

Utilización del Informix CPU elevada

Contenido

[Introducción](#)

[Información sobre la Función](#)

[Metodología](#)

[Análisis de datos](#)

[Problemas Comunes](#)

Introducción

Este documento describe cómo las actividades expresas unificadas del Centro de contacto (UCCX), que requieren el acceso a la base de datos local UCCX, pudieron realizarse lentamente. Hace las páginas de AppAdmin cargar lentamente, las actualizaciones al AppAdmin para tardar un tiempo prolongado para tomar la influencia, un retardo en la respuesta a una interrogación del tablón, administrador de la mano de obra para no poder preguntar los datos UCCX, y otro funcionamiento y problemas de estabilidad.

La **carga del** comando show process, ingresada en el CLI, muestra que el **uccxoninit** consume una gran cantidad de CPU. El proceso del **uccxoninit** representa el caso de la base de datos Informix UCCX que se ejecuta en el servidor UCCX.

Información sobre la Función

El motor de base de datos que soporta la aplicación UCCX es el Informix de IBM. La configuración y la información histórica que se agrega a la página de AppAdmin UCCX y es presentada por la aplicación UCCX se salva en el caso del Informix UCCX.

La aplicación UCCX proporciona a tres usuarios que puedan ser utilizados para acceder la base de datos UCCX directamente para extraer la información con el propósito de las aplicaciones del tablón, de la Administración de calidad, de la Administración de la mano de obra, y de la información histórica de encargo.

La información del usuario, los permisos de cada usuario, y el propósito previsto de cada usuario se describen aquí:

- **uccxhruser** - Este usuario tiene permisos selectos a muchos configuración y las tablas históricas en la base de datos UCCX y debe ser utilizado solamente para la información histórica de encargo y la Administración unificada Cisco de la mano de obra (WFM). Las interrogaciones y los procedimientos almacenados ejecutados por este usuario pudieron realizar las interrogaciones complejas, duraderas. Debido al perfil de una información histórica típica o usuario WFM, estas interrogaciones y los procedimientos almacenados no debe ser ejecutado con frecuencia como ocurriría para una aplicación del tablón.

Aunque muchas aplicaciones del tablón requieran los datos contenidos dentro de la

configuración y de las tablas históricas a las cuales el uccxhruser tiene acceso, no se soporta técnico para utilizar a este usuario para ejecutar complejo, frecuente las interrogaciones contra el UCCXdatabase con el propósito de una aplicación del tablón.

- **uccxworkforce** - El usuario del uccxworkforce tiene acceso a las tablas del equipo, del recurso, y del supervisor y debe ser utilizado para Cisco unificó la Administración de calidad (QM). La Administración de la mano de obra debe utilizar el uccxhruser mientras que requiere el acceso a las tablas de los datos históricos que no son accesibles por el usuario del uccxworkforce.
- **uccxwallboard** - Este usuario tiene permisos selectos solamente en las tablas de base de datos en tiempo real que contienen las fotos de las estadísticas en tiempo real escritas de la memoria del motor UCCX. Los permisos selectos restringidos a las tablas RTCSQsSummary y RTICDStatistics significan que el usuario del uccxwallboard debe ser utilizado para preguntar la base de datos UCCX con frecuencia con las interrogaciones simples, NON-complejas previstas para ser originado por una aplicación del tablón.

Metodología

En la versión 10.0 UCCX y posterior, ingrese el comando del **<interval> del <totalHours> del comienzo del dbperf de la base de datos del uccx del utils** para comenzar el seguimiento del funcionamiento en la base de datos UCCX. El argumento del **intervalo** en este comando determina la periodicidad de la colección de la traza y el argumento de los **totalHours** determina la cantidad total de tiempo que el seguimiento funciona con antes de que se inhabilite. Estos parámetros son opcionales. Si no se especifican cuando el comando se ejecuta los valores predeterminados de 20 minutos y se utilizan 10 horas.

Por ejemplo, ingrese el **comienzo 24 30 del dbperf de la base de datos del uccx del utils** ordenan para habilitar el seguimiento del funcionamiento en la base de datos y recoger los datos sobre las estadísticas de rendimiento cada 30 minutos por 24 horas.

Las instrucciones de recoger los datos obtenidos por el comando CLI se imprimen en la salida de comando.

```
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:█
```

Después de los **totalHours** dados, la obtención de datos para automáticamente. Para parar manualmente la obtención de datos, ingrese el **comando stop del dbperf de la base de datos del uccx del utils**.

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

Si la versión UCCX es la versión 9.0(2) o anterior y el comando del **dbperf de la base de datos del uccx del utils** no está disponible, entre en contacto el Centro de Asistencia Técnica (TAC) para la asistencia adicional.

TAC ejecutará el script de dbperf.sh asociado al Id. de bug Cisco [CSCuc68413](#) manualmente con el acceso remoto de la cuenta del soporte.

Cuando usted determina cuando comenzar la ejecución del script o manualmente o a través del comando CLI, la periodicidad, y el tiempo total, se aseguran que el CPU consumido por el proceso del **uccxoninit** fluctúe perceptiblemente o que siga siendo alto durante esos períodos para recoger la información necesaria para la Análisis de la causa de raíz.

Además, ingrese periódicamente el **comando load del proceso de la demostración** de determinar cuando el CPU fluctúa para correlacionar los registros recogidos por el script del seguimiento del dbperf.

Análisis de datos

Los registros recogidos por la ejecución del script del dbperf del **onstat - ses g 0** interrogaciones del active de la demostración que se publican contra la base de datos UCCX. CPU elevada en el **uccxoninit** el proceso es típicamente el resultado de las interrogaciones complejas que tardan un tiempo prolongado para ejecutar. La meta es determinar las interrogaciones que consumen la mayoría de los recursos, determinan al cliente de la fuente para esas interrogaciones, inhabilitan las interrogaciones del cliente para la resolución inmediata, y optimizan las interrogaciones duraderas para la resolución permanente.

En los registros recogidos por el script del dbperf, busque las interrogaciones que las altas fluctuaciones de la causa más probable en el CPU o CPU elevada consumo continuo por el proceso del **uccxoninit**.

Interrogaciones del sospechoso:

- Se publican de las sesiones conectadas como **uccxhruser** - como descrito anterior, el **uccxhruser** tiene privilegios de seleccionar la información fuera de un gran número de configuración y de tablas históricas. Como consecuencia, el complejo, las interrogaciones duraderas a través de las varias tablas se puede construir y puede tener impactos del rendimiento en la base de datos UCCX. Aunque sea no absoluto, el **uccxwallboard** y los usuarios del **uccxworkforce** tengan tal acceso limitado a las tablas dentro de la base de datos UCCX, las interrogaciones del complejo que causan el impacto del rendimiento publicado por estos usuarios son inverosímiles. Además, las interrogaciones publicadas por el **uccxhrcare** publicado por el cliente de informes históricos UCCX (HRC) o Cisco unificaron el centro de la inteligencia (CUIC) contra la base de datos UCCX. Estas interrogaciones son estáticas y no se pueden modificar y las interrogaciones, junto con los indicios relevantes, haber escrito,

haber probado, y haber ajustado para el efecto de rendimiento mínimo.

- Realice las interrogaciones intensivas en las tablas históricas - Las interrogaciones que requieren la base de datos UCCX para realizar múltiples se unen a través de las tablas, las cantidades significativas selectas de información o actúan los campos encendido NON-puestos en un índice podrían causar los impactos del rendimiento a la base de datos UCCX.

Un ejemplo con una interrogación compleja que implique una tabla HR funcionada con como **uccxhruser** se muestra aquí:

```
session #RSAM total used dynamic
id user tty pid hostname threads memory memory explain
435050 uccxhrus 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
```

El ejemplo anterior muestra una interrogación compleja, ingresada por el **uccxhruser** originado del host **WBOX** que podría causar el impacto del rendimiento en la base de datos UCCX si fue ingresada a menudo o ingresada periódicamente antes de que la interrogación anterior hubiera vuelto los resultados.

Aunque sea raro, el funcionamiento de base de datos UCCX pueda también degradar (y la utilización de la CPU del proceso del **uccxoninit** fluctúa o sigue siendo alta), como resultado del proceso incorporado de la purgación. El proceso de la purgación se diseña para borrar los datos de la configuración y las tablas históricas dentro de la base de datos UCCX para mantener el tamaño de la base de datos. La purgación se puede programar basó en el tamaño de la base de datos o del más viejo expediente contenido dentro de la base de datos.

Cuando el proceso de la purgación se ejecuta, los datos se quitan con una interrogación. No se hacen basados iterativo en la cantidad de expedientes para quitar. Esto significa que si la purgación detecta una gran cantidad de datos que deban ser quitados, publica una sola interrogación en un intento por quitar estos datos.

La modificación del horario o de los parámetros de la purgación de la página de AppAdmin UCCX para programar la purgación para quitar una gran cantidad de datos puede hacer esta sola interrogación, sobre la purgación después programada, para llevar una cantidad significativa de tiempo para completar. Por lo tanto, conduce encima de la utilización de la CPU de la instancia de la base de datos.

En la salida del script del dbperf, la interrogación de la purgación puede ser considerada. Debe ser la única interrogación ingresada por el **uccxuser** del usuario que llama el procedimiento almacenado del **sp_purge**.

```
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))));
```

Problemas Comunes

De acuerdo con la experiencia reciente de la ingeniería del TAC de Cisco y de desarrollo Cisco, éstos son lo más comúnmente posible - los problemas vistos que causan CPU elevada la utilización en el proceso del **uccxoninit**:

- Un cliente en la empresa conecta como **uccxhruser** y funciona con las interrogaciones complejas frecuentes en las tablas del tablón (RTICDStatistics y RTCSQsSummary) unidas con las tablas históricas para proporcionar un tablón o una solución de la información de la aduana. Para el uso del tablón, utilice solamente las interrogaciones del usuario y del límite del **uccxwallboard a las** tablas en tiempo real. La capacidad de preguntar el histórico o las Tablas de configuración de un tablón o con la frecuencia similar a un tablón no se soporta.
- Un cliente intenta ejecutar los informes históricos de encargo en el nodo maestro activo en vez del nodo secundario. Ejecute solamente los procedimientos almacenados, aduana o valor por defecto, que presentan los informes históricos en el nodo maestro del NON-motor. CUIC y el HRC ejecutan las interrogaciones en el nodo del NON-master por abandono, pero cuando desarrolla un informe histórico de encargo, el desarrollador tiene una opción en la cual nodo para funcionar con estas interrogaciones o para ejecutar estos procedimientos almacenados.
- La Administración de la mano de obra de Cisco (WFM) publica una interrogación compleja en la tabla de ContactRoutingDetail para intentar filtrar en el campo del startdatetime. No se crea ningún índice en este campo en esta tabla por abandono, así que el funcionamiento de esta interrogación es pobre. WFM publica esta interrogación periódicamente en un intento por sincronizar los datos de UCCX a WFM. Este problema se captura en el Id. de bug Cisco [CSCtz23710](#) y se resuelve en la versión 9.0(1)SR4 WFM. Los clientes que experimentan este problema deben actualizar a una versión de WFM que contenga un arreglo para el Id. de bug Cisco [CSCtz23710](#).
- Los umbrales de la purgación se modifican tales que la purgación después programada intenta quitar una gran cantidad de datos. Bastante que modifique perceptiblemente los parámetros de la purgación en una sola actualización, las modificaciones del horario de la purgación se hacen iterativo, con algunos días entre las modificaciones de configuración de la purgación. Esto permite que el proceso de la purgación quite a conjuntos de datos más pequeños en cada paso, que mejora el funcionamiento de la operación de eliminación.
- La tabla de DialingList es extremadamente grande. La tabla de DialingList salva todos los contactos cargados a las campañas salientes. En UCCX libera 8.0 y 8.5, después de que

millones de expedientes estén cargados a las campañas salientes, los problemas de rendimiento resultan entonces la tabla se pregunta (que causa CPU elevada en el proceso del uccxoninit y el cargamento lento de la página de AppAdmin). Para atenuar los problemas de rendimiento, abra un caso TAC para la instalación de un script de la tarea cron que limpia la tabla de DialingList. En la versión 9.0 UCCX, un índice fue agregado a esta tabla para interrogaciones más eficaces del AppAdmin en un intento por mejorar el funcionamiento. Este cambio resolvió el problema en todos pero la mayoría de los casos extremos. En la versión 10.0 el DialingList ha estado partido en dos tablas, una para los contactos activos y otra UCCX para los contactos históricos, que proporciona un arreglo completo para este problema.