

# Utilizzo elevato della CPU Informix

## Sommario

[Introduzione](#)

[Informazioni sulle funzionalità](#)

[Metodologia di risoluzione dei problemi](#)

[Analisi dei dati](#)

[Problemi comuni](#)

## Introduzione

In questo documento viene descritto il modo in cui le attività di Unified Contact Center Express (UCCX), che richiedono l'accesso al database UCCX locale, potrebbero risultare lente. Questo rallenta il caricamento delle pagine di AppAdmin, richiede tempi lunghi per l'esecuzione degli aggiornamenti di AppAdmin, comporta un ritardo nella risposta a una query dello sfondo, l'impossibilità per Workforce Manager di eseguire query sui dati UCCX e altri problemi di prestazioni e stabilità.

Il comando **show process load**, immesso nella CLI, mostra che la **funzionalità uccxoninit** utilizza una grande quantità di CPU. Il processo **uccxoninit** rappresenta l'istanza di database UCCX Informix in esecuzione sul server UCCX.

Contributo di Sridhar Chandrasekharan, Ryan LaFountain e Ben Wollak, Cisco TAC Engineers.

## Informazioni sulle funzionalità

Il motore di database che supporta l'applicazione UCCX è Informix di IBM. Le informazioni di configurazione e cronologiche aggiunte alla pagina AppAdmin di UCCX e prodotte dall'applicazione UCCX vengono archiviate nell'istanza UCCX Informix.

L'applicazione UCCX mette a disposizione tre utenti che possono essere utilizzati per accedere direttamente al database UCCX al fine di estrarre informazioni per applicazioni di wallboard, gestione della qualità, gestione della forza lavoro e creazione di rapporti cronologici personalizzati.

Di seguito sono descritte le informazioni utente, le autorizzazioni di ogni utente e lo scopo designato di ogni utente:

- **uccxhruser**: questo utente dispone delle autorizzazioni di selezione per molte tabelle di configurazione e di cronologia nel database UCCX e deve essere utilizzato solo per report cronologici personalizzati e per Cisco Unified Workforce Management (WFM). Le query e le stored procedure eseguite da questo utente potrebbero eseguire query complesse con esecuzione prolungata. A causa del profilo di un tipico utente di WFM o di reporting cronologico, queste query e stored procedure non devono essere eseguite frequentemente come accade per un'applicazione wallboard.

Sebbene molte applicazioni di wallboard richiedano dati contenuti nelle tabelle di

configurazione e cronologiche a cui l'utente uccxhruser ha accesso, tecnicamente non è supportato l'utilizzo di questo utente per eseguire complesse e frequenti query sul database UCCX per gli scopi di un'applicazione di wallboard.

- **uccxworkforce** - L'utente uccxworkforce ha accesso alle tabelle Team, Resource e Supervisor e deve essere utilizzato per Cisco Unified Quality Management (QM). Workforce Management deve utilizzare uccxhruser poiché richiede l'accesso a tabelle di dati cronologici non accessibili dall'utente uccxworkforce.
- **lavagna** - Questo utente dispone di autorizzazioni di selezione solo sulle tabelle di database in tempo reale che contengono snapshot di statistiche in tempo reale scritte dalla memoria del motore UCCX. Le autorizzazioni di selezione limitate alle tabelle RTCSQsSummary e RTICDStatistic indicano che l'utente di uccxwallboard deve essere utilizzato per eseguire frequentemente query sul database UCCX con query semplici e non complesse che devono essere originate da un'applicazione wallboard.

## Metodologia di risoluzione dei problemi

In UCCX release 10.0 e successive, immettere il comando **utils uccx database dbperf start <totalHours> <interval>** per avviare la traccia delle prestazioni sul database UCCX. L'argomento **interval** in questo comando determina la periodicità della raccolta di tracce e l'argomento **totalHours** determina la quantità totale di tempo in cui la traccia viene eseguita prima che venga disabilitata. Questi parametri sono facoltativi. Se non vengono specificati quando il comando viene eseguito, vengono utilizzati i valori predefiniti di 20 minuti e 10 ore.

Ad esempio, immettere il comando **utils uccx database dbperf start 24 30** per abilitare la traccia delle prestazioni sul database e raccogliere dati sulle statistiche delle prestazioni ogni 30 minuti per 24 ore.

Le istruzioni per raccogliere i dati ottenuti dal comando CLI sono stampate nell'output del 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:█
```

Dopo il valore **totalHours** specificato, la raccolta dei dati viene interrotta automaticamente. Per interrompere manualmente la raccolta dei dati, immettere il comando **utils uccx database dbperf stop**.

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

Se la versione UCCX è la 9.0(2) o precedente e **utils uccx database dbperf** non è disponibile, contattare il Technical Assistance Center (TAC) per ulteriore assistenza.

Lo script dbperf.sh associato all'ID bug Cisco CSCuc68413 verrà eseguito manualmente con l'accesso all'account del supporto remoto.

Quando si stabilisce quando avviare l'esecuzione dello script manualmente o tramite il comando CLI, la periodicità e il tempo totale, assicurarsi che la CPU utilizzata dal **uccxoninit** durante questi periodi, il processo varia in modo significativo o rimane elevato per raccogliere le informazioni necessarie per l'analisi della causa principale.

Inoltre, immettere periodicamente il comando **show process load** per determinare quando fluttua la CPU e correlare i log raccolti dallo script di traccia dbperf.

## Analisi dei dati

I log raccolti dall'esecuzione dello script dbperf di **onstat -g ses 0** mostrano le query attive eseguite sul database UCCX. L'elevato livello di CPU nel processo **uccxoninit** è in genere il risultato di query complesse la cui esecuzione richiede molto tempo. L'obiettivo è determinare le query che utilizzano la maggior parte delle risorse, determinare il client di origine per tali query, disabilitare le query dal client per la risoluzione immediata e ottimizzare le query a esecuzione prolungata per la risoluzione permanente.

Nei log raccolti dallo script dbperf, cercare le query che molto probabilmente causano fluttuazioni elevate nella CPU o un consumo elevato di CPU sostenuto dal processo **uccxoninit**.

Query sospette:

- Vengono emessi da sessioni connesse come **uccxhruser**: come descritto in precedenza, **uccxhruser** dispone dei privilegi per selezionare informazioni da un vasto numero di tabelle di configurazione e cronologiche. Di conseguenza, query complesse con esecuzione prolungata su più tabelle possono essere create e avere un impatto sulle prestazioni del database UCCX. Sebbene non assoluto, gli utenti **uccxwallboard** e **uccxworkforce** dispongono di un accesso così limitato alle tabelle all'interno del database UCCX, è improbabile che vengano eseguite query complesse che determinano un impatto sulle prestazioni. Inoltre, le query emesse da Cisco Unified Intelligence Center (CUIC) da parte di UCCX Historical Reporting Client (HRC) o Cisco Unified Intelligence Center rispetto al database UCCX. Queste query sono statiche e non possono essere modificate e sono state scritte, testate e ottimizzate insieme agli indici rilevanti per ridurre al minimo l'impatto sulle prestazioni.
- Esecuzione di query intensive su tabelle cronologiche: le query che richiedono che il database UCCX esegua più join tra le tabelle, selezioni quantità significative di informazioni o operi su campi non indicizzati possono influire sulle prestazioni del database UCCX.

Di seguito è riportato un esempio di query complessa che interessa una tabella HR eseguita come **uccxhruser**:

```
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
```

Nell'esempio precedente viene illustrata una query complessa, immessa da **uccxhruser** e originata dalla **WBBOX** dell'host, che potrebbe influire sulle prestazioni del database **UCCX** se è stata immessa spesso o periodicamente prima che la query precedente restituisse dei risultati.

Sebbene rari, anche le prestazioni del database **UCCX** possono peggiorare (e l'utilizzo della CPU **uccxoninit** fluttua o rimane elevato), come risultato del processo di eliminazione incorporato. Il processo di eliminazione è progettato per eliminare i dati dalle tabelle di configurazione e cronologiche all'interno del database **UCCX** al fine di mantenere le dimensioni del database. È possibile pianificare la rimozione in base alle dimensioni del database o al record meno recente contenuto nel database.

Quando viene eseguito il processo di eliminazione, i dati vengono rimossi con una query. Questa operazione non viene eseguita in modo iterativo in base alla quantità di record da rimuovere. Ciò significa che se l'eliminazione rileva una grande quantità di dati da rimuovere, esegue una singola query nel tentativo di rimuovere tali dati.

La modifica della pianificazione o dei parametri di rimozione dalla pagina **UCCX AppAdmin** per pianificare l'eliminazione in modo da rimuovere una grande quantità di dati può causare il completamento di una singola query alla successiva eliminazione pianificata. Di conseguenza, aumenta l'utilizzo della CPU dell'istanza di database.

Nell'output dello script **dbperf**, è possibile visualizzare la query di eliminazione. Deve essere l'unica query immessa dall'utente **uccxuser** che chiama la stored procedure **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))));
```

## Problemi comuni

In base alla recente esperienza di Cisco TAC e Cisco Development Engineering, si tratta dei

problemi più comuni che causano un elevato utilizzo della CPU nel processo **uccxoninit**:

- Un client dell'azienda si connette come **uccxhruser** ed esegue frequenti query complesse sulle tabelle di wallboard (RTICDStatistic e RTCSQsSummary) unite alle tabelle cronologiche per fornire uno sfondo o una soluzione di reporting personalizzata. Per l'utilizzo degli sfondi, utilizzare solo l'utente **uccxwallboard** e limitare le query alle tabelle in tempo reale. La possibilità di eseguire query sulle tabelle cronologiche o di configurazione da uno sfondo o con una frequenza simile a uno sfondo non è supportata.
- Un client tenta di eseguire report cronologici personalizzati sul nodo primario attivo anziché sul nodo secondario. Eseguire solo stored procedure, personalizzate o predefinite, che generano report cronologici sul nodo in standby. CUIC e HRC eseguono le query sul nodo in standby per impostazione predefinita, ma quando si sviluppa un report cronologico personalizzato, lo sviluppatore può scegliere su quale nodo eseguire le query o le stored procedure.
- Cisco Workforce Management (WFM) esegue una query complessa sulla tabella ContactRoutingDetail per tentare di filtrare in base al campo startdatetime. Per impostazione predefinita, in questo campo della tabella non viene creato alcun indice, pertanto le prestazioni della query sono scarse. WFM esegue periodicamente questa query nel tentativo di sincronizzare i dati da UCCX a WFM. Il problema è stato acquisito con l'ID bug Cisco [CSCtz23710](#) e viene risolto con la versione 9.0(1)SR4 di WFM. Gli utenti che riscontrano questo problema devono eseguire l'aggiornamento a una versione di WFM che contiene una correzione per l'ID bug Cisco [CSCtz23710](#).
- Le soglie di rimozione vengono modificate in modo che la successiva eliminazione pianificata tenti di rimuovere una grande quantità di dati. Anziché modificare in modo significativo i parametri di rimozione in un singolo aggiornamento, le modifiche alla pianificazione di rimozione vengono eseguite in modo iterativo, con pochi giorni tra le modifiche alla configurazione di rimozione. In questo modo il processo di eliminazione rimuove set di dati più piccoli in ogni passaggio, migliorando le prestazioni dell'operazione di eliminazione.
- La tabella DialingList è estremamente grande. Nella tabella DialingList sono memorizzati tutti i contatti caricati in Campagne in uscita. Nelle release 8.0 e 8.5 di UCCX, dopo il caricamento di milioni di record in Campagne in uscita, si verificano problemi di prestazioni e quindi viene eseguita una query sulla tabella (con conseguente aumento della CPU nel processo di uccxoninit e rallentamento del caricamento della pagina AppAdmin). Per ridurre i problemi di prestazioni, aprire una richiesta TAC per installare uno script di processo cron che pulisca la tabella DialingList. In UCCX release 9.0, è stato aggiunto un indice a questa tabella per ottenere query più efficaci da AppAdmin nel tentativo di migliorare le prestazioni. Questo cambiamento ha risolto il problema in tutti i casi tranne in quelli più estremi. In UCCX release 10.0, la lista di composizione è stata suddivisa in due tabelle, una per i contatti attivi e l'altra per i contatti storici, in modo da fornire una soluzione completa al problema.