IS607 - Project 5: Managing the Logistics of Physical Natural Gas Operations in Neo4j

Derek G. Nokes

Sunday, May 10, 2015

Contents

Business Use Case

The logistics associated with the trading of physical natural gas are complex. In simple terms, pipelines move gas from areas of excess supply to areas of excess demand. Locations where many pipelines connect are referred to as 'hubs'. Physical natural gas is priced at all hubs. These prices represent the cost of moving gas through the network of pipelines, hubs, and storage facilities.

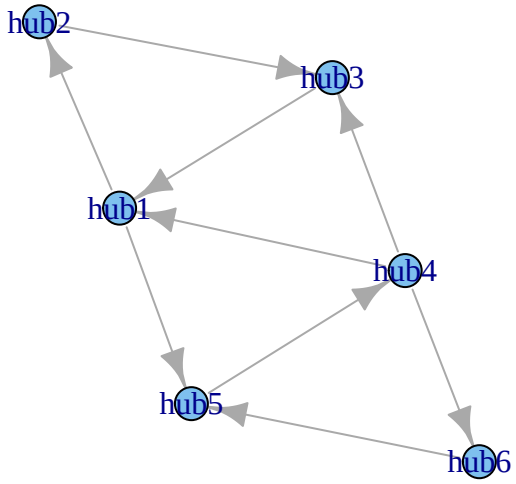
When demand exceeds supply in a particular location, prices rise in relation to other locations and gas tends to be re-routed to high-demand points. Physical constraints that prevent gas from flowing to a particular location create price spikes. Conversely, in areas of excess supply, prices tend to fall.

Gas can typically only flow in one direction along a pipeline. The physical characteristics of the pipelines determine flow capacity. Maintainence along pipelines - for instance - creates capacity constraints that temporarily prevent gas from freely following over particular paths. Such constraints can create incredible volatility in gas prices.

Contracting to buy gas at one location and sell that gas at another location for more than the transportation and other costs results in profits. To run a profitable natural gas trading business, one must be able to understand the natural gas network.

Graphs provide a very natural abstract representation for the network of physical natural gas hubs connected by pipelines.

In the diagram below we represent 6 hubs connected by pipelines. The direction of gas flow is represented by an arrow.



Nodes, Relationships, and Attributes

Each hub is connected to at least one pipeline. The cost of trasporting gas from one hub to another is roughly the price spread between the quoted prices at the two hubs.

Nodes

We define each hub as a node in our graph.

Relationships

Pipelines - which allow physical gas to flow from one hub to another - connect nodes.

Attributes/Properties

Each path from one hub to another has a particular associated transportation cost. We can imply these costs from traded prices at each hub. We define the difference in price between any two given nodes as the implied cheapest cost of transporting gas between the two hubs.

Pipelines typically flow in only one direction. The direction of flow is a property of the path between two hubs.

Data

Raw Data

Natural Gas Intelligence (NGI), is one of the leading publishers of pricing data for physical natural gas. NGI also redistributes trade data associated with the ICE exchange.

We extract the most recent ICE day-ahead natural gas price report from the Natural Gas Intelligence (NGI) service website using the 'rvest' package:

```
# load the rvest package
library(rvest)
# set the NGI URL
urlString<-'http://www.naturalgasintel.com/ext/resources/Daily-GPI/ICE.htm'
# open the session
htmlSession <- html(urlString)
# define output directory
outputDirectory<-'C:/Users/DerekG/Documents/R/IS607/Project_5/'
# define output file name
outputFileNameCSV<-'dayAheadNatGasICE'
# define the .csv output file name
outputFileNameCsvFull<-paste0(outputDirectory,outputFileNameCSV,'.csv')
# define the .RData output file name
outputFileNameRFull<-paste0(outputDirectory,outputFileNameCSV,'.RData')
# fetch the ICE day-ahead natural gas price report data
table <- htmlSession %>% html() %>% html_nodes("table") %>% html_table()
a<-table[[1]]
# extract the trade and flow dates
```

```
tradeAndFlowDates<-a$X1[1]
# make the header
header<-c(a$X1[2],a$X2[2],a$X3[2],a$X4[2],a$X5[2],a$X6[2],a$X7[2])
# extract the columns
hub<-a$X1[3:length(a$X1)]
highPrice<-as.numeric(sub('[$]','',a$X2[3:length(a$X2)],perl=TRUE))
lowPrice<-as.numeric(sub('[$]','',a$X3[3:length(a$X3)]))
avgPrice<-as.numeric(sub('[$]','',a$X4[3:length(a$X4)]))
chg<-as.numeric(a$X5[3:length(a$X5)])
volume<-as.numeric(sub('[,]','',a$X6[3:length(a$X6)]))
numberOfTrades<-as.numeric(sub('[,]','',a$X7[3:length(a$X7)]))
# bind the columns together
rawData<-cbind(highPrice,lowPrice,avgPrice,chg,volume,numberOfTrades)
# create data frame
data<-data.frame(hub,rawData)
# write data to .csv file for upload to Neo4j
write.csv(data,file=outputFileNameCsvFull,row.names=FALSE)
# write data to .RData file for quick use in R
save(list=c('data'),file=outputFileNameRFull)
# add a header for data table
colnames(data)<-header
```

The following table displays pricing information on physical fixed-price trades on ICE for Next Day physical gas for the first 10 of the 119 reported hubs:

```
# create a table with sample data
knitr::kable(head(data,10),caption='Sample ICE day-ahead natural gas prices')
```

Table 1: Sample ICE day-ahead natural gas prices

HUB	HIGH	LOW	AVG	CHG	VOLUME	# TRDS
Algonquin Citygates	2.750	2.610	2.6713	0.8168	20000	5
Col Gas TCO	2.875	2.805	2.8547	0.0800	451100	83
Dominion-North	1.900	1.600	1.8567	0.3478	65900	17
Dominion-South	1.930	1.825	1.8786	0.3534	94300	34
Iroquois (into)	3.200	3.080	3.1405	0.0795	68300	18
Iroquois-Z2	3.250	3.090	3.1906	0.1073	45900	10
Leidy-Transco	1.810	1.730	1.7776	0.3129	48000	15
Millennium EP	1.680	1.500	1.6410	0.2050	36000	10
PNGTS	3.600	3.500	3.5500	0.2862	800	3
REX E-ANR	2.840	2.790	2.8200	0.0775	111400	16

We use on the average price at each hub to determine the implied cost of transporting gas between hubs below.

Preprocessing

First, we can determine each unique two-hub spread and compute the implied costs of transporting gas between any two hubs by taking the simple difference between the prices at the hubs:

```
# create structure to store nat gas spreads
natGasSpreads<-{}

for (hub1Index in seq_along(hub)){
  # extract the name of hub 1
  hub1<-hub[hub1Index]
  # extract the average price of hub 2
  hub1Price<-avgPrice[hub1Index]

  for (hub2Index in seq_along(hub)){

    if (hub1Index != hub2Index){
      # extract the name of hub 2
      hub2<-hub[hub2Index]
      # extract the average price for hub 2
      hub2Price<-avgPrice[hub2Index]
      spread<-hub1Price-hub2Price
      ratio<-hub1Price/hub2Price
      # store the hubs, prices, and spread
      natGasSpreads<-rbind(natGasSpreads,c(hub1,hub2,
        hub1Price,hub2Price,spread,ratio))
    }
  }
}
# create the spread data
spreadData<-data.frame(natGasSpreads)
# name the columns
colnames(spreadData)<-c('hub1','hub2','priceAtHub1','priceAtHub2','spread','ratio')
# count the number of spreads
nSpreads<-nrow(spreadData)
# define output file name
outputFileNameCSV<-'dayAheadNatGasICESpreads'
# define the .csv output file name
outputFileNameCsvFull<-paste0(outputDirectory,outputFileNameCSV,'.csv')
# write the .csv file
write.csv(spreadData,file=outputFileNameCsvFull,row.names=FALSE)
```

There are 14042 distinct natural gas spreads if we constrain the spread to the difference between just two nodes.

Next, we extract a subset of the spreads. The following table shows all of the hubs spread against the 'Henry' hub:

```
# find all of the spreads to the benchmark 'Henry' hub
henryIndex<-spreadData[,1]=='Henry'
# create a table with the spreads to the benchmark 'Henry' hub
knitr::kable(head(spreadData[henryIndex,],10),
  caption='Sample ICE day-ahead natural gas price spreads')
```

Table 2: Sample ICE day-ahead natural gas price spreads

	hub1	hub2	priceAtHub1	priceAtHub2	spread	ratio
3659	Henry	Algonquin Citygates	2.8463	2.6713	0.175	1.06551117433459

	hub1	hub2	priceAtHub1	priceAtHub2	spread	ratio
3660	Henry	Col Gas TCO	2.8463	2.8547	-0.00839999999999996	0.997057484148947
3661	Henry	Dominion-North	2.8463	1.8567	0.9896	1.5329886357516
3662	Henry	Dominion-South	2.8463	1.8786	0.9677	1.51511764079634
3663	Henry	Iroquois (into)	2.8463	3.1405	-0.2942	0.906320649578093
3664	Henry	Iroquois-Z2	2.8463	3.1906	-0.3443	0.892089262207735
3665	Henry	Leidy-Transco	2.8463	1.7776	1.0687	1.60120387038704
3666	Henry	Millennium EP	2.8463	1.641	1.2053	1.73449116392444
3667	Henry	PNGTS	2.8463	3.55	-0.7037	0.801774647887324
3668	Henry	REX E-ANR	2.8463	2.82	0.0263	1.00932624113475

Queries to Acquire and Manage Data

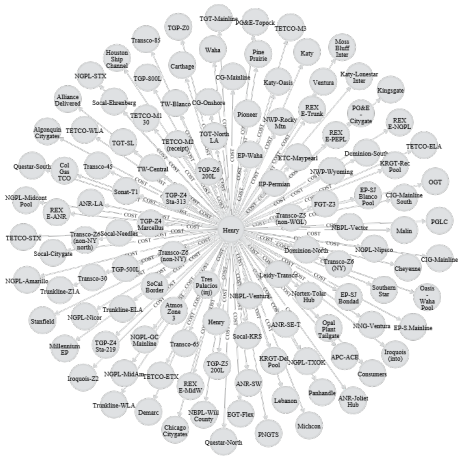
Iterate over the hubs create the nodes and relationships using the R package, RNeo4j.

```
# load the package
library(RNeo4j)
# set the connection parameters
graphURL<-'http://localhost:7474/db/data/'
userName<-'neo4j'
password<-'tgdnrx78'
# connect to the database
graph = startGraph(graphURL,username = userName,password = password)
# clear the database
clear(graph,input=FALSE)
# find the henry hub
henryIndex<-hub=="Henry"
# extract the name of hub 1
hub1<-hub[henryIndex]
# extract the average price of hub 2
hub1Price<-avgPrice[henryIndex]
# define the node for hub 1
hub1<-createNode(graph,'hub',hub=hub1,priceAtHub=hub1Price)

for (hub2Index in seq_along(hub)){

  if (hub1Index != hub2Index){
    # extract the name of hub 2
    hub2<-hub[hub2Index]
    # extract the average price for hub 2
    hub2Price<-avgPrice[hub2Index]
    spread<-hub1Price-hub2Price
    ratio<-hub1Price/hub2Price
    # define the node for hub 2
    hub2<-createNode(graph,'hub',hub=hub2,priceAtHub=hub2Price)
    # define the relationship with properties
    createRel(hub1, "COST", hub2, cost = spread)
  }
}
```

The graph database can be visualized as follows:



Queries to Access and Analyze Data

Query for the prices at the Waha and the TGT-SL hubs:

```
# query for the prices at the Waha and TGT-SL hubs
priceAtWahaTGT<-cypher(graph,"MATCH n WHERE n.hub IN {hubs} RETURN n.hub, n.priceAtHub",
       hubs = c('Waha','TGT-SL'))
knitr::kable(priceAtWahaTGT,
             caption='Prices at Waha and TGT-SL Hubs')
```

Table 3: Prices at Waha and TGT-SL Hubs

n.hub	n.priceAtHub
TGT-SL	2.8100
Waha	2.6995

Find the number of cost links between hubs (i.e., pipeline connections)

```
# query for the number of cost links between hubs
nSpreads<-cypher(graph,"match (n)-[:COST]-(m) RETURN n.hub, COUNT(m) as nSpreads")
# create a table with the number links between hubs
knitr::kable(nSpreads,
  caption='Number of Links Between Hubs')
```

Table 4: Number of Links Between Hubs

n.hub	nSpreads
NNG-Ventura	1

n.hub	nSpreads
TGP-Z4 Sta-219	1
Transco-Z6 (non-NY)	1
ANR-SE-T	1
PGLC	1
NWP-Wyoming	1
Iroquois (into)	1
ANR-LA	1
Kingsgate	1
REX E-MidW	1
CIG-Mainline South	1
Transco-30	1
NGPL-GC Mainline	1
NBPL-Ventura	1
REX E-NGPL	1
EP-SJ Bondad	1
Malin	1
Panhandle	1
ETC-Maypearl	1
OGT	1
REX E-Trunk	1
Dominion-South	1
ANR-SW	1
TGP-500L	1
Transco-Z6 (non-NY north)	1
EP-SJ Blanco Pool	1
PG&E-Topock	1
Henry	119
Transco-65	1
TGP-Z4 Marcellus	1
Pine Prairie	1
Stanfield	1
Ventura	1
TGP-Z5 200L	1
Transco-Z5 (non-WGL)	1
TGP-Z0	1
TETCO-STX	1
EP-Waha	1
TGT-North LA	1
KRGT-Rec Pool	1
CG-Onshore	1
Waha	1
NWP-Rocky Mtn	1
TW-Central	1
Socal-Citygate	1
Trunkline-Z1A	1
KRGT-Del Pool	1
TGT-Mainline	1
CIG-Mainline	1
Tres Palacios (inj)	1
EP-Permian	1
Questar-North	1
Transco-85	1

n.hub	nSpreads
EGT-Flex	1
TGP-Z6 200L	1
TGT-SL	1
Carthage	1
APC-ACE	1
Southern Star	1
TW-Blanco	1
NGPL-Amarillo	1
Socal-KRS	1
Trunkline-WLA	1
NBPL-Will County	1
NGPL-TXOK	1
Nortex-Tolar Hub	1
TETCO-WLA	1
Moss Bluff Inter	1
Trunkline-ELA	1
Pioneer	1
SoCal Border	1
ANR-Joliet Hub	1
Houston Ship Channel	1
Leidy-Transco	1
CG-Mainline	1
PG&E - Citygate	1
Col Gas TCO	1
Katy-Lonestar Inter	1
Iroquois-Z2	1
Chicago Citygates	1
TETCO-M2 (receipt)	1
NGPL-Nipsco	1
Questar-South	1
Socal-Needles	1
Oasis - Waha Pool	1
NGPL-MidAm	1
NBPL-Vector	1
EP-S.Mainline	1
TETCO-ELA	1
TGP-Z4 Sta-313	1
FGT-Z3	1
Socal-Ehrenberg	1
Michcon	1
TGP-800L	1
Lebanon	1
Algonquin Citygates	1
Atmos Zone 3	1
PNGTS	1
Sonat-T1	1
Katy-Oasis	1
REX E-PEPL	1
Transco-45	1
NGPL-STX	1
TETCO-M1 30	1
Opal Plant Tailgate	1

n.hub	nSpreads
TETCO-M3	1
Dominion-North	1
Consumers	1
Alliance Delivered	1
Transco-Z6 (NY)	1
NGPL-Nicor	1
TETCO-ETX	1
Demarc	1
REX E-ANR	1
Cheyenne	1
NGPL-Midcont Pool	1
Katy	1
Millennium EP	1

Query for the 10 most negative spreads:

```
# query for the 10 most negative spreads
worst5NegativeSpreads<-cypher(graph,
  "match (n)-[:COST]-(m) WHERE n.hub <> 'Henry' AND n.priceAtHub<m.priceAtHub RETURN n.hub,m.hub,(n.pri
# rename the columns
colnames(worst5NegativeSpreads)<-c('hub 1','hub 2','spread')
# create a table with the 5 most negative spreads
knitr::kable(worst5NegativeSpreads,
             caption='Excess Supply')
```

Table 5: Excess Supply

hub 1	hub 2	spread
TGP-Z4 Marcellus	Henry	-1.2529
Millennium EP	Henry	-1.2053
Leidy-Transco	Henry	-1.0687
Dominion-North	Henry	-0.9896
TETCO-M2 (receipt)	Henry	-0.9786
Dominion-South	Henry	-0.9677
TGP-Z4 Sta-313	Henry	-0.9264
TGP-Z4 Sta-219	Henry	-0.8468
TETCO-M3	Henry	-0.8330
Kingsgate	Henry	-0.2938

Query for the 10 most positive spreads:

```
# query for the 10 most positive spreads
best5PositiveSpreads<-cypher(graph,
  "match (n)-[:COST]-(m) WHERE n.hub <> 'Henry' AND n.priceAtHub>m.priceAtHub RETURN n.hub,m.hub,(n.pri
colnames(best5PositiveSpreads)<-c('hub 1','hub 2','spread')
# create a table with the 5 most positive spreads
knitr::kable(best5PositiveSpreads,
             caption='Excess Demand')
```

Table 6: Excess Demand

hub 1	hub 2	spread
PNGTS	Henry	0.7037
PG&E - Citygate	Henry	0.3987
Iroquois-Z2	Henry	0.3443
Iroquois (into)	Henry	0.2942
Consumers	Henry	0.1934
Michcon	Henry	0.1894
Transco-Z6 (NY)	Henry	0.1807
Transco-Z6 (non-NY)	Henry	0.1537
Transco-Z6 (non-NY north)	Henry	0.1472
Transco-Z5 (non-WGL)	Henry	0.1444

Conclusion

The graph database provides a very powerful approach to understanding the natural gas network.