

Informix高CPU利用率

目录

[简介](#)

[功能信息](#)

[故障排除方法](#)

[数据分析](#)

[常见问题](#)

简介

本文档介绍需要本地UCCX数据库访问的Unified Contact Center Express(UCCX)活动如何缓慢地执行。它会导致AppAdmin页面加载缓慢、AppAdmin更新需要很长时间才能生效、对墙板查询的响应延迟、Workforce Manager无法查询UCCX 数据以及其他性能和稳定性问题。

在CLI中输入的命令`show process load`显示`uccxoninit` 消耗了大量CPU。`uccxoninit`进程表示在UCCX服务器上运行的UCCX Informix数据库实例。

作者：Sridhar Chandrasekharan、Ryan LaFountain和Cisco TAC工程师Ben Wollak。

功能信息

支持UCCX应用程序的数据库引擎是IBM的Informix。添加到UCCX的AppAdmin页面并由UCCX应用程序生成的配置和历史信息存储在UCCX Informix实例中。

UCCX应用程序提供三个用户，可用于直接访问UCCX数据库，以便提取信息以用于壁板应用程序、质量管理、员工管理和自定义历史报告。

用户信息、每个用户的权限以及每个用户的预期用途如下所述：

- **uccxhruser** — 此用户对UCCX数据库中的许多配置表和历史表具有选择权限，并且应仅用于自定义历史报告和思科统一人力资源管理(WFM)。此用户执行的查询和存储过程可能执行复杂、长时间运行的查询。由于典型历史报告或WFM用户的配置文件，这些查询和存储过程不应像墙板应用程序那样频繁执行。

虽然许多墙板应用程序需要包含在配置表和历史表中的数据，而`uccxhruser`有权访问这些表，但在技术上不支持使用此用户执行复杂、频繁的UCCX数据库查询，以用于墙板应用程序。

- **uccxworkforce** - `uccxworkforce`用户有权访问“团队”、“资源”和“管理引擎”表，并应用于思科统一质量管理(QM)。Workforce Management应使用`uccxhruser`，因为它需要访问`uccxworkforce`用户无法访问的历史数据表。
- **uccxwallboard** — 此用户仅对包含从UCCX引擎内存写入的实时统计信息快照的实时数据库表具有选择权限。选择权限仅限于表`RTCSQsSummary`和`RTICDStatistics`，这意味着`uccxwallboard` 用户应使用简单、非复杂的查询频繁地查询UCCX 数据库，这些查询旨在由墙板应用程序进行。

故障排除方法

在UCCX 10.0版及更高版本中，输入`utils uccx 数据库dbperf start <totalHours> <interval>` 命令以开始对UCCX数据库进行性能跟踪。此命令中的`interval` 参数确定跟踪收集的周期性，而`totalHours` 参数确定在禁用跟踪之前跟踪运行的总时间量。这些参数是可选的。如果在执行命令时未指定这些值，则使用默认值20分钟和10小时。

例如，输入`utils uccx database dbperf start 24 30` 命令以启用对数据库的性能跟踪，并每30分钟收集有关性能统计数据的数据（24小时）。

命令输出中会打印有关收集CLI命令所获数据的说明。

```
admin:utils uccx database dbperf stop
Execution of dbperf has been stopped
Command Successful
admin:utils uccx database dbperf start 24 30
The script runs every 30 minutes over a total duration of 24 hours.
Please collect files after 24 hours

Use "file get activelog uccx/cli/dbperf_171013134928.log" to get the file
Use "file view activelog uccx/cli/dbperf_171013134928.log" to view the file
Command Successful
admin:█
```

在给定`totalHours`后，数据收集将自动停止。要手动停止数据收集，请输入`utils uccx数据库dbperf stop`命令。

```
admin:utils uccx database dbperf stop
Execution of dbperf has been stopped
Command Successful
admin:█
```

如果UCCX版本为9.0(2)或更低版本，`utils uccx数据库dbperf`。命令不可用，请联系技术支持中心(TAC)获取进一步帮助。

TAC将手动执行`dbperf.sh`脚本，该脚本附加到Cisco Bug ID [CSCuc68413](#)，并具有[远程支持](#)帐户访问权限。

当您确定何时手动或通过CLI命令启动脚本执行时，请确保周期和总时间消耗的CPU `uccxoninit` 流程在这些期间波动较大或保持较高，以便收集进行根本原因分析所需的信息。

此外，定期输入`show process load`命令以确定CPU何时波动，以便关联由`dbperf`跟踪脚本收集的日志。

数据分析

`dbperf`脚本执行`onstat -g ses 0`时收集的日志显示针对UCCX数据库发出的活动查询。`uccxoninit`进程上的高CPU通常是执行时间较长的复杂查询的结果。目标是确定消耗最多资源的查询、确定这些查询的源客户端、禁用来自客户端的查询以立即解决问题，以及优化长时间运行的查询以永久解决问题。

在dbperf脚本收集的日志中，查找最有可能导致CPU高波动或uccxoninit进程持续高CPU消耗的查询。

可疑查询：

- 从连接为uccxhruser的会话发出 — 如前所述，uccxhruser 具有从大量配置和历史表中选择信息的权限。因此，可以构建跨多个表的复杂、长时间运行的查询，并可能对UCCX数据库产生性能影响。虽然不是绝对的，但uccxwallboard和uccxworkforce用户对UCCX数据库中表的访问如此有限，因此不太可能出现导致这些用户发出的性能影响的复杂查询。此外，UCCX历史报告客户端(HRC)或Cisco Unified Intelligence Center(CUIC)对UCCX数据库发出的uccxhrcare的查询。这些查询是静态的，无法修改，并且已编写、测试和调整了这些查询以及相关指示，以最大限度地降低对性能的影响。
- 对历史表执行密集查询 — 要求UCCX数据库跨表执行多个联接、选择大量信息或在未索引字段上操作的查询可能会对UCCX数据库造成性能影响。

下面显示了一个复杂查询示例，该查询涉及以uccxhruser形式运行的HR表：

```
session #RSAM total used dynamic
id user tty pid hostname threads memory memory explain
435050 uccxhrus WBBOX 836 10.16.5. 1 90112 80712 off
.....

Current SQL statement :
SELECT x.resourceName, t.eventType, x.datetime, x.extension FROM ( SELECT
t1.resourceID, t1.resourceName, t1.extension, MAX(t2.eventDateTime) AS
datetime FROM Resource AS t1, AgentStateDetail AS t2 WHERE t2.agentID
= t1.resourceID AND t1.assignedTeamID = 21 and t1.active GROUP BY
t1.resourceID, t1.resourceName, t1.extension ) AS x, AgentStateDetail AS
t WHERE t.agentID = x.resourceID AND t.eventDateTime = x.datetime
ORDER BY x.resourceName
```

上例显示了一个复杂查询，该查询由源自主机WBBOX的uccxhruser输入，如果经常输入或在上一个查询返回结果之前定期输入，则该查询可能会对UCCX数据库造成性能影响。

UCCX数据库性能虽然很少，但也会降低(以及 uccxoninit 进程波动或保持较高)，这是内置清除过程的结果。清除过程旨在从UCCX数据库中的配置表和历史表中删除数据，以保持数据库的大小。可以根据数据库的大小或数据库中包含的最早记录来安排清除。

当清除进程运行时，将使用一个查询删除数据。它不会根据要删除的记录数量反复执行。这意味着，如果清除检测到必须删除的大量数据，它会发出单个查询以尝试删除此数据。

修改UCCX AppAdmin页中的清除计划或参数以安排清除以删除大量数据，可能会导致在下次计划清除时完成此单个查询花费大量时间。因此，它会提高数据库实例的CPU利用率。

在dbperf脚本的输出中，可以看到清除查询。它应是用户uccxuser输入的调用sp_purge存储过程的唯一查询。

```
session #RSAM total used dynamic
id user tty pid hostname threads memory memory explain
5628 uccxuser - -1 CC-EXPR- 1 544768 523408 off
```

```
Current SQL statement in procedure db_cra:sp_purge
```

```
proc-counter 0x0x4ccf9260 opcode SQL
```

```
delete from contactroutingdetail
where (exists
(select 1
 from contactcalldetail as ccdr
 where (and (and (and (and (and (= contactroutingdetail.sessionid,
ccdr.sessionid), (= contactroutingdetail.nodeid, ccdr.nodeid)),
(= contactroutingdetail.sessionseqnum, ccdr.sessionseqnum)),
(= contactroutingdetail.profileid, ccdr.profileid)), (>= ccdr.enddatetime,
p_purgefrom)), (< ccdr.enddatetime, p_purgeto))));
```

常见问题

根据最近的Cisco TAC和思科开发工程经验，以下是导致uccxoninit进程的CPU使用率较高的最常见问题:

- 企业中的客户端以uccxhruser身份连接，并在与历史表连接的壁板表 (RTICDStatistics和 RTCSQsSummary) 上运行频繁的复杂查询，以提供壁板或自定义报告解决方案。对于墙板，仅使用uccxwallboard用户，并将查询限制为实时表。不支持从墙板或以类似墙板的频率查询历史或配置表。
- 客户端尝试在活动主节点而不是辅助节点上执行自定义历史报告。仅执行在备用节点上生成历史报告的自定义或默认存储过程。默认情况下，CUIC和HRC在备用节点上执行查询，但在开发自定义历史报告时，开发人员可以选择在哪个节点上运行这些查询或执行这些存储过程。
- Cisco Workforce Management(WFM)在ContactRoutingDetail表上发出复杂查询，以尝试在 startdatetime 字段上进行过滤。默认情况下，此表中的此字段上未创建索引，因此此查询的性能较差。WFM定期发出此查询，以尝试将数据从UCCX同步到WFM。此问题在Cisco Bug ID [CSCtz23710](#)中捕获，并在WFM版本9.0(1)SR4中解决。遇到此问题的客户应升级到包含Cisco Bug ID [CSCtz23710](#)的修复的WFM版本。
- 清除阈值被修改，以便下次计划清除尝试删除大量数据。清除计划修改不是在单个更新中显著地修改清除参数，而是迭代地进行，在清除配置修改之间有几天。这允许清除过程在每次传递中删除较小的数据集，从而提高删除操作的性能。
- 拨号列表表非常大。“拨号列表”表存储上传到“去话活动”的所有联系人。在UCCX版本8.0和8.5中，在将数百万条记录上传到出站活动后，性能问题会导致查询表 (这会导致uccxoninit进程上的CPU使用率较高，并导致AppAdmin页面加载缓慢)。要缓解性能问题，请打开TAC案例，以安装清除DialingList表的Cron作业脚本。在UCCX 9.0版中，已向此表中添加索引，以便从AppAdmin更有效地查询，以尝试提高性能。这一变化解决了除了最极端情况之外的所有问题。在UCCX版本10.0中，拨号列表已拆分为两个表，一个用于活动联系人，另一个用于历史联系人，为此问题提供了全面的修复。