Ganos: A Multidimensional, Dynamic, and Scene-Oriented Cloud-Native Spatial Database Engine

Jiong Xie, Zhen Chen, Jianwei Liu, Fang Wang, Feifei Li, Zhida Chen, Yinpei Liu, Songlu Cai, Zhenhua Fan, Fei Xiao, Yue Chen

Alibaba DAMO & Alibaba Cloud
A New Era of City Digital Twins

• What are city digital twins?
  • Digitization copies of cities
  • Bidirectional interaction between digital and real worlds
  • Use data and data analytics to help simulation
  • Facilitate the automated governance of cities

• Broad applications
  • Urban planning
  • Smart traffic management
  • Automated environment monitoring
  • Etc.
MDS Data

- **MDS**: Multidimensional, Dynamic, and Scene-oriented spatial data

  - Multidimensional: real-world buildings as 3D entities
  - Scene-oriented: textures and materials

BIM (Building Information Modeling)

- Multidimensional: 3D position (x, y, z)
- Dynamic: positions change over time
- Scene-oriented: take off/landing events, collected images ......
Challenges to the DBMS Design

• Data types
  • Complex data structure & large scale of data size

• Query types
  • How to support different types of queries, e.g., spatio-temporal queries, scene-oriented queries, and cross-model queries

• Efficiency
  • The large scale and complex data structure result in long query time, e.g., a “big query” can take hours to finish

• Traditional spatial RDBMS have limited support for MDS data in both data types and operations
Our Solution

• Ganos: a new **cloud-native** spatial RDBMS engine
  • The name comes from the goddess of earth **Gaea** and the god of time **Chronos**

• Built on cloud-native relational database **PolarDB for PostgreSQL**

• Features of Ganos
  • Consider MDS data as first-class citizens
  • A new multidimensional data type hierarchy include 3DMesh, Trajectory, Raster, etc.
  • A systematic framework to manage the MDS data
  • Utilize cloud-native approaches to solve “big” storage / queries
Architecture of Ganos and Relation between Ganos and PolarDB
Data Types

- **3DMesh = (Shape, Visuals, General)**
  - Shape: a 3D geometry
  - Visuals = (textures, materials, UVcoords)
  - General attributes

- **Trajectory = (TPoints, Events)**
  - \( \text{TPoints} = \{(p_1, t_1, A_1), \ldots, (p_n, t_n, A_n)\} \)
  - \( \text{Events} = \{(e_1, t_1^e), \ldots, (e_m, t_m^e)\} \)

- **Raster = (Footprint, Time, Matrix)**

- **PointCloud = \{(p_1, A_1), \ldots, (p_n, A_n)\}**

- ......

Hierarchy of Ganos data types
Data Type Implementation

- **Spatial Large Object (SLOB)**
  - Compact binary sequence
  - Two Parts: profile and details
  - Profile—summary of an object and is used for filtering
  - Details—detailed information of an object

- **SLOB of 3DMesh type**

- **Indexes**
  - nD R-tree based on GIST+
  - Enriches the access methods of GiST

A building as an example

- A building = $N$ components (a roof, many doors, ……)
- Each component is stored as a SLOB of 3DMesh type
- The components share the same building id

---

*GIST: Generalized Search Tree originally provided by PostgreSQL*
Extended Storage

- Ganos allows storing SLOB profile in a database table and storing details on OSS
  - #1: Hot/cold data separation
  - #2: Heterogeneous file access
- Thoughts behind the design
  - An MDS object can be very large
  - Many queries are interested in the same subset of objects or a small part of the object
  - The extended storage can achieve a decent tradeoff between storage cost and query performance
Query Types

• Spatial queries
  • 3D relationships, 3D analysis, and 3D processing operations in 3D scenarios

• Spatio-temporal queries
  • Spatio-temporal relationships, spatio-temporal analysis, and spatio-temporal processing operations

• Scene-oriented queries
  • Operations to construct, edit, and process the scenes

• Cross-model queries
  • Hybrid queries that involve multiple data types
  • e.g., overlay analysis of 3DMesh and Trajectory

Implement a rich set of operations to support these queries
Parallel Execution

• Spatial-oriented multi-level parallelism
  • Intra-query parallelism (IQP)
    • Parallelizes a big query by assigning data slices to many RO nodes
    • The default size of each data slice is 4MB (512 pages), which can be set by users
    • Data slice assignment can be hash-based or dynamic
  • Intra-function parallelism (IFP)
    • Further parallelizes the processing of a huge cell by dividing it into small cells and calling subprocesses to process them
    • To mitigate the potential load imbalance problem that is caused by the existence of spatial objects with drastic size differences

Example of intra-function parallelism for ST_Union (huge cell)
Use Case Study

- A cross-model query in a city digital twin scene with different data types
- Handling complex 3D scenarios with simple GeoSQL

E.g. UAV must not touch the restricted-fly zone (the 3D space whose distance from a building is less than 100 meters)

```
SELECT 1 FROM t_trajectory, t_building
WHERE ST_3DIntersects(ST_3DBuffer(t_building.m, 100),
t_trajectory.traj) AND t_trajectory.id = 1;
```

E.g. The maximum height from the ground of the flight must be lower than 500 meters

```
WITH height AS (
   SELECT st_z((st_dumppoints(traj)).geom) - st_z((st_addz(rast, traj)).geom) AS h
   FROM t_trajectory, t_dem WHERE t_dem.id = 1 AND t_trajectory.id = 1)
SELECT max(h) < 500 FROM height;
```
Key Evaluation Conclusions

- Datasets: OSM data (96,648,669 trajectories) and BIM data (1,000 large buildings)
- OSS can reduce storage cost with an acceptable sacrifice of QPS.
- Although reading data from OSS is slow, with the help of the indexes, the query performance on spatio-temporal queries can become acceptable.
- Spatial-oriented multi-level parallelism with IQP + IFP can significantly accelerate the processing of big queries on MDS data.

(a) QPS varying ratio of on-OSS data
(b) Latency distribution of on-OSS data
(c) Latency vs. degree of parallelism

Testbed: 1 RW node, 3 RO nodes of PolarDB for PG, 32 CPUs & 256GB memories per node
Novel Applications

- Ganos has offered service in Alibaba Cloud for over 4 years
- It has been applied to a total of 45 industries/application directions

3D Scenes and Analytics
Achieve in-database computation acceleration by nearly 100 times in urban planning and construction of a State-Level New Area in China

Querying Dynamic Data
Provide built-in Trajectory types and transparent hot/cold data access for LBS service providers

Database for GeoAI
Give the solution of dynamic monitoring of ecological environment based on integrate aerospace data management for a satellite environmental application center of China
Conclusion

• With the rapid development of smart cities, digital twins, and cloud computing, existing spatial relational databases cannot meet the requirement of modern applications for MDS data processing
• Ganos provides a systematic framework of data models, access methods, and operations for MDS data
• Ganos optimizes the processing of queries on MDS data through cloud-native capabilities, which provides a new practice of moving from traditional on-premise spatial database to cloud-native spatial database