Q & A’s from AEM GEM Session: AEM 6 Oak: MongoMK and Queries 11/19/2015

Q. What is the unit of query cost on each index - time taken, nodes traversed, or something else?

> A. Originally, in the internal QueryIndex interface, we only had one value (the cost, a floating point number), and the unit was the expected number of disk read operations and network round-trips. That way, if the index has all the data in memory, then the cost of an index lookup is very low. Now, using the internal AdvancedQuery interface, we use 3 values: execution cost (the overhead to do an index lookup), estimated entry count, cost per entry. The unit of the costs is the same as with the old interface.

Q. How does the backup of Mongo Mk work? I know we can do the version base backup of tar MK and it is very efficient but I have not found any documents about backup and restore of mongo MK can you please touch on this topic a bit?

> A. The common solution is to use the MongoDB backup itself.
>
> Response: So when we restore the Mongo DB it is always in sync with JCR? We faced a problem few days back with our author cluster, the JCR and Mongo were somehow not in sync when we restored from Mongo backup. Ours is a video streaming solution over AEM and we use NAS to store our videos so any data that is bigger than 4kb will go in NAS. Is there any way of which you know through which we can backup mongo and NAS efficiently without any synchronization issue in backup.
>
> A. If you use a data store in the file system, you need to ensure the backup of the datastore is done after the backup of the MongoDB (I guess we need to document this). I also suggest to log a bug so we can investigate and document if we need to use a special backup procedure.

Q. Can you please document the approach to backup mongo backed by files based datastore. I just checked as suggested by you, we backup NAS after mongo but still it is sometimes not in sync.

> A. This is the recommended approach with this deployment. Please get in contact with Adobe AEM support if this process does not work reliably for you.

Q. Can this cluster set up work with Geo distributed AEM author instances? If so does it mean that the author instance in a data center will read/write to the mongo instance in the same data center for efficiency?

> A. This is something we are currently working on with tag aware sharding and content worked on by on geo would be written to a tagged shard in the local data center.

Q. In CRX2 there was an XML file for managing boost. Is there a similar configuration for OAK?

> A. This is still WIP tracked via OAK-2178.

Q. As we know XPath query is parsed and optimized to SQL2 but we are preferring XPath over SQL2 . Is this only due to simplicity in coding or any other reason for it?

> A. There is only a single internal query representation in Oak, which is closer to SQL2 and JQOM. This is why XPath is translated to SQL2.

Q. Is there any proper or standard solution for MongoMK conflict?

> A. Design a content model where conflicting writes cannot occur. For the remaining cases you can perform retries after a refresh of the session. Depending on your use case you may also try to isolate the writes on a single cluster node.

Q. Any optimization we have to configure in MongoDB?

> A. The default options should be just fine

Q. Can we cluster on different geolocation? If so how timestamp in revision handle it?

A. Timestamps are in UTC (coordinated time), and we don't rely on fully synchronized clocks so different geolocations should work just fine.

Q. Can we run oak-run when AEM instance is running?

A. This depends on the storage AEM is using. With MongoMK, you can connect oak-run as an additional cluster node. With TarMK, this is currently not possible.

Q. Is the 50+ index definition equal to the earlier AEM 5.6.1 index configuration?

A. It includes all of 5.6.1 plus new indexes to cover new features introduced since.

Response: I do not see that it covers all 5.6.1 Ex:- content finder query is not indexed

A. Queries are now optimized on a case by case basis, if there's one that's not performing as expected please reach out to Adobe AEM support for help.

Q. Which of the below negatively impact performance of a query?

Count of query results?
Filtering by path predicates?
Extracting facets?
Filtering by node type?
Search on full text?
Using order by clause?

A. Counting will definitively impact performance because it has to iterate over the complete result set. Path predicates are fairly efficient. Extracting facets will also read the complete result set. Fulltext search is quite efficient as well because it uses Apache Lucene TM. Ordering depends on an ordered index. If there is no such index, the query might be slow.

Q. Are there plans to make explain query part of AEM product?

A. Yes, http://adobe-consulting-services.github.io/acs-aem-tools/explain-query.html

Q. Can we programmatically determine which index to use rather than a framework to calculate & decide? If not as there any plan to provide such option in future?

A. Not currently, but it will be available at a later time.

Q. Can we configure to store index in a file system rather than Mongo, at least for debug purpose in lower environment?

A. Apache Solr uses external files. And Oak-Lucene has this option, but none of the other index impls do.

Q. When to use sync & async index?

A. You can use async when you can deal with having not up-to-date results because you might see stale information.

Q. What is the exact system property name to limit?

A. This is documented at jackrabbit.apache.org/oak/docs

Q. How to find storage size index is using?

A. You can use oak.indexStats() in the mongo shell with oak-mongo.js

Q & A’s from AEM GEM Session: AEM 6 Oak: MongoMK and Queries 11/19/2015

Q. Is there documentation out there on plugging in Solr for the versions of AEM?

A. It should be covered here https://jackrabbit.apache.org/oak/docs/osgi_config.html under 'Solr Server Configuration'

Q. How to perform sharding in Mongo?

A. Please consult the MongoDB documentation for this question.
Overview / Adobe Experience Manager 6.0 / Deploying and Maintaining / or The MongoDB Manual

Q. Can mongo store more than 1TB of data?

A. Yes, as long as it has enough disk space.

Q. We have a business requirement to deal with 30 TB digital assets, can Mongo can deal with that much volume of data?

A. For operational simplicity you might want to use the file system datastore for such volumes, i.e. store the nodes in MongoDB and the binaries in the FS. The file system must be shared across your various Oak instances in such cases

Q. How much data can Mongo store at max?

A. There is no theoretical limit as far as I know. Once Mongo's indexes (not to be confused with Oak's indexes) do not fit into RAM any more you will experience a performance hit, though.

Q. What is Mongo’s max capacity?

A. Data size is limited by OS virtual addressable space.

Q. Can you suggest me which MK will give good performance to maintain 20TB of data? I mean Tar or Mongo DB?

A. This depends on the kind of data you want to store. If most of the 20TB are binaries, you are probably fine with the TarMK and a data store on the file system. With 20TB of structured content / nodes, MongoMK is the better choice because you can shard the data.

Q. What are all factors to be considered for choosing MK(Tar vs Mongo)

A. TarMK is optimized for single node performance whereas MongoMK is designed for scalability and clustered deployments. Another factor is the volume of the data (see previous answer). Also keep in mind MongoMK requires additional know-how about MongoDB on how to operate it. You may also want to review the AEM GEMs Webinar: Technical Deep Dive into the AEM 6 Platform

Q. From the offline compaction documentation online - "Normal operation of the tool also requires old checkpoints to be cleared before the compaction takes place. Because of this, a full content reindex will be required after running the tool." - http://docs.adobe.com/docs/en/aem/6-0/deploy/upgrade/microkernels-in-aem-6-0.html#par_title_8, Is this true?

A. You can now remove the unreferenced checkpoints, so reindexing will no longer be required.

Q. If we have to re-index (currently we have it running on my clients production publisher server), how long it could take? (500 K content), how to check the status of reindexing? It's been running from 4 days now, Lucene and one custom index.

A. Re-indexing is usually not needed. But if it is needed, it is hard to say how long it takes. With half a million nodes, I would expect around one hour to re-index. Unless there are many documents (PDF, Word, and on so) that have a lot of content. If it takes much longer than that, then I suggest to open a ticket. You will need to describe the configuration (Tar or Mongo storage, data store configuration, file system(s) used and configuration), the size of the repository (disk space used per storage component), and around 10 or 20 full dumps (jstack –l <pid>) and heap histograms (jmap –histo <pid>). The thread dumps help identify the bottleneck.

Q. I thought Solr is developed by Lucene, What is main reason behind using both index?

A. Both are supported on different scenarios. You are not supposed to use both Lucene and Solr at the same time. You might want to use Solr when you need to offload the fulltext search to an external server via for example solrcloud.

Q. Is only one index used at the same time? Or there an indexes which can used together e.g. nodetype + property

A. Currently a single index is selected and used based on the cost.

Q. When reindex is running is old index still working?

A. Yes, except for the first time you build your index (as there's no previous version to be used)

Q. Is it ok to freely add new indexes? Are there any reasonable limits how many custom indexes can be added to do not cause performance degradations?

A. An index on very common properties (for example jcr:modifiedBy) should be avoided, because the size of the index would get very big, and it would therefore slow down write operations. Also, such an index would probably not help much, as selectivity is low (many entries per unique value). But properties that are not common or specific to a certain application are much cheaper to maintain, and usually much faster to query. I suggest to create one index per such property (do not add multiple properties to the same index). The overhead of an index is roughly one node per entry in the index, plus the related parent nodes.

Q. What is the recommended datastore impl. for today? Is default for TarMK and Mongo, or should we start with FS implementation? Can be DS changed later to FS?

A. In most cases the default configuration with binaries either in tar files MongoDB is fine. You should consider the file data store if you have special requirements for volume and/or performance. It is currently not possible to change the data store later.

Q. Is there a need in something like TarOptimization for MongoMK?

A. Not quite, there is a periodic background job that purges very old (unused) revisions, but it is very different from the TarPM optimization.

Q. How about data optimization?

A. See previous answer.

Q. Is there a way to purge older revisions?

A. To purge old revisions you can use the JMX call to the RevisionGarbageCollection MBean.

Q. Does that include non anchored contains? i.e.: %contains%

A. Sorry I don’t understand the question. Could you ask it on the Oak mailing list?

Q. Can you paste the wget command for oak-mongo.js in the chat pod please?

A. wget http://s.apache.org/oak-mongo.js

Q. So, in code, what is the best way to avoid concurrent writes?

A. Concurrent writes are OK. I assume you mean conflicting writes. Usually this means you should avoid some features like orderable child nodes.

REPLY: Yes, I meant conflicting writes. Basically, was asking if there is something similar to the ClusterAware class ala crx2.x

A. ClusterAware is deprecated and was replaced with Sling Discovery and its Topology support with a single leader.

Q. Do we have to reindex after we run an offline compaction on TarMK?

A. No

Q. Is it required to add new index for each custom property?

A. Only if the properties are used in a query

Q. Is QueryBuilder still the recommended practice to write queries in AEM?

A. Yes, QueryBuilder is where we recommend to write queries in AEM.

Q. Is it required to manually rebuild index, when new content is added?

A. No need, indexes are automatically kept up to date

Q. How do we get access to oak-run? (I don't find it in crx-quickstart.)

A. You currently have to build it from a source release. In the future we will make binaries available through the standard maven repositories.

Q. What is your updating your page 100s of times a day, is there an upper limit of the size of a revision?

A. That shouldn't be a problem. There is no size limit for revisions, but performance for small transactions is a bit better than performance for large transactions

Q. Is it possible to create an index for ordering by path name?

A. No, this is not supported currently. Feel free to add an issue in Oak with the detailed use case.

Q. Where does ordering happen in this flow? Ex. if you are ordering on a non ordered index, Or if the best index isn’t an ordered index?

A. The ordering happens at query time, taken care of by the query engine, the query engine will sort the results

Q. What is the disk usage implications of many, large indexes? If there are any - is there a (simple?) way to predict the index size?

A. In general, the more and larger indexes you have, the more disk space you will need. The size of the index mainly depends on the number of indexed nodes.

Q. Have you had any success with cross region Mongo Clusters yet? Example, cluster distribution between US and Europe?

> A. We are still working on this kind of deployment with tag aware sharding, but this is not yet production ready.

Q. How does query performance compare between MongoMK and TarMK? What about MongoMK vs CRX2?

> A. This depends on the query definition and whether it uses a property or a Lucene index. For the property index, TarMK is usually more efficient. With a Lucene index, the difference is not that big.

Q. If AEM shuts down unexpectedly does do indexes still get corrupted? With CRX2 if AEM shut down unexpectedly it was not uncommon for Lucene indexes to be corrupted and require rebuild before AEM would run (that could take hours upon hours). Is that resolved with CRX3?

> A. In Oak you can add/remove/reindex indexes at runtime without having to shut down

Q. Does MongoDB implementation helps to make memory management more efficient? There had been issues in the past with TAR

> A. Yes, performance for large repositories (many million nodes) should be better than with the TarPM (according to our tests)

Q. To what extent will Adobe help with Mongo configuration and support?

> A. In general Adobe will help with any questions around how to configure AEM with MongoDB. If you need extended support for MongoDB itself you will have to talk to MongoDB Inc.

Q. Have you thought of having a less slow / in-between default index (or other solution) for queries that result in the "consider creating an index or changing the query" message in the logs - e.g. something like full text on everything - or just something that doesn't traverse too many results (possible?)? In general how difficult is it to learn to create custom indexes where they are required?

> A. We prefer not to index each property, as the indexing overhead would be very high. We are working on a combined full-text and property index (based on Lucene / Solr) that could in theory be extended to support this case, however there are currently no plans to index all properties by default. I think it is important for developers to know how to write efficient queries. This means, you will need to understand how indexes work and how they are configured. We do want to simplify this as much as possible, for example by improving the documentation and the tools needed for this.

Q. Is there a doc to configure MongoMK shard

> A. This is straight forward and you'd just point AEM to mongos instead of mongod.

Q. Do you recommend to use both XPath and SQL query depending on which one best suits the query?

> A. Yes, some queries are easier to express in XPath

Q. Is it recommended to use the MongoDB Java driver to interact with the mongo from the server side (OSGI bundles)?

> A. You should always connect to MongoDB through Oak or some supported tool.

Q. Is oak-run tool only valid when working with MongoMK or will it also work with TarMK?

> A. Oak run console works with both MongoMK and TarMK

Q. Does it work with TokuMX?

> A. I don't think we officially support it (yet), but some people were testing with TokuMK and I'm not aware of any problems

Q. How do you setup a "share-nothing" cluster with Mongo in terms of the datastore?   Is the NAS mount shared between the AEM nodes of the cluster?

> A. The file based data store on a shared storage is still supported. Otherwise you can use the default data store implementation with MongoMK, which stores binaries in MongoDB (shared with all cluster nodes).

Q. Are there index maintenance tools needed for Mongo, like are needed for Tar indexes?   example: tar index merge, etc

> A. No, this is not needed. You may have to compact the database in MongoDB if you delete a lot of data and want to reclaim the disk space.

Q. In which case(s) is the embedded Solr of AEM 6.0 recommended instead of the standard Lucene?

> A. Embedded Solr is only recommended for development in cases where you want to deploy a standalone Solr in production (i.e. to simplify the dev setup)