ST4ML: Machine Learning Oriented Spatio-Temporal Data Processing at Scale

Kaiqi Liu, Panrong Tong, Mo Li, Yue Wu, Jianqiang Huang

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

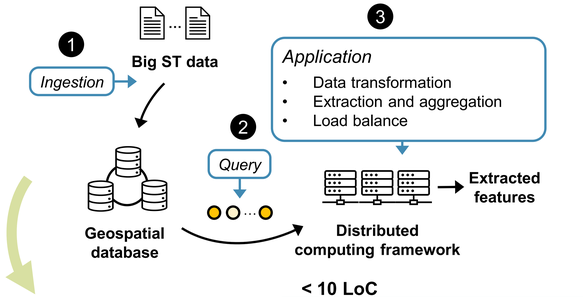
Alibaba Group

Opensource: *http://github.com/Panrong/st4ml*

Background

- Machine learning applications with large-scale ST data solve real problems

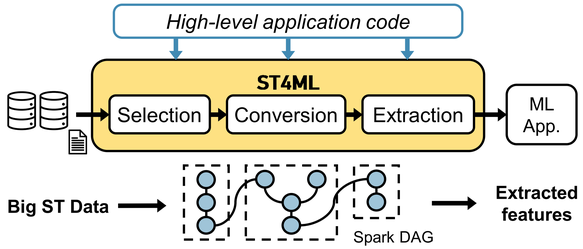
- ML with ST data take **derived features** instead of raw data as input
  - E.g., vehicle trajectories → regional speed
- Existing systems support only queries instead of data transformation and organization
- ST feature extraction with pipelining existing systems: ↑ programming burden ↓ performance



< 10 LoC

```
val raster = ReadRaster(rasterFile)
val selector = Selector[STTraj](sQuery,
tQuery, n = 100)
val converter = Traj2RasterConverter(raster)
val extractor =
RasterSpeedExtractor(unit="kmh")
val trajRDD = selector.select(dataDir)
val rasterRDD = converter.convert(trajRDD)
val speedRDD = extractor.extract(rasterRDD)
saveParquet(speedRDD, resDir)
```

System Design



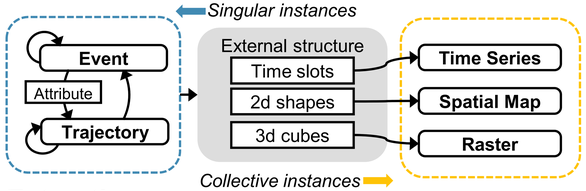
- Spark based distributed processing
- Three-stage pipelining paradigm
- 5 fundamental ST instances
- Optimized operations and user-friendly interface

Selection

- Load data into memory based on ST ranges
- Data partitioning to keep load balance in facilitating the entire pipeline
- Optional R-tree index for specific apps

Conversion

- 5 instances cover most ST applications
- Application-specific conversion



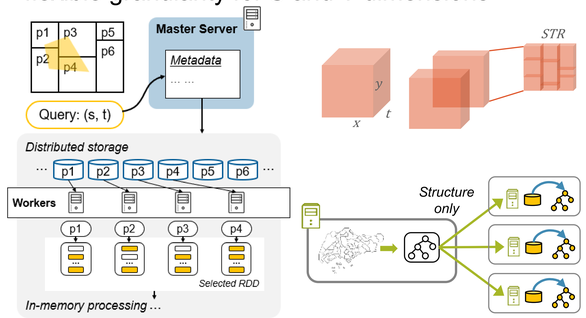
Extraction

- 3 levels of flexibility for extensibility and easy-of-use: built-in extractors, RDD-level APIs and native RDD programming

Key Optimizations

On-Disk Indexing for data loading

- Not all data need to be in-memory
- Reusing partitioning result helps save loading time and memory usage
- New T-STR partitioner extending STR with flexible granularity for S and T dimensions



Conversion with broadcasted structure index

- Optimizing singular-to-collective conversion
- Multi-dimensional R-tree on collective structure to ensure high parallelism

Evaluation

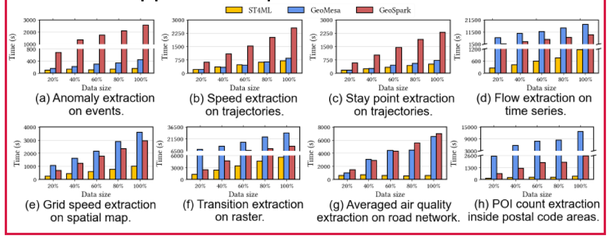
Microbenchmarks

- *Data loading with metadata indexing* saves up to **60%** data loading time and prunes **42%** to **98%** irrelevant data
- *Instance conversion with indexed structure* performs up to **105x** faster
- *T-STR partitioner* achieves better ST-aware load balance comparing to baselines and better facilitate downstream apps (up to **7x** faster)

Evaluation – Cont'd

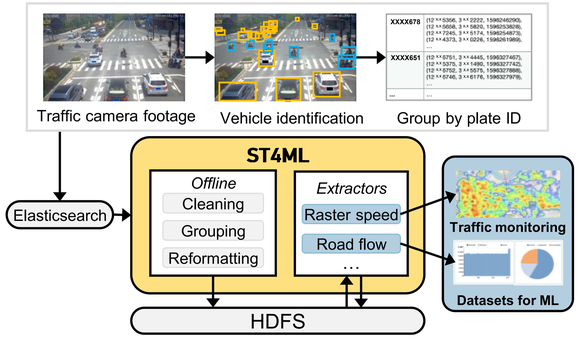
End-to-end applications

- ST4ML outperforms baselines by up to **39x** in various apps and requires **half** lines of code



Case Study

Serving Alibaba City Brain lab's business



Map matching + traffic flow extraction

- Daily city-scale trajectories: < 1h

Traffic speed extraction on raster

- 3-7x faster than GeoSpark