JanusGraph的后端存储配置

JanusGraph支持多种数据存储后端，包括：Apache Cassandra、Apache HBase、Google Cloud Bigtable和Oracle Berkeley DB。本次实践中使用了Apache Cassandra和Orcale Berkeley DB Java Edition分别作为JanusGraph的存储后端进行测试。

1. JanusGraph + Apache Cassandra 集成配置。

Apache Cassandra是一个开源的、分布式、无中心、支持水平拓展、高可用的KEY-VALUE类型的NOSQL数据库[1]。根据JanusGraph的官方文档介绍，Cassandra作为JanusGraph的存储后端时有多种集成模式[2]，这里使用了Remote Server模式，Cassandra以集群方式存在， 运行在其他主机上的JanusGraph基于Socket的读/写来访问Cassandra集群。我们通过控制Cassandra集群的节点数量，运行LDBC的社交网络图Benchmark对JanusGraph的性能进行测试。测试中使用的Cassandra版本是3.11.2。

1.1 单节点的Cassandra连接配置

在本例中JanusGraph与单个节点的Cassandra进行连接，Cassandra运行在实验主机host5上，其主要配置文件conf/cassandra.yaml中的关键配置如下：

|  |
| --- |
| # 集群名称  cluster\_name**:** 'IO Cassandra Cluster'  # 设置种子节点 种子节点为host9 ip地址为192.168.5.34  seed\_provider**:**  -class\_name**:** org.apache.cassandra.locator.SimpleSeedProvider  parameters**:**  -seeds**:** "192.168.5.34"  # 设置监听地址  listen\_address**:** 192.168.5.34  # 启用thrift rpc server，否则JanusGraph无法与Cassandra建立连接  start\_rpc**:** true  rpc\_address**:** 192.168.5.34 |

JanusGraph的相关配置文件（主要是gremlin-server.sh加载的配置文件和运行benchmark的程序加载的配置文件）中添加如下配置：

|  |
| --- |
| storage.backend**=**cassandrathrift  storage.hostname**=**192.168.5.34 |

1.2 两节点的Cassandra连接配置

在本例中我们部署了两个节点的Cassandra集群，使用host9作为集群的seed节点，host4作为普通节点，每个节点的Cassandra配置同上例中的单节点配置类似：seeds设置为host9的ip地址192.168.5.34，listen\_address和rpc\_address设置为各自对应的ip地址，启动时先启动seed节点上的Cassandra进程，之后普通节点启动会自动加入到集群中。集群启动后使用Cassandra提供的nodetool工具查看各节点的状态，如图1.1所示，我们向JanusGraph导入了约100M的数据，这些数据被分配存储在了Cassandra集群的节点中。

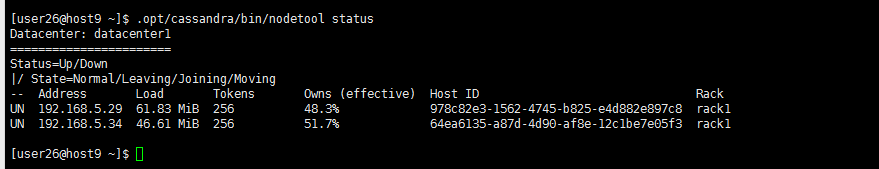
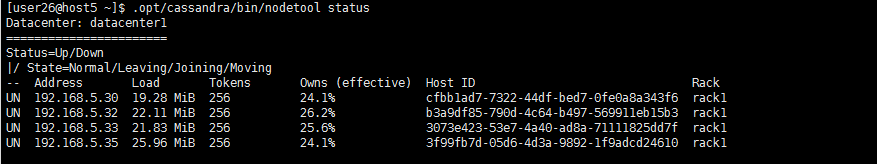


图 1. 1 两节点Cassandra集群状态

1.3 四节点的Cassandra连接配置

为了进一步验证Cassandra集群节点数量对测试性能的影响，在本例中我们使用host5、host7、host8、host10部署在四个结点的Cassandra集群，由于在这些机器上有其他用户运行的Cassandra进程造成了端口占用，除了前面例子中的配置外还需要修改Cassandra的端口，关键配置如下：



参考文献

[1]