

# Comprendere l'algoritmo per determinare la padronanza negli scenari di failover UCCX

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Failover normale del motore](#)

[Modalità ripristino isola](#)

## Introduzione

Questo documento descrive l'algoritmo utilizzato per determinare la masterizzazione dopo l'avvio del failover o del ripristino dalla modalità isola in Unified Contact Center Express (UCCX).

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- UCCX
- Meccanismo di failover

### Componenti usati

Il documento può essere consultato per tutte le versioni software o hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

## Premesse

Sono possibili due scenari in cui è necessario selezionare un master e l'algoritmo utilizzato per determinare la padronanza tra i due nodi è diverso per ogni scenario.

## Failover normale del motore

Questo scenario si verifica quando non è disponibile un dispositivo master, ad esempio quando

viene avviato un cluster, oppure quando è disponibile un dispositivo master, quando i nodi vengono sottoposti a failover in un ambiente HAoLAN o HAoWAN senza modalità isola.

Le cascate dell'algoritmo per determinare la padronanza (ad esempio, Prova 1, altrimenti Prova 2 altro Prova 3, altrimenti Prova 4 in caso di contesa):

Passaggio 1. Determinare lo stato del motore UCCX di entrambi i nodi, a seconda di quale dei due si trova nello stato migliore, ossia il nuovo master. Se entrambi sono uguali, passare al passaggio 2.

Passaggio 2. Determinare il modello hardware dei due nodi. L'hardware migliore è il nuovo master. Se sono uguali, passare al punto . Poiché molte installazioni UCCX sono ora virtuali, questo passaggio non viene spesso utilizzato.

Passaggio 3. Determinare il nodo 1, ovvero Publisher (il primo nodo UCCX installato). Il nuovo master è il nodo Publisher. CVD è programmato in modo da impostare il Nodo 1 come master predefinito. Questa è presa dal parametro PrimaryEngineComputerName nella configurazione del cluster (ClusterSpecificConfig) sul CET. Se questo valore non è corretto, Node2 acquisisce sempre la padronanza. Riferimento: [CSCuw95068](#)

Passaggio 4. Se il passaggio 3 non è in grado di determinare il nome host corretto del server di pubblicazione, impostare il nodo 2 come master (Sottoscrittore).

La logica è:

Passaggio 1. Controllare lo stato del servizio dei nodi. Se il nodo 1 è IN\_SERVICE e il nodo 2 è in PARTIAL\_SERVICE, il nodo 1 diventa il master. Se gli stati sono gli stessi (IN\_SERVICE o PARTIAL\_SERVICE), andare al PASSO 2.

Passaggio 2. Viene controllata la specifica hardware dei 2 nodi UCCX. Il server con le specifiche migliori si è trasformato nella padronanza. Se le specifiche hardware sono le stesse, andare al PASSO 3.

Passaggio 3. L'EDITORE diventa il MASTER se il nome host dell'EDITORE corrisponde all'oggetto PrimaryEngineComputerName dell'oggetto CET (ClusterSpecificConfig). Se non è presente MATCH, andare al punto 4.

Passaggio 4. Rendere il Sottoscrittore master se il passaggio precedente ha esito negativo.

## Modalità ripristino isola

Questo scenario si verifica durante il ripristino dalla modalità isola quando sono presenti due schemi. In questo caso, l'algoritmo precedente non viene eseguito. Al contrario, il nodo UCCX Publisher (il primo nodo UCCX installato) conserva la masterizzazione e il sottoscrittore la elimina.

**Nota:** È importante notare che il nome host del nodo primario deve corrispondere alla voce PrimaryEngineComputerName nell'oggetto ClusterSpecificConfig. In caso contrario, il nodo secondario viene scelto come master. Utilizzate lo strumento CET per collegarvi al nodo primario per verificare se la voce è corretta e modificarla se necessario.

Inoltre, quando il sistema controlla quale nodo si trova nello stato di servizio migliore, come indicato nel passaggio 1, questo è il modo per controllare i servizi specifici controllati

- Servizio motore
- Componente Manager nel motore

Se entrambi questi servizi sono IN\_SERVICE, questo nodo viene considerato per la masterizzazione.

Questo è un frammento dei log in cui l'algoritmo viene utilizzato per spiegare lo scenario: In questo scenario, al nodo 1 è stato assegnato il ruolo di master prima dell'interruzione della WAN; e quando la WAN è tornata, il Nodo 2 è diventato il MASTER.

Quando il collegamento WAN si interrompe:

Per prima cosa, entrambi i nodi erano Master. Il nodo 1 era il Master; Anche il nodo 2 è diventato il master:

Cisco Unified CCX Engine sul nodo 2 cambia il master da false a true

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

Questo è anche il momento in cui il nodo sospetta un arresto anomalo dell'altro nodo:

```
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
```

Quando il collegamento WAN è tornato e la CONVERGENZA è iniziata:

```
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
```

A questo punto è iniziata la convergenza. Pertanto, l'algoritmo illustrato in precedenza viene utilizzato per selezionare il master. Osservare qui gli stati di entrambi i nodi:

```
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
```

Pertanto, in base all'algoritmo, la padronanza viene consegnata al Nodo 2 (punto 1 nell'algoritmo). Questo spiega perché il nodo UCCX 2 è diventato il master dopo la convergenza.

Tuttavia, è necessario verificare il motivo per cui il Nodo 1 era in servizio parziale. Era in servizio parziale a causa del sottosistema Telefonia:

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