

了解确定UCCX故障切换场景中主机的算法

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[正常引擎故障切换](#)

[岛模式恢复](#)

简介

本文档介绍在统一联系中心快捷版(UCCX)中启动故障切换或从岛模式恢复后确定主机的算法。

先决条件

要求

Cisco 建议您了解以下主题：

- UCCX
- 故障转移机制

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

必须选择主节点的两种情况是可能的，用于确定两个节点之间主节点的算法对于每种情况都不同。

正常引擎故障切换

当没有主设备（例如启动集群时）或有一个主设备时，当节点在没有岛模式的HAoLAN或HAoWAN环境中进行故障切换时，会遇到此场景。

确定主题的算法瀑布(即Try 1，否则Try 2，否则Try 3，否则Try 4（在争用情况下））：

步骤1.确定两个节点的UCCX引擎状态（以处于更好状态为准）— 即新主节点。如果两者都相同，则转到步骤2。

第二步：确定两个节点的硬件型号。更好的硬件是新主设备。如果两者都相同，请移至步骤。由于许多UCCX安装现在都是虚拟的，因此不常使用此步骤。

步骤3.确定节点1，即发布服务器（安装的第一个UCCX节点）。新主节点是发布者节点。CVD编程为使节点1成为默认主设备。这取自CET上集群配置(ClusterSpecificConfig)中的PrimaryEngineComputerName参数。如果此值不正确，Node2始终成为主节点。请参阅：[CSCuw95068。](#)

步骤4.如果步骤3无法确定发布服务器的正确主机名，请将节点2设置为主节点（订用服务器）。

其逻辑是：

步骤1.检查节点的服务状态。如果节点1为IN_SERVICE，而节点2为PARTIAL_SERVICE，则节点1成为主节点。如果状态相同（IN_SERVICE或PARTIAL_SERVICE），请转至步骤2。

步骤2.检查2个UCCX节点的硬件规格。将具有更好规格的服务器交给主设备。如果硬件规格相同，请转至步骤3。

步骤3.如果PUBLISHER的主机名与CET(ClusterSpecificConfig)上的PrimaryEngineComputerName匹配，则PUBLISHER将成为主服务器。如果没有MATCH，请转至步骤4。

步骤4.如果上述步骤失败，则使用户成为主用户。

岛模式恢复

当有两个主设备时，从岛模式恢复时会遇到此场景。发生这种情况时，不执行上述算法。相反，UCCX发布服务器节点（安装的第一个UCCX节点）保留主节点，用户放弃主节点。

注意：需要注意的一点是，主节点的主机名必须与ClusterSpecificConfig对象中的PrimaryEngineComputerName条目匹配。否则，辅助节点将被选为主节点。使用CET工具连接到主节点以检查条目是否正确，并在必要时进行更改。

此外，如步骤1所述，当系统检查哪个节点处于更好的服务状态时，这是检查检查特定服务的方式

- 引擎服务
- 引擎中的Manager Manager组件

如果这两个服务都是IN_SERVICE，则此节点被视为主设备。

这是日志中的代码段，其中使用算法来解释场景：在此场景中，节点1在广域网中断之前被告知是主节点；当WAN恢复时，节点2成为主节点。

当WAN链路断开时：

首先，两个节点都是主节点。节点1是主节点；节点2也成为主节点：

节点2上的Cisco Unified CCX引擎将主节点从false更改为true

这也是节点怀疑另一个节点崩溃的时间：

```
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
```

当WAN链路恢复并开始融合时：

```
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
```

此时即开始融合。因此，使用前面介绍的算法来选择主设备。请注意两个节点的状态：

```
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，通过算法，主节点被交给节点2（算法中的第1点）。这解释了为什么UCCX节点2在融合后成为主节点。

但是，您必须检查节点1为什么处于部分服务状态。由于电话子系统，它处于部分服务状态：

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