

UCCX 장애 조치 시나리오에서 마스터를 결정하는 알고리즘 이해

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[일반 엔진 장애 조치](#)

[아일랜드 모드 복구](#)

소개

이 문서에서는 UCCX(Unified Contact Center Express)에서 장애 조치 또는 섬 모드에서 복구가 시작된 후 마스터를 확인하는 데 사용되는 알고리즘에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- UCCX
- 장애 조치 메커니즘

사용되는 구성 요소

이 문서는 특정 소프트웨어 및 하드웨어 버전으로 한정되지 않습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

마스터를 선택해야 하는 두 가지 시나리오가 가능하며, 두 노드 간의 마스터를 결정하는 데 사용되는 알고리즘은 각 시나리오에 대해 서로 다릅니다.

일반 엔진 장애 조치

이 시나리오는 마스터가 없거나(예: 클러스터가 시작된 경우) 마스터가 하나 있는 경우, 섬 모드가 없는 HAoLAN 또는 HAoWAN 환경에서 노드가 페일오버되는 경우 발생합니다.

마스터를 결정하기 위한 알고리즘 폭포(예: Try 1, else 2 Try 3, 그렇지 않으면 Try 4(경합 시)):

1단계. 두 노드의 UCCX 엔진 상태 중 더 나은 상태(새 마스터인 상태)를 확인합니다. 둘 다 같으면 2단계로 이동합니다.

2단계. 두 노드의 하드웨어 모델을 확인합니다. 새 마스터는 더 나은 하드웨어입니다. 둘 다 같으면 단계로 이동합니다. 이제 많은 UCCX 설치가 가상화되었으므로 이 단계는 자주 사용되지 않습니다.

3단계. 노드 1(예: Publisher(첫 번째 UCCX 노드가 설치됨))을 확인합니다. 새 마스터는 게시자 노드입니다. CVD는 노드 1을 기본 마스터로 만들도록 프로그래밍됩니다. 이는 CET의 클러스터 구성 (ClusterSpecificConfig)에 있는 PrimaryEngineComputerName 매개 변수에서 가져온 것입니다. 이 값이 올바르지 않으면 Node2는 항상 마스터를 사용합니다. 참조: [CSCuw95068](https://cscuw95068).

4단계. 3단계에서 Publisher의 올바른 호스트 이름을 확인할 수 없는 경우 노드 2를 마스터(가입자)로 설정합니다.

논리는 다음과 같습니다.

1단계. 노드의 서비스 상태를 확인합니다. 노드 1이 IN_SERVICE이고 노드 2가 PARTIAL_SERVICE에 있으면 노드 1이 마스터가 됩니다. 상태가 동일한 경우(IN_SERVICE 또는 PARTIAL_SERVICE) 2단계로 이동합니다.

2단계. 2 UCCX 노드의 하드웨어 사양이 선택되어 있습니다. 더 나은 사양을 가진 서버는 마스터를 전달합니다. 하드웨어 사양이 동일한 경우 3단계로 이동합니다.

3단계. PUBLISHER의 호스트 이름이 CET(ClusterSpecificConfig)의 PrimaryEngineComputerName과 일치하면 PUBLISHER가 MASTER가 됩니다. MATCH가 없으면 4단계로 이동합니다.

4단계. 위 단계가 실패할 경우 가입자 마스터로 설정합니다.

아일랜드 모드 복구

이 시나리오는 두 마스터가 있을 때 섬 모드에서 복구하는 동안 발생합니다. 이 경우 위의 알고리즘은 실행되지 않습니다. 대신 UCCX 게시자 노드(설치된 첫 번째 UCCX 노드)는 마스터를 유지하고 가입자가 마스터를 삭제합니다.

참고: 중요한 점은 기본 노드의 호스트 이름이 ClusterSpecificConfig 개체의 PrimaryEngineComputerName 항목과 일치해야 한다는 것입니다. 그렇지 않으면 보조 노드가 마스터로 선택됩니다. CET 툴을 사용하여 기본 노드에 연결하여 항목이 올바른지 확인하고 필요한 경우 변경합니다.

또한 1단계에서 설명한 것처럼 시스템에서 어떤 노드가 더 나은 서비스 상태에 있는지 확인할 때 이는 선택된 특정 서비스를 확인하는 방법입니다

- 엔진 서비스
- 엔진 내의 Manager Manager 구성 요소

이 두 서비스가 모두 IN_SERVICE인 경우 이 노드는 mastership으로 간주됩니다.

시나리오를 설명하는 데 알고리즘을 사용하는 로그의 조각입니다. 이 시나리오에서는 WAN 중단 전에 노드 1이 마스터가 되도록 했습니다. WAN이 돌아왔을 때 노드 2가 MASTER가 되었습니다.

WAN 링크가 다운된 경우:

첫째, 두 노드는 모두 마스터였습니다. 노드 1은 마스터였습니다. 노드 2도 마스터가 되었습니다.

노드 2의 Cisco Unified CCX Engine은 마스터를 false에서 true로 변경합니다.

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

또한 노드가 다른 노드의 충돌을 의심하는 시간입니다.

```
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
```

WAN 링크가 다시 시작되고 CONVERGENCE가 시작된 경우:

```
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
```

컨버전스가 시작된 때입니다. 따라서 앞에서 설명한 알고리즘은 마스터를 선택하는 데 사용됩니다. 여기에서 두 노드의 상태를 확인합니다.

```
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
```

따라서 알고리즘을 통해 마스터는 노드 2(알고리즘의 포인트 1)에 전달됩니다. 통합 후 UCCX 노드 2가 마스터가 된 이유를 설명합니다.

그러나 노드 1이 부분 서비스에 있는 이유를 확인해야 합니다. 텔레포니 하위 시스템으로 인해 부분 서비스에 있었습니다.

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