创新 卓越 奋斗 专业

2021年Ai.KG总结报告

北京华云安信息技术有限公司

主要内容

2021年主要工作内容

01 理论研究

02 测试验证

03 工程成果

2022年度规划

04 2022目标

05 实施计划

01. 理论研究

数据展示

数据展示技术进行页面呈现

Ai.KG

④ 知识图谱技术

将关系数据以知识的形态进行表示，以图的形式进行组织，体现出事物内在联系

② ETL技术

对海量数据进行分析、查询、清洗、转换等操作，以实现数据的格式化

③ 自然语言处理技术

通过NLP对非格式化的自然语言进行语义分析以便理解主体及主体间的关系

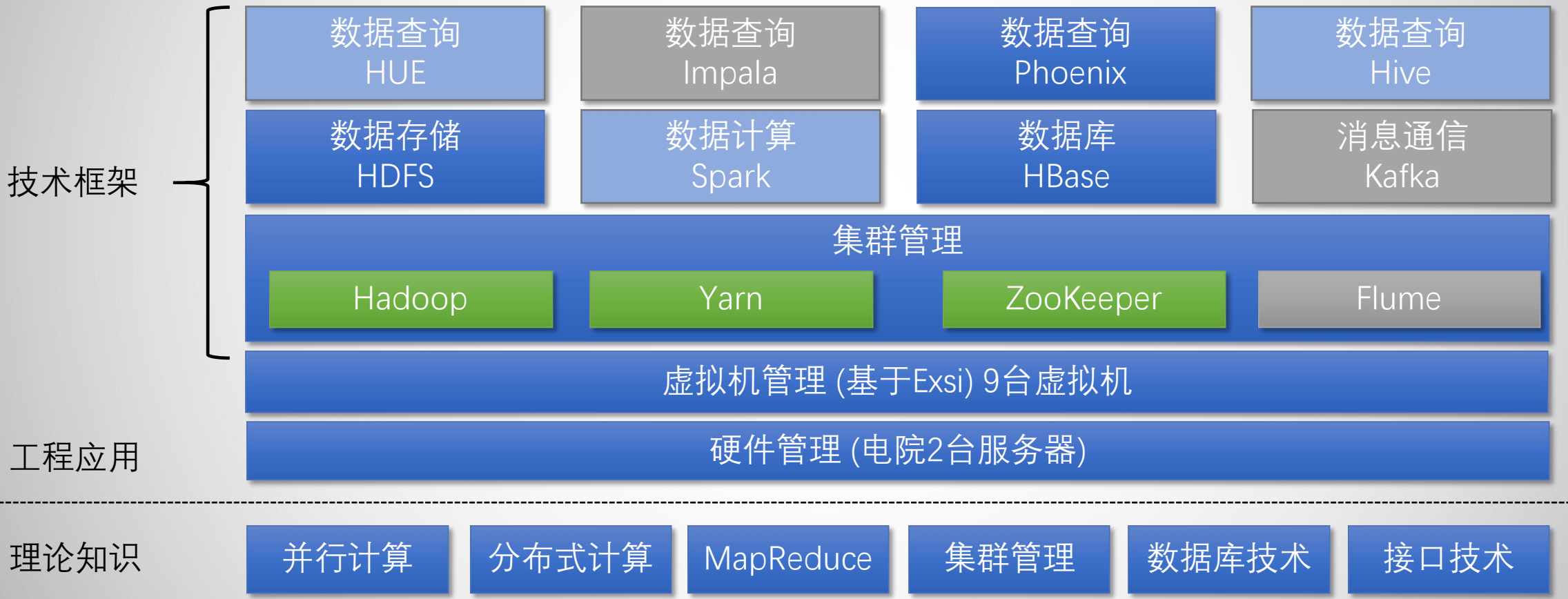
① 大数据技术

通过对计算机集群的架构化管理利用实现对海量数据的存储、查询和使用。

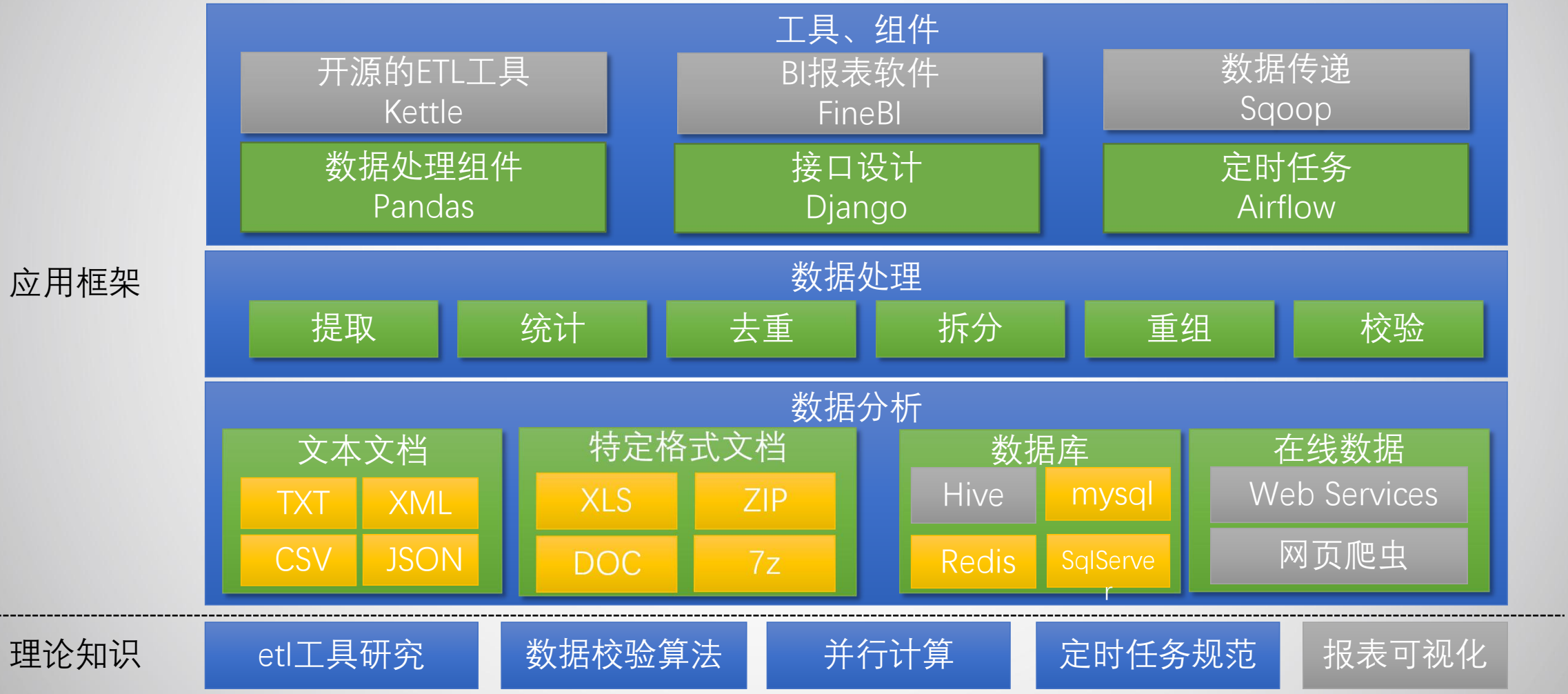
数据获取

各类数据采集技术与获得方式

① 大数据技术


技术框架
数据查询
HUE
数据查询
Impala
数据查询
Phoenix
数据查询
Hive
数据存储
HDFS
数据计算
Spark
数据库
HBase
消息通信
Kafka
集群管理
Hadoop
Yarn
ZooKeeper
Flume
虚拟机管理 (基于Exsi) 9台虚拟机
工程应用
硬件管理 (电院2台服务器)
理论知识
并行计算
分布式计算
MapReduce
集群管理
数据库技术
接口技术

② ETL技术


应用框架
工具、组件
开源的ETL工具
Kettle
BI报表软件
FineBI
数据传递
Sqoop
数据处理组件
Pandas
接口设计
Django
定时任务
Airflow
数据处理
提取
统计
去重
拆分
重组
校验
数据分析
文本文档
TXT
XML
CSV
JSON
特定格式文档
XLS
ZIP
DOC
7z
数据库
Hive
mysql
Redis
SqlServe
r
在线数据
Web Services
网页爬虫
理论知识
etl工具研究
数据校验算法
并行计算
定时任务规范
报表可视化

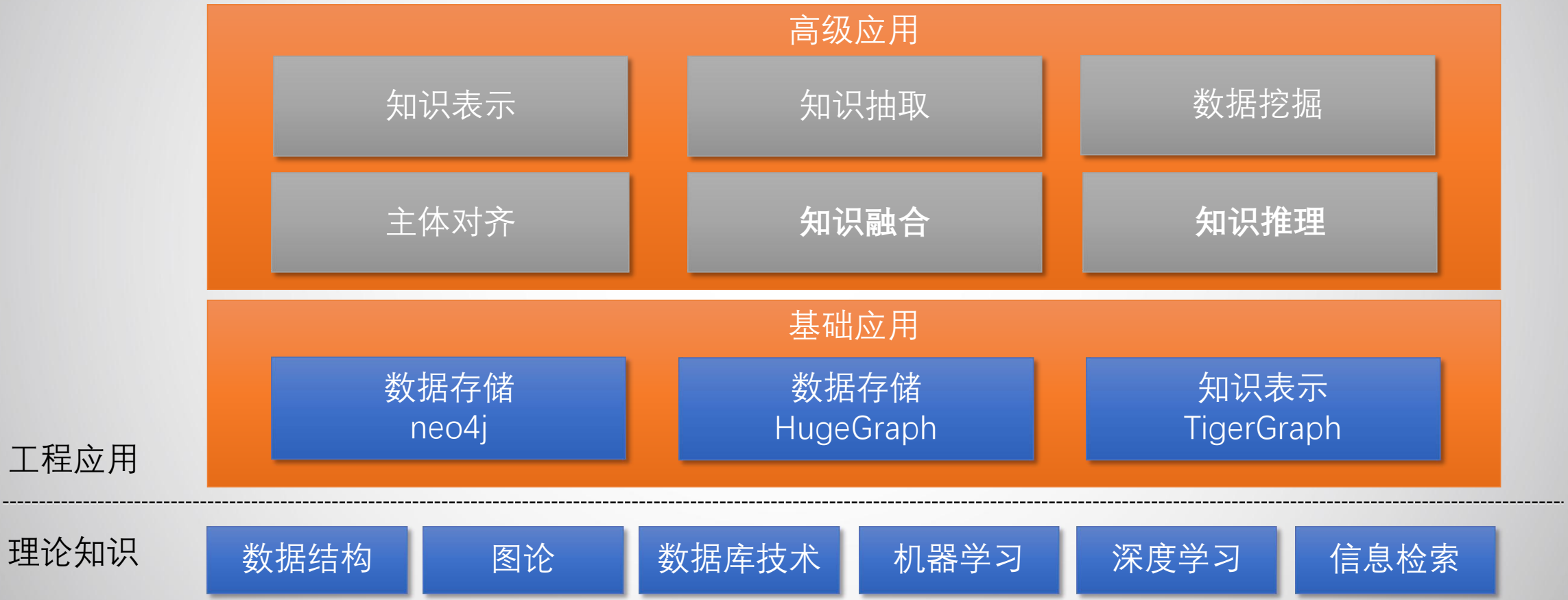
③ 自然语言处理（NLP）

技术框架

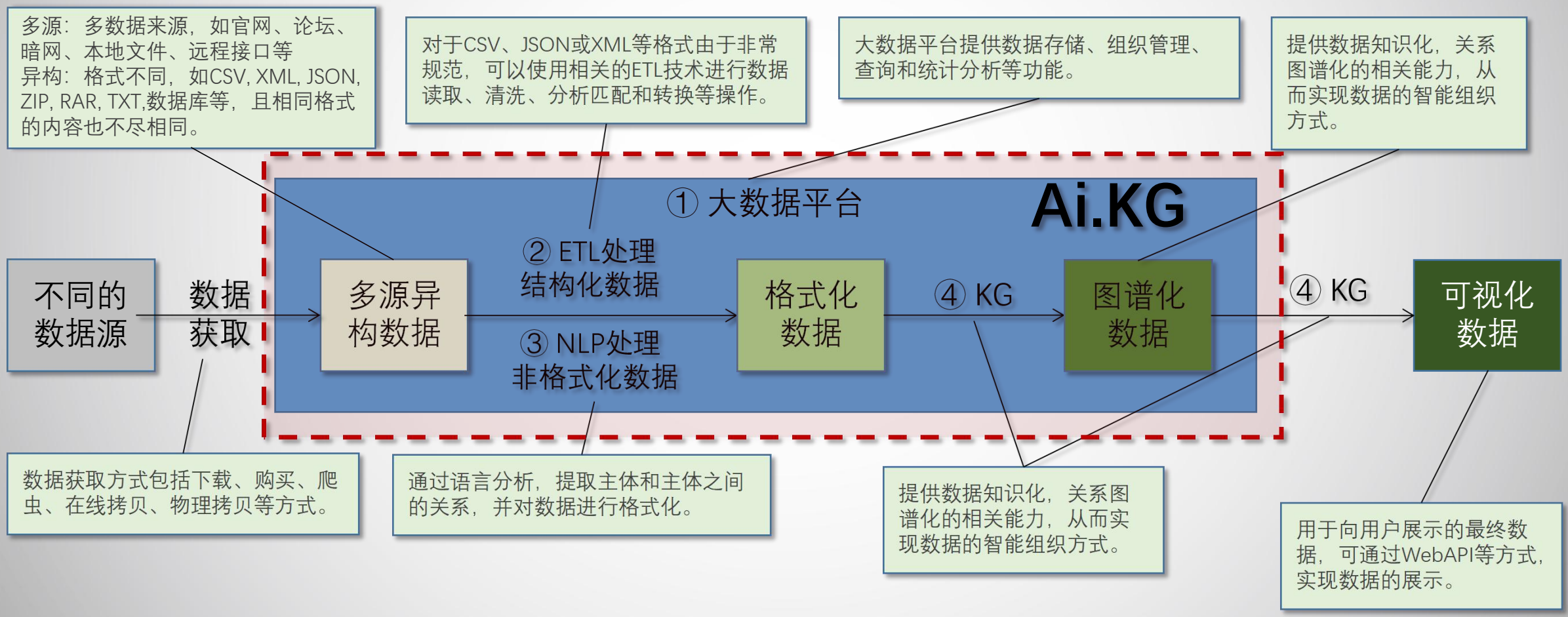
关键字抽取 PageRank	实体抽取 HanLP	情感分析	实体嵌入	三元组抽取
分词 Jieba	实体抽取 HanLP	关键字抽取 TF-IDF	谓语分析 HanLP	语料库 Corpus

应用理论	分类器	决策树	隐马尔科夫	条件随机场	矢量化	距离计算
基础理论	概率论	数理统计	语法知识	编码理论	正则表达式	信息熵论

④ 知识图谱


高级应用
知识表示
知识抽取
数据挖掘
主体对齐
知识融合
知识推理
基础应用
数据存储
neo4j
数据存储
HugeGraph
知识表示
TigerGraph
工程应用
理论知识
数据结构
图论
数据库技术
机器学习
深度学习
信息检索

Ai.KG 数据流处理框架



02. 技术验证：功能测试

主要测试内容列表

1. Hadoop原生平台和CDH的基本功能测试；
2. CDH主要组件的功能用法测试；
3. 基于Django的接口开发与调用方式测试；
4. 基于Django+REST framework+drf-yasg 的在线API文档管理与测试环境的测试；
5. 基于Airflow的定时任务的开发与使用方法测试；
6. TigerGraph的常规用法测试；
7. 集群节点的故障恢复测试；
8. 集群资源动态调整测试，如增减节点、硬盘、内存等；
9. 基于HugeGraph的对象与关系的统一编码方法测试；
10. HUE与Hive、HBase和HDFS的存储模块的整合测试；

02. 技术验证：性能测试

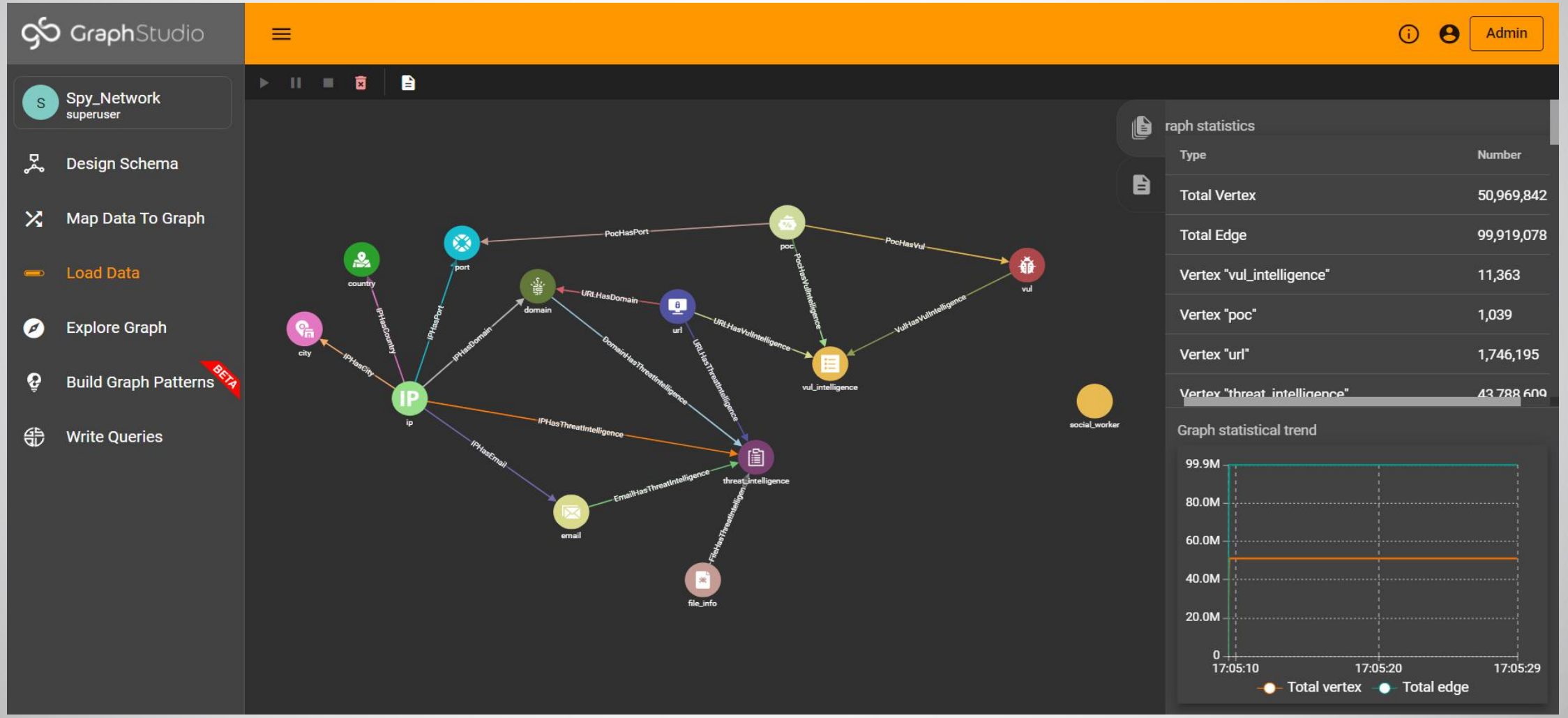
主要性能测试列表

1. neo4j 插入性能测试；
2. Hive的读写性能测试；
3. HBase的读写性能测试；
4. Phoenix索引性能测试；
5. Hive on Impala Spark 的优化机制；
6. 千万级数据数据量下的HugeGraph的读写性能测试；
7. TigerGraph的读写性能与数据容量测试；

03. 工程成果

- 平台运维
  - 搭建了基于Exsi的集群管理环境；
  - 完成了Hadoop原生平台和CDH平台构建；
  - 部署了Hive、HBase、HugeGraph、Phoenix等**10**种大数据组件。
- **数据开发**
  - 数据提取：提取了**1.13万**条漏洞信息、**4300**万条威胁情报和**13.3亿**条SG数据；
  - 图谱构建：建立了具有**5000**万主体与**1亿**关系的知识图谱。
- **接口开发**
  - V1.0，**104** 个接口，每个页面对应一个接口；
  - V1.1，**42**个接口，由于页面内容重复较多，所以根据业务内容将重复接口合并为42个；
  - V2.0，**5+26+4**个接口，包括参数化的数据接口5个和与业务逻辑相关的接口26个和工具类接口4个。
- 自动化处理
  - 编写了**5**项定时数据分析任务，如数据实时导入、数据统计分析和已有数据的正确性验证等；
  - 开发了**1**款将文本文件导入至HBase的数据转换工具。

03. 工程成果：数据展示


GraphStudio
Admin
Spy_Network
superuser
Design Schema
Map Data To Graph
Load Data
Explore Graph
Build Graph Patterns
BETA
Write Queries
raph statistics
Type
Number
Total Vertex
50,969,842
Total Edge
99,919,078
Vertex "vul_intelligence"
11,363
Vertex "poc"
1,039
Vertex "url"
1,746,195
Graph statistical trend
99.9M
80.0M
60.0M
40.0M
20.0M
0
17:05:10
17:05:20
17:05:29
Total vertex
Total edge
PocHasPort
PocHasVul
URLHasDomain
URLHasVulIntelligence
VulHasVulIntelligence
IPHasCountry
IPHasPort
IPHasDomain
IPHasCity
IPHasThreatIntelligence
IPHasEmail
DomainHasThreatIntelligence
URLHasThreatIntelligence
EmailHasThreatIntelligence
FileHasThreatIntelligence
port
poc
vul
country
domain
url
city
ip
vul_intelligence
social_worker
threat_intelligence
email
file_info

03. 工程成果：接口展示

功能页面	V1.0 接口数量	V1.1 接口数量	V2.0		
			数据层接口量	业务层接口量	性能 (以IP接口为例)
首页	2	2		1	100-200ms
首页-地球	2	2		1	100-200ms
数据的基本信息	16	7	1	8	单条：300-400ms 多条：75-80s/1000
数据的关联信息	53	17	1	5	300-400ms
流行度（阅读量）	6	4	1	5	100-200ms
画像（图谱与图例）	11	6	1	5	700-800ms
检索	14	4	1	1	模糊查询：3-4s 精确查询：100-200ms
数据统计	-	-	4	-	秒级
汇总	**104**	**42**	**5+4**	**26**	**-**

注：V1.0和V1.1缺少数据故未测试性能。

主要问题

- 缺少自动化的ETL管理工具
  - 数据上传：原始数据种类繁多，统计、上传、解压等均需手工完成；
  - 数据分析：对已上传数据内容分析，转换规则的确认，数据初步清洗；
  - 数据入库：数据转换、结果验证等；
  - 数据统计：对转换数据、已有数据等的统计信息；
- 硬件基础不可靠
  - 节点不可靠：集群的所有节点均是在2台主机上虚拟出来的；
  - 服务器不稳定：电源服务器经常停电，甚至因停电导致数据损坏1次。
- 人力资源不足（2021年总体情况及甘特图）
  - 许文林，2021/03 -- 2021/07 □□■ ■■■ ■□□ □□□
  - 吴脂娟，2021/06 -- 2022/12 □□□ □□■ ■■■ ■■■
  - 陈一根，2021/07 -- 2021/08 □□□ □□□ ■■□ □□□

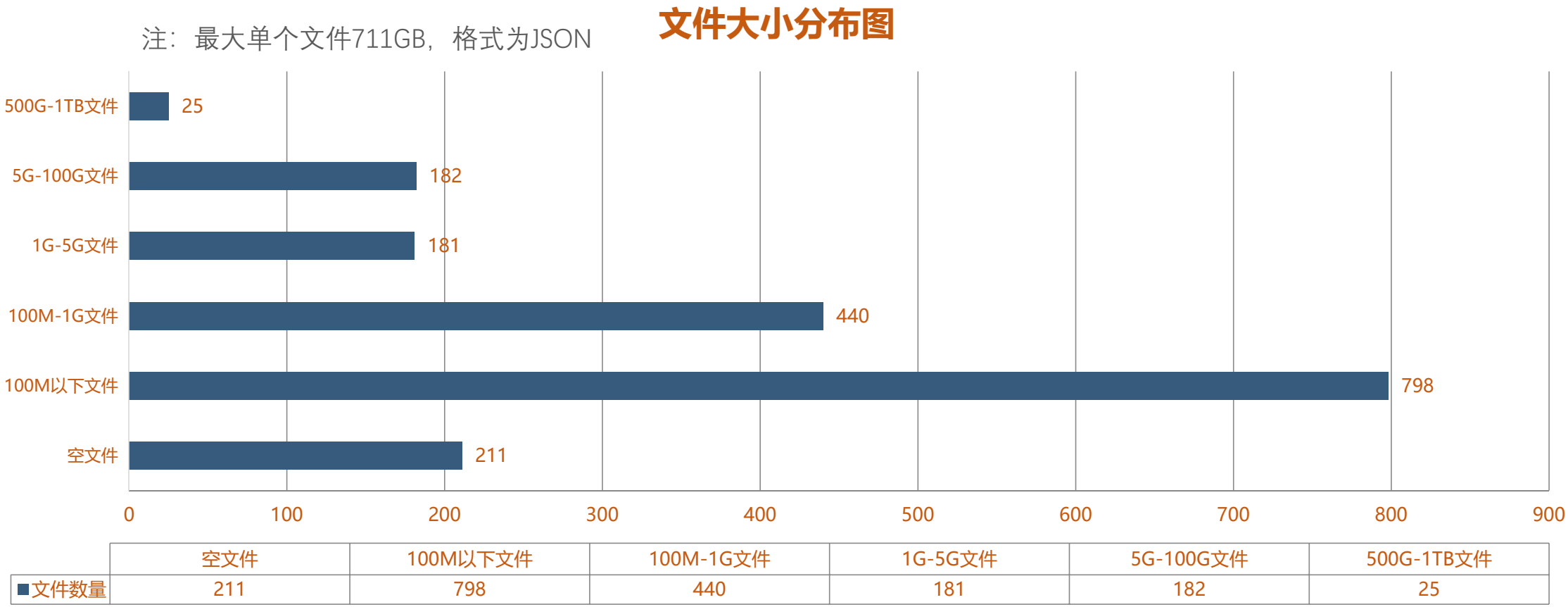
04. 2022目标：主要目标列表

- 建立功能完善的自动化的大数据管理平台；
- 进一步充实网络安全知识图谱；
- 基于此知识图谱实现至少2个产品级应用。

11.9TB数据文件大小分布图

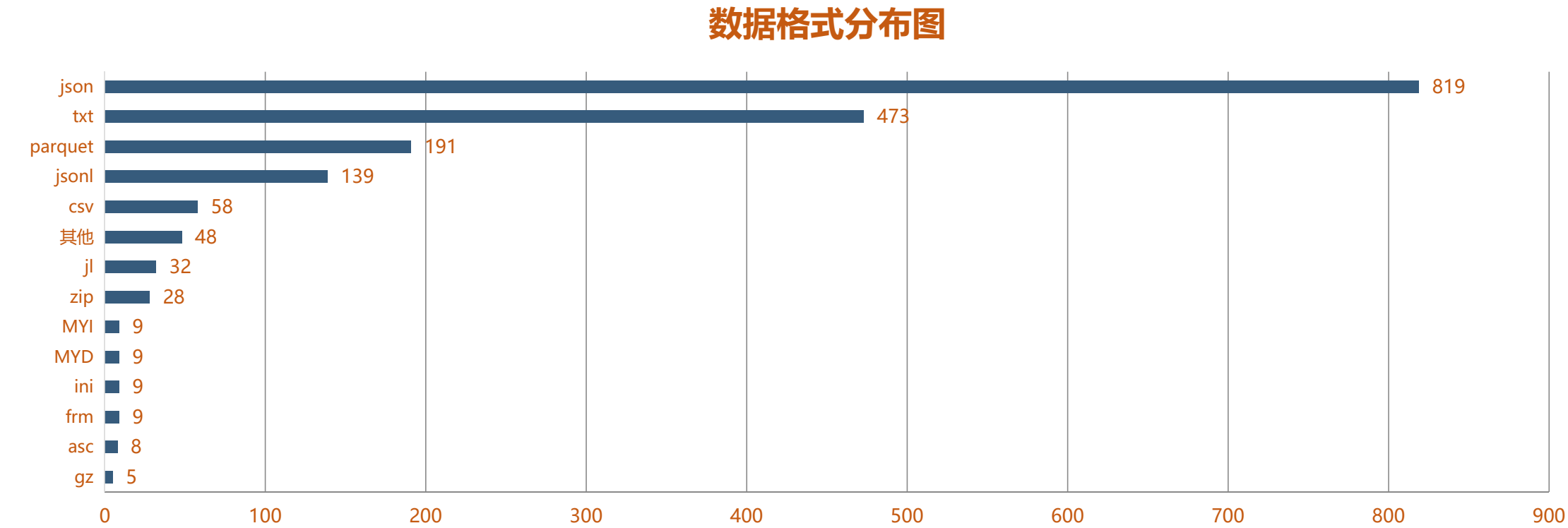
文件总容量11.9TB,文件数量 1837,文件夹数量 201

注：最大单个文件711GB，格式为JSON



11TB数据格式分布图

文件总容量11.9TB,文件数量 1837,文件夹数量 201



	gz	asc	frm	ini	MYD	MYI	zip	jl	其他	csv	jsonl	parquet	txt	json
文件数量	5	8	9	9	9	9	28	32	48	58	139	191	473	819

05. 实施规划：大数据文件管理平台

计划开发一款用于大数据文件的综合管理平台，用于管理量的数据文件，平台主要有以下功能：

1. 能够对所有所有文件信息各类信息统计并进行相关展示；
2. 能够根据条件对所有文件进行筛选、查询和对比；
3. 能够对单个文件进行必要的数据分析，如分析出文件的行数，平均行数长度，每行字段数，数字摘要MD5，主要内容等操作；
4. **能够自动识别数据类型，并使用对应转换模块对文件进行格式转换；**
5. 能够对单个或多个文件进行打标签等分类操作；
6. 支持文件的导入、导出、打包、发布下载链接等；
7. 能够记录文件的读写、分析、转换和导入导出历史；
8. 支持其他可能用到的文件管理功能。

05. 实施规划：进一步丰富Ai.KG的数据

通过多种方式的数据收集方式，进一步丰富Ai.KG的数据：

- 以漏洞、威胁情报、网络资产和SG为核心
- 提高图谱中的实体和关系数量；
- 扩展主体、关系和属性的类型；

05. 实施规划：产品级应用落地

- 产品级
  - 高性能，进一步优化数据处理能力，提高数据查询性能；
  - 高可用性，从硬件及架构层面提高平台的可用性。
- 应用落地（建议）
  - 与灵洞结合，提高灵洞的数据关联性
    - 丰富漏洞及相关信息的主体化
    - 建立更多的关系表示
  - 与灵刃结合，提高对目标的理解准确度
    - 将漏洞与POC、EXP等进行关联
  - 与灵源结合，提高识别能力程度。

05. 实施规划：人力资源岗位建议

序号	岗位名称	岗位人数 在岗/规划	岗位职责
1	KG研究员	4/4	负责实体表示、主体识别、主体对齐、关系抽取、知识计算、图谱应用等方面的研究工作。
2	系统架构师	1/1	负责整体Ai.KG的架构设计。
3	数据开发工程师	1/2	负责数据爬取、数据分析、数据转换脚本编写、数据查询验证、数据接口编写结构化数据的处理工作。
4	NLP工程师	0/1	负责非结构化数据的分析与解释，为知识图谱提供元数据。
5	平台运维	0/1	负责大数据平台的功能调研与日常运行维护。

谢谢！

附1： 部分测试内容

- 实验内容：通过50万节点插入验证neo4j插入性能。
- 测试环境1： 1核 Intel(R) Xeon(R) CPU E5-2630 @ 2.30GHz， 4G内存，16G内存。
- 主要结论：
  - 内存不足会导致实验失败；
  - 插入速度不稳定波动较大；
  - 速度越来越慢下降明显，执行速度越来越慢，执行速度从200-400节点/秒，降低到最后的0.19节点/秒
- 测试环境2：2核 Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz，8g内存，100G硬盘
- 结论：
  - 总用时33分钟，相当于每小时约90万条数据，关系的插入与节点的插入有所不同，主要包括两点：关系除了自身属性还要包括两个节点的关系；关系中的节点如何进行定位；
  - 发现使用 Where 子句匹配时，需要花费很多时间，是一个需要重要考虑的内容，所以首先进行了关于索引的测试。

01 Ai.KG主要研究内容：neo4j性能测试

- 实验内容：200万节点的索引测试
  - 使用索引查询节点，使用索引能够有效提高节点的检索速度；
  - 关系插入时的索引对性能的影响。
- 结论：
  - 对于图数据库，索引同样能够加速检索；
  - 整型比字符串的索引效率更高；
  - 所以，后面关系的插入也同样利用索引进行加速。另外，通过实验还发现 neo4j 需要一定量的内存，至少2G起步，多多益善。

01 Ai.KG主要研究内容：neo4j社区版节点添加性能测试

- 测试环境：
  - CPU：Intel(R) Xeon(R) CPU E5-2630 @ 2.30GHz （2012Q1发布）
  - MEM：MemTotal: 16,266,540 kB，MemFree: 356,032 kB，MemAvailable: 7,566,740 kB
- 测试内容1：多节点同时插入测试，插入性能可以提高约12倍，插入速度达到500个节点/秒
- 结论：
  - 通过修改每次添加节点的数量，可以实现每秒约500个节点的添加，相当于每天4000万条；
  - 对于目标数据的约240万条数据（43万个节点和196万条关系）插入时间约为 1.44 小时；

  PS. 后面又调整count和batch_count进行了些测试，发现速度基本稳定在500条/s左右。
- 测试内容2：节点插入和删除测试结果，插入节点速度为40个节点/秒
- 结论：通过远程单个节点添加的方式，大约每秒可以添加40个节点，而删除的每秒可以删除2万个节点。

01 Ai.KG主要研究内容：HugeGraph研究与测试

- 测试内容1 RocksDB单机版性能（48CPUs 128G内存 HDD盘）：
  - RocksDB单机批量写性能：关闭label index，22.8w edges/s；开启label index，15.3w edges/s；
  - RocksDB单机单条写性能：并发9000，吞吐量是8418，边的单条插入并发能力为9000；
  - RocksDB单机随机读性能：并发13000，吞吐量是12225 edges/s，边的按id查询的并发能力为13000，平均延时为12ms。
- 测试内容1 Cassandra集群版性能（15节点，48CPUs 128G内存 HDD盘）：
  - Cassandra集群批量写性能：默认开启label index，6.3w edges/s；
  - Cassandra集群单条写性能：并发4500，吞吐量是4160 edges/s，边的单条插入并发能力为4500；
  - Cassandra集群随机读性能：并发12000，吞吐量是10688 edges/s，边的按id查询的并发能力为12000，平均延时为63ms。
- 结论：达到千万级别，亿级的数据量，HugeGraph的性能会逐渐减低，不适用与要求查询性能达到秒级的大数据项目。

01 Ai.KG主要研究内容：Tigergraph研究与测试

- 测试内容1：TigerGraph写入性能测试
- 结论：TigerGraph对于超大规模数据实时写性能表现较好，同时也反映了想要获取更好的性能需要更多的服务器资源。测试中TigerGraph的Kafka实时写入和离线写入性能最终趋向一致，入库速率与批次入库数量成正比增长。TigerGraph的写入速度与单请求中记录数目具有极大关系，且随着单请求中记录数目的增加而增加，并始终保持匀速正增长，但增速缓慢。从图中可以看出，TigerGraph在本测试实时每批次写入5,000,000条事件记录和离线每批次写入500,000条事件记录时达到最高写入性能，写入速度大约为80,000事件记录/秒
- 测试内容2：TigerGraph读取性能测试
- 结论：TigerGraph的最大读取速度在12,000记录/毫秒左右，因其高级查询语句中支持累加器操作，所以我们针对测试场景定制了一个查询函数，用于累计每个实体的一度关系，它将并行的将函数预先运行到每个实体中，安装函数的步骤会花费几秒钟的时间，以后直接运行查询就会很快

01 Ai.KG主要研究内容：Hive研究与测试

- 测试内容1：运行模式（本地模式/集群模式）测试
- 测试内容2：计算引擎 （MR/TEZ/SPARK）测试
- 测试内容3：数据写入，读取性能等方面的测试
- 结论：
  - 通过对小数据集分别在本地模式运行的速度为5.402s，集群模式下运行的速度为24.387s，可以看出，数据集小的在本地模式下运行速度更快；
  - hive适用于离线数据，千万级，亿级大数据写入数据性能很好，采用TEZ/spark的计算引擎，查询性能提升了5，6倍；
  - 数据存储与hdfs中，hive建表需要指定分隔符，对于情报数据中域名，url包含各种特殊符号，容易导致此类数据错位；
  - hive sql 不支持插入数据，导致每次有新数据录入时，需要新建表格，合并数据等等，不适用与自动化项目。

01 Ai.KG主要研究内容：Hbase研究与测试

- 测试环境：8核，共享内存200G,2T硬盘，6台虚拟机，1000万数据，1.1G hbase+phoenix
- 测试内容1： 测试建立索引对数据写入性能的影响：
- 结论：不建索引，写入时间为348.253s,建一个索引，写入时间为563.222s,建两个索引677.93s,可以看出建索引会影响数据写入时间，索引越多，写入时间越长。

01 Ai.KG主要研究内容：Hbase研究与测试

- 测试内容2： 亿级数据查询性能:
  - sql:select count(id) from SOCIAL_WORKERS;　用到主键时间106.843s；
  - sql:select id from SOCIAL_WORKERS where user is not null limit 100000;　用到索引功能,时间5.126s；
  - sql:select id from SOCIAL_WORKERS where user = '! ALymnamill';　用到索引功能,时间0.071s；
  - sql:select * from SOCIAL_WORKERS where user = '! ALymnamill';　没有用到索引功能,361.74s；
  - sql:select * from SOCIAL_WORKERS where id = 'sociaworke20211108010291158035';　用到主键时间0.185s；
- 结论：
  - 批量数据写入建立采用离线方式，以文件格式导入数据库，而脚本写入数据库耗时太久，数据量小建议脚本写入；
  - 在数据没有特殊符号的情况下，使用集群MR导入模式，减少写入时间，但在特殊符号的情况下，只能采取单机导入，集群模式会报错失败，提示无效字符；
  - 要想实现秒级查询的效果，不仅需要建索引，而且在语句上格式也主要注意是否引用了索引，比如：查询某字段的某值所在行的所有字段数据，建议该字段建立索引，先查询该行的主键ID，再根据主键ID查询该行所有数据,充分利用索引查询。

01 Ai.KG主要研究内容：CDH和Apache版本大数据平台

根据部署Apache版本与CDH版本的集群的过程，以及后续的监控，问题定位的情况等得出以下结论：

- Apache社区版：
  - 版本管理比较混乱，各种版本层出不穷，很难选择，选择其他生态组件时，比如Hive，Sqoop，Flume，Spark等，需要考虑兼容性问题、版本匹配问题、组件冲突问题、编译问题等；
  - 集群安装部署复杂，需要编写大量配置文件，分发到每台节点，容易出错，效率低；
  - 集群运维复杂，需要安装第三方软件辅助。
- CDH：
  - 版本管理清晰。
  - 版本更新快。通常情况，比如CDH每个季度会有一个update，每一年会有一个release；
  - 集群安装部署简单。提供了部署、安装、配置工具，大大提高了集群部署的效率；
  - 运维简单。提供了管理、监控、诊断、配置修改的工具，管理配置方便，定位问题快速、准确，使运维工作简单，有效；
  - cdh5以及停止更新与支持，CDH5里大数据集群组件版本都比较低，建立使用CDH6。