

图引擎服务 GES 使用——实时推荐

1. 任务介绍

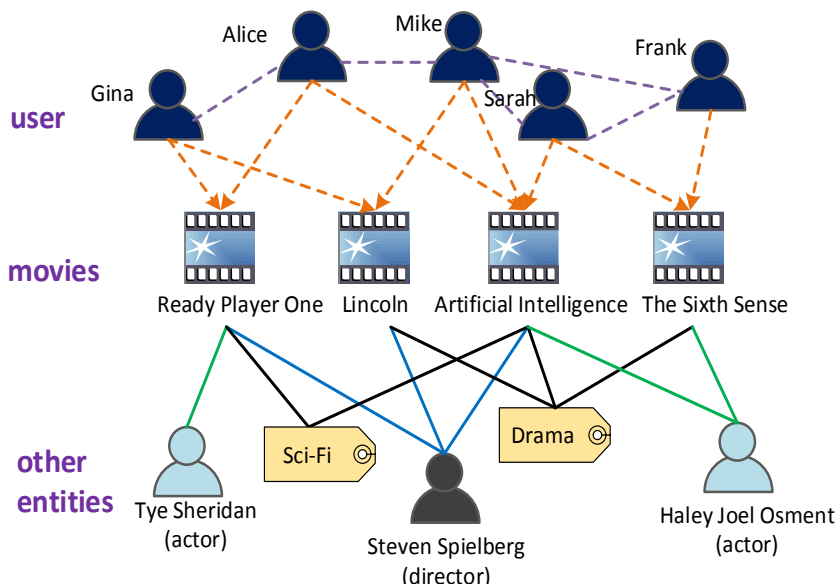
本次任务将介绍如何基于图引擎服务进行大规模、复杂场景的实时推荐。

场景介绍：

推荐技术在社交 APP、电商、音乐视频图片平台、资讯推送平台等场景中大展身手，带给用户便利的同时，提高了用户平台活跃度，增加了“物品”销量。但是，随着互联网技术的飞速发展，所能够获取的数据量和数据类型越来越多，如何利用这些错综复杂的数据，挖掘背后潜在关系，进行快速有效的推荐成为各公司必须思考的问题。

目前的推荐算法层出不穷，数据利用效率、推荐的效果、速度也不尽相同。例如，在电影推荐场景，传统基于协同过滤（CF）的方法仅考虑到左半边人对电影的 feedback 数据，而传统基于内容（CB）的方法仅考虑右半边电影的特性数据，这样单纯的依赖部分数据进行推荐难以满足用户更深层次的推荐需求，同时存在稀疏性等问题；基于深度学习的推荐是近期比较热的方向，推荐效果较传统方法也有很大的提高，但却以牺牲时间性能为代价，推荐的结果解释性也较低。

基于图引擎的实时推荐，不仅可以综合考虑多重关系，还可以针对各种复杂场景进行综合推荐，推荐更为精准，有效避免“信息茧房”、解决推荐中数据稀疏性以及冷启动等难题，同时，大数据量下能够做到快速实时推荐，可扩展性强，并支持数据动态变化。



2. 任务执行

2.1 在图引擎服务上创建图

本次课程采用的电影推荐数据（包括元数据 xml 文件、点文件、边文件）可直接从第 3 章节“打卡任务”提供的链接中下载获得（其中的，demo-data 文件夹）。

2.1.1 创建元数据：

创建方法：

目前图引擎服务支持两种元数据创建方式：一种是直接创建，一种是导入已经定义好的 xml 文件：

方法一：直接创建方式。点击“[元数据管理](#) > [创建](#)”，添加元数据中的 label 以及对应的 property，点击“确认”完成创建。

方法二：导入 xml 文件创建方式。点击“元数据管理 > 导入”，通过导入定义好的 xml 文件，创建元数据，元数据 xml 文件的格式可参考：《用户指南》中的“图数据的格式” -> schema 章节

创建好的元数据示例如下：

图引擎服务

总览

图管理

备份管理

元数据管理

元数据管理 > schema_demo

名称

schema_demo

存储路径

data-ges/demo_schema/schema_demo.xml

定义

Label名称

user

Property名称	基数	数据类型
ChineseName	单值	string
Gender	单值	enum 枚举值: FM
Age	单值	enum 枚举值: Under 18,18-14,25-34,35-44,45-49,50-55,56+
Occupation	单值	string
Zip-code	单值	char array 最大长度: 12

Label名称

movie

Property名称	基数	数据类型
ChineseTitle	单值	string
year	单值	int

Label名称

actor

Property名称	基数	数据类型
------------	----	------

Label名称

director

Property名称	基数	数据类型
------------	----	------

Label名称

genre

Property名称	基数	数据类型
------------	----	------

Label名称

like

Property名称	基数	数据类型
Datetime	单值	date

Label名称

friends

Property名称	基数	数据类型
------------	----	------

Label名称

hasActor

Property名称	基数	数据类型
------------	----	------

Label名称

hasDirector

Property名称	基数	数据类型
------------	----	------

Label名称

hasGenre

Property名称	基数	数据类型
------------	----	------

编辑

复制

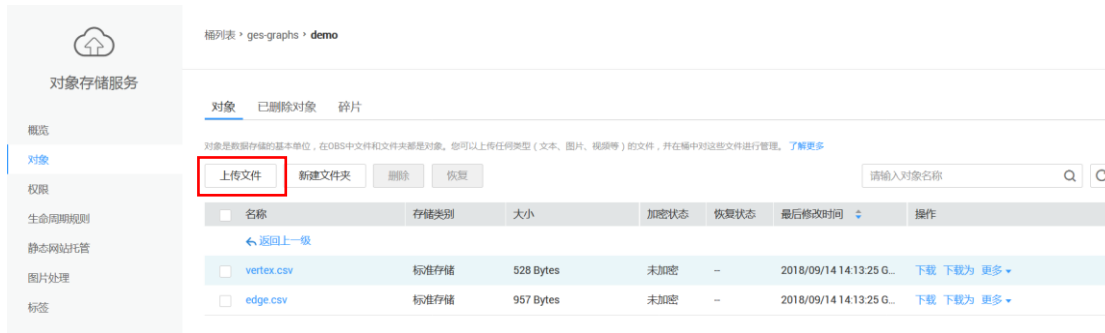
关闭

点的Label

边的Label

2.1.2 准备点和边数据：

准备点和边数据集，并上传至 [OBS](#):



点文件中，每一行为一个点的数据，每行点数据格式

id 1, label, property 1, property 2, ...

边文件中，每一行为一条边的数据，每行边数据格式：

id 1, id 2, label, property 1, property 2, ...

例如，本示例中：

（1）点数据文件 vertex.csv 内容：

```
Frank,user,弗兰克,M,18-24,college/grad student,2906
Mike,user,麦克,M,25-34,other or not specified,30606
Sarah,user,莎拉,F,18-24,other or not specified,55105
Alice,user,爱丽丝,F,25-34,academic/educator,79928
Gina,user,吉娜,F,35-44,sales/marketing,60202
Ready Player One,movie,头号玩家,2011
Lincoln,movie,林肯,2012
Artificial Intelligence,movie,人工智能,2001
The Sixth Sense,movie,灵异第六感,1999
Haley Joel Osment,actor
Tye Sheridan,actor
Steven Spielberg,director
Drama,genre
Sci-Fi,genre
```

（2）边数据文件 edge.csv 内容：

```
Frank,The Sixth Sense,like,2017/12/22 23:24
Mike,Artificial Intelligence,like,2017/12/22 20:20
Mike,Lincoln,like,2017/12/23 13:08
Sarah,The Sixth Sense,like,2017/12/22 22:00
Sarah,Artificial Intelligence,like,2017/12/23 21:05
Alice,Artificial Intelligence,like,2017/12/22 10:24
Alice,Ready Player One,like,2017/12/23 17:25
Gina,Lincoln,like,2017/12/22 16:30
Gina,Ready Player One,like,2017/12/22 21:37
Frank,Mike,friends
Frank,Sarah,friends
Mike,Sarah,friends
Mike,Alice,friends
Alice,Gina,friends
The Sixth Sense,Haley Joel Osment,hasActor
The Sixth Sense,Drama,hasGenre
Artificial Intelligence,Drama,hasGenre
Artificial Intelligence,Sci-Fi,hasGenre
Artificial Intelligence,Haley Joel Osment,hasActor
Artificial Intelligence,Steven Spielberg,hasDirector
Lincoln,Steven Spielberg,hasDirector
```

Lincoln,Sci-Fi,hasGenre
Ready Player One,Steven Spielberg,hasDirector
Ready Player One,Sci-Fi,hasGenre
Ready Player One,Tye Sheridan,hasActor

2.1.3 配置图规格，创建图：

创建图

区域：北京-1 温馨提示：页面左上角切换区域

可用区：可用区1 可用区2

图名称：ges_demo

GES 软件版本：1.1.5

虚拟私有云：vpc-ffff 查看虚拟私有云

子网：subnet-b296 (192.168.0.0/24)

安全组：eywa 如何配置安全组？ 查看安全组

公网访问：暂不使用 现在购买 使用已有

不启用弹性IP的图实例不能与互联网互通，仅可通过私有网络中已部署的弹性云网卡连接当前图实例使用。

用途：企业生产 开发学习

完整功能体验，适合开发者学习使用

图规格（边数）：1-1024

元数据：schema_demo 创建新的元数据 下载模板

边数据集：ges-graphs/demo/edge.csv 选择 下载模板

点数据集：ges-graphs/demo/vertex.csv 选择 下载模板

图引擎服务 ¥1.00/小时

参考价格，具体扣费请以账单为准。了解计费逻辑

立即购买

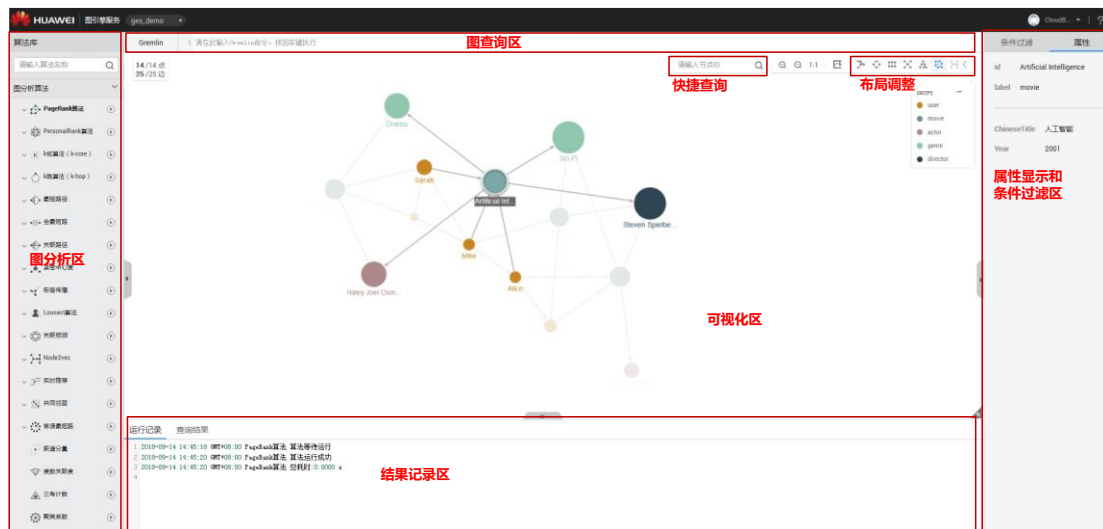
2.1.4 图运行状态变为运行中时，点击“访问”，访问对应的图界面。

图管理

您还可以创建1张图，使用1.98GB边

全部状态 输入图名称查询

图名称	运行状态	内网访问地址	公网访问地址	创建时间	操作
movie_recommender	已停止	192.168.0.209	--	2018/08/26 22:48:22 G...	访问 备份 更多
ges_demo	运行中	192.168.0.178	--	2018/08/27 16:50:15 G...	访问 备份 更多



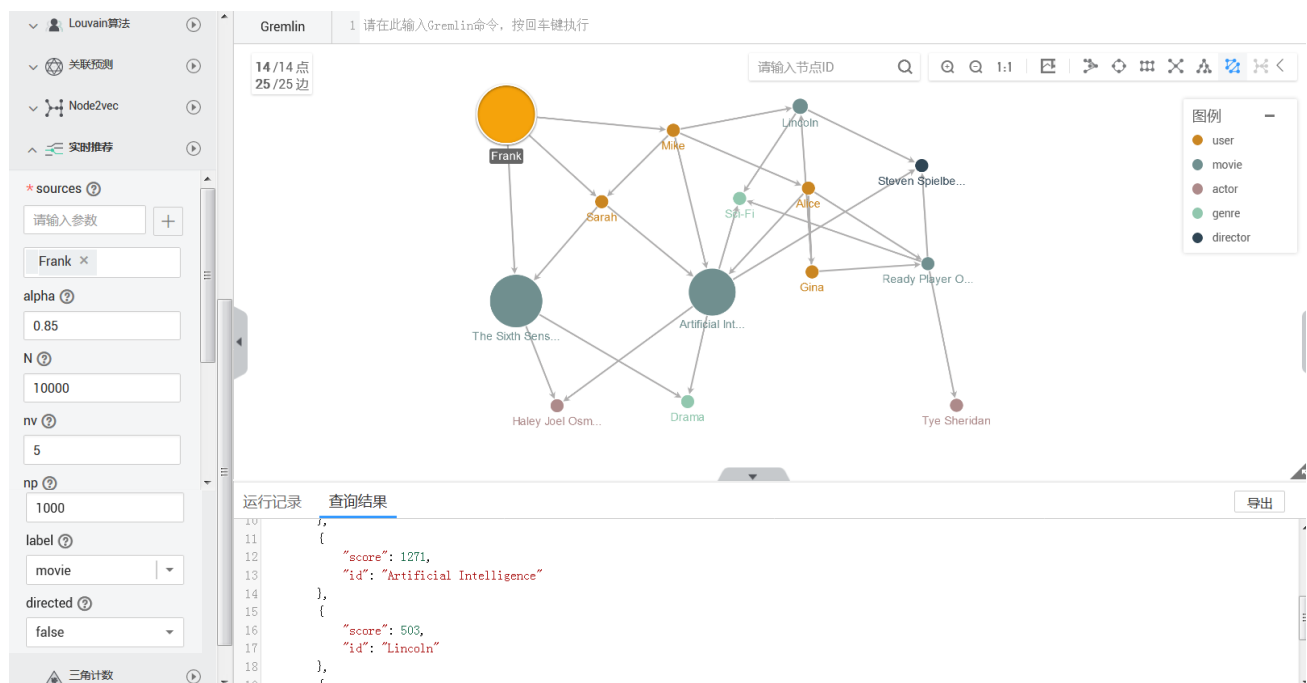
2.2 在图引擎服务上进行实时推荐

2.2.1 使用实时推荐算法进行推荐。

选择“实时推荐”(Real-time Recommendation)算法根据提示选择左侧栏的参数，这里 sources 栏用于输入请求节点（可以多个，标准的 csv 数据格式输入），label 参数过滤出期望得到的节点类型。

例如：

（1）“当 Frank 打开电影平台，向其推荐电影”时，可以输入 sources 为 Frank, label 为 movie，其他参数如下图（左），点击运行。



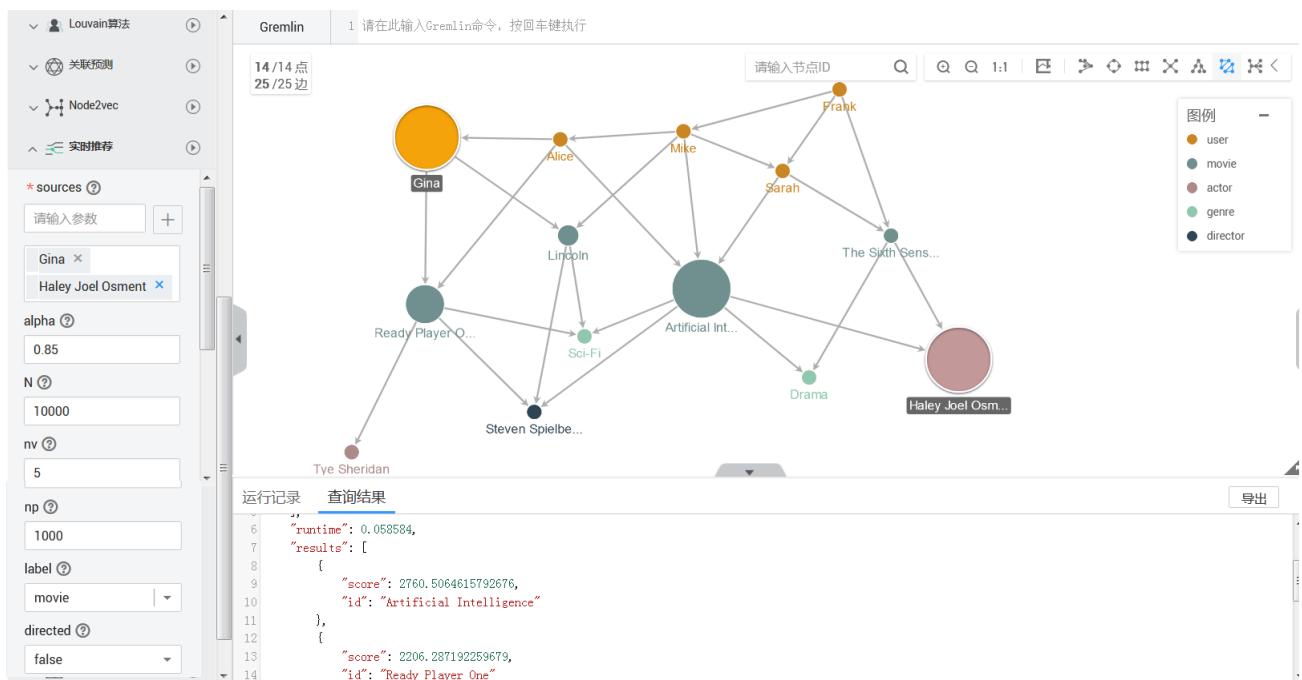
算法运行结束后，画布上将会出现含有推荐节点的子图（大小反映了推荐程度，点击右上角图例上的相关项目，可以快速锁定相应类型节点）；Pixie 的结果将会出现在下方“查询结果”栏。

由图引擎的运算结果可以很清晰显示用户、电影、演员、导演等之间的关系，根据多因素共同作用结果，演算出最为精确的推荐结果，相较于过去的推荐结果，精度更高，解释能力更强。

(2) 当某“游客”用户小易登陆平台，浏览了 Gina 主页以及搜索了演员 Haley Joel Osment 动态，这一行为具有很强的个人偏好属性说明他对 Gina 以及 Heley Joel Osment 有兴趣，这时，我们可以输入多个 sources 节点（参数如下图左），综合考虑这两种兴趣（“Gina”、“Haley Joel Osment”）进行实时推荐：

sources：Gina，Haley Joel Osment

label: movie



值得注意的是：

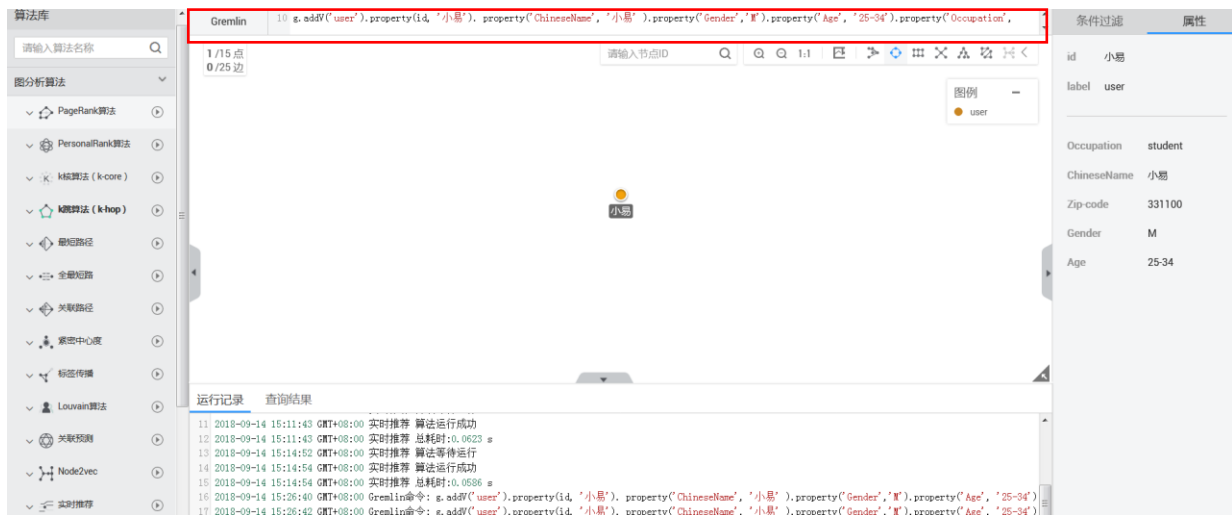
图引擎因为是基于图的结构进行实时计算，因此可以适应数据的动态变化，如上图“小易刚浏览了好友 Gina 主页，添加巨星 Matt Damon 的动态关注”这一行为，可以通过 gremlin 等语句实时更新到当前图数据里，做到实时更新，实时推荐。不仅可以考虑到考虑小易的近期倾向性行为，同时考虑小易特有的各类关系（人→电影，人→人，电影→电影等等），这样的实时推荐，可以极大增加推荐的准确度，从而增加点击率，提高转化率

2.3 动态更新图，实时更新，实时推荐

基于图的结构进行实时计算，可以适应数据的动态变化，如上图“小易加入平台，看了电影 Ready Play One”这一行为，图引擎服务可以通过 gremlin 等语句实时更新到当前图数据里，做到实时更新，实时推荐。

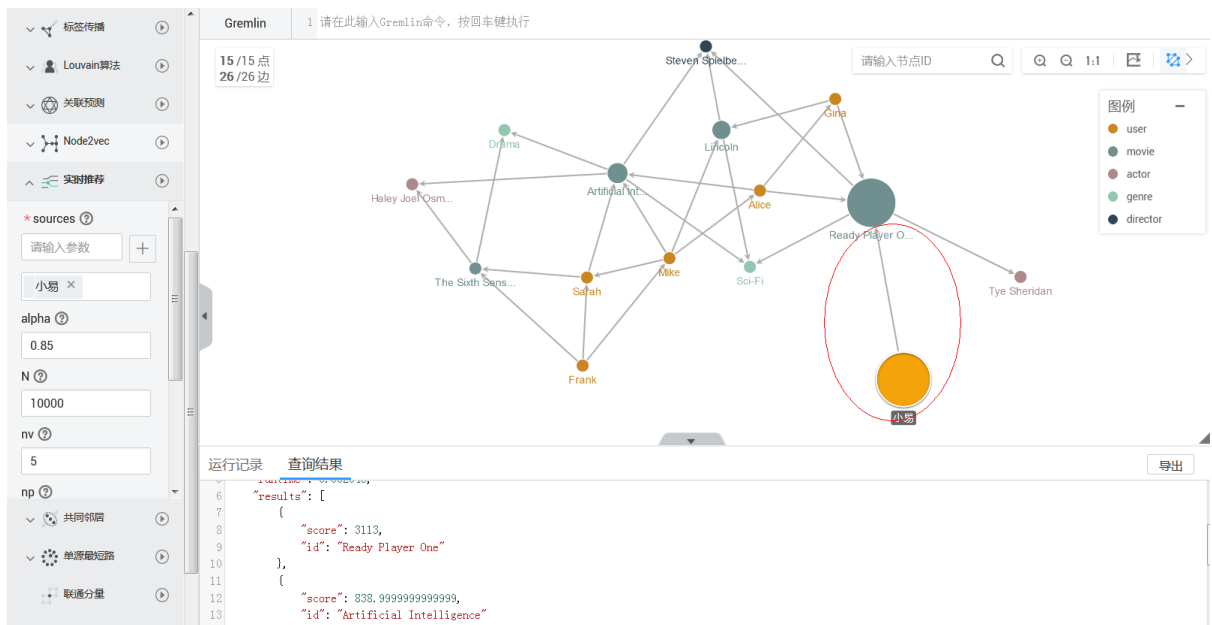
(1) Gremlin 增加点，输入 Gremlin 语句：

```
g.addV('user').property(id, '小易').property('ChineseName', '小易').property('Gender', 'M').property('Age', '25-34')
.property('Occupation', 'student').property('Zip-code', 331100)
```



(2) Gremlin 增加边，输入 Gremlin 语句：

`g.V('小明').addE('like').property('Datetime', '2018-07-27, 19:49:20').to(g.V('Ready Player One'))`



扩展：

Gremlin 是 Apache TinkerPop 框架下的图遍历语言。Gremlin 是一种函数式数据流语言，可以使得用户使用简洁的方式表述复杂的属性图（property graph）的遍历或查询。每个 Gremlin 遍历由一系列步骤（step，可能存在嵌套）组成，每一步都在数据流（data stream）上执行一个原子操作。

用户只需要将标准 gremlin 语言进行简单的组合和嵌套，就能完成复杂的查询功能，例如：

查询图上的点：g.V().limit(100)

查询 Frank 用户的信息：g.V("Frank")

查询 Frank 喜欢的电影：g.V("Frank").out("like").hasLabel("movie")；

查询 Frank 朋友喜欢的电影：g.V("Frank").out("friends").out("like").hasLabel("movie")；

等等。

综上，可以看出，图引擎服务提供的实时推荐算法，在多重关系（用户与商品的历史交互信息，人与人、商品与商品等背后潜在关系）共同作用下进行推荐，精度更高；**支持多请求节点下综合推荐，满足各种复合型、时变、多样性推荐场景需求；大数据量下速度，依然能够达到较好的实时推荐效果，可扩展性强**，欢迎大家进行试用。

3. 打卡任务

按上一章节所讲内容，在图引擎服务上创建图（创建图时，图规格参数中的“用途”选项请选择为“开发学习”），并进行电影推荐相关实践操作：

样例数据：图数据可采用第 2 章节示例数据，即，以下链接中的 demo-data；

<https://ges-training-data.obs.myhwclouds.com/huawei-ges-training-data-v1.0.zip>

实践操作：

3.1 运行 GES 提供的“实时推荐”算法：

(a) 对用户 Frank 进行电影推荐：

设置参数：sources=Frank; label=movie; directed=false；

(b) Frank 的某个朋友，浏览了 Frank 主页，添加 Steven Spielberg 导演动态关注，为其进行电影推荐

设置参数：sources=Frank,Steven Spielberg; label=movie; directed=false；

(c) 某人搜索了 Steven Spielberg、Sci-Fi 关键词，为其进行电影推荐：

设置参数：sources= Steven Spielberg,Sci-Fi; label=movie; directed=false；

3.2 图上查询以及更新图

Frank 近期接受了“实时推荐”算法推荐的电影‘Artificial Intelligence’，将这部电影标志为喜欢，通过 Gremlin 语句加入 Frank 与电影“Artificial Intelligence”这条关系，进行图的动态更新：

g.V("Frank").addE("like").property("Datetime", "2018-09-01, 19:30:21").to(g.V("Artificial Intelligence"))

Gremlin 语句查询:

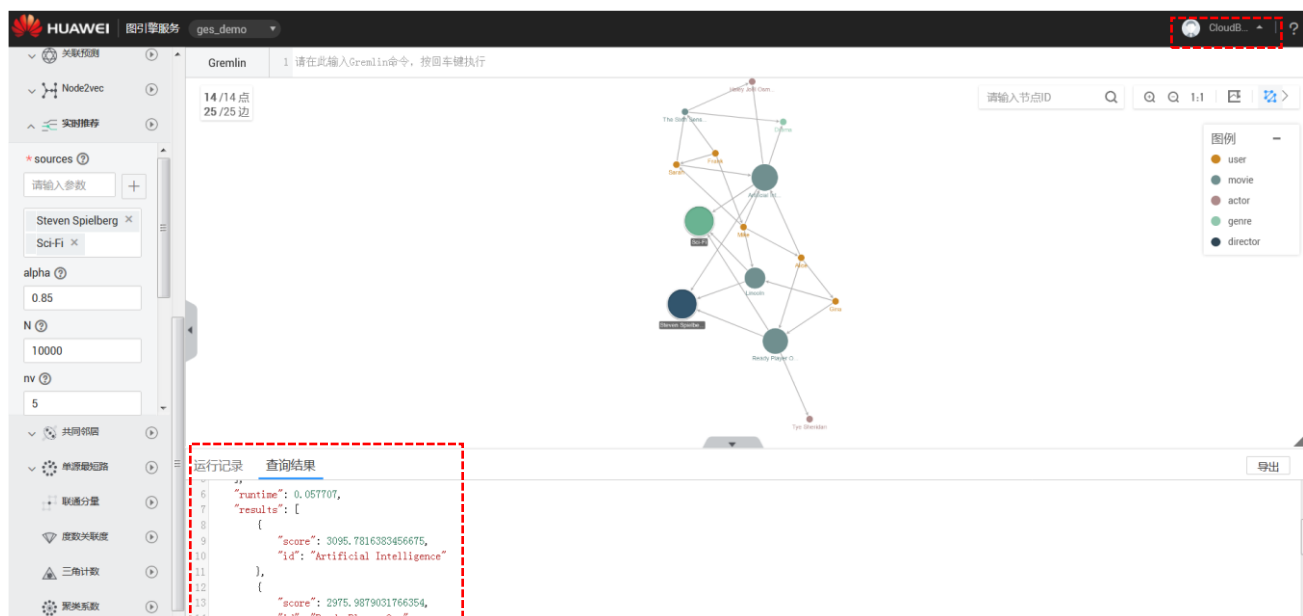
查询用户 Frank：g.V("Frank")；

查询 Frank 喜欢的电影：g.V("Frank").out("like").hasLabel("movie")；

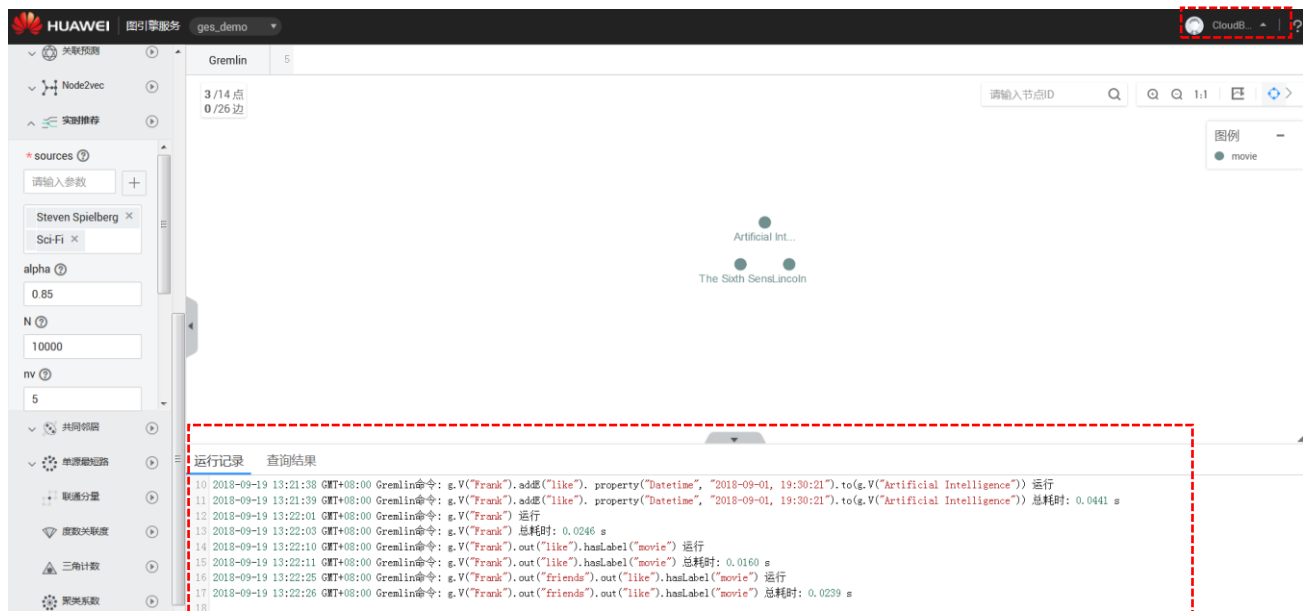
查询 Frank 朋友喜欢的电影：g.V("Frank").out("friends").out("like").hasLabel("movie")；

分别在（1）和（2）结束时，进行截图反馈打卡，示例如下(注意红色方框中的内容)：

题（1）示例:

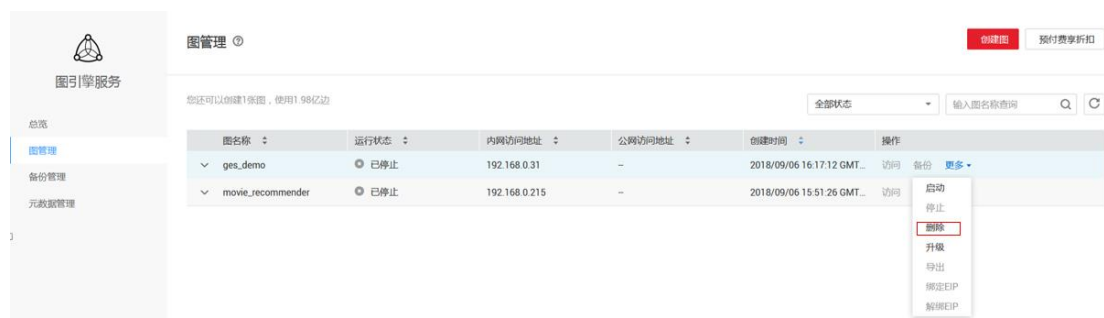


题 (2) 示例:



注意：完成打卡后，请及时到控制台删除闲置的图，以免造成欠费

删除图方法：“[图管理](#)” → 点击“更多” → 删除



附录：

GES 服务用户指南

https://support.huaweicloud.com/usermanual-ges/ges_01_0001.html#