



Parameterized Algorithm Routine: The Perfect Balance between Performance and Usability

Bing Tong

浙 江 创 邻 科 技 有 限 公 司

Createlink Technology Co., Ltd

Outlines

01.

Background

02.

Galaxybase Solution - PAR

03.

Future Outlook and Conclusion

01

Background

创邻科技 Createlink



The Irreversible Trend of the Internet of Everything

The Internet of Everything is driven by massive data connectivity, leading to an increasing demand for data assetization, service-oriented models, and intelligent solutions.



Economic Connectivity
Transportation



Information Connectivity
Internet



Social Connectivity
Facebook, Twitter, QQ



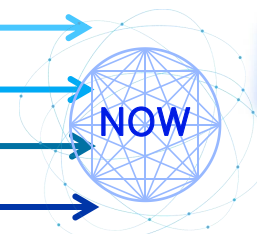
Device Connectivity
IoT

1900s

1990s

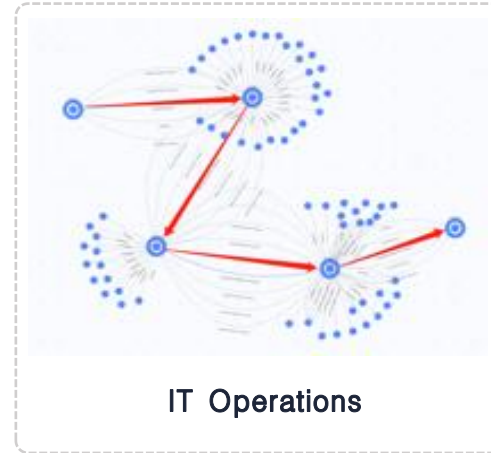
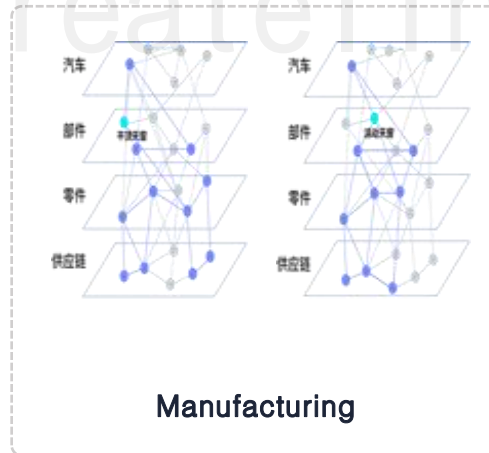
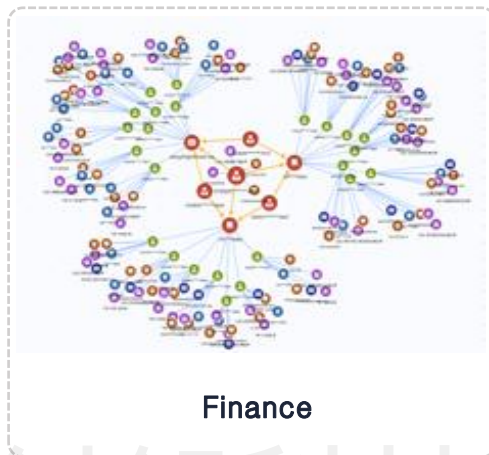
2000s

2010s





Applications of Graph Technology Across Scenarios





Graph Queries: The Key to Diverse Applications

How can graph databases be effectively applied across different scenarios?

— Through **Graph Queries**



Social Network Analysis

- Friend Relationship Queries
- Influence Center Identification
- Mutual Friends Query
- Community Detection



Finance

- Transaction Path Tracking
- Anomalous Transaction Detection
- Risk Analysis
- Transaction Pattern Analysis





Overview of Graph Queries

	Declarative Query	Imperative Query
Definition	Focuses on describing the data to retrieve, not the retrieval process.	Focuses on describing execution steps, with the user specifying how to obtain the data.
Characteristics	Closer to natural language, with automatic optimization.	Provides fine-grained control; flexible but complex.
Optimization	Optimized in real-time.	Optimized before execution.
Examples	Cypher, GQL	Procedures, functions

Declarative Query

```
MATCH (p:Person)-
[:FRIEND]->(f:Person)
WHERE p.name = 'Alice'
RETURN f.name
```

Imperative Query

```
@Procedure(name = "org.example.findFriends", mode = Mode.READ)
public Stream<FriendResult> findFriends(@Name("name") String name) {
    Node personNode = db.findNodes(Label.label("Person"), "name",
name).stream().findFirst().orElse(null);
    ....
}

CALL org.example.findFriends('Alice')
YIELD friendName
RETURN friendName;
```



Why Do Graph Databases Need Imperative Queries?

Reason

Graph databases are typically used in OLAP tasks where relational databases have limitations.

OLAP tasks often involve whole-graph or subgraph queries.

OLTP

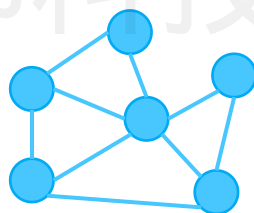


Table

OLAP



Joined Table



Graph

Application Ratio in Galaxybase

OLTP&OLAP: 70%

OLTP: 20%

OLAP: 10%

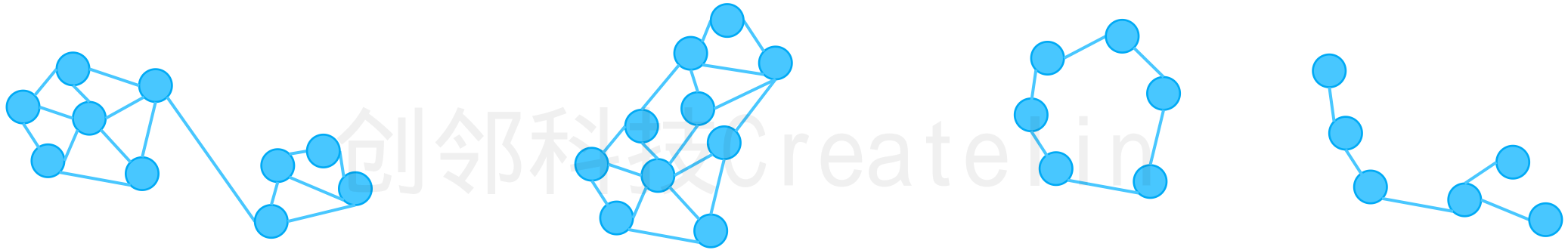
Graphs have a natural advantage in OLAP tasks.

OLAP is the most common use of graphs in real-world applications.

Why Do Graph Databases Need Imperative Queries?

Reason

For graph queries, the unstructured nature of graph data—covering data distribution, distributed storage, and query unpredictability—makes perfect query optimization theoretically impossible.



Summary

In high-performance scenarios, imperative queries are necessary because they allow for scenario-specific optimization before execution.



Imperative Queries in Graph Databases

Graph Database	Imperative Query	Support Version	Visualization Support	Permission Support	Distributed Support	Data Distribution-Based API	Query Language Invocation
Galaxybase	PAR	From v3.x	yes	yes	yes	yes	yes
TigerGraph	UDF	From v3.x		yes	yes		yes
Neo4j	UDP/UDF	From v3.x		yes			yes
TuGraph	POG	From v3.x	yes				yes

创邻科技 Createlink





Challenges of Imperative Queries



Implementation Difficulty:

Developing procedures is much more complex than writing GQL queries, and user-friendly tools for imperative queries are scarce.



Support for Distributed and Concurrent Environments:

In production environments with large datasets, distributed graph databases are often used. However, some databases may not support distributed and concurrent operations for imperative queries.



Stability and Security:

Complex queries can consume substantial resources, making system stability and security essential. However, some systems support for these requirements is still limited.



02

Galaxybase Solution - PAR

创邻科技 Createlink



What is Galaxybase PAR?

PAR: Parameterized Algorithm Routine

A high-performance, distributed, imperative query tool.

- 1 Provides pre-optimization capabilities tailored to specific scenarios
- 2 Ensures stability and reliability in production environments
- 3 Offers a user-friendly development approach





Galaxybase PAR - High Performance



Support for Distributed and Concurrent Environments:

In production environments with large datasets, distributed graph databases are often used. However, some databases may not support distributed and concurrent operations for imperative queries.

Solution

1. Distributed Support

It enables distributed optimization based on actual data storage distribution, offering distributed interfaces and data distribution insights.

2. Concurrency Support

Concurrency performance is optimized according to data partitioning, tailored to the read-write characteristics of the disk.





Implementation Difficulty:

Developing procedures is much more complex than writing GQL queries, and user-friendly tools for imperative queries are scarce.

Solution

1. Simple Registration

Users can easily upload and modify procedures via the frontend **interface**, making the process straightforward.

项目名称 ▾	创建时间 ▾	当前版本	描述	包状态 ▾	操作
test	2024-08-18 22:19:11		This is a test PAR	未上线	 包管理 注册方法 日志 编辑 删除

2. Functionality Encapsulation

The **PAR Kit** encapsulates a wide range of distributed and multithreading interfaces, simplifying development and boosting efficiency. Even if users are not familiar with distributed and parallel processing details, they can still write code that supports these features with ease.



Stability and Security:

Complex queries can consume substantial resources, making system stability and security essential. However, some graph databases support for these requirements is still limited.

Solution

1. Unified Resource Management

Provides centralized thread pool management, file operations, and memory tracking, ensuring efficient resource utilization and enhanced system stability.

2. Termination Function

Supports custom termination of executing stored procedures to prevent excessive resource consumption by inefficient programs, thereby ensuring system security.

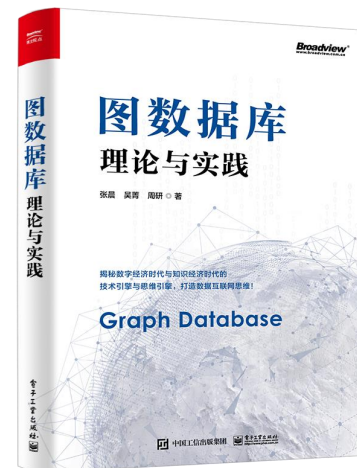




Galaxybase PAR in Use



PAR in the "Graph Databases" course at Zhejiang University



PAR in the book "Graph Databases: Theory and Practice"

第 6 章 图数据库服务编程

执行使用 Galaxybase 特定语法的自定义过程和函数。类似地, TigerGraph 也支持服务端编程的方式。使用特定语言 SQL 实现用户自定义函数 (User Defined Function, UDF)。用户可以通过创建和运行使用 CSQL 编写的 UDF 实现更复杂的逻辑。这些 UDF 可以在单独的文件中编写,并在安装后通过 CSQL 查询或其他 UDF 调用。虽然不同数据库产品的服务端编程方式存在差异,但它们都使得编写和管理复杂现代码更加方便。

6.2 Galaxybase PAR API 简介

为了给用户提供更完善的体验,并且能够在更低层分布式计算框架的能力, Galaxybase 提供了一种通过 PAR API 进行服务端编程的方式。PAR API 是 Galaxybase 提供的一套基于 Java 的高级编程开发接口,用户可以使用 PAR API 编写服务端程序。这些程序和分布式计算框架 (Par) 通过 PAR API 编写的程序可以在 Galaxybase 图数据库引擎中运行,执行复杂的查询和计算操作。图数据库引擎的查询优化、编写并执行定制化算法、分布式并行点/边数据遍历、图计算任务的分布式调度等, Galaxybase 提供了丰富的 PAR API。用户只需通过 PAR API 实现的 PAR 程序文件部署到图数据库引擎中,然后通过启动对象进行调用即可,表 6-1 描述了查询语言、驱动模式和 PAR 二者的使用方式对比。

接口	查询语言 (以 Cypher 为例)	驱动模式 (以 Java 驱动为例)	PAR
接口支持	图数据库引擎	图数据库引擎	图数据库引擎
持续使用	图数据库引擎	图数据库引擎	图数据库引擎
开发	需要 Cypher 语言基础, 需要图数据库引擎知识	需要 Java 技能, 需要具备编程技能的开发人员编程知识	需要 Java 技能, 需要具备数据库、图数据库等知识或图数据库知识
性能	图数据库引擎	图数据库引擎	图数据库引擎
扩展	图数据库引擎	图数据库引擎	图数据库引擎

简单描述, 用户可以通过编写 PAR 来实现更复杂的查询和计算逻辑。

- 用户可以根据需求定制化开发相应的 PAR。

2071

PAR is used ALL of our customers



03

Future Outlook and Conclusion

创邻科技 Createlink



We aim to drive the development of the graph database industry by leveraging both declarative and imperative approaches.

1. Support for Hybrid Declarative and Imperative Modes
2. Automatic Conversion between Declarative and Imperative Modes
3. Training and Education on Declarative and Imperative Graph Database Queries
4. PAR Cross-Platform Compatibility
5. ...

Declarative Queries



Imperative Queries





Conclusion

In certain scenarios, especially in production environments, **Imperative Queries** are often more effective than **Declarative Queries**. Thus, it's crucial to focus on developing imperative queries alongside declarative queries.

For instance, Galaxybase's **PAR** represents a significant advancement in imperative queries, providing enhanced support for a range of industry applications.

We aim to **collaborate** with various graph database vendors to advance both declarative and imperative approaches, thereby **strengthening the position of the graph database industry**.





**INSIGHT INTO CONNECTED DATA
CREATING INFINITE POSSIBILITIES**

Follow our WeChat official account to learn more