

# Informix公司高CPU利用率

## 目录

[简介](#)

[功能信息](#)

[方法](#)

[数据分析](#)

[常见问题](#)

## 简介

本文描述Unified Contact Center Express (UCCX)活动，要求本地UCCX数据库访问，如何也许迟缓地实行。它造成AppAdmin页迟缓地装载，更新到需要很长时间的Appadmin采取影响，答复的延迟对墙板查询，劳动力管理器无法查询UCCX数据和其他性能和稳定性问题。

show process命令**负载**，输入在CLI，显示uccxoninit消耗很多CPU。uccxoninit进程代表在UCCX服务器运行的UCCX Informix数据库实例。

## 功能信息

支持UCCX应用程序的数据库引擎是国际商用机器公司的Informix公司。被添加到UCCX的AppAdmin页和是由UCCX应用程序导致的配置和历史信息在UCCX Informix公司实例存储。

UCCX应用程序提供能使用直接地访问UCCX数据库为了解压缩信息为墙板应用程序、质量管理，人事管理和自定义历史报告的三个用户。

用户信息、每个用户的权限和每个用户打算的目的描述此处：

- **uccxhruser** -此用户有挑选权限对许多配置和历史表在UCCX数据库，并且应该仅使用自定义历史报告和Cisco Unified人事管理(WFM)。此用户和存储过程执行的查询也许执行复杂，长期的查询。由于一典型历史报告的配置文件或WFM用户、这些查询和存储过程不应该频繁地执行和为墙板应用程序将发生。

虽然许多墙板应用程序要求在uccxhruser访问的配置和历史表内包含的数据，不技术上支持使用此用户执行复杂，常去查询UCCX数据库为墙板应用程序。

- **uccxworkforce** - uccxworkforce用户访问团队、资源和Supervisor表，并且应该用于Cisco Unified质量管理(QM)。人事管理应该使用uccxhruser，当需要对由uccxworkforce用户不是可访问的历史数据表的访问。
- **uccxwallboard** -此用户有挑选仅权限在包含从UCCX引擎的内存写入的实时统计快照的实时数据库表。挑选权限限制对表RtCSQsSummary和RtICDStatistics含义uccxwallboard用户应该使用频繁地查询UCCX数据库与简单，打算的非复杂查询由墙板应用程序来源。

## 方法

在UCCX版本10.0及以上版本，请输入**使用情况uccx数据库dbperf启动<interval> <totalHours>**命令为了开始在UCCX数据库的性能跟踪。在此命令的**间隔**参数确定跟踪收集的周期，并且**totalHours**参数确定跟踪运行的总量时间，在禁用前。这些参数可选。如果他们没指定，当命令被执行默认值20分钟，并且时使用10个小时。

例如，请输入**使用情况uccx开始30 24**发出命令为了启用在数据库的性能跟踪和收集在性能统计数据的数据每30分钟24个小时的**数据库dbperf**。

说明收集CLI命令得到的数据在命令输出中打印。

在给的**totalHours**以后，数据收集自动地终止。为了手工终止数据收集，请输入**stop命令使用情况uccx数据库的dbperf**。

如果UCCX版本是版本9.0(2)或更加早期和**使用情况uccx数据库dbperf**命令不是可用的，请与进一步协助的技术支持中心(TAC)联系。

TAC将执行dbperf.sh脚本手工附加对Cisco Bug ID [CSCuc68413](#)与远程支持帐户访问。

当您什么时候确定手工开始脚本执行或者或时通过CLI命令，周期和总时间，保证**uccxoninit**进程消耗的CPU极大动摇或保持高在那些期限为了收集根本原因分析的必要信息。

另外，当CPU动摇为了关联dbperf跟踪脚本时，收集的日志周期地请输入**load命令的show process**确定。

## 数据分析

**onstat**的dbperf脚本的执行收集的日志- **g ses 0**显示发出UCCX数据库的活动查询。在**uccxoninit**进程的高CPU典型地是需要很长时间执行复杂查询的结果。目标是确定浪费多数资源，确定那些查询的来源客户端，禁用从客户端的查询立即解决方法的，并且优化永久性解决方法的长期的查询的查询。

在dbperf脚本收集的日志，请寻找在CPU的最可能原因高波动或由**uccxoninit**的持续的高CPU消耗处理的查询。

怀疑查询：

- 从作为**uccxhruser**连接的会话发出-如描述前，**uccxhruser**有权限选择信息在配置和历史表外面巨大数目。结果，复杂，在多个表间的长期的查询被修建并且能有在UCCX数据库的性能影响。虽然不绝对，**uccxwallboard**和**uccxworkforce**用户有对表的这样有限访问在UCCX数据库内，导致这些用户发出的性能影响是不太可能的复杂查询。另外，**uccxhrc**发出的查询由UCCX历史报告的客户端(HRC)或Cisco Unified智能中心(CUIC)发出UCCX数据库。这些查询是静态的，并且不可能与相关indices一起被修改和查询，为最小的性能影响写入，测试了和被调整了。
- 执行在历史表的密集查询-要求UCCX数据库执行在表间的多加入，选择重大数额信息或操作非被标注的字段查询可能导致性能影响UCCX数据库。

与介入作为**uccxhruser**运行的一个HR表的一复杂查询的一示例显示此处：

```
session                                #RSAM    total    used    dynamic
id user tty pid hostname threads memory memory explain
```

435050 uccxhirus WBOX 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
```

以上示例显示一复杂查询，参与由从可能导致在UCCX数据库的性能影响的主机来源的uccxhirus WBOX，如果经常被输入了或周期地被输入了，在上一个查询返回结果前。

虽然少见，UCCX数据库性能能也降低(和uccxoninit进程的CPU利用率动摇或保持高)由于内置的清除进程。清除进程设计删除从配置的在UCCX数据库内的数据和历史表为了维护数据库的大小。可以安排清除根据在数据库内或最旧的记录的包含的大小数据库。

当清除进程运行时，数据删除与一查询。它没有执行迭代根据相当数量记录删除。这意味着，如果清除检测必须删除的很多数据，发出单个查询为取消此数据。

清除日程或参数的修改从UCCX AppAdmin页为了安排清除取消很多数据能引起此单个查询，在其次被安排的清除，花些巨大数量的时间完成。所以，它驱动数据库实例的CPU利用率。

在dbperf脚本的输出中，清除查询能被看到。它应该是呼叫sp\_purge存储过程的用户uccxuser参与的唯一查询。

```
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

凭最近的Cisco TAC和思科开发工程体验，这些是导致在uccxoninit进程的高CPU利用率的最编解码器的问题：

- 在企业中连接作为uccxhirus并且运行在墙板表的常见的复杂查询(RtICDStatistics和RtCSQsSummary)加入以历史表为了提供墙板或自定义报告解决方案的一个客户端。为墙板使用，只请使用uccxwallboard用户并且对实时表限制查询。从墙板或与频率类似于墙板不支持能力查询历史或配置表。

- 客户端尝试生成关于活动主节点的自定义历史报告而不是附属节点。只请执行存储过程，自定义或默认，提供关于非引擎主节点的历史报告。CUIC和HRC在默认情况下非万事达节点的执行查询，但是，当开发一份自定义历史报告，开发者有运行这些查询或执行这些存储过程的节点的一选择时。
- 思科人事管理(WFM)在startdatetime字段发出在ContactRoutingDetail表的一复杂查询为了尝试过滤。默认情况下索引在此字段在此表里没有创建，因此此查询性能差。WFM问题周期地此查询为同步数据从UCCX到WFM。此问题在Cisco Bug ID [CSCtz23710](#)在WFM版本9.0(1)SR4捕获和被解决。遇到此问题的客户应该升级到包含Cisco Bug ID的[CSCtz23710](#)一个修正WFM的版本。
- 修改清除阈值这样其次被安排的清除尝试取消很多数据。而不是请极大修改在单个更新的清除参数，清除日程修改用在清除配置修改之间的几天迭代做。这允许清除进程删除在每张通行证的更加小的数据集，改进删除操作的性能。
- DialingList表是庞大的。DialingList表存储所有联系方式上传对出站市场活动。在UCCX版本8.0和8.5中，在数百万记录上传对出站市场活动后，性能问题发生表然后被查询(导致在uccxoninit进程和慢AppAdmin页加载的高CPU)。为了缓和性能问题，请开整理DialingList表cron工作脚本的安装的TAC案例。在UCCX版本9.0中，索引被添加到更加有效的查询的此表从Appadmin为改进性能。此更改解决了总计，但是多数极端情况的问题。在UCCX Release10.0 DialingList拆分到两个表，一活动联系方式的和别的历史联系方式的，为此问题提供一个全面的修正。