

## 1 E2017120501: Data Partitioning on Fragmented XML Data

### 1.1 XQuery queries

### 1.2 Settings

The setting of the experiments are listed below.

item	Description
XML Data	xmark600.xml
Fragmentation	maxisize=4M, Ns=16
BaseX	set main memory true
Computers	master=HaoDesk, workers: matsu-lab00 – matsu-lab15
# of partions(P)	1, 2, 3, 4, 6 8

### 1.3 Confirming Correctness

The number of hit nodes, the order of hit nodes and results size have been checked to be correct. All the successfully evaluated queries have the same number of hit nodes and correct orders. Some may be different in results size, such as xm3a.dps. Its results in original was 922,270,281, but in this experiment, it is 883,253,777. This dramatical difference is caused by line-break. The previous experiments were done on Windows, while the current is on Linux. Since BaseX using '\r\n' on Windows, while '\n' on Linux. There is one byte difference for each new line. The query has 6,502,751 hit nodes. Thus,  $(922,270,281 - 883,253,775) / 6502751 = 6$ , (Node I found two extra '\n' in the results so  $883,253,777 - 2$ ) where there is exact 6 lines of each results, such as:

```
<bidder>
  <date>08/04/1999</date>
  <time>11:15:36</time>
  <personref person="person17793"/>
  <increase>7.50</increase>
</bidder>
```

### 1.4 Discussion on Queries

- xm1.dps Since the results of xm1.dps are larger than the memory of HaoDesk. This query was not evaluated. This is a design choice about how to process the results. Based on my previous evaluation, one way is to stored the unordered intermediate results in files and then concatenate these files.

- xm2c.dps Since there is no results. I did not test this query. I propose to make a minimal change to the query, such as change "category52" to, such as "category324329", which exists in XMark600,

- xm3a.dps Here are the results (3x runs and picked the best one. FIY: xm3.org = 63.32s

query	P=1	P=2	P=3	P=4	P=6	P=8
prefix	$\approx 0.5s$					
suffix	113.2s	58.5s	30.1s	20.2s	15.8s	13.5s
merge	$\approx 2s$					

Note: for prefix and merge, the costs are similar.

Although we can get sped up by using 16 computer and 128 threads at most, the results are obvious much slower than expected. At least, when P=1, 16 computer with main memory mode is even slower than the original that was not in main memory node.

- xm3b.dps, xm4a.dps, xm4b.dps and xm6.dps Failed to evaluate them with datapartitioning. This is because on some workers, such as on matsu-lab50, there is not hit node in the suffix query. Therefore, there were some error message shown such as: "database 'xmark600\_16\_4M\_tmp' has no node with pre value 5." when executing a suffix query, where the database was as "`<root> <part> ...</part> </root>`". In the query, "pre value 5" in refers to the second "part" node. However, since there is no second "part" node, the error occurred. So the reason to this error is because a prefix query, such `/site/*` or `/site/./item`, returns not enough nodes such that we cannot make a merged tree to have at least one hit nodes for each partition of the merged tree.

- xm5a.dps and xm5b.dps Both of the queries can be evaluated and the results are correct. But the efficiency is a problems. Both query took longer time than the original (reported a couple of days ago).

In general there are three problems or design choices:

- xm1.dps returns larger results than memory.
- some queries cannot be successfully evaluated.
- some successfully evaluated queries are too slow in performance.

One more thing. A important method that affect the performance is `baseX.PreValueReceiver.process(InputStream input)`, which is used to receive and parse the results of suffix query, (i.e. PRE value + content). It takes the results of suffix query and returns an `QueryResultsPre` instance with a list of PRE values and a list of string content (of the same size). Some tests were done to evaluate the parsing speed. In the experiment, it took 465 ms to parse 52,757 KiB data with 704,430 nodes. i.e. it can process 100 MiB string data per second, which basically reach the maximum network speed and thus should be sufficient for receiving data.