Cisco Unified Contact Center Express and Cisco Unified IP IVR - Best Practices
Release 9.0(1)

Last Modified: July 6, 2012

Cisco Unified Contact Center Express (Unified CCX) 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 Unified CCX experience.

- Memory best practices
- CPU optimizations
- Others

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Memory best practices

Custom code

Customers and partners are allowed to create and deploy their own code which can be invoked from the scripts directly. This code shares the same memory and has the same privileges as the Unified CCX system main engine.

While these privileges gives tremendous amount of capability to the customer to define the behavior of the system and the logic it can execute while handling customer 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 Unified CCX system and also in Unified CCX context before deploying it into production.

You are also encouraged to use possible Java techniques to test the code outside the Unified CCX 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 Unified CCX system, as large scripts take up a big portion of the limited memory of Unified CCX Engine. The memory is actually shared with all the logic for call processing.

Limited memory is generally not an issue for most of the time as there is enough space for most needs. However, as there is no mechanism to judge the optimal size of scripts, memory consumption 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 websites, partners and customers may not be fully aware of memory needs of the scripts during execution. Once again such situation can be averted by pretesting 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, that is, 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 for different applications, then it is loaded multiple times using up precious memory.

Prompts and grammars

Prompts, grammars, and other run time document objects loaded into Unified CCX 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 that are executing simultaneously, since the XML file will be loaded into memory with every application session.

Reading enterprise database

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

Generating heap or thread dumps

In spite of your best efforts once in a while you may encounter Java memory leaks and crashes due to these leaks. In such cases TAC often requires customers to collect Unified CCX 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 Unified CCX main Engine process memory is bounded and fixed. This step has been taken to optimize real time call processing which is the core functionality of the product. It also means that there is no scope of increasing the memory pool of Unified CCX Engine even if the system has spare main memory (RAM). Unified CCX provides a mechanism to monitor the Unified CCX Engine memory usage in real time and generate alarms upon hitting certain thresholds.

Efficient and optimized coding, which are the guidelines of programming also apply to script (and custom code) development.

The above guidelines are not exhaustive, and that is 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 scripts just like they would any other software programs, taking extra care of the boundary and rare conditions.

CPU optimizations

Call volume

Call volume 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. 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. Tight loops and or CPU hogging threads, specially as part of custom code, 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 in real time in parallel with call processing, the latter can be affected.

VoiceXML scripts should be deployed with care.

HTTP requests

HTTP requests to query various kinds of status and real-time information when made frequently, say every 3-5 seconds, causes extra load on the system in terms of memory and CPU usage.

The general impression is that when there is no call to process, it is alright to run queries. 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 required for call processing.

Maximum number of executed steps

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

Others

Non-customary usage of Unified CCX database

Unified CCX database server is to be used exclusively by Unified CCX components. Hence, customers are not allowed to create databases and or tables in the Unified CCX DB server. Customers can, however, 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 and or customer to ensure that there is no impact to Unified CCX in terms of performance.

Take care not to leave non-functional DSNs since Unified CCX polls the DSNs periodically. It is not possible to judge if a DSN is temporarily or permanently unavailable. Even though there is virtually no limit on the number of DSNs that can be configured, use DSNs judiciously.

Operation in island mode due to network outage

The island mode occurs when one of the Unified CCX 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, both engines drop mastership as well as 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 the contiguous bytes of memory are broken up into smaller and smaller chunks. When the disk is heavily fragmented, any fair-sized disk space requirement cannot be satisfied. Disk reads and writes are also 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 could even be forced to freeze for seconds.

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

Low disk space

Low or insufficient disk space can also bring down the system. Care must be taken to ensure that the system never reaches such a situation.

Few of the activities which are helpful to avoid low disk space 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 defragmentation operation also runs more efficiently or with less disruption with 15% or more of free disk space.

Third party software

Cisco does not permit any unapproved third party software on the Unified CCX server. Only Cisco approved anti-virus, anti-spyware software should be used along with the compatible version of Cisco Security Agent. On similar lines, no patches or 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 Unified CCX (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 making upgrades earlier.

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 complex or large Contact Centers and downtime is not permitted.

IP address change

If a 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 and may require a full rebuild. This is why backups are a must before any major operation is attempted.
