Scalable Processing of Moving Flock Patterns

Andres Oswaldo Calderon Romero

University of California, Riverside

November 2024

Introduction

- Advances in spatio-temporal data collection have increased the need to process large datasets.
- Flock patterns represent groups moving closely together over time.
- Applications include transportation, ecology, and urban planning.

Flock Pattern Detection

- A flock pattern is a group of entities within a given radius for a specified period.
- Detection is computationally intensive, especially with large datasets.
- Current methods (e.g., BFE) lack scalability.

Sequential Approaches

- **BFE Algorithm:** Basic Flock Evaluation identifies maximal disks, or regions where entities move closely.
- **PSI Algorithm:** Optimizes BFE by reducing the search space through a half-square approach.
- PSI generally outperforms BFE on large datasets.

Challenges in Sequential Approaches

- Sequential methods suffer from high computational demands and scalability issues.
- Spatial and temporal data distribution creates performance bottlenecks.

Scalable Solutions

- **Partitioning Strategy:** Quadtree-based partitioning for spatially distributed processing.
- **Replication and Safe Zones:** Ensures correct flock detection across partitions.
- **Temporal Joins:** Strategies to efficiently handle flocks moving across partitions.

Distributed Temporal Join Strategies

- **Master, By-Level, LCA, Cube-based** methods evaluated for handling partitioned flocks.
- Cube-based approach performs best for large datasets, leveraging higher parallelism.

Experimental Evaluation

- Tested on synthetic datasets to evaluate partitioning and temporal join efficiency.
- **Results:** PSI outperformed BFE, and Cube-based approach demonstrated best scalability.

Optimizations

- Optimal partition size and configuration improve performance.
- Local, intermediate, and global reduce phases enhance processing efficiency.

Conclusions

- The scalable approach to flock pattern detection overcomes limitations of traditional methods.
- Distributed processing using partitioning and replication significantly improves efficiency.
- Results highlight effective strategies for handling large spatio-temporal datasets.