1 Experiments

1.1 E2017120401: Base Performance Evaluation

1.1.1 Purpose

1.1.2 Settings

Hardware: HaoDesk
Software: BaseX 6.8.7, Java 1.8.0_151.

XML Dataset

An XMark dataset xmark600.xml sized 66.9 GB. A BaseX databases 'xmark600' is created by the server using command:

```
create db xmark600 xmark600.xml
```

Queries

xm1 – xm6 used in ADBIS submission attached at the bottom.

1.1.3 Experiment Design

A BaseX instance Server first runs in server mode started by the following command on HaoDesk.

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

Note the databases in Server are all not in main memory mode.

Then, a java program APP runs on COM in charge of sending an input query to Server via local network and saving results returned from Server to memory(short for mem) or disk depending on settings. An input query Query that will be processed in APP is first rewritten into the following XQuery expression:

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

The results are stored either in memory or on disk depending on APP's settings: a) memory means the results are stored in memory and then discarded after the experiments. b) disk means the results are stored in disk and will be preserved after the experiments.

1.1.4 Experiment Results

Timing The execution time is measured in APP. 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 removed the results of the first run and take the average of the rest as the final execution time listed in Table 1.

Note: The result of xm2.org is always empty (still under investigation).

1.1.5 Observations

Storage has small influence on execution time

I noticed one thing that the execution time is pretty similar for xm3.org and xm5.org. This is because the bottleneck is on the worker's side but not on the

Table 1: Experiment Results.

query	storage	time(s)	result size
xm1.org	disk	3257.13	60,048,845,586
	memory	N/A	
xm2.org	disk	0.01	0
	memory	0.00	
xm3.org	disk	105.11	922,270,281
	memory	107.21	
xm4.org	disk	126.00	1,583,959,305
	memory	112.32	
xm5.org	disk	112.94	989,346,990
	memory	106.88	
xm6.org	disk	78.08	1,351,708,787
	memory	76.21	

master's side. For example, for xm3.org, it takes 105s to receive about 880MB data, i.e. around 8MB/s, which is much slower than the speed of both memory and disk. Thus, the performance are much similar.

The execution time is steady

Compared with the ADBIS study, the execution time is much more steady. My explanation to this result is that due to the large scale of data, the fluctuation has a weaker influence on the execution time (milliseconds -¿ seconds). For you reference, the results and summary are saved in matsu-lab99:/home2/hao/fragmentation/20171202) By the way, since the results are still not fully correct, I will commit my code later after solve them.

2 Environment

2.1 Computers

2.1.1 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
System: Windows 7 Professional SP1 64bit.