文章
姚 鑫 · 四月 26, 2021 阅读大约需 9 分钟

第六章 SQL性能分析工具包

第六章 SQL性能分析工具包

本章介绍可用于主动分析特定SQL语句的分析工具。这些工具收集有关这些SQL语句执行的详细信息。使用这些信息，开发人员可以采取措施提高低效SQL语句的性能。

根据请求的详细程度，此活动分析可能会显著增加服务器上的负载。因此，SQL性能分析工具包旨在进行协调一致的代码分析工作。它不是用来连续监视执行代码的。

分析工具界面

SQL性能分析工具包为开发人员和支持专家提供了分析特定SQL语句或语句组的能力。通过在执行特定SQL语句期间使用这些工具，它们可以收集详细信息，这些信息可用于单独或跨活动工作负载分析有问题的语句。

要记录的细节级别是可配置的，最细粒度的设置在模块级别收集信息，为语句的查询计划中的不同 “ 步骤 ” 提供信息。

- %SYSTEM.SQL.PTools类方法。提供以下方法来收集以下各项的性能统计信息：
  - 整个系统：setSQLStatsFlag()
  - 特定的命名空间：setSQLStatsFlagByNS()
  - 当前进程或JOB：setSQLStatsFlagJob()
  - 指定的进程或JOB：setSQLStatsFlagByPID()`
  - 可以在SELECT、INSERT、UPDATE或DELETE语句中使用%PROFILE关键字(等效于setSQLStatsFlagJob(2))或%PROFILE_ALL关键字(等效于setSQLStatsFlagJob(3))来仅收集该语句的性能分析统计信息。

使用性能分析工具包方法

可以使用%SYSTEM.SQL.PTools类方法执行以下操作：
- 激活SQL性能统计信息。
- 获取当前的SQL统计信息设置。
- 导出收集的SQL性能统计信息。显示或导出到文件。
- 删除SQL性能统计信息。

本节还包含使用这些方法的程序示例。

注意：%SYSTEM.SQL.PTools类方法是所有新编码的首选类接口。较旧的%SYS.PTools.StatsSQL类方法类似，并将继续受到支持。

激活统计信息收集

可以使用%SYSTEM.SQL.PTools类方法激活统计信息(Stats)代码生成以收集性能统计信息。提供以下方法来收集以下各项的性能统计信息：

- 整个系统：setSQLStatsFlag()

- 特定的命名空间：setSQLStatsFlagByNS()
- 当前进程或JOB：setSQLStatsFlagJob()
- 指定的进程或JOB：setSQLStatsFlagByPID()。如果第一个参数未指定，或指定为$JOB或空字符串(“ ”)，则调用setSQLStatsFlagJob()。因此，SET SQLStatsFlag=$SYSTEM.SQL.SetSQLStatsFlagByPID($JOB,3)等同于SET SQLStatsFlag=$SYSTEM.SQL.SetSQLStatsFlagJob(3).

这些方法采用整数操作选项。它们返回一个冒号分隔的字符串，其第一个元素是先前的统计操作选项。可以使用getSQLStatsFlag()或getSQLStatsFlagByPID()方法确定当前设置。

可以从ObjectScript或SQL调用这些方法，如以下示例所示：
- 从ObjectScript：rtn=##class(%SYSTEM.SQL.PTools).setSQLStatsFlag(2,,8)
- 从SQL : SELECT %SYSTEM.SQL.PTools.setSQLStatsFlag(2,,8)

操作选项

对于setSQLStatsFlag()和setSQLStatsFlagByNS()，可以指定以下Action选项之一：0关闭统计信息代码生成；1打开所有查询的统计信息代码生成，但不收集统计信息(默认设置)；2只记录查询外部循环的统计信息(在查询打开和关闭时收集统计信息)；3记录查询的所有模块级别的统计信息。模块可以嵌套。如果是，则主模块统计信息是包含数字，即完整查询的总体结果。

对于setSQLStatsFlagJob()和setSQLStatsFlagByPID()，操作选项略有不同。它们是：-1关闭此job的统计信息；0使用系统设置值。选项1、2和3与setSQLStatsFlag()相同，并覆盖系统设置。默认值为0。

要收集SQL Stats数据，需要在启用统计代码生成的情况下编译(准备)查询(选项1，默认设置)：

- 从0到1：更改SQL Stats选项后，需要编译包含SQL的例程和类以执行统计代码生成。对于xDBC和动态SQL，必须清除缓存查询以强制重新生成代码。
- 要从1变为2：只需更改SQL Stats选项即可开始收集统计信息。这使可以在运行的生产环境中启用SQL性能分析，并将中断降至最低。
- 从1到3(或从2到3)：更改SQL Stats选项后，需要编译包含SQL的例程和类，以记录所有模块级别的统计信息。对于xDBC和动态SQL，必须清除缓存查询以强制重新生成代码。选项3通常仅用于非生产环境中已识别的性能较差的查询。
- 从1、2或3变为0：要关闭统计代码生成，不需要清除缓存的查询。

收集选项

如果操作选项为2或3，则在调用这些方法之一时，可以指定收集选项值，以指定要收集的性能统计信息。默认情况下收集所有统计信息。

通过将与要收集的每种类型的统计信息相关联的整数值相加，可以指定收集选项。默认值为15(1+2+4+8)。

这些方法将此Collect选项的前值作为第二个冒号分隔的元素返回。可以使用getSQLStatsFlag()或getSQLStatsFlagByPID()方法确定当前设置。默认情况下，收集所有统计信息，返回15作为第二个元素值。

终止选项

统计数据收集将继续，直到终止。默认情况下，收集将无限期继续，直到通过发出另一个setSQLStatsFlag[nnn]()方法终止收集。或者，如果Action选项是1、2或3，可以指定setSQLStatsFlag[nnn]()Terminate选项，可以是经过的时间段(以分钟为单位)，也可以是指定的时间戳。然后指定在该时间段过后重新设置操作选项。例如，字符串“M：120：1”将M(已用分钟)设置为120分钟，结束时操作选项将重置为1。所有其他选项将重置为适用于该操作选项的默认值。

这些方法将此终止选项值的先前值作为第五个冒号分隔的元素作为编码值返回。请参见获取统计信息设置。

获取统计信息设置

第一个冒号分隔值是操作选项设置。第二个冒号分隔值是Collect选项。第三个和第四个冒号分隔值用于特定于名称空间的统计信息收集。第五个冒号分隔值编码终止选项。第六个冒号分隔值指定FlagType：0=系统标志，1=进程/作业标志。

```
  KILL
  DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL("USER")
  DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagByNS("USER",3,,7,"M:5:1")
DisplaySettings
  SET SQLStatsFlag = ##class(%SYSTEM.SQL.PTools).getSQLStatsFlag(0,.ptInfo)
  WRITE "ptInfo array of SQL Stats return value:",!
  ZWRITE ptInfo,SQLStatsFlag
```

导出查询性能统计信息

可以使用%SYSTEM.SQL.PTools的exportStatsSQL()方法将查询性能统计信息导出到文件。此方法用于将统计数据从%SYSTEM.SQL.PTools类导出到文件。

可以调用exportSQLStats()，如以下示例所示：
- 从对象脚本：SET status=##class(%SYSTEM.SQL.PTools).exportSQLStats("$IO")(默认为格式T)。
- 从 SQL: CALL %SYSTEM.SQL.PTools.exportSQLStats('$IO') (默认为格式H).

如果不指定文件名参数，此方法将导出到当前目录。默认情况下，此文件名为PTStatsSQLexportSQLStats，后跟当前本地日期和时间YYYYMMDDHHMMSS。可以指定$IO将数据输出到终端或管理门户显示。如果指定FileName参数，此方法将在当前命名空间的Mgr子目录或您指定的路径位置中创建一个文件。此导出仅限于当前命名空间中的数据。

可以将输出文件格式指定为P(文本)、D(逗号分隔数据)、X(XML标记)、H(HTML标记)或Z(用户定义分隔符)。

默认情况下，此方法导出查询性能统计信息。可以指定将其改为导出SQL查询文本或SQL查询计划数据，如以下示例所示：

- 查询文本： CALL %SYSTEM.SQL.PTools.exportStatsSQL('$IO',,0,1,0)
- 查询计划：CALL %SYSTEM.SQL.PTools.exportStatsSQL('$IO',,0,1,1)

ExportSQLStats()通过去除注释并执行文字替换来修改查询文本。

ExportSQLQuery()可以返回相同的查询文本和查询计划数据。

统计值

返回以下统计信息：
- RowCount-给定查询在主模块中返回的总行数。
- RunCount-自上次编译/准备查询以来已运行的总次数。
- ModuleCount-在查询运行期间输入给定模块的总次数。
- TimeToFirstRow-将给定查询的第一个结果集行返回到主模块所用的总时间。
- TimeSpent-给定查询在给定模块中花费的总时间。
- GlobalRefs-在给定模块中为给定查询完成的全局引用总数。
- LinesOfCode-在给定模块中为给定查询执行的ObjectScript代码的总行数。
- DiskWait(也称为磁盘延迟)-在给定模块中等待给定查询的磁盘读取所花费的总毫秒数。

删除查询性能统计信息

可以使用clearSQLStatsALL()方法删除性能统计信息。默认情况下，它删除为当前命名空间中的所有例程收集的统计信息。可以指定不同的命名空间，和/或将删除限制为特定例程。

性能统计示例

以下示例收集当前进程准备的查询主模块(操作选项2)的性能统计信息，然后使用exportSQLStats()将性能统计信息显示给终端。

```
  DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL()
  DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagJob(2)
  SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
  SET tStatement = ##class(%SQL.Statement).%New()
  SET qStatus = tStatement.%Prepare(myquery)
    IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) Q
UIT}
  SET pStatus = ##class(%SYSTEM.SQL.PTools).exportSQLStats("$IO")
    IF pStatus'=1 {WRITE "Performance stats display failed:"
       DO $System.Status.DisplayError(qStatus) QUIT}
```

下面的示例收集当前进程准备的查询(操作选项3)的所有模块的性能统计信息，然后从嵌入式SQL调用exportSQLStats()以向终端显示性能统计信息：

```
  DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL()
  DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagJob(3)
  SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
  SET tStatement = ##class(%SQL.Statement).%New()
  SET qStatus = tStatement.%Prepare(myquery)
    IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) Q
UIT}
  &sql(CALL %SYSTEM_SQL.PTools_exportSQLStats('$IO'))
```

以下示例收集当前进程准备的查询主模块(操作选项2)的性能统计信息，然后使用StatsSQLView查询显示这些统计信息：

```
  DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL()
  DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagJob(2)
  SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
  SET tStatement = ##class(%SQL.Statement).%New()
  SET qStatus = tStatement.%Prepare(myquery)
   IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QU
IT}
GetStats
  SET qStatus = tStatement.%Prepare("SELECT * FROM %SYS_PTools.StatsSQLView")
   IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QU
IT}
  SET rsstats = tStatement.%Execute()
  DO rsstats.%Display()
  WRITE !!,"End of SQL Statistics"
```

以下示例收集用户命名空间中所有查询的所有模块(操作选项3)的性能统计信息。当统计信息收集时间在1分钟后到期时，它将重新设置为操作选项2，并且所有命名空间的收集范围默认为15(所有统计信息)：

```
  DO ##class(%SYSTEM.SQL.PTools).clearSQLStatsALL("USER")
  DO ##class(%SYSTEM.SQL.PTools).setSQLStatsFlagByNS("USER",3,,7,"M:1:2")
  SET myquery = "SELECT TOP 5 Name,DOB FROM Sample.Person"
  SET tStatement = ##class(%SQL.Statement).%New()
  SET qStatus = tStatement.%Prepare(myquery)
   IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QU
IT}
GetStats
  SET qStatus = tStatement.%Prepare("SELECT * FROM %SYS_PTools.StatsSQLView")
   IF qStatus'=1 {WRITE "%Prepare failed:" DO $System.Status.DisplayError(qStatus) QU
IT}
  SET rsstats = tStatement.%Execute()
  DO rsstats.%Display()
  WRITE !!,"End of SQL Statistics",!
TerminateResetStats
  WRITE "returns:  ",##class(%SYSTEM.SQL.PTools).getSQLStatsFlag(),!
  HANG 100
  WRITE "reset to: ",##class(%SYSTEM.SQL.PTools).getSQLStatsFlag()
```

#SQL #Caché #InterSystems IRIS #InterSystems IRIS for Health

源URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E7%AB%A0-sql%E6%80%A7%E8%83%BD%E5%88%86%E6%9E%90%E5%B7%A5%E5%85%B7%E5%8C%85