Enron Data Revisited - Neighborhood Queries with FastBit Win over Popular Commercial Database System

Kurt Stockinger, Doron Rotem, Arie Shoshani, Kesheng Wu
Computational Research Division
Lawrence Berkeley National Laboratory
University of California
1 Cyclotron Road, Berkeley, CA 94720, USA

1 Introduction

In our previous work we analyzed the Enron data set and showed how FastBit [1] can be used for speeding up multi-dimensional queries [3]. We demonstrated that our approach is between a factor of 100 to 1000 faster than one of the most commonly used open-source database systems called MySQL. In this article we evaluate the performance of *neighborhood queries* (nested queries) that are important for studying communication patterns and the flow of information. Consider, for instance, that one might want to know all the recipients of emails that were sent by person P. However, some people never communicate directly with certain people and rely on either messengers or the fact that emails get forwarded to them. In SQL, this kind of query is expressed as follows:

```
SELECT recipient FROM EnronDB WHERE sender IN
  (SELECT DISTINCT recipient
   FROM EnronDB WHERE sender = 'P');
```

This query reveals all the recipients that received a message from person P via one intermediate person. We call this a neighborhood query with nesting level 1. In order to find all recipients that received messages from person P via two intermediate persons, we would need to introduce an additional nesting level to our neighborhood query, such as:

```
SELECT recipient FROM EnronDB WHERE sender IN
  (SELECT DISTINCT recipient
   FROM EnronDB WHERE sender IN
    (SELECT DISTINCT recipient
     FROM EnronDB WHERE sender = 'P'));
```

In this article we will evaluate the performance of multi-level neighborhood queries with FastBit and compare it with a popular commercial database system that we call CDBS. For the peformance evaluation we use CDBS with the default optimization option.

2 Data

For our performance measurements we used the Enron data set that was prepared by [2]. Since the data was originally stored in MySQL-format, we had to convert it and import to the format that is understood by CDBS. Like in our previous work [3] we duplicated the size of the data by a factor of 10. All the data is stored in one database table with 20 million rows and 9 columns, such as *sender*, *recipient*, *message ID*, *message folder*, *message subject*, *time* etc. Besides the data stored in CDBS, we stored each column in a separate raw binary file and built a bitmap index for each of them. The total size of the raw data is 3,780 MB. The total size of the bitmap indices is 170 MB.

3 Performance Results

Figure 1 shows the performance of a 3-level neighborhood query. For each query we have randomly chosen the sender. Thus, the number of recipients is different for each query. We can see that FastBit is on average a factor of 16 faster than the commercial database system.

One of the reasons for CDBS to be significantly slower than FastBit is the clustering. Most commercial database systems cluster the columns of a record together. The is very efficient for write-optimized databases with frequent updates. However, the disadvantage of this clustering is that all the column values have to be read even if only one column needs to be retrieved. In order to show this effect, we removed 7 columns of our table and only stored those columns that are needed for our queries, namely the *sender* and the *recipient*. This test case is equivalent to building a materialized view on two columns.

Figure 2 shows the results for the same query for two different CDBS tables. One table contains all 9 columns (CDBS-9), whereas the other table only contains 2 columns (CDBS-2). As we can see, the performance for CDBS is

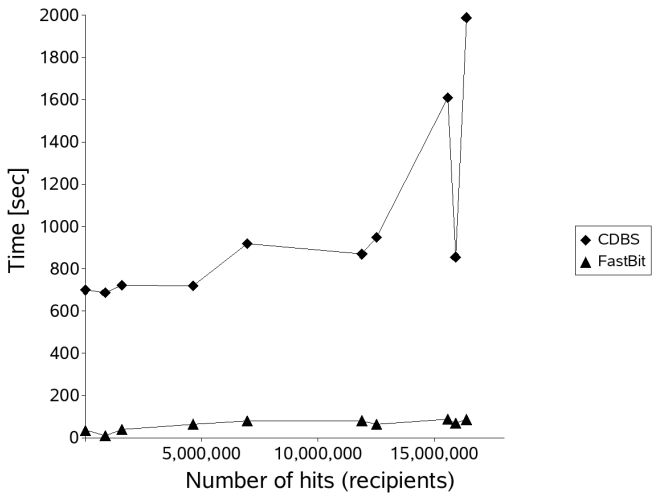

Figure 1. Performance of 3-level neighborhood queries: "Retrieve the indirect recipients of all emails that were sent by person P".

significantly faster than for CDBS-9 since only a subset of the data has to be read. However, FastBit is still about a factor of 3 faster than CDBS 3. Note the logarithmic scale on the y-axis.

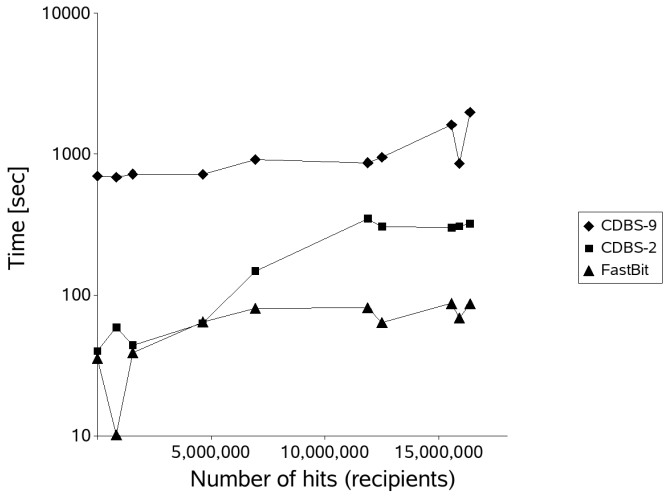

Figure 2. Performance of 3-level neighborhood queries: "Retrieve the indirect recipients of all emails that were sent by person P".

As we can see in Figure 1, the number of recipients is quite large and might not be meaningful. However, in a real life situation, some analysts might be interested in emails that are sent within a certain time period after a given event. We thus evaluate the performance of multi-level neighborhood queries with time constraints. Since these queries involve three attributes (sender, recipient and date), we built a table with only these columns (corresponding to a materialized view on three columns). Note, this is the optimal case for CDBS.

Figure 3 shows the performance results of 3-level neighborhood queries over a time interval of 3 months. We see that in this case, CDBS performs slightly better than FastBit. However, we see that CDBS takes more time for evaluating queries with 0 hits as well as more than 1 million hits. The average query processing time for CDBS and FastBit is 5 and 1.8 seconds, respectively. As the time window increases, the performance gain of FastBit over CDBS is even more significant (see Figures 4 and 5). With time intervals of 6 and 12 months, FastBit is on average a factor of 5.5 and 6.4 faster than CDBS.

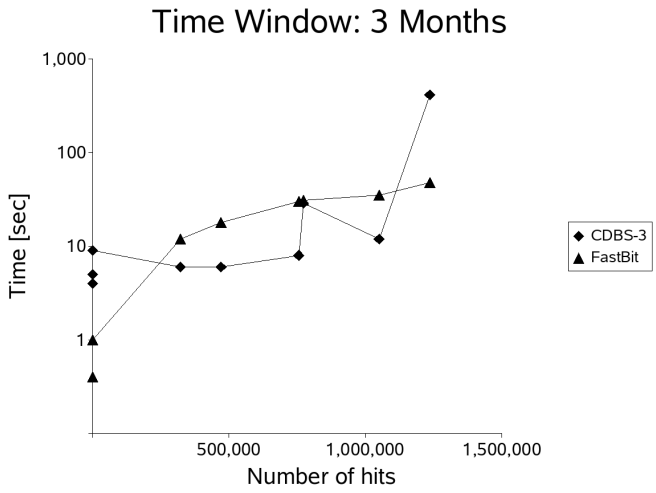

Figure 3. Performance of 3-level neighborhood queries: "Retrieve the indirect recipients of all emails that were sent by person P" over a time window of 3 months.

4 Summary

In this article we evaluated the performance of neighborhood queries that are typical for studying complex communication patterns. These patterns are very common in situation where people do not community with each other directly but through messengers. We have shown that FastBit significantly outperforms the commercial database system.

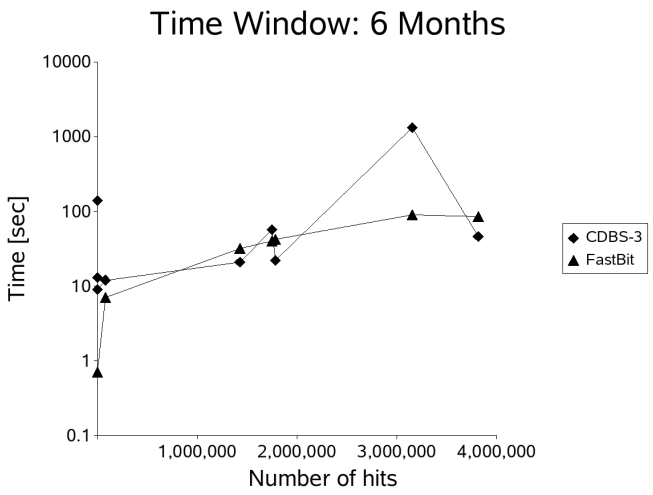

Figure 4. Performance of 3-level neighborhood queries: "Retrieve the indirect recipients of all emails that were sent by person P" over a time window of 6 months.

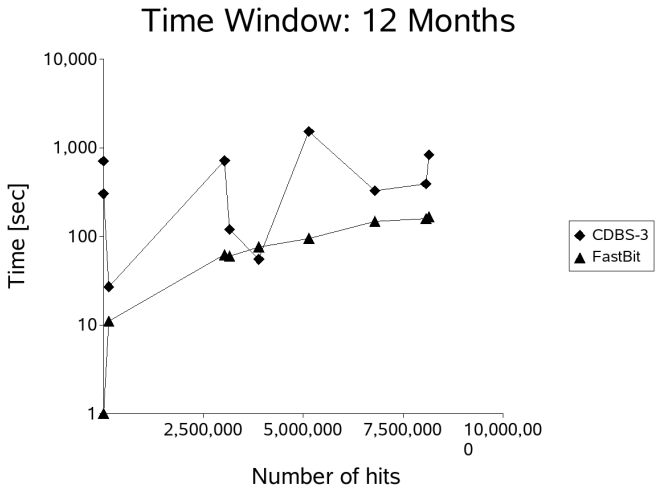

Figure 5. Performance of 3-level neighborhood queries: "Retrieve the indirect recipients of all emails that were sent by person P" over a time window of 12 months.

Acknowledgment

The work was funded by the Department of Homeland Security. We also want to thank Mark Dedlow from Berkeley Lab for his assistance on CDBS.

References

[1] FastBit, `http://sdm.lbl.gov/fastbit`. April 2006.

[2] J. Shetty, J. Adibi, The Enron Email Dataset, Database Schema and Brief Statistical Report, Retrieved from `http://www.isi.edu/~adibi/Enron/Enron_Dataset_Report.pdf`, Jan. 2006

[3] K. Stockinger, D. Rotem, A. Shoshani. K. Wu, Analyzing Enron Data: Bitmap Indexing Outperforms MySQL Queries by Several Orders of Magnitude, *Technical Report, LBNL-59437*, Berkeley Lab, Berkeley, California, USA, Jan. 2006.