向块添加一些额外的数据，用以实现数据完整性。因此，如果数据库块大小为 8192，磁带设备可处理的最大块大小为 128KB，即使 $8192 \times 16 = 128KB$ ，也不可使用 16 作为块因子。必须考虑到 Sybase IQ 会对每一 I/O 操作添加的额外数据，从而使用 15 为 **BLOCK FACTOR**。请注意，在 Windows 上，对于 128KB 的缺省数据库块大小和缺省 IQ 页大小，缺省的块因子是 15。

另请参见

- 一般性能准则（第 83 页）
- 监控性能（第 84 页）
- NTFS 高速缓存（第 85 页）
- 插入和查询（第 86 页）

索引

符号

-append | - truncate 58
 -bufalloc 54
 -c 9
 -cache 50
 -cache_by_type 52
 -ch 9
 -cl 9
 -contention 55
 -debug 59
 -file_suffix 53
 -gm 8
 -gn 9
 -interval 57
 -io 53
 -iqlgovern 8
 -iqmt 10
 -summary 50
 -threads 57

A

AGGREGATION_ALGORITHM_PREFERENCE
 76
 AGGREGATION_PREFERENCE 70

B

BT_PREFETCH_MAX_MISS 23

保留空间选项 19

备份

 调优块大小 86

表

 连接 27

 折叠 27

C

CACHE_PARTITIONS 61

CPU

 监控 67

 监控 (UNIX) 63

 监控 (Windows) 63

 可用性 20

设置数目 20
 统计信息 43
 策略文件位置 16
 插入操作
 NT CreateFile 选项 86
 调优 86
 查询 76
 Group BY 限制 70
 HG 删除操作 79
 ORDER BY, 增强 71
 TEXT 删除操作 81
 WD 删除操作 80
 查询处理 75
 查询计划 74
 查询树 74
 查询优先级 76
 调优 86
 负载监控 78
 高速缓存方法 72
 计划 72
 控制 76
 连接 76
 评估选项 72
 删除操作 79
 时间限制 75
 条件提示 77
 限制按行 21
 限制并发 19
 优化 27, 76
 优化, 删除选项 79
 优化程序简化 76
 子查询性能 71
 查询,
 GROUP BY 影响 69
 结构化 69
 优化查询 69
 查询处理
 监控 78
 控制 75, 77
 优先级 76
 查询服务器
 平衡负载 25
 查询计划 72
 评估选项 72

- 生成但不执行 72
 - 使用 74
 - 图形 74
 - 查询计划, 选项
 - INDEX_ADVISOR 72
 - NOEXEC 72
 - QUERY_DETAIL 72
 - QUERY_PLAN 72
 - QUERY_PLAN_AFTER_RUN 72
 - QUERY_PLAN_AS_HTML 72
 - QUERY_PLAN_AS_HTML_DIRECTORY 72
 - QUERY_PLAN_TEXT_ACCESS 72
 - QUERY_PLAN_TEXT_CACHING 72
 - QUERY_TIMING 72
 - 查询树 74
 - 查询优化选项
 - AGGREGATION_ALGORITHM_PREFERENCE 76
 - DEFAULT_HAVING_SELECTIVITY 76
 - DEFAULT_LIKE_MATCH_SELECTIVITY 76
 - DEFAULT_LIKE_RANGE_SELECTIVITY 76
 - EARLY_PREDICATE_EXECUTION 76
 - IN_SUBQUERY_PREFERENCE 76
 - INDEX_PREFERENCE 76
 - JOIN_ALGORITHM_PREFERENCE 76
 - MAX_HASH_ROWS 76
 - 查询执行
 - 分布式 25
 - 超线程
 - 服务器开关 20
 - 磁带设备
 - Windows 86
 - 磁盘高速缓存
 - 定义 26
 - 性能影响 26
 - 磁盘空间
 - Multiplex 数据库 25
 - 交换空间 2
 - 磁盘条带化
 - Sybase IQ 14
 - 定义 14
 - 规则 14
 - 内部 15
 - 在装载中使用 14
 - 存储 I/O 统计信息 46
 - 存储过程
 - 查看分析数据 36
 - 性能监控 35
- ## D
- DEFAULT_HAVING_SELECTIVITY 76
 - DEFAULT_LIKE_MATCH_SELECTIVITY 76
 - DEFAULT_LIKE_RANGE_SELECTIVITY 76
 - 低碎片堆 10
 - 调优
 - 32 位 Windows 服务器 83
 - 插入操作 86
 - 查看环境 35
 - 查询 86
 - 性能 35
 - 性能和调优问题 86
 - 动态性能监视器 40
 - 抖动, 缓冲区管理器
 - HASH_PINNABLE_CACHE_PERCENT 62
 - HASH_THRASHING_PERCENT 62
 - 要执行的操作 62
 - 段
 - 数据库, 使用多个 16
 - 堆
 - 低碎片 10
 - 多列索引 71
 - 多线程
 - 性能影响 12
- ## E
- EARLY_PREDICATE_EXECUTION 76
- ## F
- FLATTEN_SUBQUERIES 71
 - FORCE_NO_SCROLL_CURSORS 22
 - FROM 子句 30
 - 非规范化
 - 原因 29
 - 分布式查询处理 25
 - 分区
 - 定义 13
 - 分区表 30
 - 分页
 - 管理 2
 - 在 UNIX 上监控 63
 - 在 Windows 上监控 63

服务器
 监控性能 40
负载监控 78
负载平衡
 在查询服务器间 25

G

GROUP BY
 查询影响 69
 拆分查询示例 70
 限制 70
 性能建议 69
高速缓存 85
 IQ 主缓冲区大小和临时缓冲区大小 6
 NTFS 85
 缓冲区 85
 另请参见缓冲区高速缓存 48
 统计信息 41
 预取页 23
高速缓存方法
 使用 72
工作空间 18
固定长度的 I/O
 Windows 86
管理, 资源 1
 缓冲区高速缓存 1, 3
过程, 系统
 sp_iqcolumnuse 78
 sp_iqindexuse 78
 sp_iqtableuse 78
 sp_iqunusedcolumn 78
 sp_iqunusedindex 78
 sp_iqunusedtable 78
 sp_iqworkmon 78
过程分析
 sa_procedure_profile 40
 sa_procedure_profile_summary 40
 过程 36
 过程摘要 38
 在 Interactive SQL 中查看数据 38
过程分析信息
 ISQL 39
 sa_procedure_profile 39

H

HASH_PINNABLE_CACHE_PERCENT 62
HASH_THRASHING_PERCENT 62

HG 索引
 多列 71
函数
 查看分析数据 36
缓冲区
 禁用操作系统缓冲 10
缓冲区高速缓存 85
 IQ 主缓冲区和临时缓冲区 6
 布局 61
 大小要求 6
 高速缓存大小 6
 管理 3
 监控器 48
 监控器输出选项 50
 监控清单 64
 开销 4
 块大小 7
 临时 6
 临时存储 6
 内存, 操作系统 4
 内存, 节省 7
 内存, 应用程序 4
 内存使用 4
 确定大小 4
 设置, 目录 8
 设置大小 6
 数据库访问, 多用户 4
 数据压缩 7
 物理内存 6
 线程堆栈 4
 页大小 7
 主 6
 主数据库 6
 注意事项 6
缓冲区高速缓存监控器 48
缓冲区高速缓存选项
 MAIN_CACHE_MEMORY_MB 6
 TEMP_CACHE_MEMORY_MB 6
缓冲区管理器
 抖动 62
缓冲区管理器抖动
 HASH_PINNABLE_CACHE_PERCENT 62
 HASH_THRASHING_PERCENT 62
 要执行的操作 62

I

I/O
 性能建议 13

- 直接 10
- IN_SUBQUERY_PREFERENCE 76
- INDEX_ADVISOR 72
- INDEX_PREFERENCE 76
- IOS_FILE_CACHE_BUFFERING 11
- IQ PATH 选项
 - 选择原始设备 13
- IQ 存储
 - 工作空间 18
 - 缓冲区高速缓存大小 6
- IQ_USE_DIRECTIO 11
- iqgovern 开关
 - 限制查询以提高性能 19
- IQGOVERN_MAX_PRIORITY 选项 76
- IQGOVERN_PRIORITY 76
- IQMSG 日志
 - 设置最大大小 18
- iqnumbercpus
 - 设置 CPU 数 20
- iqwmem 开关 10
- J**
 - JAVA_HEAP_SIZE 12
 - JOIN_ALGORITHM_PREFERENCE 76
 - JOIN_PREFERENCE 30
 - 计划
 - 查询 72
 - 查询计划 74
 - 监控
 - 事务状态 40
 - 物理内存 83
 - 监控负载 78
 - 监控器
 - 设置输出文件位置 49
 - 监控器输出选项
 - append | - truncate 58
 - bufalloc 54
 - cache 50
 - cache_by_type 52
 - contention 55
 - debug 59
 - file_suffix 53
 - interval 57
 - io 53
 - summary 50
 - threads 57
 - 监控性能
 - 工作集 84

- 虚拟地址空间 84
- 页面故障 85
- 交换
 - 磁盘空间要求 2
 - 监控 2
- 交换文件
 - 对性能的影响 2
- 结构化查询 69
- 进程
 - 工作集 84
 - 增加 10
- 进程线程模型 12

K

- 开销
 - 缓冲区高速缓存 4
- 块大小
 - 与 IQ 页大小的关系 7

L

- 连接
 - 连接请求 8
 - 统计信息 44
 - 限制语句 23
- 连接索引
 - 性能影响 27
- 连系内存 10
- 列
 - 大量空值 69
- 临时存储
 - 缓冲区高速缓存大小 6
- 逻辑服务器 25

M

- MAIN_CACHE_MEMORY_MB 6
- MAX_CURSOR_COUNT 22
- MAX_HASH_ROWS 76
- MAX_QUERY_TIME 选项 75
- MAX_STATEMENT_COUNT 23
- monitor
 - IQ UTILITIES syntax 49
 - starting and stopping 49
- Multiplex
 - 性能监视器 40
- Multiplex 数据库
 - 磁盘空间 25

- 内存 3
- Multiplex 资源
动态调整 25
- 目录存储
文件增长 28
- 目录缓冲区高速缓存设置 9
- N**
- NOEXEC 72
- NTFS 高速缓存
提高性能 85
- 内部条带化 15
- 内存
IOS_FILE_CACHE_BUFFERING 11
IQ_USE_DIRECTIO 11
JAVA_HEAP_SIZE 12
Multiplex 数据库 3
操作系统 4
多线程 12
服务器 3
缓冲区高速缓存 3
缓冲区高速缓存大小 4
交换 2
进程线程模型 12
开销 4
连接请求 8
连系 10
另请参见
 缓冲区高速缓存 6
平衡 I/O 13
启动选项 8
启用 Java 的数据库 12
轻量进程 12
数据库访问, 多用户 4
碎片 10
特定于平台的内存选项 10
为用户优化 8
文件系统缓冲 11
线程堆栈 4
应用程序 4
优化 1
原始分区 4
增加 2
- 内存, 节省
 页大小 7
- 内存使用
 其它 4
- 内存使用统计信息 41
- O**
- ORDER BY
 查询性能 71
- ORDER BY 子句 71
- OS_FILE_CACHE_BUFFERING 11
OS_FILE_CACHE_BUFFERING_TEMPDB 11
- P**
- PREFETCH_BUFFER_LIMIT 23
- 平衡 I/O
 保留空间选项 19
 策略文件位置 16
 磁盘条带化 14
 多个文件 16
 工作空间 18
 内部条带化 15
 原始 I/O 13
评估选项
 查询 72
- Q**
- 启动选项 8
 -c 8
 -ch 8
 -cl 8
 -gm 8
 -gn 8
 -iqgovern 8
 -iqmt 8
 轻型进程 12
 清理程序线程 61
 请求统计信息 45
- R**
- RAWDETECT
 磁盘条带化选项 15
- 任务管理器 84
- S**
- sa_procedure_profile 39
SET OPTION 12
sp_iqcolumnuse 78

- sp_iqindexuse 78
- sp_iqtableuse 78
- sp_iqunusedcolumn 78
- sp_iqunusedindex 78
- sp_iqunusedtable 78
- sp_iqworkmon 78
- SUBQUERY_CACHING_PREFERENCE 72
- SUBQUERY_FLATTENING_PERCENT 71
- SUBQUERY_FLATTENING_PREFERENCE 71
- SWEeper_THREADS_PERCENT 61
- 删除操作
 - HG 79
 - TEXT 81
 - WD 80
 - 优化 79
- 设备
 - Windows 上的 I/O 86
- 使用
 - 典型 24
- 事件
 - 查看分析数据 36
- 事务日志
 - 关于 17
 - 截断 17
 - 停止的数据库 18
 - 脱机数据库 18
- 事务统计信息 45
- 事务状态
 - 监控 40
- 输出选项
 - 缓冲区高速缓存监控器 50
- 输出选项, 监控器 58
 - bufalloc 54
 - cache 50
 - cache_by_type 52
 - contention 55
 - debug 59
 - file_suffix 53
 - interval 57
 - io 53
 - summary 50
 - threads 57
- 数据
 - 存储 85
- 数据库
 - 段 16
 - 对象分析 37
 - 对象分析信息 37
 - 非规范化以提高性能 29
- 分析 37
- 分析设置 37
- 分析统计信息 36
- 管理 28
- 过程 36
- 过程分析 36
- 数据库访问
 - 多用户 4
- 数据库空间
 - 定位以达到最佳性能 16
 - 使用情况统计信息 47
 - 限制使用 20
- 数据压缩
 - 页大小 7
- 顺序磁盘 I/O 16
- 碎片 10
- 索引
 - HG 27, 71
 - LF 27
 - 多列 71
 - 类型 27
 - 索引顾问 27
 - 选择 27

T

- TEMP_CACHE_MEMORY_MB 6, 70
- 条件
 - 用户提供的 77
- 统计信息
 - 动态 40
- 吞吐量 1
 - 最大化 83

U

- UNION ALL
 - 规则 30
 - 视图 30
 - 视图性能 31
 - 装载 29
- USER_RESOURCE_RESERVATION 24

V

- vmstat 命令
 - 监控 UNIX 上的缓冲区高速缓存 63

W

WASH_AREA_BUFFERS_PERCENT 61
 WD 删除操作 80
 Windows
 32 位服务器 83
 I/O 设备 86
 NTFS 高速缓存 85
 备份操作 86
 插入和查询 86
 磁带设备 86
 调优 32 位 Windows 服务器 83
 工作集 84
 过度分配内存 83
 监控性能 84, 85
 进程 84
 内存使用情况 84
 任务管理器 84
 文件系统 83
 物理内存 83
 性能准则 83
 虚拟地址空间 84
 最大化吞吐量 83
 最低内存要求 83

网络

大型数据传输 31
 设置 31
 网络 31
 性能 31
 性能建议 31

网络统计信息 47

文件

定位以达到最佳性能 16

文件系统缓冲 11

物理内存

缓冲区高速缓存 6
 监控 83

X

系统触发器

查看分析数据 36

系统过程

sp_iqcolumnuse 78
 sp_iqindexuse 78
 sp_iqtableuse 78
 sp_iqunusedcolumn 78
 sp_iqunusedindex 78
 sp_iqunusedtable 78

sp_iqworkmon 78

系统资源

管理 1
 连接 8
 内存 1
 启动选项 8
 性能注意事项 1
 资源使用选项 19

下推连接 30

线程

缓冲区高速缓存 61
 监控 57

线程堆栈

内存 4
 线程统计信息 43

响应时间 1

消息日志

Sybase IQ 18

性能

磁盘高速缓存 26
 定义 1
 动态性能监视器 40
 多用户 23
 监控 48
 监控和调优 35
 平衡 I/O 13
 设计 1
 数据库过程分析信息 36
 选择正确索引类型 27
 用 iqgovern 限制查询 19
 注意事项 1
 子查询 71

性能调优

简介 35

性能监视器

服务器级 40

虚拟地址空间 84

虚拟内存

碎片 10

选项

AGGREGATION_ALGORITHM_
 PREFERENCE 76

AGGREGATION_PREFERENCE 70

BT_PREFETCH_MAX_MISS 23

CACHE_PARTITIONS 61

DEFAULT_HAVING_SELECTIVITY 76

DEFAULT_LIKE_MATCH_SELECTIVITY
 76

- DEFAULT_LIKE_RANGE_SELECTIVITY 76
 EARLY_PREDICATE_EXECUTION 76
 FLATTEN_SUBQUERIES 71
 HASH_PINNABLE_CACHE_PERCENT 62
 HASH_THRASHING_PERCENT 62
 IN_SUBQUERY_PREFERENCE 76
 INDEX_ADVISOR 72
 INDEX_PREFERENCE 76
 IQ_USE_DIRECTIO 11
 JAVA_HEAP_SIZE 12
 JOIN_ALGORITHM_PREFERENCE 76
 JOIN_PREFERENCE 30
 MAIN_CACHE_MEMORY_MB 6
 MAX_HASH_ROWS 76
 MAX_STATEMENT_COUNT 23
 NOEXEC 72
 OS_FILE_CACHE_BUFFERING 11
 OS_FILE_CACHE_BUFFERING_TEMPDB 11
 PREFETCH_BUFFER_LIMIT 23
 QUERY_DETAIL 72
 QUERY_PLAN 72
 QUERY_PLAN_AFTER_RUN 72
 QUERY_PLAN_AS_HTML 72
 QUERY_PLAN_AS_HTML_DIRECTORY 72
 QUERY_PLAN_TEXT_ACCESS 72
 QUERY_PLAN_TEXT_CACHING 72
 QUERY_TIMING 72
 SET OPTION 12
 SUBQUERY_CACHING_PREFERENCE 72
 SUBQUERY_FLATTENING_PERCENT 71
 SUBQUERY_FLATTENING_PREFERENCE 71
 SWEEPER_THREADS_PERCENT 61
 TEMP_CACHE_MEMORY_MB 6, 70
 USER_RESOURCE_RESERVATION 24
 WASH_AREA_BUFFERS_PERCENT 61
- 选项, 查询计划
 INDEX_ADVISOR 72
 NOEXEC 72
 QUERY_DETAIL 72
 QUERY_PLAN 72
 QUERY_PLAN_AFTER_RUN 72
 QUERY_PLAN_AS_HTML 72
 QUERY_PLAN_AS_HTML_DIRECTORY 72
 QUERY_PLAN_TEXT_ACCESS 72
- QUERY_PLAN_TEXT_CACHING 72
 QUERY_TIMING 72
 选项, 查询优化
 AGGREGATION_ALGORITHM_PREFERENCE 76
 DEFAULT_HAVING_SELECTIVITY 76
 DEFAULT_LIKE_MATCH_SELECTIVITY 76
 DEFAULT_LIKE_RANGE_SELECTIVITY 76
 EARLY_PREDICATE_EXECUTION 76
 IN_SUBQUERY_PREFERENCE 76
 INDEX_PREFERENCE 76
 JOIN_ALGORITHM_PREFERENCE 76
 MAX_HASH_ROWS 76
- 选项, 缓冲区高速缓存
 MAIN_CACHE_MEMORY_MB 6
 TEMP_CACHE_MEMORY_MB 6
- 选项值
 截断 72
- Y**
- 页大小
 减少内存 7
 块大小 7
 内存, 节省 7
 确定 7
 数据压缩 7
- 页面
 解压缩 85
- 页面故障
 监控 85
- 用户提供的条件
 查询 77
- 优化查询 27, 69, 71
- 游标
 强制为非滚动游标 22
 限制数目 22
- 语句
 限制语句 23
- 预取的高速缓存页 23
- 预取的行
 控制 24
- 原始分区
 内存使用 4
 文件系统 4
- 原始设备
 对性能的影响 13

Z

在 Interactive SQL 中查看过程分析信息 38
直接 I/O 10
主数据库
 缓冲区高速缓存大小 6
装载数据
 调优 86
 使用条带化磁盘 14
资源
 Multiplex 25
资源管理
 缓冲区高速缓存 3
资源使用
 Multiplex 磁盘空间 25
 磁盘高速缓存 26
 负载平衡 25
 使用 UNION ALL 装载 29
索引 27

提高 25
网络性能 31
限制数据库访问 26
资源使用选项 19
 典型使用 24
 强制为非滚动游标 22
 设置可用 CPU 20
 限制按行查询 21
 限制并发查询 19
 限制数据库空间使用 20
 限制游标 22
 限制语句 23
 预取的行 24
 预取高速缓存页 23
子查询
 提高性能 71
 展平 71
子查询性能 71
子查询展平 71

