Name: Kiran Nayak (ID: 015271332)

Wavelet

Database Architecture CECS 521 Final Project

1. Implementation:

This project intends to implement the wavelet concept in JAVA and compare the obtained result with the expected results. There implementation enables user to either Query or Update the values of the wavelet. As we know wavelets provide quick approximate results on a large data, the project tries to find the real costs in terms of time in nanoseconds for the Query and update. Important point to note is , this is a Haar Wavelet (h=0.5,0.5 ; g=0.5,-0.5) with value of coefficients equal to 50.

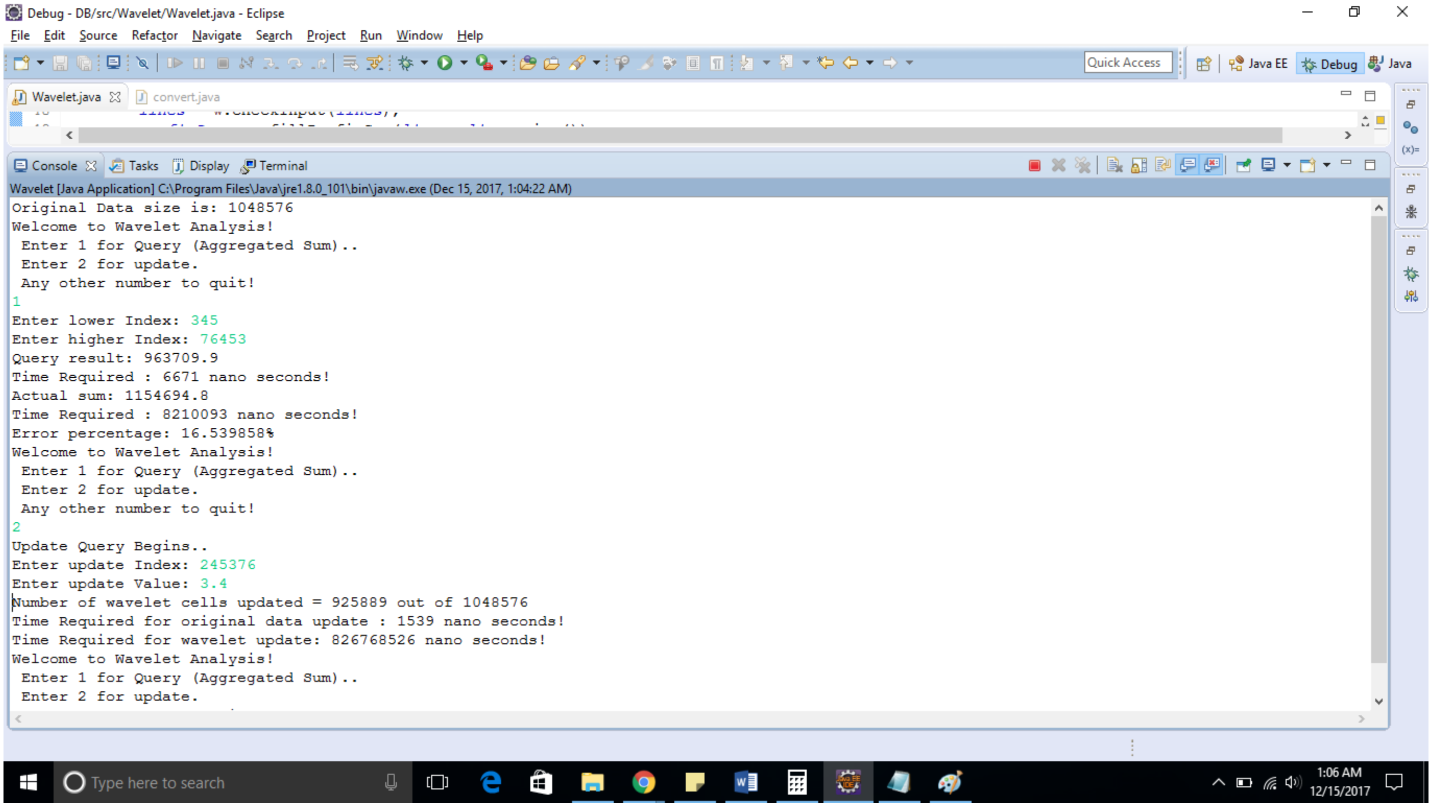
For Query it implements the Aggregated sum query, where it asks the user for the upper and the lower index between which it has to query. Once the indexes have been entered, the program calculates the prefix, then calculates the wavelet and the final data array on which the query can be implemented. The program then starts the timer and calculated the time for query on original data. The timer is them stopped to provide time in nanoseconds for the time for original data. Similarly, it calculates time while query is implemented in the wavelet. Hence both the times are displayed, by which we can easily compare time between the two. The wavelet is finally written to a file as output file called output.txt with computed wavelet.

For update, the project asks user for an index which needs to be updated and the value. Once these two are entered by the user, the program analyses which elements have to be updated in the prefix sum. Only the needed elements are updated and the rest are keep as it is, to save time. Later the program starts a timer and prints the time required for the original array to be updated and the time required for the wavelet to be updated. Hence, we can directly compare the cost of update in both cases.

2. Screen Captures

Following are the actual screenshots, when the program was run on eclipse with a million records. The input file was a csv file containing average Global temperatures from 1750 till 2015 month wise taken multiple times, from Kaggle (website) which provides various datasets. The input has to be in csv format.

Name: Kiran Nayak (ID: 015271332)



Wavelet [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Dec 15, 2017, 1:04:22 AM)

```
Original Data size is: 1048576
Welcome to Wavelet Analysis!
 Enter 1 for Query (Aggregated Sum)..
 Enter 2 for update.
 Any other number to quit!
1
Enter lower Index: 345
Enter higher Index: 76453
Query result: 963709.9
Time Required : 6671 nano seconds!
Actual sum: 1154694.8
Time Required : 8210093 nano seconds!
Error percentage: 16.539858%
Welcome to Wavelet Analysis!
 Enter 1 for Query (Aggregated Sum)..
 Enter 2 for update.
 Any other number to quit!
2
Update Query Begins..
Enter update Index: 245376
Enter update Value: 3.4
Number of wavelet cells updated = 925889 out of 1048576
Time Required for original data update : 1539 nano seconds!
Time Required for wavelet update: 826768526 nano seconds!
Welcome to Wavelet Analysis!
 Enter 1 for Query (Aggregated Sum)..
 Enter 2 for update.
```

3. Results

When the program was run for a million records (1048576), following results were obtained:

1. Query:
   Sum (A[345 : 76543])

   Cost Required for Query on Original Data = 8210093 ns (nano seconds).
   Cost Required for Query on Wavelet Data = 6671 ns
   With just an error rate of 16.54%

2. Update:
   Update A[245376] with value 3.4

   Cost Required for Update on Original Data = 1539 ns
   Cost Required for Query on Wavelet Data = 826768526 ns

   Updated Cells : 925889 out of total 1048576 cells; which means 122,687 cells were unchanged!

4. Conclusion

The findings were clear that Wavelets are great to find an approximate answer to a query very quickly as we saw in above result 6671ns as opposed to a costly query of 8210093ns on original data. For updates, it was exactly opposite wavelets are very costly on updates as compared to updates on original data.

Therefore, we can conclude, wavelet is a great technique for quick approximate query and is good only if we do very few updates.

5. Submitted Content:

1. Report

2. Executable Code

3. Input File used

4. Output file produced