

LDBC Benchmark on JanusGraph

0：IO | 大数据系统与大规模数据分析课程报告 | 2018.6.19

张俊阳 | 2017E8018661163 | zhangjunyang@iie.ac.cn

张小洋 | 2017E8018661065 | zhangxiaoyang@iie.ac.cn

王传仁 | 2017E8018661118 | wangchuanren@iie.ac.cn

刘志磊 | 201728018629141 | liuzhilei@iie.ac.cn

冀海川 | 2017E8018661165 | jihaichuang@iie.ac.cn

# LDBC SNB简介

LDBC社交网络图数据Benchmark是一项针对专业图数据库的基准测试工具，它包含一个数据生成器生成社交网络数据，三种工作负载（workloads）：交互式(Interactive)，商业智能(Business Intelligence)和图分析( Graph Analytics)。目前，只有交互式工作负载在初稿阶段发布。商业智能工作负载的只读部分目前能够预览。

主要的SNB主要部件有下列四项

* SNB benchmark 规范文档 <https://github.com/ldbc/ldbc_snb_docs>
* SNB数据生成器 <https://github.com/ldbc/ldbc_snb_datagen>
* LDBC驱动（实现查询的驱动）<https://github.com/ldbc/ldbc_driver>
* 交互式工作负载实现 <https://github.com/ldbc/ldbc_snb_implementations>

# JanusGraph简介

JanusGraph是一个可扩展的图数据库，可以把包含数千亿个顶点和边的图存储在多机集群上。它支持事务，支持数千用户实时、并发访问存储在其中的图。

除此之外JanusGraph还具有下列特点：

* 弹性，线性可扩展
* 针对性能和容错的数据分发和复制机制
* 多数据中心高可用性和热备份
* 支持ACID和eventual consistency
* 支持多个后端存储
* 通过与大数据平台集成支持全局图数据分析，报表和ETL
* 通过集成索引引擎支持地理位置，数字和全文搜索
* 原生集成Apache Tinker Pop图技术栈
* 开源，基于Apache 2 Licenses
* 通过可视化工具展示存储在JanusGraph中的图

基于LDBC 社交网络图实现Benchmark，并比较不同的选项对性能的影响，具体分为以下五个章节：

***{数据生成部分***

***数据导入***

***工作负载实现***

***Benchmark运行***

***性能比较及分析}***

# Importer 数据导入到JanusGraph

该部分承接上一步数据生成的完成，由于实现具体的benchmark查询(Query)和生成数据高度相关，因此对于数据生成所用到的参数会在下一章节工作负载实现中使用，该部分主要着重阐述将生成的social\_network数据导入到JanusGraph后端存储中，通过执行com.ldbc.snb.janusgraph.importer.Main类来导入，基于LDBC ldbc\_snb\_implementations开源代码修改实现。

导入前需要将JansusGraph server（gremlin-server）启动

具体执行方式为：

java -cp target/janusgraphSNBInteractive-0.1-SNAPSHOT-jar-with-dependencies.jar com.ldbc.snb.janusgraph.importer.Main -n 2 -s 2000 -d test-data-100m/social\_network -c bdb.conf

其中参数解释：

--numThreads/-n 加载过程中线程数

--transactionSize/-s 读取文件事务的大小（每个读取任务读取的行数）

--dataset/-d 要导入数据集的文件夹路径

（数据集的时间戳必须采用Long，使用ldbc.snb.datagen.serializer.formatter.LongDataFormatter生成数据 ）

--backend-config/-c 配置后端存储的文件路径。示例文件resources/bdb.conf

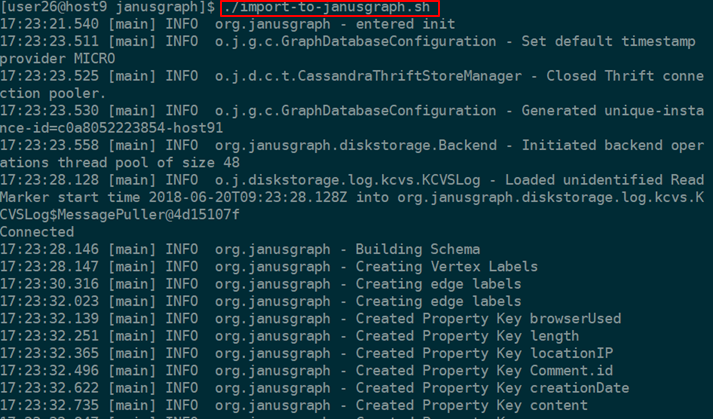
通过shell 脚本import-to-janusgraph.sh 导入数据

./import-to-janusgraph.sh 导入到Cassandra后端存储

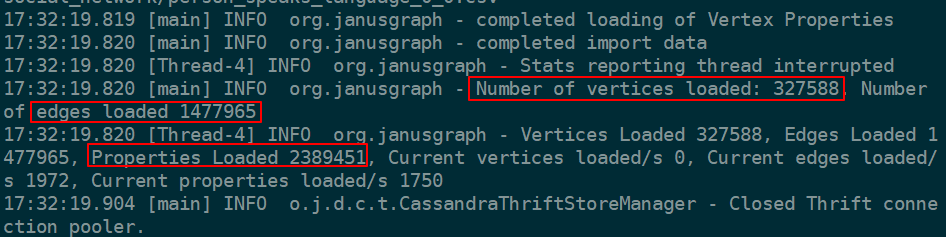
./import-to-janusgraph-berkeley.sh 导入到BerkeleyDB后端存储

./import-to-janusgraph-hbase.sh 导入到Hbase后端存储

示例导入过程



导入完成



测试100m数据导入了327588个点，1477965条边，2389451条properties。

# 工作负载（workload）实现

该部分基于<https://github.com/ldbc/ldbc_snb_implementations>官方LDBC Interactive代码中针对Titan实现的交互式工作负载benchmark部分实现。

主要组件实现有：

ReadOperation、读操作

UpdateOperation、更新操作

JanusGraphDb、JanusGraph数据库连接器

LdbcQuery[1-14]Handler、14个复杂查询

LdbcQueryU[1-8]Handler.java、8个更新操作

LdbcShortQuery[1-7]Handler，7个简单查询

具体的Query规范在ldbc-snb-specification.pdf 文件中有详细描述。具体实现代码请看代码仓库或者附件。

该部分需要LDBC SNB Dirver包，打包前在pom.xml引入如下依赖

<dependency>

<groupId>com.ldbc.driver</groupId>

<artifactId>jeeves</artifactId>

<version>0.3-SNAPSHOT</version>

</dependency>

# Benchmark运行

## 配置后端janusgraph

## 交互式工作负载benchmark

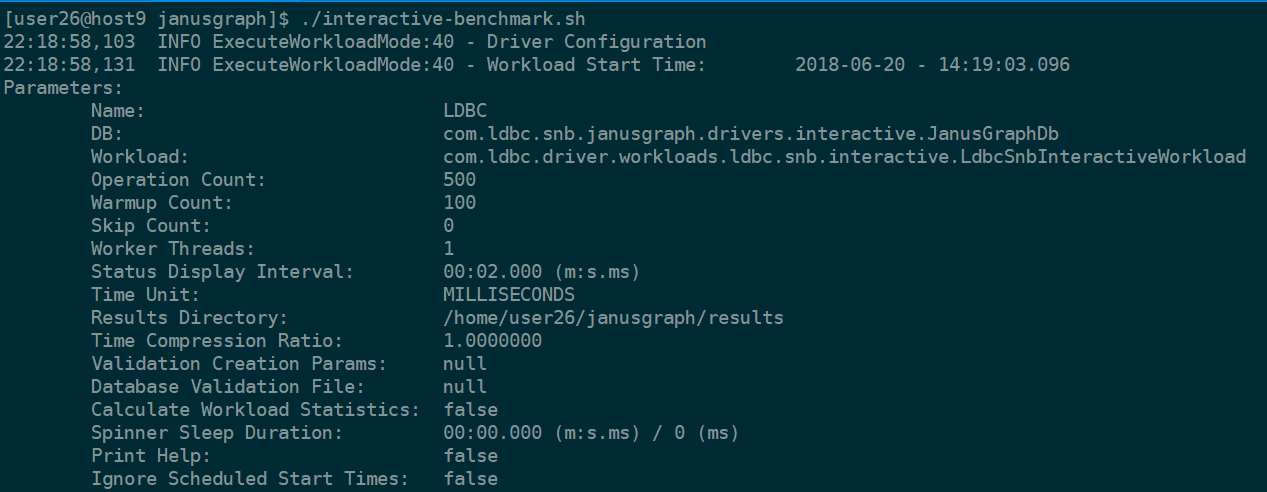
首先获取<https://github.com/ldbc/ldbc_snb_driver>中的源码，然后使用maven编译安装driver包

mvn clean install -DskipTests

接着进入项目janusgraph目录下生成实现benchmark的JAR包

mvn clean assembly:assembly -DskipTests

对于我们的Janusgraph 交互式工作负载benchmark实现，首先根据系统配置修改ldbc\_snb\_interactive.properties文件，接着使用脚本interactive-benchmark.sh执行驱动程序运行基准测试。



# 性能比较及分析

同样参数情况下对于不同的Cassandra集群测试结果如下：

Operation Count: 482

embedded Cassandra

Throughput: 65.78 (op/s)

Throughput: 65.92 (op/s)

Throughput: 65.92 (op/s)

Throughput: 65.59 (op/s)

Cassandra cluster 2 节点

Throughput: 24.34 (op/s)

Throughput: 66.05 (op/s)

Throughput: 65.89 (op/s)

Throughput: 65.96 (op/s)

Cassandra cluster 4节点

Throughput: 21.35 (op/s)

Throughput: 45.25 (op/s)

Throughput: 66.05 (op/s)

Throughput: 48.00 (op/s)

由于测试图数据集较小结果差距不大，但多节点Cassandra集群benchmark稳定性较弱，这可能和集群之间需要较多通信有关。