Experiments on Fragmentation

Hao, Wei

December 6, 2017

Contents

1 E2017120401: Baseline Performance Evaluation

1.1 Purpose

To understand the baseline performance by using the queries in ADBIS submission for XMark600.

1.2 Experiment Design

We run a BaseX server *Server* for processing XPath queries on specified databases. Server starts by the following command on HaoDesk.

```
java -Xmx4g -xms2g -cp BaseX897.jar org.basex.BaseXServer
```

Note the databases in Server are NOT in main memory mode.

We run a java program *Japp* (that is the class `basex.ORIG` in repository/src/fragmentation/java/basex) on HaoDesk in charge of sending an input query to Server via local network and saving results returned from Server. An input query Query that will be processed in Japp is first rewritten into the following XQuery expression:

```
for $node in db:open('xmark600')Query return $node
```

The results of the input query are stored either in memory or on disk depending on Japp's settings. When in memory, the results will be then discarded after the experiments, while on disk, the results will be preserved in files. One more thing, the maximum available memory for Japp was set to 12 GB.

1.3 Settings

- **Hardware** HaoDesk (see A.1.1)
- **Software** BaseX 6.8.7, Java 1.8.0_151(x64).
- **XML Dataset** XMark600.xml (see Table 2), from which a BaseX databases 'xmark600' is created in a BaseX server using command:
  `create db xmark600 xmark600.xml`
- **Queries** xm1.org – xm6.org (see Table 3).

1.4 Experiment Results

Timing The execution time is measured in Japp. The time period between starting sending a query and finishing receiving the results is measured as execution time. Each query is evaluated 5 times.

Process Results We disgard the execution time of the first run and take the average of the rest as the final execution time listed in Table 1.

Original Experiment Data

Table 1: Experiment Results of E2017120401.

query	storage	time(s)	result size
xm1.org	disk	3191.60	60,048,845,586
	memory	N/A	
xm2.org	disk	0.01	0
	memory	0.01	
xm3.org	disk	71.25	922,270,281
	memory	73.34	
xm4.org	disk	113.05	1,583,959,305
	memory	113.84	
xm5.org	disk	83.59	989,346,990
	memory	88.75	
xm6.org	disk	78.42	1,351,708,787
	memory	78.90	

All the original results containing execution time of queries and scripts used in the experiments are stored to `experiments/E2017120401` relative to the current folder that stores this report.

Note: The result of xm2.org is always empty. This is because there is no `incategory` node with attribute that has the content of `category52`.

1.5 Observations

- **Storage has small influence on execution time**

  We noticed one thing that the execution time is pretty similar for all the available queries. This is because the bottleneck is on the worker's side but not on the master's side. For example, for xm4.org, it takes 113s to receive about 1500 MB data, i.e. around 13.36 MB/s, which is much slower than the maximum speed of both memory and disk. Thus, the performance are much similar. We also notice that for some queries such as xm3.org and xm5.org, in-memory case is even a bit slower than on-disk case, one possible explanation is that the time was taken by calling System.gc().

- **The execution time is steady**

  Compared with the ADBIS study, the execution time of each run is more steady. Our explanation to this result is that for very large scale of data, the fluctuation has a weaker influence on the execution time (increased from milliseconds to seconds).

2 E2017120501: Why xm2.org on XMark600 returns nothing?

2.1 Check With BaseX

To find the reason, I used count() function tested the following queries, Q1–Q5 (Q1 is the original) over XMark600 using HaoLab.

```
Q1:/site//incategory[./@category="category52"]/parent::item/@id
Q2:/site//incategory[./@category="category52"]
Q3:/site//incategory[@category]
Q4:/site//incategory[contains(@category, "52")]
Q5:/site//incategory[contains(@category, "category52")]
```

The results are listed as below:

query	count(Q1)	count(Q2)	count(Q3)	count(Q4)	count(Q5)
result	0	0	49,514,754	2,720,824	895,154

We then outputted the results of Q4 and Q5. After checking the results, there was no node that matchs "category52" found. The nodes found were `interest` nodes that have the contents started with `category52` such as `<incategory category="category526763"/>`.

2.2 Check Without BaseX

From the above results, it is clear that no "incategory" node has an attribute with content of "category52". To make sure this conclusion, we did another test without using BaseX.

The test is to find string `category="category52"` by reading and comparing each line in the xmark600.xml. The java code is listed as below:

```
public static void main(String[] args) throws Exception {
    FileReader reader = new FileReader(args[0]);
    BufferedReader br = new BufferedReader(reader);
    String str = null;
    int lineno = 0;
    while ((str = br.readLine()) != null) {
        lineno++;
        if (str.contains(args[1]))
            System.out.println("Found in line " +
              lineno + ": " + str);
    }
    br.close();
    reader.close();
}
```

We executed it by the following command.

```
java -cp test.jar basex.Finder xmark600.xml category=\"category52\"
```

Table 2: Statistics of XMark datasets

key	# elements	# attributes	# context	total nodes	file size
xmark1.xml	1,666,315	381,878	1,173,732	3,221,925	113.06 MB
xmark600.xml	1,002,327,042	229,871,111	705,824,967	1,938,023,120	66.99 GB

Then, what we found are listed as below.

```
Found in line 424449141: <interest category="category52"/>
...
Found in line 658701472: <interest category="category52"/>
```

(The full results are stored in the file "`experiments/E2017120501/results.txt`").

There is no `incategory` node in xmark600 found with attribute category="category52".

Therefore, the results of xm2.org E2017120401 in report are correct.

A Environments

A.1 Computers

A.1.1 HaoDesk

CPU: Intel Core (TM) i7 3930K@3.2 GHz (turbo to 3.8 GHz), 6 cores 12 threads
Memory: 32 GB DDR3 1333 GHz
Disk: 256GB SSD + 4TB HDD
Windows 7 Professional SP1 64bit

B DataSets

B.1 XMark Dataset

The XMark datasets are listed in Table 2.

C XPath Queries

C.1 For XMark Dataset

The queries for XMark datasets are listed in Table 3.

Table 3: Original XPath queries For XMark datasets.

key	query
xm1.org	`/site//*[name(.)="emailaddress" or name(.)="annotation" or name(.)="description"]`
xm2.org	/site//incategory[./@category=”category52”]/parent::item/@id
xm3.org	/site//open_auction/bidder[last()]
xm4.org	/site/regions/*/item[./location=”United States” and ./quantity > and ./payment=”Creditcard” and ./description and ./name]
xm5.org	/site/open_auctions/open_auction/bidder/increase
xm6.org	/site/regions/*[name(.)=”africa” or name(.)=”asia”]/item/description/parlist/listitem