Temporal Databases and Parallelism

Thomas Zurek
Department of Computer Science
Edinburgh University

May 1995

Overview

1. temporal databases

2. parallelism in databases

3. an example

   - a parallel equi-join

   - a parallel temporal join

4. summary & possible solutions

Temporal Database Systems

In general

A database system (DBS) that offers a functionality for coping with

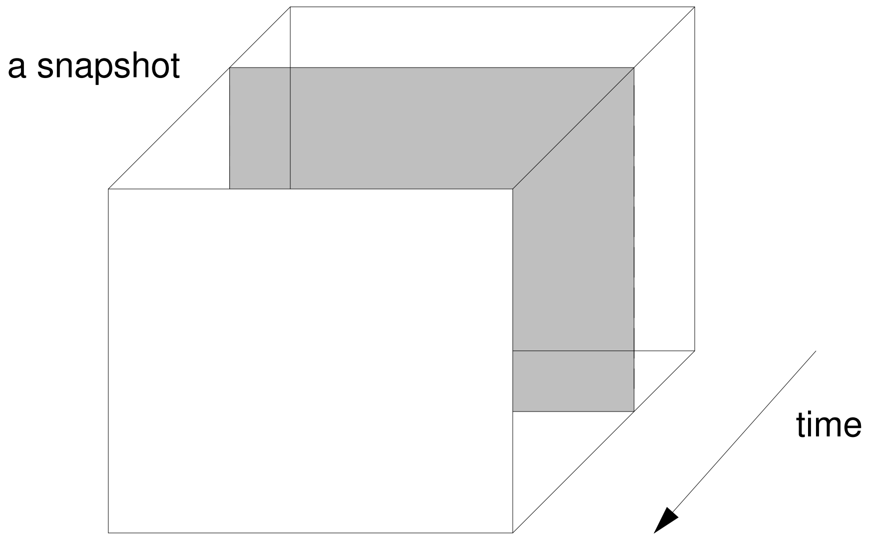
- time,
- time-dependent,
- evolutionary data

In particular

- temporal query language
- temporal query processing

Differences to Traditional DBS

- logical vs. physical deletion
- schema versioning
- new language constructs
- new temporal operators



Examples

- medicine: patients' histories
- banking: accounts' histories
- administration: legal regulations' validity
- natural sciences: experimental data
- industry, commerce: planning

Main Performance Problems

- immense amount of data
  (by logical deletion)

- temporal operators
  - temporal join
  - time slice
  - coalescing
  - temporal selection $\sigma^t = \sigma \circ C$

  ...

Types of Parallelism

- inter-query parallelism
- intra-query parallelism
  - pipelining
  - inter-operator parallelism
  - ▷ intra-operator parallelism

Staff

Name	Start	End
Alex	3	8
Elisabeth	2	10
Frank	4	9
Henry	1	8
Mary	6	10
Vicky	1	10

Students

Name	Start	End
Charles	1	4
Frank	1	4
Karen	7	9
Mary	2	5
Olga	1	3
Steve	7	10

```
SELECT  Staff.Name
FROM    Staff, Students
WHERE   Staff.Name = Students.Name
```

Processor 1: A – F

Staff			Students		
Alex	3	8	Charles	1	4
Elisabeth	2	10	Frank	1	4
Frank	4	9			

Processor 2: G – M

Staff			Students		
Henry	1	8	Karen	7	9
Mary	6	10	Mary	2	5

Processor 3: N – Z

Staff			Students		
Vicky	1	10	Olga	1	3
			Steve	7	10

```
SELECT  Staff.Name, Students.Name
FROM    Staff, Students
WHERE   VALID(Staff) INTERSECTS VALID(Students)
```

Processor 1: Intervals intersecting [1,3]

Staff			Students		
Alex	3	8	Charles	1	4
Elisabeth	2	10	Frank	1	4
Henry	1	8	Mary	2	5
Vicky	1	10	Olga	1	3

Processor 2: Intervals intersecting [4,5]

Staff			Students		
Alex	3	8	Charles	1	4
Elisabeth	2	10	Frank	1	4
Frank	4	9	Mary	2	5
Henry	1	8			
Vicky	1	10			

Processor 3: Intervals intersecting [6,10]

Staff			Students		
Alex	3	8	Karen	7	9
Elisabeth	2	10	Steve	7	10
Frank	4	9			
Henry	1	8			
Mary	6	10			
Vicky	1	10			

Goals

1. maximum parallelism
2. minimum overlaps
3. good load balance

In general, these goals are incompatible

- Is this problem NP-complete?
- How to find a good partition among 2^{n-1} that are possible?

Possible Strategies

- graph partitioning
- statistical partitioning
- heuristic partitioning

applied

- to entire data
- to sample data
- incremental