

SYBASE  
技术服务中心  
连载(66)

SYBASE TECHNOLOGY SERVICE FIELD

解决方案

开发人员升级至 ASE 15.0 的 10 大理由  
(七)

(接上期)

## 7.2.4 分布式查询调优

当需要进行包含数据移动的跨服务器查询时,一个重要的方面就是需要确保合适的 CIS 配置被正确调优了。查看以下列表中的配置和值:

表 14 分布式查询调优配置

配置参数	缺省	推荐	注释
cis bulk insert array size	50	1 000 至 10 000	当 ASE 在服务器间传送数据时, 它在源服务器中创建一个数组并一次传送该数组大小的内容。数组越大, 过程越快。
cis bulk insert batch size	0?	0 或 n*array size	它由 CIS 用在服务器间的数据移动上, 并与 bcp 的 -B 选项行为类似。如果保留 0 值, 所有的行传送完成后进行提交。如果是非 0, 它必须被设置为 cis bulk insert array size 的偶数倍。在执行分布式查询时, ASE 能将数据重定位至一张本地工作表, 或可选择对远程表进行游标扫描。它控制游标检索块的大小降低反复检索较大数据集的网络冗余。
cis cursor rows	50	500 至 5000	在 ASE 15.0 中, 缺省连接将此值自动提升至 2 048 但如果归档系统仍旧是 12.5.x 的话, 则有必要增加它。注意, 它假设服务器在不同的站点上; 如果在同一站点, 更好的配置值可能生效, 但是需要配置“additional network memory”参数。
cis packet size	512	2048	

原先的缺省配置对于 OLTP 服务器执行包含几行数据的分布式 OLTP 查询是合适的。通常, 重定位数据被重定位到 tempdb, 缺省就支持 select/intobulkcopy。但是如果使用用户自定义 tempdb, 可能也需要检查该选项。

## 8 新客户端工具(dbisql、计划查看器(plan viewer))

ASE 15.0 中一项更受欢迎的改变是引入了新的

客户端实用工具 Interactive SQL, 也称为 dbisql。虽然在列举了之前的客户端后会看起来有点迷惑 (Data WorkBench、wisql、SQL Advantage 等等), dbisql 为开发人员提供了之前的工具都很少能提供的有趣功能。这些功能包括: 查询编辑器 / 构造器帮助不常开发 SQL 语句的初级开发人员; 直接对结果集进行插入、更新或删除的能力而无需 SQL; 将结果导出为常见导出格式的能力 (html、csv 等); 通过总执行时间百分比来快速定位存储过程中执行最慢的语句; 图形化显示查询计划并提供每个节点的成本; 检索高级优化器参数, 如上次是何时运行统计更新的 (或如果柱状图被优化器认为“缺失”), 优化使用的资源和查询抽象计划。

对于老练的开发人员来说, 第 1 条就无关紧要了。但是, 第 2 条却很有趣, 因为 Sybase 曾经在 System 10 的 Data WorkBench 中提供了轻松录入和修改测试数据的功能, 如图 7。



图 7 dbisql 数据录入和 SQL 生成功能

这将不仅能帮助开发人员, 也可以帮助在参照表中维护数据的数据管理员, 可在需要时使用 dbisql 来输入变化。例如需要删除阻止了索引创建的重复行, 则该行可通过以下步骤快速删除。

(1) 运行查询以查找重复行, 例如:

```
select <col_list>, count(*)
from <table>
group by <col_list> having count(*) > 1
```

(2) 右键点击结果中的每个重复行, 并创建删

除语句。

(3) 在删除语句前加上“`set rowcount 1`”，然后执行。

有其他用户，如在截断其它行后创建插入语句来重新填充一个特定行等。但是，dbisql 真正有价值的是“计划查看器”。在查询运行完后，可从“工具”菜单中选择“计划查看器”并生成查询树的图形计划，树中包含了每个节点相应的成本。用鼠标悬停在任一节点上，节点细节即显示如图 8。



图 8 dbisql 计划查看器查看包含了节点细节的图形化查询计划(鼠标悬停)

如果倾向于 12.5 风格的文本计划，则可通过在图 9 中展现的 Text 标签页中获取。通常有 3 方面值得注意：排序、矢量聚合和表扫描。他们相应的影响可轻松通过在上述的图形化树中选择合适的节点来追溯。



图 9 dbisql 中的 showplan 文字计划

计划查看器也包含“高级”选项页，显示了查询计划优化的细节，诸如查询抽象计划（否则需要在查询中增加 `plan` 子句，这是手工编写的真正快捷方式），如图 10。

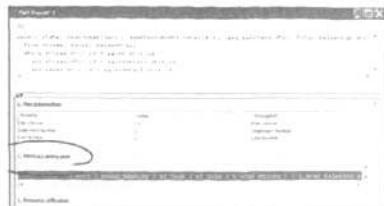


图 10 dbisql 计划查看器高级选项页—优化器信息和索引统计柱状图使用

AQP(抽象查询计划)可简单地被双击并选择，用 Ctrl-C 复制，并粘贴至“PLAN”子句，可在需要的时候修改。除了 AQP 之外，在“高级”标签页中更常被考虑的包括查询成本、查询优化成本和任何索引统计描述，如图 11。

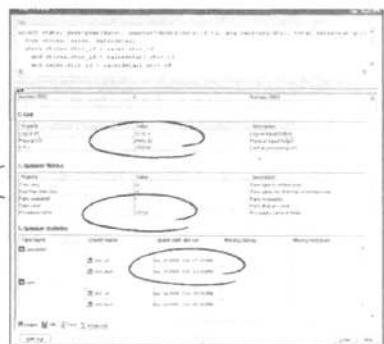


图 11 Dbisql 高级标签页中的查询优化成本

正如所展现的，CPU 的使用率较高（可能是由于排序），使用了约 235KB 的过程缓存来优化查询（例如保持查询计划和柱状图统计）。另外，有一些较早的表统计，上次更新统计时间在 2008 年的圣诞节前。

计划查看器对存储过程有一些有趣的方面。虽然存储过程为应用程序开发人员提供了很多优势，但在以前的 ASE 发布中它们却也是一些最难排除性能问题的代码段。Dbisql 计划查看器和 ASE 中的一些存储过程增强一道，能较为快速地定位存储过程中性能差的语句。之前版本的 dbisql 在为存储过程生成计划时速度较慢，但是 dbisql v11.0 已能够对大型过程快速生成并显示计划，如图 12。

以上是一个存储过程的输出，它调用了其它多个存储过程，总共执行路线是 100 条多表语句。当计划被检索后，语句被列在中部的下拉列表框中（正好在图形查询树和语句文本之上），根据每行所

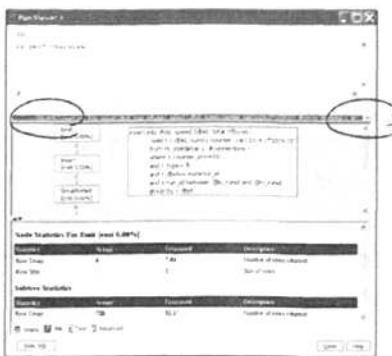


图 12 Dbisql 计划查看器中的存储过程分析

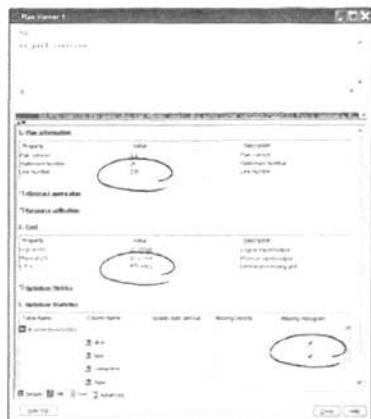
花的时间来排序。因此，存储过程中最慢的行可被快速鉴定，因为它们就在列表的最上方（查看图 13 中的完整下拉语句列表）。



图 13 dbisql 计划查看器中按经历时间排序的存储过程语句列表

正如在第 1 个屏幕截图中展现的，如果将鼠标悬停在选中的语句上，语句的 SQL 文本就会显示。问题是，因为是按照花费的时间来展现，如何找到应该去修复的存储过程行呢？答案在“高级”选项页中，如图 14。

虽然上面的一些输出（AQP 和资源）屏幕截图中由于空间的原因隐藏了一部分，但可以看见，存储过程中的行号被快速找到（118 行），和一些高

图 14 Dbisql 计划查看器—存储过程语句定位和柱状图  
(高级选项页)

CPU 成本，还有缺失的列柱状图（很可能导致合并连接，因此造成 CPU 成本高）。当然，问题是如何甄别存储过程中的第 118 行？在 ASE 15.0 中，使用以下新语句，`sp_helptext` 已被改变可用来输出存储过程中仅需要的行：

```
sp_helptext <proc_name>, <line num>, <num lines>,  
'showsql [ ,linenumbers ]'
```

所以，可以通过类似以下的调用快速定位第 118 行的查询：

```
-- Get 30 lines of SQL text starting at line 110 and  
print line numbers sp_helptext rs_perf_overview, 110,  
30, 'showsql,linenumbers'
```

除此之外，Dbisql 还有其它功能和特性，例如干净执行脚本的能力（无需 SQL Advantage 那样总是乱糟糟的），以及将结果导出为常用格式的能力，比起之前工具来说，也提供了同等巨大的价值。

文/赛贝斯软件（中国）有限公司

(未完待续)

