Comprendre l'algorithme permettant de déterminer la maîtrise dans les scénarios de basculement UCCX

Contenu

[Introduction](#)
[Conditions préalables](#)
[Conditions requises](#)
[Components Used](#)
[Informations générales](#)
[Basculement normal du moteur](#)
[Récupération en mode île](#)

Introduction

Ce document décrit l'algorithme utilisé pour déterminer la maîtrise après le démarrage du basculement ou de la récupération à partir du mode île dans Unified Contact Center Express (UCCX).

Conditions préalables

Conditions requises

Cisco vous recommande de prendre connaissance des rubriques suivantes :

- UCCX
- Mécanisme de basculement

Components Used

Ce document n'est pas limité à des versions de matériel et de logiciel spécifiques.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l’incidence possible des commandes.

Informations générales

Deux scénarios dans lesquels un maître doit être sélectionné sont possibles et l'algorithme utilisé pour déterminer la maîtrise entre les deux noeuds est différent pour chaque scénario.

Basculement normal du moteur

Ce scénario se produit lorsqu'il n'y a pas de maître (par exemple lorsqu'un cluster est démarré) ou qu'il y a un maître, lorsque les noeuds sont basculés dans un environnement HAoLAN ou HAoWAN sans mode île.

L'algorithme cascade pour déterminer la maîtrise (c'est-à-dire Try 1, else Try 2 else Try 3, else Try 4 in case contention) :

Étape 1. Déterminez l'état du moteur UCCX des deux noeuds, selon celui qui est le mieux en état, c'est-à-dire le nouveau maître. Si les deux sont identiques, passez à l'étape 2.

Étape 2. Déterminez le modèle matériel des deux noeuds. Le meilleur matériel est le nouveau maître. Si les deux sont identiques, passez à l'étape . Étant donné que de nombreuses installations UCCX sont désormais virtuelles, cette étape n'est pas souvent utilisée.

Étape 3. Déterminez le noeud 1, à savoir Publisher (le premier noeud UCCX installé). Le nouveau maître est le noeud Publisher. CVD est programmé pour faire du noeud 1 le maître par défaut. Ceci est extrait du paramètre PrimaryEngineComputerName dans la configuration de cluster (ClusterSpecificConfig) sur CET. Si cette valeur est incorrecte, Node2 prend toujours la maîtrise. Reportez-vous à : CSCuw95068.

Étape 4. Si l'étape 3 ne parvient pas à déterminer le nom d'hôte correct de Publisher, définissez Node 2 comme maître (Abonné).

La logique est la suivante :

Étape 1. Vérifiez l'état du service des noeuds. Si le noeud 1 est IN_SERVICE et le noeud 2 est dans PARTIAL_SERVICE, le noeud 1 devient le maître. Si les états sont identiques (IN_SERVICE ou PARTIAL_SERVICE), passez à l'ÉTAPE 2.

Étape 2. La spécification matérielle des 2 noeuds UCCX est vérifiée. Le serveur avec la meilleure spécification est remis au maître. Si les spécifications matérielles sont identiques, passez à l'ÉTAPE 3.

Étape 3. Le PUBLISHER devient MASTER si le nom d'hôte du PUBLISHER correspond au PrimaryEngineComputerName sur CET (ClusterSpecificConfig). S'il n'y a pas de CORRESPONDANCE, passez à l'étape 4.

Étape 4. Créer le maître d'abonné si l'étape ci-dessus échoue.

Récupération en mode île

Ce scénario est rencontré lors de la récupération à partir du mode île lorsqu'il y a deux maîtres. Dans ce cas, l'algorithme ci-dessus n'est pas exécuté. Au contraire, le noeud UCCX Publisher (le premier noeud UCCX installé) conserve la maîtrise et l'abonné abandonne la maîtrise.

> **Note**: Il est important de noter que le nom d'hôte du noeud principal doit correspondre à l'entrée PrimaryEngineComputerName de l'objet ClusterSpecificConfig. Sinon, le noeud secondaire est choisi comme maître. Utilisez l'outil CET pour vous connecter au noeud principal afin de vérifier si l'entrée est correcte et de la modifier si nécessaire.

En outre, lorsque le système vérifie quel noeud est en meilleur état de service, comme indiqué à l'étape 1, c'est la manière de vérifier les services spécifiques vérifiés

- Service du moteur
- Composant Manager Manager dans le moteur

Si ces deux services sont IN_SERVICE, ce noeud est pris en compte pour la maîtrise.

Voici un extrait des journaux dans lesquels l'algorithme est utilisé pour expliquer le scénario : Dans ce scénario, le noeud 1 devait être le maître avant une panne WAN ; et lorsque le WAN est revenu, le noeud 2 est devenu le MASTER.

Lorsque la liaison WAN est tombée en panne :

Premièrement, les deux noeuds étaient Master. Le noeud 1 était le maître ; Le noeud 2 est également devenu le maître :

Cisco Unified CCX Engine sur le noeud 2 passe de false à true

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

C'est également le moment où le noeud suspecte un plantage de l'autre noeud :

```
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
```

Lorsque la liaison WAN est revenue et que CONVERGENCE a démarré :

```
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
```

C'est à ce moment que la convergence a commencé. Par conséquent, l'algorithme expliqué précédemment est utilisé pour sélectionner le maître. Ici, notez les états des deux noeuds :

```
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
```

Par conséquent, en se basant sur l'algorithme, le maître est remis au noeud 2 (point 1 de l'algorithme). Ceci explique pourquoi le noeud UCCX 2 est devenu le maître après la convergence.

Cependant, vous devez vérifier pourquoi le noeud 1 était en service partiel. Il était en service partiel en raison du sous-système de téléphonie :

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