Energy Efficient Access in Multiversion Broadcast Environment

Oleg Shigiltchoff
University of Pittsburgh
oleg@cs.pitt.edu

Panos K. Chrysanthis
University of Pittsburgh
panos@cs.pitt.edu

Evaggelia Pitoura
University of Ioannina
pitoura@cs.uoi.gr

Motivation. Broadcasting provides an efficient means for disseminating information in both wired and wireless settings, especially for popular data items. Multiversion (MV) data broadcast, i.e., data broadcast in which more than one value is broadcast per data item, has the advantage of allowing more client transactions to read consistent data and complete their operation successfully. Naturally, the MV broadcast can satisfy both *Historical* queries in which a client accesses many versions of the same data item and *Snapshot* queries in which a client accesses different data items of the same version. Furthermore, multiple versions increase clients' tolerance to network disconnections that are common in wireless communications.

Contribution. In this paper [4], we expand our previous work on MV broadcasts [1, 2] to support efficient selective tuning. Selective tuning is important for energy constraint mobile devices since it enables the client to be active only when data of interest appear on the broadcast. The rest of the time the client stays in doze mode with its antenna powered down. In its simplest form, we adopt (1,1)-Indexing [3] and organize multiversion data in buckets, so that the average *Energy Consumption* is significantly reduced while minimally affecting the average *Access Time*.

MV (1,1)-Indexing. In MV (1,1)-Indexing, an index is broadcast at the beginning of every a broadcast cycle. The index is organized as an ordered set of entries of the form (*Did*, *Vno*, *bucketId*), where *Did* is the data id, *Vno* is the version number, and *bucketId* is the bucket offset of that data item on broadcast (bucket is the broadcast unit).

Grid Bucket Organization. In constructing the data segment of the broadcast, there are two basic questions: (1) how to map *Did* and *Vno* into *bucketId*; and (2) in which order should the buckets be broadcast to satisfy different sets of clients. We propose a general model of organization of multiversion data in buckets for the broadcast called the *Grid* bucket organization.

A set of multiversion data can be represented as a two-dimensional array, where dimensions correspond to *Vno* and *Did*, and the array elements are the data values *Dval*. To distribute data among buckets we convert the two-dimensional array of data and versions into a two-dimensional array of Grid-cells (cell size n x k). The size of a Grid-cell is determined by the size of the broadcast bucket so that one Grid-cell can be placed in one bucket. If possible, the dimensions I and J of a Grid-cell should be selected so that Grid-cells do not span across the dimensions of the data and version array. For a data and version array with *Did=1,..,N* and *Vno=1,..K*, if the Grid-cell dimensions are set to be *I=(N/n)* and *J=(K/k)* to be integers (a more general case is also applicable), then *Grid-cell(i,j)*, where *i=1,...,I, j=1,...,J*, contains the data elements with *Did=(i-1)*n+1, (i-1)*n+2, ..., (i-1)*n+n* and *Vno=(j-1)*k+1, (j-1)*k+2, ..., (j-1)*k+k*.

The buckets are broadcast by linearizing the Grid-cell array either "vertically" (*Grid-Vertical*): *Grid-cell(1,1), ..(I,1), (1,2), ..(I,2), ..(1,J), ..(I,J)*, or "horizontally" (*Grid-Horizontal*): *Grid-cell(1,1), ..(1,J), (2,1), ..(2,J), .. ,(I,1), ..(I,J)*. There are two special cases of grid bucket organizations which could minimize the number of buckets to be read by a client. One is *Vertical* in which each bucket contains data items of the same version. The other is *Horizontal* in which each bucket contains only versions of a data item.

Evaluation. Our simulation results showed that the choice of the best organization depends on the distribution of client access patterns. If the vast majority of the clients is Snapshot/Historical, the Vertical/Horizontal bucket organization must be used. If there is no overwhelming majority of clients with a specific access pattern, the Grid-Vertical/Grid-Horizontal bucket organization must be used. Indexing resulted in energy savings from 2.5 to 5 times. We also found that there is an optimal granularity in terms of the bucket size. Such optimal granularity exploits the advantage of indexing while keeping the indexing overhead small.

References

[1] O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. Multiversion Data Broadcast Organizations. *ADBIS Conf.*, (2002) 135-148

[2] O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. Broadcast Data Organizations and Client Side Cache. *IEEE Workshop on Mobile Distributed Computing*, (2003) 420-426

[3] T. Imielinski, S. Viswanathan, B. R. Badrinath. Energy Efficient Indexing on Air. *ACM SIGMOD Conf.*, (1994) 25–36

[4] O. Shigiltchoff, P. K. Chrysanthis and E. Pitoura. Energy Efficient Access in Multiversion Broadcast Environment. *CS TR-03-112*, University of Pittsburgh, (2003)

Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04)
0-7695-2070-7/04 $20.00 © 2004 **IEEE**