Cisco Unified Contact Center Express and Cisco Unified IP IVR - Best Practices

Cisco Unified Contact Center Express product is a fairly open product which is programmed by the customer or partner to run various types of scripts and upload documents, prompts and grammars, and even deploy their own custom code. All this flexibility can lead to memory and CPU issues if not used carefully.

Below are some optimizations and best practices that will enhance the UCCX experience.

- Memory Best Practices
- CPU Optimizations
- Others

Memory Best Practices

Custom Code

Customer/Partners are allowed to create and deploy their own code which can be invoked from the scripts directly. This code shares the same memory as the UCCX system main engine and has the same privileges as the main engine.

While this gives tremendous amount of capability to the customer to define the behavior of the system and what logic it can execute while handling its contacts, it also means that any faulty or malicious code can bring down the system. Problems that could arise due to custom code include memory leak, access violations, and other CPU and timing issues. While these are normal for any program, care must be taken to test the custom code outside the UCCX system and also in UCCX context before deploying it into production.

It is also encouraged to use possible Java techniques to test the code outside the UCCX environment. These include JTest unit testing, memory profiling and any other suitable forms of stress test. Attention should also be provided to handling of error situations and exceptions.

Large Scripts

Care should be taken while deploying very large scripts into the UCCX system. Since large scripts take up the limited memory of the UCCX Engine which is actually shared with all the logic for call processing.

This is not an issue most of the time, since there is enough space for most needs but since there is no mechanism to judge the optimal size of scripts, this will remain a potential pitfall for the existing releases.

An equivalent of large scripts is very large VoiceXML scripts. Here again since the whole script is loaded into memory for each invocation (each call) in the parsed form, it can lead to excessive memory usage and subsequent crash. Note that since VoiceXML scripts may be loaded from external web sites, the partner/customer may not be fully aware of its memory needs during execution. Once again such situation can be averted by pre testing these scripts.

Multiple Applications

While large scripts remain a potential issue, creating a huge number of applications which use the same script or very similar scripts can also lead to excessive memory consumption leaving very little for other activities like call processing.

Again script developers can be imaginative and build logic inside the script to handle calls in different ways for different originating triggers. Thus, a single application can be deployed with multiple triggers. The script can be coded to handle the calls to these triggers slightly differently e.g. playing a different prompt or presenting a different language. Another way of customizing parameters for different triggers is to store the information either in a custom database or in an XML document that can then be read within the script to determine parameter values based on the dialed number.

The point to be noted here is that with each application one script is loaded into memory. If the script happens to be the same one, then it is loaded multiple times using up precious memory.

Prompts and Grammars

Prompts, grammars and other run time document objects loaded into UCCX Engine memory during execution can bring down the system if they are too large in size.

Such situation can arise if there are large advertisement prompts.

Excessive Usage of Script Variables (especially Strings)

Using a large number of string variables coupled with large call volume can lead to excessive memory usage and subsequent crash.

The same can occur with other variable types. This issue can be curbed by suitable optimization.

Reading or Parsing large XML files

This has impact during runtime, depending on the number of instances of the application executing simultaneously, since the XML file will be loaded into memory per application session.

Reading Enterprise Database

While using Database steps in Unified CCX Editor, use DB Release step after a DB Read or DB Write step to utilize the connection pool effectively.

Generating Heap/thread Dumps

In spite of best efforts once in a while we do encounter Java memory leaks and crashes due to that. In such cases TAC often requires customers to collect UCCX Engine Java heap dumps (and possibly thread dumps) for analysis. Such heap and thread dumps should strictly be done only during off-peak hours to minimize disruption to active calls.

Here it should also be highlighted that the UCCX main Engine process memory is bounded and fixed. This has been done to optimize real time call processing which is the core functionality of the product. This also means that there is no scope of increasing the memory pool of UCCX Engine even if the system has spare main memory (RAM). UCCX 7.0(1) SR3 and other future releases will provide a mechanism to monitor the UCCX Engine memory usage in real time and have the ability to generate alarms upon hitting certain thresholds [CSCsz21189].

Since Script (and custom code) development is just another form of programming, many of the usual guidelines for efficient and optimized coding also applies here.

The above guidelines are not exhaustive, and that's where the knowledge, care and imagination of the script programmers come into play. And to re-emphasize, script programmers must as a rule test the script(s) just like they would any other software programs, taking extra care of the boundary and rare conditions.

CPU Optimizations

Call Volume

This is always highlighted even before the system is designed and bought. So issues due to excessive call volume are not common.

However, one must keep in mind that call volume is not just the BHCC or BHCA figure which is calculated across an hour or several hours. If there is a sudden surge in calls over a few seconds, CPU can peak and lead to dropped or failed calls. Please refer to the config and ordering tool to make sure that the system is within the specified guidelines for expected call volume.

Usage of Heavy Scripts

Script developers must be aware of any complex (CPU intensive) computation that the scripts might be required to do. If there are tight loops and/or CPU hogging threads especially as part of custom code, it can use up CPU and lead to starvation for actual call processing. This in turn will manifest as dropped or failed calls.

Voice XML

VoiceXML scripts are in general more CPU intensive due to a whole lot of parsing of XML documents. When this is done real time in parallel to call processing, the later can be affected.

Voice XML scripts should be deployed with care.

HTTP Requests

Usage of HTTP requests to query various kinds of status and real time information, when done very often, say every 3-5 seconds or so, it can cause extra load on the system in-terms of memory and CPU usage.

The general impression is that since there is no call processing, it is OK. However one must keep in mind that these HTTP requests also need to be processed and they share the same resources – CPU and memory as call processing.

Max Number of Executed Steps

Customer should not change the "max number of executed steps" parameter unless instructed by TAC. The purpose is to prevent infinite loops which can impact system performance greatly.

Others

Non-customary Usage of UCCX Database

UCCX Database server is for the exclusive use by UCCX components. Hence, customers are not allowed to create Databases and/or tables in the UCCX DB server. However customers can read historical data records for analysis where the canned reports do not suffice.

In such cases, all DB access should be tested thoroughly by partner/customer to ensure that there is no impact to UCCX in terms of performance.

Care must be taken to not leave non-functional DSNs since UCCX polls those periodically. There is not good method of judging if the DSN is temporarily down or permanently. Also even if there is virtually no limit on the number of DSNs configured, it should be used judiciously.

Operation in Island mode due to Network Outage

The Island mode occurs when one of the UCCX node(s) gets detached from the LAN while still functioning. In the Island mode, both nodes run as masters of all components. It must be noted that when the network connectivity is restored for the detached node, it goes through convergence when a single master is elected. As a result of that both Engines drop mastership and with that all active calls.

Similarly, if master re-election is done on purpose it must be noted that there will be disruption of calls. Also mastership moves to the more powerful machine hardware and if both Engine nodes are of equal power, the election outcome is non-deterministic.

Disk Fragmentation in Windows Environment

Windows OS (more specifically NTFS) has the unique issue of disk fragmentation. With prolonged usage, various files are created in memory and deleted/purged. However this leads to fragmentation as in the contiguous bytes of memory are broken up into smaller and smaller chunks. When the disk is heavily fragmented, any fair size disk space requirement cannot be satisfied. Disk reads and writes also are affected since the files spread across multiple fragments.

This can cause pauses in the system with catastrophic impact on call processing as all the processes are forced to freeze for seconds even.

Hence, Disk De-fragmentation activity should be added to regular maintenance operations. However all disk de-fragmentation operation should strictly be run during off-peak hours.

Low Disk Space

Low or insufficient disk space can also bring down the system. Hence care must be taken that the system does not reach such a situation ever.

Few of the activities which are helpful are

- Properly set DB purging
- Set the number of trace files to be generated at optimal levels/numbers.
- Turn off excess tracing when not needed. This usually happens after a debugging session when traces are turned ON, but administrators forget to turn them OFF once done.

Free Disk space should be maintained at 15% or higher for best all round results. Disk de-fragmentation operation also runs more efficiently (i.e. less disruption) with 15% or more of free disk space.

Third Party Software

Cisco does not permit any unapproved third party software on the UCCX server. Only Cisco approved anti-virus, anti-spy ware software should be used along with the compatible version of Cisco Security Agent. On similar lines no patches/upgrades should be applied to SQL Server, Java or Windows itself. Only Cisco provided or approved patches are allowed.

Regular Backup and Recovery Testing

Customers are encouraged to take advantage of the regular backup mechanism built into UCCX (and associated products like Cisco UCM). Since a backup is meaningful only if the recovery mechanism works fine it is recommended that a replica system be used to test the recovery without touching the production system. This is only if a replica system can be afforded.

Following the documentation step by step is also important during any restore/recovery operation.

Upgrades

Two crucial guidelines during major upgrades (not Service Releases or Engineering Specials) are as follows

Step 1 Follow the steps described in Upgrade document to the point. This is even of the customer has had experience in upgrades before hand.

Step 2 Use a replica system to build an upgraded system before attempting the upgrade on the production system. This becomes important for customers who have a complex or large Contact Centers and downtime is not permitted.

IP Address Change

If ever any customer needs to change the IP Address of one or more machines in the solution, it is strongly recommended to follow the guidelines and detailed procedure mentioned in the corresponding document. Any deviation from the steps can lead to a irrecoverable damage to the system needing a full rebuild. Yet another reason to have backups before this major operation is attempted.

This document is to be used in conjunction with *Cisco Unified Contact Center Express* Support Documentation.