

瞭解UCCX故障切換場景中確定主幹能力的演算法

目錄

[簡介](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[正常引擎故障轉移](#)

[島式模式復原](#)

簡介

本文檔介紹在Unified Contact Center Express(UCCX)中啟動從島式模式進行故障切換或恢復後，用於確定主幹的演算法。

必要條件

需求

思科建議您瞭解以下主題：

- UCCX
- 故障切換機制

採用元件

本文件所述內容不限於特定軟體和硬體版本。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

兩個必須選擇主節點的方案是可能的，並且每個方案用於確定兩個節點之間的主節點的演算法不同。

正常引擎故障轉移

在沒有島式模式的HAoLAN或HAoWAN環境中，當沒有主節點時（例如啟動集群時）或只有一個主節點時，就會發生這種情況。

演算法會進行瀑布化以確定主控（即，若發生爭用，則嘗試1，否則嘗試2，否則嘗試3，否則嘗試4）：

步驟1. 確定兩個節點的UCCX引擎狀態，以處於較佳狀態（即新主機）為準。如果兩者相同，則轉到步驟2。

步驟2. 確定兩個節點的硬體型號。新的主裝置是更好的硬體。如果兩者都相同，請轉到步驟。由於現在許多UCCX安裝都是虛擬的，因此通常不使用此步驟。

步驟3. 確定節點1（即Publisher）（已安裝第一個UCCX節點）。新的主節點是發佈器節點。CVD被程式設計為使節點1成為預設主節點。此引數取自CET上群集配置(ClusterSpecificConfig)中的PrimaryEngineComputerName引數。如果此值不正確，Node2始終獲取主控。請參閱：[CSCuw95068](#)。

步驟4. 如果步驟3不能確定Publisher的正確主機名，請將節點2設定為主（訂戶）。

其邏輯是：

步驟1. 檢查節點的服務狀態。如果節點1為IN_SERVICE，而節點2為PARTIAL_SERVICE，則節點1成為主節點。如果狀態相同（IN_SERVICE或PARTIAL_SERVICE），請轉到步驟2。

步驟2. 檢查2個UCCX節點的硬體規格。具有良好規格的伺服器被授予主控。如果硬體規格相同，請轉至步驟3。

步驟3. 如果PUBLISHER的主機名與CET(ClusterSpecificConfig)上的PrimaryEngineComputerName匹配，則PUBLISHER將成為MASTER。如果沒有MATCH，請轉到步驟4。

步驟4. 如果上述步驟失敗，則使訂戶成為主使用者。

島式模式復原

當有兩個主節點時，從島式模式進行恢復時遇到此情況。發生這種情況時，不會執行上述演算法。相反，UCCX發佈伺服器節點（安裝的第一個UCCX節點）保留主控，而使用者放棄主控。

附註：需要注意的重要一點是，主節點的主機名必須與ClusterSpecificConfig對象中的PrimaryEngineComputerName條目匹配。否則，會選擇輔助節點作為主節點。使用CET工具連線到主節點，檢查條目是否正確，並在必要時進行更改。

此外，當系統檢查哪個節點處於較佳服務狀態時（如步驟1所述），這是檢查所檢查的特定服務的方式

- 引擎服務
- 引擎中的Manager Manager元件

如果這兩個服務都是IN_SERVICE，則此節點被視為主控。

以下是日誌中的一個片段，其中演算法用於解釋場景：在此案例中，節點1在WAN中斷之前被通知為主節點；當廣域網恢復時，節點2成為主節點。

當WAN鏈路斷開時：

首先，兩個節點都是主節點。節點1是主節點；節點2也成為了主節點：

節點2上的Cisco Unified CCX Engine主節點從false更改為true

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

這也是節點懷疑另一個節點崩潰的時間：

```
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
```

當WAN鏈路恢復並開始融合時：

```
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
```

這是收斂開始的時間。因此，前面解釋的演算法用於選擇主機。請注意兩個節點的狀態：

```
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
```

因此，通過演算法，主節點被傳遞給節點2（演算法中的點1）。這解釋了為什麼在收斂之後UCCX節點2成為主節點。

但是，您必須檢查節點1處於部分服務狀態的原因。由於電話子系統，它處於部分服務中：

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