



2017第八届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2017

gStore——一种开源图数据库系统及其在企业中的应用

彭 鹏

湖南大学信息科学与工程学院

hnu16pp@hnu.edu.cn

提纲

- 知识图谱概述
- RDF数据管理系统——gStore
- 系统应用
- 总结

提纲

- 知识图谱概述
- RDF数据管理系统——gStore
- 系统应用
- 总结

知识图谱 (Knowledge Graph)

2012年5月16日，Google发布“知识图谱”的新一代“智能”搜索功能。

The screenshot shows a Google search result for '北京大学' (Peking University). The search bar contains the text '北京大学'. Below the search bar, there are tabs for '全部', '地图', '图片', '新闻', '视频', '更多', and '搜索工具'. The search results show approximately 5,690,000 results found in 0.53 seconds. The main result is for '北京大学' (Peking University) with the URL 'www.pku.edu.cn/'. A snippet of text describes a 2016 visit by Li Keqiang, Premier of the State Council, to Peking University and Tsinghua University. To the right of the main result is a knowledge graph for '北京大学' (Peking University), which includes the university's logo, a map showing its location at '北四环西路', and a list of related information such as '北京大学校内信息门户', '北京大学图书馆', '北大未名BBS', and '院系设置'. Below the knowledge graph, there are links to '北京大学- 维基百科, 自由的百科全书' and '北京大学_百度百科'.

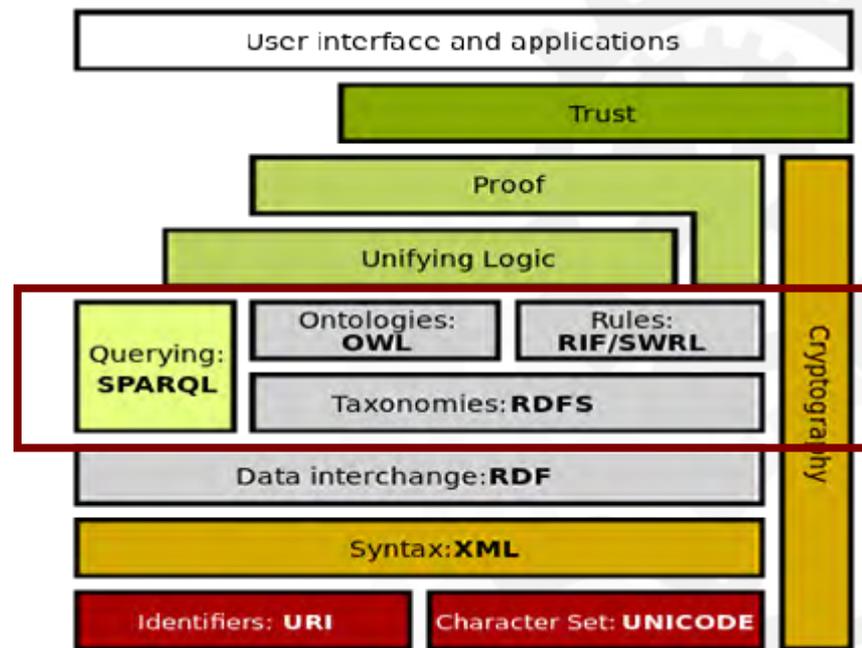
知识图谱 (Knowledge Graph)

本质上是基于图的语义网络，表示实体和实体之间的关系！



资源描述框架(RDF)数据

- RDF是知识图谱数据的事实标准
- RDF是由W3C组织提出的一种描述资源概念模型的语言
- RDF是语义网的一个基石
(Building Block)
- 语义网的目标是网络上的资源是“机器可理解”(Machine understandable)



Google发布
知识图谱

2012-05-16

搜狗发布
“知立方”

2012-11-12

Facebook
社交知识图谱的
图搜索功能上线

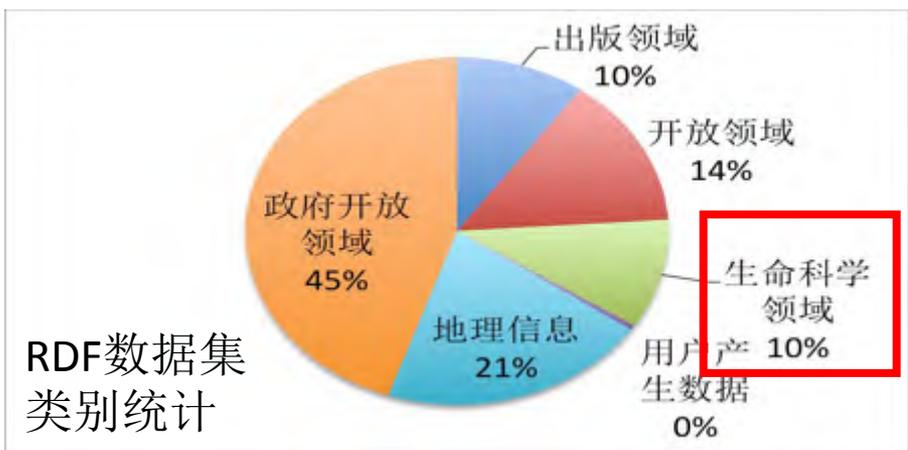
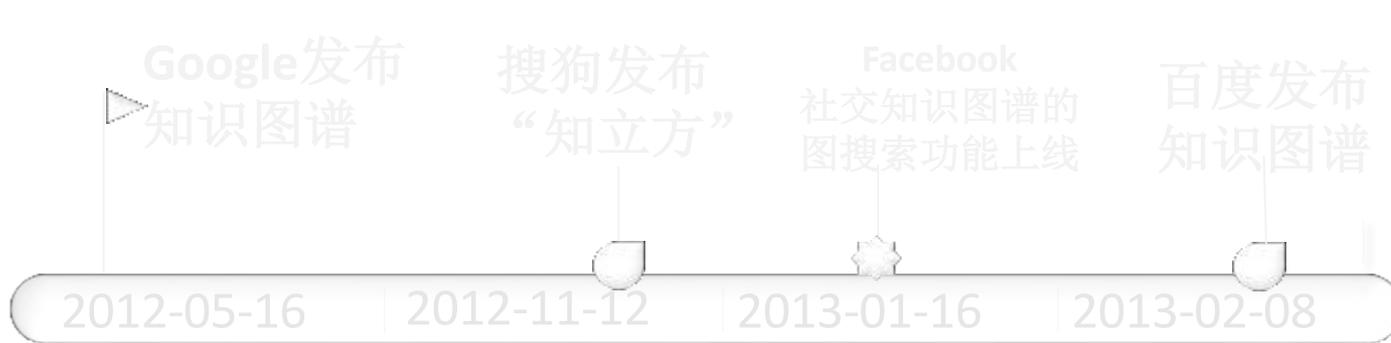
2013-01-16

百度发布
知识图谱

2013-02-08

工业应用





BioModels, Biosamples, ChEMBL, Ensembl, Atlas, Reactome and UniProt

语义网？-----一个简单例子(RDFa)

传统的HTML只是考虑网页的显示，例如字体、段落格式等；而不是网页中的内容的语义。

```
<html>
  <font size="3" color="red"> Lei Zou </font>
  <br>
  Email:<a href= "mailto: zoulei@pku.edu.cn">
zoulei@pku.edu.cn </a>
  <p>
    <font size="3" color="black">Publications: </font>
  </p>
  <div>
    Lei Zou, Jinhui Mo, Lei Chen, M. Tamer Ozs,
    Dongyan Zhao, gStore: Answering SPARQL Queries Via
    Subgraph Matching, VLDB, 2011
  </div>
</html>
```

语义网？-----一个简单例子(RDFa)

语义网考虑的是内容的语义。

```
<html>
<div resource="#me" typeof="Person" >
<font size="3" color="red"> <span property= http://xmlns.com/foaf/0.1/name> Lei
Zou </span> </font>
<br/>
<a property=" http://xmlns.com/foaf/0.1/mbox" href= "mailto: zoulei@pku.edu.cn "
> zoulei@pku.edu.cn </a>
<p>
<font size="3" color="black">Publications: </font>
</p>
<div resource="www.vldb.org/pvldb/vol4/p482-zou.pdf">
  <span property=" http://purl.org/dc/terms/contributor"> Lei Zou </span>,
  <span property="http://purl.org/dc/terms/contributor"> Jinghui Mo </span>,
  <span property=" http://purl.org/dc/terms/contributor"> Lei Chen </span>,
  <span property=" http://purl.org/dc/terms/contributor"> M. Tamer Özsu</span>,
  <span property=" http://purl.org/dc/terms/contributor"> Dongyan Zhao</span>,
  <span property=" http://purl.org/dc/terms/title"> gStore: Answering SPARQL
Queries Via Subgraph Matching </span>,
  <span property=" http://purl.org/dc/terms/Publisher"> VLDB </span>
  <span property=" http://purl.org/dc/terms/Date">2011</span>
</div>
</html>
```

语义网? ----- 一个简单例子(RDFa)

Google结构化数据测试工具

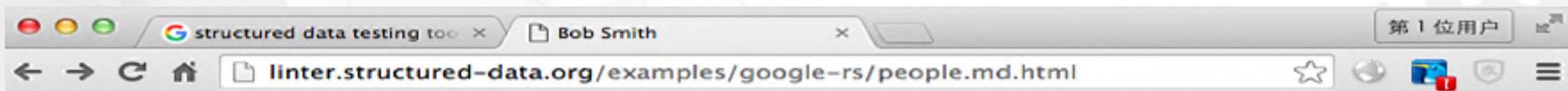
The screenshot shows the Google Structured Data Testing Tool interface. On the left, there is a code editor with the following HTML code:

```
1 <html>
2 <div resource="#me" typeof="Person" >
3 <font size="3" color="red" > <span property=
4 http://xmlns.com/foaf/0.1/name> Lei
5 Zou </span> </font>
6 <br/>
7 <a property=" http://xmlns.com/foaf/0.1/mbox" href= "mailto:
8 zoulei@pku.edu.cn "
9 > zoulei@pku.edu.cn </a>
10 <p>
11 <font size="3" color="black">Publications: </font>
12 </p>
13 <div resource="www.vldb.org/pvldb/vol4/p482-zou.pdf">
14 <span property=" http://purl.org/dc/terms/contributor"> Lei Zou
15 </span>,
16 <span property="http://purl.org/dc/terms/contributor"> Jinghui Mo
17 </span>,
18 <span property=" http://purl.org/dc/terms/contributor"> Lei Chen
19 </span>,
20 <span property=" http://purl.org/dc/terms/contributor"> M. Tamer
21 Özsu</span>,
22 <span property=" http://purl.org/dc/terms/contributor"> Dongyan
23 Zhao</span>,
24 <span property=" http://purl.org/dc/terms/title"> gStore: Answering
25 SPARQL
```

On the right, the tool displays a table of extracted data:

@type	未指定类型
未指定类型	
ID: http://www.vldb.org/pvldb/vol4/p482-zou.pdf	
@type	未指定类型
@id	http://www.vldb.org/pvldb/vol4/p482-zou.pdf
http://purl.org/dc/terms/contributor	Lei Zou
http://purl.org/dc/terms/contributor	Jinghui Mo
http://purl.org/dc/terms/contributor	Lei Chen
http://purl.org/dc/terms/contributor	M. Tamer Özsu
http://purl.org/dc/terms/contributor	Dongyan Zhao
http://purl.org/dc/terms/title	gStore: Answering SPARQL Queries Via Subgraph Matching
http://purl.org/dc/terms/Publisher	VLDB
http://purl.org/dc/terms/Date	2011

语义网？-----一个简单例子(RDFa)



My name is Bob Smith, but people call me Smithy. Here is my homepage: www.example.com. I live in Albuquerque, NM and work as an engineer at ACME Corp. My friends: [Darryl](http://darryl-blog.example.com), [Edna](http://edna-blog.example.com)

RDFa (source lint)

```
<div xmlns:v="http://rdf.data-vocabulary.org/#" typeof="v:Person">
  My name is <span property="v:name">Bob Smith</span>,
  but people call me <span property="v:nickname">Smithy</span>.
  Here is my homepage:
  <a href="http://www.example.com" rel="v:url">www.example.com</a>.
  I live in
  <span rel="v:address">
    <span typeof="v:Address">
      <span property="v:locality">Albuquerque</span>,
      <span property="v:region">NM</span>
    </span>
  </span>
  and work as an <span property="v:title">engineer</span>
  at <span property="v:affiliation">ACME Corp</span>.
  My friends:
  <a href="http://darryl-blog.example.com" rel="v:friend">Darryl</a>,
  <a href="http://edna-blog.example.com" rel="v:friend">Edna</a>
</div>
```

语义网? ----- 一个简单例子(RDFa)

Enhanced search result preview

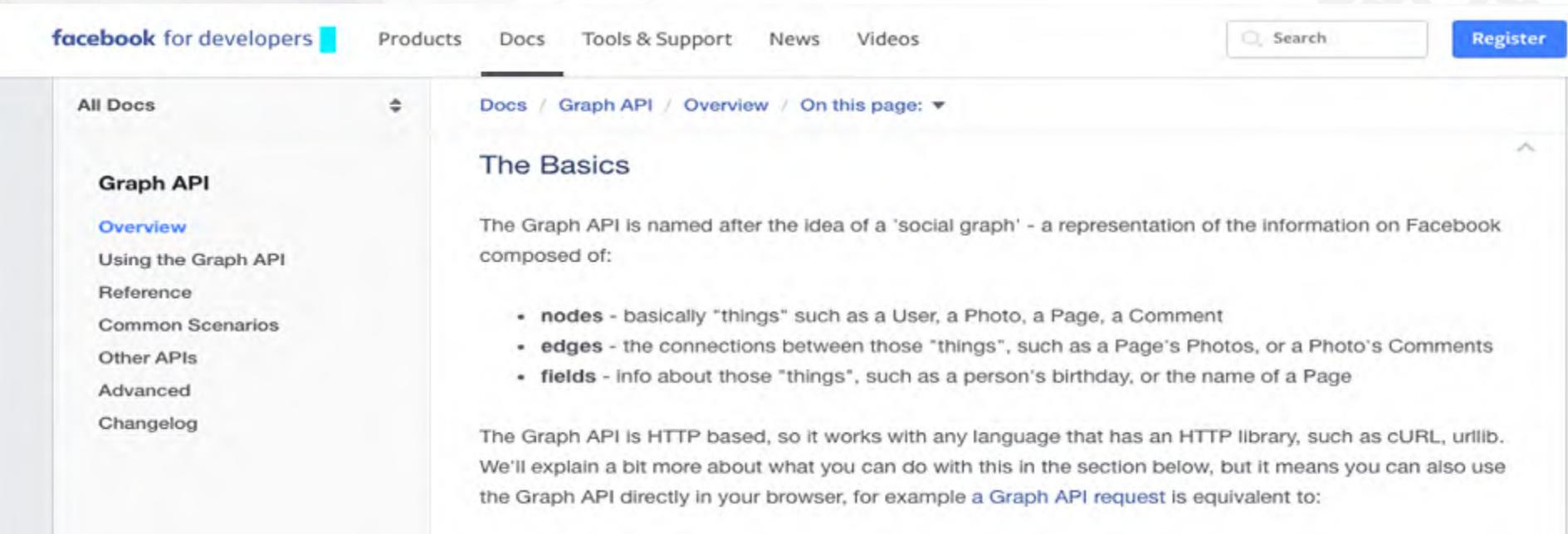
Disclaimer: this preview is only shown as a example of what a search engine **might** display. It is to the discretion of each search engine provider to decide whether your page will be displayed as an enhanced search result or not in their search results pages.

Bob Smith
linter.structured-data.org/examples/google-rs/people.rdfa.html
Albuquerque, NM - engineer, ACME Corp
an **actual** search result **may** display other content **relating** to your search terms here.

Raw structured data extracted from the page:

@id	http://rdf.data-vocabulary.org/#Address(1)						
rdf:type	rdfs:Class						
@id	http://rdf.data-vocabulary.org/#Person(1)						
rdf:type	rdfs:Class						
rdf:type	v:Person rdfs:Resource						
v:address	<table border="1"><tr><td>rdf:type</td><td>v:Address rdfs:Resource</td></tr><tr><td>v:locality</td><td>Albuquerque</td></tr><tr><td>v:region</td><td>NM</td></tr></table>	rdf:type	v:Address rdfs:Resource	v:locality	Albuquerque	v:region	NM
rdf:type	v:Address rdfs:Resource						
v:locality	Albuquerque						
v:region	NM						
v:affiliation	ACME Corp						
v:friend	<ul style="list-style-type: none">• http://edna-blog.example.com• http://darryl-blog.example.com						
v:name	Bob Smith						
v:nickname	Smithy						

Facebook Social Graph



The screenshot shows the 'facebook for developers' website. The navigation bar includes 'Products', 'Docs', 'Tools & Support', 'News', and 'Videos'. A search bar and a 'Register' button are also present. The left sidebar lists 'All Docs' with a dropdown arrow, and under 'Graph API', it lists 'Overview', 'Using the Graph API', 'Reference', 'Common Scenarios', 'Other APIs', 'Advanced', and 'Changelog'. The main content area shows the breadcrumb 'Docs / Graph API / Overview / On this page: ▼' and the title 'The Basics'. The text explains that the Graph API is named after the idea of a 'social graph' - a representation of the information on Facebook composed of:

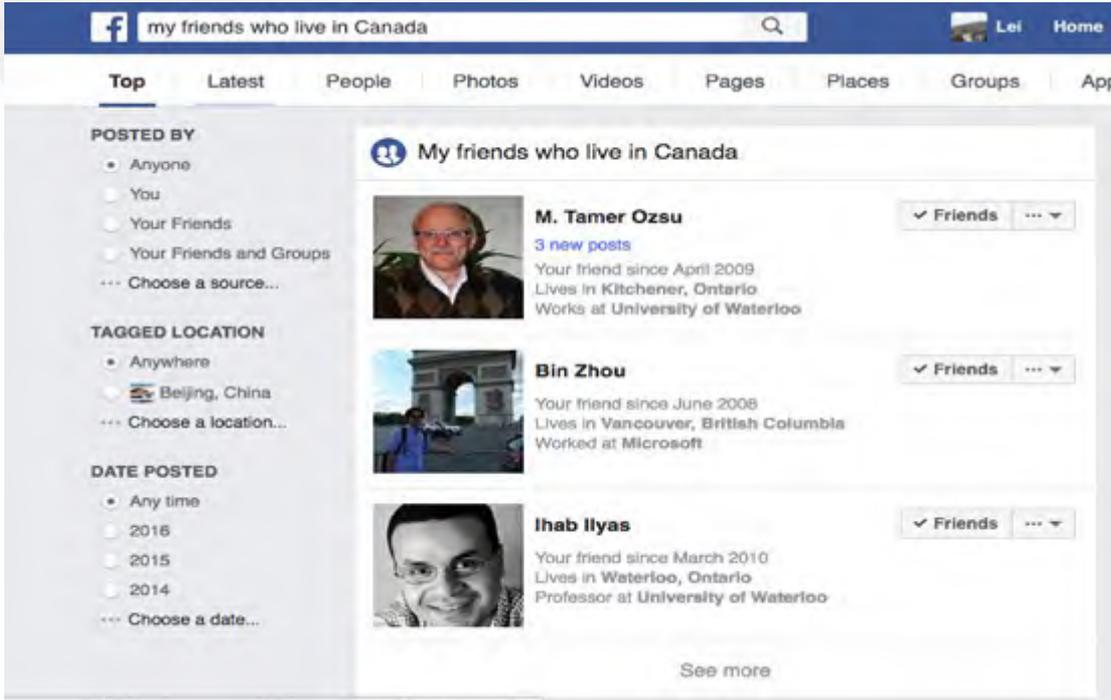
- **nodes** - basically "things" such as a User, a Photo, a Page, a Comment
- **edges** - the connections between those "things", such as a Page's Photos, or a Photo's Comments
- **fields** - info about those "things", such as a person's birthday, or the name of a Page

The text continues: 'The Graph API is HTTP based, so it works with any language that has an HTTP library, such as cURL, urllib. We'll explain a bit more about what you can do with this in the section below, but it means you can also use the Graph API directly in your browser, for example a [Graph API request](#) is equivalent to:'

Facebook Graph Search

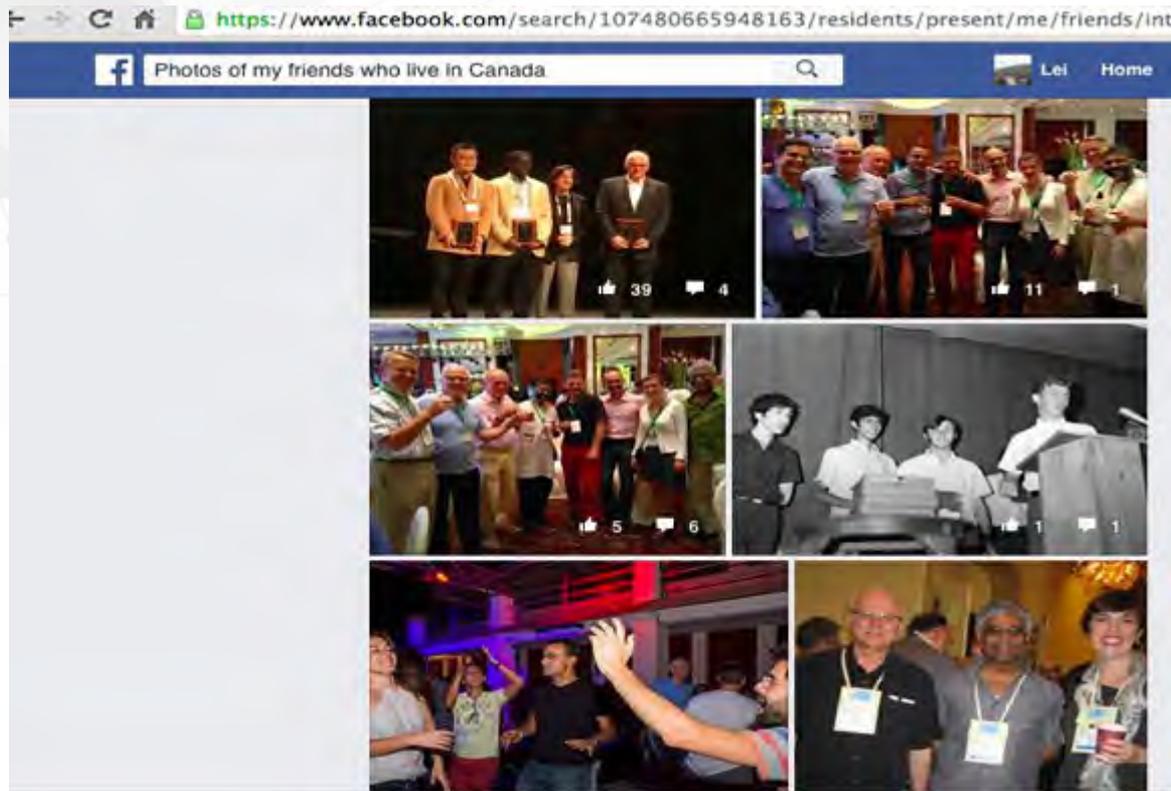
2013年1月16日 Facebook Graph Search
产品发布会---Mark Zuckerberg

“My friends who live in Canada”

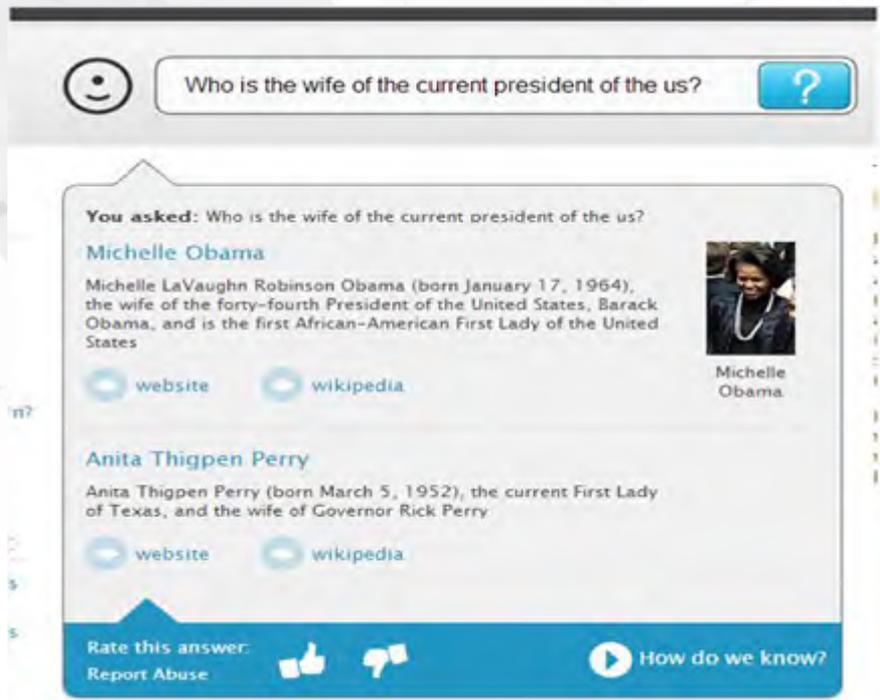


Facebook Graph Search

“Photos of my friends who live in Canada”



EVI---原名True Knowledge



年度	获得风投
2007-09	120万 USD
2008-07	400万 USD
2012-01	被亚马逊收购

William Tunstall-Pedoe: *True Knowledge: Open-Domain Question Answering using Structured Knowledge and Inference*. AI Magazine 31(3): 80-92 (2010)

RDF 数据模型

- RDF中任何实体都被称之为资源 (Resource)，用URI来表示。
- 实体的属性需要被定义
- 实体间关系需要被定义
- 不同数据集直接互相链接构成海量的关联数据
 - 一个集成的Web” 数据库”

xmlns:y=http://en.wikipedia.org/wiki
y:Abraham Lincoln



Abraham Lincoln:hasName "Abraham Lincoln"
Abraham Lincoln:BornOnDate: "1809-02-12"
Abraham Lincoln:DiedOnDate: "1865-04-15"

DiedIn



y:Washington_DC

RDF 数据 & SPARQL查询语言

RDF 数据库

主语	谓词	宾语
Abraham_Lincoln	hasName	"Abraham Lincoln"
Abraham_Lincoln	BornOnDate	"1809-02-12"
Abraham_Lincoln	DiedOnDate	"1865-04-15"
Abraham_Lincoln	DiedIn	Washington_DC
Abraham_Lincoln	bornIn	Hodgenville KY
Reese-Witherspoon	bornOnDate	"1976-03-22"
Reese-Witherspoon	bornIn	New_Orleans_LA
New_Orleans_LA	foundingYear	"1718"
New Orleans LA	locatedIn	United_States
United_States	hasName	"United States"
United_States	hasCapital	Washington_DC
United_States	foundingYear	"1776"

“找到出生1976年生的，并且出生地是1718年构建的城市的人有哪些？”

```
SELECT ?name SPARQL
WHERE {
  ?m <bornIn> ?city .
  ?m <hasName> ?name .
  ?m <bornOnDate> ?bd .
  ?city <foundingYear> ``1718`` .
  FILTER( regex (str (?bd ), "1976" ) )
}
```

提纲

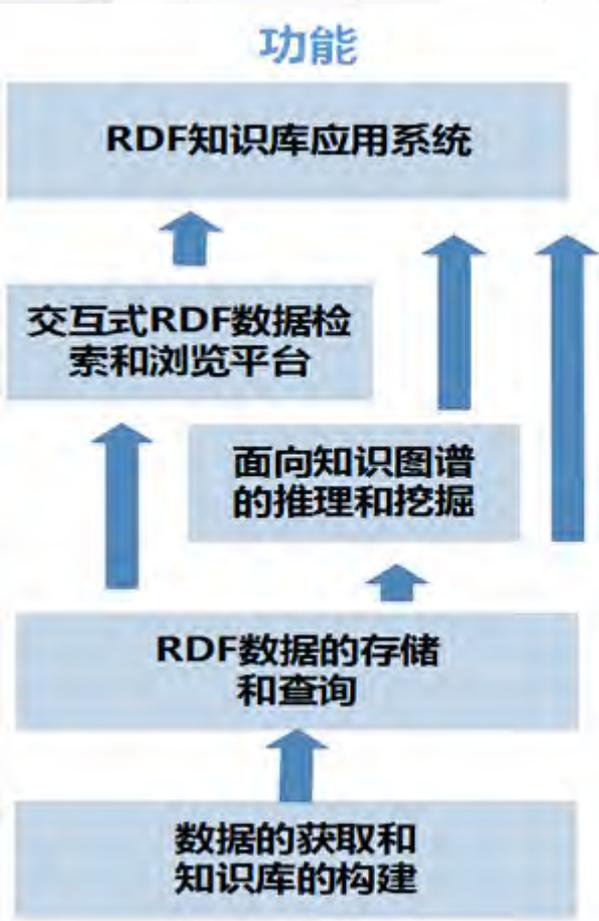
- 知识图谱概述
- **RDF数据管理系统——gStore**
- 系统应用
- 总结

RDF数据管理问题

应用

系统

数据



涉及的技术

互联网的开放域知识图谱和特定应用领域的应用

自然语言问题理解、图数据的可视化、CHI界面的设计

基于规则的推理；大规模并行推理系统；异质信息网络的挖掘

基于关系数据库的RDF引擎；图数据库技术；分布式RDF数据管理

实体、关系的抽取；Deep Web数据；数据集知识融合；知识图谱数据质量控制

RDF数据管理问题

应用

系统

数据



一个基本的问题：如何存储RDF数据和回答SPARQL查询

主语	谓词	宾语
Abraham_Lincoln	hasName	"Abraham Lincoln"
Abraham_Lincoln	BornOnDate	"1809-02-12"
Abraham_Lincoln	DiedOnDate	"1865-04-15"
Abraham_Lincoln	DiedIn	Washington_DC
Abraham_Lincoln	bornIn	Hodgenville KY
Reese_Witherspoon	bornOnDate	"1976-03-22"
Reese_Witherspoon	bornIn	New_Orleans_LA
New_Orleans_LA	foundingYear	"1718"
New Orleans LA	locatedIn	United_States
United_States	hasName	"United States"
United_States	hasCapital	Washington_DC
United_States	foundingYear	"1776"

```
SPARQL  
SELECT ?name  
WHERE {  
  ?m <bornIn> ?city .  
  ?m <hasName> ?name .  
  ?m <bornOnDate> ?bd .  
  ?city <foundingYear> `1718` .  
  FILTER( regex (str (?bd ), "1976" ) )  
}
```

怎样快速回答SPARQL?

现有方法：求助于关系数据库技术

主语	谓词	宾语
Abraham_Lincoln	hasName	"Abraham Lincoln"
Abraham_Lincoln	BornOnDate	"1809-02-12"
Abraham_Lincoln	DiedOnDate	"1865-04-15"
Abraham_Lincoln	DiedIn	Washington_DC
Abraham_Lincoln	bornIn	Hodgenville KY
Reese_Witherspoon	bornOnDate	"1976-03-22"
Reese_Witherspoon	bornIn	New_Orleans_LA
New_Orleans_LA	foundingYear	"1718"
New Orleans LA	locatedIn	United_States
United_States	hasName	"United States"
United_States	hasCapital	Washington_DC
United_States	foundingYear	"1776"

```

SELECT ?name
WHERE {
  ?m <bornIn> ?city .
  ?m <hasName> ?name .
  ?m <bornOnDate> ?bd .
  ?city <foundingYear> ``1718`` .
  FILTER( regex (str (?bd ), "1976" ) )
}
    
```

SPARQL



```

SELECT T2.object
FROM T
T as T4
WHERE T1.property="bornIn "
AND T2.property="hasName"
AND T3.property="bornOnDate "
AND T1.subject=T2.subject
AND T2.subject=T3.subject
AND T1.object=T4.subject
AND T4.propety="foundingYear "
AND T4.object=" 1718 "
AND T3.object LIKE '%1976%'
    
```

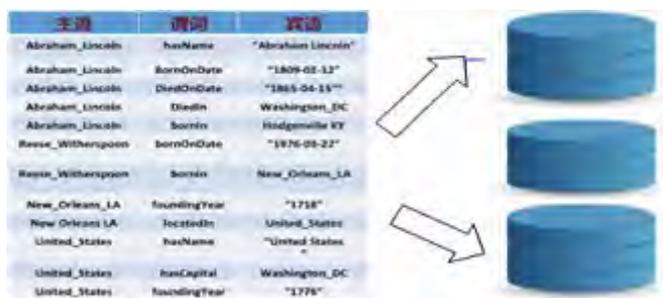
SQL



三种典型基于关系数据库的优化策略

- 属性表方法 Jena [Wilkinson et al., 2003], FlexTable [Wang et al., 2010], DB2-RDF [Bornea et al., 2013]
- 垂直划分方法 SW-store [Abadi et al., 2009]
- 全索引方法 RDF-3X [Neumann and Weikum, 2008], Hexastore [Weiss et al., 2008]

基本思路： 划分三元组表、生成更加简单的查询。



- 邹磊, 陈跃国, 海量RDF数据管理, 中国计算机学会通讯, 第8卷, 第11期, 2012年11月
- 杜方 陈跃国 杜小勇, RDF数据查询处理技术综述, 软件学报, 2013
- M. T. Özsu. "A Survey of RDF Data Management Systems", Front. Comp. Sci., 2016.

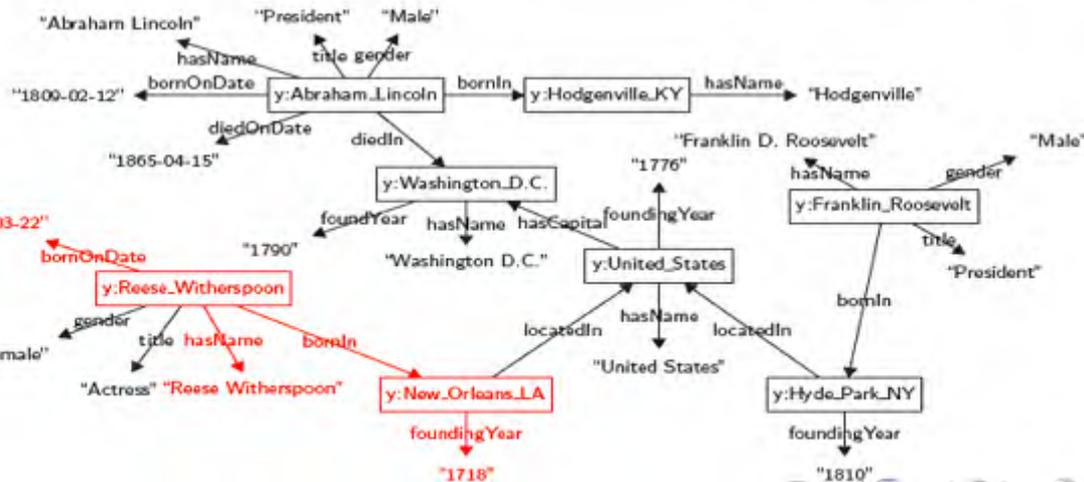
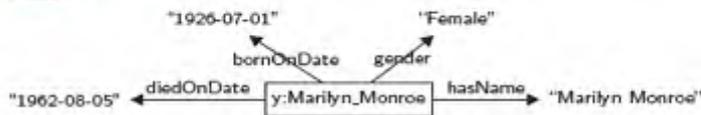
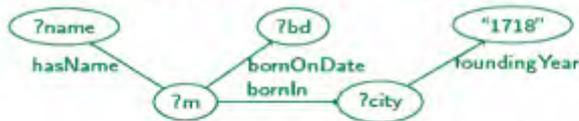


换个角度

我们的方法---gStore [Zou et al.,

WDD 11 WDD T 14 7

FILTER(regex(str(?bd), "1976"))



回答SPARQL查询
== 子图匹配

我们基于图的RDF数据管理研究路线图

gStore-D: 分布式RDF图数据管理系统

gStore: 基于子图匹配的SPARQL查询系统

研究主题

子图模式
匹配查询

建模

基于结构感知的图
数据库索引和子图
匹配查询优化理论

姓名	出生日期	婚姻状况
Wright, George	1867-01-01	Married

```
SELECT ?name  
WHERE {  
  ?c <bornOnDate> ?d  
  ?c <bornName> ?name  
  ?c <bornOnDate> ?d  
  ?c <?y foundingYear> "1718"  
  FILTER regex(str(?d), "1976")  
}
```

SPARQL

自然语言问题

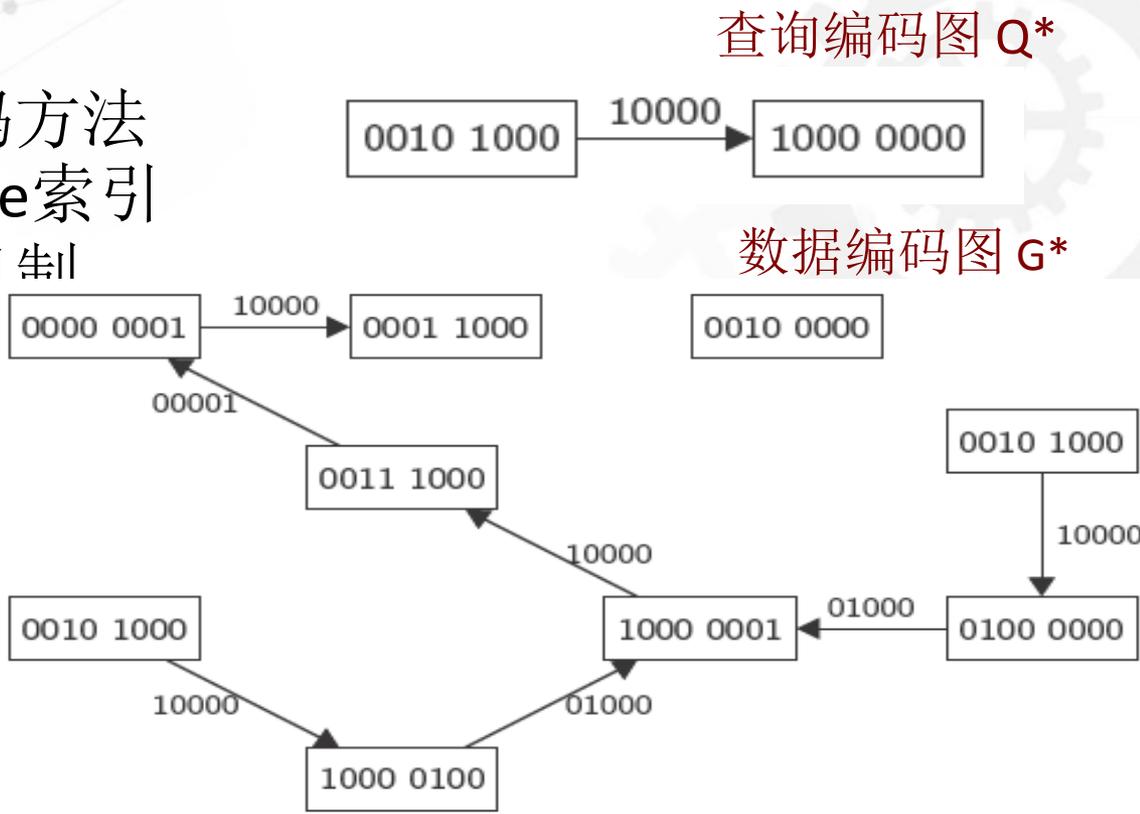
“结婚的人的著作有哪些？”
“Who was married to an actor that
play in Philadelphia?”

RDF数据

gStore---一种基于图的RDF存储和查询系统

主要技术手段:

- 统一的结构和内容编码方法
- 一种高度平衡树 VS-tree索引
- 基于索引的多级过滤机制



找到 Q^* 在 G^* 上的匹配



测试每个这样的匹配

构建的系统



代码：除了SPARQL语法解析器外均为独立开发的，目前有14万行C++，完成自主知识产权；共计6人年，目前是版本v 0.3.0 。

开源地址：<https://github.com/Caesar11/gStore/>

包括全部的**系统代码**；详细的**用户手册**；与目前最好的开源和工业系统在多个Benchmark数据集上的对比**测试报告**；系统使用**演示视频**。

开源协议：基于BSD 协议

部署方法：单机和C/S方式部署

接口：C++，Java，python，PHP等API接口；接收标准的RDF文件格式（N3，Turtle等格式）



对比测试

实验环境

- Linux服务器(CentOS 7.2)
- CPU: E5-2640 v3@2.60GHz
- RAM: 128GB RDIMM,2133Mt/s
- gStore版本: 0.4.0
- 对比系统: Apache-Jena 3.0.1, Virtuoso-openlinksw 7.2
- 数据集: WatDiv, LUBM, BSBM, DBpedia



对比系统	系统性质	所属单位
Virtuoso	目前最好的、使用最广泛的 商业 RDF数据库系统	Openlink公司 (美国伯灵顿、1992成立)
Apache Jena	目前最有影响力的 开源 RDF数据库项目	来源于HP研究院 (HP Lab) 2000年开始的项目; 2010年以后成为Apache开源项目

对比测试

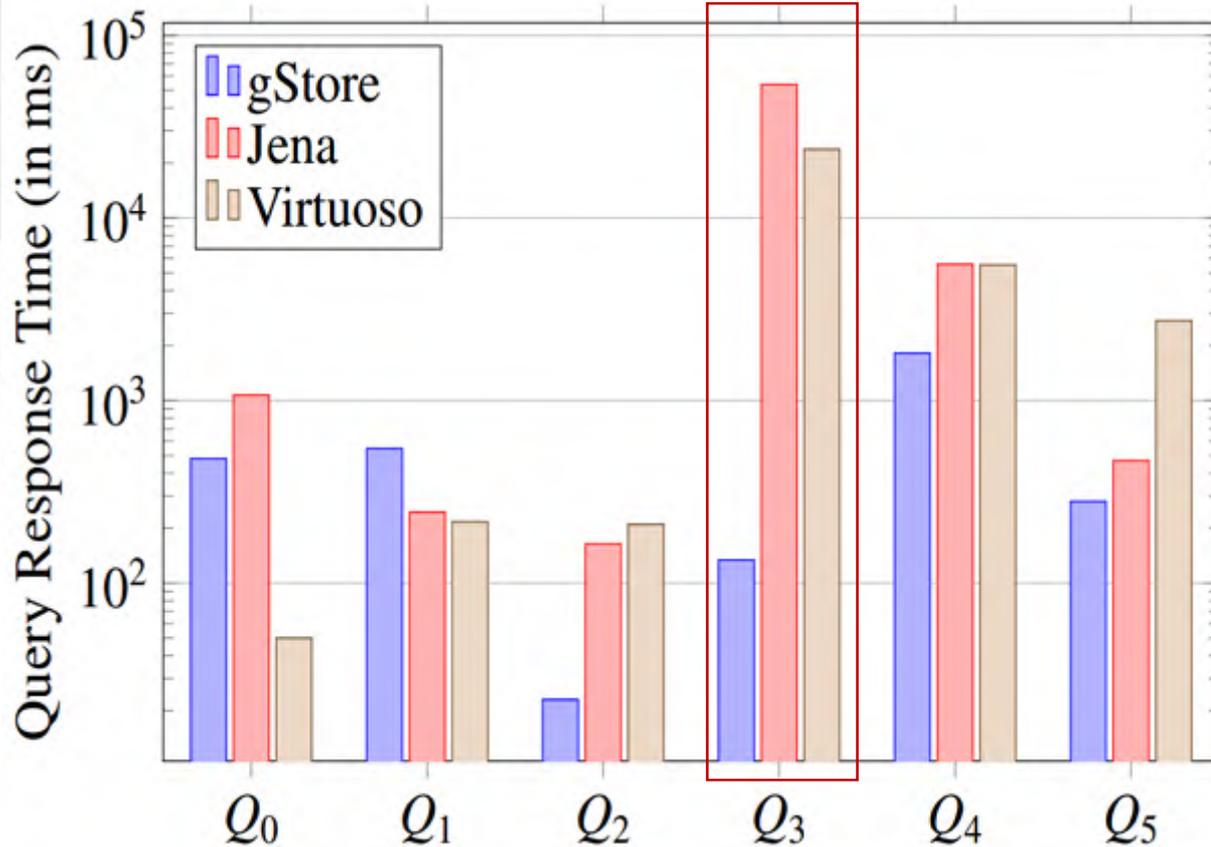
数据集规模

Dataset	Size	Triple	Predicat e	Entity	Literal
Dbpedia 170M	23GB	170,784,508	57,354	7,123,915	14,971,449
WatDiv 300M	47GB	329,584,783	86	15,636,745	14,748,846
LUBM 500M	85GB	500,000,000	18	81,342,489	41,804,418
BSBM 300M	82GB	311,957,992	40	46,514,164	25,176,573

对比测试

gStore 0.4.0: 2017年1月

Query Performance over DBpedia 170M

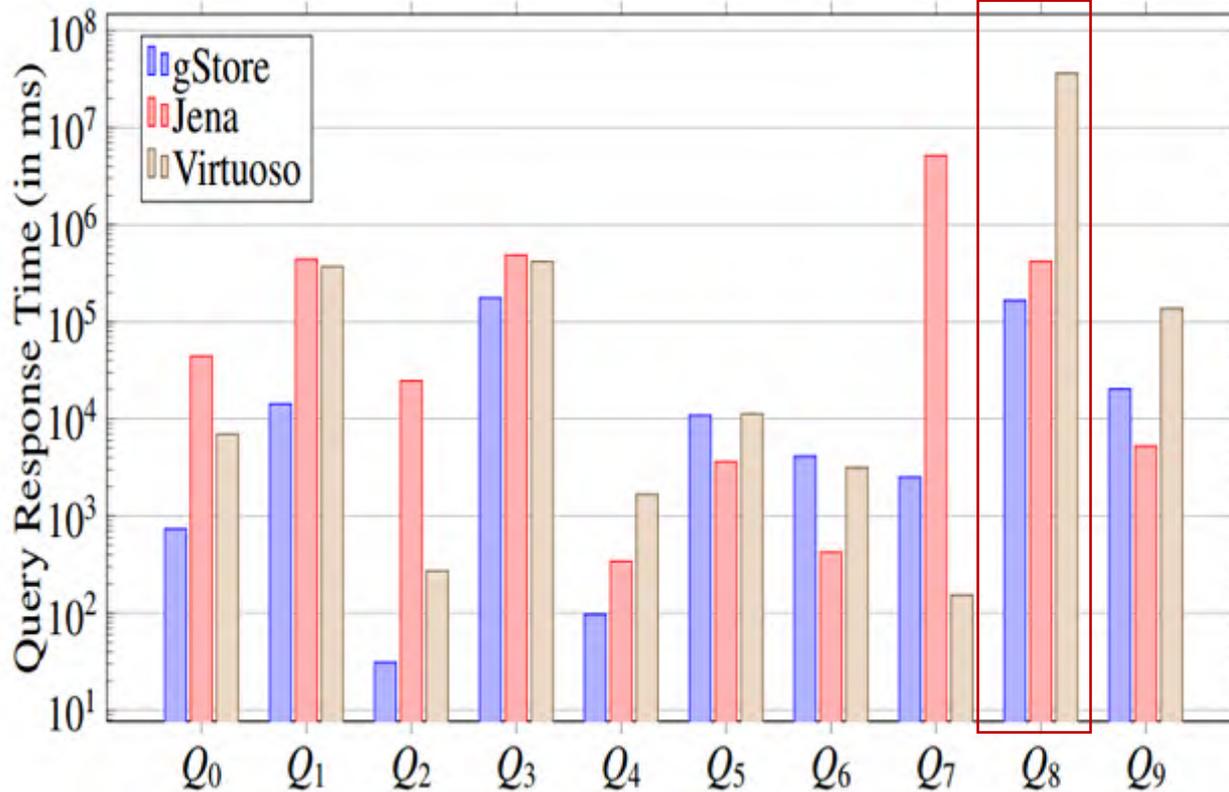


查询	三元组数	返回列数
Q0-Q2	≤3	1
Q3-Q5	1-3	>1

对比测试

gStore 0.4.0: 2017年1月

Query Performance over BSBM 300M

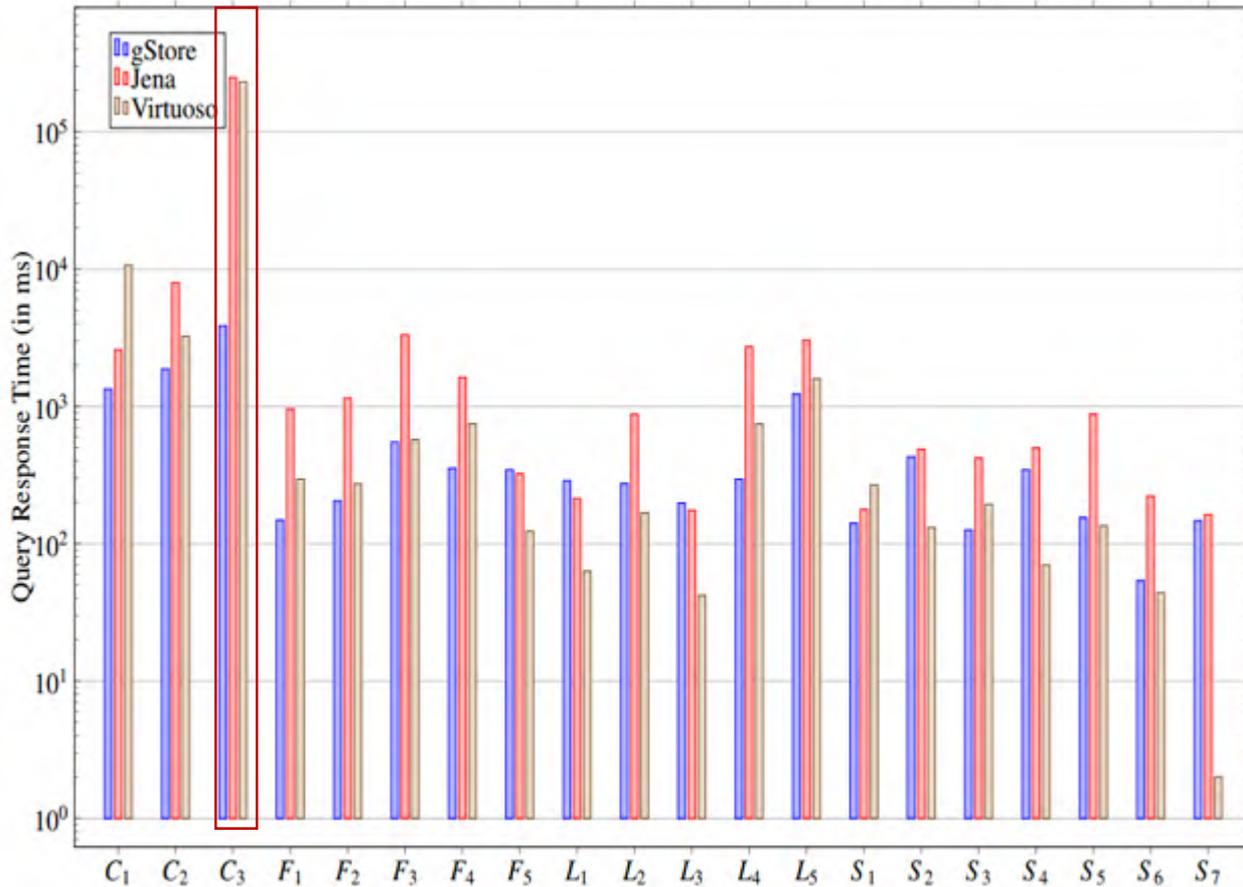


查询	三元组数	返回列数
Q0-Q2	≤2	≤2
Q3-Q5	2-4	3-4
Q6-Q8	5-6	5-7
Q9	11	9

对比测试

gStore 0.4.0: 2017年1月

Query Performance over WatDiv 300M

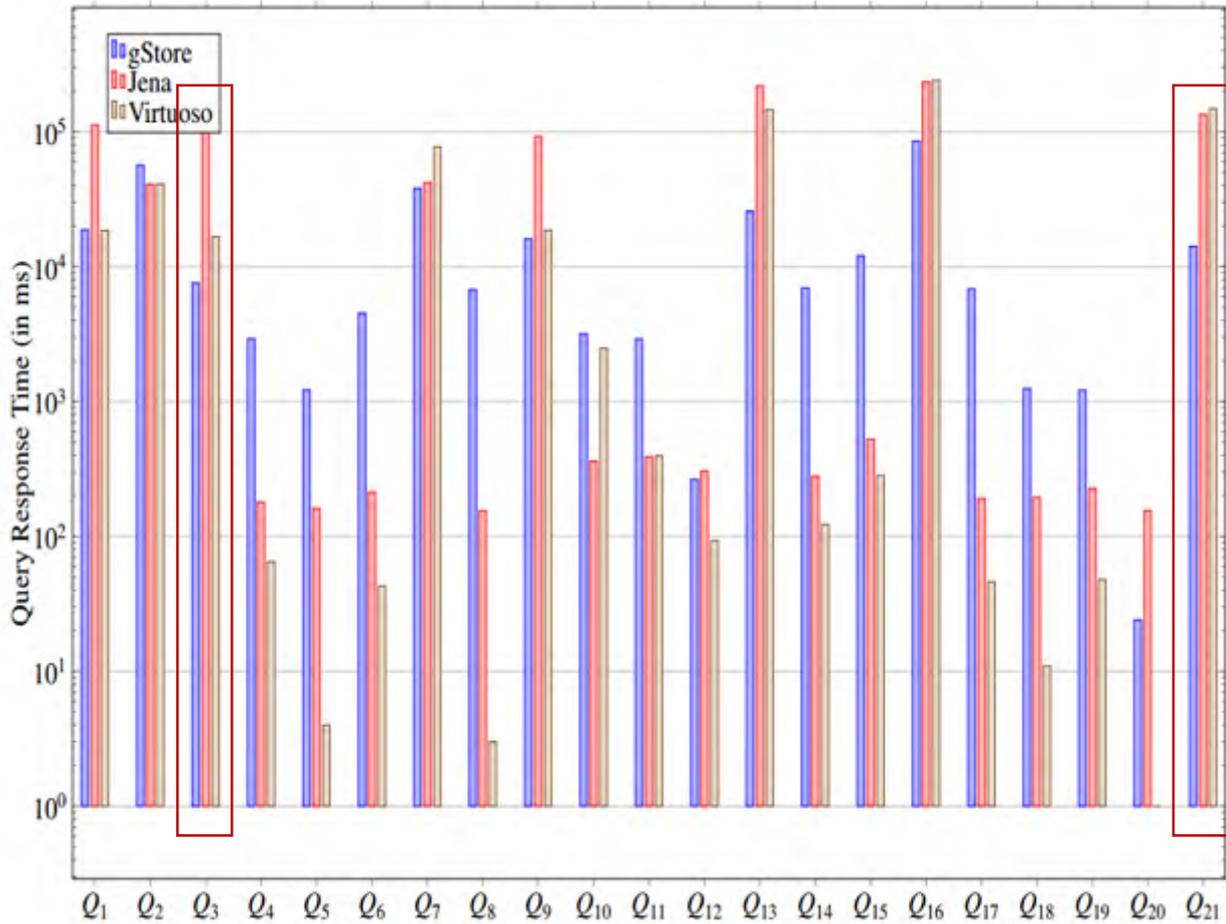


查询	三元组数	返回列数
C1,C2	8,10	4
C3	6	1
F1-F3	6-8	5-7
F4	9	8
F5	6	6
L1-L5	≤3	≤3
S1	9	9
S2-S7	3-4	3-4

对比测试

gStore 0.4.0: 2017年1月

Query Performance over LUBM 500M



查询	三元组数	返回列数
Q1,Q2,Q3	6,2,6	1
Q4	5	4
Q5	2	1
Q6	4	2
Q7	6	3
Q8	2	1
Q9	6	3
Q10	2	1
Q11	5	3
Q12,Q13	1	1
Q14	4	2
Q15,Q16	5	1,3
Q17-Q21	≤3	≤2



性能比较（国外同行的第三方测试）

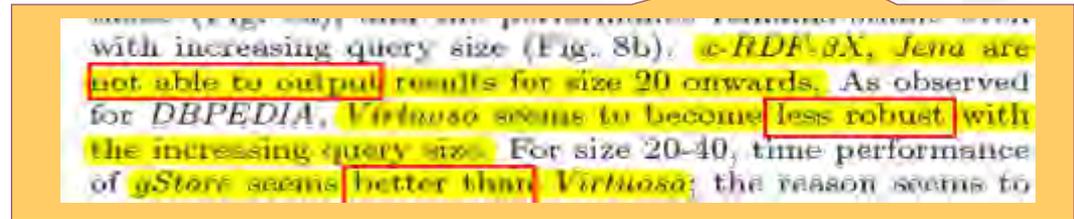
【Vijay Ingalalli, Dino Ienco, Pascal Poncelet, Serena Villata: Querying RDF Data Using A Multigraph-based Approach. EDBT 2016: 245-256】

- 法国LIRMM实验室（隶属于蒙彼利埃大学和法国国家科学研究中心CNRS）
- I3S实验室 CNRS(法国国家科学研究中心)

DBpedia 2014		
1.7亿三元组	7百万实体	23.8 GB 文件大小

对比系统	系统性质	同行实验总结
Apache Jena	目前最有影响力的 开源 RDF数据库系统	“当查询图大小超过20以后， <u>系统就不能输出查询结果了</u> ”
x-RDF-3x	影响力较大的 学术界 系统	
Virtuoso	目前最好的、使用最广泛的 商业 RDF数据库系统	“对于查询图大小增长，系统性能的 <u>鲁棒性较差</u> ”
gStore (我们的系统)	Github上开源 【Zou et al., VLDB 2011】	“gStore性能要好于其他对比系统”

gStore	Virtuoso	RDF-3x
11.96 (秒)	20.45 (秒)	>60 (秒)



对20个针对Dbpedia数据集的
查询的平均查询时间 ---
摘自 Ingalalli et. EDBT 16

gStore---API 接口 (C++; Java; python,

PHP

```
// initialize the Gstore server's IP address and port.
GstoreConnector gc("127.0.0.1", 3305);

// build a new database by a RDF file.
// note that the relative path is related to gserver.
gc.build("LUBM10.db", "example/LUBM_10.n3");

// then you can execute SPARQL query on this database.
std::string sparql = "select ?x where \
{ \
  ?x <rdf:type> <ub:UndergraduateStudent>. \
  ?y <ub:name> <Course1>. \
  ?x <ub:takesCourse> ?y. \
  ?z <ub:teacherOf> ?y. \
  ?z <ub:name> <FullProfessor1>. \
  ?z <ub:worksFor> ?w. \
  ?w <ub:name> <Department0>. \
}";

std::string answer = gc.query(sparql);

// unload this database.
gc.unload("LUBM10.db");

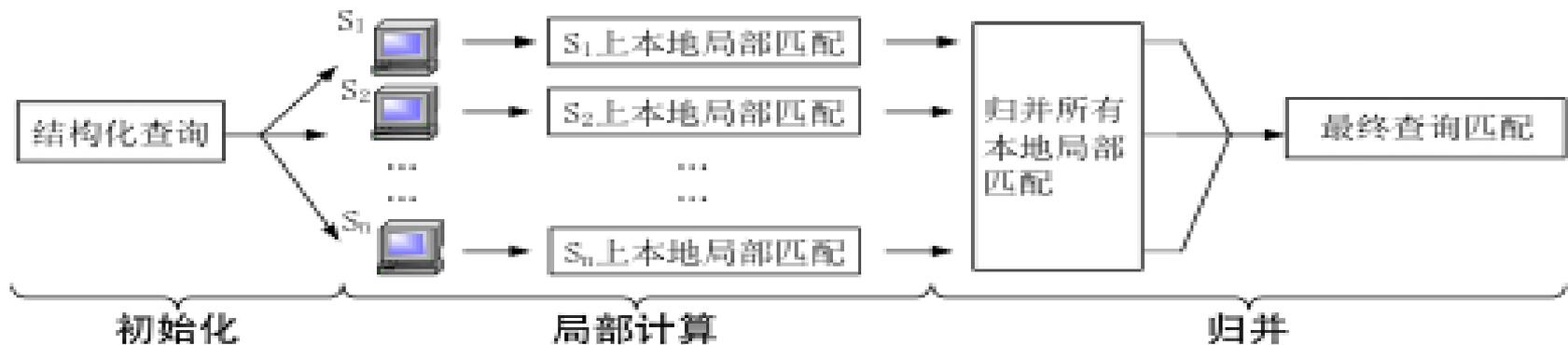
// also, you can load some exist database directly and then query.
gc.load("LUBM10.db");

// query a SPARQL in current database
answer = gc.query(sparql);
```

gStore-D: 分布式系统 [Peng; Zou, et al., VLDB J 16]

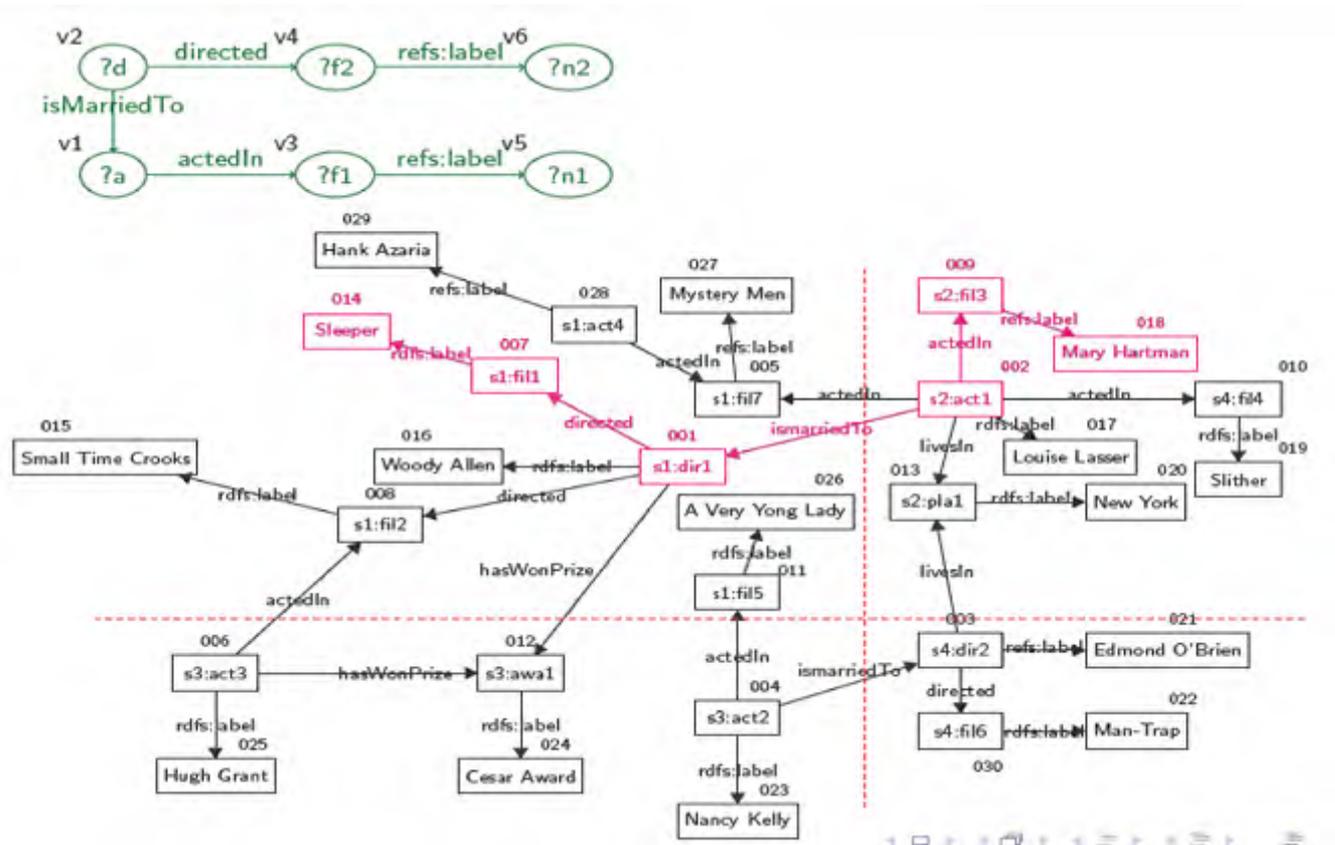
主要技术手段:

- 利用Partial Evaluation and Assembly方案来解决分布式SPARQL匹配;
- 分布式环境下的优化归并策略



gStore-D: 分布式系统

主要技术问题：如何找到“跨界匹配”



gStore-D: 分布式系统

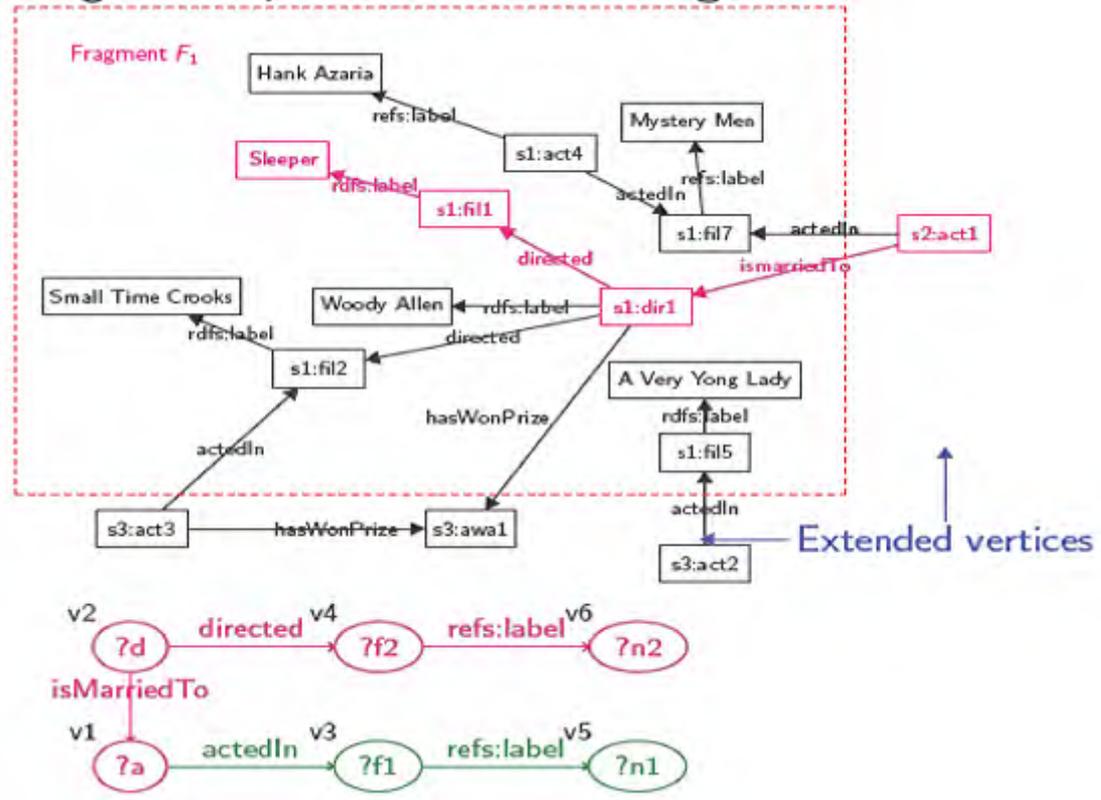
背景：部分执行 (Partial Evaluation) [Jones, 1996; Fan et al., 06; Shuai et al., 2012]

$f(x) \Rightarrow f(s, d) \Rightarrow f''(f'(s), d) \Rightarrow$ 最终结果



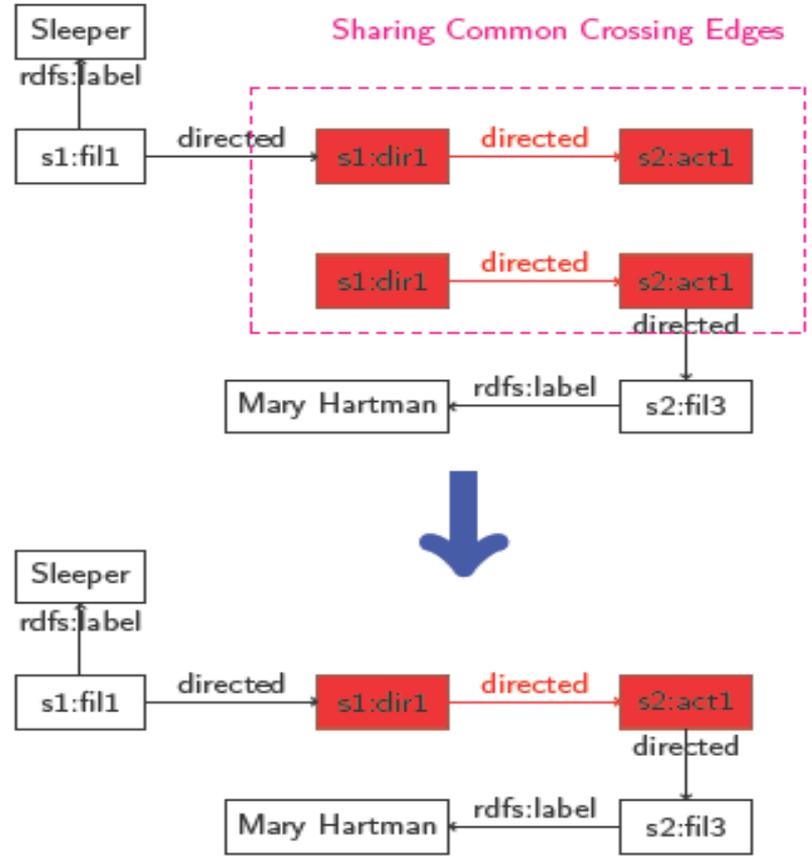
gStore-D: 分布式系统

哪些是“已知输入”和“部分解”？



gStore-D: 分布式系统

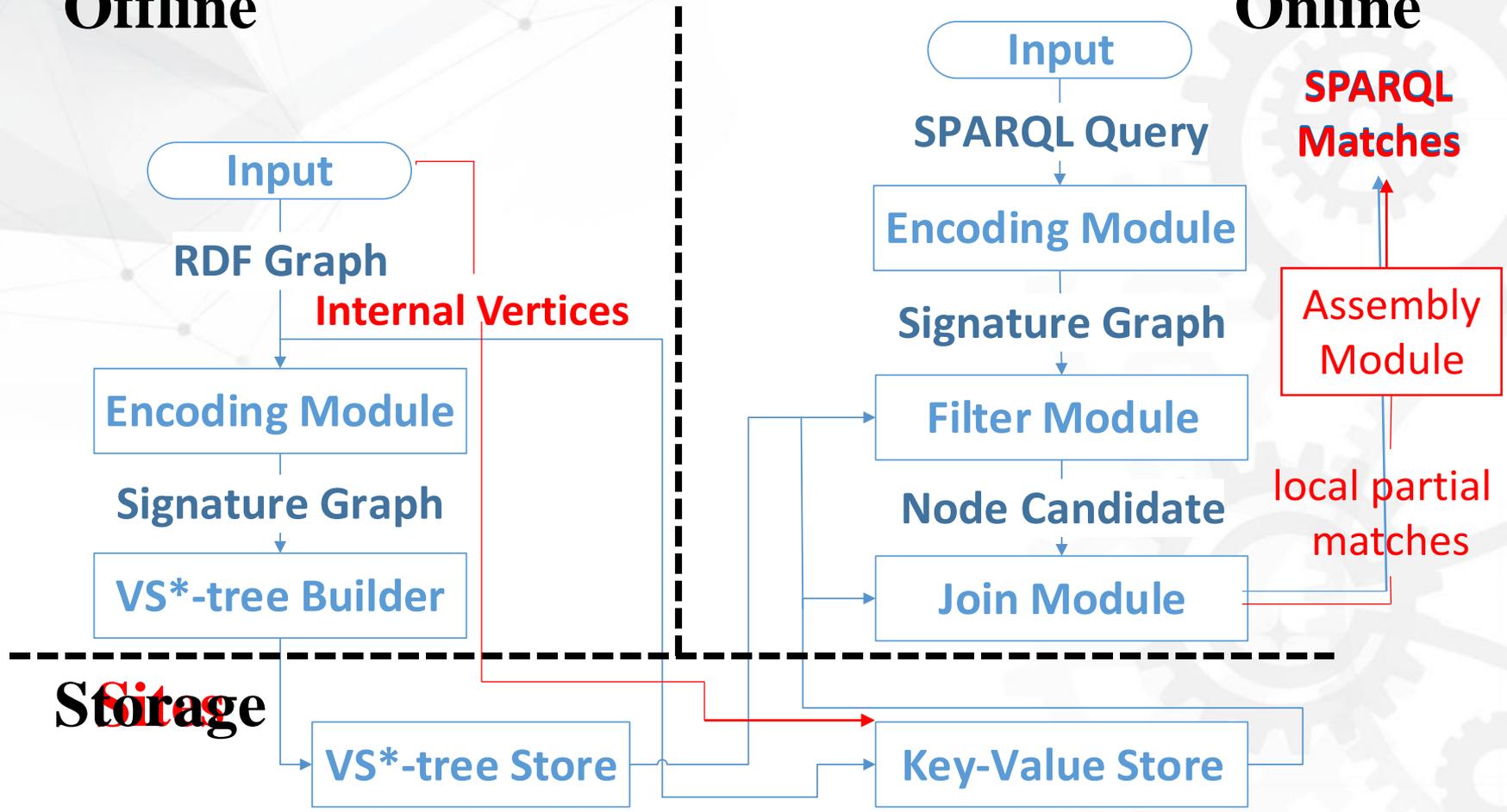
部分解合并



gStore-D: 分布式系统

Offline

Online



gStore-D@Github

Personal Open source Business Explore Pricing Blog Support This repository Search Sign in Sign up

bnu05pp / gStoreD Watch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Pulse Graphs

A Distributed RDF Data Management System for Processing SPARQL Queries Over Distributed RDF Graphs

6 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

bnu05pp Update README Latest commit 64c4d81 4 minutes ago

.debug	Initially upload	21 minutes ago
.objs	Initially upload	21 minutes ago
.settings	Initially upload	21 minutes ago
.tmp	Initially upload	21 minutes ago

提纲

- 知识图谱概述
- RDF数据管理系统——gStore
- **系统应用**
- 总结

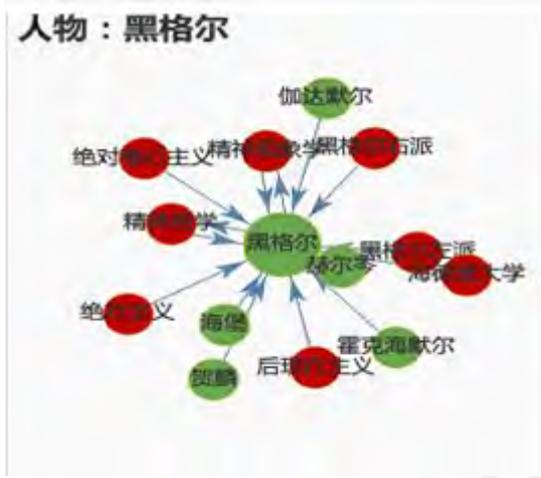
gStore应用

- 方正电子知识出版系统



```
SELECT *
WHERE
{
  ?s <http://www.founder.106.attr:name> ?name.
  {
    ?s <http://www.founder.106.attr:dmlID> "9759". //思想学派
    ?s <http://www.founder.106.link:12855> ?o. //受影响于
  }
}
UNION
{
  ?s <http://www.founder.106.attr:dmlID> "7630". //人物
  ?s <http://www.founder.106.link:12856> ?o. //受影响于
}
?o <http://www.founder.106.attr:name> "黑格尔".
}
```

```
//删除所有与黑格尔有关的三元组
DELETE {?s ?p ?o}
WHERE
{
  ?s ?p ?o.
  { ?s <http://www.founder.106.attr:name> "黑格尔".}
}
UNION
{ ?o <http://www.founder.106.attr:name> "黑格尔".}
}
```



gStore应用

中科院微生物所-全球微生物中心 

三元组	实体
286440147	33183085

Bacteria > Terrabacteria group > Actinobacteria > Actinobacteria > Micrococcales > Micrococcaceae > Micrococcus > Micrococcus luteus

Overview Taxonomy Genome Feature GO Pathway Literature

Species Information

Taxonomy	Bacteria > Terrabacteria group > Actinobacteria > Actinobacteria > Micrococcales > Micrococcaceae > Micrococcus > Micrococcus luteus
NCBI taxonomy ID	1270
Scientific Name	Micrococcus luteus
	Micrococcus luteus CD1_FAA_NB_1
	Micrococcus luteus J28
	Micrococcus luteus Mu201
Children	Micrococcus luteus NCTC 2665
	Micrococcus luteus SK58
	Micrococcus luteus str. modasa
	More
Reference Title In IJSEM	
Type Strains	
Strains	

PREFIX annotation:
`<http://gcm.wdcm.org/ontology/gcmAnnotation/v1/>`

PREFIX taxonomy:
`<http://gcm.wdcm.org/data/gcmAnnotation1/taxonomy/>`

```

SELECT ?taxonId ?name
WHERE
{
    ?taxonId annotation:parentTaxid taxonomy:1270.
    ?nameId annotation:taxid ?taxonId.
    ?nameId annotation:nameclass 'scientificName'.
    ?nameId annotation:taxname ?name.
}
    
```

gStore应用

中科院微生物所-全球微生物中心



三元组	实体
286440147	33183085

PREFIX annotation:
 <http://gcm.wdcm.org/ontology/gcmAnnotation/v1/>
 PREFIX taxonomy:
 <http://gcm.wdcm.org/data/gcmAnnotation1/taxonomy/>

```
SELECT (COUNT(?geneid) AS ?num)
WHERE
{
    {
        ?taxonid annotation:ancestorTaxid taxonomy:1270.
        ?geneid a annotation:GeneNode.
        ?geneid annotation:x-taxon ?taxonid.
    }UNION
    {
        ?geneid a annotation:GeneNode.
        ?geneid annotation:x-taxon taxonomy:1270.
    }
}
```

Number of Gene	54824
Number of Protein	16229

Annotation summary

Proteins with PDB structures	15
Proteins with Pfam assignments	2008
Proteins with GO assignments	32453
Proteins with EC number assignments	680
Proteins with Pathway assignments	2398

Publications and Patents

Publications
Patents

gStore应用

中科院微生物所-全球微生物中心 

三元组	实体
286440147	33183085

Genome

Export Excel

Organism Name	Genome Accession	Description
<input type="checkbox"/> Micrococcus luteus str. modasa	AMYK02000110	Micrococcus luteus str. modasa contig_110, whole genome shotgunsequence.
<input type="checkbox"/> Micrococcus luteus NCTC 2665	CP001628	Micrococcus luteus NCTC 2665, complete genome.
<input type="checkbox"/> Micrococcus luteus SK58	ADCD01000097	Micrococcus luteus SK58 ctg1119142780327, whole genome shotgunsequence.
<input type="checkbox"/> Micrococcus luteus str. modasa	AMYK02000273	Micrococcus luteus str. modasa contig_273, whole genome shotgunsequence.
<input type="checkbox"/> Micrococcus luteus str. modasa	AMYK02000081	Micrococcus luteus str. modasa contig_81, whole genome shotgunsequence.
<input type="checkbox"/> Micrococcus luteus str. modasa	AMYK02000252	Micrococcus luteus str. modasa contig_252, whole genome shotgunsequence.
<input type="checkbox"/> Micrococcus luteus str. modasa	AMYK02000060	Micrococcus luteus str. modasa contig_60, whole genome shotgunsequence.
		Micrococcus luteus str.

```

PREFIX annotation:
<http://gcm.wdcm.org/ontology/gcmAnnotation/v1/>
PREFIX taxonomy:
<http://gcm.wdcm.org/data/gcmAnnotation1/taxonomy/>

SELECT ?taxonid ?name ?genomeid ?description ?strain
WHERE
{
    ?taxonid annotation:ancestorTaxid taxonomy:1270.
    ?nameId a annotation:TaxonName.
    ?nameId annotation:taxid ?taxonid.
    ?nameId annotation:nameclass 'scientificName'.
    ?nameId annotation:taxname ?name.
    ?genomeid a annotation:GenomeNode.
    ?genomeid annotation:x-taxon ?taxonid.
    ?genomeid annotation:definition ?description.
    optional{?genomeid annotation:strain ?strain.}
}
    
```

提纲

- 知识图谱概述
- RDF数据管理系统——gStore
- 系统应用
- **总结**

Take-home Message

1. 路线



基于图的RDF数据管理是可行的技术路线

2. 方法



图匹配查询是有效的技术手段



THANKS