Stavros Papadopoulos, Yin Yang, Dimitris Papadias

Hong Kong University of Science and Technology

CADS: Continuous Authentication on Data Streams

VLDB 2007, Vienna

Database Outsourcing

Definition:

- A data owner (DO) delegates its database functionality to a third party service provider (SP)

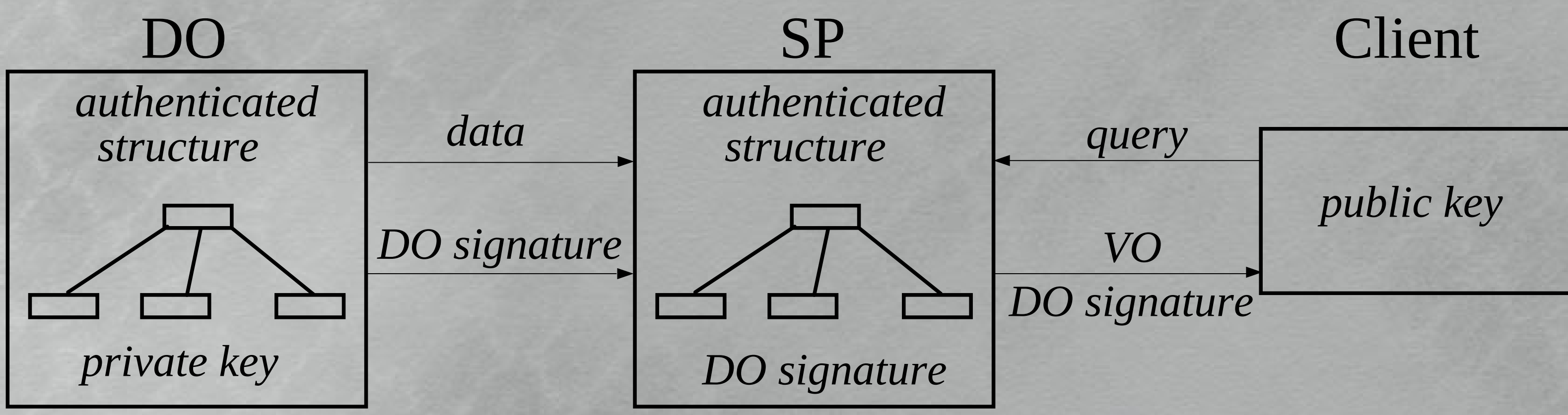
Motivation:

- Some companies may not have the sufficient resources for running a full-scale DBMS to administrate their data
- The SP achieves economies of scale by serving multiple DOs
- The network latency is reduced, since the SPs are located closer to user clusters
- The system robustness is improved, because the SP ceases to be the single point of failure

Challenge:

- Since the SP is not the real owner of the data, it must prove to the clients (i) that the returned results are unaltered (***soundness***) and (ii) that no record that satisfies a query is missing (***completeness***)

The Database Outsourcing Model

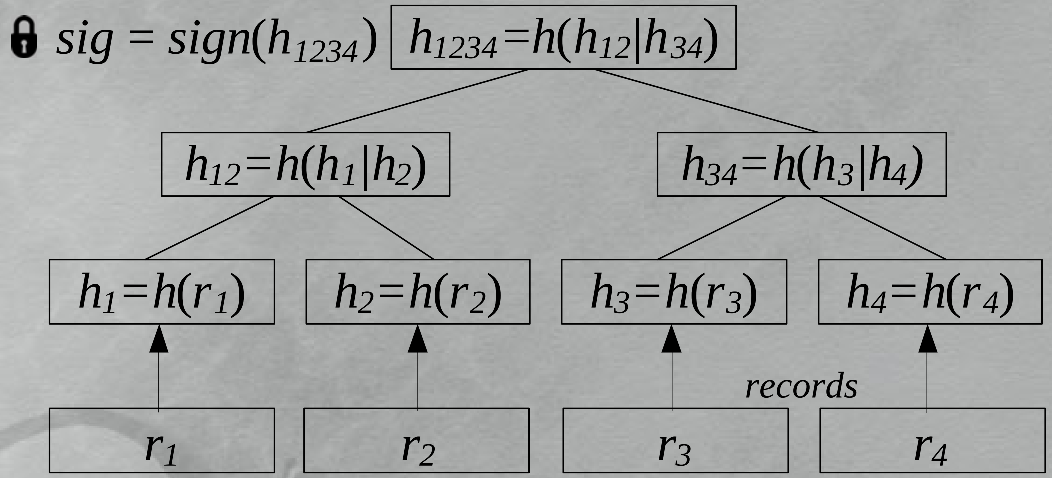


Cryptographic Primitives

- *One-way, collision-resistant hash function*
  - It is intractable to derive M from $h(M)$
  - It is intractable to find M_1 and M_2, such that $h(M_1) = h(M_2)$
  - We use SHA1, which produces a 20-byte digest
- RSA Public Key Cryptosystem
  - Private key: a
  - Public key: (b, c)
  - To sign M, the signer computes $sig = sign(M, a, c) = h(M)^a \bmod c$
  - To verify that M was signed by the proper signer, the verifier computes $verify(M, sig, b, c) = sig^b \bmod c$ and checks if it matches $h(M)$

The Merkle Hash Tree (MH-Tree)

(Merkle, *CRYPTO 1989*)



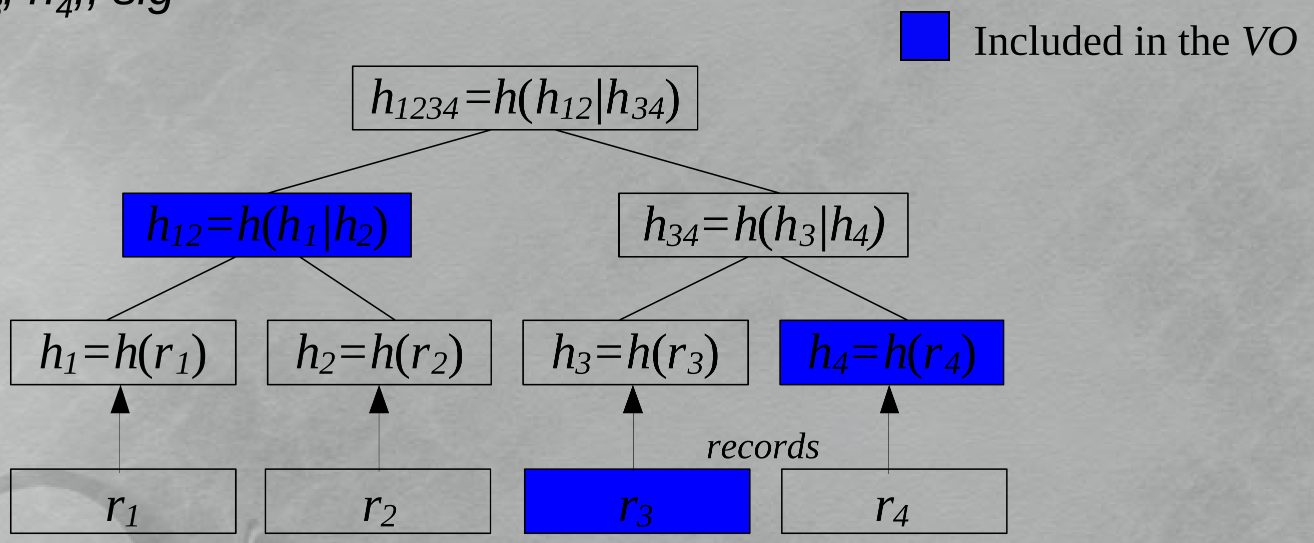
The Merkle Hash Tree (MH-Tree)

(Merkle, *CRYPTO 1989*)

Query result: r_3

VO: h_{12}, r_3, h_4,, *sig*

Example



Streaming Environments

- Record updates are constantly being introduced to the system
- We adopt the *positive-negative model*, which is more general, but our methods apply to the *sliding window model* as well
- The users issue continuous range queries and they expect to receive feedback whenever their results change

Example Application:

- The SP receives current values from one or more stock exchanges
- Subscribers register long-running queries at the SP
- Whenever a stock update influences a query, the corresponding client is immediately informed

Streaming Environments

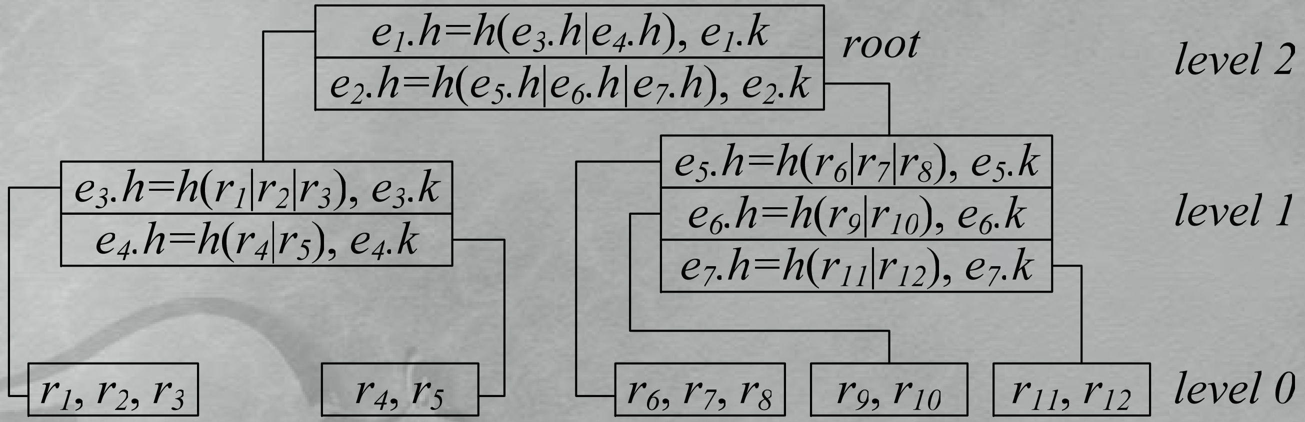
<u>*Challenges:*</u>

- Need for soundness and completeness
- The system must accommodate very fast updates and support efficient query processing
- It must include effective mechanisms for minimizing the communication overhead and the verification cost
- Need for ***temporal completeness***:
  - The SP must guarantee that all clients receive *all* the updates that are relevant to their queries

A Reference Solution - REF

$$sig = sign(h(e_1.h \mid e_2.h \mid LT \mid ST))$$

DMH-Tree



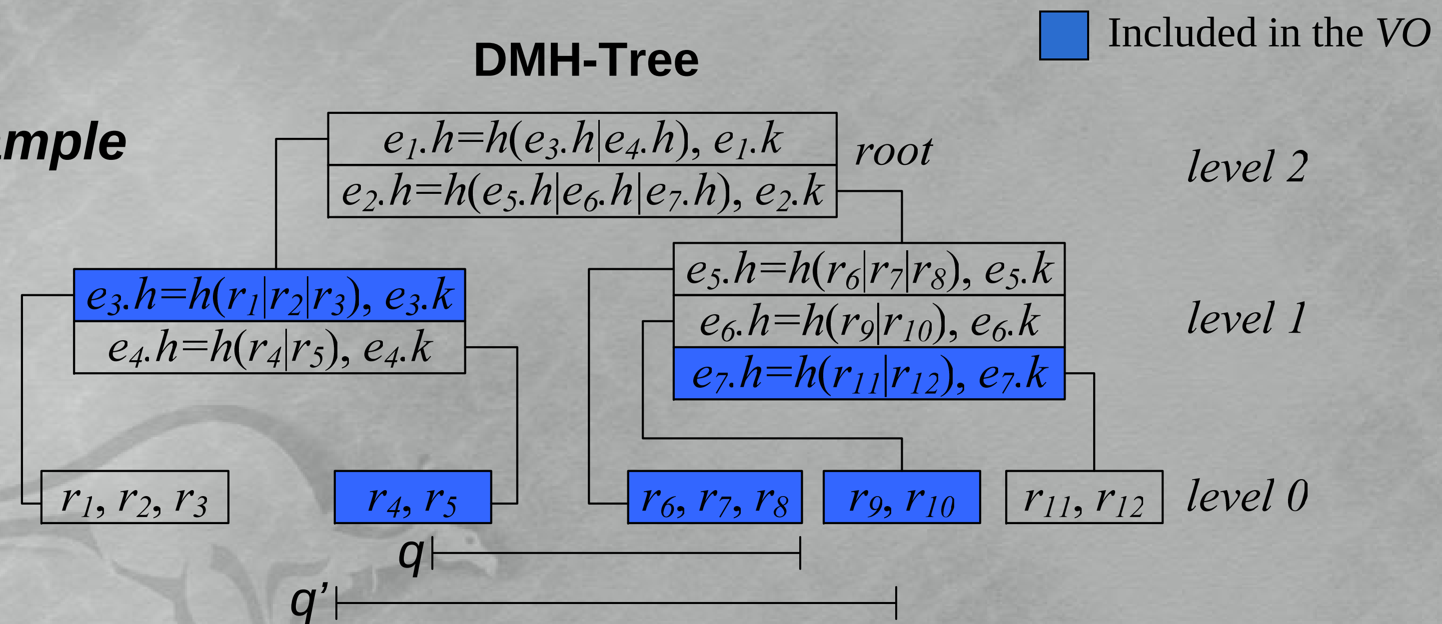
A Reference Solution - REF

<u>Query result:</u> r_5, r_6, r_7, r_8

<u>*VO*:</u> $e_3.h, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, e_7.h, sig, LT, ST$

Example

DMH-Tree



A Reference Solution - REF

Query Processing:

- The SP sends a new *VO,* the signature, *LT and ST* to *every* client for *every* update it receives
- Soundness and completeness are proven as in the MB-Tree
- Timestamps *ST* and *LT* ensure temporal completeness

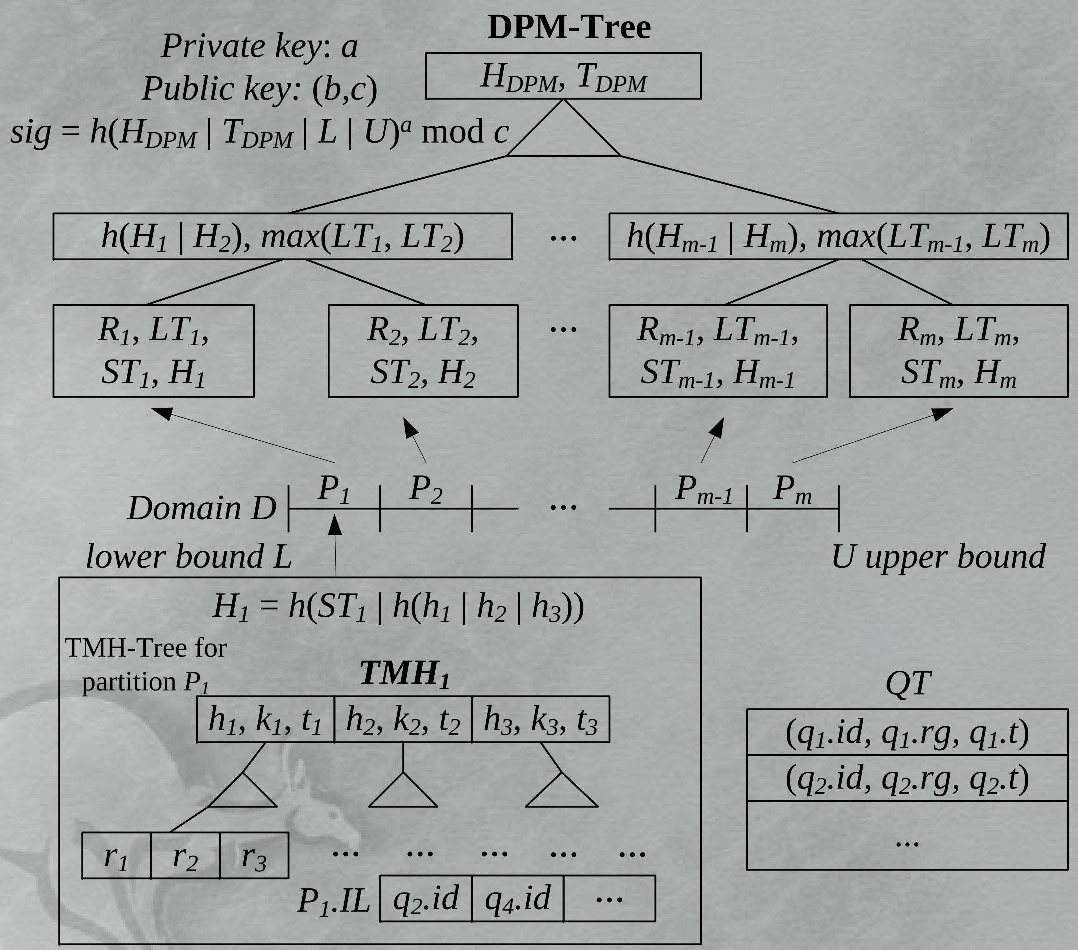
Example:

- τ = 1: client *C* obtains a result
- τ = 2: the SP receives a new record r_1, but it does not inform *C*
- τ = 3: r_1 is deleted and a new record r_2 becomes part of the result. The SP sends a new *VO* to *C* along with the signature, *LT*=3 and *ST*=2.

Main drawback:

- False transmissions

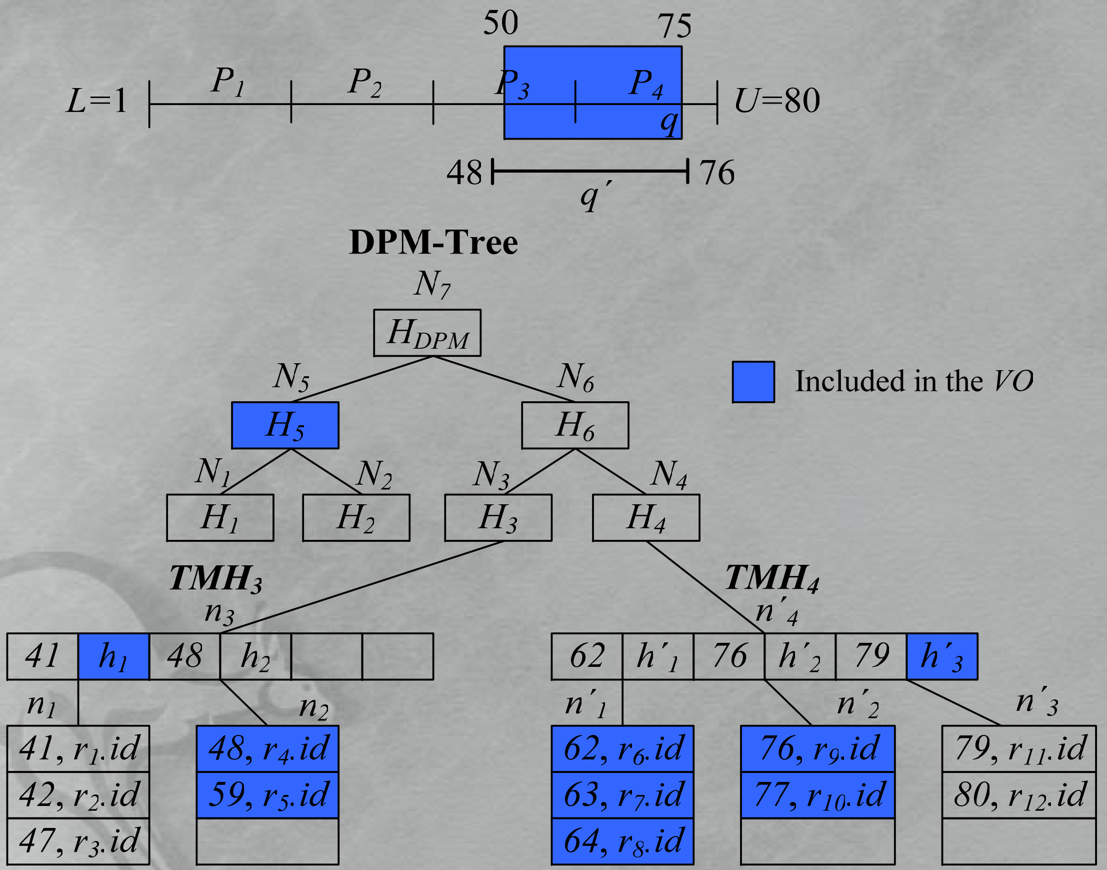
CADS – Indexing scheme



CADS – Initial Computation

VO: $[H_5, [begin\_TMH, N_3.ST, [h_1, [r_4, r_5], dummy], end\_TMH, begin\_TMH, N_4.ST, [[r_6, r_7, r_8],[r_9, r_{10}], h'_3], end\_TMH]]$

Example



CADS – Monitoring Module

Key points:

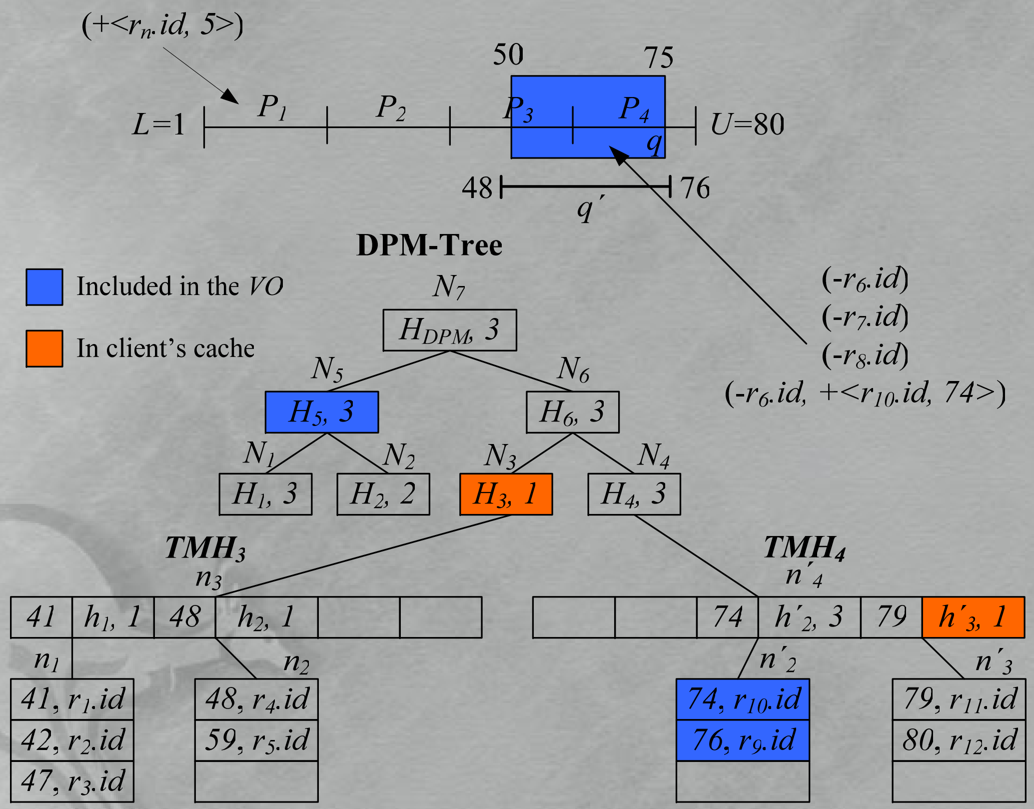
- The SP sends a new *VO* only to the clients whose queries overlap with the partitions where the updates have occurred
- We also employ a *Virtual Caching Mechanism* (*VCM*).
  - It prevents the SP from sending *VO* components (hashes/records) that already exist in the client's cache
  - The term *virtual* means that the SP does not maintain any *VO* for any query

CADS – Monitoring Module

$\tau = 1$ ***cachedVO***: [H_5, [*begin_TMH*, N_3.*ST*, [h_1, [r_4, r_5], *dummy*], *end_TMH*, *begin_TMH*, N_4.*ST*, [[r_6, r_7, r_8],[r_9, r_{10}], h'_3], *end_TMH*]]

$\tau = 3$ ***newVO***: [H_5, [*Hit*, *begin_TMH*, N_4.*ST*, [*dummy*, [r_9, r_{10}], *Hit*], *end_TMH*]]

Example



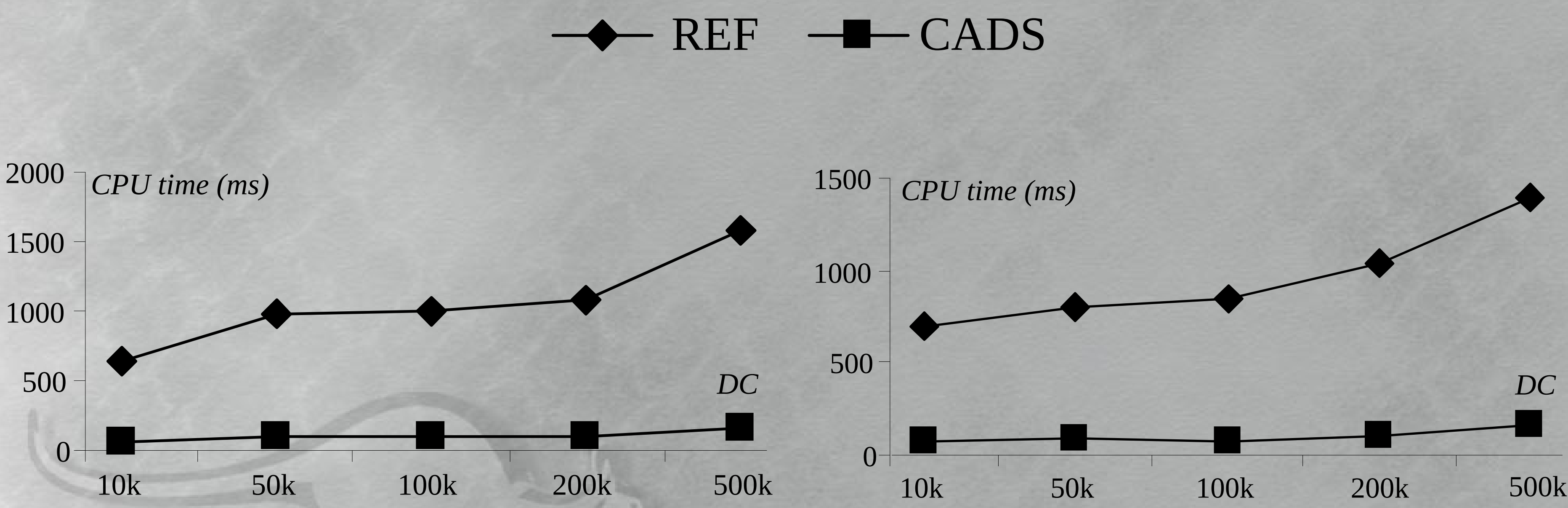
Experimental Evaluation

Setting:

- Two datasets (UNI and SKD)
- Record size: 100 bytes
- Each experiment is a simulation of 100 timestamps
- Continuous queries are uniformly distributed having 0.1% selectivity
- *m* is set to 8192 after a fine tuning step
- Parameters:
  - *Data Cardinality* (*DC*): 10K, 50K, **100K**, 200K, 500K
  - *Query Cardinality* (*QC*): 100K, 500K, **1K**, 2K, 5K
  - *Arrival rate* (*AR*): 10, 50, **100**, 200, 500

Experimental Evaluation

Query processing time vs. *DC*

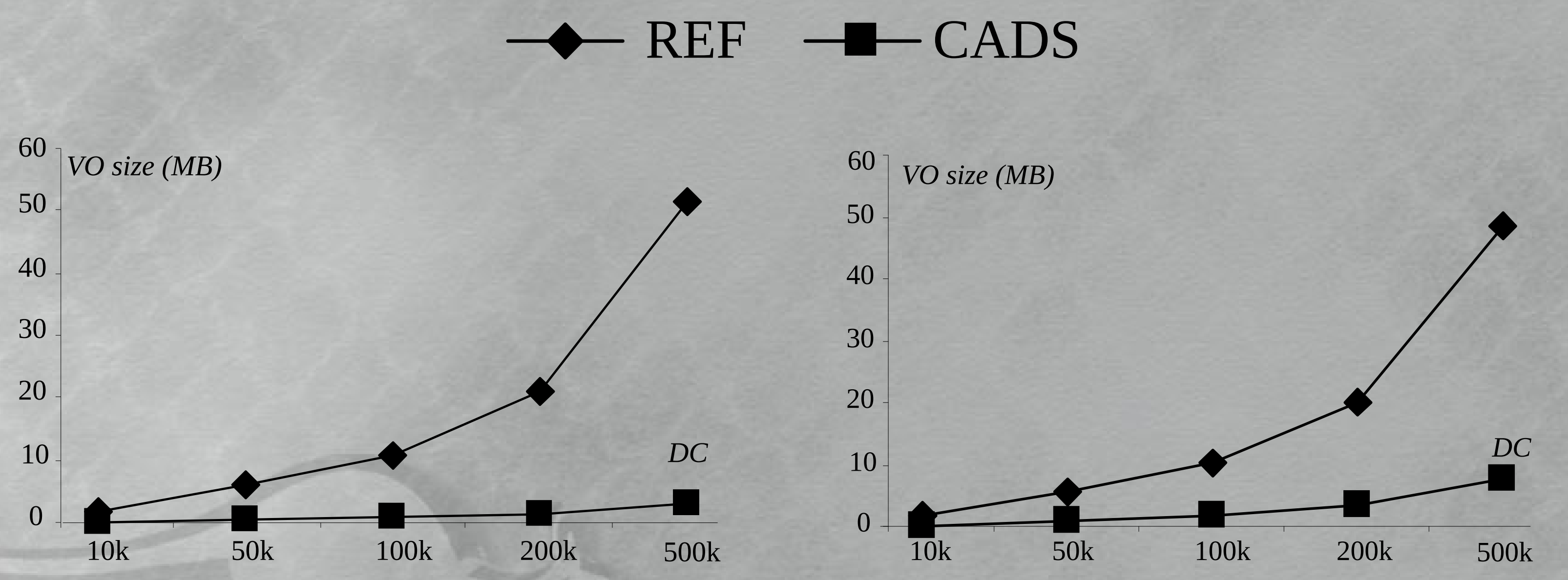


UNI

SKD

Experimental Evaluation

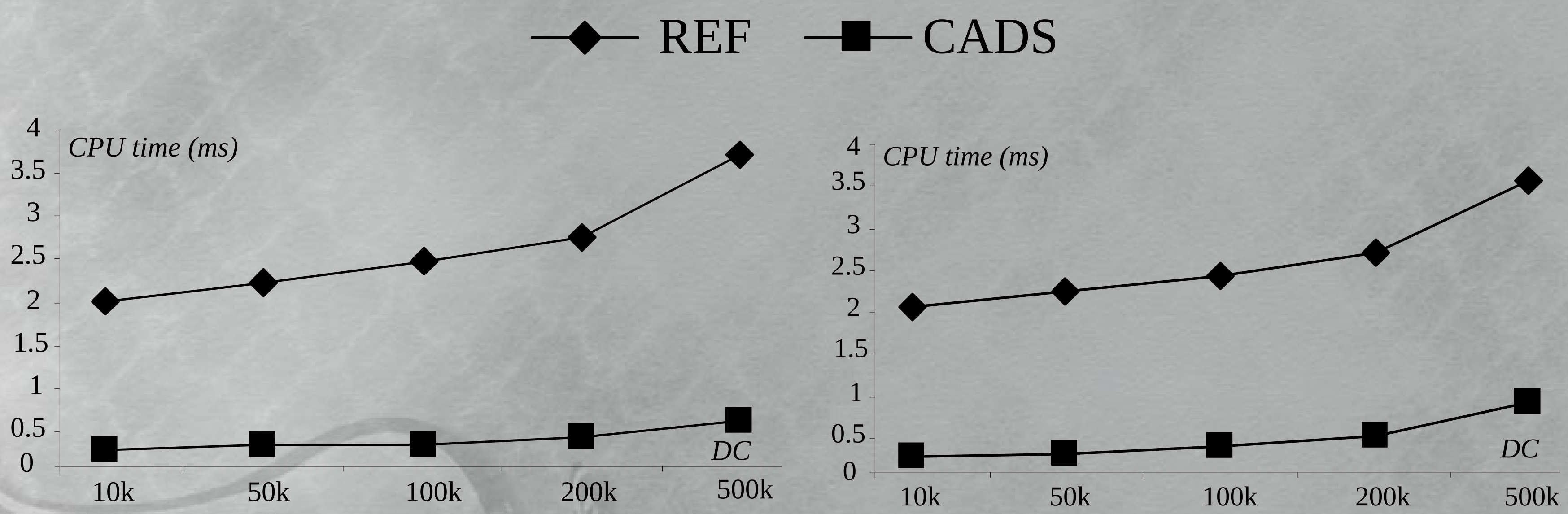
VO* size vs. *DC



UNI SKD

Experimental Evaluation

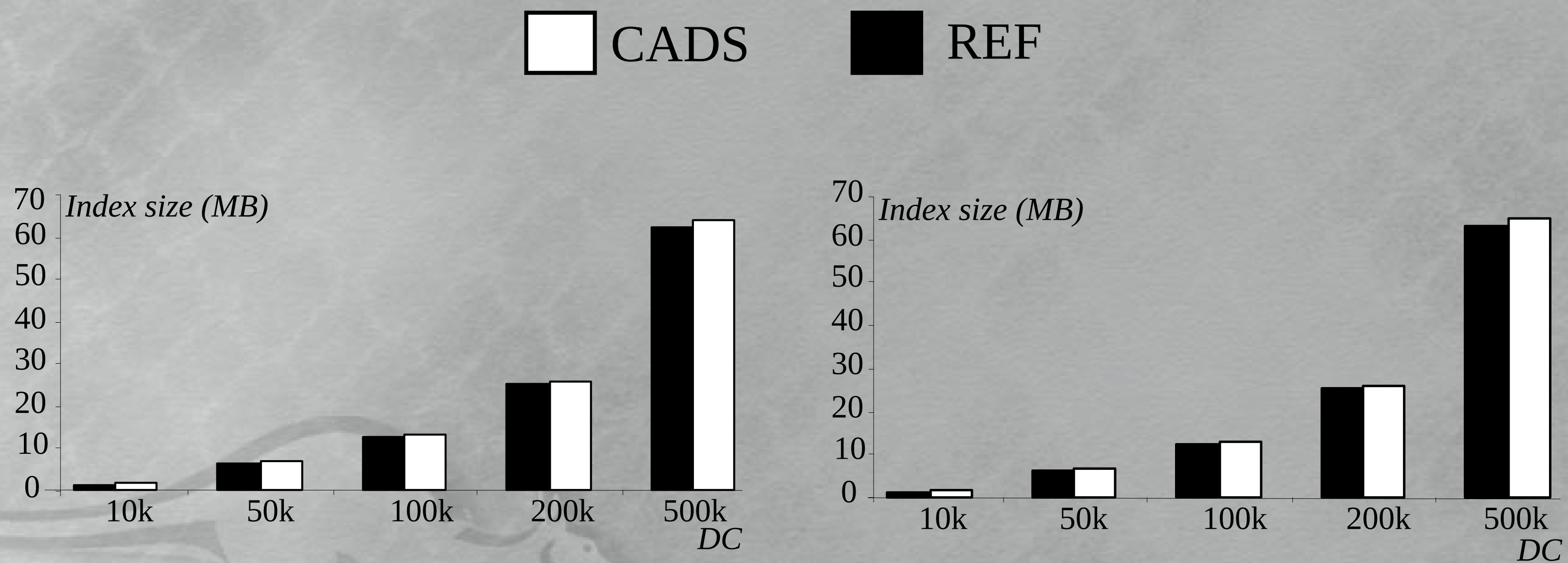
Verification time vs. *DC*



UNI SKD

Experimental Evaluation

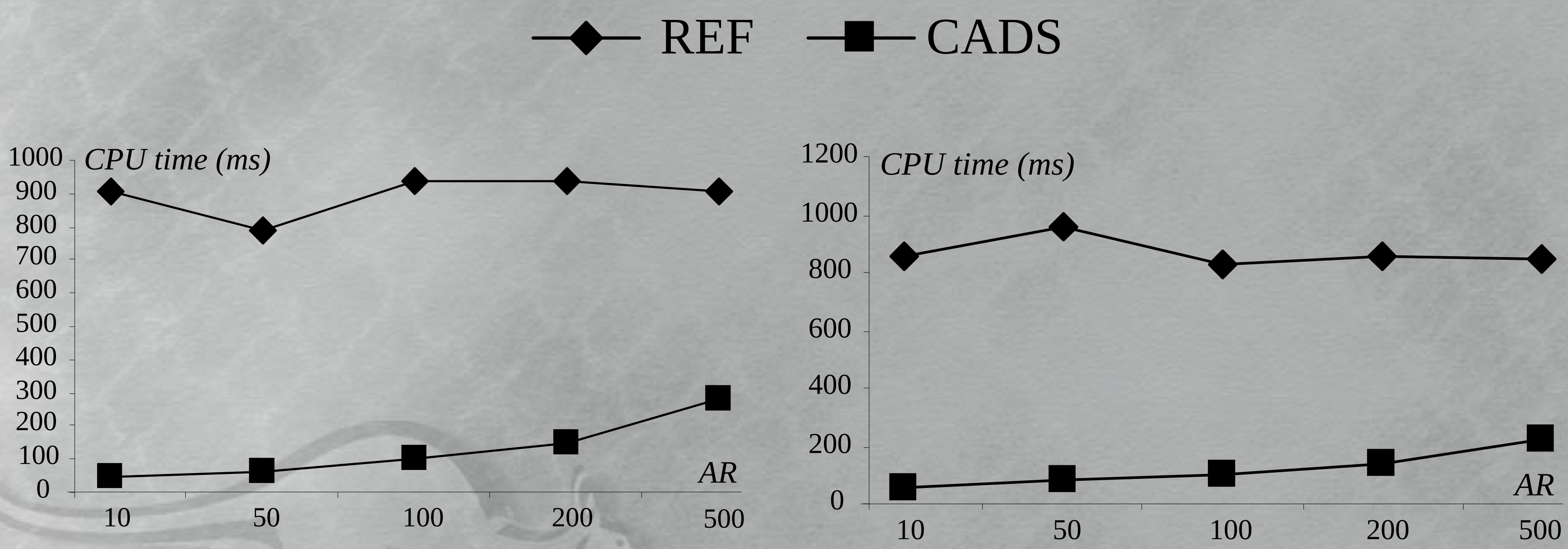
Index size vs. *DC*



UNI SKD

Experimental Evaluation

Query processing time vs. *AR*

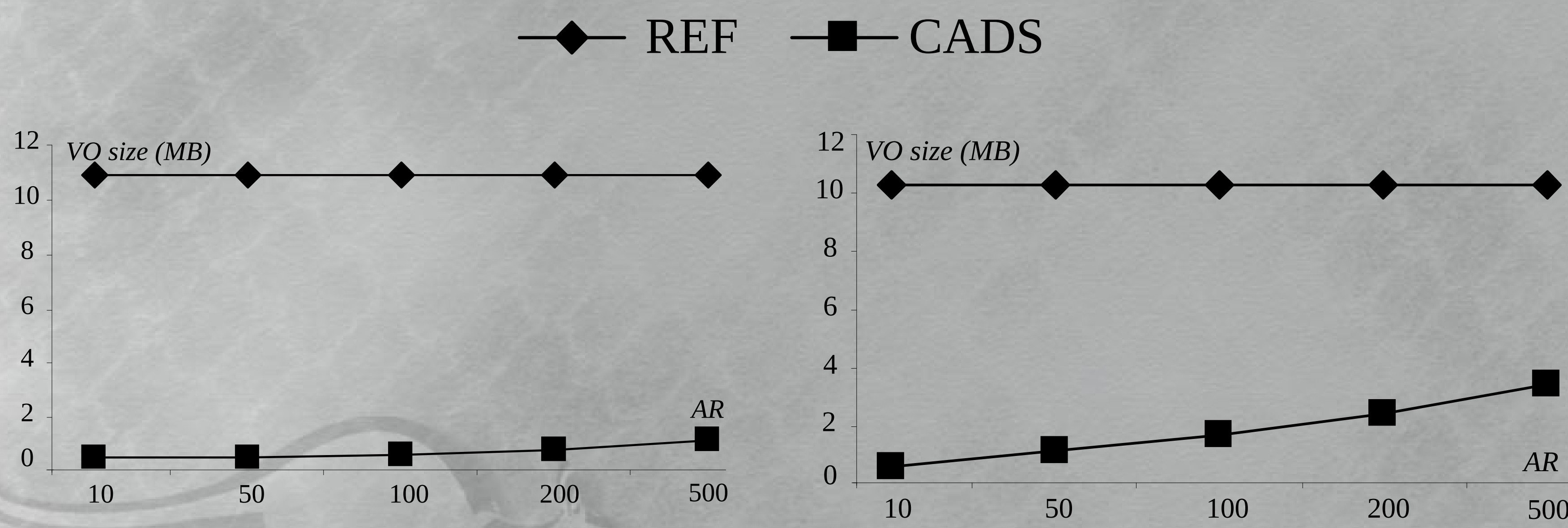


UNI

SKD

Experimental Evaluation

VO* size vs. *AR



UNI SKD

Future Work

- Extend CADS to solve the spatial version of the problem
- Find the partitioning granularity that optimizes the performance of CADS
- Handle the case where there are multiple empty partitions. One solution is to attribute *imbalance* to the DPM-Tree, which can reduce the size of the tree and, thus, the query processing time and the *VO* size.

Questions

?