文章
姚 鑫 · 四月 24, 2021 阅读大约需 12 分钟

第五章 优化查询性能（三）

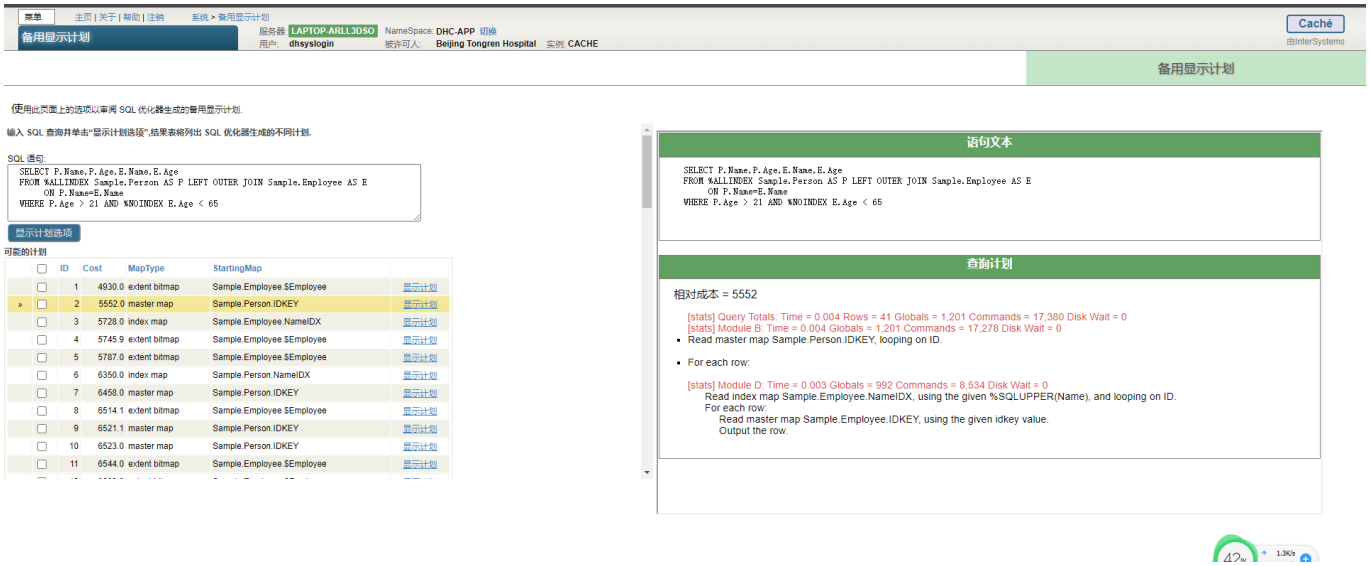
第五章 优化查询性能（三）

查询执行计划

可以使用解释或显示计划工具来显示SELECT、DECLARE、UPDATE、DELETE、TRUNCATE TABLE和一些INSERT操作的执行计划。这些操作统称为查询操作，因为它们使用SELECT查询作为其执行的一部分。InterSystems IRIS在准备查询操作时生成执行计划；不必实际执行查询来生成执行计划。

默认情况下，这些工具显示InterSystems IRIS认为的最佳查询计划。对于大多数查询，有多个可能的查询计划。除了InterSystems IRIS认为最佳的查询计划外，还可以生成和显示备用查询执行计划。

InterSystems IRIS提供以下查询计划工具：
- $SYSTEM.SQL.ExPlan()方法可用于生成和显示XML格式的查询计划以及备选查询计划(可选)。
- SQL EXPLAIN命令可用于生成XML格式的查询计划，还可以选择生成备选查询计划和SQL统计信息。所有生成的查询计划和统计信息都包含在名为Plan的单个结果集字段中。请注意，EXPLAIN命令只能与SELECT查询一起使用。
- 管理门户 - >系统资源管理器 - >SQL界面显示计划按钮。
- 管理门户 — >系统资源管理器 — >工具—>SQL性能工具。



对于生成的%PARALLEL和分片查询，这些工具显示所有适用的查询计划。

使用Explain()方法

可以通过运行$SYSTEM.SQL.Explain()方法生成查询执行计划，示例如下:

```
/// w ##class(PHA.TEST.SQL).SQLExplain()
ClassMethod SQLExplain()
```

```
{
    SET mysql=2
    SET mysql(1)="SELECT TOP 10 Name,DOB FROM Sample.Person "
    SET mysql(2)="WHERE Name [ 'A' ORDER BY Age"
    SET status=$SYSTEM.SQL.Explain(.mysql,{"all":0,"quiet":1,"stats":0,"preparse":0},
,.plan)
    IF status'=1 {WRITE "Explain() failed:" DO $System.Status.DisplayError(status) QU
IT}
    ZWRITE plan
}
```

设置 “ all ” :0选项会生成InterSystems IRIS认为最优的查询计划。
设置 “ all ” :1选项会生成最佳的查询计划和备选的查询计划。
默认值为 “ all ” :0。

结果被格式化为表示xml格式文本的下标数组。

如果指定单个查询计划("all":0)，上述方法调用中的plan变量将具有以下格式:

- plan:显示结果中的下标总数。
- plan(1):总是包含XML格式标签 “ <plan> ” 。
  最后一个下标总是包含XML格式标记 “ </plan> ” 。
- plan(2):总是包含XML格式标签"<sql>"
- plan(3): 总是包含查询文本的第一行。
  如果""preparse":0(默认值)，则返回字面查询文本，并为多行查询的每一行使用额外的下标;在上面的例子中，查询有两行，因此使用了两个下标(plan(3)和plan(4))。如果"prepare":1，则规范化查询文本返回为单行:plan(3)。
- plan(n):总是包含XML格式标签 “ </sql> ” ;
  在上面的例子中，3+mysql = plan(5)。
- plan(n+1):总是包含XML格式的查询cost"<cost value=""407137""/>".
- plan(n+2):总是包含执行计划的第一行。
  这个plan可以是任何长度，可以包含<module>…</module>标签作为单独的下标行，包含生成的执行模块的查询计划。

如果指"all":1 Explain()将生成备用查询计划。计划变量遵循相同的格式，不同之处在于它们使用第一级下标来标识查询计划，而使用第二级下标来标识查询计划的行。因此，plan(1)包含第一个查询计划结果中的二级下标计数，plan(2)包含第二个查询计划结果中的二级下标计数，依此类推。在此格式中，plan(1，1)包含第一个查询计划的XML格式标记 "<plan>" ；plan(2，1)包含第二个查询计划的XML格式标记 "<plan>"，依此类推。唯一不同的是，备用查询计划包含二级零下标(plan(1，0)变量，该变量包含成本和索引信息；此零下标不计入一级下标(plan(1))值。

如果指"stats":1, Explain()将为每个查询计划模块生成性能统计信息。
每个模块的这些统计数据都使用<stats> ... </stats>标记，并立即出现在查询成本之后("<cost value=""407137""/>")和查询计划文本之前。
如果查询计划包含额外的<module>标记，则生成的模块的<stats>将紧接在<module>标记之后，在该模块的查询计划之前列出。
对于每个模块，将返回以下项:
- <ModuleName>:模块名。
- <TimeSpent>:模块的总执行时间，以秒为单位。
- <GlobalRefs>:全局引用的计数。
- <LinesOfCode>:执行的代码行数。
- <DiskWait>:磁盘等待时间，单位为秒。
- <RowCount>:结果集中的行数。
- <ModuleCount>:此模块被执行的次数。
- <Counter>:这个程序被执行的次数。

使用显示计划从InterSystems SQL工具

可以使用Show Plan以以下任何一种方式显示查询的执行计划
- 从管理门户SQL接口。
选择System Explorer，然后选择SQL。
在页面顶部选择带有Switch选项的名称空间。
(可以为每个用户设置管理门户的默认名称空间。)
编写查询，然后按Show Plan按钮。
(还可以通过单击列出查询的Plan选项，从Show History列表调用Show Plan。)
- 从管理门户工具界面。
选择“ 系统资源管理器 ”，然后选择“ 工具 ”，然后选择“ SQL性能工具 ”，然后选择“ SQL运行时统计信息 ”：
 - 在Query Test选项卡中:在页面顶部选择一个带有Switch选项的名称空间。
在文本框中写入查询。
然后按下Show Plan with SQL Stats按钮。
这将在不执行查询的情况下生成一个显示计划。
 - 在View Stats选项卡中:对于列出的查询之一，按Show Plan按钮。
列出的查询包括在执行查询时编写的查询和在查询测试时编写的查询。
- 在SQL Shell中，可以使用SHOW PLAN和SHOW PLANALT Shell命令来显示最近执行的查询的执行计划。
- 通过对缓存的查询结果集运行Show Plan，使用:i%Prop语法将文本替换值存储为属性:

```
SET cqsql=2
SET cqsql(1)="SELECT TOP :i%PropTopNum Name,DOB FROM Sample.Person "
SET cqsql(2)="WHERE Name [ :i%PropPersonName ORDER BY Age"
DO ShowPlan^%apiSQL(.cqsql,0,"",0,$LB("Sample"),"",1)
```

默认情况下，Show Plan以逻辑模式返回值。但是，当从管理门户或SQL Shell调用Show Plan时，Show Plan使用运行时模式。

执行计划:语句文本和查询计划

显示计划执行计划由两个组件组成，即语句文本和查询计划：

语句文本复制了原始查询，但进行了以下修改：管理门户SQL界面中的显示计划按钮显示删除了注释和换行符的SQL语句。空格是标准化的。显示计划按钮显示还执行文字替换，将每个文字替换为？，除非已通过将文字值括在双圆括号中来取消文字替换。使用EXPLAIN()方法显示显示计划时，或者使用SQL运行时统计信息或备用显示计划工具显示显示计划时，不会执行这些修改。

查询计划显示将用于执行查询的计划。查询计划可以包括以下内容：

- 如果查询计划已经冻结，则查询计划的第一行为冻结计划，否则第一行为空。
- “ Relative cost ”是一个整数值，它是从许多因素中计算出来的抽象数字，用于比较同一查询的不同执行计划的效率。这种计算考虑了查询的复杂性、索引的存在和表的大小(以及其他因素)。相对成本对于比较两个不同的查询是没有用的。" Relative cost not available"由某些聚合查询返回，例如COUNT(*)或MAX(%ID)不带WHERE子句。
- 查询计划由一个主模块和(在需要时)一个或多个子组件组成。可以显示一个或多个模块子组件，按字母顺序命名， B: Module:B, Module:C，等等开始，并按执行顺序列出(不一定按字母顺序)。

默认情况下，模块执行处理并使用其结果填充内部临时文件(内部临时表)。
通过指定 /*#OPTIONS {"NoTempFile":1}
*/，可以强制查询优化器创建不生成内部临时文件的查询计划，如注释选项中所述。

对于查询中的每个子查询，都会显示一个命名子查询模块。子查询模块按字母顺序命名。子查询命名在每个命名子查询之前跳过一个或多个字母。因此 Module:B, Subquery:F or Module:D, Subquery:G.当到达字母表末尾时，会对其他子查询进行编号，解析Z=26并使用相同的跳过序列。下面的示例是以Subquery开头的每三个子查询命名序列：F：F，I，L，O，R，U，X，27，30，33。下面的示例是以Subquery开头的每秒一次的子查询命名序列：G：G，I，

K，M，O，Q，S，U，W，Y，27，29。如果子查询调用模块，模块将按字母顺序放在子查询之后，不会跳过。因此，Subquery:H calls Module:I。

- “ Read master map ”作为主模块中的第一个项目符号表示查询计划效率低下。查询计划使用以下映射类型语句之一开始执行Read master map... (no available index), Read index map... (use available index), or Generate a stream of idkey values using the multi-index combination...因为master map读取的是数据本身，而不是数据的索引，所以Read master map...。几乎总是指示低效的查询计划。除非表相对较小，否则应该定义一个索引，以便在重新生成查询计划时，第一个映射显示为read index map...。

某些操作会创建表示无法生成查询计划的显示计划：
- 非查询插入:INSERT... VALUES()命令不执行查询，因此不生成查询计划。
- 查询总是FALSE:在少数情况下，InterSystems IRIS可以在准备查询时确定查询条件总是FALSE，因此不能返回数据。“ 显示计划 ”会在“ 查询计划 ”组件中通知这种情况。例如，包含条件的查询WHERE %ID IS NULL 或 WHERE Name %STARTSWITH('A') AND Name IS NULL不能返回数据,因此，InterSystems IRIS不生成执行计划。查询计划没有生成执行计划，而是表示“ Output no rows ”。如果查询包含具有这些条件之一的子查询，则查询计划的子查询模块表示“ Subquery result NULL, found no rows ”。这种条件检查仅限于涉及NULL的几种情况，并不是为了捕捉所有自相矛盾的查询条件。
- 无效的查询:Show Plan为大多数无效查询显示SQLCODE错误消息。然而，在少数情况下，Show Plan显示为空。例如, WHERE Name = $$$$$ or WHERE Name %STARTSWITH('A")。在这些情况下，Show Plan不显示语句文本，而Query Plan[没有为该语句创建的计划]。这通常发生在分隔文字的引号不平衡时。
当为用户定义的(“ 外部 ”)函数指定了两个或多个前置美元符号而没有指定正确的语法时，也会出现这种情况。

交替显示计划

可以使用管理门户或Explain()方法显示查询的替代执行计划。

使用以下任意一种方法，从管理门户显示查询的备选执行计划:

- 选择系统资源管理器，选择工具，选择SQL性能工具，然后选择备用的显示计划。
- 选择System Explorer，选择SQL，然后从Tools下拉菜单中选择Alternate Show Plans。

使用备用的“ 显示计划 ”工具:
1. 输入一个SQL查询文本，或使用Show History按钮检索一个。
可以通过单击右边的圆形“ X ”圆来清除查询文本字段。
2. 按显示计划选项按钮以显示多个备用显示计划。 Run ... in the background...默认情况下不选中复选框，这是大多数查询的首选设置。建议选择RUN...。对于大型或复杂的查询，请在后台复选框中。当一个长查询在后台运行时，会显示一个View process按钮。单击查看进程将在新选项卡中打开进程详细信息页面。在“ 进程详细信息 ”页中，可以查看进程，还可以挂起、继续或终止进程。
3. 可能的计划按成本升序列出，并带有映射类型和起始映射。
4. 从可能的计划列表中，使用复选框选择要比较的计划，然后按比较显示计划与统计信息按钮以运行这些计划并显示其SQL统计信息。

带有ALL限定符的EXPLAIN()方法显示查询的所有执行计划。它首先显示IRIS认为最优(成本最低)的计划，然后显示备选计划。备选计划按成本升序列出。

以下示例显示最佳执行计划，然后列出备选计划：

```
DO $SYSTEM.SQL.SetSQLStatsFlagJob(3)
SET mysql=1
SET mysql(1)="SELECT TOP 4 Name,DOB FROM Sample.Person ORDER BY Age"
DO $SYSTEM.SQL.Explain(.mysql,{"all":1},,.plan)
ZWRITE plan
```

Stats

显示计划选项列表为每个备用显示计划分配一个成本值，使可以在执行计划之间进行相对比较。

Alternate Show Plan Details为每个查询计划提供了一组查询总数的统计信息(统计信息)，以及(如果适用)每个查询计划模块的统计信息。每个模块的统计信息包括时间(整体性能，以秒为单位)、全局引用(全局引用数)、命令(执行的行数)和读取延迟(磁盘等待，以毫秒为单位)。查询总计统计信息还包括返回的行数。

将查询优化计划写入文件

以下实用程序列出了针对文本文件的一个或多个查询的查询优化计划。

```
QOPlanner^%apiSQL(infile,outfile,eos,schemapath)
```

- infile 包含缓存查询列表的文本文件的文件路径名。指定为带引号的字符串。
- outfile 要列出查询优化计划的文件路径名。指定为带引号的字符串。如果该文件不存在，系统将创建该文件。如果该文件已存在，则InterSystems IRIS会覆盖该文件。
- eos 可选-语句末尾分隔符，用于分隔Infile列表中的各个缓存查询。指定为带引号的字符串。默认值为 “ GO ”。如果此EOS字符串与缓存的查询分隔符不匹配，则不会生成输出文件。
- schemapath 可选-以逗号分隔的方案名列表，用于为未限定的表名、视图名或存储过程名指定方案搜索路径。可以包括DEFAULTSCHEMA，这是当前系统范围内的默认架构。如果infile包含#Import指令，QOPlanner会将这些#Import包/架构名称添加到schemapath的末尾。

以下是调用此查询优化计划列表实用程序的示例。该实用程序将ExportSQL^%qarDDLExport()实用程序生成的文件作为输入，如 “ 缓存查询 ” 一章的 “ 将缓存查询列出到文件 ” 一节中所述。可以生成此查询列表文件，也可以将一个(或多个)查询写入文本文件。

```
  DO QOPlanner^%apiSQL("C:\temp\test\qcache.txt","C:\temp\test\qoplans.txt","GO")
```

从终端命令行执行时，进度会显示在终端屏幕上，如下例所示：

```
Importing SQL Statements from file: C:\temp\test\qcache.txt

Recording any errors to principal device and log file: C:\temp\test\qoplans.txt

  SQL statement to process (number 1):
      SELECT TOP ? P . Name , E . Name FROM Sample . Person AS P ,
      Sample . Employee AS E ORDER BY E . Name
  Generating query plan...Done

  SQL statement to process (number 2):
      SELECT TOP ? P . Name , E . Name FROM %INORDER Sample . Person AS P
      NATURAL LEFT OUTER JOIN Sample . Employee AS E ORDER BY E . Name
  Generating query plan...Done

Elapsed time: .16532 seconds
```

创建的查询优化计划文件包含如下条目：

```
<pln>
```

```
<sql>
 SELECT TOP ? P . Name , E . Name FROM Sample . Person AS P , Sample . Employee AS E
ORDER BY E . Name
</sql>
Read index map Sample.Employee.NameIDX.
Read index map Sample.Person.NameIDX.
</pln>
######
<pln>
<sql>
 SELECT TOP ? P . Name , E . Name FROM %INORDER Sample . Person AS P
    NATURAL LEFT OUTER JOIN Sample . Employee AS E ORDER BY E . Name
</sql>
Read master map Sample.Person.IDKEY.
Read extent bitmap Sample.Employee.$Employee.
Read master map Sample.Employee.IDKEY.
Update the temp-file.
Read the temp-file.
Read master map Sample.Employee.IDKEY.
Update the temp-file.
Read the temp-file.
</pln>
######
```

可以使用查询优化计划文本文件来比较使用不同查询变体生成的优化计划，或者比较不同版本的InterSystems IRIS之间的优化计划。

将SQL查询导出到文本文件时，来自类方法或类查询的查询将以代码行开头：

```
#import <package name>
```

这个#Import语句告诉QOPlanner实用程序使用哪个默认包/模式来生成查询计划。从例程导出SQL查询时，例程代码中SQL语句之前的任何#import行也将位于导出文件中的SQL文本之前。假设从缓存查询导出到文本文件的查询包含完全限定的表引用；如果文本文件中的表引用不是完全限定的，则QOPlanner实用程序使用在运行QOPlanner时在系统上定义的系统范围的默认模式。

#SQL #Caché

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%94%E7%AB%A0-%E4%BC%98%E5%8C%96%E6%9F%A5%E8%AF%A2%E6%80%A7%E8%83%BD%EF%BC%88%E4%B8%89%EF%BC%89