Towards Building an Analytics Platform in the Cloud

Valentina Salapura
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
salapura@us.ibm.com

Kirk A Beaty
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
kirkbeaty@us.ibm.com

Alan Bivens
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
jbivens@us.ibm.com

Minkyong Kim
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
minkyong@us.ibm.com

Min Li
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
minli@us.ibm.com

ABSTRACT
Recently enterprises have been able to leverage two revolutionary new tools for gaining a competitive advantage for their business – cloud computing and analytic applications. Cloud computing unburdens them from running and maintaining their compute resources, whereas analytic applications comb through their big data to provide new insights for a competitive advantage in the market. Analytic applications are carefully tailored to their target problems. While there is a lot of work published on both the mechanics of cloud computing as well as analytic methods for distilling insights from a variety of data, there is little work available about the cloud influence on the analytics platforms which aim at lowering the barrier for the creation, deployment, scaling and maintenance of next generation analytic workloads. This paper discusses the challenges we are facing today in order to provide an analytics platform to reduce cost and increase performance of analytics applications in the cloud computing environment.

Keywords
Cloud computing, analytics application, analytics platform, agile cluster provisioning, cluster sizing, MapReduce, Spark, in memory computing.

1. INTRODUCTION
Cloud computing is a new compute platform that offers agility, elasticity and cost savings. The main attributes of cloud computing are scalable, on-demand computing resources delivered over the network, and pay-per-use pricing. This offers flexibility to end user in exploiting as many resources as needed at any point in time without investing in huge infrastructure and management costs.

The enterprises using proprietary IT structure need to manage and maintain these machines, consuming a significant amount of their available IT budget (can be up to 80%), just to keep status quo. Many enterprises are moving from the proprietary IT structure towards pay-as-you-go model computing. They are opting for commodity hardware available in public or private clouds, being operated by a third party.

The utility business model frees up their IT budget allowing corporations the flexibility to scale up their operations as they need it, and reducing their operational risks. This offers flexibility in using as few or as many IT resources as needed at any point in time. Thus, users do not need to predict future resources they might need, and to commit to hardware capital investments in advance. This is especially advantageous for start-ups, and small and medium businesses which might otherwise not be able to afford the IT infrastructure they need to support their growing business. At the same time, redirecting capital investment from IT infrastructure to the core business is attractive even for large and financially strong businesses.

The term analytics is used for mathematical or scientific methods that discover new insights based on data, which is typically unstructured. The amount of unstructured data that can be mined to generate business value is exploding, reaching as much as exabytes daily. Usage of analytics to provide new insights out of collected big data in order to make better business decisions gives companies a competitive advantage.

Insights given by analytics evolved from analyzing engineering-based processes, such as product design and manufacturing, into optimization of logistic processes such as supply chain operations, and to human centric processes such as workforce management. Analytics is becoming an important part in the decision making process of enterprises, and the competitive advantage it gives them drives increasing demand.

The industry has seen analytic workloads emerging in many different forms. We are bringing here a taxonomy to capture their nature:

- **Passive Analytics:** Analytic workloads are designed and deployed on top of a dataset which is acquired and

CF'15, May 18 - 21, 2015, Ischia, Italy
Copyright 2015 ACM 978-1-4503-3358-0/15/05…$15.00
http://dx.doi.org/10.1145/2742854.2747279

maintained for a primary traditional workload. An example of a passive analytic workload would be analytics deployed on transactional bank data in order to determine usage patterns which may point to the likelihood of a client's interest in additional financial products. In this example, the primary reason for the dataset is the transactional workload of the bank, but the analytic results may provide insights which could be used in direct marketing efforts.

- **Decision support and business intelligence:** This type of analytic workloads usually answers critical business questions providing summarized facts or deep analytical insights such as revenue forecasting, predictive modeling, loyalty / churn analysis
- **Operational Efficiency Analytics:** The industry currently have the ability to monitor, collect, and store more data about our operations than ever before. Many process and operations owners have designed and built operational efficiency analytic workloads to model their processes and determine factors for optimization and troubleshooting.
- **Analytic Solutions:** End-to-end analytics solutions attempt to solve a mathematic or scientific problem using data collected for the purpose of solving the problem. One example of these analytics applications is a smarter planet solution which combines data from sensors (e.g., windmills) and from several mathematical models (e.g., weather forecast and an energy consumption model) to achieve optimal control and management of resources such as a power plant.

The elasticity and agility of cloud infrastructure, its flexibility to provide virtually unlimited resources in the cloud when needed at low cost has a potential to radically change how analytics applications on big data are built. Analytics application developers are focusing on the problem they are solving, and not on building analytics platform in the cloud in order to reduce the amount of programming needed for the development of analytics applications.

In this paper, we look at the trends and discuss the challenges we are facing today in order to provide an efficient platform for analytics developers. An analytics platform would reduce complexity involved in developing a new analytics application, would reduce amount of coding needed and with it their cost and time needed for development, and would increase performance of analytics applications in cloud computing environments, enabling real time analytics. We look into the existing analytics algorithms and compute methods, and into novel trends and compute methods in the cloud environment. In order to provide an analytics platform, we look into what analytics applications need, based on a case study of a real-world end-to-end analytics solution. Finally, we discuss how such an analytics platform might benefit analytics applications and transform how they are developed.

2. SCIENTIFIC AND BUSINESS ANALYTICAL WORKLOAD

Big data and cloud computing created a major shift in computing, moving the focus of computing from scientific workloads and business analytics into new born-in-the-cloud applications and compute models.

Examples of scientific applications are weather modeling, nuclear

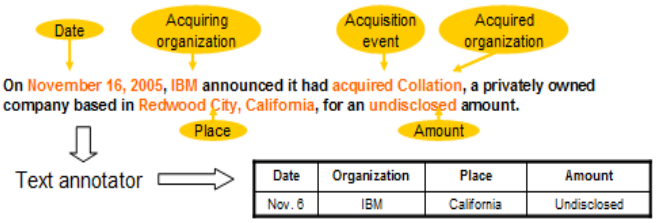

Figure 1. Acquisition annotator converts input unstructured text into structured data.

test simulations, wave propagation, or molecular dynamic simulations. These applications are divided into tasks manageable to be performed on a single compute node, and are processed typically on a supercomputer. Supercomputers are systems with massive numbers of processors and with fast interconnect. Supercomputer users carefully split the overall problem into tasks manageable to perform on a single compute node. For example, for weather modeling, each compute node will process only data related to a small area, and will mostly communicate with the compute nodes processing its neighboring areas. Data exchange between nodes is typically performed by using MPI (Message Passing Interface) programming model.

Compute problems are typically mapped to hardware nodes in order to minimize physical distance between the nodes (to reduce the number of hops) and to take advantage of nearest neighbor network topology. Such example applications and supercomputers with multi-dimensional torus topology are found in Blue Gene supercomputer systems [6].

A different type of application driving a need for compute systems is business analytics and big data processing. Business analytics applications allow processing and mining unstructured data in order to discover meaningful patterns in data so as to extract business value. Data can be in the form of unstructured text, such as in blogs and other social media, e-mails and documents, stored in variable-length descriptive formats such as XML, images, voice, video, and in various other forms.

Business applications transitioned from automating processes and their reporting to discovery of information, prediction, prescription and decision support. Modern business analytics applications analyze significant amounts of unstructured data in conjunction with past business performance to gain new insight and improve business outcomes. New insights and understanding of business performance is based on past data, statistical and quantitative analysis, and predictive modeling. The extracted insights are used to drive future business decisions.

Analytics relies on the simultaneous application of statistics, computer programming and operations research to quantify performance. Business analytics applications employ different methods and algorithms, such as statistic methods, simulation, optimization, data mining, machine learning, artificial intelligence, and cognitive computing. An example business analytics application is use of data analysis by banks or telecommunication companies to differentiate customers based on their credit risk, or on their usage of mobile networks.

Many business analytics applications use text analytics to process unstructured input data. The data ingress modules need to parse unstructured data for a wide range of data formats and encryption

methods. Additional requirements may be posed on data ingress to process it at the speed of data feeds to enable real-time analytics, for example for sentiment analysis based on twitter feeds for real-time advertising.

Acquisitions Annotator [14] is an example application which uses a set of Java libraries that provide natural language processing features such as language identification, tokenization, relationship extraction, and semantic analysis. Internally, it relies on FSM (finite state machines) based algorithms and techniques that are similar to those used in regex (regular expressions) matching. An annotator is a text analyzer that takes unstructured text and converts it to annotated or structured data. The goal is to detect items of interest in the text analyzed, ranging from simple patterns like e-mail addresses and phone numbers to complex relationships like one company acquiring another. Figure 1 illustrates the typical operation of an annotator. Our prior research [21] identified that the application spends approximately 50% of its time in the FSM-based code.

The FSM-based algorithms are very different from typical high performance scientific applications. In a typical scientific application, the same calculation is performed on large data sets arranged in arrays. Once operation is performed certain number of times, the calculation is completed. The number of operations to perform depends typically on the number of data elements. Data values and their indexes are separated from the control flow information.

Unlike in scientific applications, text processing and FSM based processing applications change their flow depending on the current input. Input data is consumed, character by character, and depending on the character read, the action is determined. For example, depending on the ingress character, a different word can be identified, and a different flow is determined. Here, data values are used for both data calculation and for determining the control flow. FSM codes exhibit a similar memory access pattern as a pointer chasing kernel, and are difficult to optimize.

Scientific applications pose demands on computing resources which are not a good match for cloud computing environment. An example is deterministic placement of workloads in a data center. The cloud compute environments are frequently virtualized, and placement of each individual virtual machine (VM) is not pre-defined. Instead, placement of a VM is determined at the time of provisioning to balance workload on physical servers in a data center. VM placement and variable network traffic in a data center cannot guarantee network latency.

Instead, cloud computing is well suited for shared-nothing highly parallel applications. Low cost and easy access to computing infrastructure of a cloud influenced wide usage of distributed algorithms for processing large data sets in a massively parallel manner, which are described in more detail in the next chapter.

3. PROGRAMMING MODELS FOR CLOUD

Since Google published the paper on MapReduce in 2004 [7] and the Apache Hadoop project officially started in 2006 [2], MapReduce programming paradigm became a huge success and Hadoop has been used widely. The success could be attributed to many of its outstanding capabilities such as ease of programming, scalability and fault tolerance.

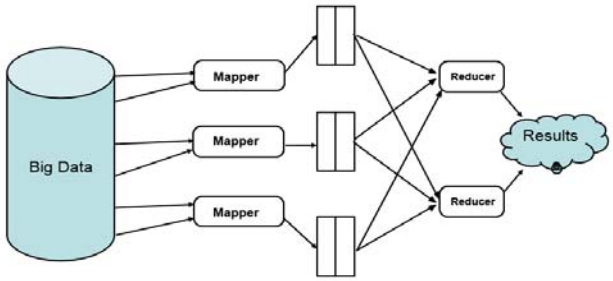

Figure 2. Illustration of a MapReduce application.

Among many reasons, the most important reason for MapReduce to be widely adapted would be ease of programming. In Message Passing Interface (MPI) programming, the burden of message passing between processes was laid on programmers. The key functions in the MPI library are *send* and *receive* operations, and programmers need to explicitly specify the parties involved in the communication. MapReduce framework completely eliminated this burden from programmers. Instead, the framework automatically distributes the tasks. This allows programmers to simply specify the computation without worrying about the task distribution and message passing. Seeing this advantage, there have been some efforts on supporting MapReduce operations on top of MPI to make use of MapReduce programming model while leveraging the existing high performance infrastructures [19]. These attempts had a limited success, and MapReduce blossomed in cloud.

The emergence of cloud propelled the success of MapReduce. Note that Amazon officially launched EC2 in 2006. The cloud allowed programmers to host MapReduce workload at a low cost. MapReduce frameworks take advantage of cloud's capability to provide scalability and fault tolerance. For example, MapReduce's capability to restart the failed tasks individually can be easily supported by unlimited and on-demand resources that cloud can provide.

Another reason for the success of MapReduce would be its wide applicability. Although the MapReduce programming model started as a distributed way to count words from a large volume of data, MapReduce turned out to be applicable for various problems. Many of business analytics problems, especially those that aim to extract useful information from big data, were able to use the MapReduce programming model, as illustrated in Figure 2. For example, telephone companies collect call logs. Using these logs, they want to infer various user information: home/work location, commute path/time, nationality, languages, etc. These profiles lead Telco to do better resource management, infrastructure planning, value-added services, and advertisement. The main challenge stems from the fact that the volume of logs for each user quickly adds up to be terabytes of data over time and becomes costly to perform analytics. MapReduce has become the most successful programming model used for these kinds of business analytics for the past decade.

While MapReduce provides a powerful way to perform a batch job against big data, it is not suitable for processing stream data.

Most enterprise systems run monitoring agents 24-7 and generate an enormous amount of data. As the number of users increases, the volume of data produced even by people, let alone systems and sensors, became huge; the log data that Telco industry collects and Twitter would be good examples.

Another weakness of MapReduce is its simple model: Map and Reduce. While this simple paradigm allows an ease of programming, it also has limitations. This programming model does not support iterative algorithms of which are commonly used for scientific applications and machine learning algorithms. There have been a few efforts to extend MapReduce programming models to support iterative algorithms [17] [9] without much commercial success.

A well-known bottleneck for MapReduce is the disk I/O. As most Hadoop jobs generate a significant amount of intermediate data, disk I/O often becomes the bottleneck. Although LZO compression may reduce the amount of disk I/O during shuffle phase, the amount of time spent for disk I/O still remains significant.

Even with the enormous success of MapReduce, recognition of its shortcomings and the emergence of stream analytics have bought new technologies for business analytics. One technology that has been gaining attention recently is *Spark* [4], which was initially started at the University of California Berkeley in 2009 and moved to Apache in 2013. *Apache Spark* enables applications to store data in memory. This avoids costly disk I/O and boosts up the performance significantly. Keeping the data in memory and allowing applications to access the same data set repeatedly improve the performance of iterative algorithms. Also, *Spark* is applicable for stream analytics. *Spark Streaming* provides an API to perform operations against stream data, such as data filtering and counting words periodically.

As the importance of stream data processing is ever increasing, Google finally abandoned MapReduce and announced Cloud Dataflow in June 2014. Cloud Dataflow supports programming primitives for both batch and stream data processing. Google open-sourced Cloud Dataflow SDK to allure developers to Google Cloud Platform. Amazon, who is another strong player in cloud, also offers a solution for stream processing: Amazon Kinesis. Although not coming from the MapReduce camp, IBM InfoSphere Streams also focuses on stream processing. InfoSphere Streams has been widely used in various industry sectors including financial services, healthcare, manufacturing, and environmental monitoring.

4. ANALYTICS PLATFORM

Various big data analytic platforms are designed and developed to meet the needs of various business analytics demands. For example, Hadoop, a widely used open source implementation of MapReduce, is known to be good at offline big data batch processing. *Spark* [4], an in memory data analytic programming paradigm, has recently attracted significant attention due to the superiority in handling iterative and interactive big data applications. On the other hand, there are also some specialized graph processing frameworks such as Giraph [5], Pregel [16] and Power graph [10] which are designed and specially optimized for graph analytics workloads. These data analytics platforms provide a set of easy-to-use APIs allowing business analytics to perform in depth data analytics on large scale data sets on large scale

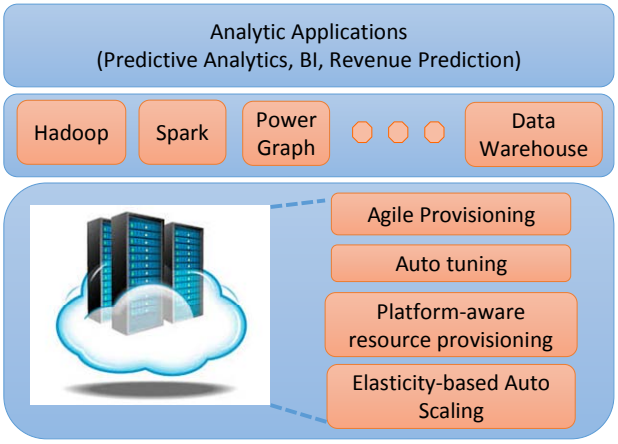

Figure 3. Desired properties of cloud for Analytic Platforms

distributed systems in a real time without worrying about the task partitioning, distribution, and failure recovery details.

Bringing data analytics platforms in the cloud enables the business analytics to inherit the merits brought by clouds. Cloud provides an illusion of infinite resources that are available to meet the resource demand of business analytic workloads. The users can choose to scale up the workloads to a large number of machines speeding them up to meet the service level agreements (SLAs). In addition, the elasticity of cloud provisioning empowers customers the real flexibility to dynamically provision analytic workloads based on the demand in a more fine time granularity. Small and medium companies including startups can start with a small cluster and increase/decrease the cluster size as their business expands/shrinks. The elasticity of cloud can also help cloud providers to better match the resources with demands, improve the cluster utilization by multiplexing more workloads and make higher profits. Moreover, the pay-as-you-go model of clouds allows companies to rent compute and storage for analytic workloads in a short term basis eliminating the need to reserve resource upfront or to overprovision resources.

Containerized clouds [22] represent an important trend that shapes the cloud offerings and the way how business analytics can benefit from cloud. The concept of containerized clouds has burst into news headlines announcing Docker container offerings by IBM SoftLayers, Google App Engine, etc. Containerization alleviates the virtualization overhead of VMs by deploying applications within light weight virtualized layer that runs directly on a single instance of OS on a physical machine. Compared with VMs, containers can be provisioned order of magnitude faster and consume fewer resources since they eliminate the need of hosting multiple copies of OS images in a single bare metal machine.

There are two usage models for analytics workloads in the cloud. One usage model is to provision a dedicated virtualized cluster for each submitted workload. The customers first ingest the data and data processing programs into object stores such as Amazon S3 [2] or Apache Swift [18]. They then request a cluster to run their jobs. Upon receiving the request, the cloud providers provision a dedicated cluster for the workloads, copy the data and data processing programs into the allocated cluster, run the job, copy the data back to the object store after the jobs are finished, and tear down the allocated cluster. This usage model provides better performance isolation and better data locality since data and

compute are usually collocated together on the same set of physical machines. Another usage model from cloud providers is to provision a multitenant analytics cluster and host workloads from different clients to the same provisioned cluster. In the case where performance isolation is not the top priority of customers, using a multi-tenant cluster can rule out the overhead to provision/de-provision a cluster for each client and simplify the cluster management process.

As Figure 3 shows, in order for cloud providers to better serve the need for business analytics, there are a couple of important capabilities that cloud providers should provide. First, for the case where single tenant clusters are needed, the cloud providers should be able to provision the virtualized cluster and tear down the cluster when the workloads are finished in an agile way. While traditional virtualization supports VM provisioning and de-provisioning in the order of minutes, containerization techniques can achieve faster provisioning capabilities. This is particularly useful when there are a large number of workload submission requests waiting in the job queue.

Second, the capability of providing an easy-to-use set of tools that helps users to configure the cluster setup and workload parameters can greatly reduce the barrier of using the cloud enabled data analytic platforms. For example, an automated cluster provisioning component that analyzes the workload characteristics and recommends cluster setup and workload configuration parameters can alleviate the cluster configuration burden from users. Starfish [11] is a Hadoop self-tuning tool that automatically decides the cluster size and workload parameters. It also provides a what-if-engine to answer users' queries in term of exploring the trade-off of low cost and fast turn-around time. MROnline [12][15] is an online parameter tuning tool for Hadoop workloads. It dynamically monitors the Hadoop workloads characteristics such as resource consumptions and data flow patterns and changes parameter configurations to better allocate resources to each task. However, Starfish and MROnline are specifically designed for tuning and configuring Hadoop workloads. It would be desirable to have similar automated cluster provisioning and configuration tuning tools for other frameworks as well. Moreover, given that various analytics platforms are available and each has different strength in handling different types of workloads, it can be beneficial to provide a unified framework that automatically chooses a framework that best fits the resource need of a particular application.

Third, being able to auto scale up and down the cluster quickly can help reduce the cost and increase the cluster utilization for cloud providers. The Amazon elastic MapReduce [1] allows users to scale up and down the clusters as needed. However, it requires users' involvement to manually scale up and down the clusters. Moreover, the data components of the system have less flexibility to scale up and down since it can involve expensive data movement. Auto-scaling capability in a single tenant usage model can help users to reduce the computation cost while it can help cloud providers to adaptively respond to the aggregated workload demands without overprovisioning the cluster resources or violating the SLAs. Moreover, containerization technology brings new opportunities to scale up and down the containerized clusters significantly faster than VM clusters. It opens up possibilities to support new types of workloads, such as online streaming applications, that require clusters which can scale up fast.

5. ANALYTICS APPLICATIONS

To bring the discussion from the vision detailed in the prior sections of the paper to real-world use cases, this section outlines a contemporary application that illustrates the requirements and benefits from the realization of such a cloud-based analytics platform. The application selected, a genomic analytics cloud service, is one that the authors here have detailed knowledge, and which provides examples of different aspects of analytics requirements.

The Genomic Analytics Service (GenAS) is a cloud–based service that accepts samples of DNA data obtained from cancer cells as input, analyzes genetic mutations from the DNA data, searches medical data and literature for drugs reported as targeting those specific mutations, (that is drugs which target the cancer pathways where the mutations are identified), and provides visualization of the affected pathways and how the drugs may be effective in inhibiting the cancer pathways. The GenAS service is targeted at oncologists, and cancer researchers as it users. It relies on large sets of ingested reference data, related to cancer pathways, and the drug relationships to these pathways. This data is compiled from many sources to form a "curated" corpus of well-established knowledge which the analysis needs and uses to form its report of the analysis information to the doctor and research users. To make the service more valuable, it adds to the curated corpus, the advances in research by employing natural language processing to analyze millions of research abstracts and papers, extracting from them new findings of cancer pathways to target or newly discovered effective drug uses. Figure 4 shows the high-level architecture of this application.

The GenAS service has been deployed to the IBM SoftLayer [12] cloud. It provides a user portal for oncologists and cancer researchers to manage patient cases and submit samples, representing DNA data from cancer cells to be processed. Following analysis, the results can be viewed in report form as well as graphic visualizations of mutation findings and the effected cancer pathways by the drugs identified in the analysis report.

An analytics platform has to start with how to get data into the platform, often with challenges such as large size or sensitive data, both of which are relevant to GenAS. The size of the DNA data can vary from less than 100 MB to more than half of a terabyte – the latter being raw data that needs pre-processing analysis to get to a variant or "differences" format (this can be done outside the service as well, and presented as the smaller size input mentioned). Due to sensitivity of genomic patient data, encryption techniques are used to secure the data before it enters the system and is at rest. Decryption is done as needed to perform the analysis, and scrubbing performed not to leave any clear form of the data anywhere within the service.

GenAS uses the IBM SoftLayer provided Swift-based file system, Object Store [13], as a persistent store of the input DNA data and resulting analysis output. Use of Object Store benefits the service by allowing for easy and direct input of data into the service without a bottleneck that might occur if the service itself were to be an intermediary or were involved itself in the data encryption mentioned, instead this is all done on the client side and data directly deposited already encrypted into Object store for the service to access. Object store further facilitates the sharing needed between components in the analysis pipelines - sharing of data being another key component irrespective of the type of analysis used.

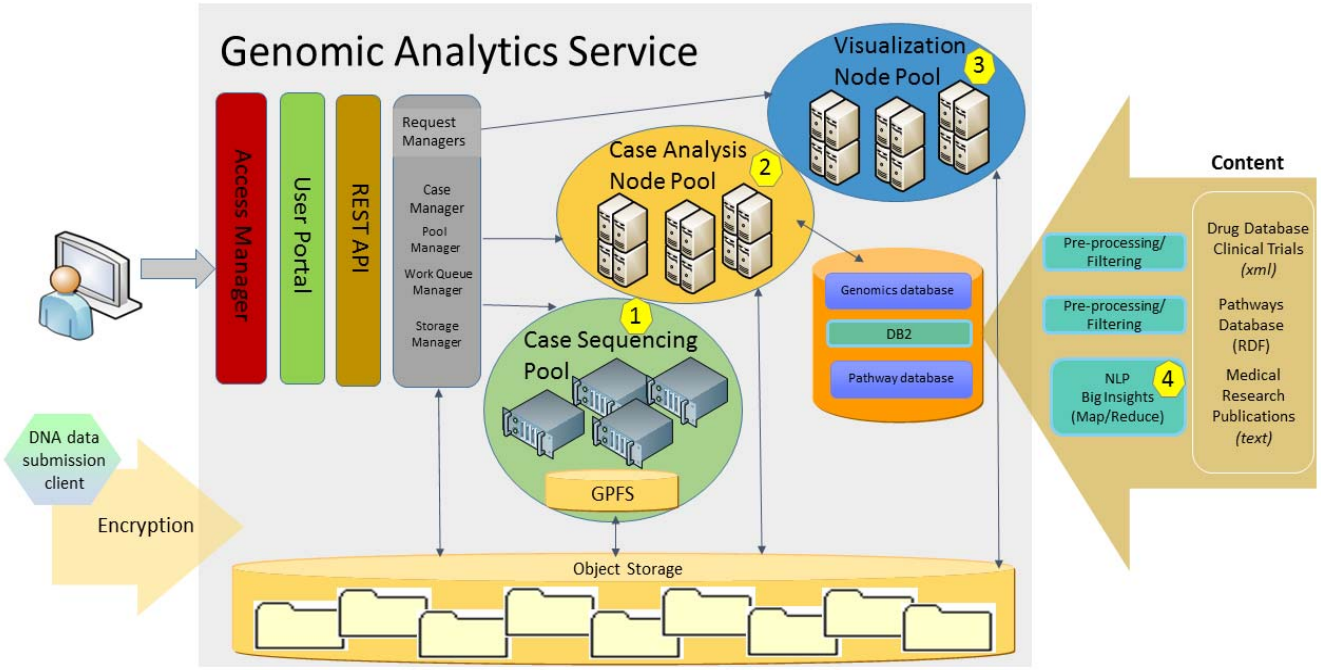

Figure 4. Genomic analytics service (GenAS) component diagram, highlighted items 1-4 identify 4 different analytics types in this service.

With regards to the variety of analysis types used by GenAS, refer first to the case sequencing pool (reference #1 in Figure 4). To accomplish the sequencing steps to transform a raw DNA data sample to a variant data form, there is a set of highly intensive compute and I/O processing steps needed. The case sequencing calls for large machines with a high-throughput file system. GenAS uses large physical (bare-metal) machines put together in a pool each with solid-state drives forming the disk-array base for the high-performance GPFS file system needed to perform the processing. The raw DNA data that has been input to the service, and persisted in Object Store, is brought into GPFS for the sequencing analytics. The result of this processing is a variant format file which is then stored back into the Object Store container with the original raw data – this then serves as input into next step of the analytics pipeline. The computational analysis processing for this has been found to perform best with parallelization across many cores of a single machine instance versus distributed across a cluster of machines. So the machines in this pool are used to each service the sequencing of a single raw DNA data sample.

Similarly, the case analysis (refer to #2 in figure 4), uses a virtual machine pool for performing the analysis of identifying mutations, cancer pathways, and related drugs. It has a pipeline of processing steps which are generally not benefitted from distributing across the likes of a MapReduce or Spark cluster for parallelization. And since data sensitivity requires that unencrypted data not be allowed to be accessed by any but the owner of the data (user who submitted it), both the sequencing and case analysis use the pool model to batch process analysis requests on behalf of the user to avoid other users from ever accessing the system where the data is opened and not encrypted to be analyzed. A queue manager is used by GenAS to manage the queue of analysis requests, and prioritize the requests when they exceed the current pool size. The pool manager serves to manage the active pool of virtual servers – providing elasticity to grow the pool size as demand warrants. While GenAS currently uses SoftLayer virtual servers, experimentation with the use of container (docker) based pools to reduce overhead and time need to elastically grow the analysis pools, is planned as future work

The analysis related to visualization (refer to #3 in Figure 4) is about finding the best way to layout cancer pathway graphs from what was identified in the case analysis, so that the user can better understand how the drugs identified may be effective in thwarting the patient's cancer. This analysis too uses a resource pool, but in this case the active pool members are shared by multiple users, so that the least loaded of these will be used for the next visualization analysis request. The sensitive input DNA data is not exposed on these pool members, and therefore users do interact, albeit indirectly (via secured reverse proxy technology), to get the visualization served to their browser.

In contrast, the analysis processing for extracting new pathway and drug treatments from the large body of new research papers employs MapReduce to parallelize across a cluster of virtual servers (refer to #4 in Figure 4). This processing has to do natural language processing on abstracts and papers to glean from these the relationships stated between drugs and cancer types/pathways. These are added to corpus of knowledge that the service uses to identify drugs for consideration by the doctors in making their treatment decisions. There are millions of research publications processed, and given that the machine learning employed is ever improving, the need to re-run over older data is important as well as running for the new research. This calls for high-performance analytics and the MapReduce model of processing fits well.

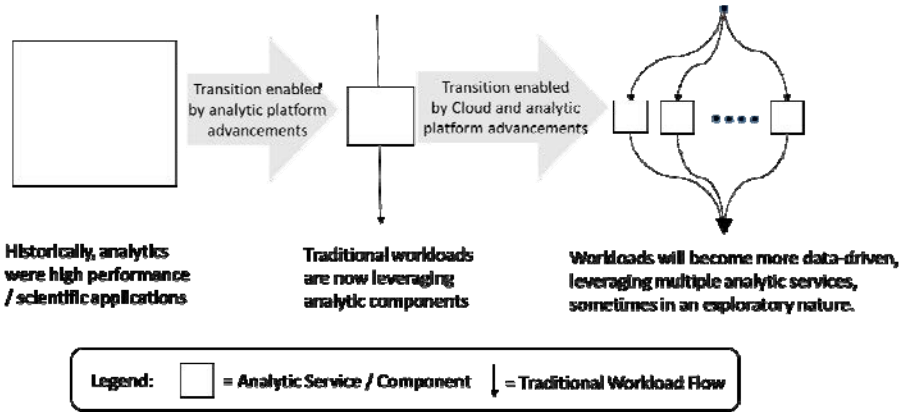

Figure 5. Workload Evolution.

Having a cloud-based analytics platform as outlined in Section 4 for applications such as the Genomic Analytics Service provides the versatility to easily spin-up the type of analysis required, and in the combination and size needed. Having the ease of cloud deployment and variety of analytics options available to experiment and discover what is optimal is key for reducing development time, optimizing the performance and the operational costs by elastically growing and tearing down on demand.

6. A CONTINUED EVOLUTION

As referenced in Section 3, the development of analytic platforms such as Hadoop has helped the industry advance from the point where analytics were primarily high performance computing applications to the point where many begin to leverage analytic components as part of traditional workloads. In addition, as Figure 5 illustrates, the emergence of cloud and related delivery models have driven the cost and deployment inhibitors to levels where new analytic workloads will continue to appear and more traditional workloads will continue to embrace analytic components. In this section, we will describe a few factors contributing to this continued evolution as well as how it may affect the industry's workloads.

6.1 Evolving Cloud Capabilities

Many well-known cloud capabilities have already served to fuel the transition of analytics into more traditional workloads. As described in Section 1, Cloud's on-demand and pay-as-you-go delivery model allow any size analytic frameworks to be constructed and deconstructed based on need, reducing the cost and infrastructure requirements. Having cloud expenses seen as operating expenses versus a more complicated capital expense also helps the workload owners clearly understand the cost and return on investments of specific workloads. However, the growing popularity of a few emerging trends may also help to accelerate this trend. Micro-services architectures in which the workload is comprised of several API-connected services will permit a standard interface to analytic components that will enable portability and ease of maintenance. The growing popularity of PaaS (Platform as a Service) environments and offerings will provide a fertile ground for new analytic components to be created, advertised, and brought into rapidly deployed composeable workloads.

6.2 Emerging Analytic Workloads

As computing cost is driven lower and our ability to gather insights from data grows, we will continue to see rapid growth in every type of analytic workload. Data owners will continue to drive innovation to capitalize on their data at rest through various forms of passive analytics. Sensor technology, including methods of monitoring and encoding complex manual processes, will become more cost efficient and pervasive, inviting many forms of operational efficiency analytics to drive optimization. The ease of use of new analytic platforms and the low cost of cloud will attract solutions owners to build cloud based analytic workload to address mathematical challenges which may not have warranted large system investments in the past. In fact, as cloud computing cost drives towards rock bottom levels and analytic capabilities advance, IT organizations may find that they can explore many different analytic solutions simultaneously with little investment. As Figure 5 suggests, with analytics blocks becoming smaller, more cost effective, and readily available, development time and deployment cost of analytics will continue to reduce. This will lead many workload owners to find ways to make their workloads data driven. For example, workloads which may have previously taken a static or user-provided configuration may instead leverage analytics to determine configuration recommendations or optimizations for the user to consider. The advent of analytics platform in the Cloud will enable wide usage and inexpensive and rapid implementation of analytics applications.

6.3 Continuous Evolution

Active research in strategic areas will continue to drive the creation and use of cloud-backed analytic platforms in today's workload mix. Data transfer and storage advancements have begun which will help facilitate data movement, resiliency, and cost effective storage in the cloud. Once the data volumes exist in the cloud, all types of passive analytics and creative analytic applications can be applied. Early research in data privacy [19][20] and data derivation avoidance will also help meet the requirements of datasets containing sensitive information such as personal or patient data. Lastly, advancements in hybrid cloud which permit the union of on premise and off premise hosted environments will help more organizations ease into the cloud computing paradigm and perform analytic functions across these environments.

7. CONCLUSION

Cloud computing and analytics applications are radically changing the way enterprises utilize IT to gain a competitive advantage for their business. Cloud computing provides low cost efficient IT resources, whereas analytics provides new insights for running their business. Currently, analytics applications are carefully tailored for the target problems, and are custom made. A number of services and programming models born in the Cloud such as MapReduce or Spark, or fast movement of big data in the Cloud are making application of analytics in the Cloud more feasible. Analytics Platform in the Cloud can enable a dramatic shift for analytics with low cost resources, easy to use programming models, and even readily available analytics services which enable rapid building of custom analytics applications.

REFERENCES

[1] Amazon EMR. http://aws.amazon.com/elasticmapreduce/

[2] Amazon S3. http://aws.amazon.com/s3/

[3] Apache. Hadoop. http://hadoop.apache.org/

[4] Apache. Spark. http://spark.apache.org/

[5] C. Avery. *Giraph: Large-scale graph processing infrastructure on Hadoop.* Proceedings of the Hadoop Summit. Santa Clara, 2011.

[6] D. Chen, N. Eisley, P. Heidelberger, R. Senger, Y. Sugawara, S. Kumar, V. Salapura, D. Satterfield, B. Steinmacher-Burow, J. Parker. 2012. *The IBM Blue Gene/Q Interconnection Fabric,* in IEEE Micro, vol. 32, no.01 pp. 32-43. 2012.

[7] J. Dean and S. Ghemawat. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM*, vol 51, no. 1, pp. 107-113. 2008. http://doi.acm.org/10.1145/1327452.1327492

[8] W. Ding and G. Marchionini. 1997. *A Study on Video Browsing Strategies*. Technical Report. University of Maryland at College Park. 1997.

[9] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.H. Bae, J. Qiu, G. Fox, *Twister: A Runtime for Iterative MapReduce*, The First International Workshop on MapReduce and its Applications (MAPREDUCE'10) – HPDC 2010.

[10] J. E Gonzalez, Y. Low, H. Gu, D. Bickson, C. Guestrin. *PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs*. InOSDI, vol. 12, no. 1, pp. 2. 2012.

[11] H. Herodotou, H. Lim, G. Luo, N. Borisov, L. Dong, F. Bilgen Cetin, S. Babu. *Starfish: A Self-tuning System for Big Data Analytics.* In CIDR, vol. 11, pp. 261-272. 2011.

[12] IBM SoftLayer. http://www.softlayer.com/

[13] IBM SoftLayer Object Store. http://www.softlayer.com/object-storage/

[14] LanguageWare Resources Workbench. http://www.alphaworks.ibm.com/tech/lrw.

[15] M. Li, L. Zeng, S. Meng, J. Tan, L. Zhang, A. Butt, N. Fuller. *MRONLINE: MapReduce online performance tuning.* In Proceedings of the 23rd international symposium on High-performance parallel and distributed computing (HPDC '14) 2014.

[16] G. Malewicz, M. Austern, A. Bik, J. Dehnert, I. Horn, N. Leiser, G. Czajkowski. *Pregel: a system for large-scale graph processing.* In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data (SIGMOD '10). ACM, New York, NY, USA.

[17] Microsoft. Daytona. http://research.microsoft.com/en-us/projects/daytona/

[18] Openstack Swift. http://docs.openstack.org/developer/swift/

[19] S. Plimpton and K. Devine. *MapReduce in MPI for Large-scale graph algorithms.* 2011. Parallel Computing. vol. 37, no. 9, pp 610-632. 2011.

[20] R. Popa, J. Lorch, D. Molnar, H. Wang, L. Zhuang. *Enabling Security in Cloud Storage SLAs with CloudProof.* In Proceedings of USENIX ATC 2011, Portland OR.

[21] V. Salapura, T. Karkhanis, P. Nagpurkar, and J. Moreira. 2012. *Accelerating business analytics applications*, In High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on.

[22] P. Wainewright. Virtualization is dead, long live containerization. http://diginomica.com/2014/07/02/virtualization-dead-long-live-containerization/