

Analyse des Algorithmus zur Bestimmung des Mastering in UCCX-Failover-Szenarien

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Hintergrundinformationen](#)

[Normaler Engine-Failover](#)

[Wiederherstellung im Inselmodus](#)

Einführung

In diesem Dokument wird der Algorithmus beschrieben, der zur Bestimmung der Beherrschung verwendet wird, nachdem der Failover- oder Recovery-Modus in Unified Contact Center Express (UCCX) initiiert wurde.

Voraussetzungen

Anforderungen

Cisco empfiehlt, über Kenntnisse in folgenden Bereichen zu verfügen:

- UCCX
- Failover-Mechanismus

Verwendete Komponenten

Dieses Dokument ist nicht auf bestimmte Software- und Hardwareversionen beschränkt.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Hintergrundinformationen

Es sind zwei Szenarien möglich, in denen ein Master ausgewählt werden muss, und der Algorithmus, der zur Bestimmung der Meisterschaft zwischen den beiden Knoten verwendet wird, ist für jedes Szenario unterschiedlich.

Normaler Engine-Failover

Dieses Szenario tritt auf, wenn entweder kein Master vorhanden ist (z. B. wenn ein Cluster gestartet wird) oder ein Master vorhanden ist, wenn Knoten in einer HAoLAN- oder HAoWAN-Umgebung ohne Inselmodus Failover-Vorgänge durchführen.

Der Algorithmus übernimmt die Wasserfälle, um die Meisterschaft zu bestimmen (d. h. versuchen Sie 1, andernfalls geben Sie 2 weitere Try 3 ein, andernfalls versuchen Sie 4 im Streitfall):

Schritt 1: Ermitteln Sie den UCCX-Modulstatus beider Knoten, je nachdem, welcher Status besser ist - das ist der neue Master. Wenn beide identisch sind, fahren Sie mit Schritt 2 fort.

Schritt 2: Bestimmen Sie das Hardwaremodell der beiden Knoten. Die bessere Hardware ist der neue Master. Wenn beide identisch sind, fahren Sie mit Schritt fort. Da viele UCCX-Installationen jetzt virtuell sind, wird dieser Schritt nicht häufig verwendet.

Schritt 3: Bestimmen Sie den Node 1, d. h. Publisher (der erste installierte UCCX-Knoten). Der neue Master ist der Publisher-Knoten. CVD ist so programmiert, dass Knoten 1 als Standard-Master festgelegt wird. Dies wird aus dem PrimaryEngineComputerName-Parameter in der Clusterkonfiguration (ClusterSpecificConfig) im CET übernommen. Wenn dieser Wert nicht korrekt ist, übernimmt Node2 immer die Meisterschaft. Siehe: [CSCuw95068](#).

Schritt 4: Wenn in Schritt 3 der richtige Hostname des Herausgebers nicht bestimmt werden kann, sollte Knoten 2 als Master (Subscriber) festgelegt werden.

Die Logik lautet:

Schritt 1: Überprüfen Sie den Dienststatus der Knoten. Wenn Knoten 1 IN_SERVICE und Knoten 2 in PARTIAL_SERVICE ist, wird Knoten 1 zum Master. Wenn die Zustände identisch sind (IN_SERVICE oder PARTIAL_SERVICE), fahren Sie bei SCHRITT 2 fort.

Schritt 2: Die Hardwarespezifikation der 2 UCCX-Knoten wird überprüft. Der Server mit der besseren Spezifikation wird beherrscht. Wenn die Hardware-Spezifikationen identisch sind, fahren Sie bei SCHRITT 3 fort.

Schritt 3: Der PUBLISHER wird zum MASTER, wenn der Hostname des PUBLISHER mit dem PrimaryEngineComputerName im CET (ClusterSpecificConfig) übereinstimmt. Wenn keine MATCH vorliegt, fahren Sie mit Schritt 4 fort.

Schritt 4: Master für Teilnehmer festlegen, wenn der obige Schritt fehlschlägt.

Wiederherstellung im Inselmodus

Dieses Szenario tritt während der Wiederherstellung aus dem Inselmodus auf, wenn es zwei Master gibt. In diesem Fall wird der oben genannte Algorithmus nicht ausgeführt. Vielmehr behält der UCCX Publisher-Knoten (der erste installierte UCCX-Knoten) die Master-Rolle bei, und der Subscriber verwirft die Meisterschaft.

Hinweis: Wichtig ist, dass der Hostname des primären Knotens mit dem Eintrag PrimaryEngineComputerName im ClusterSpecificConfig-Objekt übereinstimmen muss. Andernfalls wird der sekundäre Knoten als Master ausgewählt. Verwenden Sie das CET-Tool, um eine Verbindung zum primären Knoten herzustellen, um zu überprüfen, ob der

Eintrag korrekt ist, und um ihn ggf. zu ändern.

Wenn das System überprüft, welcher Knoten den besseren Service-Status hat, wie in Schritt 1 erwähnt, können Sie auf diese Weise die ausgewählten Services überprüfen.

- Engine-Service
- Manager Manager-Komponente in Engine

Wenn beide Dienste IN_SERVICE sind, gilt dieser Knoten als Meisterschaft.

Dies ist ein Ausschnitt aus den Protokollen, in denen der Algorithmus verwendet wird, um das Szenario zu erklären: In diesem Szenario wurde Knoten 1 vor einem WAN-Ausfall als Master bezeichnet. Als das WAN zurückkam, wurde Knoten 2 zum MASTER.

Wenn die WAN-Verbindung ausfiel:

Zunächst waren beide Knoten Master. Knoten 1 war der Master; Knoten 2 wurde auch Master:

Cisco Unified CCX Engine auf Knoten 2 ändert Master von false in true

```
3162: Dec 15 12:41:17.607 IST %MCVD-CLUSTER_MGR-7-UNK:JavaService167
```

Dies ist auch der Zeitpunkt, an dem der Knoten einen Absturz des anderen Knotens vermutet:

```
3111: Dec 15 12:41:17.481 IST %MCVD-CVD-4-HEARTBEAT_SUSPECT_NODE_CRASH:CVD suspects node crash:
state=Heartbeat State,nodeInfo=Node id=1 ip=172.30.72.2 convId=69 cmd=16 viewLen=1,dt=1022
```

Als die WAN-Verbindung zurückkam und KONVERGENZ begann:

```
9777: Dec 15 12:42:28.859 IST %MCVD-CVD-4-MASTER_DETECTS_NODE_JOIN:More than one master
detected, when processing node join: name=Cisco Unified CCX Database,nodeId=2,masterCnt=1
9778: Dec 15 12:42:28.859 IST %MCVD-CVD-7-UNK:Split after network partition is detected, new
nodeId=2
```

```
Node id=002, addresses=[172.30.83.2], MAC addresses=[279f2d5ba86d], compName=UCCXSUB, state=IN
SERVICE, en=true, rmiPort=6999, masterPort=1994
```

```
VersionInfo: [ Version=8.5.1.11003-32, crsRelease=8.5.1.11003-32, crsServiceRelease=,
crsEngineeringSpecial=, dbEdition=IDS, dbVersion=V11, installTime=1348139852000,
upgradeTime=1348139852000, jtapiClientVersion=8.6(2.10000)-2 ]
```

```
cT=969, uT=969, rT=528, serVer=3, cvdVer=3, points=0
```

```
Component201: type=CRS Historical Datastore, state=IN SERVICE, en=true, prim=false, node=002,
activationTime=1348141153000, parent=null, uT=492, rT=193, rootDir /opt/cisco/uccx,
version=8.5.1.11003-32, serVer=1
```

```
Service163: name=Cisco Unified CCX Database, Feature Service, isActivationSupported=false,
node=002, state=IN SERVICE, master, parent=null, type=DB Services, logDir:
/common/informix/crs/???, en=true, uT=928, rT=0, version=8.5.1.11003-32, serVer=4
```

```
Component202: type=Cisco Recording, state=IN SERVICE, en=true, prim=false, node=002,
activationTime=1348140987000, parent=null, uT=439, rT=198, rootDir /opt/cisco/uccx,
version=8.5.1.11003-32, serVer=1
```

```
9823: Dec 15 12:42:38.866 IST %MCVD-CLUSTER_MGR-7-UNK:Post Convergence Event:
CONVERGENCE_STARTED, name=Cisco Unified CCX Engine
```

```
9824: Dec 15 12:42:38.866 IST %MCVD-CLUSTER_MGR-7-UNK:Cl Mgr: Cisco Unified CCX Engine
Convergence Started
9825: Dec 15 12:42:38.866 IST %MCVD-CLUSTER_MGR-7-UNK:try to process
MasterConvergenceCompletedCmdImpl: name Cisco Unified CCX Engine, nodeId=1, type=MASTER_DROPPED,
uniqueId=66, master=false, updateTick=3101, baseTick=3100, nodeCurrentTick=3101
9826: Dec 15 12:42:38.866 IST %MCVD-CLUSTER_MGR-7-UNK:process MasterConvergenceCompletedCmdImpl:
name Cisco Unified CCX Engine, nodeId=1, type=MASTER_DROPPED, uniqueId=66, master=false,
updateTick=3101, baseTick=3100, nodeCurrentTick=3101
9827: Dec 15 12:42:38.866 IST %MCVD-CLUSTER_MGR-7-UNK:JavaService66: Cisco Unified CCX Engine on
node 1 change master from true to false
```

Zu diesem Zeitpunkt begann die Konvergenz. Daher wird der zuvor erläuterte Algorithmus zur Auswahl des Masters verwendet. Beachten Sie hier die Zustände beider Knoten:

```
Node id=001, addresses=[172.30.72.2], MAC addresses=[95eab6e4c4cb], compName=UCCXPUB,
state=PARTIAL SERVICE, en=true, rmiPort=6999, masterPort=1994
  VersionInfo: [ Version=8.5.1.11003-32, crsRelease=8.5.1.11003-32, crsServiceRelease=,
crsEngineeringSpecial=, dbEdition=IDS, dbVersion=V11, installTime=1348064353000,
upgradeTime=1348064353000, jtapiClientVersion=8.6(2.10000)-2 ]
  cT=3275, uT=3275, rT=534, serVer=3, cvdVer=3, points=0

Node id=002, addresses=[172.30.83.2], MAC addresses=[279f2d5ba86d], compName=UCCXSUB, state=IN
SERVICE, en=true, rmiPort=6999, masterPort=1994
  VersionInfo: [ Version=8.5.1.11003-32, crsRelease=8.5.1.11003-32, crsServiceRelease=,
crsEngineeringSpecial=, dbEdition=IDS, dbVersion=V11, installTime=1348139852000,
upgradeTime=1348139852000, jtapiClientVersion=8.6(2.10000)-2 ]
  cT=969, uT=969, rT=528, serVer=3, cvdVer=3, points=0
```

Daher wird die Meisterschaft nach dem Algorithmus an Node 2 übergeben (Punkt 1 im Algorithmus). Dies erklärt, warum UCCX Node 2 nach der Konvergenz Master wurde.

Sie müssen jedoch überprüfen, warum der Knoten 1 teilweise verwendet wurde. Der Dienst wurde teilweise aufgrund des Telefonie-Subsystems ausgeführt:

```
name=Unified CM Telephony Subsystem, Feature Service, isActivationSupported=false, node=001,
state=PARTIAL SERVICE
```