Createlink

Zhejiang Createlink Technology Co., Ltd.

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

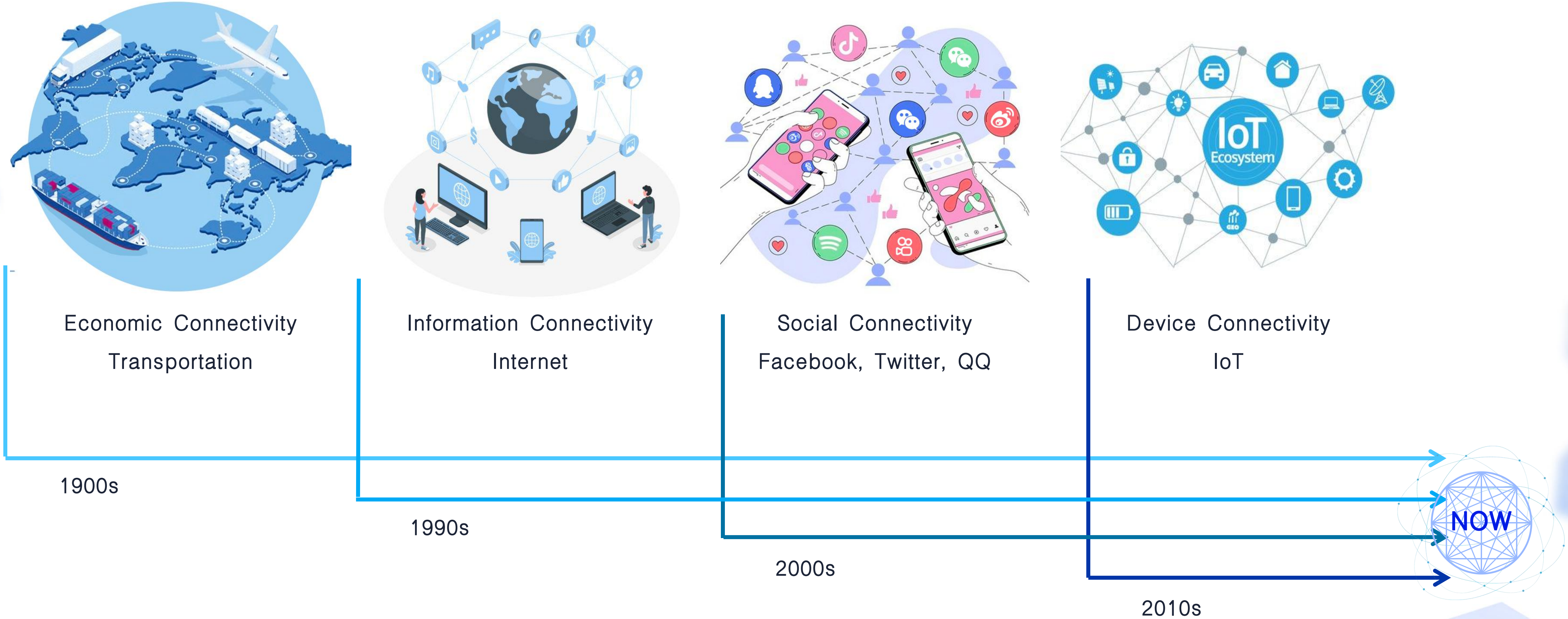
03.
Future Outlook and Conclusion

01

Background

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

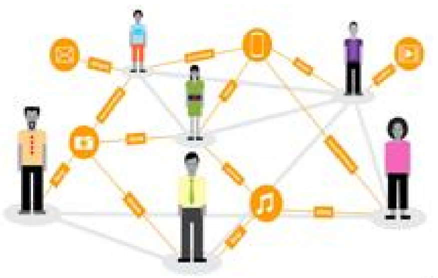
Social Connectivity
Facebook, Twitter, QQ

Device Connectivity
IoT

Applications of Graph Technology Across Scenarios

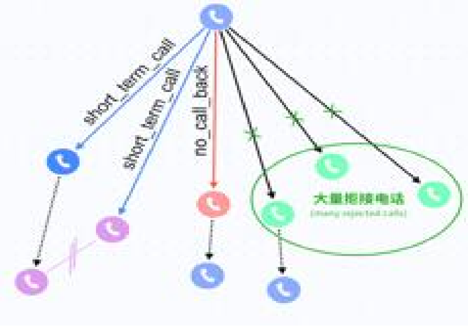
Social Networks

Finance



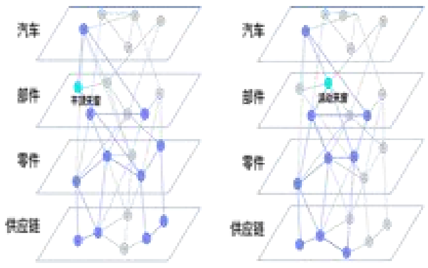
Retail

Energy

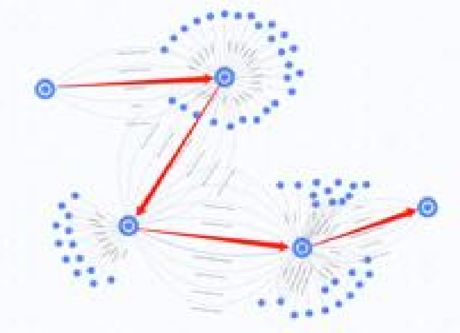


Telecommunications

Smart Governance



Manufacturing



IT Operations

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;
```

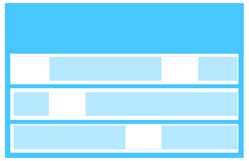
Why Do Graph Databases Need Imperative Queries?

Reason

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

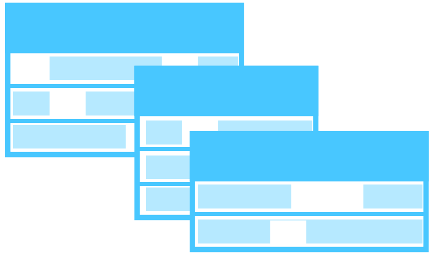
OLAP tasks often involve whole-graph or subgraph queries.

OLTP

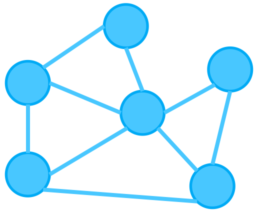


Table

OLAP



Joined Table



Graph

Graphs have a natural advantage in OLAP tasks.

Application Ratio in Galaxybase

OLTP&OLAP: 70%

OLTP: 20%

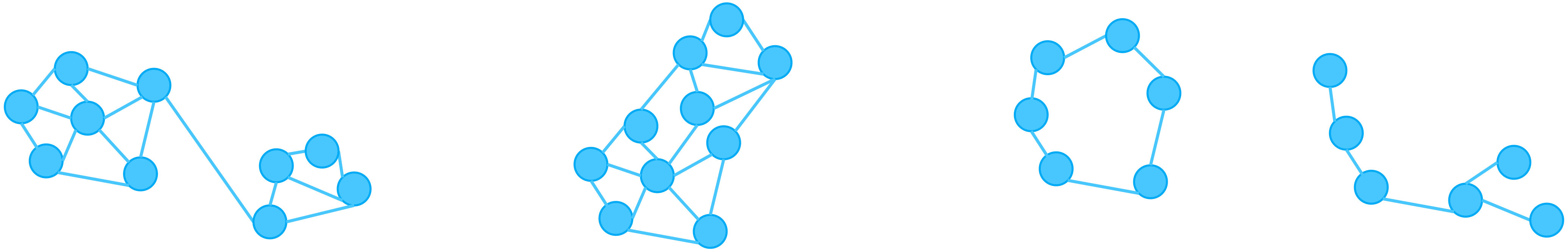
OLAP: 10%

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

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

What is Galaxybase PAR?

Createlink
Zhejiang Createlink Technology Co., Ltd.

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.

Galaxybase PAR - Usability

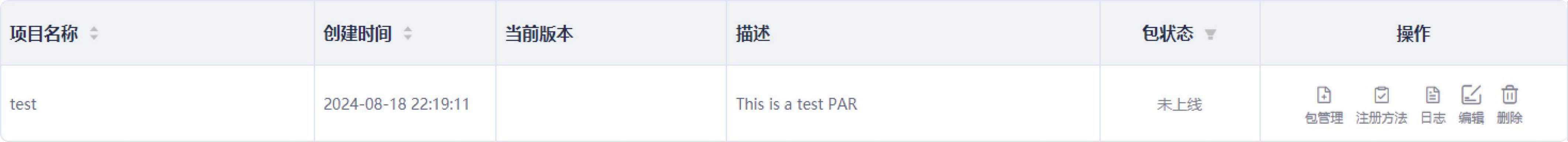
Implementation Difficulty:

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.

Galaxybase PAR - Security

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

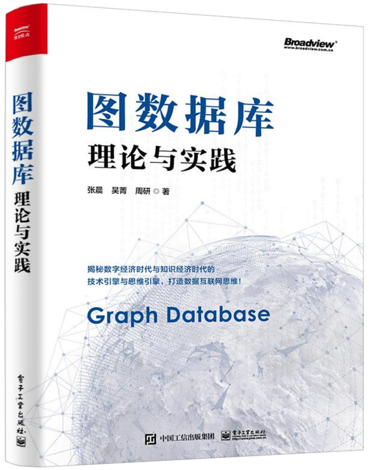
Provides centralized thread pool management, file operations, and memory tracking, ensuring efficient resource utilization and enhanced system stability.

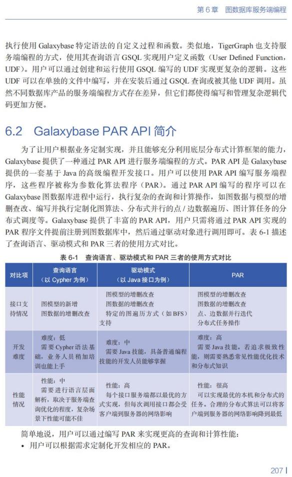
2. Termination Function

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





第 6 章　图数据库服务端编程

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，使用其查询语言 GSQL 实现用户定义函数（User Defined Function，UDF）。用户可以通过创建和运行使用 GSQL 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 GSQL 查询或被其他 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逻辑代码更加方便。

6.2　Galaxybase PAR API 简介

为了让用户根据业务定制实现，并且能够充分利用底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，这些程序被称为参数化算法程序（PAR）。通过 PAR API 编写的程序可以在 Galaxybase 图数据库进程中运行，执行复杂的查询和计算操作，如图数据与模型的增删查改、编写并执行定制化图算法、分布式并行的点 / 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，用户只需将通过 PAR API 实现的 PAR 程序文件提前注册到图数据库中，然后通过驱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 三者的使用方式对比。

表 6-1　查询语言、驱动模式和 PAR 三者的使用方式对比

对比项	查询语言 （以 Cypher 为例）	驱动模式 （以 Java 接口为例）	PAR
接口支持情况	图模型的新增 图数据的增删改查	图模型的增删改查 图数据的增删改查 特定的图遍历方式（如 BFS）支持	图模型的增删改查 图数据的增删改查 点、边数据并行迭代 分布式任务操作
开发难度	难度：低 需要 Cypher 语法基础，业务人员稍加培训也能上手	难度：中 需要 Java 技能，具备普通编程技能的开发人员能够掌握	难度：高 需要 Java 技能。若追求极致性能，则需要熟悉常见性能优化技术和分布式知识
性能情况	性能：中 需要进行语言层面解析，取决于服务端查询优化的程度，复杂场景下性能可能不佳	性能：高 每个接口服务端都以最优的方式实现，但每次调用接口都会受客户端到服务器的网络影响	性能：很高 可以实现最优的本机和分布式的任务。合理的分布式算法可以将客户端到服务器的网络影响降到最低

简单地说，用户可以通过编写 PAR 来实现更高的查询和计算性能：

- 用户可以根据需求定制化开发相应的 PAR。

207

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

PAR is used ALL of our customers

中国农业银行 AGRICULTURAL BANK OF CHINA · 交通银行 BANK OF COMMUNICATIONS · 中国民生银行 CHINA MINSHENG BANK · 北京银行 BANK OF BEIJING · 国家电网公司 STATE GRID · 中国南方电网 CHINA SOUTHERN POWER GRID · Tencent 腾讯 · 个推 GeTui · 浙江大学 ZHEJIANG UNIVERSITY · 香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY · •••

03

Future Outlook and Conclusion

Future Outlook

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.**

Createlink
Zhejiang Createlink Technology Co., Ltd.

INSIGHT INTO CONNECTED DATA

CREATING INFINITE POSSIBILITIES

Follow our WeChat official account to learn more