White Paper

GALACTIC EXCHANGE™

Simplifying the Challenges of Big Data

The ClusterGX™ Architecture & Integrated AppStore

Galactic Exchange, Inc.

Author: Robert Mustarde

Date of Last Revision: 09.26.2016

Table of Contents

Introduction

There is little dispute across the industry that the world of Big Data is a complex one. There are no internationally accepted standards bodies that spend years crafting definitions and then passing final specifications for the way big data technology should work and inter-operate. Instead we have an industry whose whole is actually made up of an endlessly expanding primeval soup of open source solutions. The software core around which many of these add-on tools rotate – Apache Hadoop and Apache Spark - are also open source, with no central body owning the on-going development of the entire ensemble.

As a direct result, for the uninitiated, the learning curve is immense and for many a step too far. The net result has been that as the market for Big Data has rapidly expanded, the big winners are clearly defined. They are companies that can hire their own teams of skilled big data professionals, or who have the financial muscle to outsource their requirements to other people to do the work for them. No surprise then that every big data forecast, past and present - when broken down into hardware, software and services – shows the services side as the biggest chunk of what is a very large pie.

At Galactic Exchange we believe it does not have to be this way. Whilst we cannot change the underlying complex nature of multiple open source tools – what we can do is create an abstraction layer which allows the user to dial-down (or dial-up) the complexity to fit their needs and capabilities. We believe customers should be free to pick from any of the multitude of applications (analytics, machine learning and other “big data” applications) without having to fear the underlying complexity of the infrastructure that supports those applications. There are multiple elements to the overall process of extracting data, transforming it and ultimately storing it and running it through the selected preferred business application. As a result at

Galactic Exchange we are on a path to simplify all these elements. Our starting point is our clustering platform, ClusterGX™.

Clustering & Hadoop

At a macro level, the concept of clustering and Hadoop is pretty straightforward and can be explained in 3 words – divide and conquer. That’s pretty much where the simplicity ends however.

The problem with Big Data is – well, the volume of data….it’s Big! These days we are able to gather and store, in perpetuity, information on nearly every variable affecting our businesses. From web site clicks, to client invoices, product orders and deliveries to employee records, phone data to network traffic and alarms – the list is endless and it is only getting bigger. The technology now exists which allows us to analyze these large volumes of data and see patterns,

which we were never able to see before. In fact today we can find connections and patterns in data sets that were once totally de-coupled from each other.

Benjamin Franklin explains chaos theory very succinctly (although perhaps not knowingly!) in his poem – **For The Want of a Nail** - below.

For the want of a nail the shoe was lost,

For the want of a shoe the horse was lost,

For the want of a horse the rider was lost,

For the want of a rider the battle was lost,

For the want of a battle the kingdom was lost,

And all for the want of a horseshoe-nail.

Benjamin Franklin

What Big Data promises is the ability to make that connection between the nail and the kingdom being lost. The corollary here is that fine tuning or pro-active action related to details hidden within business data can make very significant positive downstream impacts. You can spot that missing nail early!

Clustering

But how can a single computer analyze such huge volumes of data in a reasonable amount of time? The short answer is, it cannot. Even with Moore's law promising a doubling of computer power roughly every two years, the quantum jump to processing such huge volumes of data is a step too far for your typical humble x86 server. If we couple that quantum jump in data volume with the laws of physics which indicates that Moore's law, in the traditional sense, is reaching the end of the road anyway - then we have a compute power conundrum.

But that conundrum is solved by the concept of clustering. Instead of having one server do the work – have two or more servers do the work – divide and conquer. Server clustering is in many respects the proof that Moore's Law will in fact continue unabated, just in a different manner. Instead of the compute power residing inside one server, simply consider the entire cluster as one giant "virtual" server with unlimited expansion capacity.

Hadoop

Hadoop is designed to run on top of a physical server cluster to facilitate the processing of large volumes of data. Think of it almost like a multi-server operating system, allowing the separate pieces to operate as one.

At a macro level, Hadoop is essentially responsible for 2 discrete functions. First, it takes the data sets that need processing (the "big" data) and splits it up into chunks which it allocates across the drives of the various server nodes that make up the cluster. This is called the Hadoop Distributed File System or HDFS for short. HDFS is designed with redundancy in mind so that if any server node fails, additional copies of the data chunks are also stored on other nodes. Net-net the processing can continue unabated.

The second function that Hadoop facilitates is something called MapReduce. This is the concept of a master node (a dedicated server in the cluster) sending out specific tasks to each slave node to execute (Map) specific tasks on specific data and then consolidate the responses from each node after the processing has occurred (Reduce). MapReduce works by each slave server pulling data off its local HDFS files stored on disk, processing the data and then sending out the results. The nature of the MapReduce architecture means it is somewhat high latency and best used for batch processing where data volumes are large, the data maybe mostly static (i.e. not changing in real-time) and speed it not of the essence.

As the big data world has developed the need for faster processing has arisen and alternate technologies to MapReduce have been developed which facilitate processing data stored in memory rather than on disk and also to be able to cater for streaming data that is hitting the cluster in real-time. The most well know and successful of these technologies is called Spark, also an Apache project. Spark is often considered competitive to Hadoop but in reality the two technologies work well together and should be considered complimentary. Spark can utilize the data stored in HDFS and can run in parallel to MapReduce applications running on the same cluster.

Whilst the principles and benefits of clustering and Hadoop are relatively simple to explain at a macro level – like most things in life, the devil is in the details. And what details they are. With more than 50 big data-centric open source projects all offering some additional bell and whistle to make the "whole" more efficient, the reality is that customers with little to no experience are simply over-whelmed.

The process of deploying a Hadoop Cluster alone can take even a skilled and experienced engineer a day or more to perform. For the inexperienced you can assume many days or even weeks, with many failed attempts along the way. This does not take into consideration the workload of adding some of those more complex bolt-on open source big-data tools. Nor does it deal with the complexity of actually deploying vendor applications (analytics, machine-

learning etc.) across your Hadoop cluster and tuning the numerous Hadoop attributes accordingly

A Better Way – Big Data Clustering with ClusterGX™

At Galactic Exchange our vision is to create an abstraction layer between the ultra-complex big data open source world and the day to day environment and skills of a typical IT department.

ClusterGX™ is the first element in that abstraction layer. This free to download software can be installed on Linux or Bare Metal servers (and even on Windows and Mac desktop devices). It can be deployed on-premise or on the cloud provider of your choice. Once installed it will auto-generate a container virtualized Hadoop/Spark cluster (with Apache Hadoop, Cloudera Hadoop or other commercially available distributions) - all in just a few minutes and with zero required knowledge of clustering, virtualization, containers, Hadoop or Spark. In a nutshell – if you are capable of installing an app like Skype or Spotify on your PC, then you can deploy a Hadoop/Spark cluster in minutes, no experience needed.

Inside ClusterGX™ we also support an increasing number of the other open source tools and frameworks. Now it becomes very easy to build – with a single click here or there - just the right open-source infrastructure to optimally support the various application(s) you wish to run.

The Power of Containers

ClusterGX™ uses container based virtualization, and specifically Docker containers at this time, although adding different container architectures such as LXC would be straight forward.

Containers provide incredible flexibility when it comes to virtualization, with the ability to spin up and spin down containers in seconds across a cluster. Unlike Virtual Machines, each of which creates a significant resource overhead on its host machine, containers are lightweight and nimble with virtually zero host overhead. Containers can be used to provide separation between application instances, separation between virtual client clusters across a physical cluster and separation of the underlying big data infrastructure associated with a specific application. In other words, containers make it really easy to run multiple Hadoop/spark frameworks (think “different versions of Hadoop”) across the same physical infrastructure, apply them to a chosen application, and spin them up and down quickly.

When ClusterGX™ is deployed, there is no server pre-preparation required to support containers. Installation of the entire Docker container architecture is handled automatically as part of the automated ClusterGX™ deployment.

The ClusterGX™ Container-based Architecture

Consider a deployment of ClusterGX™ with Cloudera CDH as the chosen Hadoop distribution. Assume we are running a business application (we can call that "Application-X") on this Cloudera cluster.

Everything is run inside Docker containers across the cluster. There are three types of containers:

1. Cloudera Master container:

Runs all Hadoop master services, including HDFS NameNode, YARN Resource Manager, Hive Metastore, Impala State-store, Impala Catalog, Spark-Master, Zookeper and others.

2. Cloudera Slave container:

Runs all Cloudera slave node services such as the HDFS datanode, YARN Task Tracker, Impala Server, Kafka Server and others.

3. Application container:

Runs the selected analytics/machine-learning or other "big data" application software (in this example it runs Application-X)

Master Nodes in the Cloud

By default, and as part of the immense simplification of deployment that ClusterGX™ provides, the Cloudera Master container for each cluster runs in the Galactic Exchange Cloud service*. The unique master node for each customer cluster is automatically spun up upon initial account creation.

The Cloudera Slave containers either run on the customer's machines on-premise or on cloud instances from their preferred cloud provider. Each slave container node connects back to the master over a secure encrypted VPN connection when the user account credentials are applied during initial set-up. These will then form a cluster of slave nodes where the actual data-processing will happen.

The Application-X container will run on a customer on-prem/cloud slave machine that the user selects during the installation (note: some applications will actually run distributed across the cluster, in which case we would show an Application-X container on each slave node).

whilst the **Master Node runs on the Galactic Exchange cloud service by default, it is possible to run this service on the customer's premise or an instance in their own cloud service.*

Secure VPN Connectivity

There is a secure OpenVPN connection between the Cloudera Slave and Application-X containers running on the customer premise/cloud, and the Galactic Exchange Cloud VPN Gateway. After a remote slave container is connected to the Galactic Exchange Cloud VPN Gateway, it can connect securely to Cloudera Master container running in the Galactic Exchange Cloud service.

All containers see each other as belonging to the same network. All VPN set-up is fully automated with no user configuration required.

Figure-1 shows a macro-level view of a Cloudera CDH cluster deployment of "n" slave nodes connected to the Galactic Exchange master node cloud service.

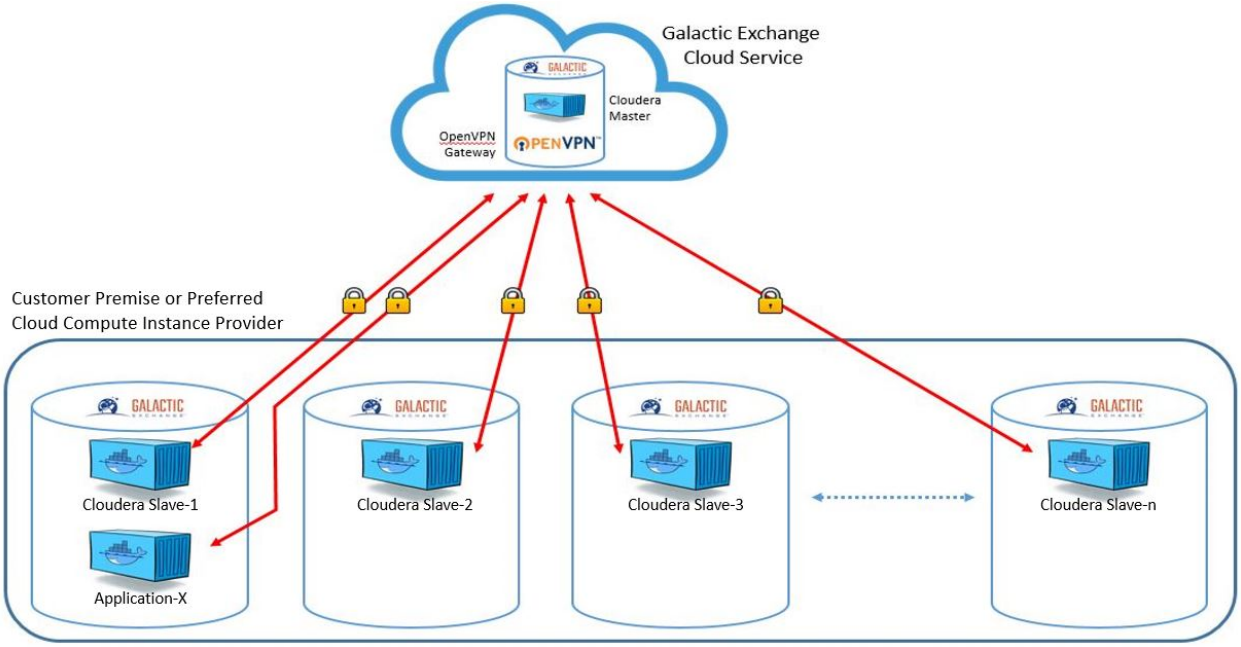

Figure-1

Simplified ClusterGX™ Network Addressing

In the example shown, the Cloudera Slave and Application-X containers acquire local IP addresses on the customer network, either statically or using DHCP. ClusterGX™ DNS automatically provides resolution of these addresses to host names. Browsers and other tools on the customer network can then easily communicate with the containers as local hosts.

In addition, the Cloudera Slave and Application-X containers acquire global IP addresses on the ClusterGX™ secure VPN network, so that they can talk to the ClusterGX™ Master. Therefore, in this scheme each container acquires two addresses, one global and one local as shown in Figure-2

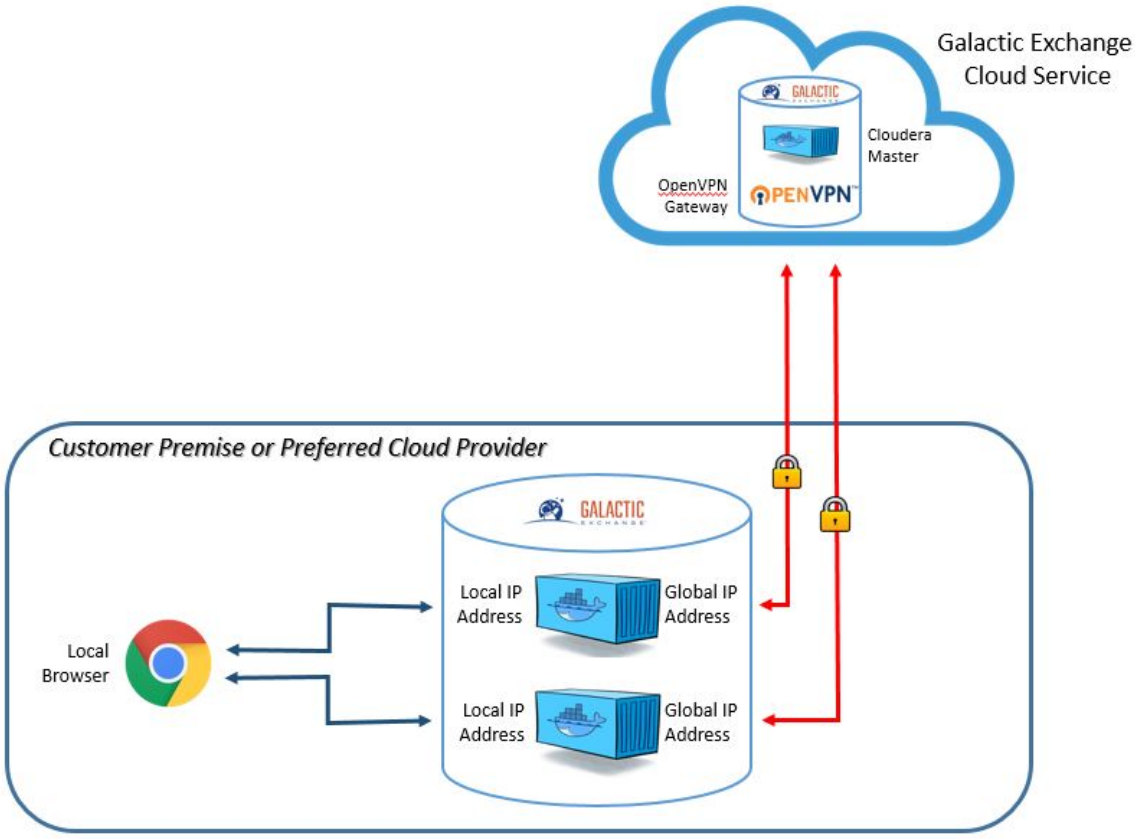

Figure-2

Applications and the ClusterGX™ Integrated AppStore

Once ClusterGX™ is installed on several machines, a Cloudera CDH cluster is established consisting of a master node running in the Galactic Exchange cloud, and several slaves nodes, each running in a container inside each physical machine. At this point the user can manually load any chosen application to take advantage of the Cloudera CDH cluster.

Loading applications manually can still be a laborious process for the inexperienced and will require configuring the Hadoop settings per the application vendors recommended preferred default settings. Deploying enterprise class applications in this way can take several hours even for experienced engineers.

With ClusterGX™ the user has an easier alternative for certain applications. Galactic Exchange has pre-integrated certain applications within an embedded AppStore. The AppStore is accessible via the ClusterGX™ user interface on the left main menu. Clicking on the AppStore menu option will highlight a selection of pre-integrated Application options.

When an AppStore application is selected, the user is prompted to select the ClusterGX™ slave machine where the application container is to be installed. The application container image is

then automatically downloaded from the internet. At the same time, a script is executed on the Cloudera Master running in the Galactic Exchange cloud, to set the relevant Cloudera configuration required by the selected application, such as creation of HDFS directories and Kafka topics.

All necessary information about the Hadoop cluster, such as IP addresses of the master and slaves, is automatically injected into the Application image when it is executed for the first time. This is done by creating templates out of the Application config. files and processing templates during the initial execution of the Application container.

From a user perspective – the only important point is that the entire Application is launched with a single click, and an installation that might otherwise take hours or longer, is completed automatically within a few minutes.

The ClusterGX™ Unified Command Center

ClusterGX™ automatically displays the Web UI of all installed Applications by integrating them into the ClusterGX™ UI framework. In this way, the user sees the Application(s) Web UI alongside the Hadoop Web UIs and Hue. The Unified Command Center ensures that there is always a single place to go to access all management consoles, including every Application, Hadoop/Spark and ClusterGX™ itself. The user has the option of switching to “native” full screen mode for any installed Application at any time.

Conclusion:

The open source tools that are the foundation of the big data industry were developed by software engineers for use by software engineers. This is reflected in the immense complexity that many experience when they try and immerse themselves into the technology. The success that the Hadoop ecosystem has demonstrated over the last decade is a triumph of value over complexity – businesses get considerable value, if they can just unlock their data.

But success comes at a price. Either you buy in skilled engineers or you outsource to skilled engineers. As a result, only those with deep pockets get to play and most of those feel they pay far too much.

To ensure the broadest possible adoption, any technology eventually has to appeal to and be useable by the mainstream population. By adding an abstraction layer to the complexity of the big data open source software soup, Galactic Exchange is a pioneer in ensuring that all businesses will have the power to unlock the value inside their data.

Thank you for reading

For more information on Galactic Exchange and ClusterGX™ please contact:

rob@galacticexchange.io

Follow Galactic Exchange:

Twitter @GetGalactic

LinkedIn

Tel: (415) 767 1007

www.galacticexchange.io