

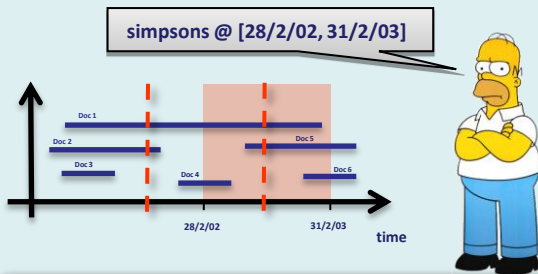
Index Partitioning Strategies for Efficient Time-Travel Search

Avishek Anand

Introduction

Time-travel queries provide for search, exploration and other forms of rich temporal analysis over versioned text-collections like Web Archives. **Time-travel queries** are of the form $\langle \text{keywords} \rangle @ \text{time}$. The time-travel Index (TTIX) [3], extends the standard inverted index list for efficiently answering time-point queries. We consider strategies to partition the TTIX along the time-axis to efficiently support time-range queries.

Indexing for Time-Travel Queries



Partitioning index lists into smaller sub-lists results in reading fewer entries but it also introduces **replication** of index entries at partition boundaries, hence incurring a **blowup** in index size.

Optimization Problems

- **Maximum Replication** : Partition to maximize the number of entries in a list which are ever partitioned, or replicated at least once, given a space budget.

$$\arg \max_{\mathcal{M}} \mathcal{R}(\mathcal{M})$$

$$s.t. \sum_{m \in \mathcal{M}} |L_v : m| \leq \gamma |L_v|$$

\mathcal{M} = set of partition points
 γ = desired blowup
 $\mathcal{R}(\mathcal{M})$ = replication
 $|L_v|$ = #entries in the list
 $|L_v : m|$ = #entries in the partition m .

- **Weighted Replication** : Partition to maximize the overall score of the partition set, given a space budget. Useful in ranked retrieval where weights or scores are term-frequencies (tf) or tf-idf values.

Unfortunately : Both problems are NP-Hard[2].

Fortunately: Greedy solutions exist with proven approximation guarantees.

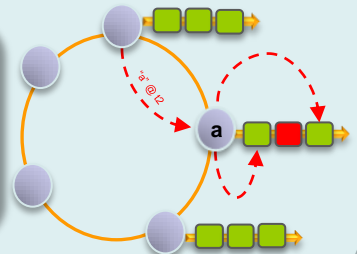
Exploiting Replication of Index Entries

- For **Time-range queries** : Time-range queries might lead to selection of multiple partitions during query processing.

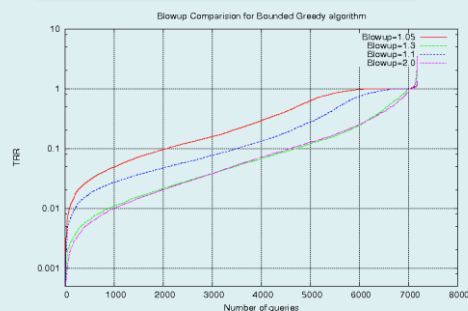
Partition Selection: Exploit overlapping of entries among partitions for selecting a subset of affected partitions with an acceptable (user defined) relative recall.

- In **Peer-to-Peer Retrieval** : Partitions of Index lists reside on autonomous peers. Peers might fail resulting in index unavailability.

Maximize replication of entries across peers. This improves availability and reconstruction of the index by contacting neighbors of the affected peer [1].



Experimental Results



AOL queries (with time-ranges) on Wikipedia versions.

$$TRR = \frac{\# \text{entries}_{\text{read}}(\text{partitioned}) - \# \text{ground}_{\text{truth}}}{\# \text{entries}_{\text{read}}(\text{unpartitioned}) - \# \text{ground}_{\text{truth}}}$$

References

- [1] EverLast: A Distributed Architecture for Preserving the Web : Avishek Anand, Srikanta Bedathur, Klaus Berberich, Ralf Schenkel, Christos Tryfonopoulos. In JCDL, Jun 2009.
- [2] Indexing Strategies for Peer-to-Peer Web Archival : Avishek Anand, Master's Thesis, Universität des Saarlandes and Max-Planck Institut für Informatik, Jan 2009.
- [3] A Time Machine for Text Search : Klaus Berberich, Srikanta Bedathur, Thomas Neumann, Gerhard Weikum. SIGIR 2007, July 2007.



MAX-PLANCK-GESELLSCHAFT