

Informix-gebruik met hoge CPU's

Inhoud

[Inleiding](#)

[Functieinformatie](#)

[Methode voor probleemoplossing](#)

[Gegevensanalyse](#)

[Vaak voorkomende problemen](#)

Inleiding

Dit document beschrijft hoe langzaam Unified Contact Center Express (UCCX) activiteiten, die lokale UCCX-databases vereisen, kunnen worden uitgevoerd. De AppAdmin-pagina's worden langzaam geladen, updates aan AppAdmin duren lang om effect te sorteren, een vertraging in de reactie op een mapvraag, Workforce Manager kan UCCX-gegevens niet-gegevens vragen en andere problemen met prestaties en stabiliteit.

De opdracht **toont proceslading** die in de CLI is ingevoerd, en toont aan dat de **CUXINIT** een grote hoeveelheid CPU verbruikt. Het **uccxoninit**-proces vertegenwoordigt de UCCX Informix-gegevensinstantie die op de UCCX-server draait.

Bijgedragen door Sridhar Chandrasekharan, Ryan LaFountain en Ben Wollak, Cisco TAC-engineers.

Functieinformatie

De database engine die de UCCX-toepassing ondersteunt, is de Informix van IBM. Configuratie- en historische informatie die aan de AppAdmin-pagina van UCCX wordt toegevoegd en door de UCCX-toepassing wordt geproduceerd, wordt opgeslagen in de UCCXInformix-instantie.

De UCCX-toepassing biedt drie gebruikers die kunnen worden gebruikt om direct toegang te krijgen tot de UCCX-database om informatie te extraheren voor wallbordtoepassingen, Quality Management, Workforce Management en aangepaste historische rapportage.

De gebruikersinformatie, de permissies van elke gebruiker en het beoogde doel van elke gebruiker worden hier beschreven:

- **Gebruiker** - Deze gebruiker heeft toegangsrechten voor veel configuratie en historische tabellen in de UCCX-database en moet alleen gebruikt worden voor aangepaste historische rapportage en Cisco Unified Workforce Management (WFM). Vraagstukken en opgeslagen procedures die door deze gebruiker worden uitgevoerd kunnen complexe, lange lopende vragen uitvoeren. Vanwege het profiel van een typische historische verslaggever of WFM-gebruiker zouden deze vragen en opgeslagen procedures niet frequent moeten worden uitgevoerd zoals voor een wallbordapplicatie zou gebeuren.

Hoewel veel maptoepassingen gegevens vereisen die zijn opgenomen in de configuratie en

historische tabellen waartoe de gebruiker toegang heeft, wordt het technisch niet ondersteund om deze gebruiker te gebruiken voor het uitvoeren van complexe, frequente vragen tegen de UCCX-database ten behoeve van een walleteltoepassing.

- **uccxpersoneel** - De gebruiker van uccxpersoneelsbestand heeft toegang tot de tabellen Team, Resource en supervisor en moet worden gebruikt voor Cisco Unified Quality Management (QM). Workforce-beheer dient gebruik te maken van gebruikers omdat dit toegang vereist tot historische datatabellen die niet toegankelijk zijn voor de uccxpersoneelsgebruiker.
- **behangsel** - Deze gebruiker heeft alleen recht op de tabellen in de real-time database die momentopnamen bevatten van realtime statistieken die zijn geschreven uit het geheugen van UCCX Engine. De selectiecompensiteiten die beperkt zijn tot tabellen RTCSQsSummary en RTICDS betekenen dat de uccxwallboard-gebruiker gebruikt moet worden om de UCCX-database regelmatig te vragen met eenvoudige, niet-complexe vragen die bedoeld zijn om uit een maptoepassing te komen.

Methode voor probleemoplossing

In UCCX release 10.0 en hoger voert u de **DBF-opdracht** `<totaaluren> <interval>` uit om met het herstellen van de prestaties te beginnen in de UCCX-database. Het **interval** argument in deze opdracht bepaalt de frequentie van de spoorverzameling en het **totale** uur argument bepaalt de totale hoeveelheid tijd die de overtrekbeurt loopt voordat het wordt uitgeschakeld. Deze parameters zijn optioneel. Als deze niet worden opgegeven wanneer de opdracht wordt uitgevoerd, worden de standaardwaarden van 20 minuten en 10 uur gebruikt.

Voer bijvoorbeeld de **utils uccx database dbperf start 24 30** opdracht in om prestatietracering op de database mogelijk te maken en verzamel om de 30 minuten gegevens over prestatie-statistieken voor 24 uur.

Instructies voor het verzamelen van de gegevens die de CLI-opdracht heeft verkregen, worden in de opdrachtoutput afgedrukt.

```
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:█
```

Nadat de **totale** uren is gegeven, wordt de gegevensverzameling automatisch stopgezet. Om de gegevensverzameling handmatig te stoppen, voert u de opdracht **utils uccx-database dbperf stop** in.

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

Als de UCCX-versie release 9.0(2) of eerder is en de **utils uccx-database dbperf** Aangezien deze opdracht niet beschikbaar is, neemt u contact op met het TAC (Technical Assistance Center) voor verdere assistentie.

TAC voert het DBP.sh-script dat aan Cisco bug ID is toegevoegd handmatig uit [CSCuc68413](#) met toegang tot de externe ondersteuningsaccount.

Wanneer u bepaalt wanneer u de uitvoering van het script handmatig of via de CLI-opdracht wilt starten, de periodiciteit en de totale tijd, dient u ervoor te zorgen dat de CPU die door de **ucxoninit** het proces tijdens die perioden sterk fluctueert of hoog blijft om de nodige informatie te verzamelen voor de analyse van de basisoorzaak .

Daarnaast voert u periodiek de opdracht **voor het verwerken van beelden in** om te bepalen wanneer de CPU fluctueert, zodat de logbestanden die zijn verzameld door het dbperf-overtreksript gecorreleerd worden.

Gegevensanalyse

De door de uitvoering van **onstat -g ses 0** verzamelde logboeken van het dbperf script tonen actieve vragen die tegen de UCCX database zijn afgegeven. Hoge CPU's in het **ucxonit**-proces zijn doorgaans het resultaat van complexe vragen die lang duren om uit te voeren. Het doel is om de vragen te bepalen die de meeste middelen verbruiken, de bronclient voor die vragen te bepalen, de vragen van de client voor onmiddellijke oplossing uit te schakelen en de lange lopende vragen voor permanente oplossing te optimaliseren.

In de logbestanden die door het dbperf-script zijn vergaard, zoekt u naar vragen die waarschijnlijk hoge fluctuaties in CPU's of een duurzaam hoog CPU-verbruik bij het **ucxonit**-proces veroorzaken.

Verdachte vragen:

- Wordt uitgegeven uit sessies die als **gebruiker** zijn aangesloten - Zoals eerder beschreven, **heeft** Aironet rechten om informatie uit een groot aantal configuratie en historische tabellen te selecteren. Als resultaat hiervan kunnen complexe, lange-termijn vragen over meerdere tabellen worden geconstrueerd en kunnen prestaties impact hebben op de UCCX database. Hoewel niet absoluut, hebben de gebruikers van het **uccxwallboard** en de **uccxploddiensten** zo beperkte toegang tot tabellen in de UCCX-database, zijn complexe vragen die door deze gebruikers afgegeven prestatiekerneffecten veroorzaken onwaarschijnlijk. Daarnaast zijn vragen die zijn verstrekt door UCCX door de UCCX Historical Reporting Client (HRC) of Cisco Unified Intelligence Center (CUIC) tegen de UCCX-database verstrekt. Deze vragen zijn statisch en kunnen niet worden gewijzigd en de vragen, samen met de relevante indexen, zijn geschreven, getest en afgestemd op een minimale impact op de prestaties.
- Voer intensieve vragen uit over historische tabellen - vragenlijsten die vereisen dat de UCCX-database meerdere verbindingen tussen tabellen moet uitvoeren, significante hoeveelheden

informatie selecteert of op niet-geïndexeerde velden moet werken, kunnen prestatieketens in de UCCX-database veroorzaken.

Een voorbeeld met een complex query dat een HR tabel run zoals **CUBE** wordt weergegeven:

```
session #RSAM total used dynamic
id user tty pid hostname threads memory memory explain
435050 uccxhrus WBB0X 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
```

Het bovenstaande voorbeeld toont een ingewikkelde vraag, die door **een** frauderende **gebruiker** is ingevoerd en afkomstig is van de **WBB0X** van de gastheer die een invloed op de UCCX-database kan hebben als deze vaak werd ingevoerd of periodiek werd ingevoerd voordat de vorige query resultaten had teruggegeven.

Hoewel zeldzaam, kunnen UCCX-databases ook worden gedegradeerd (en kan het CPU-gebruik van de **ucxoninit** het proces fluctueert of blijft hoog) als gevolg van het ingebouwde zuiveringsproces. Het zuiveringsproces is ontworpen om gegevens uit de configuratie en historische tabellen in de UCCX-database te verwijderen om de grootte van de database te behouden. De zuivering kan worden gepland op basis van de grootte van de database of het oudste record in de database.

Wanneer het zuiveringsproces wordt uitgevoerd, worden de gegevens met één query verwijderd. Dit gebeurt niet op basis van de hoeveelheid gegevens die moet worden verwijderd. Dit betekent dat als de purge een grote hoeveelheid gegevens detecteert die moet worden verwijderd, er één query wordt gegeven in een poging om deze gegevens te verwijderen.

De wijziging van het zuiveringsschema of de parameters van de UCCX AppAdmin-pagina om de purge te plannen voor het verwijderen van een grote hoeveelheid gegevens kan ertoe leiden dat deze enkele query bij de volgende geplande purge een aanzienlijke hoeveelheid tijd nodig heeft om te voltooien. Daarom wordt het CPU-gebruik van de gegevensinstantie erdoor bevorderd.

In de output van het dbperf script kan de purge query worden gezien. Het moet de enige query zijn die is ingevoerd door gebruiker **uccxuser** die de **sp_purge** opslagprocedure aanroept.

```
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.enddatetetime,
p_purgefrom)), (< ccdr.enddatetetime, p_purgeto))));
```

Vaak voorkomende problemen

Gebaseerd op recente ervaring van Cisco TAC en Cisco Development Engineering, zijn dit de meest voorkomende problemen die een hoog CPU-gebruik bij het productieproces veroorzaken:

- Een cliënt in de onderneming verbindt zich als **koper** en voert frequente complexe vragen uit op de wallbordtabellen (RTICDStatistics en RTCSQs Summary), die met de historische tabellen worden meegeleverd om een wallboard of een aangepaste meldingsoplossing te bieden. Voor het gebruik van het muurbord, gebruikt u alleen de gebruiker **van het** klembord en beperkt u de vragen tot de tabellen in realtime. De mogelijkheid om de historische of configuratietabellen vanaf een muurbord te vragen of met een frequentie die vergelijkbaar is met een muurbord wordt niet ondersteund.
- Een client probeert aangepaste historische rapporten over het actieve primaire knooppunt uit te voeren in plaats van het secundaire knooppunt. Alleen opgeslagen procedures uitvoeren, op maat of standaard, die historische rapporten over het standby knooppunt opleveren. CUIC en HRC voeren vragen uit over het stand-by knooppunt per default, maar bij het ontwikkelen van een aangepast historisch rapport heeft de developer een keuze op basis waarvan u deze vragen kunt uitvoeren of deze opgeslagen procedures kunt uitvoeren.
- Cisco Workforce Management (WFM) geeft een complexe vraag uit in de ContactRoutingDetail-tabel om te proberen een filter in het veld Begintijd in te stellen. Er wordt in deze tabel standaard geen index voor dit veld aangemaakt, dus de prestaties van deze query zijn laag. WFM geeft deze query regelmatig uit in een poging om gegevens van UCCX naar WFM te synchroniseren. Deze kwestie wordt opgenomen in Cisco bug-ID [CSCtz23710](#) en wordt opgelost in WFM release 9.0(1)SR4. Klanten die deze kwestie ervaren moeten upgrade uitvoeren naar een versie van WFM die een oplossing bevat voor Cisco bug-ID [CSC23710](#).
- De zuiveringsdrempels worden zodanig aangepast dat de volgende geplande verwijdering probeert een grote hoeveelheid gegevens te verwijderen. In plaats van de zuiveringsparameters in één update aanzienlijk te wijzigen, worden de wijzigingen in het zuiveringsschema iteratief gemaakt, met een paar dagen tussen wijzigingen in de zuiveringsconfiguratie. Hiermee kan het zuiveringsproces worden gebruikt om kleinere gegevensreeksen in elke pas te verwijderen, die de prestaties van de verwijderd handeling verbeteren.
- De tabel DialingList is extreem groot. De tabel DialingList slaat alle contacten op die naar uitgaande campagnes zijn geüpload. In UCCX release 8.0 en 8.5, nadat miljoenen records zijn geüpload naar uitgaande camera's, worden er prestatiekwesties weergegeven die ertoe leiden dat de tabel niet waterpas staat (waardoor een hoge CPU in het uccxoninit-proces ontstaat en de pagina-lading van AppAdmin langzaam verloopt). Om de problemen met de prestaties te verzachten, opent u een TAC-case voor de installatie van een cron job script dat de DialingList-tabel reinigt. In UCCX release 9.0 is er een index toegevoegd aan deze tabel voor effectievere vragen vanuit AppAdmin in een poging de prestaties te verbeteren. Deze verandering heeft het probleem in alle behalve de meest extreme gevallen opgelost. In UCCX

release 10.0 is de DialingList opgesplitst in twee tabellen, een voor actieve contacten en een ander voor historische contacten, die een uitgebreide oplossing voor deze kwestie biedt.