

排除CVP礼貌回拨(CCB)网关容量验证故障故障

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[背景信息](#)

[症状](#)

[Troubleshoot](#)

[解决方案](#)

[永久解决方案](#)

[测试结果解决方案](#)

Introduction

本文描述如何排除Customer Voice Portal (CVP) CCB问题故障，当呼叫人没获得CCB提供时，因为Trunk网关容量超出了。

Prerequisites

Requirements

Cisco 建议您了解以下主题：

- CVP
- Cisco CVP礼貌回拨

Components Used

本文档中的信息基于以下软件版本：

- CVP服务器10.5
- 统一的联系中心企业(UCCE) 10.5

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. If your network is live, make sure that you understand the potential impact of any command.

背景信息

在排除故障前网关容量问题，了解在CCB的Trunk验证过程是重要的。基本上，进程首先确定呼叫的数量从与EventTypeID的Callback_current表(21,22,23)；待定，Inprogress，试探性特定网关和位置的。

其次，从同一张Callback_current表，请确定，呼叫的数量完成与被连接的原因： EventTypeID = 24 (完成)和CauseID = 27 (连接)。

最终进程添加这两值并且和Trunk相比的数量被配置在Survivability.tcl服务下。

如果结果在被配置的Trunk阈值，进程退还一个故障(回归1)，否则退还ok (回归0)。

总之，验证Trunk的公式用于CCB是：

CCB建立中继< (与EventTypeID的Callback_current表(21,22,23);待定， Inprogress， 试探性特定网关的) + EventTypeID Callback_current表= 24 (完成)和CauseID = 27 (连接)

如果CCB Trunk值是更低的验证发生故障。

症状

一次入局呼叫没获得CCB提供。呼叫直接地去不管怎么样排队估计的等待时间(EWT)

Troubleshoot

步骤1.从语音扩展标记语言(VXML)服务器的CallbackEntry应用程序收集活动日志。

步骤2.在活动日志内搜索验证是无的所有呼叫：

```
Validate_02,data,result,none
```

哪些意味着验证没有通过。获得此呼叫的GUID。 由活动callid过滤呼叫并且寻找一callid类似此示例：

```
start,parameter,callid=BBBBAAAACCCDDDDDEEEFFFAAAAABBBB
```

步骤3.收集报告的CVP报告的服务器的日志。查找在报告的CVP的同样callid日志。

```
ValidateHandler:ValidateHandler.exec: ValidateHandler GUID=BBBBAAAACCCDDDDDEEEFFFAAAAABBBB  
results:none validation status bitmask=0x00000103
```

步骤4.转换bitmask编号成二进制。请使用一个程序员计算器：**0001 0000011**

第5.步。检查CVP报告指南bitmask CCB表。您应该看到验证失效由于“EXCEED_CAPACITY_GW”。

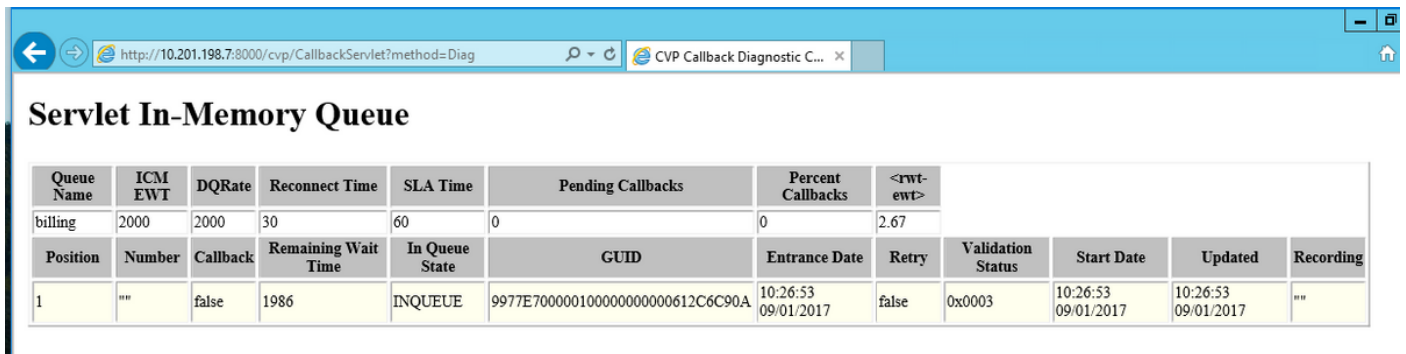
```
00000000 00000001 OK  
00000000 00000010 ICM_NO_SCHEDULED_ALLOWED  
00000000 00000100 ICM_NO_PREEMPTIVE_ALLOWED  
00000000 00001000 NOT_IN_QUEUE  
00000000 00010000 TOD  
00000000 00100000 EWT  
00000000 01000000 PROBE_FAILED_NO_RESPONSE  
00000000 10000000 PROBE_FAILED_NO_CONFIG  
00000001 00000000 EXCEED_CAPACITY_GW  
00000010 00000000 EXCEED_CAPACITY_QUEUE
```

Note: ICM_NO_SHCEDULED_ALLOWED

步骤6.缩小问题对一个特定队列。检查从报告的CVP的CCB Servlet服务器为了确定是否有提供CCB的任何特定队列。打开Web浏览器并且键入。

http:// {报告服务器IP Address}:8000/cvp/CallbackServlet?method=Diag

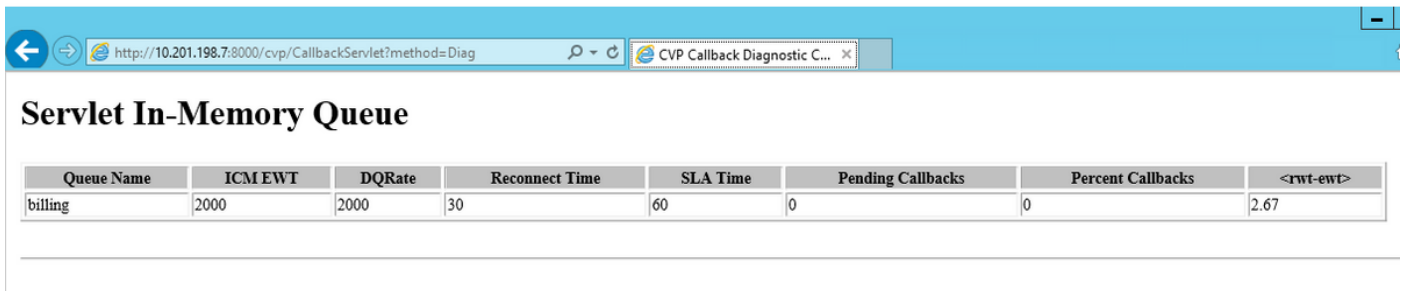
这是提供CCB队列的示例：



Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rw-ewt>
billing	2000	2000	30	60	0	0	2.67

Position	Number	Callback	Remaining Wait Time	In Queue State	GUID	Entrance Date	Retry	Validation Status	Start Date	Updated	Recording
1	""	false	1986	INQUEUE	9977E7000010000000000612C6C90A	10:26:53 09/01/2017	false	0x0003	10:26:53 09/01/2017	10:26:53 09/01/2017	""

这是没有提供CCB队列的示例



Queue Name	ICM EWT	DQRate	Reconnect Time	SLA Time	Pending Callbacks	Percent Callbacks	<rw-ewt>
billing	2000	2000	30	60	0	0	2.67

第7.步。检查队列是否由一个特定网关服务。检查网关配置(生存能力应用程序参数)。

```
application
service new-call flash:bootstrap.vxml
!
service survivability flash:survivability.tcl
  paramspace callfeature med-inact-det enable
  param ccb id:10.201.198.21;loc:CALO;trunks:512
```

第8.步。如果配置是正确的，请检查在报告的服务器数据库存储的信息(Informix公司)确定呼叫的数量在此特定网关和位置的。您能由CCB id (10.201.198.21在这种情况下)或locattion (在本例中的CALO检查)。

第9.步。在报告的服务器上，请访问Informix数据库。

打开CMD提示并且键入：**dbacces**

连接对**连接>连接**

选择**cvp实例**

键入用户名cvp_dbadmin

键入密码

选择callback@cvp数据库

退出并且连接对查询语言

步骤10.运行查询：

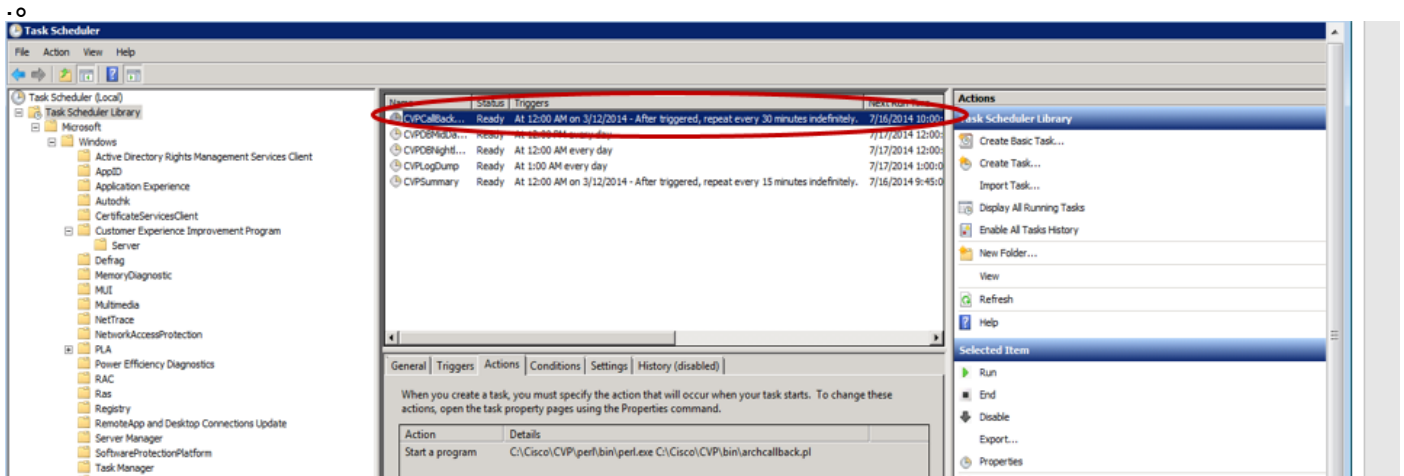
选择计数(*)从callback_current其中位置=="CALO";

第11步。如果值高于在位置的网关配置的Trunk值是相同的或，这是原因为什么validation发生故障，因为允许的Trunk的最大数量在Callback_Current表里到达了。

Note: 作为被参考CVP报告指南，回拨表是两张表视图：Callback_Current和Callback_Historical。两张表是相同的。每30分钟，数据为完整呼叫从Callback_Pending被拉并且被移动向Callback_Historical。

步骤12。如果Trunk值每个位置在Callback_Current表里达到了其限额，并且这表明的没有在队列的回拨有在移动回拨记录的一个问题从Callback_Current向Callback_Historical表。

第13步。保证CVPCallbackArchive运行在日程表任务(报告的CVP下服务器)。连接对Start > Programs > Accessories ->System工具->被安排的任务。



步骤14。如果此任务CVPCallbackArchive完成请保证退出代码是(0x0)。

Name	Status	Triggers	Next Ru...	Last Run Ti...	Last Run Result	Author
CVPCallBack...	Ready	At 12:00 AM on 1/31/2017 - ...	8/30/20...	8/30/2017 4...	The operation completed successfully. (0x0)	Administrator
CVPDBMidD...	Ready	At 12:00 PM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPDBNight...	Ready	At 12:00 AM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPLogDump	Ready	At 1:00 AM every day	8/31/20...	8/30/2017 1...	The operation completed successfully. (0x0)	Administrator
CVPSummary	Ready	At 12:00 AM on 1/31/2017 - ...	8/30/20...	8/30/2017 4...	The operation completed successfully. (0x0)	Administrator
GoalLeds	Ready	Multiple triggers defined	8/30/20...	8/30/2017 2...	The operation completed successfully. (0x0)	Administrator

第15步。如果第13步和第14步优良是，但是仍然没有数据在Callback_Historical表里，您将需要确定信息为什么在数据库没有被添加。检查在当前和历史表里存储的信息的完整性。运行在informix公司dbaccess CMD窗口的此查询：

```
Select count (*) from callback_current where surrogateid in (select surrogateid from callback_historical);
```

第16步。如果计数1或更高，意味着在当前表的主密钥在历史表里已经存在，并且信息没有被添加到数据库。在大多这些方案，加入的竞争状态原因重复项记录callback_current表。

对surrogateid映射的GUID在队列表发生。在呼叫从回拨等待移动到回拨队列脚本的情况下，看起来似乎档案工作从当前移动记录到历史记录窗口，并且应用程序在与同一surrogateid的当前表里输入一个新建记录。此问题与此CDETS [CSCuq86400](#)有关

解决方案

步骤1.访问Informix数据库。打开CMD提示并且键入：**dbaccess**

步骤2.连接对**连接>连接**挑选cvp实例。键入用户名cvp_dbadmin并且键入密码

步骤3.选择callback@cvp数据库退出并且连接对**查询语言**

步骤4.运行这些命令：

从callback_current的删除其中surrogateid (从callback_historical的挑选surrogateid);

如果有一个临时表错误：

丢弃表T1;

步骤5.运行从当前移动信息向从查询语言窗口dbaccess的历史回拨表的sp程序。

执行程序sp_arch_callback();

第6.步。检查没有许多个记录在当前表里作为以前。

选择计数(*)从callback_current其中位置=="CALO";

永久解决方案

步骤1.连接对Cisco \ CVP \ informix_frag并且打开在文本编辑的sp_arch_callback.sql。

步骤2.在文件初注释删除此线路：**--丢弃程序sp_arch_callback;** (请去除--在线路的开始)。

步骤3.添加此线路：**从callback_current的删除其中指定代理(从callback_historical的挑选 surrogateid);**以后

创建程序sp_arch_callback()线路。

步骤4.保存文件。

第5.步。这是关于怎样的一个示例文件的第一部分应该看起来象。

```
{ *****  
Stored procedure to move completed calls out of the active table into the
```

```
historical table.
*****}
drop procedure sp_arch_callback;
create procedure sp_arch_callback()

DEFINE p_ageoff INTEGER;

-- delete any duplicates found in current table.

delete from callback_current where surrogateid in (select surrogateid from callback_historical);
```

测试结果解决方案

步骤1.打开CMD提示并且运行命令：**dbschema**

dbschema -d回拨-f sp_arch_callback

Note: 如果有一个授权问题再次当运行dbschema命令、登录作为cvp_dbadmin到报告的服务器和尝试。

Step 2.从输出，请保证从命令的删除被执行。

```
C:\Users\Administrator>dbschema -d callback -f sp_arch_callback
DBSCHEMA Schema Utility          INFORMIX-SQL Version 12.10.FC3

create procedure "Administrator".sp_arch_callback()
DEFINE p_ageoff INTEGER;
-- delete any duplicates found in current table.
delete from callback_current where surrogateid in (select surrogateid from callb
ack_historical);
SELECT surrogateid
FROM Callback_current
WHERE EventTypeID in (24,29) -- Completed, Too many callbacks
AND CauseID in (27,28) -- Connected, Cancelled
INTO TEMP t1 WITH NO LOG;
```