

Evolving Principles of Big Data Virtualization

Alexander Bogdanov[1], Alexander Degtyarev[1,3], Nadezhda Shchegoleva[1(✉)], Valery Khvatov[2], and Vladimir Korkhov[1,3]

[1] Saint Petersburg State University, St. Petersburg, Russia
{a.v.bogdanov, n.shchegoleva, v.korkhov}@spbu.ru
[2] DGT Technologies AG., Enschede, The Netherlands
valery.khvatov@gmail.com
[3] Plekhanov Russian University of Economics,
36, Stremyanny per, Moscow 117997, Russia
http://dgt.world/

Abstract. The fact that over 2000 programs exist for working with various types of data, including Big Data, makes the issue of flexible storage a quintessential one. Storage can be of various types, including portals, archives, showcases, data bases of different varieties, data clouds and networks. They can have synchronous or asynchronous computer connections. Because the type of data is frequently unknown a priori, there is a necessity for a highly flexible storage system, which would allow to easily switch between various sources and systems. Combining the concept of virtual personal supercomputer with the classification of Big Data that accounts for different storage schemes would solve this issue.

Keywords: Big Data · Data virtualization · Virtual personal supercomputer · Data network · Data marketplaces

1 Introduction

One of the advance computer paradigms is a virtual personal supercomputer [1]. Virtualization brings all computing objects, such as applications, computers, machines, networks, data and even services, to the ability to overcome physical limitations through a wide range of technologies, tools and methods, and provide significant operational advantages for the entire infrastructure. In an increasingly virtualized world, the most effective approach to data are structures that allow it to be virtualized. The well-known data virtualization approach complements the paradigm mentioned above. Big Data has a few features that must be considered when virtualizing them.

To understand the main problems of storage systems for Big Data, it's enough to notice that they arise when you have to match the type of Big Data, how the storage is connected to data servers, and finally, how to organize work with data. The most natural way to classify Big Data follows from Brewer's theorem and leads to six different types of data [2]. The use of cloud storage in addition to the traditional in-band and out-of-band connection allows to implement various ways of connecting storage with data servers, most often hybrid [3]. And finally, you can work with data through

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12254, pp. 67–81, 2020.
https://doi.org/10.1007/978-3-030-58817-5_6

electronic archives, registers, databases, knowledge bases, streaming libraries, data lakes, data meshes, etc. Most often, a distributed system implements several ways of organizing such work.

Thus, the central problem for storage systems now is their flexibility. Now it's already obvious that to achieve flexibility it is most efficient to use virtualization. Gartner's report [4] gives such a definition of data virtualization - the federalization of queries to different data sources into virtual images, which are then used by applications or middleware to create analytical conclusions. However, with this head-on understanding of virtualization, the user will require a sufficiently high qualification and a lot of technical effort to achieve efficiency. The way out of this situation is considered intellectual virtualization [5]. The main idea of this approach is to organize the main part of calculations on remote resources, which are grouped to increase processing speed by data types and tools used. Despite the attractiveness of this approach, it is still quite difficult to implement, and in addition, on distributed systems, there is a problem of a decrease in processing speed due to the need to control errors in data combined into one pool (the situation is quite similar to the problem of a decrease in the speed of consensus processing in distributed ledgers) [6, 7].

It seems to us that a significant part of the problems can be solved if we use the paradigm of the Virtual Personal Supercomputer [8], which was developed for computing, but also used to build a framework for distributed ledgers [9]. The idea of this approach is to virtualize not only the processing itself, but also the entire field on which the processing is performed, namely the network, file system and shared memory. This allows you to create a single image of the operating environment, which simplifies the user's work and increases the processing speed. In this paper, we show how our approach allows us to create an ecosystem that combines the features of federated databases, data lakes, and data networks.

2 Approaches to Store Super Large Amounts of Data

Without context, data is essentially raw information. Information as the very concept means that data is woven into some context. However, even so, it is just material suitable for reporting. To be truly meaningful, data should be focused on business context or carry some functionality, thus becoming knowledge.

The recipient of data needs to pay attention to its sources. Another significant factor is the difference in data volume, especially noticeable for scientific endeavours. In the context of an IT environment, data's physical accessibility is just as important as its structure. The issues lie with the number of resources available to developers and the notable possibility of moving to the cloud to greatly simplify development and further support.

Data can further be divided into several categories. Chaotic data (Unstructured), organized data (Structured), and weakly structured data (Semi-Structured). Structured data follows a specific defined model, for example that of the database. Unstructured data does not and is most frequently stored in a binary format, such as image. Weakly structured data is textual and is stored according to some preset pattern. Weakly structured data can be found in files with extensions of .log, json, .xml. Unstructured

data is much more common than semi-structured and structured types – by several factors in fact. Even though unstructured data can accumulate quickly and is hard to sort, it can carry vital information. Semi-structured data is much more popular though.

Modern data platforms possess almost uniform principal characteristics. They are centralized rather than decentralized, monolithic rather than modular, and have tightly coupled pipelined architecture that could only be managed by a group of highly skilled data engineers.

These platforms typically employ one of three ways to organize data storage:

1. Proprietary corporate data warehouses and business analytic platforms. These are very expensive solutions understood only by a small number of specialists. This small spread leads to their positive impact being severely underestimated in business settings.
2. Big Data ecosystems. These could also be called a Data Marketplace. They possess a data lake and are managed by a central team of specialized high-class engineers.
3. Previous Generation-based. These solutions are more or less similar to the previous generations, but slant towards streaming data. This ensures that data is available in real time. Kappa architecture is most frequently used (Fig. 1). Batch and streaming processing for data conversion that are frequently used are Kappa (Fig. 1). Batch and streaming processing for data conversion are combined within such platforms as Apache Beam. Other features often include fully managed cloud storage services, data pipeline mechanisms, and most recently machine learning platforms. It is obvious that such a data platform eliminates some of the key weaknesses of the previous ones. For instance, data is available in real-time and Big Data infrastructure is less expensive to maintain. However, other problems of the previous solutions remain unsolved.

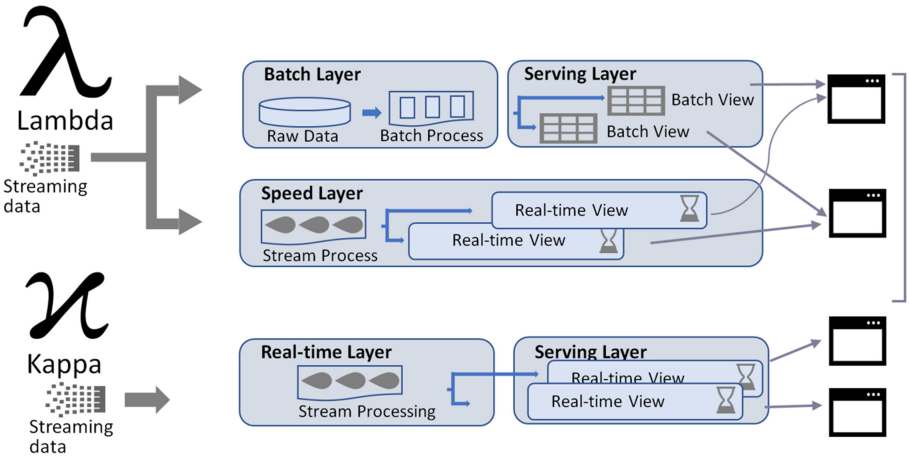


Fig. 1. Lambda and Kappa architectures.

Centralized data platform architectures often fail due to the following notable flaws:

- Inability to manage the consistent emergence of new information sources. As the availability of data increases, the capacity to utilize and organize it under the control of one centralized platform decreases.
- The requirements of building new associations within variously combined data leads to an ever-growing number of transformations. Their aggregates, projections, and slicings significantly increase response time. This has consistently been an issue and remains so even in modern data platform architectures.
- Accounting for the impact of previous data platform architecture generations, specialists distinguish several phases of data processing. The problem arises in relation to the structure of the teams that create and manage the platform. Some of them have high-class data engineers who understand the sources of data origin and how it is used to make decisions. Others are overwhelmed by specialists with extensive experience in technical work with Big Data, but without any knowledge of business and the subject area.

These problems can be resolved by an entirely new paradigm. Such is the distributed data network, an entirely new corporate architecture of a data platform.

To success in decentralizing the monolithic data platforms, our very understanding of data, its location, and ownership needs to shift ideologically. Instead of transferring data from domains to a common lake or central platform, domains should be the ones that house and maintain their datasets, preferably in accessible form. This implies in turn that data can be duplicated in different domains and converted to a form suitable for whichever domain required.

Within this system, the source domain data sets must be absolutely separate from the internal datasets of source systems. The very nature of domain datasets is radically different from internal data required for the operating systems to work. These are much larger in volume, are invariably synchronized facts, and change much less frequently than their respective systems, as business facts do not change so often. The actual underlying storage should be suitable for Big Data, but separate from existing operational databases. The datasets of the source domain are the most quintessential datasets and represent raw data at the time of creation, not customized or modeled for a particular client.

The domain-specific data platform should be able to easily restore these user data sets from the source.

In this case, ownership of the data sets is delegated from the central platform to the domains that are supposed to provide data cleaning, preparation, aggregation and maintenance, as well as the use of the data pipeline. The teams that manage the domains provide the ability to process their data to other specialists in the organization in the form of APIs.

To ensure a quick search for the required data, a registry must be implemented, a data catalog of all available data containing meta-information, such as their owners, source of origin, origin, samples of data sets, etc.

Note that a secure and manageable global control of access to data sets should be implemented. This requirement is mandatory, whether the architecture is centralized or not.

The proposed distributed data network [10] as a platform is focused on domains belonging to independent groups that have data processing engineers and data owners using a common data infrastructure as a platform for hosting, preparing and maintaining their data assets.

A mesh data network platform is a specially designed distributed data architecture with centralized management and standardization for interoperability that is provided by a common and consistent data self-service infrastructure.

Thus, a formal set of requirements can be generalized to form a virtual data model:

- Abstract representation of data in terms of the object model and its sections (rejection of a rigid structure due to a mesh data network).
- Differential confidentiality allowing to determine access parameters on the fly depending on the general role model.
- API-centering data management systems for loading data on demand.
- Refusal from strict separation of streaming and batch processing of data with the necessary switching on the fly, as a part of the implementation of the KAPPA architecture); building a feedback system based on a generalized metadata model.

3 Big Data Types

In [2] it was shown that the solution to this problem is should be based on new specification of Big Data types. This article proposed a method for determining the types of Big Data, the formation of ecosystems (software stacks) for different types of data, and substantiates the Data Lake concept.

Let's consider in more detail the data itself. We propose an approach based on the CAP theorem, albeit the theorem itself concerns distributed data storage, not the data itself. We can divide data into 6 classes (Error! Reference source not found.), however only 5 classes out of 6 potential are possible because PA-class cannot exist by itself, and modern corporate architectures are distributed - that is, divided - by default. Then we have the following data classes.

C – class (consistency): It is characterized by data that: agreed - this is a guarantee that simultaneous reading from different places will return the same value; that is, the system does not return outdated or conflicting data; stored in one place (usually); may not have backups (there is too much data to do backup for them); often analytical data with a short life span.

A – class (availability): It is characterized by data that: should always be available; can be stored in different places; have at least one backup or at least one other storage location; are important data, but do not require significant scaling.

CA – class: data must be consistent and accessible; potentially a monolithic system, without the possibility of scaling or scaling under the condition of instant exchange of information about the changed data between the master-slave nodes; there is no resistance to distribution, if scaling is provided for (branches), then each branch works with a relatively independent database.

In this case, the CA class is divided into 3 subclasses:

1) Big Data of large sizes that cannot be represented in a structured way or they are too large (stored in Data Lake or Data Warehouse): data has any format and extension (text, video, audio, images, archives, documents, maps, etc.); whole data collected, the so-called "raw data"; large data that is unreasonable to place in the database (unstructured data in the case of data warehouses); multidimensional data.
   Medical data that cannot be stored in tabular form (x-ray, MRI, DNA, etc.) are the example of this type.
2) Data of a specific format that can be represented in a structured form (biological data, DNA and protein sequences, data on a three-dimensional structure, complete genomes, etc.) characterized by multidimensional data; data must be analyzed and their sizes reach gigantic values. Medical and bioinformatics data that need to be searched and stored in a relational table with extensions of xml, json, etc. are the example of this type.
3) Other data well presented in relational databases which: have a clear structure or can be represented in the concept of a relational database; the size of the stored data does not matter (provided that lightweight objects or links to large objects are stored in the storage); transactional required;

"Raw" data, though not recommended (an exception - if the logs are stored), customer data, logs, clicks, weather statistics or business analytics, personal data, rarely updated, customer base, etc. are the example of this type.

CP – class It is characterized by data that: must be consistent and at the same time there is support for the distributed state of the system, which has the potential for scaling; structured, but can easily change their structure; must be presented in a slightly different format (graph, document), that is, data for social networks, geographic data and any other data that can be presented in the form of a graph; have a complex structure, because of which there is a potential need for storing files in a document-oriented format; they accumulate very quickly, so a distribution mechanism is needed; no permanent availability requirements.

Frequently recorded, rarely read statistics, as well as temporary data (web sessions, locks, or short-term statistics) stored in a temporary data store or cache are the example of this type.

PA – class: It is characterized by data that: should be available and at the same time there is high support for the distributed state of the system, which has the potential for scaling; have a complex structure, the potential need to store files in a different format with the ability to change the scheme without the need to transfer all the data to a new scheme (Cassandra); accumulate quickly.

This class is suitable for data that is historical in nature. The main task here is to store large amounts of data with the potential growth of this information every day, statistical and other processing of information online and offline in order to obtain certain information (for example, about the interests of users, mood in conversations, to identify trends and etc.).

Before determining the type of system, we must estimate the total system parameters (maximum number of users for simultaneous operation, the ability to scale services, the availability of personalized access), evaluate the project (having its own server capacity, cost comparison with the cost of building rental of services), evaluate

time data access, query performance evaluation for cloud infrastructures, construct the automatic allocation system and send requests in a distributed database (Fig. 2).

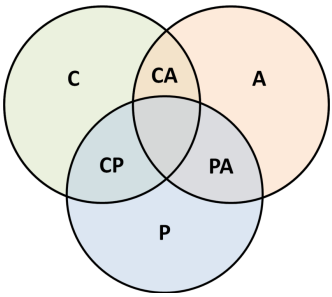


Fig. 2. Graphical representation of Big Data classes.

4 Virtual Supercomputer Solution

The virtual supercomputer is an idea of making a utility-centric computational environment with configurable computation and community characteristics based completely on virtualization technologies used in distributed systems. Such an environment permits flexible partitioning of available physical assets depending on software requirements and priorities of execution. Here we will present some of the widespread standards that must be adhered to while creating one. These concepts are beneficial for fixing problems of a large scale on a virtual supercomputer and some of them can be neglected for small-scale issues [5]:

- A virtual supercomputer is completely determined through its application programming interface (API) and this API must be unbiased of the platform. The API takes form of a high-level programming language. It is the only method to interact with the virtual supercomputer. Moreover, the API does not cover all of the functions of the scheduling system underlying the computer. It is just the means to avoid complications with integration and connection.
- The API of a virtual supercomputer supplies the functionality for seamless integration with other systems. Large-scale problems can be solved when various allotted systems cooperate effectively and their seamless interplay composes the dynamic hybrid distributed systems capable of extending capacity on a need basis. It can even be thought of a s method of scaling a virtual computer to solve problems that are too complicated for just one virtual supercomputer.
- Efficient data processing is achieved with the aid of the virtual supercomputer by distributing data among available nodes and be means of running small programs called queries – essentially on each host where the required data resides. This approach not only supports concurrent operations when running queries on each host, but also has the benefit of minimizing the transfers of data. It should be noted that in current implementations, these programs are not generic. They are parts of an algorithm and are designed specifically to fit the data model which the said

algorithm was initially developed. The shared interface of the virtual memory allows for the efficient processing of data located on any host. To summarize, the storage of large data sets is done by distributed databases, while the general-purpose programs required to process them are written in the virtual shared memory.

- Even using light-weight virtualization could become advantageous in regards to performance. Container-based virtualization could be a good choice for achieving the elusive balance between high performance, process isolation and control, and ease of administering the system in a distributed environment.
- Load balance is achieved using virtual processors with controlled clock rate, memory allocation, network access and process migration when possible.
- Virtual supercomputer uses complex grid-like security mechanisms, since it is possible to combine GRID security tools with cloud computing technologies.

To summarize, the virtual supercomputer is quintessentially an API that offers functions to run programs, work with data stored in a distributed database and to work with virtual shared memory in the application-centric manner based on application requirements and priorities.

5 Virtualizing Data

5.1 Generic Approaches

In the recent times there has been a significant growth in demand for operational access to data. The term "analytics on demand" has appeared to signify a very rapid data-informed business decision making process. Meanwhile, there is practically no time allocated for traditional processes of transforming and loading data as a result of this process. The situation is made more complex by the volume and speed of new data, which appears at a rate beyond the abilities of typical modern corporate infrastructures.

The most effective way to surpass these limitations is to have "virtualized data access". Data virtualization has appeared long ago and some of the older examples are datasets from relational data bases, NoSQL databases, Big Data platforms and even corporate applications, which allows for the creation of logical data storages that may be accessed through SQL, REST and etc. (Error! Reference source not found.). This grants access to data from a large number of distributed sources and various formats, without the requirement for the users to know where it is stored. This eliminates the necessity to move data or to allocate resources for its storage. Apart from greater effectiveness and faster data access, data virtualization may give the necessary basis for fulfilling the requirements of data management (Fig. 3).

Virtualization has three features that support the scalability and operational efficiency required for big data environments:

- Partitioning: sharing resources and moving to streaming data.
- Isolation: Transition to the object representation of data with reference to the domain model.
- Encapsulation: Logical storage as a single entity

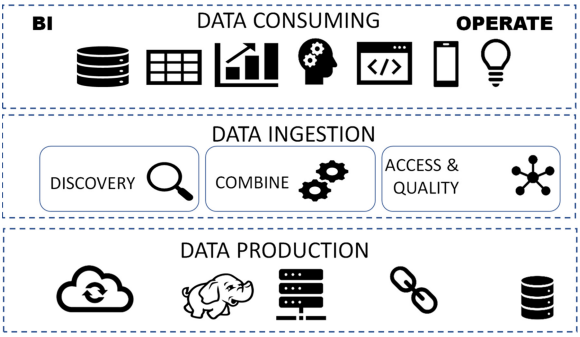


Fig. 3. Data virtualization.

This solution changes the general approach to data in data access abstraction, semantic storage, real-time data access and decentralized security (Error! Reference source not found.) (Fig. 4).

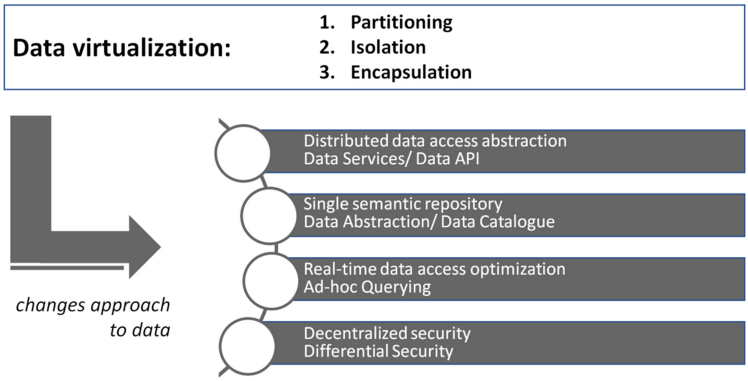


Fig. 4. Features of data virtualization.

Finding and training data processing engineers may be a slow process. **Meanwhile, the results may be suboptimal, if the data specialist does not understand** the business user requirements or is unaware of which methods should be used to accomplish the goals they set. Therefore, suppliers frequently develop analytic products that allow users to resolve these problems themselves.

The main problem lies in the fact that companies have often various data types in different formats, which are located on different systems and servers. A part of them is in the cloud, a part may be located on local services, and access to all of these is determined by varying security policies and practices.

To enable effective work there must be a way to collect disparate data from all of the organization's resources in to one place and represented in one precise way. To

allow for unified data access, organizations typically perform a process known as transforming cloud data.

Transforming cloud data makes data of all formats and sources (both cloud and local ones) readable and accessible. However, there is a set of problems, which make data transformation a long, complicated, and frequently expensive task.

Many suppliers of data virtualization technologies force clients to transform data into their own format before it could be read and used. However, this process of data transformation may lead to data distortion or loss during this translation. Moreover, proprietary formats of many suppliers are incompatible with other technologies. Therefore, you face new problems of continuous integration due to the attachment to a certain supplier. As the data sizes rise, so does the volume of engineering that is required to control the different data sources to quickly fulfill requests.

A solution to these problems is data virtualization, which would create complete independences from the data source format. That means that data does not have to be replicated or transformed in some way. Instead of relying on complex and time-consuming data transformation and transfer processes, it would be more effective to use some business language that would allow the users to easily work with data.

Consequentially, there is a need for a solution that would intellectually virtualize all of the disparate data from various sources into a single unified representation. From there, different BI instruments could receive quick and consolidated answers to make business-decisions. Data is requested "as is", but users perceive it as a singular data storage.

Therefore, smart data virtualization is the new paradigm for controlling data. The "smart" virtualization of data resolves problems of scalability and productivity. Smart data virtualization platforms allow users to avoid large traffic volumes due to federative connections, which create a distributed cache that is optimized for the data platform. By avoiding unnecessary data transfers, the smart data virtualization allows for a more stable productivity of requests with far smaller resources required.

Meanwhile, it is necessary to implement work with different data sources: relationship databases (for example, Oracle, Teradata, Snowflake), file-based ones (CSV, JSON, XML, HDFS, S3), API-based ones (REST, HTML), and application-based ones (Salesforce, Workday, Service Now). This will allow for the use of practically any data: local, cloud, structured and unstructured without using ETL or transferring the data by hand.

At the same time, the virtualization platform must allow for even better productivity than native platforms that they work with, since the level of virtualization must correspond to or surpass the current solutions that they are replacing.

We must note that because data virtualization platforms are an intermediary software for analytic requests, it is necessary that the platform must integrate with the security structure of the enterprise.

All of the aforementioned challenges could be received using a virtual supercomputer.

5.2 Big Data Virtualization

Virtualization of Big Data through logical constructs and object access (the data themselves may be stored in different sources and be collected on request and/or be accessed (interpreted) at various trigger points (event-based integration)):

- Logical storage of data according to function is analogous to a traditional storage of data, with several exceptions. For a start, in a logical data warehouse (LDW) data is not stored, unlike in data storages where data is prepared, filtered, and placed.
- Logical abstraction and division: heterogeneous data sources may now easily interact through data virtualization.
- Differential confidentiality (intersecting access levels).

The fact that over 2000 programs exist for working with various types of data, including Big Data, makes the issue of flexible storage a quintessential one. Storage can be of various types, including portals, archives, showcases, data bases of different varieties, data clouds and networks. They can have synchronous or asynchronous computer connections. Because the type of data is frequently unknown a priori, there is a necessity for a highly flexible storage system, which would allow to easily switch between various sources and systems.

Combining the virtual personal supercomputer with the classification of Big Data that accounts for different storage would solve this issue.

5.3 Data Networks and Data Marketplaces

We looked at several important characteristics of modern data platforms: centralized, monolithic, and with rigid conveyor architecture controlled by a group of highly specialized data engineers.

There are three core approaches that should be noted within the classification of data storage.

The first is proprietary data storage. These storages and business analytics platforms are highly inflexible, very expensive solutions. Using them involves a small group of specialists, which leads to wasted potential that this storage may have had on the business operations.

The second is a Big Data ecosystem. It contains a data lake managed by a centralized team of highly specialized data engineers.

Finally, the third type is the data marketplace (Error! Reference source not found.). These solutions are similar to the first two, but lean towards streaming of data and real-time access to insight. Batch and streaming data conversion processes are combined through platforms like Apache Beam, Kappa architectures are used, as well as fully controllable cloud storage services, data conveyor mechanisms, and machine learning platforms.

Real-time analysis and expensive Big Data infrastructures are problems for the first two approaches, but not for the latter. If we are looking into the main problems of using a centralized data platform architecture, the following should be mentioned:

Continuous emergence of new data sources. The amount of data accessible is increasing at an exponential rate and the ability to use and reconcile this data under one platform's control is diminishing proportionally.

Organizations seek to combine data in different ways to reflect their fluid business environments and demands. This leads to an increasing number of data transformations, aggregates, projections, and slicing. The response time rises beyond acceptable levels, which is a problem faced even by modern data platform architectures.

When implementing data platform architectures, specialists are influenced by past architecture generations when identifying data processing stages. In particular, this is seen when structuring commands that create and control the platform. These specialists are mostly high-class data engineers, who understand data sources and the principles of using data to make decisions. Some specialists have great technical experience, but frequently lack the knowledge of business and areas of application.

The new paradigm for corporate data platform architectures is the decentralized data network, as it allows for the successful resolution of aforementioned problems. This paradigm requires a shift in the understanding of data, its location and belonging (Fig. 5).

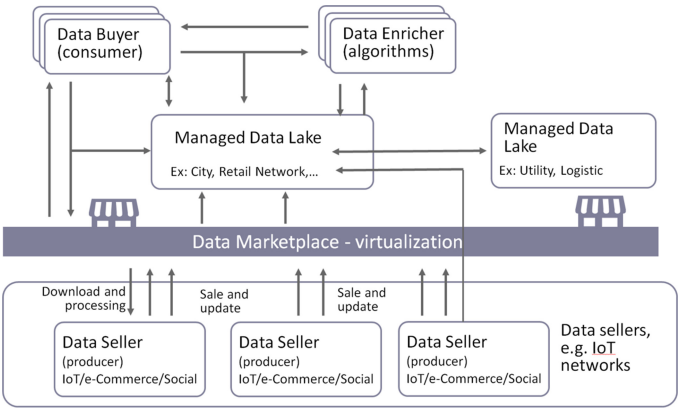


Fig. 5. Data marketplace.

Instead of transferring data from domains into lakes or from centrally owned platforms, there must be an easier way to store and service data, including duplicating data in different domains to allow greater flexibility in its transformation.

A recent example of such a decentralized platform for data storage is the DGT Network [11]. It creates a virtual data mesh, connecting different sources of data across corporate information borders into unified analytics accessed by authorized users in a manner conductive to differential confidentiality.

6 Big Data Virtualization Applications

Studies have shown that data marketplaces already offer several important advantages for companies seeking to put their data to effective work. One of these uses is the construction of ecosystems, best illustrated by the data marketplace powered by the DGT Network. DGT Network allows for horizontal integration by creating distinctive clusters of enterprise-operated nodes, which exchange data through a secure F-BFT Protocol and record it in a unified ledger (the Direct Acyclic Graph ledger). Even though this ledger serves as a "unified source of truth" for its participants, the differential anonymity protects corporate privacy of source data, while still allowing analytics to provide relevant insights to participants in real time.

Other advantages of data virtualization may include new monetization opportunities. One particular digital marketplace adds value to Europe's electric-automobile market by enabling data and transactional gateways for a diverse group of businesses, including charging-infrastructure providers, vehicle manufacturers, mobility-service players and others. These participants use customer habits and market trends as raw data that informs their dynamic pricing structures.

Many industries are beginning to embrace virtualization of Big Data within their operations. One example is the Data Marketplace launched by the IOTA Foundation as a proof of concept in 2017. IOTA initially launched an open-source distributed ledger that connects Internet of Things devices to process micro transactions in exchange for crypto currency. Unlike a blockchain, IOTA's distributed ledger – the IOTA Tangle does not group transactions in blocks, like a typical blockchain, but instead sees them as a stream of individual transactions entangled together through a relatively simple network algorithm – in order to participate, a node needs to perform a small amount of computational work to verify the two previous transactions. The IOTA Data Marketplace was designed to be a simplified platform that simulates how an IOTA-connected device is economically incentivized for sharing secure data over to a web browser. Despite its limited functionality as a Proof-of-Concept for the larger IOTA platform, the IOTA Data Marketplaces has been utilized by over 70 organizations, including Accenture, Bosch, KPMG, T-Mobile, Fujitsu, Philips, Tele2 and other organizations working with Internet of Things devices and sensors.

A study by McKinsey notes how highly applicable digital marketplaces are to IOT data as a field through combining data sets and data streams to ensure consistency, high quality, and security. As part of their research on the underutilization of IoT data by corporate enterprises, they note a typical example where one oil rig with 30,000 sensors examines only 1 percent of the data collected to detect and control anomalies, ignoring the greatest value that comes in predictive analytics and optimization. In this application, a dynamic data would harness that additional information by making both its sourcing and processing effective and inexpensive.

Some dynamic business models are especially adept at using Kappa Architecture and data virtualization. Uber, for example, uses data processing systems like Apache Link and Apache Spark to calculate real-time pricing, to enable optimized driver dispatching, and to minimize the fraud on their platform. With its widespread operations, Uber relies on this architecture to process data on a massive scale with exactly-once

semantics. In order to backfill Uber's streaming pipeline, Uber has tested Kappa architecture through two of the most commonly used methods: relating data from Apache Hive to Kafka and backfilling as a batch job. Even though neither approach was found to be scalable enough for Uber's data velocity, they have found a solution by modeling their Hive connector as a streaming source in Spark Streaming.

7 Conclusion

Data virtualization is a method of organizing access to data without requiring the information about its structure or place in any particular information system.

The main goal is to simplify access and use of data by turning it into a service, essentially shifting the paradigm from storage to usage.

Previously, the task of using data was solved through integration into an intermediary storage system. It already had some elements of virtualization through data marts created by data producers. Now the data consumer is coming into the focus.

There are three core characteristics of virtualization that support scalability and operational effectiveness necessitated by Big Data environments. These include: portioning, which is the division of resources and a shift to streamed data; isolation, which is an object-oriented approach to data with domain application in mind; and encapsulation, keeping the logical storage as a singular object.

Data services and API change the way distributed information is accessed. Data abstraction forms from a single semantic repository. Access to data in real-time and ad-hoc querying are optimized. Finally, differential security and privacy are achieved. Data virtualization is more than just a modern approach, it is an entire new way of seeing data.

References

1. Bogdanov, A.: Private cloud vs personal supercomputer. In: Distributed Computing and GRID Technologies in Science and Education, pp. 57–59. JINR, Dubna (2012)
2. Bogdanov, A.V., Shchegoleva, N.L., Ulitina, I.V.: Database ecosystem is the way to data lakes. In: Proceedings of the 27th Symposium on Nuclear Electronics and Computing (NEC 2019), vol. 2507, pp. 147–152. Aahen University (2019)
3. Nemade, R., Nitsure, A., Hirve, P., Mane, S.B.: Detection of forgery in art paintings using machine learning. Int. J. Innov. Res. Sci. Eng. Technol. **6**(5), 8681–8692 (2017)
4. Menon, S., Beyer, M., Zaidi, E., Jain, A.: Market guide for data virtualization, 16 November 2018. ID: G00340606. https://www.gartner.com/en/documents/3893219/market-guide-for-data-virtualization
5. Gankevich, I., et al.: Constructing virtual private supercomputer using virtualization and cloud technologies. In: Murgante, B. (ed.) ICCSA 2014. LNCS, vol. 8584, pp. 341–354. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09153-2_26
6. Bogdanov, A., Degtyarev, A., Korkhov, V., Gaiduchok, V., Gankevich, I.: Virtual supercomputer as basis of scientific computing. In: Horizons in Computer Science Research, Chap. 5, vol. 11, pp. 159–198. NOVA Science Publishers (2015)

7. Bogdanov, A., Degtyarev, A., Korkhov, V.: New approach to the simulation of complex systems. In: EPJ Web of Conferences, vol. 108, pp. 1–12 (2016). Article number: 01002
8. Bogdanov, A., Degtyarev, A., Korkhov, V.: Desktop supercomputer: what can it do? Phys. Part. Nuclei Lett. **14**(7), 985–992 (2017). https://doi.org/10.1134/S1547477117070032. ISSN 1547-4771
9. Korkhov, V., Kobyshev, S., Degtyarev, A., Bogdanov, A.: Light-weight cloud-based virtual computing infrastructure for distributed applications and hadoop clusters. In: Gervasi, O. (ed.) ICCSA 2017. LNCS, vol. 10408, pp. 399–411. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62404-4_29
10. Zhamak, D.: How to move beyond a monolithic data lake to a distributed data mesh. https://martinfowler.com/articles/data-monolith-to-mesh.html
11. DGT, the decentralized enterprise platform. http://dgt.world/