

Utilização elevada da CPU de Informix

Índice

[Introdução](#)

[Informação da característica](#)

[Metodologia](#)

[Análise de dados](#)

[Problemas comuns](#)

Introdução

Este original descreve como as atividades expressas unificadas do centro de contato (UCCX), que exigem o acesso à base de dados local UCCX, puderam executar lentamente. Faz com que as páginas appadmin carreguem lentamente, as atualizações ao AppAdmin para tomar um muito tempo tomar a influência, um atraso na resposta a uma pergunta do quadro, gerente da mão de obra para ser incapaz de perguntar dados UCCX, e outros desempenho e problemas de estabilidade.

A carga do comando show process, incorporada ao CLI, mostra que o **uccxoninit** consome uma grande quantidade de CPU. O processo do **uccxoninit** representa o exemplo do base de dados Informix UCCX que é executado no server UCCX.

Informação da característica

O motor de base de dados que apoia o aplicativo UCCX é Informix do IBM. A configuração e a informação histórica que é adicionada à página appadmin de UCCX e produzida pelo aplicativo UCCX são armazenadas no exemplo UCCX Informix.

O aplicativo UCCX fornece três usuários que podem ser usados para alcançar diretamente o base de dados UCCX a fim extrair a informação para fins dos aplicativos do quadro, do Gerenciamento de qualidade, do Gerenciamento da mão de obra, e do relatório histórico feito sob encomenda.

A informação sobre o usuário, as permissões de cada usuário, e a finalidade pretendida de cada usuário são descritas aqui:

- **uccxhruser** - Este usuário tem permissões seletas aos muitos configuração e tabelas de histórico no base de dados UCCX e deve ser usado somente para o relatório histórico feito sob encomenda e o Gerenciamento unificado Cisco da mão de obra (WFM). As perguntas e os procedimentos armazenados executados por este usuário puderam executar perguntas complexas, longas. Devido ao perfil de um relatório histórico típico ou usuário WFM, estas perguntas e procedimentos armazenados não deve ser executado frequentemente como ocorreria para um aplicativo do quadro.

Embora muitos aplicativos do quadro exijam os dados contidos dentro da configuração e das tabelas de histórico a que o **uccxhruser** tem o acesso, não é apoiado tecnicamente para usar

este usuário para executar complexo, frequente perguntas contra o UCCXdatabase para fins de um aplicativo do quadro.

- **uccxworkforce** - O usuário do uccxworkforce tem o acesso às tabelas da equipe, do recurso, e do supervisor e deve ser usado para Cisco unificou o Gerenciamento de qualidade (QM). O Gerenciamento da mão de obra deve usar o uccxhruser enquanto exige o acesso às tabelas dos dados históricos que não são acessíveis pelo usuário do uccxworkforce.
- **uccxwallboard** - Este usuário tem permissões seletas somente nas tabelas de base de dados do tempo real que contêm instantâneos das estatísticas em tempo real redigidas da memória do motor UCCX. As permissões seletas restringidas às tabelas RTCSQsSummary e RTICDStatistics significam que o usuário do uccxwallboard deve ser usado para perguntar frequentemente o base de dados UCCX com as perguntas simples, NON-complexas pretendidas ser originado por um aplicativo do quadro.

Metodologia

Em UCCX libere 10.0 e mais atrasado, incorpore o comando do **<interval> do <totalHours> do começo do dbperf do base de dados do uccx dos utils** a fim começar o desempenho que segue no base de dados UCCX. O argumento do **intervalo** neste comando determina a periodicidade da coleção do traço e o argumento dos **totalHours** determina a quantidade total de tempo onde o traçado executa antes que esteja desabilitado. Estes parâmetros são opcionais. Se não são especificados quando o comando está executado os valores padrão de 20 minutos e horas 10 estão usados.

Por exemplo, entre nos **utils o começo 24 30 do dbperf que do base de dados do uccx** comanda a fim permitir o desempenho que segue no base de dados e recolher dados em estatísticas de desempenho cada 30 minutos por 24 horas.

As instruções para recolher os dados obtidos pelo comando CLI são imprimidas na saída do 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:█
```

Após os **totalHours** dados, o levantamento de dados para automaticamente. A fim parar manualmente o levantamento de dados, inscreva o **comando stop do dbperf do base de dados do uccx dos utils**.

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

Se a versão UCCX é a liberação 9.0(2) ou mais adiantado e o comando do **dbperf do base de dados do uccx dos utils** não está disponível, contacte o centro de assistência técnica (TAC) para a assistência adicional.

O TAC executará o script de dbperf.sh anexado à identificação de bug Cisco [CSCuc68413](#) manualmente com acesso remoto da conta do apoio.

Quando você determina quando começar manualmente a execução do script ou através do comando CLI, a periodicidade, e o tempo total, asseguram-se de que o CPU consumido pelo processo do **uccxoninit** flutue significativamente ou permaneça alto durante aqueles períodos a fim recolher a informação necessária para a análise da causa raiz.

Adicionalmente, inscreva periodicamente o **comando load do processo da mostra** determinar quando o CPU flutua a fim correlacionar os logs recolhidos pelo script do traçado do dbperf.

Análise de dados

Os logs recolhidos pela execução do script do dbperf do **onstat - ses g 0** perguntas do active da mostra que são emitidas contra o base de dados UCCX. A alta utilização da CPU no processo do **uccxoninit** é tipicamente o resultado das perguntas complexas que tomam um muito tempo executar. O objetivo é determinar as perguntas que consomem a maioria de recursos, determinam o cliente da fonte para aquelas perguntas, desabilitam as perguntas do cliente para a resolução imediata, e aperfeiçoam as perguntas longas para a definição permanente.

Nos logs recolhidos pelo script do dbperf, procure as perguntas que as flutuações altas da causa mais provável no CPU ou o consumo sustentado da alta utilização da CPU pelo **uccxoninit** processam.

Perguntas do suspeito:

- São emitidos das sessões conectadas como o **uccxhruser** - como descrito mais cedo, o **uccxhruser** tem privilégios selecionar a informação fora de um grande número de configuração e de tabelas de histórico. Em consequência, o complexo, perguntas longas através das tabelas múltiplas pode ser construído e pode ter impactos no desempenho no base de dados UCCX. Embora não absoluto, o **uccxwallboard** e os usuários do **uccxworkforce** têm tal acesso limitado às tabelas dentro do base de dados UCCX, as perguntas do complexo que causam o impacto no desempenho emitido por estes usuários são improváveis. Adicionalmente, as perguntas emitidas pelo **uccxhrcare** emitido pelo Cliente de relatório histórico UCCX (HRC) ou Cisco unificaram o centro da inteligência (CUIC) contra o base de dados UCCX. Estas perguntas são estáticas e não podem ser alteradas e as perguntas, junto com indícios relevantes, ter sido escritas, testado, e ajustado para o impacto de desempenho mínimo.
- Execute perguntas intensivas em tabelas de histórico - As perguntas que exigem o base de dados UCCX para executar múltiplo juntam-se através das tabelas, quantidades significativas seletas de informação ou operam-se campos sobre NON-posicionados poderiam causar impactos no desempenho ao base de dados UCCX.

Um exemplo com uma pergunta complexa que envolva uma tabela HR executada como o **uccxhruser** é mostrado aqui:

```
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
```

O exemplo acima mostrou uma pergunta complexa, incorporada pelo **uccxhruuser** originado do host **WBOX** que poderia causar o impacto no desempenho no base de dados UCCX se foi incorporado frequentemente ou incorporado periodicamente antes que a pergunta precedente retornou resultados.

Embora raro, o desempenho de base de dados UCCX pode igualmente degradar (e a utilização CPU do processo do **uccxoninit** flutua ou permanece alta), em consequência do processo incorporado da remoção. O processo da remoção é projetado suprimir de dados da configuração e de tabelas de histórico dentro do base de dados UCCX a fim manter o tamanho do base de dados. A remoção pode ser programada baseou no tamanho do base de dados ou do registro o mais velho contido dentro do base de dados.

Quando o processo da remoção é executado, os dados estão removidos com a uma pergunta. Não são feitos baseados iterativamente na quantidade de registros para remover. Isto significa que se a remoção detecta uma grande quantidade de dados que devem ser removidos, emite uma única pergunta na tentativa de remover estes dados.

A alteração da programação ou dos parâmetros da remoção da página appadmin UCCX a fim programar a remoção para remover uma grande quantidade de dados pode fazer com que esta única pergunta, em cima da remoção em seguida programada, tome uma quantidade significativa de tempo terminar. Consequentemente, conduz acima da utilização CPU da instância de base de dados.

Na saída do script do dbperf, a pergunta da remoção pode ser considerada. Deve ser a única pergunta incorporada pelo **uccxuser** do usuário que chama o procedimento armazenado do **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 comuns

Baseado na experiência recente da engenharia do tac Cisco e de desenvolvimento Cisco, estas são o mais geralmente - as edições vistas que causam a utilização elevada da CPU no processo do **uccxoninit**:

- Um cliente na empresa conecta como o **uccxhruser** e executa perguntas complexas frequentes nas tabelas do quadro (RTICDStatistics e RTCSQsSummary) juntadas com as tabelas de histórico a fim fornecer um quadro ou uma solução do relatório do costume. Para o uso do quadro, use somente o usuário do **uccxwallboard** e limite perguntas às tabelas em tempo real. A capacidade para perguntar o histórico ou as tabelas de configurações de um quadro ou com a frequência similar a um quadro não é apoiada.
- Um cliente tenta executar relatórios de histórico feitos sob encomenda no nó mestre ativo em vez do nó secundário. Execute somente os procedimentos armazenados, costume ou padrão, que produzem relatórios de histórico no nó mestre do NON-motor. CUIC e o HRC executam perguntas no nó do NON-mestre à revelia, mas quando desenvolvendo uns relatórios de histórico feitos sob encomenda, o colaborador tem uma escolha em que nó para executar estas perguntas ou para executar estes procedimentos armazenados.
- O Gerenciamento da mão de obra de Cisco (WFM) emite uma pergunta complexa na tabela de ContactRoutingDetail a fim tentar filtrar no campo do startdatetime. Nenhum deslocamento predeterminado é criado neste campo nesta tabela à revelia, assim que o desempenho desta pergunta é deficiente. WFM emite esta pergunta periodicamente na tentativa de sincronizar dados de UCCX a WFM. Esta edição é capturada na identificação de bug Cisco [CSCtz23710](#) e resolvida na liberação 9.0(1)SR4 WFM. Os clientes que experimentam esta edição devem promover a uma versão de WFM que contém um reparo para a identificação de bug Cisco [CSCtz23710](#).
- Os pontos iniciais da remoção são alterados tais que a remoção em seguida programada tenta remover uma grande quantidade de dados. Um pouco do que altere significativamente os parâmetros da remoção em uma única atualização, as alterações da programação da remoção são feitos iterativamente, com alguns dias entre modificações de configuração da remoção. Isto permite que o processo da remoção remova os conjuntos de dados menores em cada passagem, que melhora o desempenho da operação da supressão.
- A tabela de DialingList é extremamente grande. A tabela de DialingList armazena todos os contatos transferidos arquivos pela rede às campanhas de partida. Em UCCX libera 8.0 e 8.5, depois que milhões de registros são transferidos arquivos pela rede às campanhas de partida, problemas de desempenho resultam a tabela é perguntado então (que causa a alta utilização da CPU no processo do uccxoninit e na carga lenta da página appadmin). A fim abrandar os problemas de desempenho, abra um caso de TAC para a instalação de um script do cron job que limpa a tabela de DialingList. Em UCCX libere 9.0, um deslocamento predeterminado foi adicionado a esta tabela para umas perguntas mais eficazes do AppAdmin na tentativa de melhorar o desempenho. Esta mudança resolveu a edição em tudo mas na maioria de casos extremos. Em UCCX libere 10.0 que o DialingList foi rachado em duas tabelas, um para contatos ativos e outro para contatos históricos, que fornece um

reparo detalhado para esta edição.