Overview
To comply with recent regulations such as the Sarbanes Oxley Act and SEC Rule 17a-4, corporations are archiving their records on Write-Once-Read-Many (WORM) storage. However, WORM storage is insufficient to ensure records’ trustworthiness: the records can be hidden by tampering with the index used to access them, even if the index is kept on WORM. It is very challenging to provide strong security guarantees (and good performance) for indexes on WORM.

Inverted Indexes on WORM
Updating the index on WORM requires as many disk seeks as the number of terms in the document, and hence is prohibitively slow.

Approach and Impact

Trustworthy Inverted Indexes
• Merged posting lists
  • Jump index, for conjunctive queries

Research Impact
• High-performance queries and updates
  • Trustworthy lookups and conjunctive queries

Merged Posting Lists
Merging posting lists into as many lists as the number of cache blocks results in each update hitting the cache. Merging does increase the length of the posting list and hence slows down queries. The merging strategy must be chosen to minimize the workload run time.

Jump Indexes on WORM
In a jump index, the pointers followed to look up an element are the same as those used to insert the element (B+ trees do not have this property). Hence, lookups always succeed if the pointers are kept on WORM. Jump indexes support lookups in O (log N) time.

Results
• Merged inverted index is only 10% slower than a regular inverted index on a real life document and query workload from IBM’s intranet.
• Jump Index performance is within a factor of 1.5 times that of a B+ tree on the same workload.