Slalom: Coasting Through Raw Data via Adaptive Partitioning & Indexing
Matthais Olma, Manos Karpathiotakis, Yannis Alagiannis, Manos Athanassoulis and Anastasia Ailamaki

In the Era of Data Deluge
Data collections are increasingly larger
Exploratory workloads become popular e.g., smart meter networks

Problem:
- DBMS startup cost
- Unknown data, workload

Goal: Create efficient turn-key solution

Minimizing data access with Slalom

1. In-situ query processing
   - Positional map
   - Binary caches

2. Logical Partitioning
   - Sequential partitions
   - Maximize skipped data

3. Adaptive Indexing
   - Existence indexes
   - Position indexes

Conventional Approaches

DBMS
Fast but Expensive Initialization

In-situ query processing
No initialization but Slow

Evaluation: Compare with PostgreSQL, PostgresRaw, in-memory DBMS, Database Cracking

Benchmark: Smart meter exploration workload 680m tuples (59GB), 25 attributes, Sel-Proj-Agg, sel 0.1-10%

Slalom in Action

Slalom dynamically refines its indexes to achieve high query performance

Slalom does not incur loading cost

Slalom uses less memory

Data collections are increasingly larger
Exploratory workloads become popular e.g., smart meter networks

Problem:
- DBMS startup cost
- Unknown data, workload

Goal: Create efficient turn-key solution

Minimizing data access with Slalom

1. In-situ query processing
   - Positional map
   - Binary caches

2. Logical Partitioning
   - Sequential partitions
   - Maximize skipped data

3. Adaptive Indexing
   - Existence indexes
   - Position indexes

Conventional Approaches

DBMS
Fast but Expensive Initialization

In-situ query processing
No initialization but Slow

Evaluation: Compare with PostgreSQL, PostgresRaw, in-memory DBMS, Database Cracking

Benchmark: Smart meter exploration workload 680m tuples (59GB), 25 attributes, Sel-Proj-Agg, sel 0.1-10%

Slalom in Action

Slalom dynamically refines its indexes to achieve high query performance

Slalom does not incur loading cost

Slalom uses less memory