Informix の高 CPU 使用率

目次

概要

ローカル UCCX データベースアクセスを必要とする Unified Contact Center Express （UCCX）アクティビティがどのようにゆっくり実行するかもしれませんかこの資料に記述されています。 それにより UCCX データを問い合わせることができないために AppAdminページは労働力マネージャおよび他のパフォーマンスおよび安定性の問題影響を、ウォールボード クエリへの応答の遅延奪取 するのに、長い 時間をかける AppAdmin に、更新ゆっくりロードします。

CLI で入るコマンド **show process ロードは uccxoninit が**多量の CPU を消費することを示します。 **uccxoninit** プロセスは UCCX サーバで動作する UCCX Informixデータベース 例を表します。

機能情報

UCCX アプリケーションをサポートするデータベース エンジンは IBM の Informix です。 UCCX の AppAdminページに追加され、UCCX アプリケーションによって生成 される 設定およびヒストリ情報は UCCX Informix 例で保存されます。

UCCX アプリケーションはウォールボード アプリケーション、品質 管理、人事管理およびカスタム史的記事の目的で情報を得るために UCCX データベースに直接アクセスするのに使用できる 3 人のユーザを提供します。

各ユーザのユーザ情報、各ユーザの権限および意図されていた目的はここに記述されています:

- **uccxhruser** -このユーザは UCCX データベースで多くの設定およびヒストリテーブルに選択権限があり、カスタム史的記事および Cisco Unified 人事管理（WFM）のためにだけ使用する必要があります。 このユーザが実行するクエリおよび保存された手順は複雑な、長期クエリを行うかもしれません。 ウォールボード アプリケーションのために発生するようにか WFM ユーザ、これらのクエリおよび保存された手順典型的な史的記事のプロファイルが原因で頻繁に実行されるべきではないです。

  多くのウォールボード アプリケーションが uccxhruser にアクセスできるヒストリテーブルおよび設定の内で含まれているデータを必要とするがこのユーザを複雑実行するのに使用することを、ウォールボード アプリケーションの目的で UCCXdatabase に対して技術的に度々行きますクエリにサポートしません。
- **uccxworkforce** - uccxworkforce ユーザはチーム、リソースおよびスーパバイザ 表にアクセスでき、Cisco Unified 品質 管理（QM）に使用する必要があります。 人事管理は

uccxworkforce ユーザによってアクセスが不可能である履歴データ 表にアクセスを必要とすると同時に uccxhruser を使用する必要があります。

- **uccxwallboard** -このユーザは UCCX エンジンのメモリから書かれているリアルタイム統計情報のスナップショットが含まれているリアルタイム データベーステーブルのだけ選択権限があります。 表 RTCSQsSummary および RTICDStatistics に制限 される選択権限は uccxwallboard ユーザがウォールボード アプリケーションによってソースをたどられるように意図されている簡単で、非複雑なクエリを用いる UCCX データベースを頻繁に問い合わせるのに使用する必要があることを意味します。

方法論

UCCX リリース 10.0 と それ以降では、UCCX データベースのパフォーマンス トレースを始めるために **utils uccx データベース dbperf 開始する <totalHours> <interval>** コマンドを入力して下さい。 このコマンドの**間隔** 引数はトレース 収集の周期性を判別し、**totalHours** 引数は無効である前にトレースが実行する時間の総量を判別します。 これらのパラメータはオプションです。 コマンドがデフォルト値 20 分のおよび実行されるとき彼らが規定 されなければ 10 時間は使用されます。

たとえば、データベースのパフォーマンス トレースを有効に し、30 分毎にパフォーマンス統計情報のデータを 24 時間収集するために **24 30 が**命じる **utils uccx データベース dbperf 開始する を**入力して下さい。

CLI コマンドによって得られるデータを収集する指示はコマンド 出力で印刷されます。

```
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:
```

与えられる **totalHours** の後でデータ収集は自動的に停止します。 手動で データ収集を停止するために、**utils uccx データベース dbperf 停止**コマンドを入力して下さい。

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

UCCX バージョンがリリース 9.0(2) またはそれ以前であり、**utils uccx データベース dbperf** コマンドが利用できなかったら、詳細事項に関してはテクニカル アシスタンス センタ （ TAC ）に連絡して下さい。

TAC はリモート サポート アカウント アクセスと Cisco バグ ID CSCuc68413 に手動で接続された dbperf.sh スクリプトを実行します。

スクリプト実行をいつ手動で開始するかまたは判別するかとき CLI コマンドによって、周律および合計時間は、**uccxoninit** プロセスによって消費される CPU をかなり変動するか、または原因解析のための必要な情報を収集するために残りますそれらの期間の間に高く確認します。

dbperf トレース スクリプトによって集められる口グを関連させるために CPU が変動するときさらに、定期的に判別する **show process load コマンドを**入力して下さい。

データ分析

dbperf スクリプトの **onstat** の実行によって集められる口グ**- g SES 0 は** UCCX データベースに対して発行されるアクティブなクエリを示します。 **uccxoninit** プロセスの高CPU は一般的に実行するのに長い 時間をかける複雑なクエリの結果です。 目標はほとんどのリソースを消費し、それらのクエリのための出典 クライアントを判別し、即時解決のためのクライアントからのクエリをディセーブルにし、常置解決のための長期クエリを最適化するクエリを判別することです。

dbperf スクリプトによって集められる口グでは CPU の可能性が高い原因高い変動か **uccxoninit** による支えられた高CPU 消費が処理するクエリを探して下さい。

クエリを疑って下さい:

- **uccxhruser** として接続されるセッションから発行されます-先に記述されていて、**uccxhruser に**設定およびヒストリテーブルの膨大な数から情報を選択する特権があるように。 その結果、複合体は、複数のテーブルを渡る長期クエリ組み立て、UCCX データベースのパフォーマンス影響を持つ場合があります。 絶対が、**uccxwallboard に**および **uccxworkforce** ユーザに UCCX データベース内の表にそのような制限されたアクセスが、これらのユーザが発行するパフォーマンス影響をまずない引き起こす複雑なクエリあります。 さらに、UCCX データベースに対して UCCX 史的記事クライアント（HRC）がまたは Cisco Unified Intelligence Center （CUIC）発行する **uccxhrcare** によって発行されるクエリ。 これらのクエリは静的で、関連した indicies と共に修正され、クエリ パフォーマンスへの最小の影響用の書かれ、テストされ、調整することができません。
- 行って下さいヒストリテーブルの集中的なクエリを- UCCX データベースが実行するように要求するクエリは UCCX データベースに表を渡って倍数、選定された相当数の情報加入するかまたは非指標付けされたフィールドをパフォーマンス影響を引き起こす可能性があります操作します。

uccxhruser として動作する 1 HR 表を含む複雑なクエリを用いる例はここに示されています:

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

上述の例は上のクエリに返される結果があった前に頻繁に入るか、または定期的に入ったら UCCX データベースのパフォーマンス影響を引き起こす可能性があるホスト **WBBOX** からソース

をたどられる uccxhruser によって入る複雑なクエリを示します。

まれが、UCCX データベースのパフォーマンスはまた組み込みパージ プロセスの結果として（uccxoninit プロセスの CPU稼働率は高く変動するか、または残ります）、低下でき。 パージ プロセスはデータベースのサイズを維持するために設定からデータおよび UCCX データベース内のヒストリテーブルを削除するように設計されています。 パージはデータベースの内で含まれているデータベースまたは最も古いレコードのサイズに基づいていましたスケジュールすることができます。

パージ プロセスが動作するとき、データは 1 つのクエリと取除かれます。 取除くことをレコードの量に繰り返して基づいてされません。 これはパージが取除く必要がある多量のデータを検出するこのデータを取除くために一つのクエリを発行することを意味します。

UCCX AppAdminページからのパージ スケジュールまたはパラメータの修正により多量のデータを取除くためにパージをスケジュールするために完了するのに時間をかけますこの単一クエリーは、次にスケジュールされたパージに場合があります。 従って、それはデータベース の インスタンスの CPU稼働率の上で駆動します。

dbperf スクリプトの出力では、パージ クエリは見られる場合があります。 それは sp_purge 保存された手順を呼出すユーザ uccxuser によって入る唯一のクエリであるはずです。

```
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))));
```

一般的な問題

最近の Cisco TAC および Cisco 開発工学エクスペリエンスに基づいて、これらは uccxoninit プロセスの CPU使用率が高い状態を引き起こす最もよく見られる問題です:

- 企業のクライアントは uccxhruser として接続し、ヒストリテーブルとウォールボードかカスタム レポート ソリューションを提供するために加入されるウォールボード表の頻繁な複雑なクエリを（RTICDStatistics および RTCSQsSummary）実行します。 ウォールボード使用に関しては、リアルタイム テーブルにだけ uccxwallboard ユーザおよび制限クエリを使用して下さい。 ウォールボードからのまたはウォールボードと同じような周波数の歴史的かコンフィギュレーションテーブルを問い合わせる機能はサポートされません。
- クライアントはセカンダリ ノードの代りにアクティブなマスター ノードのカスタム史的記事を提出するように試みます。 非エンジン マスター ノードの史的記事を生成する保存された

手順だけを、カスタムかデフォルト実行して下さい。 CUIC および HRC は非マスター ノードのクエリをデフォルトで実行しますが、ときカスタム史的記事を開発して、開発者にこれらのクエリを実行するか、またはこれらの保存された手順を実行するノード選択がある。

- Cisco 人事管理（WFM）は startdatetime フィールドでフィルタリングするように試みるために ContactRoutingDetail 表の複雑なクエリを発行します。 インデックスはこの表のこのフィールドでデフォルトで作成されません、従ってこのクエリのパフォーマンスは粗末です。 WFM 問題定期的にこのクエリ UCCX から WFM にデータを同期するため。 この問題は Cisco バグ ID CSCtz23710 でキャプチャ され、WFM リリース 9.0(1)SR4 で解決されます。 この問題に直面する顧客は Cisco バグ ID CSCtz23710 のための修正が含まれている WFM のバージョンにアップグレードする必要があります。
- パージしきい値は次にスケジュールされたパージが多量のデータを取除くように試みることそのような物修正されます。 よりもむしろかなり単一 アップデートのパージ パラメータを、パージ スケジュール修正繰り返して作られますパージ コンフィギュレーションの変更間の数日と、修正して下さい。 これはパージ プロセスが削除オペレーションのパフォーマンスを改善する各パスのより小さいデータセットを取除くようにします。
- DialingList 表は膨大です。 DialingList 表は送信キャンペーンにアップロードされるすべての連絡先を保存します。 UCCX リリース 8.0 および 8.5 では、数百万のレコードが送信キャンペーンにアップ ロードされた後、パフォーマンス上の問題は表それから（uccxoninit プロセスおよび遅い AppAdmin ページ ロードの高CPU を引き起こす）問い合わせられます生じます。 パフォーマンス上の問題を軽減するために、DialingList 表をクリーンアップする cron ジョブ スクリプトのインストールのための TAC ケースをオープンして下さい。 UCCX リリース 9.0 ではパフォーマンスを改善するために、インデックスは AppAdmin からより有効なクエリのためのこの表に追加されました。 この変更はすべてほとんどの極端なケースの問題を解決しました。 この問題に広範囲の修正を提供する歴史的連絡先のアクティブな連絡先のための UCCX リリース 10.0 DialingList は 2 つの表に、1 および別のものでは分割されました。