2019
05
08-10
北京新云南皇冠假日酒店

数据风云十年变迁

DTCC 第十届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2019

IT168 ChinaUnix ITPUB

HugeGraph图数据库应用案例与存储原理

李章梅

2019.5

Baidu 百度

目录

图数据库是什么

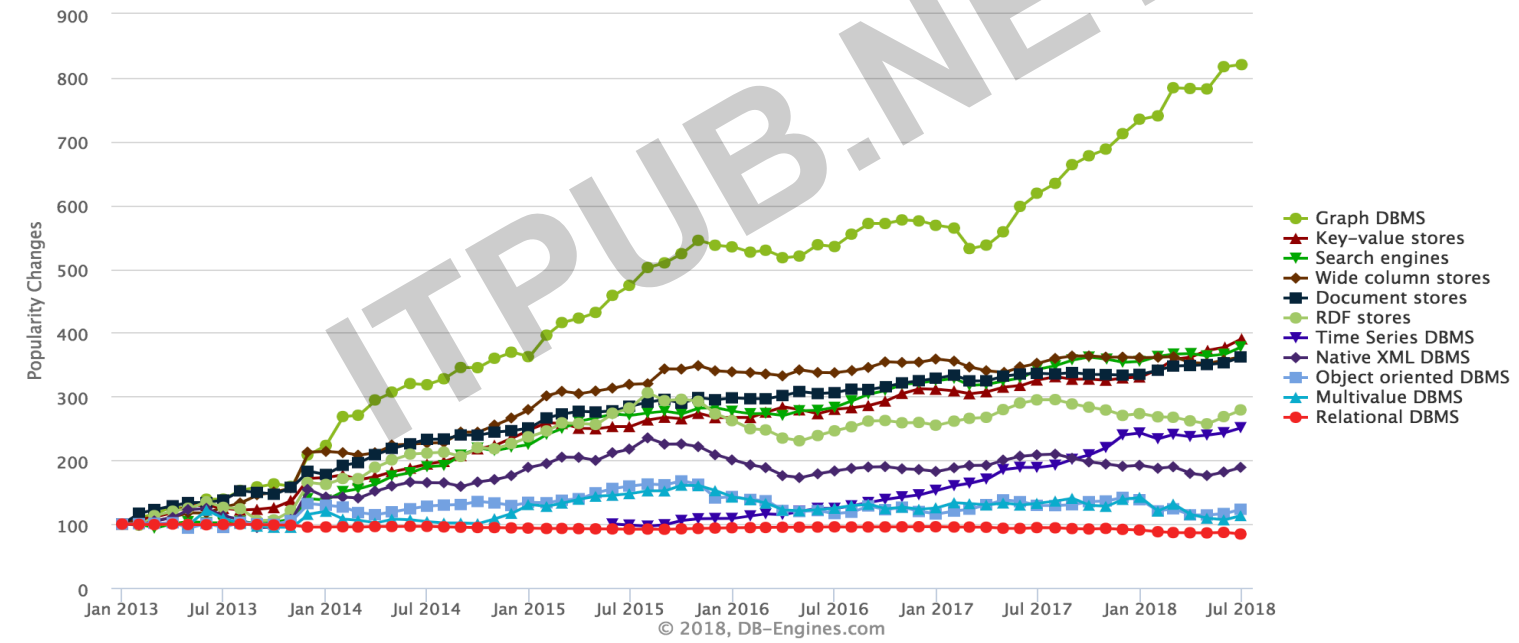
新型的NoSQL数据库

图数据库是什么

图数据库是一种新型的NoSQL数据库

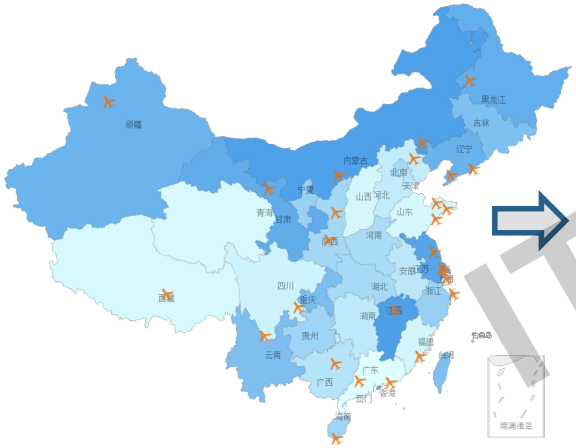
图数据库是什么

图数据库在近几年广受关注



图数据库是什么

图数据库：存储实体与实体之间的关联关系

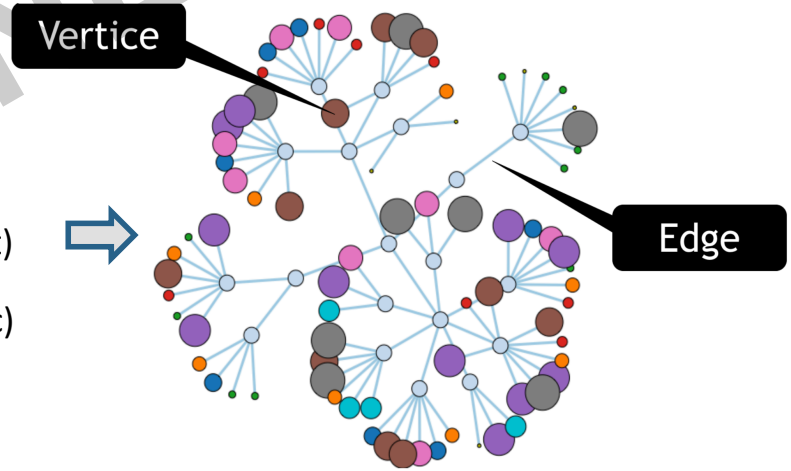


现实世界

G = (V, E)

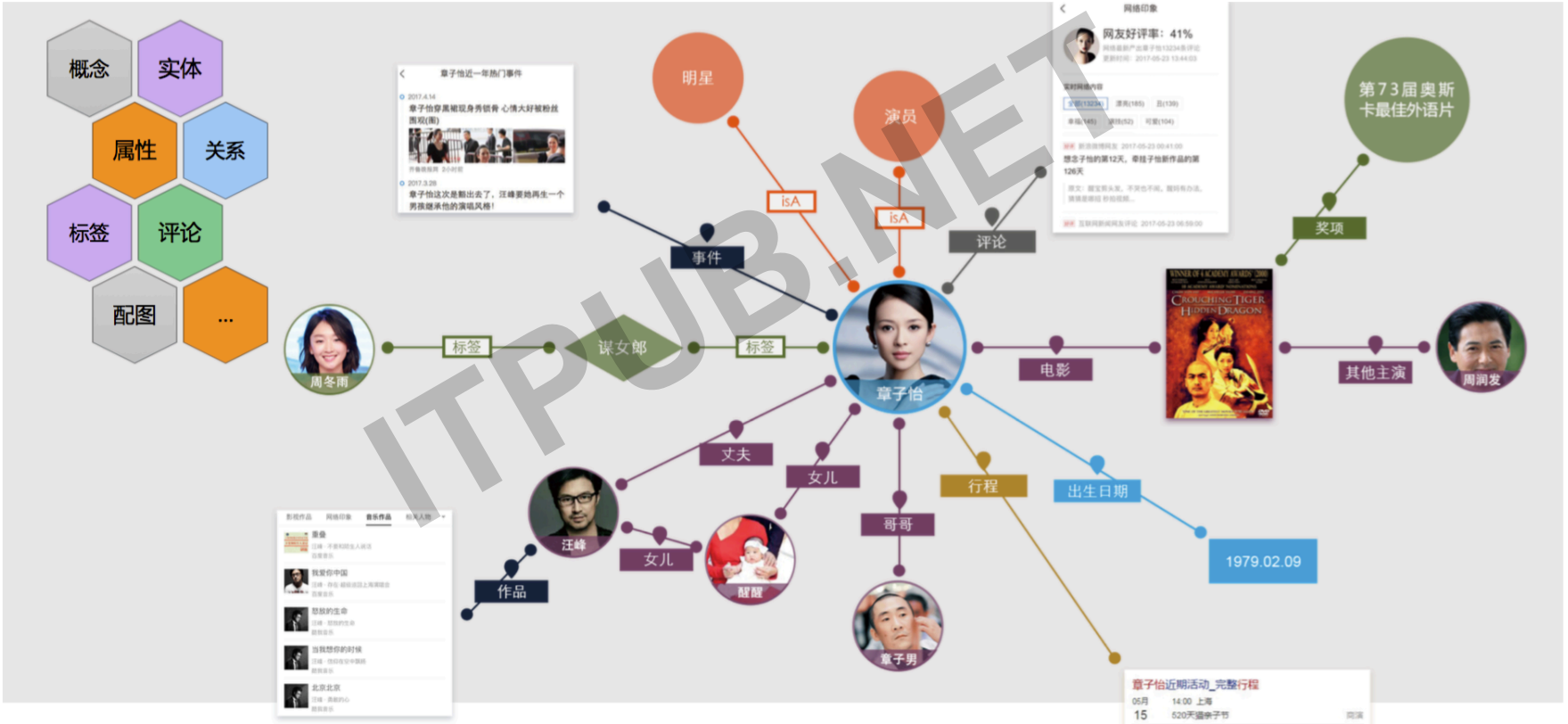
Vertex (Node, Entity, Object)

Edge (Relationship, Link, Arc)



图数据库

图数据库是什么



图数据库带来的改变

SQL

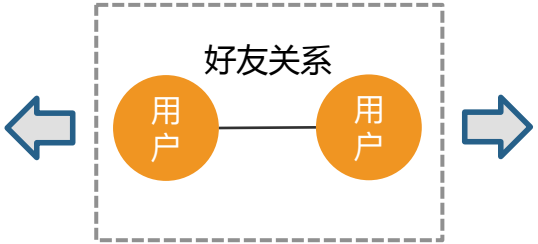
1. user

id	name	age	phone
1	Tom	22	188****1111
2	Mike	23	158****2222
…	…	…	…

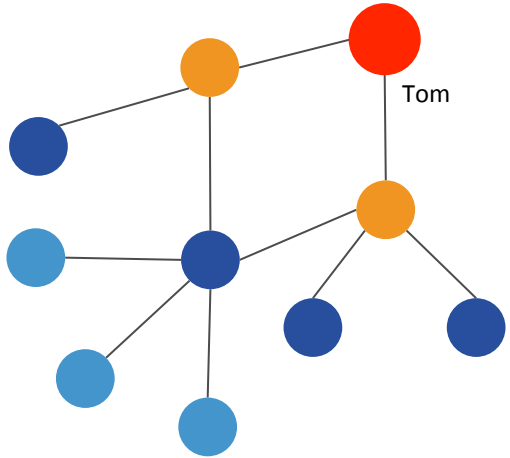
2. friend

id	user1	user2	since
1	1	2	2016-01-02
2	…	…	…

```
select * from friend a
    join friend b on b.user1=a.user2
    join user on id=b.user2
where a.user1 in (select id from
   user where name='Tom')
```



图数据库




```
g.V().has('name', 'Tom')
  .out('friend')
  .out('friend')
```

图数据库带来的改变

深度	关系型数据库执行时间 / s	图数据库执行时间 / s	返回结果数量
2	0.016	0.01	2500
3	30.27	0.168	110000
4	1543.51	1.359	600000
5	~	2.132	800000

数据来自《程序员》

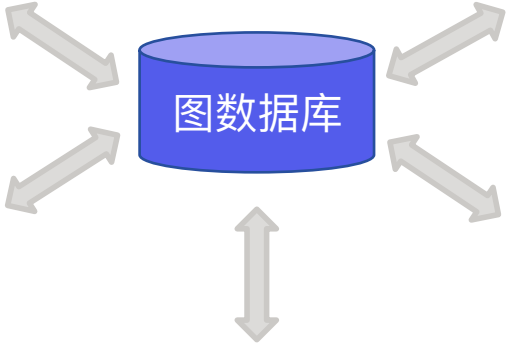
图数据库应用行业

反欺诈

发现人员、事件、地点和时间之间的异常联系

知识图谱

构建实体与实体间的关联关系，描述真实世界



网络安全

服务器、域名、IP、文件等关联关系，形成网络安全情报

IT运维

系统调用，数据库访问，缓存读取等全链路监控

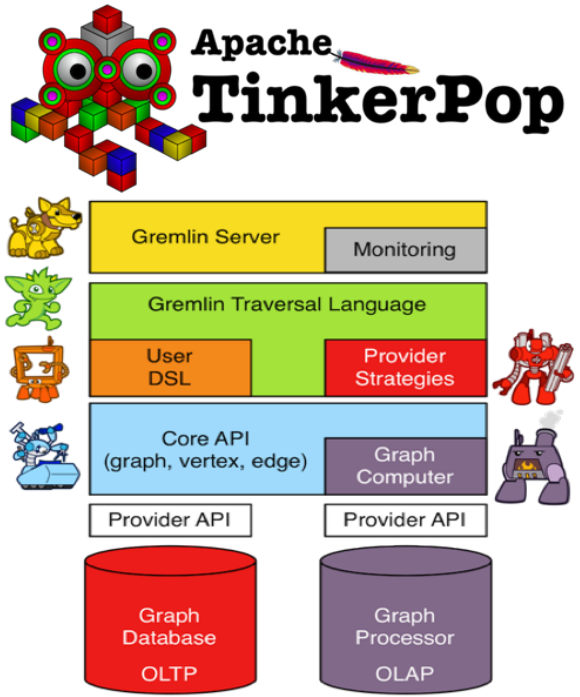
更多图数据库应用

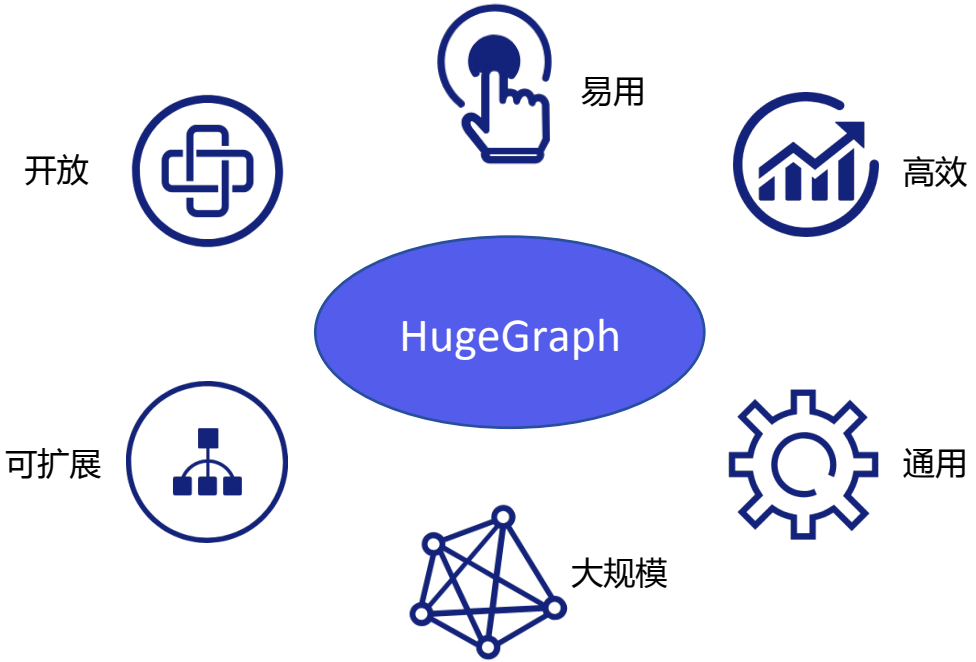
金融 社交 招聘 物流 医疗 电信 零售

HugeGraph是什么

百度安全自研的开源图数据库

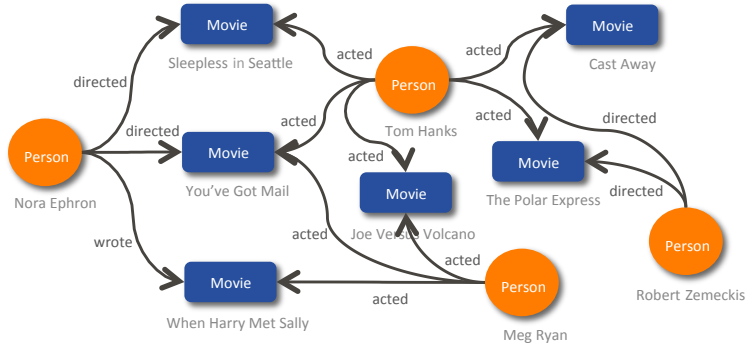
HugeGraph特点



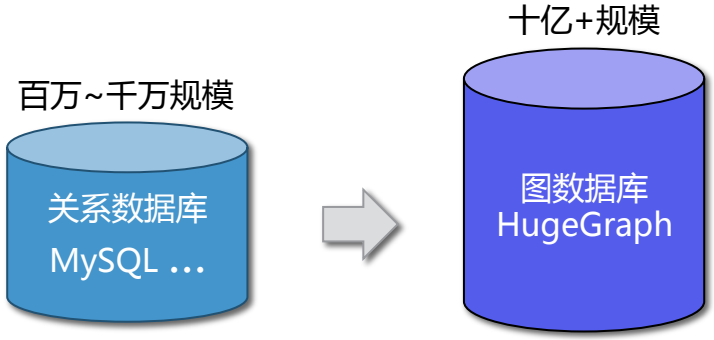


HugeGraph优势

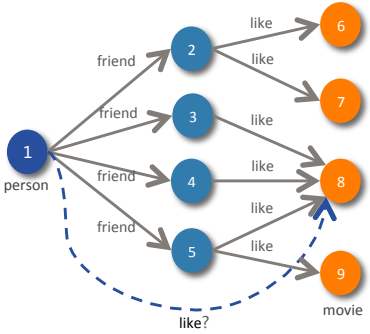
易建模

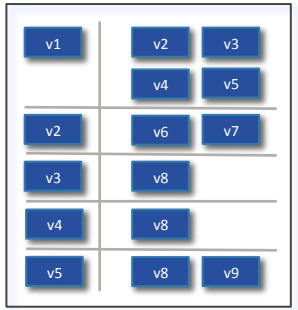


大规模



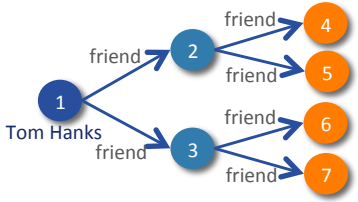
高效关联分析



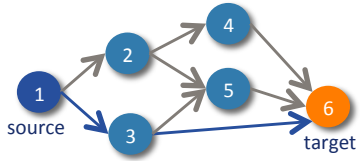


灵活查询语言

```
// Gremlin 2层好友查询
g.V()
  .has('name', 'Tom Hanks')
  .out('friend').out('friend')
```

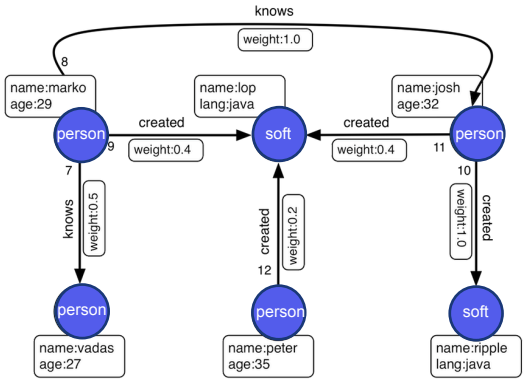



```
// 查询2点之间的最短路径
g.V(source_id)
  .repeat(out().simplePath())
  .until(hasId(target_id))
  .path().limit(1)
```



HugeGraph技术选型

1. 概念模型



- **Property Graph**
- RDF

2. 存储模型

A: (B,D)

B: (C,D)

C: (E)

D: (B,C,E)

E: (A,C)

F: (D,E)

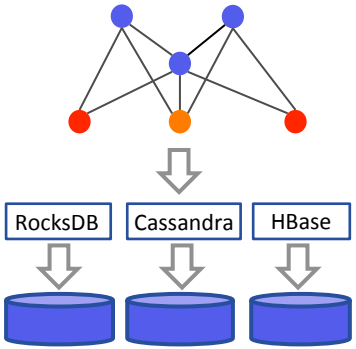
- **Adjacency List**
- Adjacency Matrix

3. 查询语言

```
g.V().has('name','gremlin')
     .out('knows')
     .out('knows')
     .values('name')
```

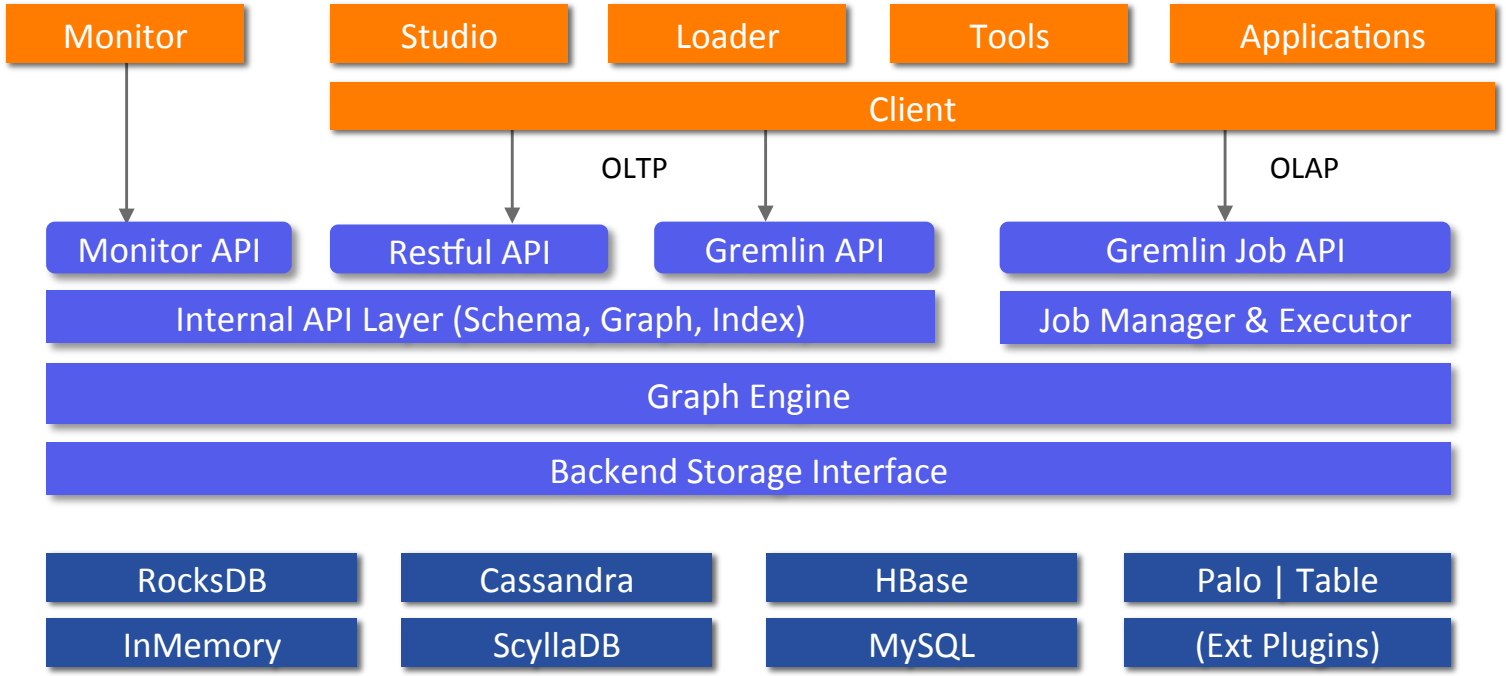
- **Gremlin**
- Cypher
- SPARQL

4. 持久化方案

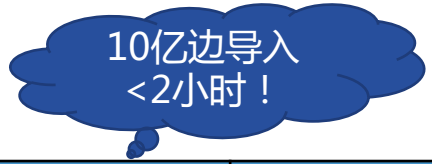


- **Non-Native Storage**
- Native Storage

HugeGraph整体架构



HugeGraph性能



批量写入性能

后端 \ 数据集	email-enron (30w edge)	amazon0601 (300w edge)	com-youtube.ungraph (300w edge)	com-lj.ungraph (3000w edge)
HugeGraph	1.726	13.066	13.009	141.212
Titan	14.02	125.975	154.926	1467.159
Neo4J	4.694	19.396	22.199	447.488

最短路径查询性能

后端 \ 数据集	email-enron (30w edge)	Amazon0601 (300w edge)	com-youtube.ungraph (300w edge)	com-lj.ungraph (3000w edge)
HugeGraph	2.289	0.242	10.218	28.78
Titan	13.326	0.577	544.492	848.36
Neo4J	2.001	3.899	5.937	28.925

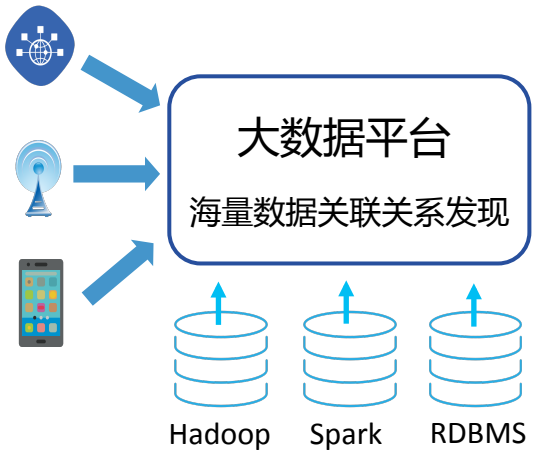
HugeGraph典型案例

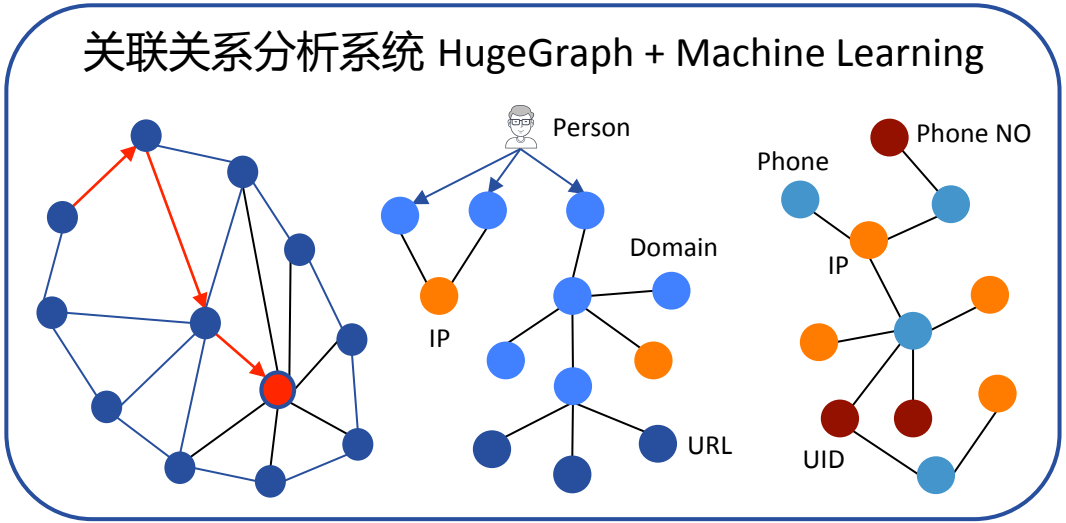
案例1：反黑产侦查　　案例2：基于知识图谱的广告推荐
案例3：金融反欺诈　　案例4：大数据安全治理

反黑产侦查

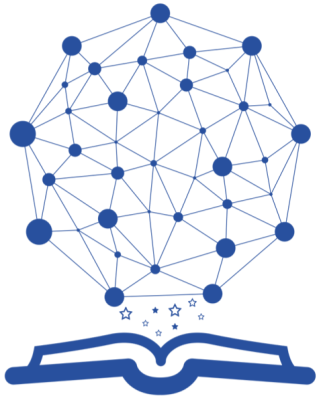
威胁情报
网址安全检测
IP信誉度
反刷单与作弊

活动事件关联图谱
网址 IP/URL关联图谱
设备ID、手机号账号关联图谱

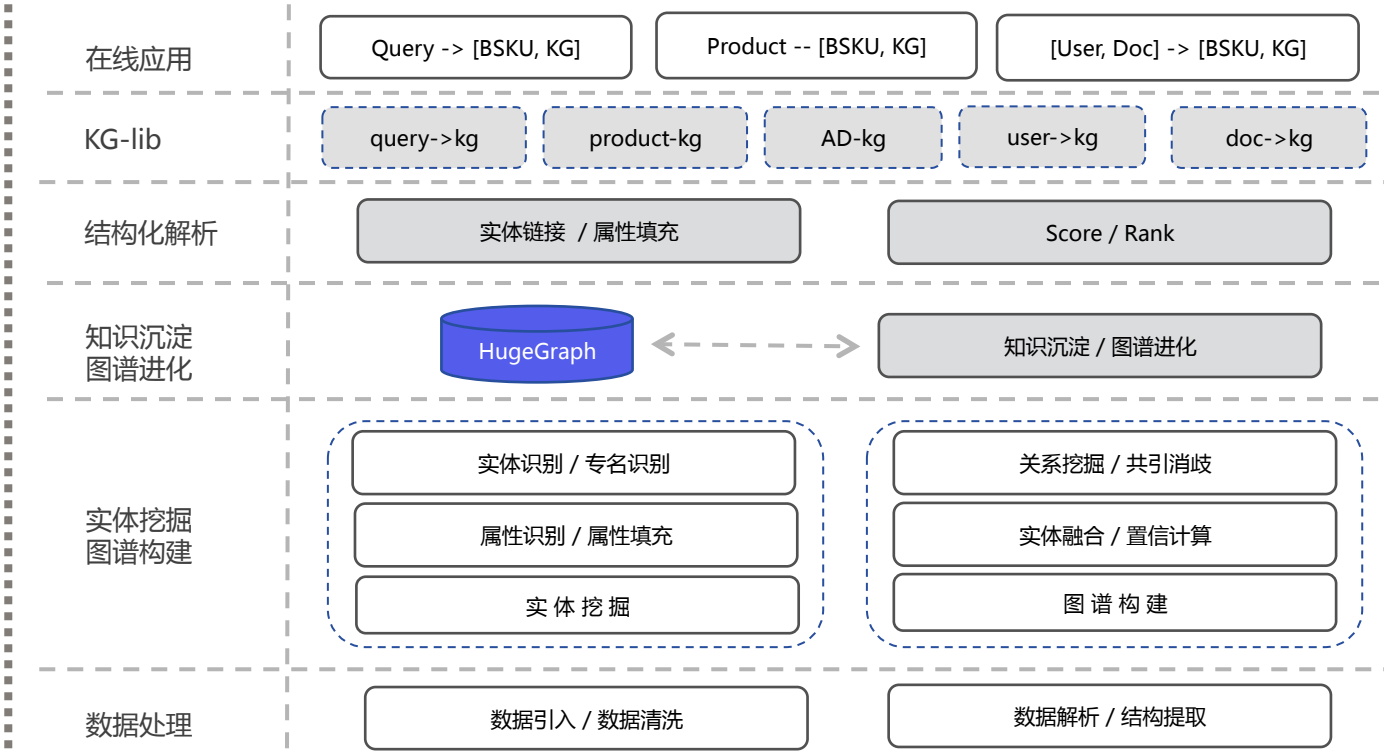




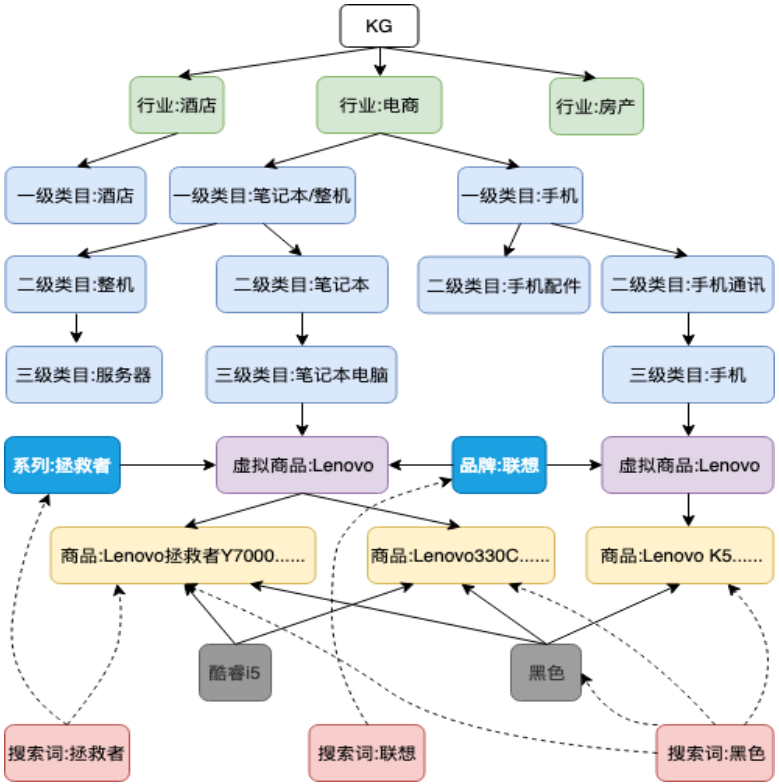
基于知识图谱的广告推荐

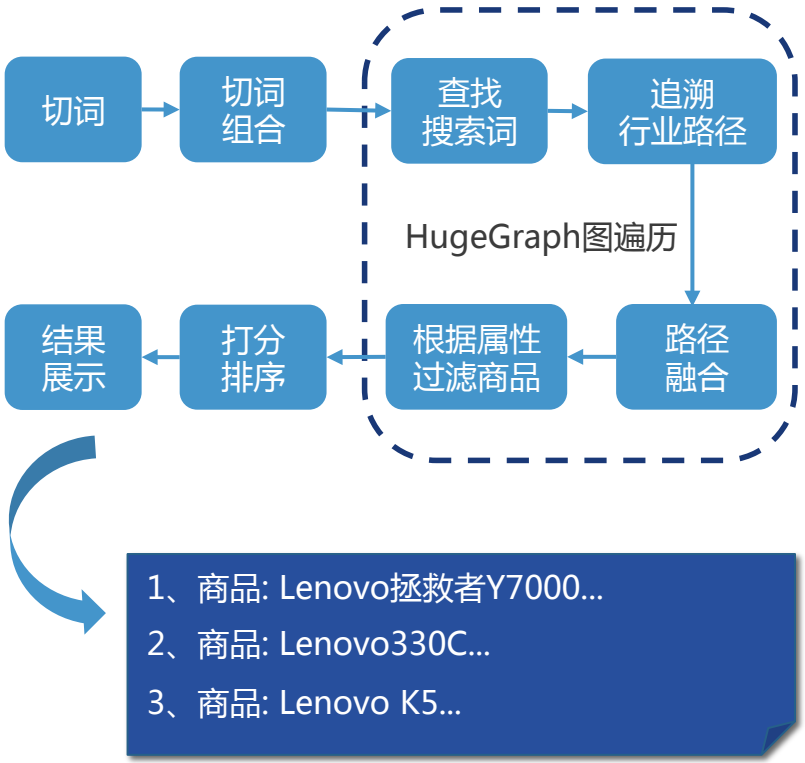


- 实体和实体关系
- 蕴含规则和知识
- 典型图数据应用场景

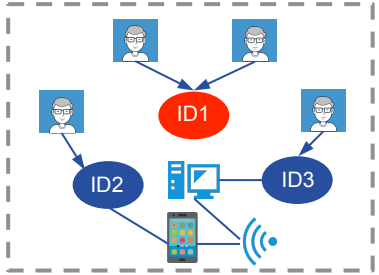


基于知识图谱的广告推荐

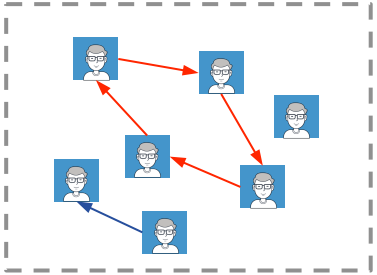




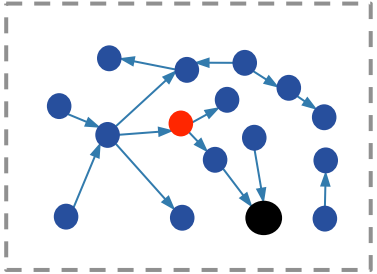
金融风控



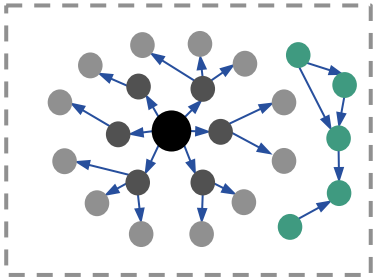
ID唯一性检查



循环担保检测

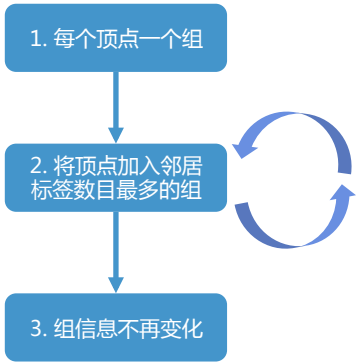


二度关系触黑

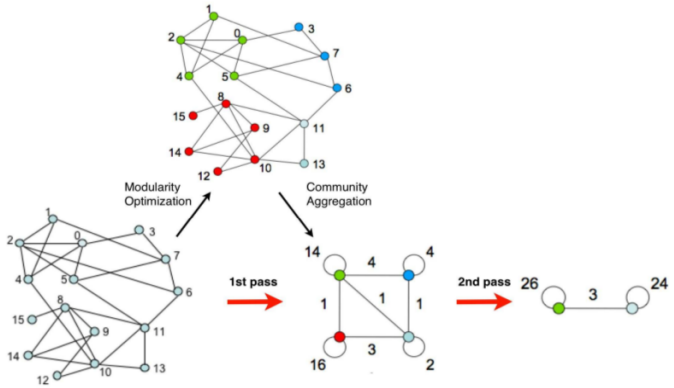


黑用户风险评分扩散

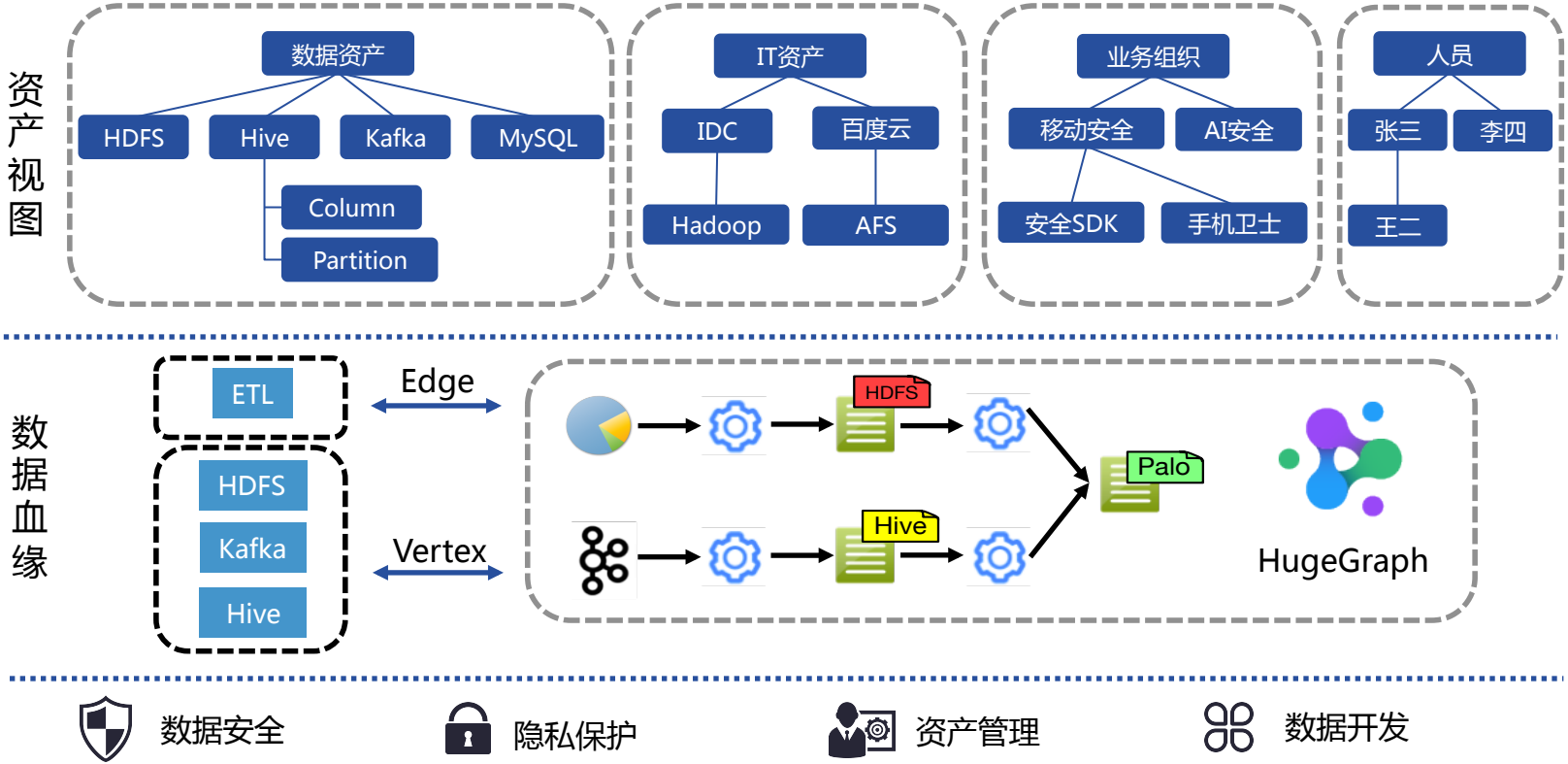
LPA 算法



Louvain 算法



大数据安全治理

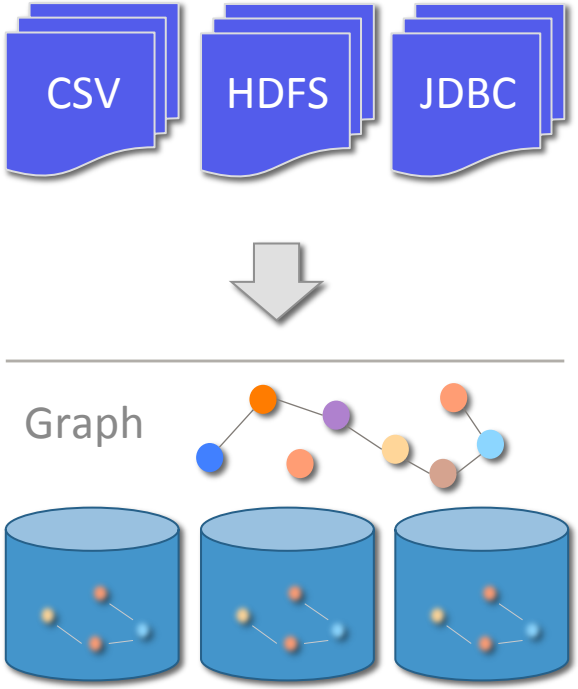

资产视图
数据资产
HDFS
Hive
Kafka
MySQL
Column
Partition
IT资产
IDC
百度云
Hadoop
AFS
业务组织
移动安全
AI安全
安全SDK
手机卫士
人员
张三
李四
王二
数据血缘
ETL
HDFS
Kafka
Hive
Edge
Vertex
HDFS
Hive
Palo
HugeGraph
数据安全
隐私保护
资产管理
数据开发

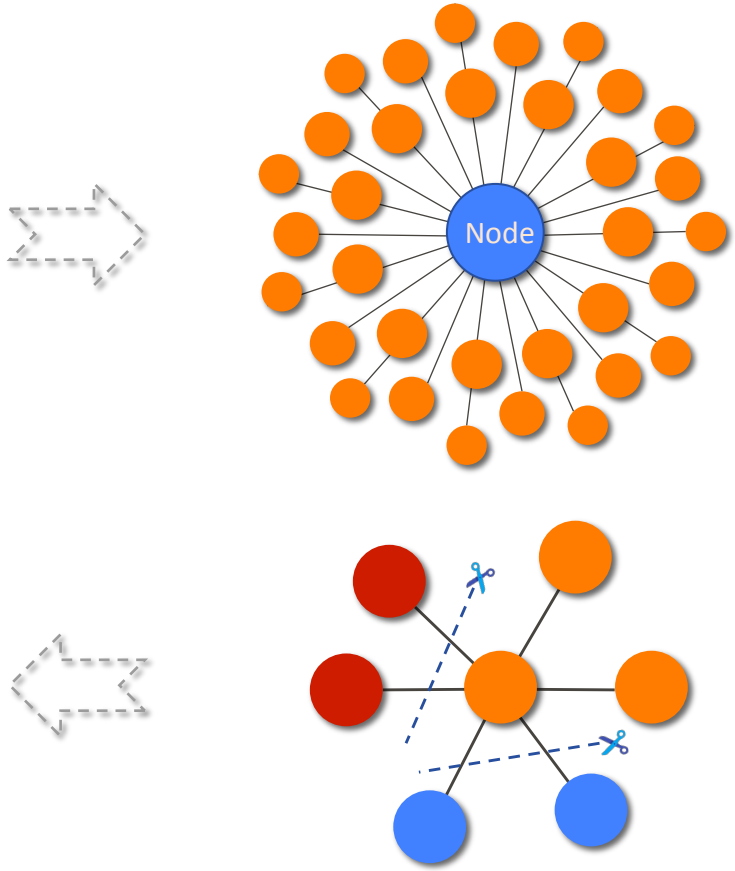
HugeGraph存储原理

存储结构：边集数组、邻接矩阵、邻接表、十字链表

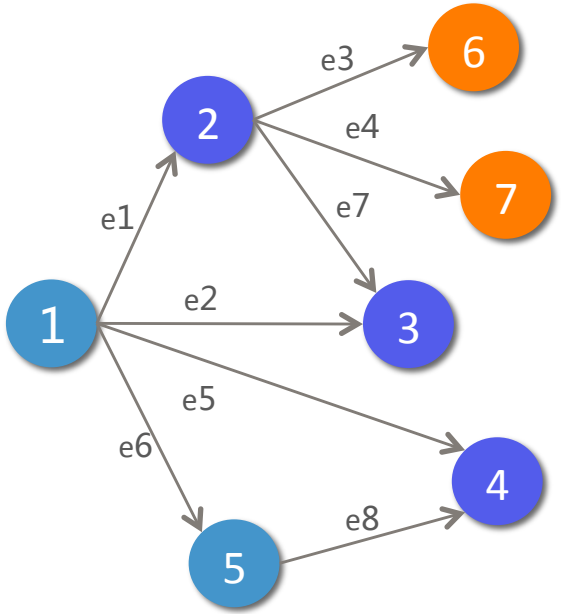
HugeGraph存储结构：邻接表（顺序表）

图存储结构



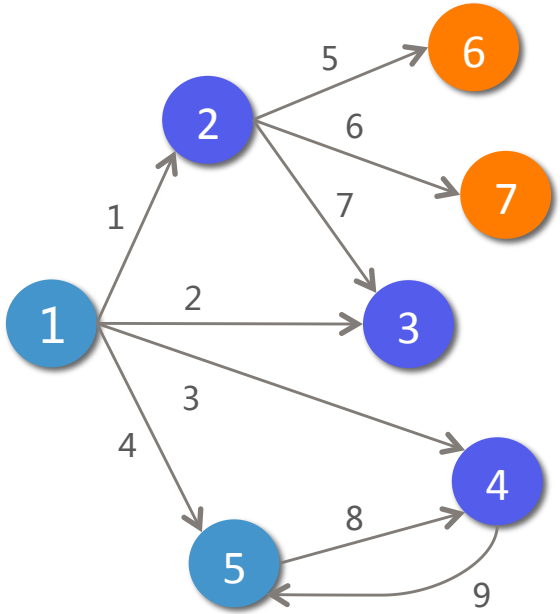


边集数组



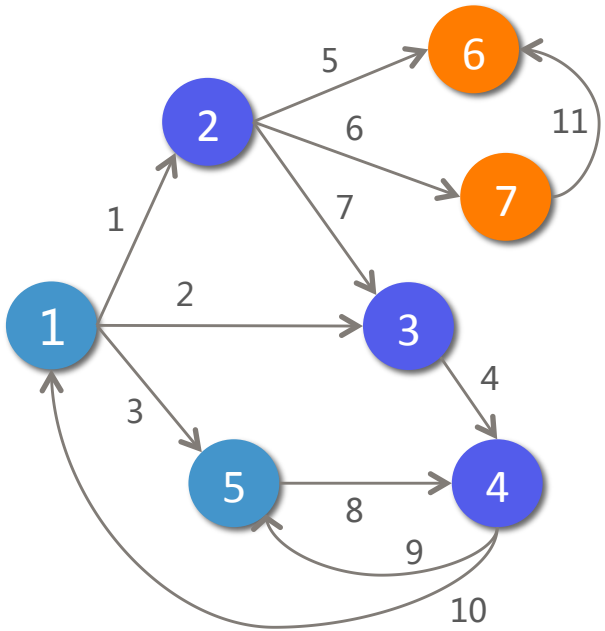
边ID	源顶点	目标顶点	边属性
e1	v1	v2	weight
e2	v1	v3	weight
e3	v2	v6	score
e4	v2	v7	score
e5	v1	v4	weight
e6	v1	v5	weight
e7	v2	v3	weight
e8	v5	v4	weight

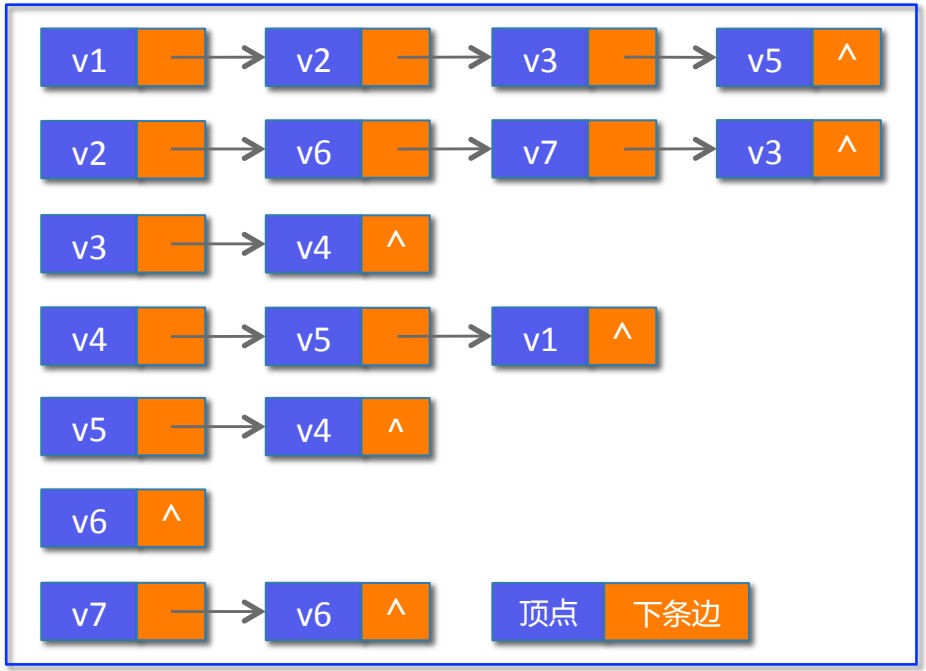
邻接矩阵



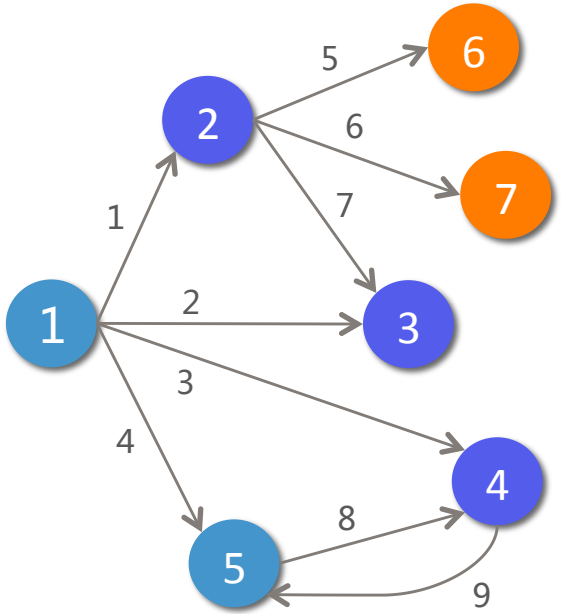
	v1	v2	v3	v4	v5	v6	v7
v1	0	1	2	3	4	0	0
v2	0	0	7	0	0	5	6
v3	0	0	0	0	0	0	0
v4	0	0	0	0	9	0	0
v5	0	0	0	8	0	0	0
v6	0	0	0	0	0	0	0
v7	0	0	0	0	0	0	0

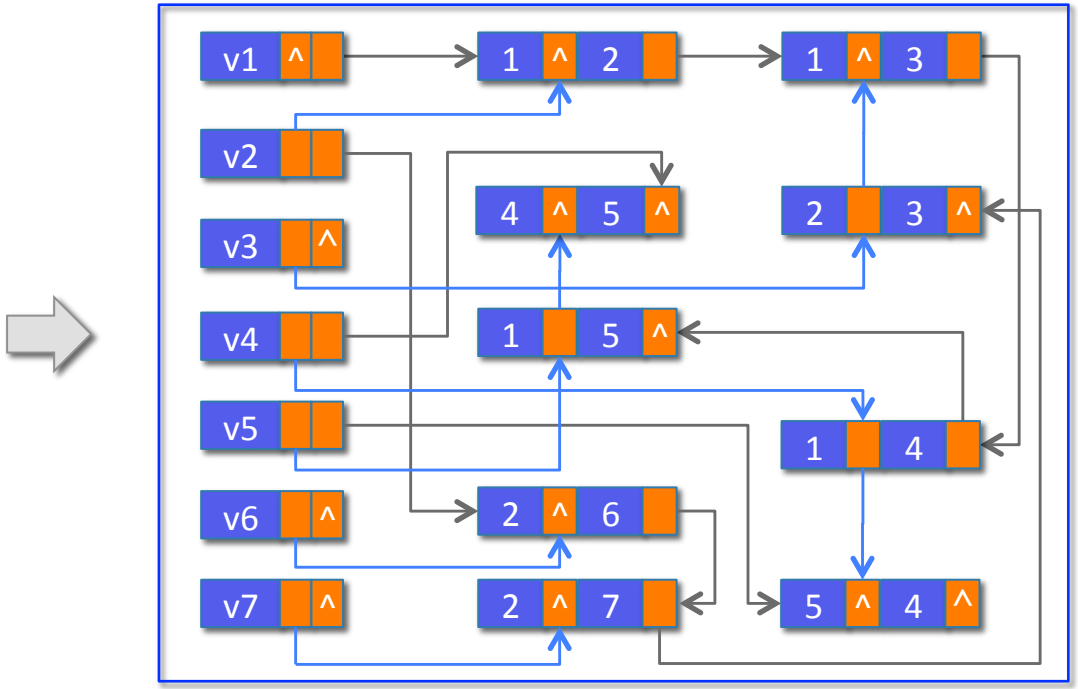
邻接表


1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11

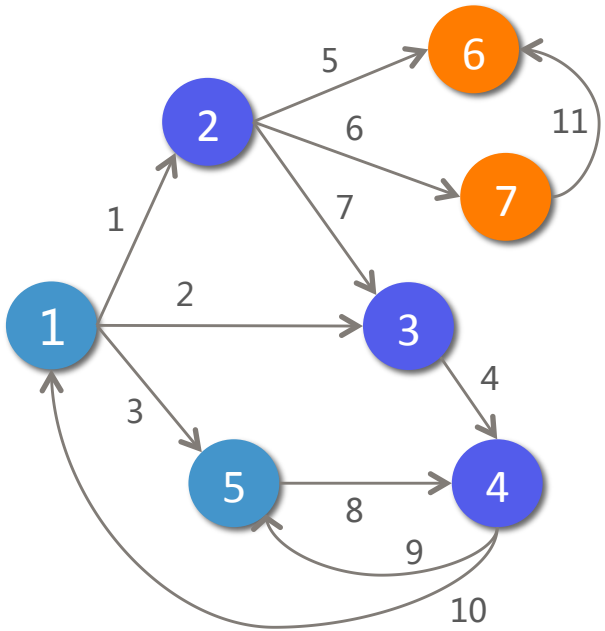

v1
v2
v3
v5
^
v2
v6
v7
v3
^
v3
v4
^
v4
v5
v1
^
v5
v4
^
v6
^
v7
v6
^
顶点
下条边

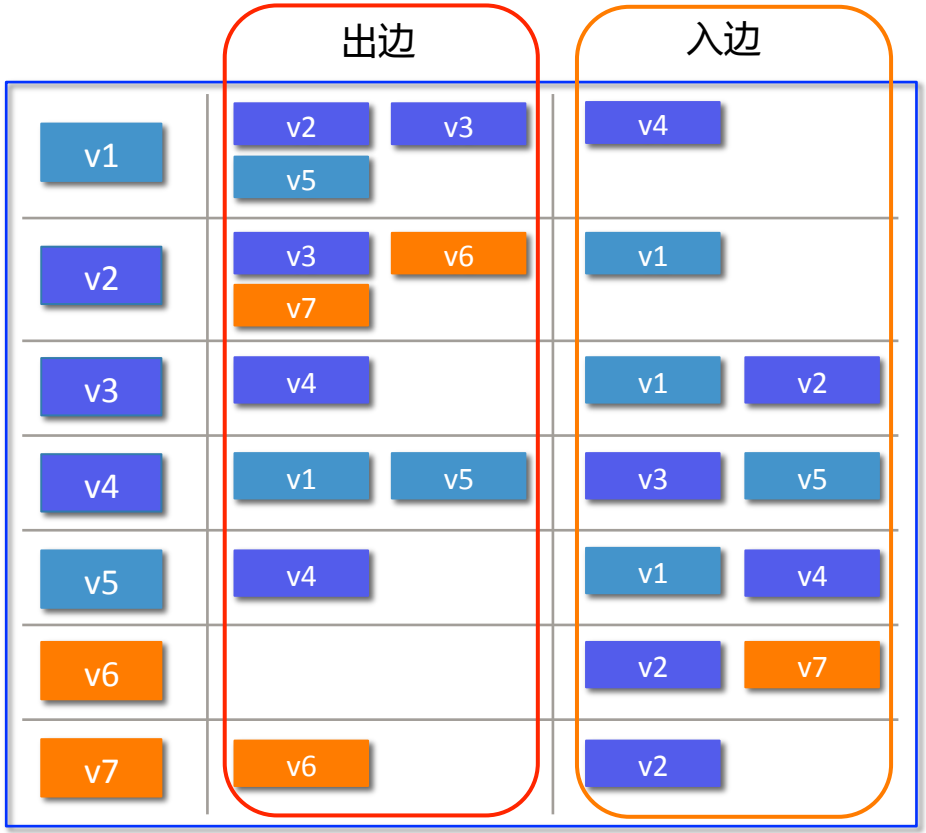
十字链表





邻接表（顺序表）


1
2
3
4
5
6
7
1
2
3
4
5
6
7
8
9
10
11

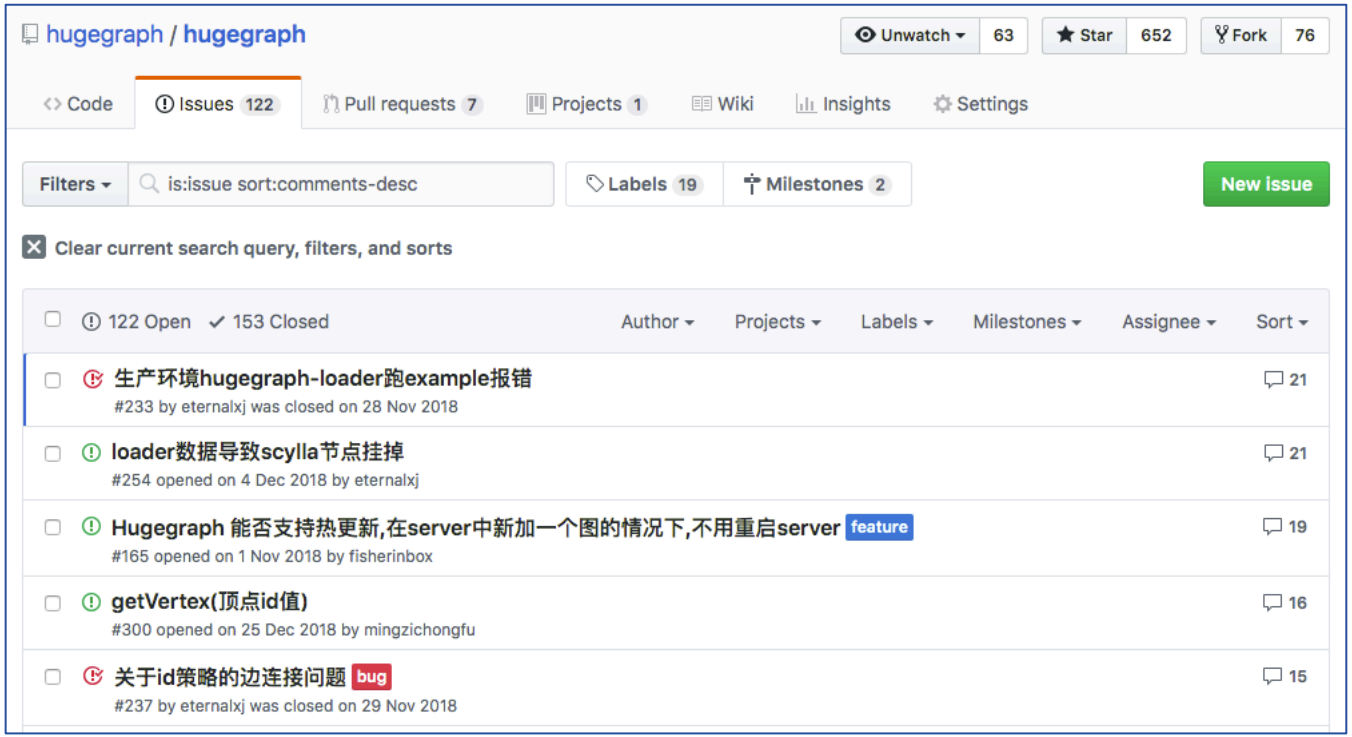

出边
入边
v1
v2
v5
v4
v2
v3
v6
v7
v1
v3
v4
v1
v2
v4
v1
v5
v3
v5
v5
v4
v1
v4
v6
v2
v7
v7
v6
v2

如何参与开源贡献

Issue 问题与需求反馈

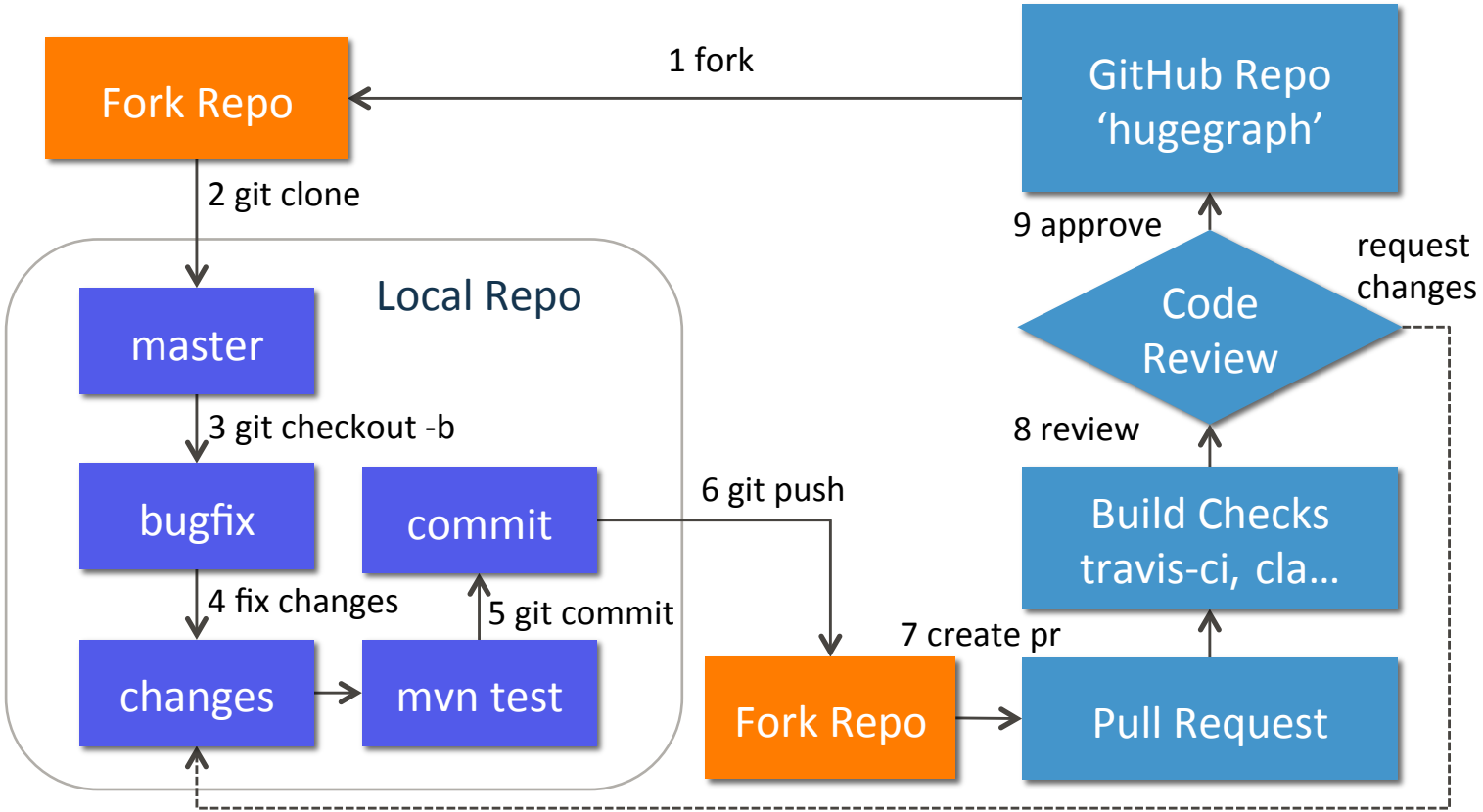
Pull Request 贡献代码与评审

问题与需求反馈



https://github.com/hugegraph/hugegraph/issues

贡献代码与评审


Fork Repo
1 fork
GitHub Repo
'hugegraph'
2 git clone
9 approve
request
changes
Code
Review
Local Repo
master
3 git checkout -b
8 review
bugfix
commit
6 git push
Build Checks
travis-ci, cla...
4 fix changes
5 git commit
7 create pr
changes
mvn test
Fork Repo
Pull Request

微信群

Thanks！ Q & A ？

总结：

图数据库是什么、HugeGraph是什么、HugeGraph典型案例、HugeGraph存储原理、如何参与开源贡献