esg116147  BOCS pre-caching and DMS related information

|  |  |
| --- | --- |
| Branch Office Caching Services 6.5 SP3, 6.5 SP2, 6.5 SP1, 6.5 |  |

**Symptoms**

The following solution provides Branch Office Caching Services (BOCS) pre-caching and DMS related information.

**Resolution**

The following information was gathered from discussions with Engineering about how pre-caching job, DMS and BOCS work together:

1. BOCS pre-caching and Asynchronous operations rely on the message which is either transferred by or grabbed from DMS depending on whether BOCS is configured in push or pull mode.

When BOCS Asynchronous operation is triggered by WDK-based application (or DFS), the DFC in WDK side generates a message for the content file to be uploaded.  This message contains all the required information (including credentials) for BOCS to upload the file into proper ACS instance.  This message is sent to DMS and kept in DMS database (Sybase instance). If DMS is not up, error is not raised. As a back-up, Async write job needs to be executed on a schedule. Job generates content upload message for each content that has not been uploaded yet. Meanwhile, content is fully available for all users that can use BOCS where the content is stored.

Similarly when BOCS pre-caching job runs, the DFC used by the Java Method Server generates one message for each content file to be pre-cached.  Then this message is also sent to DMS. In order to pre-cache content on BOCS, pre-caching job sends a message (~2Kb) for each content (to pre-cache) to DMS. If you need to pre-cache 1000 files, 1000 messages will be added to DMS, regardless of a number of BOCS to pre-cache content on.

2. DMS uses the following message statuses:

0 - Not processed by DMS, i.e.. not sent to BOCS. It means that the message has been received by DMS and stored in the DMS DB.

1 - Delivered, it means the DMS has sent the message to BOCS. When DMS sends the message to BOCS, it doesn't wait any reply from BOCS, simply it marks the message as delivered.

2 - Success, it means that BOCS has processed message successfully and has notified DMS.

3 - Failed, it means that BOCS faced an issue when processing message and has notified DMS.

This numbering may be changed in a future.

3. For security reasons, each message is only delivered once and the URL used by BOCS to pre-cache contents is valid for 6 hours.

4. DMS provides ability to define the lenght of time to keep messages on DMS. This time period in days is defined by dms.cleanup.period parameter in dms.properties file. If you modify the dms.properties file, you need to restart DMS.  The file is located in [jboss]/server/DctmServer\_DMS/deploy/DMS.ear/APP-INF/classes where [jboss] designates the DMS Jboss instance installation path.

5. You can perform operations on messages stored in DMS DB or configure some DMS parameters through a DMS resource agent (needs to be created through DA). Note that if the DMS matching has an “\_” (underscore) in its name, the DMS resource agent cannot be created.

Create the DMS resource Agent by specifying the JMX service URL for DMS JBoss instance (the information can be retrieved from the DMS installation logs) (see DMSAgent1.JPG)

Once created, use the “View Resources on Agent” function to retrieve all the administration beans related to the agent. The DMSMessageModifyMBean allows you to delete messages, the DmsMessageListMBean allows you to list existing messages, and the DmsServerConfigMBean administration bean allows you to set some configuration parameters. (see DMSAgent2.JPG)

Methods “getWaitingDestinationMessageCount” and “getWaitingMessageCount” in DmsMessageListMBean return the number of messages with status 0 and 1. These are messages that have not been sent to BOCS and messages that have been sent to BOCS, but have not been reported as failed or succeeded. In other words, waiting messages are “messages not delivered to BOCS” OR “messages delivered to BOCS that BOCS did not process yet” OR “messages delivered to BOCS message that BOCS processed but did not send to DMS status for.

6. Content size is irrelevant for DMS as DMS only handles messages. Thus DMS performance is not related to a size of cached data, only to a number of messages in DMS DB.

7. When BOCS receives messages, it stores them using JMS (Java Message Service), provided by JBoss. JBoss JMS stores messages in Sybase DB. BOCS does not access this DB directly. Before BOCS processes messages, a few of them are retrieved from JMS. Once messages are removed from JMS, they are never put back. Failed messages are not reprocessed. By default, there are only 2 workers/threads caching contents on BOCS. The number of workers is NOT configurable in version 6.5 of the product.

When BOCS sees that the message represents BOCS pre-caching operation and BOCS does not have the full content yet, it contacts ACS using URL in the message to download the content file from Documentum repository into the folder represented by acsCache folder.  For each file, BOCS keeps some information from which it can tell whether the content file requested by the message already exists in its file system.

If the file doesn't exist in the file system or content is partial, BOCS downloads the content file and creates another xml descriptive file. As a result, every cached content on BOCS has 2 files, which means that to count the number of pre-cached contents; you can count the number of files in the cache store (acsCache) and divide it by 2.

8.  If the BOCS message is already expired, it is simply discarded and the content file is not downloaded.  At this stage, we are not clear what the status of the discarded messages is at DMS level. Does BOCS notify DMS that these messages are in “Failed” status? Or simply BOCS doesn't notify DMS?

9. As mentioned earlier, each message sent to BOCS by DMS is only valid for 6 hours. As a result, the pre-caching job shouldn't send more message than what BOCS can handle in 6 hours timeframe. For example, the pre-caching job sends 30 000 messages to DMS and the DMS delivers these 30 000 messages to BOCS. If BOCS can only handle 20 000 messages (which means BOCS can cache only 20 000 contents) during the 6 hours timeframe, then 10 000 messages will be discarded by BOCS since they will be expire within the time frame.

In other words, proper size of set of documents should be selected in BOCS pre-caching job.

10. BOCS will not pre-cache more than 100,000 contents by default. Once it reaches the limit it will remove some cached content. There are undocumented BOCS parameters to go beyond this limit. Default value of cache.paging.max\_count = 100 in BOCS' acs.properties allows to cache up to 100,000 documents (1000 times count parameter value). Setting cache.paging.max\_count=1000 in BOCS' acs.properties allows to cache up to 1,000,000 contents (1000 times count parameter value).

Please note that the structure that keeps track of documents on a BOCS server is in-memory only (for performance purposes). Each document takes in this structure about 100 bytes. It means that if we want to allow 1,000,000 documents cached on a BOCS we need ~100MB for this structure. So it is impractical to keep 10 million documents in BOCS running 32 bit JVM, since 1 GB memory shall be required only to keep the structure of the documents.

11. BOCS has an internal housekeeping mechanism. The housekeeper runs once a day by default. Additionally, if a new content to be stored in the cache cannot be stored due to a lack of space according to defined space quota or if the number of contents reaches the number specified in cache.paging.max\_count, the housekeeper will run outside of the scheduled execution to purge content to free space in the cache.

This is specified in the BOCS release notes under the section “Configuring cache housekeeping”.

12. Underscore in the machine name of DMS or BOCS is watched to prevent DMS or BOCS from running properly.  Besides of the problem to register an agent through DA (see point 5) BOCS pre-caching job fails to run properly when BOCS host includes a “\_” in its name. Even though a customer could work around this limitation by adding alias for machine name, it is unclear whether it clears all the problems. Therefore, “\_” should simply be avoided in DMS and BOCS hosts names.

13. Content pre-caching is a performance optimization. As such it might improve user experience when content is already on BOCS. It can also negatively affect users of all systems that use repository by loading content server during pre-caching.

Therefore, pre-caching needs to be used cautiously and knowingly that pre-cached content is going to be used by BOCS users. In any case, BOCS works fine when content is not present on BOCS, so failing to pre-cache content cannot be viewed as a system problem under any circumstances.

**Additional Q&A:**

==============

**Question 1.**

Let's say pre-caching job, version 6.5 SP2 or newer, needs to process 200,000 contents and it fails in the middle.   Is it going to start next time from the beginning and process all the content again?   Or, is it going to start in the vicinity of the place where it stopped last time?

**Answer**

Pre-caching job will not start from the beginning. It should start from the time of the 'newest' dmr\_content object from the last successful batch processing.

**Question 2.**

So all BOCS use the same encryption key?

**Answer**

Each BOCS uses its own encryption key, which is stored in C:\Documentum\bocs\config\bocs.keystore file.  The first key is created when BOCS is installed.  You can always generate a new key using resource manager.  The key will be used with content that is going to be stored on BOCS after key is created.

**Question 3**  
On BOCs sever acs.properties, there are certain parameters to increase the throughput by increasing content download threads.  Please help us understand these parameters in acs.properties file:  
  
################### 3. Performance Enhancements : Policy Implementation   
policy=Throttle,1,10,0  
policy.1=ReplaceRequest,50,1

**Answer**

UCF client performs multithreaded download of content when this is beneficial for the user.  In a similar way, BOCS might download content from ACS using multiple threads.  Each thread opens HTTP connection to a server.  If lots of clients open multiple connections, server can be overstressed. To prevent this situation, ACS and  BOCS use these policies that define whether clients can use many connections or need to be limited to one, depending on a number of already opened connections.

Policy **Throttle,X,Y,Z**  means that if overall number of active connections is more than X then each new client will get Y new connections and Z number of connections will be taken from existing clients but only those that have more than Y number of threads.  For example, policy

policy=Throttle,1,10,0

means that this policy gets active as soon as first connection to server gets established.  It allows up to 10 concurrent connections per client and client doesn't take any additional connections from other clients

Policy **ReplaceRequest,X,Y** means that if overall number of active connections is more than X then each new client will get up to Y new connections and these number of connections will be taken from the existing clients, that is from one of the existing clients will lose Y connections, one by one.  We don't allow any client to lose all of it connections (at least one will be left).  For example, policy

policy.1=ReplaceRequest,50,1

means that if overall number of active connections is more than 50 then each new client will get  1 new connection and this connection will be taken from the existing client that has more than 1 connection.  No clients will have less than 1 connection and there is no pre-set limit on a number of clients