Adobe CQ Help / Performance Tuning Tips

Issue

My CQ5 instance performance is poor.

Cause

Performance problems in CQ5 can be due to many things in combination. The most common cause is due to application code. However, some measures can be taken within CQ5 configuration to improve performance.

Resolution

To improve the overall performance of the CQ instance, we recommend that you do the following (on author and publish instances):

1. Increase CRX's bundleCacheSize

Increase the `bundleCacheSize` param of CRX `TarPersistenceManager` by doing the following:

Edit each of the `PersistenceManager` elements in your `repository.xml` and all of your `workspace.xml` files by adding the following parameter within the `<PersistenceManager class="com.day.crx.persistence.tar.TarPersistenceManager" />` element:
e.g.

```
<PersistenceManager class="com.day.crx.persistence.tar.TarPersistenceManager">

    <param name="bundleCacheSize" value="256" />

</PersistenceManager>
```

Per default, this cache is set to 8mb; increase this to at least 256, or even512, depending on how much memory you can assign to the jvm. If you have a jvm with the Xmx param set to 10gb then you should set the `bundleCacheSize` to 1024.If you're using a 32 bit jvm on Windows, your Xmx param is probably less than 1500mb, thus a reasonable value for the bundleCacheSize may be 128mb. It is highly recommend to upgrade to a 64 bit JVM in order to allocate more memory to the jvm heap.

2. Use CRX's FineGrainedISMLocking (CRX 1.4.x only)

To configure it, simply add following the following ISMLocking class to the `workspace.xml` and `repository.xml`, directly after the SearchIndex block:

```
<Workspace ...>
```

```
    ...

    </SearchIndex>

    <ISMLocking class="org.apache.jackrabbit.core.state.FineGrainedISMLocking"/>

    ...

</Workspace>
```

3. Disable the CQ5 AssetSynchronizationService (CQ5.3 only)

The AssetSynchronizationService is used to synchronize assets from mounted repositories (eg. LiveLink, Documentum, ...). When enabled. It has been observed that this service periodically allocates many objects which get garbage collected, thus it has an impact on the performance of the jvm. If you do not use mounted repositories, this service can be disabled.

By configuration, the synchronization period can be set to a higher number of seconds (default 300) thus factually preventing it from running.

The attached package sets the sync period to 1 year, thus disables the service.

4. Set CRX Search Index's resultFetchSize param

If the result set of a jcr query is large, then the loading the complete setand checking ACLs on them is quite expensive.
To remedy this, limit the fetch size to 50 as via the `SearchIndex` element in the `workspace.xml`:

```
<param name="resultFetchSize" value="50"/>
```

e.g.

```
<SearchIndex class="com.day.crx.query.lucene.LuceneHandler">

    <param name="path" value="${wsp.home}/index"/>

    <param name="resultFetchSize" value="50"/>

</SearchIndex>
```

5. Set the CRX Search cacheSize param

The search cacheSize can also be set in the `SearchIndex` element in `workspace.xml`. Set this parameter to `100000`:

```
<param name="cacheSize" value="100000" />
```

e.g.

```
<SearchIndex class="com.day.crx.query.lucene.LuceneHandler">

    <param name="path" value="${wsp.home}/index"/>

    <param name="resultFetchSize" value="50"/>

    <param name="cacheSize" value="100000" />

</SearchIndex>
```

This parameter is documented here.

6. Disable CRX clustering (CRX 1.4, 2.0, 2.1)

When running a system with high write load (e.g., with massive imports of DAM assets), sometimes even relatively small write performance gains can help achieve required performance benchmark. In such case, you might consider turning off clustering journal in your deployment.

By default CRX is setup to run in cluster mode, but if there is only one instance in the cluster,then it adds some I/O overhead. To gain write performance, we can disable clustering.

In the repository.xml, comment out the `<Cluster ...>...</Cluster>` section.
Then use the sample script to move the repository files from cluster mode to single mode.

Before doing this, please contact your daycare or your account manager to discuss this option.Please note that most of customer's deployments run with the default, journaled configurations.It is best to test this script on a test instance doing it on a production system.

7. Disable Content Finder Refresh and Auto Load (Author instance only)

See this article [2]

8. Disable DAM Sub Asset Generation

See this article [3]

9. Limit the Max Journal Size in CRX

If the CRX journal files under share/journal grow too large then your application can experience some slowdown. See this article [4] for how to limit the journal size and maximum number of files.

10. Disable DAM workflows on publish (CQ5.3 Publish instance only)

See this article [5]

11. Disable the Link Checker

A small gain in performance can be had by disabling the link checker. You should only do this if you decide that you do not need it in your environment.
The link checker validates that resources addressed by URLs on your pages are reachable. It does this by doing an HTTP HEAD request.
If a link is marked invalid on author, then the link checker displays a broken chain icon around it. If a link is marked invalid on publish, then the surrounding `<a href="...">` tag is removed.

Some clients disable it in their publish instances to gain performance. Although this gains performance, please note that broken links affect your site's SEO. Please consider the effects carefully before deciding to disable it.

For instructions on disabling this feature see this article [6].

12. Cache Tar PM index

Until CRX 2.1

Using a cron job to read the index*.tar file from time to time, can help to improve the performance, as the tar index will be cached in the hardware I/O cache. On Unix you can load the index*.tar using the "cat index*.tar > /dev/null" command. Doing that every hour should help to improve the performance of the Tar Persistence Manager while executing TarFile.readFully() method.

From CRX 2.2 + hotfixpack 2.2.0.26 and later

Check the global size of the index_*.tar files in the crx.default workspace, increase the max heap size by such size, and set the indexInMemory param to true for the TarPersistenceManager section of the workspace.xml :

```
<PersistenceManager class="com.day.crx.persistence.tar.TarPersistenceManager">
    <param name="indexInMemory" value="true" />
</PersistenceManager>
```

The heap space required for one JCR node is 128 bytes. If the largest index*.tar file of a workspace is 1 GB, that means there are about 16 million nodes in this workspace, as each node needs 64 bytes of disk space. Using the in-memory index option will need about 2 GB of heap space (the file size needs to be doubled to calculate the heap space required because there can be

two versions per index file during index merge).

13. LDAP cache expiration

To improve the performance and reduce latency during authentication process, it's good to increase the LDAP cache expiration, you can set cache.expiration to 86400 (1 day) in your ldap configuration.

References

[1] http://wiki.apache.org/jackrabbit/Search
[2] How to change ContentFinder refresh interval
[3] How to remove subasset generation from DAM Workflow
[4] Journal consumes too much disk space
[5] How to disable DAM Workflows on publish
[6] Disable Linkchecker

Applies To

CQ 5.2 / 5.3, CRX 2.X

crx_disable_clustering.sh

AssetSynchronizationServiceConfiguration-1.0.zip