



性能和调优指南

SAP Sybase IQ 16.0 SP03

文档 ID: DC00953-01-1603-01

最后修订日期: 2013 年 11 月

© 2013 SAP 股份公司或其关联公司版权所有, 保留所有权利。

未经 SAP 股份公司明确许可, 不得以任何形式或为任何目的复制或传播本文的任何内容。本文包含的信息如有更改, 恕不另行事先通知。

由 SAP 股份公司及其分销商营销的部分软件产品包含其它软件供应商的专有软件组件。各国的产品规格可能不同。

上述资料由 SAP 股份公司及其关联公司 (统称“SAP 集团”) 提供, 仅供参考, 不构成任何形式的陈述或保证, 其中如若存在任何错误或疏漏, SAP 集团概不负责。与 SAP 集团产品和服务相关的保证仅限于该等产品和服务随附的保证声明 (若有) 中明确提出之保证。本文中的任何信息均不构成额外保证。

SAP 和本文提及的其它 SAP 产品和服务及其各自标识均为 SAP 股份公司在德国和其它国家的商标或注册商标。

如欲了解更多商标信息和声明, 请访问: <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>。

目录

读者	1
性能注意事项	3
硬件配置	5
设置可用 CPU 数	5
进程线程模型	5
网络性能	6
服务器配置	9
了解内存	9
服务器内存	9
所需内存	9
高速缓存内存	11
大内存	11
IQ 页大小	12
绑定内存	12
调优选项	13
针对典型使用进行优化	13
为大量用户进行优化	14
限制并发查询	15
限制查询临时空间	16
限制按行返回的查询	17
强制游标为非滚动游标	17
限制游标数	18
限制语句数	19
预取高速缓存页	19
控制预取行数	20
控制文件系统缓冲	20
优化高速缓存分区	22
平衡输入/输出	23
原始设备	23
磁盘条带化	23
内部条带化	24

随机和顺序文件访问	25
事务和消息日志	25
监控性能	26
数据库分析过程	27
事件分析过程	28
关键性能指标	29
缓冲区高速缓存性能	30
Multiplex 性能	35
管理 Multiplex 磁盘空间	35
管理逻辑服务器资源	35
平衡查询负载	35
模式设计	37
索引	37
索引提示	37
使用索引的时间和位置	38
简单索引选择标准	39
HG 索引装载	40
多列索引	41
连接列	41
主键	42
外键	43
适当确定数据类型的大小	43
空值	44
无符号数据类型	45
LONG VARCHAR 和 LONG VARBINARY	45
大对象存储	46
临时表	47
非范式化以提高性能	48
可用来实现更快装载的 UNION ALL 视图	49
引用 UNION ALL 视图的查询	50
UNION ALL 视图性能	50
散列分区	51
故障排除	53
隔离性能问题	53

诊断工具	53
常见性能问题	54
分页和磁盘交换	54
索引和行碎片	55
目录文件增大	55
抖动和查询执行	56
查询和删除	57
结构化查询	57
增强 ORDER BY 查询性能	57
提高了子查询性能	58
使用高速缓存方法	58
生成查询计划	58
查询评估选项	59
使用查询计划	60
控制查询处理	61
设置查询时间限制	61
设置查询优先级	61
设置查询优化选项	62
设置用户提供的条件提示	63
监控负载	63
优化删除操作	64
HG 删除操作	64
WD 删除操作	65
TEXT 删除操作	66
索引	67

读者

数据库管理员、数据库设计人员和开发人员可借助此文档来配置 SAP® Sybase® IQ 以提高其性能。

读者

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。

响应时间是完成单个任务所用的时间。可影响响应时间的因素包括：

- 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。

为使 SAP Sybase IQ 在正确配置的系统上实现最大性能增益运行，需精心设计并选择适合的索引策略。

其它考查事项（如硬件和网络分析）可确定安装中的瓶颈。

硬件配置

影响 SAP Sybase IQ 性能的硬件问题。

设置可用 CPU 数

设置 **-iqnumbercpus** 启动开关可以指定可用的 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 并启用超线程的计算机上，可将 **-iqnumbercpus** 设置为实际内核数
- 已使用操作系统实用程序的计算机可将 SAP Sybase IQ 限定为计算机内 CPU 的子集

其它信息

《管理：数据库》 > “运行数据库服务器” > “命令行开关” > “性能的命令行选项” > “内存选项” > “CPU 开关数”。

进程线程模型

SAP Sybase IQ 使用操作系统内核线程，以获得最佳性能。缺省情况下，SAP Sybase IQ 以系统中的 CPU 数为依据分配线程数。

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。操作系统并不知道用户线程是什么，但是能够检测到用户线程是否正在等待或能够运行。

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

单个、高度线程化进程服务于所有 SAP Sybase IQ 用户。数据库服务器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，SAP 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 SAP Sybase IQ 线程。还有可能是针对连接数设置的 **-iqmt** 过低。

用于管理线程用法的 SAP Sybase IQ 选项

- 使用服务器启动选项 **-iqmt** 可以设置最大线程数。缺省值通过连接数和 CPU 数计算得来，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深度超过此限制的错误的消息，则该值可能会增加。
- 使用 `SET OPTION MAX_IQ_THREADS_PER_CONNECTION` 命令可以设置单个用户的最大线程数。`SET OPTION MAX_IQ_THREADS_PER_TEAM` 命令可以设置可用于一组线程的线程数，允许您约束分配到单个操作的线程数（以及由此导致的系统资源数量）。
- 使用这些选项可以控制特定操作消耗的资源数量。例如，可在发出 **INSERT**、**LOAD**、**BACKUP DATABASE** 或 **RESTORE DATABASE** 命令之前设置此选项。

网络性能

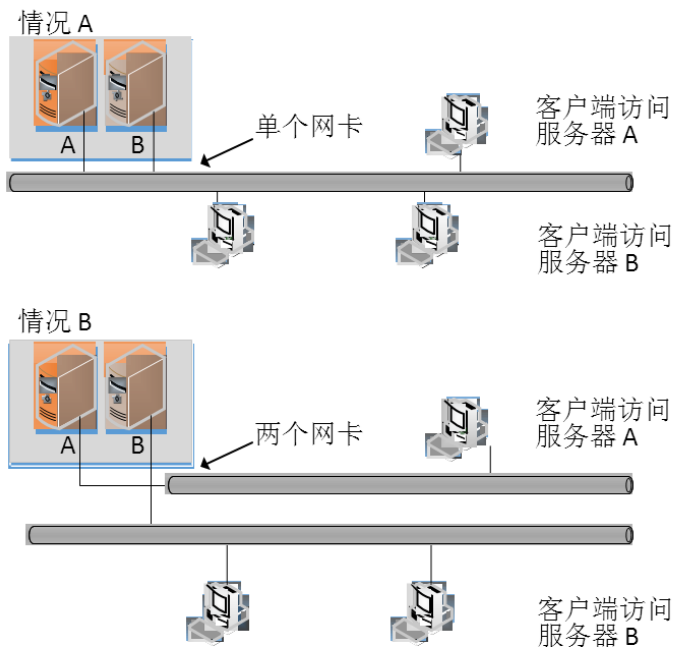
环境中的微小更改可以解决某些网络性能问题。

要提高网络吞吐量，请提供多个网络适配器。根据服务级别协议，可以将各类用户分配到不同的网络。

在下图示例 **A** 中，访问两个不同数据库服务器的客户端使用一个网卡。即，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：隔离重网络负荷用户



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。

服务器配置

了解内存

了解 SAP Sybase IQ 分配内存的方式可以帮助系统达到最佳性能。

服务器内存

SAP 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，SAP 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 SAP 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。确保在运行 SAP Sybase IQ 之前将操作系统内核正确配置为使用共享内存。

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。

用于 SAP Sybase IQ 主要和临时缓冲区高速缓存的内存总量，加上 SAP 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。

另请参见

- 所需内存（第 9 页）
- 高速缓存内存（第 11 页）
- 大内存（第 11 页）
- IQ 页大小（第 12 页）
- 绑定内存（第 12 页）

所需内存

在确定操作系统和其它应用程序所需的物理内存之后，计算 SAP Sybase IQ 所需的剩余内存。

原始分区与文件系统

对于类似 UNIX 的操作系统，使用文件系统而不是原始分区的数据库可能需要另外 30% 的剩余内存来处理操作系统的文件缓冲。在 Windows 上，应当通过设置 `OS_FILE_CACHE_BUFFERING = 'OFF'`（针对新数据库的缺省设置）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，SAP 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，可能有 30 个用户连接到 SAP Sybase IQ，但仅有 10 个左右用户可能在某一时间同时使用数据库。

用于线程堆栈的内存

处理线程需要少量内存。您使用的处理线程越多，需要的内存就越多。**-iqmt** 服务器开关控制线程数。**-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约为： $(-gn * (-gss + -iqtss)) + (-iqmt * -iqtss)$ 。

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并发执行的任务数（包括用户和系统请求）。**-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。SAP Sybase IQ 使用以下公式计算这些工作线程的堆栈大小： $(-gss + -iqtss)$ 。

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，可通过在 **BACKUP DATABASE** 命令中调整 **BLOCK FACTOR** 来提高备份性能，但增加 **BLOCK FACTOR** 也会增加所使用的内存量。
- 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表及表的各个字段和索引。根据表的数量以及这些表中的累计列数和索引数，**sp_iqcheckdb** 可能需要非常少或非常多的虚拟内存。要限制需要的内存量，请使用 **sp_iqcheckdb** 选项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表、文件和索引，因此它所使用的虚拟内存同 **sp_iqcheckdb** 在检查整个数据库时使用虚拟内存相当。它使用临时缓冲区高速缓存来跟踪所使用的块。

另请参见

- 服务器内存（第 9 页）
- 高速缓存内存（第 11 页）
- 大内存（第 11 页）
- IQ 页大小（第 12 页）
- 绑定内存（第 12 页）

高速缓存内存

请尽可能多地为 IQ 主要和临时缓冲区高速缓存分配内存以获得最佳性能。更改缺省设置以适应装载、查询和应用程序。

主缓冲区高速缓存和临时缓冲区高速缓存的缺省大小都为 64MB。高速缓存大小通过 `-iqmc` (主缓冲区高速缓存) 和 `-iqtc` (临时高速缓存) 服务器启动选项进行控制。这些启动选项仅在服务器运行时保持有效, 因此每次重新启动服务器时也需要包括它们。

大内存需求是分配到 SAP Sybase IQ 的总可用物理内存的三分之一。为确保有足够的内存可供主 IQ 存储库和临时 IQ 存储库使用, 应设置 `-iqlm`、`-iqmc` 和 `-iqtc` 启动参数, 这样每个参数都会接收到总可用物理内存的三分之一。

另请参见

- 服务器内存 (第 9 页)
- 所需内存 (第 9 页)
- 大内存 (第 11 页)
- IQ 页大小 (第 12 页)
- 绑定内存 (第 12 页)

大内存

`-iqlm` 启动参数指定 SAP Sybase IQ 可从操作系统动态申请的最大的大内存量。

某些装载操作可能需要比缺省提供的 2GB 更大的内存。如果内存需求超过了缺省内存量, 请使用 `-iqlm` 启动选项来增加 SAP Sybase IQ 能够从操作系统动态申请的内存。将 `-iqlm` 设置为开关, 将其作为启动服务器的命令或配置文件的一部分。

大内存分配

一般情况下, 大内存需求是分配到 SAP Sybase IQ 的总可用物理内存的三分之一。为确保有足够的内存可供主 IQ 存储和临时 IQ 存储使用, 应设置 `-iqlm`、`-iqtc` 和 `-iqmc` 启动参数, 这样每个参数都会接收到分配给 SAP Sybase IQ 的总可用物理内存的三分之一。

在大多数情况下, 应将总物理内存的 80% 分配到 SAP Sybase IQ, 以避免换出 SAP Sybase IQ 进程。调整实际内存分配, 以容纳在同一系统中运行的其它进程。例如, 在拥有 32 个内核和 128GB 总可用物理内存的计算机上, 应将 100GB (大约是总计 128GB 的 80%) 分配给 SAP Sybase IQ 进程。根据一般规则, 要将 `-iqlm`、`-iqtc` 和 `-iqmc` 参数分别设置为 33GB。

另请参见

- 服务器内存 (第 9 页)
- 所需内存 (第 9 页)
- 高速缓存内存 (第 11 页)

- IQ 页大小 (第 12 页)
- 绑定内存 (第 12 页)

IQ 页大小

IQ 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。

SAP Sybase IQ 以页大小为单位执行 I/O。创建数据库时，应为目录存储库和 IQ 存储库指定单独的页大小。临时存储库的页大小与 IQ 存储库相同。

目录存储库的页大小不会对性能产生实际影响。缺省值 4096 字节应当足够。IQ 页大小决定着其它两个性能因素：缺省 I/O 传输块大小和数据库最大数据压缩。SAP Sybase IQ 可压缩所有数据。压缩量取决于 IQ 页大小。

节省内存

如果计算机没有足够的内存，请增加内存并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。

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。

另请参见

- 服务器内存 (第 9 页)
- 所需内存 (第 9 页)
- 高速缓存内存 (第 11 页)
- 大内存 (第 11 页)
- 绑定内存 (第 12 页)

绑定内存

在 HP 和 Solaris 平台上，可以指派指定数量的内存作为绑定内存。绑定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。

连系内存池

在 HP 和 Solaris 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，绑定内存可能会提高 SAP Sybase IQ 的性能。但是，SAP Sybase IQ 的专用绑定内存不能用于该计算机上的其它应用程序。

要仅在这些 UNIX 平台上创建绑定内存池，请指定 **-iqwmem** 命令行开关，以指示绑定内存大小(MB)。(您必须是 **root** 用户才能设置 **-iqwmem**，在 Solaris 上除外。)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：仅当您有足够内存可专用于为此目的指定的内存量时，才使用 **-iqwmem**。否则，可能会导致性能大幅下降。

- 在 Solaris 上，**-iqwmem** 始终提供绑定内存。
- 在 HP 上，如果您作为 **root** 启动服务器，则 **-iqwmem** 提供绑定内存。如果您不是以 **root** 启动服务器，则提供非绑定内存。此行为可能在未来版本中有所更改。

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` 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 SAP Sybase IQ 的虚拟内存量。有关建议值，请参见《安装和配置指南》。

另请参见

- 服务器内存（第 9 页）
- 所需内存（第 9 页）
- 高速缓存内存（第 11 页）
- 大内存（第 11 页）
- IQ 页大小（第 12 页）

调优选项

通过调优选项可加快查询速度。

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。

SAP Sybase IQ 可跟踪打开的游标数，并相应地分配内存。在某些情况下，可以使用 `USER_RESOURCE_RESERVATION` 选项调整 SAP Sybase IQ 认为正在使用产品的当前游标的最小数量，从而节省从临时高速缓存中分配的内存。

只在需要时设置此选项。如果需要设置此选项，请与技术支持部门联系，以了解详细信息。

另请参见

- 为大量用户进行优化（第 14 页）

- 限制并发查询 (第 15 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 预取高速缓存页 (第 19 页)
- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 页)

为大量用户进行优化

为大量用户调整启动参数。

表 1. 服务器启动选项

参数	描述
-gm	<p>设置缺省连接数。用法： <code>-gm #_connections_to_support</code></p> <p>尽管这表示服务器支持的连接总数，但并非所有连接都同时处于活动状态。</p>
-iqgovern	<p>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 -iqgovern 限制，则新查询将排队，直到某个活动查询完成。用法： <code>-iqgovern #_ACTIVE_queries_to_support</code></p> <p>-iqgovern 的最佳值取决于查询的性质、CPU 数和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，将此选项设置为 $2 * numCPU + 4$ 可能会提供更好的吞吐量。</p>
-gn	<p>设置多个用户同时运行时将用于目录存储库和连接的执行线程数。用法： <code>-gn number of tasks (both user and system requests) that the database server can execute concurrently</code></p> <p>-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 设置过低会阻止服务器正确操作。建议不要将 -gn 设置为超过 480。</p>

参数	描述
-c	<p>设置目录存储高速缓存大小。用法：</p> <pre>-c catalog_store_cache_size</pre> <p>目录高速缓存大小很大程度上取决于模式大小与对象数。目录存储缓冲区高速缓存也是目录存储库的常规内存池。要以 MB 为单位指定，请使用 -c nM 格式，例如 -c 64M。</p> <p>如果高达 1000 名用户，将 -c 设置为 16MB 或更高。如果高达 2000 名用户，将 -c 设置为 48MB（缺省值）。</p>
-cl and -ch	<p>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过小，请设置 -cl 和 -ch 参数。不得将 -c 和 -ch 或 -cl 用于同一配置文件或命令行中。有关信息，请参见 -ch cache-size 选项。</p>
-iqmt	<p>设置处理线程数。如果对于 -gm 设置而言，-iqmt 设置过低，则可能会导致线程匮乏。</p>

注意： 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命令中执行以下操作之一（但不要同时执行这两项操作）：

- 设置 **-c** 参数
- 使用 **-ch** 和 **-cl** 参数为目录存储库高速缓存大小设置特定的上限和下限

另请参见

- 针对典型使用进行优化（第 13 页）
- 限制并发查询（第 15 页）
- 限制查询临时空间（第 16 页）
- 限制按行返回的查询（第 17 页）
- 强制游标为非滚动游标（第 17 页）
- 限制游标数（第 18 页）
- 限制语句数（第 19 页）
- 预取高速缓存页（第 19 页）
- 控制预取行数（第 20 页）
- 控制文件系统缓冲（第 20 页）
- 优化高速缓存分区（第 22 页）

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，连接数由许可证控制。

最佳 **-iqgovern** 值会提供并发查询访问的正确数量以提供最佳吞吐量。如果通过此阈值设置 **-iqgovern**，连接或资源不足时，所有请求的速度都会降低。

通过指定 **-iqgovern** 开关，可以帮助 SAP Sybase IQ 优化将缓冲区数据调度到磁盘上的分页过程，避免过度使用内存。**-iqgovern** 的缺省值是 $(2 \times \text{CPU 数}) + 10$ 。您可能需要试验确定理想值。对于具有大量活动连接的站点，可尝试将 **-iqgovern** 设置得略低些。

另请参见

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 预取高速缓存页 (第 19 页)
- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 页)

限制查询临时空间

设置 `QUERY_TEMP_SPACE_LIMIT` 以指定拒绝查询前临时空间量的最大估计值。

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。

SAP Sybase IQ 可估计处理该查询所需的临时空间。如果估计值超过当前的 `QUERY_TEMP_SPACE_LIMIT` 设置，SAP 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。

要限制每个连接的实际临时存储库使用情况，需为所有 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库使用情况。如果连接超过 `MAX_TEMP_SPACE_PER_CONNECTION` 选项设置的限额，则会返回错误并回退当前语句。

另请参见

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制并发查询 (第 15 页)
- 限制按行返回的查询 (第 17 页)

- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 预取高速缓存页 (第 19 页)
- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 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量结果集的查询。

`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
```

设置此选项只能拒绝可能消耗过多资源的查询。

另请参见

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制并发查询 (第 15 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 预取高速缓存页 (第 19 页)
- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 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。

使用未声明主机变量的滚动游标时，**SAP 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相互独立。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 **SAP Sybase IQ** 对每个向后滚动重新执行查询。

如果应用程序很少执行向后滚动，请将 `FORCE_NO_SCROLL_CURSORS = 'ON'` 设置为永久性 `PUBLIC` 选项。它将使用更少内存，并能提高查询性能。

另请参见

- 针对典型使用进行优化（第 13 页）
- 为大量用户进行优化（第 14 页）
- 限制并发查询（第 15 页）
- 限制查询临时空间（第 16 页）
- 限制按行返回的查询（第 17 页）
- 强制游标为非滚动游标（第 17 页）
- 限制游标数（第 18 页）
- 限制语句数（第 19 页）
- 预取高速缓存页（第 19 页）
- 控制预取行数（第 20 页）
- 控制文件系统缓冲（第 20 页）
- 优化高速缓存分区（第 22 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` 选项限制连接一次最多可以使用的游标数。缺省值为 `50`。如果将此选项设置为 `0`，则允许使用无限数目的游标。

另请参见

- 针对典型使用进行优化（第 13 页）
- 为大量用户进行优化（第 14 页）
- 限制并发查询（第 15 页）
- 限制查询临时空间（第 16 页）
- 限制按行返回的查询（第 17 页）
- 强制游标为非滚动游标（第 17 页）
- 限制语句数（第 19 页）
- 预取高速缓存页（第 19 页）

- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 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` 选项限制连接一次最多可以使用的预准备语句数。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)，则可将 `MAX_STATEMENT_COUNT` 选项设置为更高的值。

另请参见

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制并发查询 (第 15 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 预取高速缓存页 (第 19 页)
- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 页)

预取高速缓存页

设置 `BT_PREFETCH_MAX_MISS` 选项可以控制预取内存行为。

`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` 索引进行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

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制并发查询 (第 15 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 控制预取行数 (第 20 页)

- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 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。这是客户端选项，可在 ODBC 连接对话框或在 `.odbc.ini` 文件中进行设置。

预取可以提高仅提取 `relative 1` 或 `relative 0` 的游标的性能。可通过两个连接参数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。在特定条件下，增加预取的行数可能会提高性能：

- 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

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制并发查询 (第 15 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 预取高速缓存页 (第 19 页)
- 控制文件系统缓冲 (第 20 页)
- 优化高速缓存分区 (第 22 页)

控制文件系统缓冲

在某些文件系统中，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` 角色设置此选项。必须关闭并重新启动数据库，更改才能生效。

由于 `Multiplex` 功能需要共享的原始存储库，所以 `Multiplex` 数据库不支持 `direct I/O` 文件系统设备。只有 `Simplex` 数据库支持 `direct I/O` 性能选项。

此 **direct I/O** 性能选项仅在 **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**，**direct 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 **direct I/O**，还要将环境变量 **IQ_USE_DIRECTIO** 设置为 **1**。缺省情况下，**direct I/O** 在 **Linux** 内核版本 **2.6** 和 **AIX** 上处于禁用状态。**IQ_USE_DIRECTIO** 对 **Solaris** 和 **Windows** 不起作用。

注意：

- 在 **Linux** 内核版本 **2.4** 中，**SAP Sybase IQ** 不支持 **direct I/O**。如果在 **Linux** 内核版本 **2.4** 中设置了 **IQ_USE_DIRECTIO** 环境变量，则 **SAP Sybase IQ** 服务器不会启动。系统将报告错误 `'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'`。
如果 **LOAD TABLE** 输入文件为指向某数据文件的符号链接，则在 **Linux** 内核版本 **2.6** 中 **direct I/O** 可能会失败并提示相同的错误消息。
 - **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缓冲区高速缓存页，因而导致操作系统分页活动过多并且性能降低。请尽可能在 **Solaris** 上使用数据库的原始设备。
 - **Windows** 可以通过损耗一定的文件系统性能来调整缓存分页算法，使之符合程序的行为特征以达到性能提升的目的。建议使用 **Windows** 的此特性，以保证 **SAP Sybase IQ** 的性能。
-

另请参见

- 针对典型使用进行优化（第 13 页）
- 为大量用户进行优化（第 14 页）
- 限制并发查询（第 15 页）
- 限制查询临时空间（第 16 页）
- 限制按行返回的查询（第 17 页）
- 强制游标为非滚动游标（第 17 页）
- 限制游标数（第 18 页）
- 限制语句数（第 19 页）
- 预取高速缓存页（第 19 页）
- 控制预取行数（第 20 页）
- 优化高速缓存分区（第 22 页）

优化高速缓存分区

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

SAP 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预期慢，通过更改 `CACHE_PARTITIONS` 数据库选项的值可能会有所提高。

当缓冲区接近高速缓存的最近最少使用 (LRU) 末尾时，它们将越过清洗标记。SAP Sybase IQ 会将最旧页 (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 SAP 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 SAP 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 缓冲区被脏抢夺因而 SAP Sybase IQ 必须先将其写出到磁盘才能使用的次数。

通常，SAP 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。

其它信息

- 《参考：语句和选项》> “数据库选项”> “按字母顺序排列的选项列表”> “`CACHE_PARTITIONS` 选项”
- 《参考：语句和选项》> “数据库选项”> “按字母顺序排列的选项列表”> “`SWEEPER_THREADS_PERCENT` 选项”
- 《参考：语句和选项》> “数据库选项”> “按字母顺序排列的选项列表”> “`WASH_AREA_BUFFERS_PERCENT` 选项”

另请参见

- 针对典型使用进行优化 (第 13 页)
- 为大量用户进行优化 (第 14 页)
- 限制并发查询 (第 15 页)
- 限制查询临时空间 (第 16 页)
- 限制按行返回的查询 (第 17 页)
- 强制游标为非滚动游标 (第 17 页)
- 限制游标数 (第 18 页)
- 限制语句数 (第 19 页)
- 预取高速缓存页 (第 19 页)
- 控制预取行数 (第 20 页)
- 控制文件系统缓冲 (第 20 页)

平衡输入/输出

使用磁盘条带化、随机和顺序文件磁盘访问来平衡输入/输出 (I/O)。

原始设备

对于类似 UNIX 的操作系统，您可在原始设备或文件系统文件上创建数据库或 `dbspace`。

磁盘分区通常有两种访问模式：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，它每次都在缓冲区中收集数据，直到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 `dbspace`。SAP 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。原始分区可以为任意大小。

另请参见

- 磁盘条带化（第 23 页）
- 内部条带化（第 24 页）
- 随机和顺序文件访问（第 25 页）
- 事务和消息日志（第 25 页）

磁盘条带化

对多个磁盘上的数据进行条带化处理是获得良好性能的重要技术。

可在系统的不同位置执行磁盘条带化，通常可作为 RAID 硬件和软件的一部分来执行，例如：

- 在设备层，例如磁盘阵列或控制器。
- 在操作系统或专用设备管理软件（例如 Veritas）中。
- 在应用程序中。

缺省情况下，SAP Sybase IQ 对 `dbspace` 中的所有文件范围内的页面进行内部条带化，因此无需额外在软件或硬件级别条带化来优化性能。当然，作为实现数据库存储冗余的一部分（如在使用 RAID-5 时），可能需要进行额外的条带化。

可通过简单的镜像或 "RAID-1" 获得 SAP Sybase IQ 存储冗余的最佳性能。如上所述，SAP Sybase IQ 将在 `dbspace` 内的所有双磁盘镜像集中分配数据。

考虑到成本，大多数 SAP Sybase IQ 将不会使用镜像，而是通过执行 RAID-5 或类似 RAID 级别来实现冗余。因为 RAID-5 的写入开销很大，通过 RAID-5 选择合适的块大小（移动到下一个磁盘之前写入磁盘的数据量）将对系统性能产生重大影响。若您的应用程序执行经常性的或时效性的装载、更新或删除，或经常查询临时 `dbspace I/O`，则 SAP Sybase IQ 数据库页大小 25-50% 范围内的较小的块大小将有可能实现最

佳性能。如果您的应用程序大多执行读取活动，几乎没有写入活动，则 SAP Sybase IQ 页大小 75-100% 范围内的较大的块大小将有可能实现最佳性能。

因为 SAP Sybase IQ 通常试图并行预取多页读取或刷新多页写入（即使只有单个活动查询），所以在多个磁盘中的每个页读取或写入活动中使用极小的块大小没有任何益处，而且通常有损性能。

使用 RAID 时，通常可通过基于 RAID 的硬件（如控制器或数组）获得最佳性能。基于 RAID 工具的软件将运行良好，但可能在服务器 CPU 上添加附加的中等性能装载。

另请参见

- 原始设备（第 23 页）
- 内部条带化（第 24 页）
- 随机和顺序文件访问（第 25 页）
- 事务和消息日志（第 25 页）

内部条带化

磁盘条带化利用多个磁盘主轴，并提供并行磁盘写入的速度。

SAP Sybase IQ 提供磁盘条带化选项，而无需使用第三方软件。如果您已经拥有通过第三方软件和硬件实现的磁盘条带化解决方案，则应改用此方法。可以通过为 CREATE DBSPACE 命令指定 STRIPING ON 选项启用磁盘条带化。

在创建 dbspace 时更改缺省条带化：

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

对于所有平台，DEFAULT_DISK_STRIPING 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 dbspace。当磁盘条带化为 OFF 时，dbspace（磁盘段）在逻辑文件上从前填充，每次填充一个磁盘段。

如果更改 DEFAULT_DISK_STRIPING 的值，不指定条带化首选项的所有后续 CREATE DBSPACE 操作都将受到影响。

启用磁盘条带化时，可以使用 ALTER DBSPACE DROP 命令从 dbspace 中删除文件。但是，在删除 dbspace 之前，必须使用 sp_iqemptyfile 存储过程重新定位该 dbspace 中的所有数据。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 dbspace 上。

另请参见

- 原始设备（第 23 页）
- 磁盘条带化（第 23 页）
- 随机和顺序文件访问（第 25 页）
- 事务和消息日志（第 25 页）

随机和顺序文件访问

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 存储库、临时存储库、目录存储库的文件以及程序（包括 SAP 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。

为了避免磁盘瓶颈：

- 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 dbspace 应仅使用物理设备（磁盘或硬件 RAID 集）的一个分区。
- 将 SAP Sybase IQ 数据库 I/O 与其它数据库 I/O 或密集型应用程序 I/O 隔离。
- 将数据库文件、临时 dbspace 和事务日志文件放在数据库服务器所在的物理计算机上。

另请参见

- 原始设备（第 23 页）
- 磁盘条带化（第 23 页）
- 内部条带化（第 24 页）
- 事务和消息日志（第 25 页）

事务和消息日志

管理事务和消息日志的大小以节约磁盘空间。

事务日志文件包含恢复和审计信息。为了避免产生数据库文件碎片和介质故障，建议您在使用该选项时将事务日志存放在与数据库所在位置不同的单独设备或分区上。

事务日志可随着时间的增加消耗大量的磁盘空间。定期截断事务日志以节约磁盘空间。

为了截断日志：

1. 关闭服务器。
2. 使用 `-m` 参数作为 `start_iq` 命令或 `.cfg` 文件的一部分来启动服务器。
3. 在不使用 `-m` 参数的情况下关闭并重新启动服务器。

请不要永久性设置 `-m` 开关。如果设置了 `-m`，对保存数据库文件的设备上的介质故障将没有保护。请在重新启动服务器后将 `-m` 从 `.cfg` 中删除。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`)。

警告！ SAP 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。如果由于某些原因导致数据库文件丢失，您将失去数据库（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。

消息日志

SAP 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。限制此文件的大小以节约磁盘空间。

在某些站点上，消息日志文件常常会迅速增长。为了限制此文件的大小：

- 设置最大文件大小并在活动消息日志已满时存档日志文件
- 增加 NOTIFY_MODULUS 数据库选项设置
- 使用 NOTIFY 参数关闭 LOAD TABLE 中的通知消息。INSERT 和 CREATE INDEX 语句
- 使用 **-iqmsgsz** 开关限制消息日志的大小

其它信息

- 实用程序指南 > “start_iq 数据库服务器启动实用程序” > “start_iq 服务器选项” > “-iqwmem iqsrv16 服务器选项”
- 实用程序指南 > “start_iq 数据库服务器启动实用程序” > “start_iq 服务器选项” > “-m iqsrv16 服务器选项”
- 《参考：语句和选项》 > “数据库选项” > “按字母顺序排列的选项列表” > “NOTIFY_MODULUS 选项”。
- 《参考：语句和选项》 > “SQL 语句” > “CREATE INDEX 语句”。
- 《参考：语句和选项》 > “SQL 语句” > “INSERT 语句”。
- 《参考：语句和选项》 > “SQL 语句” > “LOAD 语句”。

另请参见

- 原始设备（第 23 页）
- 磁盘条带化（第 23 页）
- 内部条带化（第 24 页）
- 随机和顺序文件访问（第 25 页）

监控性能

用于确定系统是否充分利用可用资源的工具。

数据库分析过程

返回数据库使用情况统计的存储过程。

表 2. 数据库分析过程

名称	描述
sp_iqconnection	<p>显示有关连接和版本的信息，包括哪些用户正在使用临时 dbspace、哪些用户正在使版本保持活动状态、连接在 SAP Sybase IQ 内部执行哪些操作、连接状态、数据库版本状态，等等。用法：</p> <pre>sp_iqconnection [connhandle]</pre> <p>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”</p>
sp_iqcontext	<p>按照连接跟踪和显示有关当前执行的语句的信息。用法：</p> <pre>sp_iqcontext [connhandle]</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcontext 过程”</p>
sp_iqcheckdb	<p>检查当前数据库的有效性。可选择更正 dbspace 或数据库的分配问题。用法：</p> <pre>sp_iqcheckdb 'mode target [...] [resources resource-percent]'</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqcheckdb 过程”</p>
sp_iqdbstatistics	<p>报告最近一次 sp_iqcheckdb 的结果。用法：</p> <pre>sp_iqdbstatistics</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqdbstatistics 过程”</p>
sp_iqdbsize	<p>显示当前数据库的大小。用法：</p> <pre>sp_iqdbsize([main])</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqdbsize 过程”</p>
sp_iqspaceinfo	<p>显示数据库中每个对象的空间使用情况。用法：</p> <pre>sp_iqspaceinfo ['main [table table-name index index-name] [...] ']</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqspaceinfo 过程”</p>

名称	描述
sp_iqstatus	显示有关数据库的杂项状态信息。用法： <pre>sp_iqstatus</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqstatus 过程”</p>
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。用法： <pre>sp_iqtablesize (table_owner.table_name)</pre> <p>请参见《构件块、表和过程》>“系统过程”>“系统存储过程”>“sp_iqtablesize 过程”</p>

另请参见

- 事件分析过程 (第 28 页)
- 关键性能指标 (第 29 页)
- 缓冲区高速缓存性能 (第 30 页)

事件分析过程

事件分析过程返回存储过程、功能、事件以及触发器的性能统计信息。

表 3. 事件分析过程

名称	描述
sa_server_option	在 Interactive SQL 中设置数据库分析选项。用法： <pre>CALL sa_server_option ('procedureprofiling', 'ON')</pre>
sa_procedure_profile	返回数据库过程、函数、事件或触发器的每一行的执行时间。用法： <pre>sa_procedure_profile [filename [, save_to_file]])</pre>
sa_procedure_profile_summary	汇总所有过程、函数、事件和触发器的执行时间。用法： <pre>sa_procedure_profile_summary [filename [, save_to_file]])</pre>

其它信息

《参考：构件块、表和过程》>“系统过程”>“按字母顺序排列的目录存储过程列表”：

- sa_server_option 系统过程
- sa_procedure_profile 系统过程
- sa_procedure_profile_summary 系统过程

另请参见

- 数据库分析过程 (第 27 页)
- 关键性能指标 (第 29 页)
- 缓冲区高速缓存性能 (第 30 页)

关键性能指标

在 SAP Control Center 中建立一个统计信息集合来监控服务器上的关键性能指标 (KPI)。KPI 值分组为集合并显示在 SCC 监控器中。

关键性能区域包括 SAP Sybase IQ 服务器、Multiplex 服务器及逻辑服务器

SAP Sybase IQ 服务器统计信息

各种服务器状态和使用统计信息

表 4. SAP Sybase IQ 服务器统计信息

关键性能区域	使用统计信息
SAP Sybase IQ 可用性统计信息	资源状态、CPU 使用、内存分配、高速缓存使用以及活动连接。
概述统计信息	服务器状态、CPU 使用、内存分配及当前活动连接。
连接统计信息	活动的、可用的、恢复的、回退的、挂起的用户和节点间连接。还显示平均每分钟的连接数和连接断开数。
DBSpace 和 DBSpace 文件统计信息	DbSPACE 和 dbSPACE 文件大小和可用百分比。
存储输入和输出统计信息	存储输入和输出统计信息标识每秒存储的读取和写入次数。
高速缓存统计信息	主缓冲区、目录以及临时高速缓存统计信息。
操作和请求统计信息	平均、最小、最大和总计等待、活动以及总的操作数。
网络统计信息	网络统计信息显示网络活动。
事务统计信息	当前正在服务器上运行的事务详细信息。

Multiplex 和节点相关的统计信息

Multiplex 和 Multiplex 服务器节点相关的统计信息

表 5. Multiplex 和节点相关的统计信息

关键性能区域	
Multiplex 可用性	每个 Multiplex 节点的可用性统计信息。
Multiplex 状态	Multiplex 的状态。
Multiplex 节点属性	每个 Multiplex 节点的角色、状态以及故障转移状态。
Multiplex 链接可用性	辅助节点和协调器间的节点间通信状态。

逻辑服务器统计信息

逻辑服务器和逻辑服务器节点相关的统计信息

表 6. 逻辑服务器统计信息

关键性能区域	使用统计信息
逻辑服务器可用性	逻辑服务器状态。
逻辑服务器引擎统计信息	CPU 使用、连接和连接统计信息。
逻辑服务器连接	可用连接的平均、最小、最大和总计数量。
逻辑服务器事务	平均、最小、最大和总计事务数。还显示平均和最小装载事务的数量。
逻辑服务器高速缓存统计信息	针对目录、临时以及主缓冲区高速缓存的平均、最小和最大高速缓存使用统计信息。
逻辑服务器操作和请求统计信息	平均、最小、最大、总计等待和活动的操作数。

其它信息

有关详细信息，请参见 SCC 中针对 SAP Sybase IQ 的 SAP Control Center 联机帮助，或访问 <http://sybooks.sybase.com/sybooks/sybooks.xhtml?prodID=10680>。

另请参见

- 数据库分析过程（第 27 页）
- 事件分析过程（第 28 页）
- 缓冲区高速缓存性能（第 30 页）

缓冲区高速缓存性能

缓冲区高速缓存性能是总体性能中的关键因素。IQ UTILITIES 语句启动收集缓冲区高速缓存统计信息的高速缓存监控器。使用高速缓存监控器的输出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。

请查看此清单以查找出超出正常范围的高速缓存行为。

表 7. 缓冲区高速缓存监控器清单

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超过 90%。开始时它可能低于 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 <code>-iqmc</code> 和 <code>-iqtc</code> 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 <code>PREFETCH_THREADS_PERCENT</code> 选项尝试增加预取线程数。</p>
Gdirty (脏抢夺)	<p>在有中等高速缓存大小 (< 10GB) 的系统中为 0。</p>	<p><code>GDirty > 0</code></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 <code>GDirty/GrabbedDirty</code> 大于 0 并且 I/O 率 (写入) 很低, 则系统可能直接被轻装载, 而不需要执行任何操作。</p>	<p>调整 <code>SWEEPER_THREADS_PERCENT</code> 选项 (缺省值 10%) 或 <code>WASH_AREA_BUFFERS_PERCENT</code> 选项 (缺省值 20%), 以增加清洗区的大小。</p>

统计信息	正常行为	需要调整的行为	建议操作
BWaits (缓冲区忙等待)	0	持久 > 0, 表示多个作业正在争夺相同缓冲区。	<p>如果 I/O 率 (写入) 很高, 则忙等待可能因高速缓存抖动导致。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>向主高速缓存区或临时高速缓存区分配更多的内存, 以 BWaits 始终 > 0 的为准。</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。几乎相同的查询包括任何接触数据和使用缓存区的查询, 如 INSERT、UPDATE、DELETE 和 SELECT。</p>
LRU 等待 (调试报告中的 LRUNum TimeOuts 百分比)	20% 或更少	> 20%, 表示发生严重争用问题。	检查操作系统补丁级别和其它环境设置。此问题常常是操作系统问题。
IOWait (IONum-Waits)	10% 或更低	> 10%	检查是否存在磁盘错误或 I/O 重试
FLWait (FLMutexWaits)	20% 或更低	> 20%	<p>检查 dbspace 配置:</p> <p>数据库空间即将不足?</p> <p>DISK_STRIPING 是否为 ON?</p> <p>sp_iqcheckdb 报告碎片是否大于 15%?</p>
HTWait (BmapHTNum-Waits) MemWts (MemNtimesWaited) (PFMgrCondVar-Waits)	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, 并存在足够使系统繁忙的作业, 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, 确定是否存在高速缓存抖动。还应查看 cache_by_type (或调试) 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? 系统是否有许多 CPU? 可能需要诸如 Multiplex 配置的策略。
InUse% (使用的缓冲区)	除了启动期间以外等于或接近 100%	约小于 100%	缓冲区高速缓存可能过大。 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%, 表示系统正在危险地接近缓冲区不足条件, 这将会导致事务回退	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。
可用线程数 (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, 可能会导致系统线程匮乏。	尝试执行下列操作之一: 通过设置 -iqmt 增加线程数。 减少与线程相关的选项: MAX_IQ_THREADS_PER_CONNECTION, MAX_IQ_THREADS_PER_TEAM . 通过设置 USER_RESOURCE_RESERVATION 限制查询引擎资源分配。 通过设置 -iqgovern 限制作业数。
FlOutOfSpace (仅调试)	0, 表示此存储区的空闲列表不完整; 未分配的页可用	1, 表示已完全分配此存储 (主存储区或临时存储区)	将更多 dbspace 添加到该存储区

注意： 如果一个高速缓存执行的 I/O 明显多于另一高速缓存，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。重新分配之后重新运行负载，并监控性能变化。

其它信息

《参考：语句和选项》 > “SQL 语句” > “IQ UTILITIES 语句”

另请参见

- 数据库分析过程（第 27 页）
- 事件分析过程（第 28 页）
- 关键性能指标（第 29 页）

Multiplex 性能

调整系统以获得最佳性能或更好地利用磁盘空间。

Multiplex 中的每个服务器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。

管理 Multiplex 磁盘空间

令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。指定 `auto_commit` 选项有助于减少版本积累进而减少占用的空间。

如果任何服务器上有任何用户可能处在可能需要旧版本的事务中，**SAP Sybase IQ** 则不能删除这些旧版表。因此在 **Multiplex** 数据库中同时发生表更新和查询时，**SAP Sybase IQ** 可能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。

管理逻辑服务器资源

通过逻辑服务器，您可以最有效地管理 **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。

平衡查询负载

SAP Sybase IQ 可通过分布式查询处理提供部分负载平衡，或可通过网络客户端提供全部负载平衡（需要第三方软件）。

分布式查询处理

DQP（分布式查询处理）将针对 **Multiplex** 服务器上符合条件的查询自动进行。

Multiplex 性能

Multiplex 服务器必须已建立 MIPC (**Multiplex** 进程间通信连接)。当前连接的逻辑服务器必须至少拥有另一个可用的成员节点。共享临时 **dbspace** 必须拥有可用的可写文件。

通过网络客户端平衡负载

要使用网络客户端在 **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。

要使用此方法，需要使用此中间负载平衡系统的 IP 地址和端口号、常规服务器名并将 **VerifyServerName** 连接参数设置为 **NO**，以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，负载平衡器将与它认为负载最低的计算机建立连接。

注意： 需要使用第三方软件。**VerifyServerName** 仅允许此方法工作。

其它信息

«管理：数据库» > “附录：连接和通信参数参考” > “网络通信参数” > “**VerifyServerName** 通信参数 [Verify]”

«管理：**Multiplex**» > “分布式查询处理”

模式设计

良好的数据库性能始于良好的数据库设计。开发时花些时间将设计功能整合到您的模式中可减少响应时间并加快查询处理速度。

索引

SAP Sybase IQ 索引选择和解决方案。

另请参见

- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用来实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

索引提示

选择正确的列索引类型，以加快查询运行速度。

SAP Sybase IQ 自动提供某些索引（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才能尽快处理特定查询。

INDEX_ADVISOR

INDEX_ADVISOR 可以在查询中生成关于优化器何时会从一个或多个列的其它索引中受益的消息。

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。

LF 或 HG 索引

如果列未使用枚举 FP 存储，需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。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。

其它信息

《参考：语句和选项》> “数据库选项”> “按字母顺序排列的选项列表”> “INDEX_ADVISOR 选项”。

另请参见

- 使用索引的时间和位置（第 38 页）
- 简单索引选择标准（第 39 页）
- HG 索引装载（第 40 页）
- 多列索引（第 41 页）

使用索引的时间和位置

索引是 SAP Sybase IQ 内的主要调优机制。获悉应在何时何处使用索引可以加快查询运行速度。

在以下列中 始终使用索引：

- 连接列（不考虑基数的 HG 索引）
- 可搜索列（基于基数的 HG 或 LF 索引）
- DATE、TIME 和 DATETIME/TIMESTAMP 列（DATE、TIME 和 DTTM）
DATE、TIME 或 DATETIME/TIMESTAMP 列还应拥有一个基于数据基数的 LF 或 HG 索引。
- 如果并不确定此列是否将频繁使用，则在此列放置一个 LF 或 HG 索引。随后可启动负载管理监控索引的使用情况。

- 适当情况下使用 PRIMARY KEY、UNIQUE CONSTRAINT 或 UNIQUE HG 索引，因为它们可为 SAP Sybase IQ 提供有关索引列中唯一数据的附加信息。
- 具有 HNG 或 CMP 索引的列应具有相应的 LF 或 HG 索引
- 对于仅向客户端（投影）返回数据的列，无需使用索引

另请参见

- 索引提示（第 37 页）
- 简单索引选择标准（第 39 页）
- HG 索引装载（第 40 页）
- 多列索引（第 41 页）

简单索引选择标准

一些简单问题的解答，可帮助您为列选择正确的索引。

要在不考虑查询的情况下为数据模型确定最佳索引，请针对每一列向自己提问以下简单问题：

- 基数是否大于 1500-2000？
如果答案是肯定的，则在此列上放置 HG 索引。如果否定，则在此列上放置 LF 索引。
- 此列是否包含 DATE、TIME、DATETIME 或 TIMESTAMP 数据？
如果答案是肯定的，则在此列上放置 DATE、TIME 或 DTTM 索引。还应在此列上放置一个 LF 或 HG 索引。
- 该列是否将用于范围搜索或集合？
如果答案是肯定的，则在此列上放置 HNG 索引。还应放置一个 LF 或 HG 索引。如果集合不仅仅包含列，则 HNG 可能不合适。大多数情况下并不需要 HNG 索引，因为 LF 或 HG 足以执行集合。但这不适用于 DATE、TIME 或 DATETIME 类型。
- 此列是否用于单词搜索？
如果答案是肯定的，则在此列上放置 WD 索引。不需要 LF 或 HG 索引，它们会占用大量的空间。
- 此列是否用于全文本搜索？
如果答案是肯定的，则在此列上放置 TEXT 索引。不需要 LF 或 HG 索引，它们会占用大量的空间。
- 是否会比较相同表中的两列 ($A = B$, $A < B$, $A > B$, $A \leq B$, $A \geq B$)？
如果答案是肯定的，则在这两列上放置 CMP 索引。
- 此列或此组列是否会用于 GROUP BY 或 ORDER BY 语句？
如果答案是肯定的，则在 GROUP BY 或 ORDER BY 语句的此列或此组列上放置 HG 索引。每一列还应具有一个相应的 HG 或 LF 索引。
- 此列是否是多列主键、约束或索引的一部分？
如果答案是肯定的，则在多列索引的每一列上放置一个 HG 或 LF 索引。

另请参见

- 索引提示 (第 37 页)
- 使用索引的时间和位置 (第 38 页)
- HG 索引装载 (第 40 页)
- 多列索引 (第 41 页)

HG 索引装载

相对于其它索引，在数据装载和删除期间，HG 索引需要更多维护成本。HG 索引的性能很大程度上取决于 HG 索引结构内数据的位置：稀疏操作或密集操作。

在密集 HG 操作中，受影响的行会围绕特定键紧密组合。在稀疏操作中，每个键可能只有少量行一定会受到影响。例如，数据日期通常围绕操作记录时间、数据修改时间等组合。这表示新数据将会放置在 HG 索引结构的末尾。删除日期 HG 索引中的数据时，要删除的数据通常出现在若干天、周、月等时间段中，因此需要从 HG btree 的开头进行删除或围绕若干键紧密组合以进行删除。相对而言，这些操作运行速度非常快，因为 SAP Sybase IQ 针对少量页面进行操作但是影响大量行。

如价格、客户 ID、城市和国家/地区等的非常稀疏的数据则大不相同。例如，“定价”数据装载时，索引中已存在的所有数据的每个值都会有很大差异。如果列跟踪存储为数字字段的库存价格，则数据将进行紧密更新，因为要更改的数据的范围将涉及几乎所有已装载的值。这些操作相对要慢些，因为索引页的数量较多且必须针对每一个受影响的行进行维护。在最坏的情况下，SAP Sybase IQ 强制为将装载或删除的 EACH ROW 读取或写入 1 页。尽管这并非最佳选择，但 SAP Sybase IQ 专用于并行处理 HG 索引装载和删除的阶段 2，因此大大减小了影响。

这都很好，但对数据模型设计和索引的有何影响？SAP Sybase IQ 中典型的调优和优化通常归结于索引或由此产生的缺失。在决定要放置和拿开哪些索引时，了解数据和加载对索引有何影响非常重要。因为与其它索引相比较，HG 索引装载所花费的时间更长，在进行使用和设计时它们往往是关注的重点。当然，HG 索引可帮助提高查询性能。有时，尽管添加索引可能会对查询产生些微积极影响，但对数据装载的负面影响更大。在这些情况下，了解装载或删除费时的原因以及如何应对很重要。

对于当前已装载的数据，新数据是稀疏还是密集对此至关重要。如果是对客户 ID 相对随意的列，则必须对列建立索引以进行更为快速的查询，并且索引必须位于此列上。假设表上存在主键且它是客户 ID 和存储交易日期时间的 "Date" 字段。若顺序为 (customer_id, transaction_date)，则大多数情况下数据会从表中稀疏装载或删除。数据按交易日期进行装载。由于客户 ID 列处于多列索引的首位，因此 SAP Sybase IQ 会接触到整个 HG 索引结构中的数据。

对 (transaction_date, customer_id) 顺序的简单更改会改变此行为。索引仍然可控制主键的参照完整性。列的顺序对主键执行没有影响。因此，可更改列的顺序而不会引起任何后续不良影响。此简单更改现将使得所有按交易日期装载的新数据以非常密集的方式插入到 HG 索引结构的末尾。随着时间推移，数据将始终装载到 HG 结构的末尾。

简单的更改多列索引中列顺序可能会对性能产生巨大的影响。HG 索引的大小不应有较大更改，因为不考虑顺序的情况下，数据的宽度相同。将更改的是数据从表中装载或删除的速度。

另请参见

- 索引提示 (第 37 页)
- 使用索引的时间和位置 (第 38 页)
- 简单索引选择标准 (第 39 页)
- 多列索引 (第 41 页)

多列索引

当前，只有 HG、UNIQUE HG、UNIQUE CONSTRAINT 和 PRIMARY KEY 索引支持多列索引创建，但多列索引对 GROUP BY 和 ORDER BY 语句也颇为有用。

从统计角度来看，多列索引可提供多列表连接的大量信息，以供优化程序获取连接的精确统计信息并判断连接为多对多还是一对多。优化程序同时非常智能，足以利用统计信息进行优化，但在实际工作中使用各个 HG/LF 索引。优化程序可估算所有连接的成本并对方案进行排序，最后决定最适合于操作的索引。统计信息有助于优化程序完成以上操作。

HG 索引须知：

- HG 插入成本最高
- 尽量确保插入发生在索引末尾

请在索引列表的开头放置交易日期或批次编号（序列数据）等一般增量数据。这样可确保顺序键。

另请参见

- 索引提示 (第 37 页)
- 使用索引的时间和位置 (第 38 页)
- 简单索引选择标准 (第 39 页)
- HG 索引装载 (第 40 页)

连接列

针对连接，请尽可能减少数据类型以减少磁盘 I/O 和内存需求。

因为整数比较比字符比较速度更快，所以在连接中使用（如有可能，无符号的）整数数据类型。尽可能减少数据类型可减少磁盘 I/O 和内存需求从而提高连接性能。因为 HG 索引从连接角度来看具有更多功能，所以在连接列上使用 HG 索引而非相应的基数索引（LF 或 HG）。应该权衡 LF 索引和 HG 索引在提高装载时间潜力方面的比较结果。

另请参见

- 索引 (第 37 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

主键

多列主键应在主键指定的每列添加 LF 或 HG 索引。必须手动执行此操作，因为 SAP Sybase IQ 只在复合列上创建 HG 索引。

UNIQUE constraint、UNIQUE HG 和主键共享相同的结构。此结构使用无 G-Array 的 HG 索引来存储行 id。请尽量在表上使用主键。优化程序可借此制定更加明智的查询路径决策（即使未使用索引）。索引结构可提供详细的统计信息，优化程序可借此作出更明智的选择并提供遍历数据的结构。

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

外键

与主键一样，使用外键可以提高查询连接性能。**SAP Sybase IQ** 还可借此获取表的连接方法以及连接统计信息的相关信息。**SAP Sybase IQ** 可自动在外键列上创建 **HG** 索引，所以不需要再添加 **HG** 或 **LF** 索引。外键要求引用的表中存在主键。

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

适当确定数据类型的大小

尽可能准确地确定所有数据类型（特别是基于字符的数据类型）的大小。

要确定对某列使用哪种数据类型，请考虑以下因素：

- **SAP Sybase IQ** 包含大量数据类型。针对应用程序使用适当的数据类型可以获得最佳性能。
- 如果不需要 **HOUR**、**MINUTE** 和 **SECOND** 信息，请使用 **DATE** 替代 **DATETIME**。
- 如果 **TINYINT** 或 **SMALLINT** 数据类型的数据适用，则使用上述类型替代 **INTEGER** 或 **BIGINT**。
- 定义 **NUMERIC()** 或 **DECIMAL()** 时不要分配过多存储空间，因为对于无需此精度级别的数据而言开销过大。
- **CHAR()** 和 **VARCHAR()** 类型在缺省 **Flat FP** 索引中宽度固定。唯一的差异就是表示所用字节数的每个 **VARCHAR()** 行多出 1 个字节。

SAP Sybase IQ 包含可压缩 **BINARY()**、**CHAR()**、**VARCHAR()** 以及 **VARBINARY()** 数据类型中常见的大型重复模式的压缩算法。

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

空值

将列定义为 **NULL** 或 **NOT NULL** 有助于优化程序更有效地工作。

通过指定 **NULL** 或 **NOT NULL**，优化程序可获取更多有关数据特性的信息，从而获得更有根据性的连接和搜索条件推测。同在其它数据库中一样，**NULL** 数据不会节省数据库页的空间。但是，由于 **SAP Sybase IQ** 压缩算法和优化索引，存储在磁盘上的 **NULL** 数据将被压缩。

- 始终指定 **NULL** 或 **NOT NULL**
- **Open Client** 和 **ODBC** 连接在创建表时拥有不同的缺省行为
- 为优化程序提供有关连接和搜索参数数据特征的更多信息

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)

- 散列分区 (第 51 页)

无符号数据类型

在某些情况下，使用无符号数据类型可避免符号比较，从而创建更快的查询。

在所有数据始终大于或等于零，而数据符号不太重要的情况下，请使用无符号数据类型。没有符号的存储在进行列比较时，将不必执行符号比较。这样可以提高性能，并在连接和搜索数据（特别是键列的数据）时减少一个步骤。

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用来实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

LONG VARCHAR 和 LONG VARBINARY

在不使用大对象存储机制的情况下，利用 `VARCHAR()` 和 `VARBINARY()` 增加列存储空间。

通常，研发人员和 **DBA** 认为 `VARBINARY()` 和 `VARCHAR()` 数据限制为 255 个字节。**SAP Sybase IQ** 支持字节宽度高达 32K 的 `VARCHAR()` 和 `VARBINARY()`（也称作 `LONG VARCHAR` 或 `LONG VARBINARY`）。这样便可支持占据更多存储空间文本或二进制数据，而无需将其移动到专门针对 `BLOB/CLOB` 或 `IMAGE/TEXT` 数据类型的大对象存储机制中。

- 可用于存储中等数量的文本或二进制数据
- 最大宽度为 32K（针对 `VARBINARY()`，则为 64K ASCII 十六进制数）
- `WORD` 和 `TEXT` 索引是 `VARCHAR()` 数据上唯一允许的宽度大于 255 字节的索引。
- 存储将按 256 字节块进行分配
- 257 字节的字符串将需要 512 字节的存储空间

- 511 字节的字符串也将需要 512 字节的存储空间

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非规范化以提高性能 (第 48 页)
- 可用来实现更快装载的 UNION ALL 视图 (第 49 页)
- 散列分区 (第 51 页)

大对象存储

对于需要 32 K 以上存储的数据，请使用大对象数据类型。

- 大对象数据类型存储 ASCII (TEXT/CLOB) 以及二进制 (IMAGE/BLOB) 数据。数据的每个 BLOB/CLOB 单元存储在一个或多个页面中：
 - 假设页大小为 128K
 - 如果数据为 129K，则需要 2 页来存储信息
 - 如果数据为 1K，则需要 1 页来存储数据
 - 在任一情况下，磁盘中都会将页压缩成块大小的倍数
- 可用于存储二进制或基于文本的对象
- 将长二进制数据类型的最大大小从 6K 扩展至无限大小
- TEXT 索引是唯一可行的索引
- 可通过 TEXT 索引及其搜索功能进行彻底地搜索
- 拥有可返回对象大小的特殊函数 (byte_length64)
- 拥有可返回对象的部分 (但不是全部内容) 的特殊函数 (byte_substr64)
- 可使用 BFILE () 函数将二进制对象单元的内容提取到单个文件

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)

- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

临时表

如果希望在事务提交过程中保留数据，需在创建全局临时表或声明局部临时表时使用 **ON COMMIT PRESERVE ROWS** 选项。

有三类临时表：

- # 表

```
CREATE TABLE #temp_table( coll int )
```

- 本地临时表

```
DECLARE LOCAL TEMPORARY TABLE temp_table ( coll int )
```

局部临时表与 # 表相似

- 全局临时表

```
CREATE GLOBAL TEMPORARY TABLE temp_table ( coll int )
```

全局临时表结构在连接和重新启动过程中为静态

表不需要 **ON COMMIT PRESERVE ROWS** 选项，因为 # 表中的数据在事务提交过程中始终保留。

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 非范式化以提高性能 (第 48 页)
- 可用于实现更快装载的 **UNION ALL** 视图 (第 49 页)

- 散列分区 (第 51 页)

非范式化以提高性能

非范式化数据库可以提高性能，但存在一些风险和缺点。

非范式化只能在您全面了解应用程序时才能成功执行，并且仅当性能问题显示需要这样做时，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（它们经常需要汇总大量数据）与事务处理需求（它们执行不连续的数据修改）的区分的典型示例。非范式化通常以其它处理的开销来支持某些处理。

非范式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。

决定非范式化

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？
- 并发查询有多少？
- 最常访问的表有多大？
- 任何进程都计算汇总吗？

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- **LONG VARCHAR** 和 **LONG VARBINARY** (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 用来实现更快装载的 **UNION ALL** 视图 (第 49 页)
- 散列分区 (第 51 页)

可用来实现更快装载的 UNION ALL 视图

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。

SAP Sybase IQ 允许您将数据拆分到多个单独的基表中（例如，按日期）。数据将装载到这些较小的表中。然后，通过 UNION ALL 视图将这些表重新连接在一起，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。

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，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
```

可以每月将数据装载到单个基表中（在此示例中为 JANUARY、FEBRUARY 或 MARCH）。在下一月再将数据装载到有相同列和索引类型的新表中。

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。

另请参见

- 索引（第 37 页）
- 连接列（第 41 页）
- 主键（第 42 页）
- 外键（第 43 页）
- 适当确定数据类型的大小（第 43 页）
- 空值（第 44 页）
- 无符号数据类型（第 45 页）
- LONG VARCHAR 和 LONG VARBINARY（第 45 页）
- 大对象存储（第 46 页）
- 临时表（第 47 页）

- 非范式化以提高性能（第 48 页）
- 散列分区（第 51 页）

引用 **UNION ALL** 视图的查询

要调整引用 **UNION ALL** 视图的查询的性能，需设置 `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优化工作定义的完整索引。如果查询使用 `DISTINCT`，则其使用 **UNION ALL** 视图的运行速度将比使用基表慢。

SAP Sybase IQ 包含 **UNION ALL** 视图的优化技术，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：

- 它只包含一个或包含多个 **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。
- 第一个 **UNION** 分支的 `SELECT` 列表中的列数据类型序列与 **UNION** 每个随后分支中的序列相同。

另请参见

- **UNION ALL** 视图性能（第 50 页）

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

- 引用 **UNION ALL** 视图的查询（第 50 页）

散列分区

散列表分区将数据分配到可并行执行的逻辑分区中，可以增强大表和分布式查询 (PlexQ) 的连接性能。

IQ 提供了利用散列表分区的新的连接算法和聚合算法，这些算法不仅可以减少运算过程的中间结果和网络传输的数据量的大小，并且，通过对表分而治之，提高了查询处理的并行度。通过数据感知和工作负载的感知提高高速缓存行为可在 PlexQ 环境中提供进一步的可伸缩性。

使用散列表分区连接或散列表分区集合算法

要使用散列表分区连接或散列表分区集合算法，针对表的所有散列表分区键列都要用于这些表的等连接条件或分组表达式中，这一点至关重要。也可使用其它连接条件或分组表达式，但若作为散列表分区键一部分的列未用于查询，则分区算法将不符合条件。在此情况下，其它非分区算法仍然符合条件，可选择用于非分区表。一组表的散列表分区应该包含用于对这些表进行应用程序查询的最小的一组公用列。通常，单个列即足以用于分区。分区表间连接中列的数据类型必须一致。

大表相对小表可从散列表分区获取更多好处。请考虑对至少有 10 亿行的表进行散列表分区。

大内存

某些装载操作可能需要比缺省提供的 2GB 更大的内存。如果内存需求超过缺省值，请使用 `-iqlm` 启动选项来增加 SAP Sybase IQ 能够从操作系统动态申请的内存。

一般情况下，大内存需求是分配到 SAP Sybase IQ 的总可用物理内存的三分之一。为确保有足够的内存可供主 IQ 存储和临时 IQ 存储使用，应设置 `-iqlm`、`-iqtc` 和 `-iqmc` 启动参数，这样每个参数都会接收到分配给 SAP Sybase IQ 的总可用物理内存的三分之一。

在大多数情况下，应将总物理内存的 80% 分配到 SAP Sybase IQ，以避免换出 SAP Sybase IQ 进程。调整实际内存分配，以容纳在同一系统中运行的其它进程。例如，在拥有 32 个内核和 128GB 总可用物理内存的计算机上，应将 100GB（大约是总计 128GB 的 80%）分配给 SAP Sybase IQ 进程。根据一般规则，要将 `-iqlm`、`-iqtc` 和 `-iqmc` 参数分别设置为 33GB。

其它信息

- 《参考：语句和选项》 > “数据库选项” > “按字母顺序排列的选项列表” > “JOIN_PREFERENCE 选项”
- 《参考：语句和选项》 > “数据库选项” > “按字母顺序排列的选项列表” > “AGGREGATION_PREFERENCE 选项”。
- 《实用程序指南》 > “start_iq 数据库服务器启动实用程序” > “start_iq 服务器选项” > “-iqlm iqsrv16 服务器选项”

- 《实用程序指南》 > “start_iq 数据库服务器启动实用程序” > “start_iq 服务器选项” > “-iqmc iqsrv16 服务器选项”
- 《实用程序指南》 > “start_iq 数据库服务器启动实用程序” > “start_iq 服务器选项” > “-iqtc iqsrv16 服务器选项”

另请参见

- 索引 (第 37 页)
- 连接列 (第 41 页)
- 主键 (第 42 页)
- 外键 (第 43 页)
- 适当确定数据类型的大小 (第 43 页)
- 空值 (第 44 页)
- 无符号数据类型 (第 45 页)
- LONG VARCHAR 和 LONG VARBINARY (第 45 页)
- 大对象存储 (第 46 页)
- 临时表 (第 47 页)
- 非范式化以提高性能 (第 48 页)
- 可用来实现更快装载的 UNION ALL 视图 (第 49 页)

故障排除

标识和解决常见性能问题

隔离性能问题

确定问题对于 SAP Sybase IQ 而言是外部问题还是内部问题。

SAP Sybase IQ 外部

- 针对任何瓶颈或问题监控操作系统、硬件和存储
- 查找高 CPU 使用、高 CPU 系统时间、低 CPU 用户时间和高等待时间
- 查找高于 10ms 的 I/O 服务时间

SAP Sybase IQ 内部

- 启用 INDEX_ADVISOR 并查找缺失索引
- 若单个查询出现问题，则启用 QUERY_PLAN_AS_HTML 和 INDEX_ADVISOR

另请参见

- 诊断工具 (第 53 页)
- 常见性能问题 (第 54 页)

诊断工具

操作系统上用于监控系统活动的实用程序。

使用这些实用程序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。

OS	实用程序	描述
UNIX	top, topas	实时地持续查看处理器活动。 top 适用于 Solaris、Linux 和 HP-UX 平台。 topas 适用于 AIX 平台。
	ps	报告进程状态。
	vmstat	显示有关系统进程、内存、分页、块 IQ、捕获和 CPU 活动的信息。

OS	实用程序	描述
	iostat -x	显示磁盘子系统信息。
	sar	将所选的 OS 活动结果写入标准输出。
Windows	“任务管理器”，“资源监视器”	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。

注意：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 dbspace 设备所在位置的磁盘上。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

- 隔离性能问题（第 53 页）
- 常见性能问题（第 54 页）

常见性能问题

常见性能问题的解决方案。

另请参见

- 隔离性能问题（第 53 页）
- 诊断工具（第 53 页）

分页和磁盘交换

良好的内存管理可避免发生页面交换。要最大程度减少对操作系统文件的使用，需增加或重新分配物理内存。

内存不足导致性能大幅降低。如果发生这种情况，您需要寻找某种方式获得更多可用内存。分配给 SAP Sybase IQ 的内存越多越好。

因为系统的内存量始终存在固定限制，所以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。操作系统必须转向磁盘并检索任何数据才能满足内存请求，这时将会发生分页或交换。良好的内存管理可避免或最大程度减少发生分页和交换。

最常使用的操作系统文件是交换文件。内存用完时，操作系统会将内存页交换到磁盘，以便为新数据腾出空间。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。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。

为了最大程度地利用物理内存，SAP Sybase IQ 针对所有数据库读写操作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多的空间，才能容纳所有物理内存。必须使交换/分页空间条带化分布在快速磁盘上。

另请参见

- 索引和行碎片（第 55 页）
- 目录文件增大（第 55 页）
- 抖动和查询执行（第 56 页）

索引和行碎片

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 FP 索引级别的行碎片。
- **sp_iqindexfragmentation** 报告补充索引中的内部碎片。 .

查看输出并决定是重新创建索引、重组索引还是要重新构建索引。可以创建其它索引来补充 FP 索引。

其它信息

- 《参考：构件块、表和过程》 > “系统过程” > “sp_iqrowdensity 过程”。
- 《参考：构件块、表和过程》 > “系统过程” > “sp_iqindexfragmentation 过程”。

另请参见

- 分页和磁盘交换（第 54 页）
- 目录文件增大（第 55 页）
- 抖动和查询执行（第 56 页）

目录文件增大

目录文件增大是正常情况，根据应用程序和目录内容不同而各不相同。 .db 文件的大小不影响性能，并且 .db 文件中的可用页将根据需要重用。

为最大限度地抑制目录文件增大：

- 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。
- 在长时间运行事务之后发出 COMMIT 语句。

另请参见

- 分页和磁盘交换 (第 54 页)
- 索引和行碎片 (第 55 页)
- 抖动和查询执行 (第 56 页)

抖动和查询执行

调整 `HASH_THRASHING_PERCENT` 选项以限制涉及散列算法的查询执行期间的抖动。

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 **SAP 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。**SAP 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，为执行查询提供所需的资源：

- 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。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同样大小以防止系统发生抖动。
- 针对此语句，尝试标识 **SAP Sybase IQ** 未能估计一个或多个散列大小的原因并缓解后果。例如，请检查所有需要 LF 或 HG 索引的列都至少拥有其中一种索引。并考虑多列索引是否合适。
- 减少数据库选项 `HASH_PINNABLE_CACHE_PERCENT` 的值。

为标识查询中可能存在的问题，请使用临时数据库选项 `QUERY_PLAN='ON'` 和 `QUERY_DETAIL='ON'` 运行查询，以生成查询计划。检查查询计划中的估计值。

另请参见

- 分页和磁盘交换 (第 54 页)
- 索引和行碎片 (第 55 页)
- 目录文件增大 (第 55 页)

查询和删除

帮助您计划、结构化和控制查询的建议。

结构化查询

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。
- 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。

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。

可以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。此更改对于用户来说是透明的，但改进了查询性能。

使用多列 HG 索引可以提高在 ORDER BY 子句中对多列进行查询的运行速度。例如，如果用户在表 T 上具有多列索引 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 58 页）
- 使用高速缓存方法（第 58 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。**SAP Sybase IQ** 会展平多个子查询但不是所有子查询。当优化程序选择使用此优化方式时，需使用 `SUBQUERY_FLATTENING_PREFERENCE` 和 `SUBQUERY_FLATTENING_PERCENT` 进行控制。

另请参见

- 增强 `ORDER BY` 查询性能 (第 57 页)
- 使用高速缓存方法 (第 58 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，每次引用列发生更改时相关子查询都会重新执行。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

- 增强 `ORDER BY` 查询性能 (第 57 页)
- 提高了子查询性能 (第 58 页)

生成查询计划

生成查询计划可帮助您了解优化程序开发的执行计划。

在执行任何查询之前，查询优化程序都会创建查询执行计划。查询执行计划是数据库服务器用来访问数据库中与语句相关信息的步骤集。无论语句的执行计划是否刚经过优化、是否跳过了优化程序，也无论其计划是否是从先前执行进行高速缓存，均可进行保存和查看。尽管查询执行计划可能不会与初始语句中所使用的语法完全对应，但初始查询与执行计划中所描述的操作在语义上是等效的。

查询计划生成一个包含一系列表示处理阶段的节点的执行树。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。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。

装载执行计划

装载执行计划详细描述数据库引擎将数据插入到表的步骤。装载计划使用与查询执行计划相同的数据库选项和输出选项。数据流对象 (DFO) 树标识装载各阶段处理的行数。不同的 `SQL` 语句会生成不同的 `DFO` 树，相同的 `SQL` 语句针对不同种类的表（如未分区表、范围分区表、散列分区表以及散列范围分区表等）生成不同的树。

生成查询计划

要生成查询计划，请设置相应评估选项并执行查询。计划的文本版本将写入 `.iqmsg` 文件。HTML 版本可在 **“Interactive SQL 计划查看器”** 或大多数 Web 浏览器中显示。

注意： 仅使用查询计划来评估特定查询或装载的效率。将 `QUERY_PLAN` 选项设置为 `ON` 来运行 SAP Sybase IQ 对性能有重大影响，尤其是 `INSERT...VALUE` 语句量增加时。

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，索引顾问程序会输出索引建议以作为 Sybase IQ 查询计划的一部分；如果未启用查询计划，则将其作为单独的消息输出在 Sybase IQ 消息日志文件中。这些消息以字符串 **“Index Advisor:”** 开头，并且您可以使用该字符串在 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，缺省设置为 `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`（缺省设置）时，Sybase IQ 会生成有关查询的消息。
- `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。这将允许该计划包括其它信息，例如从查询的每个节点传递过来的实际行数。为使此选项有效，`QUERY_PLAN` 必须为 `ON`。缺省情况下，此选项为 `OFF`。

- `QUERY_PLAN_AS_HTML` - 生成 **HTML** 格式的图形式查询计划，以便在 **Web** 浏览器中进行查看。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` 选项将查询名称包含在查询计划的文件名中。此选项缺省设置为 `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。
- `QUERY_PLAN_MIN_TIME` - 指定查询执行的阈值。只有查询执行时间超出阈值（以微秒为单位）时，才会生成查询计划。这样便可通过不为查询执行时间较短的查询生成计划来提高系统性能。

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` 也为 `ON` 时，最好启用消息日志包装或消息日志存档，以避免填满消息日志文件。

另请参见

- 使用查询计划（第 60 页）

使用查询计划

设置 `QUERY_PLAN_AS_HTML` 选项以生成 **HTML** 版本的查询计划，以便在 **Web** 浏览器中进行查看。

HTML 版本可以在 **Interactive SQL “计划查看器”** 中显示。在 **HTML** 查询计划中，树中的每个节点均超链接至处理的详细信息。您可以单击任意节点快速浏览整个计划。

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。

在 **Interactive SQL “计划查看器”** 中查看图形式查询计划。“**计划查看器**”不支持文本计划，将返回错误消息不支持此计划类型。使用 **SQL** 函数 `GRAPHICAL_PLAN` 和 `HTML_PLAN` 返回查询计划作为字符串结果。

其它信息

- 《参考：构件块、表和过程》 > “SQL 函数” > “GRAPHICAL_PLAN 函数 [字符串]”
- 《参考：构件块、表和过程》 > “SQL 函数” > “HTML_PLAN 函数 [字符串]”
- 《参考：语句和选项》 > “数据库选项” > “QUERY_PLAN_TEXT_ACCESS 选项”
- 《参考：语句和选项》 > “数据库选项” > “QUERY_PLAN_TEXT_CACHING 选项”。

另请参见

- 查询评估选项（第 59 页）

控制查询处理

任何用户都可以对用于处理特定查询的时间量设置限制。如果用户具有 SET ANY PUBLIC OPTION 系统特权，则其查询优先级可能高于其他用户的查询，或者他们可以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 选项可以限制查询可运行的时间。如果查询的执行时间超过 MAX_QUERY_TIME，SAP Sybase IQ 会停止查询并显示相应的错误。

注意： SAP 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 61 页）
- 设置查询优化选项（第 62 页）
- 设置用户提供的条件提示（第 63 页）
- 监控负载（第 63 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。

查询和删除

- `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` - 允许具备 `SET ANY PUBLIC OPTION` 系统特权的用户为用户或角色设置 `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。

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 61 页）
- 设置查询优化选项（第 62 页）
- 设置用户提供的条件提示（第 63 页）
- 监控负载（第 63 页）

设置查询优化选项

优化选项影响查询处理速度。

- `AGGREGATION_PREFERENCE` - 控制用以处理集合（`GROUP BY`、`DISTINCT` 和 `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，并且匹配字符串并非以一组常量字符开头后跟单个通配符，则优化程序会使用此选项。
- `DEFAULT_LIKE_RANGE_SELECTIVITY_PPM` - 设置前导常量 `LIKE` 谓词的缺省选择性，形式为 `LIKE 'string%'`，其中匹配字符串是一组常量字符后跟一个通配符（`%`）。如果未提供其它选择性信息，优化程序会使用此选项。
- `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 61 页）
- 设置查询优先级（第 61 页）

- 设置用户提供的条件提示 (第 63 页)
- 监控负载 (第 63 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。

SAP 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因素是条件的选择；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。

其它信息

- 《参考：构件块、表和过程》 > “SQL 语言元素” > “用户提供的条件提示”

另请参见

- 设置查询时间限制 (第 61 页)
- 设置查询优先级 (第 61 页)
- 设置查询优化选项 (第 62 页)
- 监控负载 (第 63 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。

SAP Sybase IQ 提供用于收集和分析已定义负载统计信息的工具。DBA 可快速确定查询正在引用因而应该保留的数据库对象。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。

另请参见

- 设置查询时间限制 (第 61 页)
- 设置查询优先级 (第 61 页)
- 设置查询优化选项 (第 62 页)
- 设置用户提供的条件提示 (第 63 页)

优化删除操作

SAP Sybase IQ 选择最佳算法来处理含 HG 和 WD 索引列的删除操作。

HG 删除操作

SAP Sybase IQ 在三种算法中选择任意一种来处理含 HG (**High_Group**) 索引列的删除操作。

- 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 删除的开销

删除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。

在具有 HG、LF 或 enumerated FP 索引的列上指定谓词会大大改善开销。为使 HG 开销选择大型删除以外的算法，必须能够确定受删除操作影响的不同值 (组) 的数量。非重复计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、更精确的非重复计数估计值。

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 删除性能选项

您可以使用 HG_DELETE_METHOD 选项来控制 HG 删除的性能。

如果通过 HG_DELETE_METHOD 选项指定参数值，则必须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》的“数据库选项”中的“`HG_DELETE_METHOD` 选项”。

另请参见

- `WD` 删除操作 (第 65 页)
- `TEXT` 删除操作 (第 66 页)

WD 删除操作

SAP Sybase IQ 在三种算法中选择任意一种来处理含 `WD` (Word) 索引列的删除操作。

- 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

如果通过 `WD_DELETE_METHOD` 选项指定参数值，则必须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 64 页）
- `TEXT` 删除操作（第 66 页）

TEXT 删除操作

SAP Sybase IQ 在两种算法中选择任意一种来处理含 `TEXT` 索引列的删除操作。

- 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

如果通过 `TEXT_DELETE_METHOD` 选项指定参数值，则必须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

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 64 页）
- `WD` 删除操作（第 65 页）

索引

符号

-gm 14
 -iqgovern
 限制查询以提高性能 15
 -iqwmem 12

A

AGGREGATION_ALGORITHM_PREFERENCE
 62

B

BT_PREFETCH_MAX_MISS 19
 绑定内存
 -iqwmem 开关 12

表

连接 37
 折叠 37

C

CACHE_PARTITIONS 22
 CMP 索引 38, 39
 CPU

可用性 5
 设置数目 5

查询

HG 删除操作 64
 ORDER BY 性能 57
 TEXT 删除操作 66
 WD 删除操作 65
 按行限制 17
 查询处理 61
 查询计划 60
 查询优先级选项 61
 负载监控 63
 高速缓存方法 58
 控制 62
 连接 62
 删除操作 64
 设置时间限制 61
 条件提示 63
 限制并发 15

优化 37, 62
 优化程序简化 62

查询,
 结构化 57
 优化 57

查询处理
 监控 63
 控制 61

查询服务器
 平衡负载 35

查询计划
 查询评估计划 58
 查询评估选项 59
 生成 58
 生成但不执行 59
 使用 60
 装载评估计划 58

查询计划, 选项

INDEX_ADVISOR 59
 NOEXEC 59
 QUERY_DETAIL 59
 QUERY_PLAN 59
 QUERY_PLAN_AFTER_RUN 59
 QUERY_PLAN_AS_HTML 59
 QUERY_PLAN_AS_HTML_DIRECTORY
 59
 QUERY_PLAN_TEXT_ACCESS 59
 QUERY_PLAN_TEXT_CACHING 59
 QUERY_TIMING 59

查询评估选项 59

INDEX_ADVISOR 59
 NOEXEC 59
 QUERY_DETAIL 59
 QUERY_PLAN 59
 QUERY_PLAN_AFTER_RUN 59
 QUERY_PLAN_AS_HTML 59
 QUERY_PLAN_AS_HTML_DIRECTORY
 59
 QUERY_PLAN_TEXT_ACCESS 59
 QUERY_PLAN_TEXT_CACHING 59
 QUERY_TIMING 59

查询优化选项

AGGREGATION_ALGORITHM_
 PREFERENCE 62
 DEFAULT_HAVING_SELECTIVITY 62

索引

DEFAULT_LIKE_MATCH_SELECTIVITY 62
DEFAULT_LIKE_RANGE_SELECTIVITY 62
EARLY_PREDICATE_EXECUTION 62
IN_SUBQUERY_PREFERENCE 62
INDEX_PREFERENCE 62
JOIN_ALGORITHM_PREFERENCE 62
MAX_HASH_ROWS 62

查询优先级选项 61

查询执行

分布式 35

常见性能问题 54

超线程

服务器开关 5

磁盘空间

Multiplex 数据库 35

交换空间 54

节约 25

截断事务日志 25

限制消息日志大小 25

磁盘条带化 23

内部条带化 24

存储过程

查看分析数据 27

D

DATABASES

非范式化以提高性能 48

DATE 索引 38, 39

Dbospace

限制使用 16

DEFAULT_HAVING_SELECTIVITY 62

DEFAULT_LIKE_MATCH_SELECTIVITY 62

DEFAULT_LIKE_RANGE_SELECTIVITY 62

DELETE OPERATIONS

HG 删除操作 64

DTTM 索引 38, 39

大对象存储 46

大内存

- iqmc 11

- iqtc 11

Multiplex 服务器 11

Simplex 服务器 11

调优

性能 26

调优选项

非滚动游标 17

高速缓存分区 22

为用户进行优化 14

文件系统缓冲 20

限制按行返回的查询 17

限制并发查询 15

限制查询临时空间 16

限制语句数 18, 19

预取高速缓存页 19

预取行 20

针对典型使用进行优化 13

调整数据类型大小 43

抖动和查询执行 56

多列索引 41, 57

多线程

性能影响 5

多用户性能 19

E

EARLY_PREDICATE_EXECUTION 62

F

FLATTEN_SUBQUERIES 58

FORCE_NO_SCROLL_CURSORS 17

FROM 子句 50

分布式查询处理 35

分区 23

分区表 50

分页 54

分页和磁盘交换 54

服务器内存

堆内存 9

共享内存 9

限制 9

服务器配置

内存 9

服务器统计信息 29

负载监控 63

负载平衡

在查询服务器间 35

G

GRAPHICAL_PLAN 60

高速缓存方法 58

高速缓存监控器清单 30

高速缓存内存

- iqmc 11

- iqtz 11
- 参数 11
- 高速缓存大小 11
- 预取页 19
- 故障排除 53
 - 常见性能问题 54
 - 抖动和查询执行 56
 - 分页和磁盘交换 54
 - 目录文件增大 55
 - 索引和行碎片 55
 - 性能问题, 隔离 53
 - 诊断工具 53
- 关键性能指标
 - Multiplex 和节点相关的统计信息 29
 - 服务器统计信息 29
 - 逻辑服务器统计信息 29
- 管理, 资源
 - 缓冲区高速缓存 3
- 过程, 系统
 - sp_iqcolumnuse 63
 - sp_iqindexuse 63
 - sp_iqtableuse 63
 - sp_iqunusedcolumn 63
 - sp_iqunusedindex 63
 - sp_iqunusedtable 63
 - sp_iqworkmon 63
- 过程分析
 - 过程 27

H

- HG 删除操作 64
- HG 索引 38, 39, 57
 - 操作 40
 - 负载 40
 - 模式设计 40
- HNG 索引 38, 39
- HTML_PLAN 60
- 缓冲区高速缓存
 - 布局 22
 - 开销 9
 - 块大小 12
 - 内存, 节省 12
 - 内存使用 9
 - 数据库访问, 多用户 9
 - 数据压缩 12
 - 线程堆栈 9
 - 页大小 12
- 缓冲区高速缓存性能 30
 - 高速缓存监控器 30

- 高速缓存监控器清单 30

I

- I/O
 - 性能建议 23
- IN_SUBQUERY_PREFERENCE 62
- INDEX_ADVISOR 59
- INDEX_PREFERENCE 62
- IOS_FILE_CACHE_BUFFERING 20
- IQ_USE_DIRECTIO 20
- IQGOVERN_MAX_PRIORITY 61
- IQGOVERN_PRIORITY 61
- iqnumbercpus
 - 设置 CPU 数 5

J

- JOIN_ALGORITHM_PREFERENCE 62
- JOIN_PREFERENCE 50
- 监控负载 63
- 监控性能
 - 高速缓存监控器 30
 - 关键性能指标 29
 - 缓冲区高速缓存性能 30
 - 事件分析过程 28
 - 数据库分析过程 27

键

- 外键 43
- 主键 42
- 交换 54
- 交换文件 54
- 结构查询
 - 高速缓存方法 58
- 结构化查询
 - 提高性能 58
- 截断
 - 事务日志 25
- 进程线程模型 5

K

- 开销
 - 缓冲区高速缓存 9
- 空值 44
- 块大小
 - 与 IQ 页大小的关系 12

L

- LF 索引 38, 39
- LONG VARBINARY 45
- LONG VARCHAR 45
- 连接
 - 连接请求 14
 - 限制语句 19
- 连接列 41
- 列
 - Null 值 57
- 临时表 47
- 逻辑服务器 35
- 逻辑服务器统计信息 29

M

- MAX_CURSOR_COUNT 18
- MAX_HASH_ROWS 62
- MAX_QUERY_TIME 61
- MAX_STATEMENT_COUNT 19
- Multiplex 35
- Multiplex 和节点相关的统计信息 29
- Multiplex 数据库
 - 磁盘空间 35
- Multiplex 资源
 - 动态调整 35
- 模式设计
 - HG 索引负载 40
 - LONG VARBINARY 45
 - LONG VARCHAR 45
 - UNION ALL 50
 - 大对象存储 46
 - 多列索引 41
 - 非范式化 48
 - 简单索引选择条件 39
 - 空值 44
 - 连接列 41
 - 临时表 47
 - 散列分区 51
 - 使用索引 38
 - 适当调整数据类型大小 43
 - 索引 37
 - 外键 43
 - 无符号数据类型 45
 - 主键 42
- 目录文件增大 55

N

- 内部条带化 24

内存 9

- IOS_FILE_CACHE_BUFFERING 20
- IQ_USE_DIRECTIO 20
- 绑定内存 12
- 大内存 11
- 堆内存 9
- 多线程 5
- 服务器内存 9
- 高速缓存大小 11
- 高速缓存内存 11
- 进程线程模型 5
- 连接请求 14
- 平衡 I/O 23
- 轻量进程 5
- 所需内存 9
- 文件系统缓冲 20
- 限制 9
- 页大小 12
- 内存使用
 - 其它 9

O

- ORDER BY 子句 57
- OS_FILE_CACHE_BUFFERING 20
- OS_FILE_CACHE_BUFFERING_TEMPDB 20

P

- PREFETCH_BUFFER_LIMIT 19
- PRIMARY KEY 38
- 评估
 - 查询 58
 - 装载 58

Q

- QUERY_PLAN_TEXT_ACCESS 60
- QUERY_PLAN_TEXT_CACHING 60
- 启动参数 14
 - iqwmem 12
 - iqmc 11
 - iqtc 11
- 轻型进程 5
- 清理程序线程 22

S

sp_iqcolumnuse 63
 sp_iqindexuse 63
 sp_iqtableuse 63
 sp_iqunusedcolumn 63
 sp_iqunusedindex 63
 sp_iqunusedtable 63
 sp_iqworkmon 63
 SUBQUERY_CACHING_PREFERENCE 58
 SUBQUERY_FLATTENING_PERCENT 58
 SUBQUERY_FLATTENING_PREFERENCE 58
 SWEEPER_THREADS_PERCENT 22
 散列分区 51
 删除操作
 TEXT 删除操作 66
 WD 删除操作 65
 优化 64
 使用索引
 CMP 38
 DATE 38
 DTTM 38
 HG 38
 HNG 38
 LF 38
 PRIMARY KEY 38
 TIME 38
 UNIQUE CONSTRAINT 38
 UNIQUE HG 38
 使用索引 38
 事件分析过程 28
 事务日志
 截断 25
 输入/输出
 磁盘条带化 23
 内部条带化 24
 事务日志 25
 顺序文件访问 25
 随机文件访问 25
 消息日志 25
 原始设备 23
 数据库
 过程 27
 过程分析 27
 数据库访问
 多用户 9
 数据类型
 LONG VARBINARY 45
 LONG VARCHAR 45

 调整大小 43
 空值 44
 无符号数据类型 45
 数据模型推荐 37
 数据压缩
 页大小 12
 顺序磁盘 I/O 25
 顺序文件访问 25
 随机文件访问 25
 所需内存
 备份 9
 多用户访问 9
 删除泄漏块 9
 数据库校验 9
 线程堆栈 9
 原始分区 9

索引

CMP 39
 DATE 39
 DTTM 39
 HG 37, 39, 57
 HG 索引 40
 HG 索引装载 40
 HNG 39
 LF 37, 39
 TIME 39
 WD 39
 多列 57
 多列索引 41
 类型 37
 索引 40
 索引顾问 37
 索引选择 39
 选择 37
 索引和行碎片 55
 索引选择 38, 39
 CMP 39
 DATE 39
 DTTM 39
 HG 39
 HNG 39
 LF 39
 TIME 39
 WD 39

T

TEXT 删除操作 66
 TIME 索引 38, 39

索引

条件提示
 设置 63
吞吐量 3

U

UNION ALL
 规则 50
 视图 50
 视图性能 50
 装载 49
UNIQUE CONSTRAINT 38
UNIQUE HG 38
USER_RESOURCE_RESERVATION 13

W

WASH_AREA_BUFFERS_PERCENT 22
WD 删除操作 65
WD 索引 38, 39
外键 43
网络
 大型数据传输 6
 设置 6
 网络 6
 性能 6
 性能建议 6
文件访问
 顺序文件 25
 随机文件 25
文件系统缓冲 20
无符号数据类型 45

X

系统过程
 sp_iqcolumnuse 63
 sp_iqindexuse 63
 sp_iqtableuse 63
 sp_iqunusedcolumn 63
 sp_iqunusedindex 63
 sp_iqunusedtable 63
 sp_iqworkmon 63
系统资源
 性能注意事项 3
 资源使用选项 13
下推连接 50
线程
 缓冲区高速缓存 22

线程堆栈
 内存 9
响应时间 3
消息日志
 限制大小 25
性能
 定义 3
 监控和调优 26
 平衡 I/O 23
 设计 3
 限制并发查询 15
 选择正确的索引类型 37
 注意事项 3
 子查询 58
性能问题, 隔离 53
选项

AGGREGATION_ALGORITHM_
 PREFERENCE 62
BT_PREFETCH_MAX_MISS 19
CACHE_PARTITIONS 22
DEFAULT_HAVING_SELECTIVITY 62
DEFAULT_LIKE_MATCH_SELECTIVITY
 62
DEFAULT_LIKE_RANGE_SELECTIVITY
 62
EARLY_PREDICATE_EXECUTION 62
FLATTEN_SUBQUERIES 58
IN_SUBQUERY_PREFERENCE 62
INDEX_PREFERENCE 62
IQ_USE_DIRECTIO 20
JOIN_ALGORITHM_PREFERENCE 62
JOIN_PREFERENCE 50
MAX_CURSOR_COUNT 18
MAX_HASH_ROWS 62
MAX_STATEMENT_COUNT 19
OS_FILE_CACHE_BUFFERING 20
OS_FILE_CACHE_BUFFERING_TEMPDB
 20
PREFETCH_BUFFER_LIMIT 19
QUERY_ROWS_RETURNED_LIMIT 17
QUERY_TEMP_SPACE_LIMIT 16
SUBQUERY_CACHING_PREFERENCE 58
SUBQUERY_FLATTENING_PERCENT 58
SUBQUERY_FLATTENING_PREFERENC
 E 58
SWEEPER_THREADS_PERCENT 22
USER_RESOURCE_RESERVATION 13
WASH_AREA_BUFFERS_PERCENT 22

选项, 查询计划

INDEX_ADVISOR 59
 NOEXEC 59
 QUERY_DETAIL 59
 QUERY_PLAN 59
 QUERY_PLAN_AFTER_RUN 59
 QUERY_PLAN_AS_HTML 59
 QUERY_PLAN_AS_HTML_DIRECTORY
 59
 QUERY_PLAN_TEXT_ACCESS 59
 QUERY_PLAN_TEXT_CACHING 59
 QUERY_TIMING 59

选项, 查询优化

AGGREGATION_ALGORITHM_
 PREFERENCE 62
 DEFAULT_HAVING_SELECTIVITY 62
 DEFAULT_LIKE_MATCH_SELECTIVITY
 62
 DEFAULT_LIKE_RANGE_SELECTIVITY
 62
 EARLY_PREDICATE_EXECUTION 62
 IN_SUBQUERY_PREFERENCE 62
 INDEX_PREFERENCE 62
 JOIN_ALGORITHM_PREFERENCE 62
 MAX_HASH_ROWS 62

Y

页大小

减少内存 12
 块大小 12
 内存, 节省 12
 缺省大小 12
 确定 12
 数据压缩 12

优化

查询 57

优化查询 37, 57

游标

强制为非滚动 17
 限制 18

语句

限制语句 19
 预取高速缓存页 19
 预取行
 控制 20
 原始分区
 内存使用 9
 文件系统 9
 原始设备 23

Z

诊断工具 53

主键 42

装载计划 58

资源

Multiplex 35

资源使用 35

Multiplex 磁盘空间 35

负载平衡 35

使用 UNION ALL 装载 49

索引 37

网络性能 6

资源使用选项 13

典型使用 13

强制为非滚动游标 17

设置可用 CPU 5

限制 Dbpace 使用 16

限制按行返回的查询 17

限制并发查询 15

限制游标 18

限制语句 19

预取高速缓存页 19

预取行 20

子查询

性能 58

展平 58

子查询性能

提高性能 58

