



My boy Sir Isaac Newton is famous for a few laws he wrote about motion. His first such law on the topic says:

An object at rest will remain at rest unless acted on by an unbalanced force. An object in motion continues in motion with the same speed and in the same direction unless acted upon by an unbalanced force

Apparently back in his day motion was not only a big deal but it was also apparently out of control and needed to have some laws written to govern it. If I lived back then I would have been mad about apples falling on my head as well and probably would have come up with some pretty awesome laws of my own.

But the Qlik Dork is living in a day and age where 0's and 1's are falling out of the sky and totally are totally out of control. Not sure there is any governing body to prevent me from doing so therefore I figured it was time for me dictate a few Laws of Data:

Qlik Dork's First Law of Data - "Data at REST is stupid. Put that data into motion by visualizing it and your company will pick up momentum."

Qlik Dork's Second Law of Data - "Data at REST is expensive. Put that data into motion by visualizing it and the data will start paying for itself."

Qlik Dork's Third Law of Data - "Data at REST is useless. Unless you expect an 8'th day of the week to be added to the calendar I've got bad news for you ... your 'Someday' won't get here. So start making use of your data 'Today.'"

["Data at REST is stupid. Put that data into motion by visualizing it and your company will pick up momentum."](#)

[Click To Tweet](#)

BIGger DATA

[Big Data is kind of old news. How old? Even I jumped on the bandwagon all the way back in September of 2016.](#) In this post I'm not going to talk about Big Data instead as the heading suggests I'm going to write about BIGger DATA.

Marketing slides would say that Qlik has 10 points of integration with Cloudera. Woo-hoo. You can actually get the value out of all that Big Data with Qlik. But those of you reading this blog aren't novices. You already know that Qlik is a data ingesting beast and already knew that Big Data is simply another source of data. Qlik's Associative Model will gladly allow you to associative your 0's and 1's from cocktail napkins, spreadsheets, flat files, databases, EDW's and of course Big Data sources.

Kind of crazy that I can actually make light of that point, but let's face we already know that. Nothing new to look at there. Where I'm going today is beyond even that. I'm suggesting that the data Cloudera collects about your ingestion of Big Data is also a source of meaningful data which is now just sitting at rest but can and should be visualized so you can get value out of it as well. In other words **BIGger** DATA.

Don't laugh too hard but my use of the uppercase when using the word REST isn't accidental. It's intentional. What I'm driving at this post is using the Qlik REST data connector to take advantage of all of the data sitting there, at rest, about what you are doing with your Big Data. Sneaky huh??? That's how I roll.

Visualizing Data at REST

Let me lay the ground work for what I'm talking about. Cloudera provides a nice Cloudera Manager tool that you can utilize to see a lot of different things about your implementation. I can see what's going across the system.



The screenshot shows the Cloudera Manager web interface. At the top, there are navigation tabs for Cloudera, Hue, Hadoop, HBase, and Impala. Below this is a dark blue header with the text "cloudera MANAGER" and dropdown menus for "Clusters" and "Hosts". The main content area is titled "Home" and features a "Status" section with a red indicator for "All Health Issues" (1) and a yellow indicator for "Configuration" (6). Below this, a "Cloudera QuickStart" (CDH 5.10.0) is listed. A section titled "Parcels" contains a table of services:

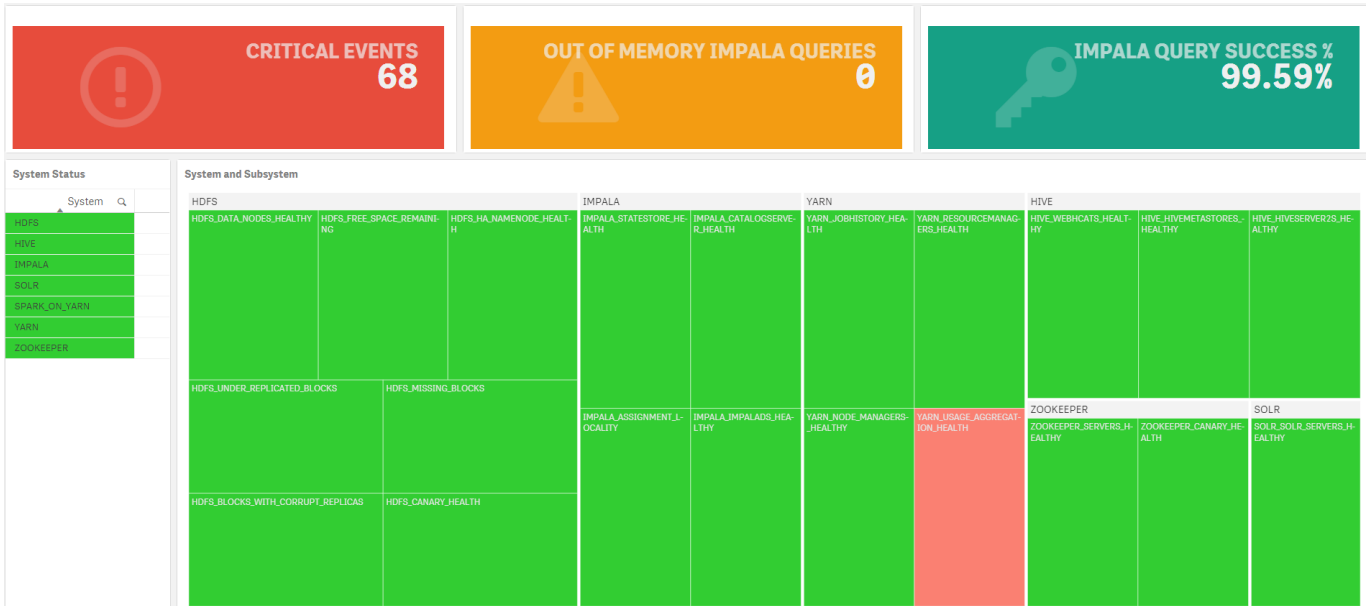
Hosts	✖ 1	
HBase		
HDFS	✖ 2	
Hive	✖ 2	
Hue		
Impala	ⓘ 1	
Key-Value Stor...		
Oozie		
Solr		
Spark		
Sqoop 1 Client		

I can also drill into different systems to see what's going on with them specifically.

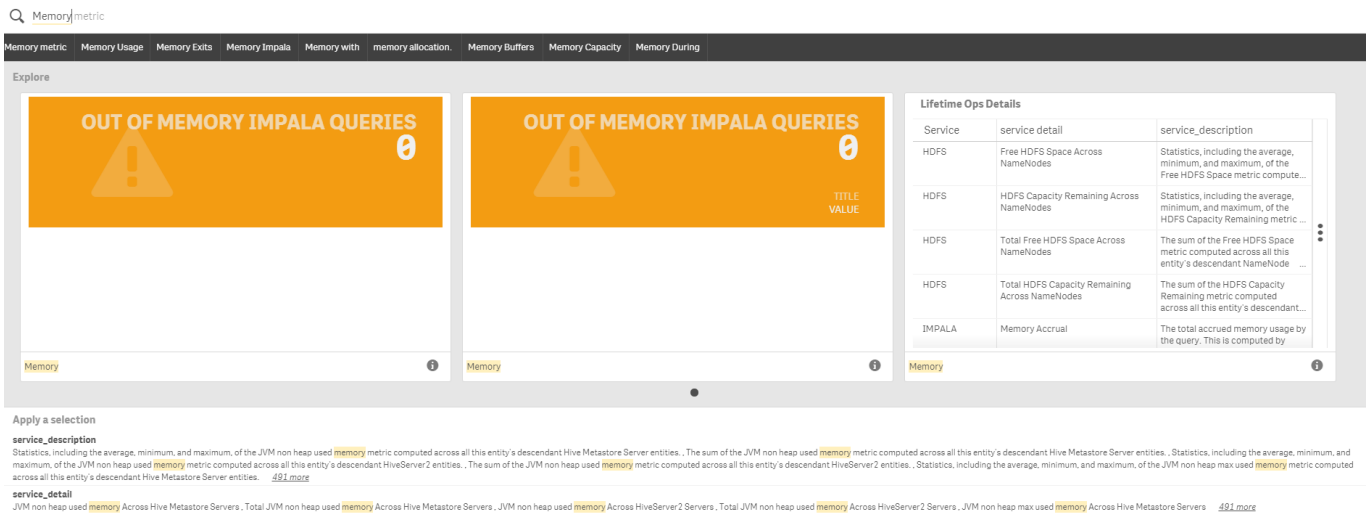


The screenshot shows the Cloudera Manager interface for HDFS monitoring. At the top, there's a navigation bar with various services like Cloudera, Hue, Hadoop, HBase, Impala, Spark, Solr, Oozie, and Cloudera Manager. Below that, the 'HDFS (Cloudera QuickStart)' section is active, with a sub-menu containing 'Status', 'Instances', 'Configuration', 'Commands', 'File Browser', 'Charts Library', 'Cache Statistics', and 'Audits'. The 'Status' tab is selected, showing an 'HDFS Summary' with a progress bar for 'Configured Capacity' at 14.7 GIB/54.5 GIB. Below this is a 'Health Tests' section with a 'Create Trigger' button and a 'Show 4 Good' indicator. The 'Status Summary' section lists components and their health: Balancer (1 None), DataNode (1 Good Health), NameNode (1 Good Health), SecondaryNameNode (1 Good Health), and Hosts (1 Good Health). The 'Health History' section shows a list of events with timestamps and descriptions, such as '7:47:16 AM' where 2 became good, and '7:47:10 AM' where 1 became bad, 1 became concerning, and 5 became good.

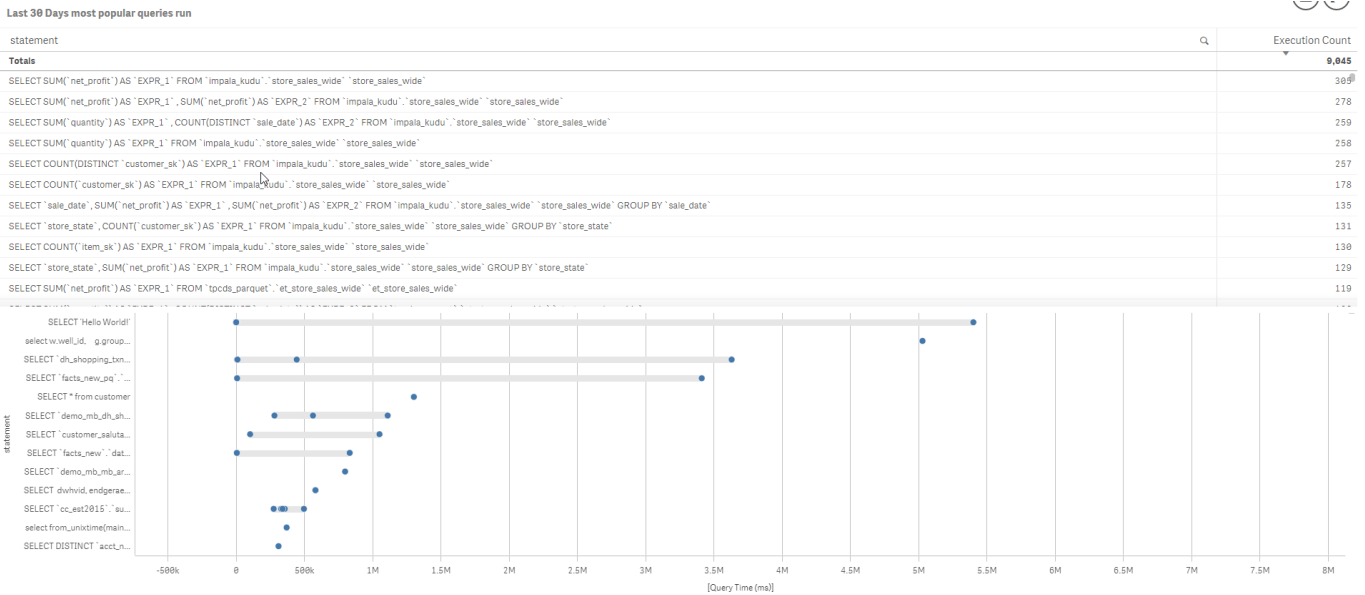
There is data behind those things. Meaningful. Important. Useful. Data. Visualizing all of that wonderful information inside Qlik provides you the ability to get an overall and subsystem visual as well:



No big deal you say? Happy to click around and hunt and peck to find everything are you? Well how would you find any issues in any of the systems that have to do with memory or see their history? Well in Qlik I would simply use the Smart Search feature to find anything that has to do with the term “memory” but that’s just me.



What about the metadata collected about the queries that are used to pull the Big Data? Doesn't have value? OF COURSE IT DOES so you know I want to visualize that as well. Not only can we show you which queries were fired and how many times they were fired, we can show a distribution plot of how long the queries took each time. Glad I was the guy who selected “Hello World” in 18 ms and not 5.5 minutes. Hate to be that guy.



More importantly I'd hate to be the team that is just leaving all this data lay at rest on your system. If you don't understand why... see my 3 Laws of Data above.

Using REST to get your data at rest about your Big Data usage

I'd like to begin this section by sharing that all of the real work in coding that you will see was done by my buddy, and Cloudera implementation stud David Frericks. Yes, there are others at Qlik that are bigger data dorks than I. And David is the guy the Qlik Dork goes to.

Did you even know that Cloudera had a fully supported set of REST API's that could be used to pull this wonderful data that is only resting as of now? Guessing the answer is no or it would be redundant to write the question. [You can check it all out right here.](#)

Let me walk you through a very simple REST API call. Let's say we want to get all of the Impala queries that were run on the system. There is a REST API for that.

```
http://your_VM_IP_here:7180/api/v15/clusters/Cloudera
QuickStart/services/impala/impalaQueries?from=2017-01-01&limit=1000&offset=0
```

We would implement that call using the Qlik Rest Connector like so. Notice that I've filled in my Cloudera system IP address and put my Cloudera system credentials in.

Create a connection - REST

Request

URL

Timeout

Method

Data options

Auto detect response type

Key generation strategy

Authentication

Use Windows authentication

User name

Name

That will then allow me to see what that REST API call will surface in terms of data.

REST
Impala Queries

Response type: Preload symbols count: Include HTTP response header

Tables: Fields:

queries	queryId	statement	queryT...	querySt...	startTime	endTime	rowsProdu...	user	detailsAva...	da
	ec447c49b4b4ac1da83c2708000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T17:03:33.797Z	2017-07-11T18:59:57.588Z		cloudera	True	default
	f74fa0ef70ebb3e993817838000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T17:28:11.562Z	2017-07-11T18:59:57.588Z		cloudera	True	default
	fb438dc24e52c19cdb0e69d0000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T17:26:19.901Z	2017-07-11T18:59:57.587Z		cloudera	True	default
	e49256be3f2d7cf33365d2e0000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T17:30:43.284Z	2017-07-11T18:59:57.586Z		cloudera	True	default
	3540f72989ee712edc9e1675000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T15:17:31.944Z	2017-07-11T18:59:57.586Z		cloudera	True	tpcds
	6f42278db306755cb1aa86bd000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T17:30:00.601Z	2017-07-11T18:59:57.586Z		cloudera	True	default
	374c21fb74b4bb0872db179d000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T15:13:43.009Z	2017-07-11T18:59:57.585Z		cloudera	True	tpcds
	8c487f089365ee676f824b3f0000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T15:20:50.376Z	2017-07-11T18:59:57.584Z		cloudera	True	tpcds
	3479f310cabdb6a1116cb190000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T14:10:43.416Z	2017-07-11T18:59:57.583Z		cloudera	True	default
	21467c86110d0842e269b88500	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T15:22:20.456Z	2017-07-11T18:59:57.583Z		cloudera	True	tpcds
	4b407b964f0f49229f48822c0000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T14:29:35.246Z	2017-07-11T18:59:57.583Z		cloudera	True	default
	654cad13e76ec38eae981757000	SELECT 'Hello World!'	QUERY	EXCEPTION	2017-07-11T13:59:57.690Z	2017-07-11T18:59:57.582Z		cloudera	True	default
	c4a44d2b4d6bcf7.253bd080000	SELECT 'tpcdstosresales', custom	QUERY	FINISHED	2017-07-11T18:11:38.071Z	2017-07-11T18:11:39.457Z			True	default
	5545b1adccc6ff17:e599d0920000	DESCRIBE 'tpcds'; tpcdsstoreale DDL	QUERY	FINISHED	2017-07-11T18:11:34.344Z	2017-07-11T18:11:38.066Z			True	default

I say heck yeah, go get that for me ... and boom it loads that data. It's important to understand what's happening behind the scenes at this point. Script is written that will execute and process that call.

```

92      (SELECT
93          "thread_storage_wait_time",
94          "admission_result",
95          "session_id",
96          "planning_wait_time",
97          "oom",
98          "stats_missing",
99          "memory_spilled",
100         "network_address",
101         "admission_wait",
102         "file_formats",
103         "planning_wait_time_percentage",
104         "client_fetch_wait_time",
105         "client_fetch_wait_time_percentage",
106         "pool",
107         "memory_per_node_peak_node",
108         "memory_per_node_peak",
109         "connected_user",
110         "session_type",
111         "thread_network_receive_wait_time",
112         "thread_network_send_wait_time",
113         "impala_version",
114         "estimated_per_node_peak_memory",
115         "query_status",
116         "hdfs_bytes_read_remote_percentage",
117         "hdfs_bytes_read_from_cache",
118         "hdfs_bytes_read_local_percentage",
119         "hdfs_scanner_average_bytes_read_per_second",
120         "hdfs_bytes_read_from_cache_percentage",
121         "bytes_streamed",
122         "memory_aggregate_peak",
123         "hdfs_average_scan_range",
124         "memory_accrual",
125         "hdfs_bytes_read_local",
126         "hdfs_bytes_read_short_circuit",
127         "hdfs_bytes_read_short_circuit_percentage",
128         "hdfs_bytes_read_remote",
129         "hdfs_bytes_read",
130         "ddl_type",
131         "__FK_attributes"
132     FROM "attributes" FK "__FK_attributes")
133 FROM JSON (wrap off) "queries" PK "__KEY_queries";
134
135 [Coordinator]:
    
```

Phooey you say? TMI you say?

No my friends here is where the rubber meets the road ... with script you can overcome any

kind of REST API barriers that may exist with Cloudera or with others ... say those pesky times when they implement a MAXIMUM number of rows being returned.

Qlik can loop and get all of the rows for you. But wait there's more.

What about those times where you have 1 REST API call that returns some key information and you then need to go get the details using another REST API call? Or when you have multiple systems implemented and want to get all of the information about the systems, the calls that were fired, their history, their blah-blah-blah-blah. Yeah Qlik let's you do all of that because of it's ability to do the ETL on the fly.

When I started the section with a shout out to my friend David Frericks it wasn't in jest. Dude provided a serious baseline of work in making the Cloudera REST API's fly information and has produced some great work.

Follow up

"Hope isn't a Strategy." I read that recently in a book called "Wintality" by Baylor Barbee and that it was an awesome quote to summarize what I see happening right now in most people when it comes to big data. Lot's of HOPE. They are hoping that the time will come when they will learn, understand, begin diving into "Big Data." I'm assuming that you are reading this post because you are the kind that is more action oriented and you have or are formulating you strategy for visualizing your Big Data and hopefully now your Big'ger' Data as I've laid it out for you.

Be sure to check out <http://cloudera.qlik.com/> to see some great examples of how Qlik brings Big Data to life.

Hungry for even more? Want to see a phenomenal example of how Qlik can call SOLR Search on the fly using the REST API, ingest the data and produce a web application using the Qlik Sense Visualization API calls then check this out. <http://cloudera.qlik.com:3000>