

XQuery tests with eXist XML DB

Environment:

- Laptop Pentium M 1.6Ghz / 1Gb RAM
- OS: Windows XP Pro
- Java: JVM SUN 1.5
- eXist: snapshot 2005-08-05 on standalone mode, launched with -Xmx512000k
- eXist conf.xml: db-connection cacheSize="500M" / recovery mode="no" sync-on-commit="no" / watchdog output-size-limit="100000"

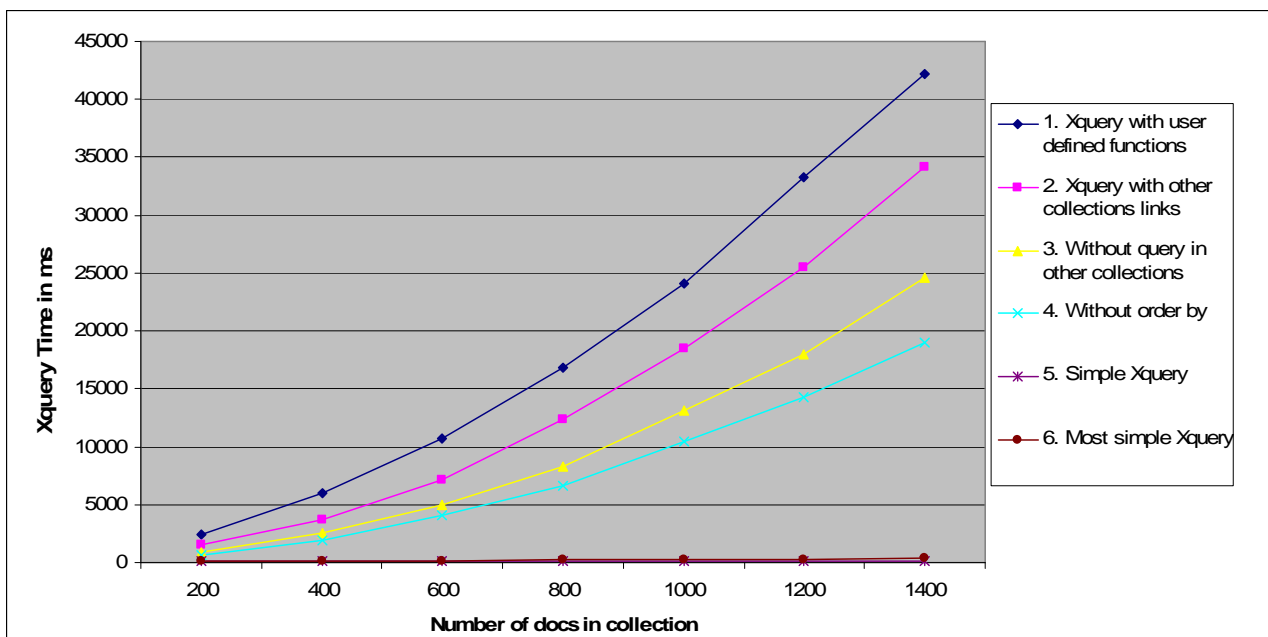
Procedure of the tests:

- The XQueries were done in a collection of documents
- Average of documents: 1.42Ko
- Each time I added new documents to the collection I did a re-index of this collection.
- Each test was done 6 times (the best and the worst results in milliseconds were ignored) and the total result is the average of the 4 normal results.
- I did tests on 200 documents in the collection. I added 200 new docs each time (till 1400 docs)
- I did tests on 6 different XQueries. From the easier to the most complicated I could imagine. (The details of the XQueries are at the end of this document)
- I used the eXist Java client to launch the queries. The results are the "execution time" provided with the new java client.

Results of the tests:

Number of docs in the collection :	200	400	600	800	1000	1200	1400
1. Xquery with user defined functions	2384	5989	10756	16781	24078	33210	42214
2. Xquery with other collections links	1493	3688	7098	12371	18542	25544	34117
3. Without query in other collections	849	2526	4942	8300	13184	17989	24540
4. Without order by	681	1975	4026	6595	10453	14228	19013
5. Simple Xquery	65	100	133	136	95	100	115
6. Most simple Xquery	75	125	186	196	275	291	338

The results are in milliseconds



Observations:

1. If we try to retrieve some information from documents in a collection containing more than 400 docs it is really really slow...
2. The tests were done on entire collections. When you query a single document the time stay normal.
3. The XQuery time in milliseconds has nothing to do with the amount of datas eXist should return. For example, the first XQuery (the slowest) returns only 10 results each time and the 5th XQuery (the fastest) returns an XML up to 1.5Mb!
4. The time of the queries is not linear with the number of docs.
5. I know that eXist is not a relational DB but it should be able to retrieve information a bit faster even on cross collections...
6. The difference between an "ordered by" query and a "non-ordered" query is more than 3.5 sec. for a collection of 1'000 documents!
7. The difference between a query in a single document or a query that call other collections (with a simple XPath expression) is more than 5 sec for 1'000 docs. It means eXist need 5 sec. to make 2 joints on other collections on 1'000 queried documents.

Questions:

1. What do you think about these tests? Are they realistic? Are they a representation of what the eXist database should do?
2. What do you think about the time of queries? For example, the XQuery number 4 is only returning some information on each doc in the collection. It takes 10 sec to list 1'000 documents! Is that normal? (the xquery source is at the end of this document)
3. We use such XQueries to query lists of document from the DB. Isn't eXist done to list docs in collections?
4. Are my XQueries totally wrong written?
5. Is the problem coming from Windows? Do you have better perfs on Linux? Even the perfs are divided by 2 on Linux, you don't find it's really slow?
6. What's your reaction with this? (tests, speed of eXist, etc.)

Typical XML document in the collection:

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <idDoc>5042c554db8bb8b</idDoc>
  <computerDatas>
    <droits>
      <droit niveau="1">
        <entreprise id="2" all="0">
          <user>1</user>
          <role>-1</role>
        </entreprise>
      </droit>
    </droits>
  </computerDatas>
  <versions>
    <version number="1">
      <uploadDate>2005-07-01</uploadDate>
      <uploadTime>16:36:11</uploadTime>
      <uploaderId>1</uploaderId>
      <sizeBytes>128000</sizeBytes>
      <documentName>ipvr_html.ppt</documentName>
      <documentMimeType>application/mspowerpoint</documentMimeType>
      <downloadNumber>1</downloadNumber>
      <description>Premier essai du document. Voici sa description pour la première version.</description>
    </version>
    <version number="2">
      <uploadDate>2005-07-01</uploadDate>
      <uploadTime>16:50:27</uploadTime>
      <uploaderId>1</uploaderId>
      <sizeBytes>88576</sizeBytes>
      <documentName>ipvr_html.ppt</documentName>
      <documentMimeType>application/mspowerpoint</documentMimeType>
      <downloadNumber>2</downloadNumber>
      <description>Je n'avais pas sauvée celle d'avant, celle-ci est complète.</description>
    </version>
  </versions>
  <general>
    <description>Powerpoint de résumé sur le HTML nécessaire à la création du site test.</description>
    <author>
      <id>1</id>
      <name/>
      <firstname/>
      <email/>
    </author>
  </general>
</document>
```

XQueries used during the tests:

1. Xquery with user defined functions

```

import module namespace droits="droits.uri" at
"http://localhost/ouistites/traitements/scripts/xquery/functions/droits.xqm";
let $idEntreprise := 2,
    $idRessource := 12,
    $idRole := -1,
    $max := 10,
    $start := 1,
    $results :=

        for $document in
collection("/db/cms_dev/contenus/gestionDocuments/documents/")/document
    let $level := droits:getLevelOfRessource($document, $idEntreprise, $idRole,
$idRessource)
        let $author :=
collection("/db/cms_dev/contenus/droits/ressources/")/ressource[idRessource =
$document/general/author/id]
            let $last_version := max($document/versions/version/@number)
            let $uploader :=
collection("/db/cms_dev/contenus/droits/ressources/")/ressource[idRessource =
$document/versions/version[@number = $last_version]/uploaderId/text()]
                order by $document/versions/version[@number =
$last_version]/uploadDate/text() descending, $document/versions/version[@number =
$last_version]/uploadTime/text() descending
            return
                if ($level < 100000)
                then

<document>
  <id>{$document/idDoc/text()}</id>
  {
    if ($document/general/author/id = '-1') then
      <author>
        <name>{$document/general/author/name/text()}</name>
        <firstname>{$document/general/author/firstname/text()}</firstname>
      </author>

      <author>
        <name>{$author/coordonnees/nom/text()}</name>
        <firstname>{$author/coordonnees/prenom/text()}</firstname>
      </author>
    }
    <description>{$document/general/description/text()}</description>
    <lastUploader>
      <name>{$uploader/coordonnees/nom/text()}</name>

      <firstname>{$uploader/coordonnees/prenom/text()}</firstname>
    </lastUploader>
    <lastUploadDate>{$document/versions/version[@number =
$last_version]/uploadDate/text()}</lastUploadDate>
    <lastUploadTime>{$document/versions/version[@number =
$last_version]/uploadTime/text()}</lastUploadTime>
    <lastSizeBytes>{$document/versions/version[@number =
$last_version]/sizeBytes/text()}</lastSizeBytes>
    <lastDocumentName>{$document/versions/version[@number =
$last_version]/documentName/text()}</lastDocumentName>

    <lastDocumentMimeType>{$document/versions/version[@number =
$last_version]/documentMimeType/text()}</lastDocumentMimeType>

```

```

        <lastDownloadNumber>{$document/versions/version[@number =
$last_version]/downloadNumber/text()}</lastDownloadNumber>
        <levelRessource>{$level}</levelRessource>
    </document>

    null,
    $count := count($results),
    $send := if($count >= $max) then $max else $count
return
<documents>
{
for $i in $start to $end
return $results[$i]
}
</documents>

```

2. Xquery with other collections links

```

let $idRessource := 12
for $document in collection("/db/cms_dev/contenus/gestionDocuments/documents/"/)document
let $author := collection("/db/cms_dev/contenus/droits/ressources/"/)ressource[idRessource =
$document/general/author/id]
let $last_version := max($document/versions/version/@number)
let $uploader := collection("/db/cms_dev/contenus/droits/ressources/"/)ressource[idRessource =
$document/versions/version[@number = $last_version]/uploaderId/text()]
order by $document/versions/version[@number = $last_version]/uploadDate/text() descending,
$document/versions/version[@number = $last_version]/uploadTime/text() descending
return
    <document>
        <id>{$document/idDoc/text()}</id>
        {
            if ($document/general/author/id = '-1') then
                <author>
                    <name>{$document/general/author/name/text()}</name>
                    <firstname>{$document/general/author/firstname/text()}</firstname>
                </author>
            else
                <author>
                    <name>{$author/coordonnees/nom/text()}</name>
                    <firstname>{$author/coordonnees/prenom/text()}</firstname>
                </author>
        }
        <description>{$document/general/description/text()}</description>
        <lastUploader>
            <name>{$uploader/coordonnees/nom/text()}</name>

            <firstname>{$uploader/coordonnees/prenom/text()}</firstname>
        </lastUploader>
        <lastUploadDate>{$document/versions/version[@number =
$last_version]/uploadDate/text()}</lastUploadDate>
        <lastUploadTime>{$document/versions/version[@number =
$last_version]/uploadTime/text()}</lastUploadTime>
        <lastSizeBytes>{$document/versions/version[@number =
$last_version]/sizeBytes/text()}</lastSizeBytes>
        <lastDocumentName>{$document/versions/version[@number =
$last_version]/documentName/text()}</lastDocumentName>
        <lastDocumentMimeType>{$document/versions/version[@number =
$last_version]/documentMimeType/text()}</lastDocumentMimeType>
        <lastDownloadNumber>{$document/versions/version[@number =
$last_version]/downloadNumber/text()}</lastDownloadNumber>
    </document>

```

3. Without query in other collections

```

for $document in collection("/db/cms_dev/contenus/gestionDocuments/documents")/document
let $last_version := max($document/versions/version/@number)
order by $document/versions/version[@number = $last_version]/uploadDate/text() descending,
$document/versions/version[@number = $last_version]/uploadTime/text() descending
return
  <document>
    <id>{$document/idDoc/text()}</id>
    <description>{$document/general/description/text()}</description>
    <lastUploadDate>{$document/versions/version[@number =
$last_version]/uploadDate/text()}</lastUploadDate>
    <lastUploadTime>{$document/versions/version[@number =
$last_version]/uploadTime/text()}</lastUploadTime>
    <lastSizeBytes>{$document/versions/version[@number =
$last_version]/sizeBytes/text()}</lastSizeBytes>
    <lastDocumentName>{$document/versions/version[@number =
$last_version]/documentName/text()}</lastDocumentName>
    <lastDocumentMimeType>{$document/versions/version[@number =
$last_version]/documentMimeType/text()}</lastDocumentMimeType>
    <lastDownloadNumber>{$document/versions/version[@number =
$last_version]/downloadNumber/text()}</lastDownloadNumber>
  </document>

```

4. Without order by

```

for $document in collection("/db/cms_dev/contenus/gestionDocuments/documents")/document
let $last_version := max($document/versions/version/@number)
return
  <document>
    <id>{$document/idDoc/text()}</id>
    <description>{$document/general/description/text()}</description>
    <lastUploadDate>{$document/versions/version[@number =
$last_version]/uploadDate/text()}</lastUploadDate>
    <lastUploadTime>{$document/versions/version[@number =
$last_version]/uploadTime/text()}</lastUploadTime>
    <lastSizeBytes>{$document/versions/version[@number =
$last_version]/sizeBytes/text()}</lastSizeBytes>
    <lastDocumentName>{$document/versions/version[@number =
$last_version]/documentName/text()}</lastDocumentName>
    <lastDocumentMimeType>{$document/versions/version[@number =
$last_version]/documentMimeType/text()}</lastDocumentMimeType>
    <lastDownloadNumber>{$document/versions/version[@number =
$last_version]/downloadNumber/text()}</lastDownloadNumber>
  </document>

```

5. Simple XQuery

```

for $document in collection("/db/cms_dev/contenus/gestionDocuments/documents")/document
return $document

```

6. Most simple XQuery

```

for $document in collection("/db/cms_dev/contenus/gestionDocuments/documents")/document
return $document/idDoc

```