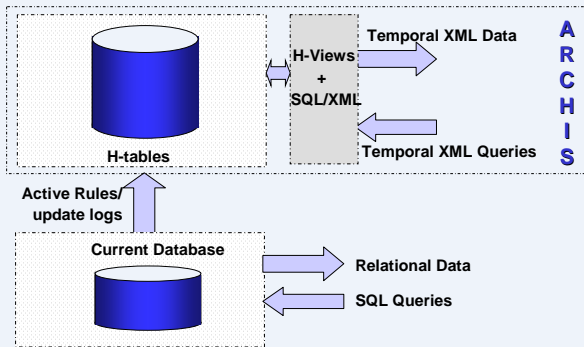


ArchIS: An Efficient Transaction-Time Temporal Database System Built on Relational Databases and XML

Fusheng Wang, Xin Zhou and Carlo Zaniolo

The ArchIS System: Architecture



ArchIS (Archival Information System)

- Expressive temporal representation and data models beyond relational limitations
- Powerful language for temporal queries: no change in standards; temporal libraries ok
- Indexing, clustering and query optimization techniques for efficient query support
- An architecture that integrates all of these into current DBMSs

Why XML

- Timestamped tuples: projection needs coalescing

name	empno	salary	title	deptno	DOB	start	end
Bob	10003	60000	Engineer	d01	1945-04-09	1995-01-01	1995-05-31
Bob	10003	70000	Engineer	d01	1945-04-09	1995-06-01	1995-09-30
Bob	10003	70000	Sr Engineer	d02	1945-04-09	1995-10-01	1996-01-31
Bob	10003	70000	Tech Leader	d02	1945-04-09	1996-02-01	1996-12-31

Temporally grouped history: projection w.o. coalescing

name	empno	salary	title	deptno	DOB	
Bob	10003	60000	Engineer	d01	1945-04-09	
		1995-01-01:1995-05-31		1995-01-01:1995-09-30		1995-01-01:1995-09-30
		70000	Sr Engineer	d02		1995-10-01:1996-01-31
		1995-01-01:1996-12-31		1995-10-01:1996-12-31		1995-01-01:1996-12-31
		70000	Tech Leader	d02		
		1995-06-01:1996-12-31		1996-02-01:1996-12-31		

Mapping Temporal XQuery to SQL/XML

- Temporal projection: retrieve the salary history of "Bob":

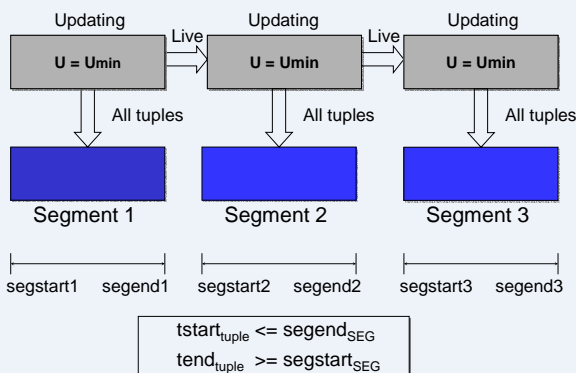
```

element salary_history {
  for $s in doc("employees.xml")/
    employees/employee[name="Bob"]/salary
  return $s
}
    
```

```

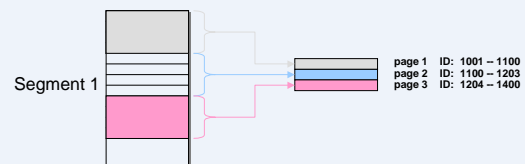
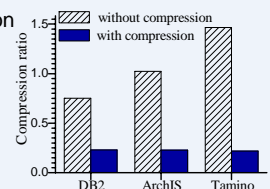
select
  XMLElement (Name "salary_history",
    XMLAgg (XMLElement (Name "salary",
      XMLAttributes (S.tstart as "tstart",
        S.tend as "tend"), S.salary)))
  from employee_salary as S, employee_name as N
  where N.id = S.id and N.name = "Bob"
  group by N.id
    
```

Storage: Segment-based Clustering Scheme

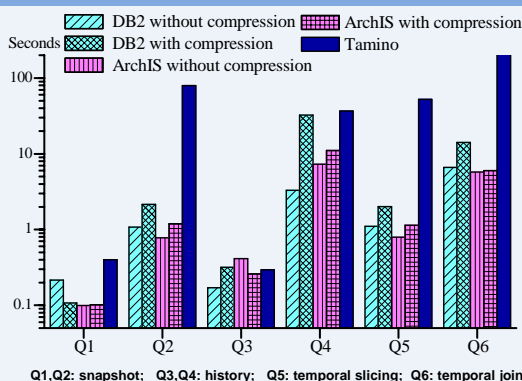


BlockZIP: Block-based Data Compression

- BlockZIP: Block-based compression (LZW&Huffman)
- Indexing and clustering structures preserved
- Effective on Archives because of read-only and fast CPUs



Query Performance



Conclusion

- Temporally grouped data model in XML
- Complex historical queries in standard XQuery
- Efficient support in current DBMSs by:
 - ♦ mapping to SQL/XML
 - ♦ new user-defined temporal aggregates in DBMS
 - ♦ temporal clustering
- Efficient block-based compression (LZW&Huffman)
- Approach applicable to general XML documents for content management and archiving