kinetica

Inside Kinetica

Inside

WHITE PAPER

Summary

For years, the CPU clock speed race trumped almost every other IT metric and thus many applications were optimized around single CPU architectures.

Big data applications were benchmarked by the volume, velocity, and variety of data they processed, almost entirely ignoring the complexity of data analysis involved and the unpredictable nature of data: long-lived vs. perishable, human vs. machine, and structured vs. unstructured, among others.

Today's big data applications are getting even bigger, and are being challenged with an extreme data problem with growing unpredictable data and the increasing complexity of analysis that brings. The challenge is that delivering real, actionable insights is rapidly becoming every company's most valuable resource, and therefore, businesses need to act with unprecedented agility to grasp the bigger picture.

Even though traditional systems are common, they are not well designed for handling large-scale analytical and streaming workloads such as those found in modern data-warehousing, decision support, and business intelligence applications.

Kinetica is a GPU-powered insight engine for the Extreme Data Economy that is best for running large-scale analytics and streaming workloads.

Whether you are a data scientist or architect, this paper is for you. It goes deep into the internal architecture of Kinetica. It assumes you have a basic understanding of Kinetica and want to explore what's under the hood.

1 / Core Concepts

1.1 GPU-accelerated technology

As the focus of innovation shifted away from greater clock speeds, hardware designs moved from single core architectures to multicore. This greatly changed application execution, from single instruction at a time to parallel instructions on many symmetric cores.

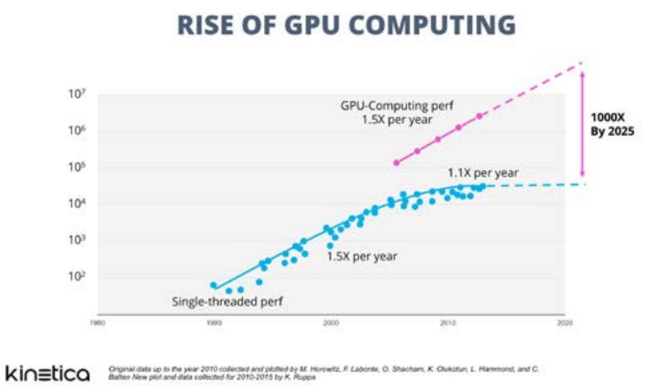


Figure 1 : Growth of multi-core architectures (Number of transistors over time)

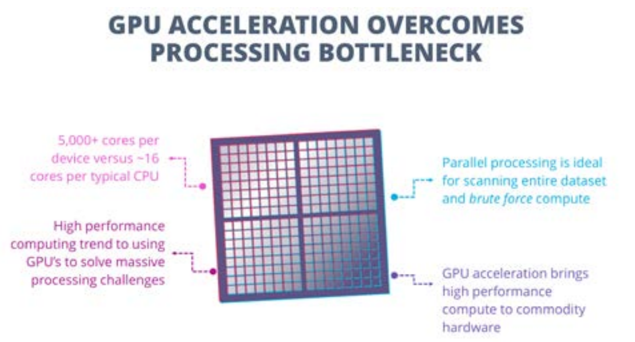
"

The CPU (central processing unit) has often been called the brains of the PC. But increasingly, that brain continues to be enhanced by another part of the PC – the GPU (graphics processing unit), which is its soul"

Source: Nvidia

Although, it seems like the central processing units (CPUs) got all the glory for comp uting horsepower in part by using the massively parallel processing paradigm with horizontal scale out at the server level, the graphical processing units (GPUs) have become the processor of choice for many types of intensive parallel applications to accelerate workloads and provide faster insights.

The most fundamental architectural ingredients that make Kinetica unique are its in-memory GPU architecture and advanced GPU abstraction technology, that allow it to utilize the combination of CPUs and GPUs to deliver superior performance at a lower cost. This ingredient is special because most other similar systems are either not fully memory-based, or still use either GPU or CPU compute alone, leaving the other resource under or un-utilized.



GPU ACCELERATION OVERCOMES PROCESSING BOTTLENECK
5,000+ cores per device versus ~16 cores per typical CPU
High performance computing trend to using GPU's to solve massive processing challenges
Parallel processing is ideal for scanning entire dataset and *brute force* compute
GPU acceleration brings high performance compute to commodity hardware

Figure 2 : Key differentiators of a GPU

As shown in Figure 3, Kinetica offloads computationally-intensive operations to the GPU cores, while the remainder of the operations are run on the CPU cores. The GPU has hundreds to thousands of cores that can speed up ingestion and analysis of data by up to 100x.

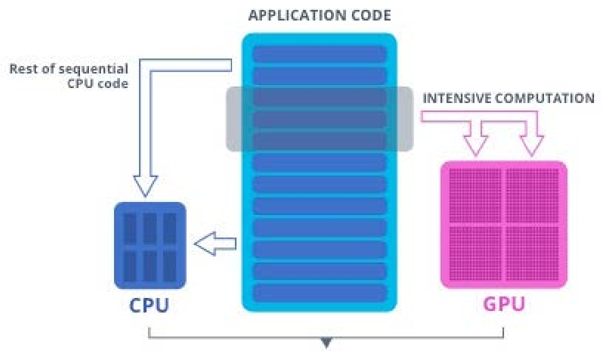


Figure 3 : Accelerating application performance by parallelizing between CPU and GPU

1.2 Memory first approach - all the way down to the chip

Kinetica adopted a memory-first architecture that includes using not only system memory or host memory but also chip memory or VRAM co-located with the GPU. This allows Kinetica to support not only quick operational point lookups but also operational analytics with results in milliseconds.

1.3 Column-oriented design for analytics

Typically, analytical datasets are large because they contain huge amounts of historical data. Kinetica's native column-oriented design provides efficient data storage and blazing-fast query performance for analytical datasets.

This section dives deeper into the columnar design of Kinetica and why using GPUs is optimal for analytics.

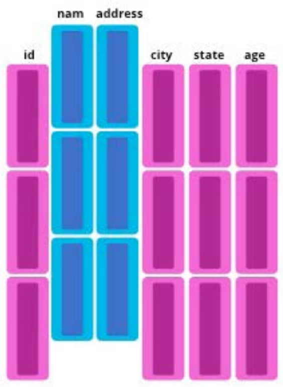


Figure 4 : Data represented as columns

As shown in Figure 4 above, analytical fact tables are tall and thus column values can be co-located together for faster processing. In addition, column data can also be compressed to efficiently save space.

In Kinetica, there are several compression algorithms you can choose from to compress a table. As shown in Figure 5, individual columns are broken up into multiple column segments, which are individually compressed using dictionaries. Dictionaries drastically reduce the memory usage of tables and help in fast movement of character-based column values from CPU to GPU.

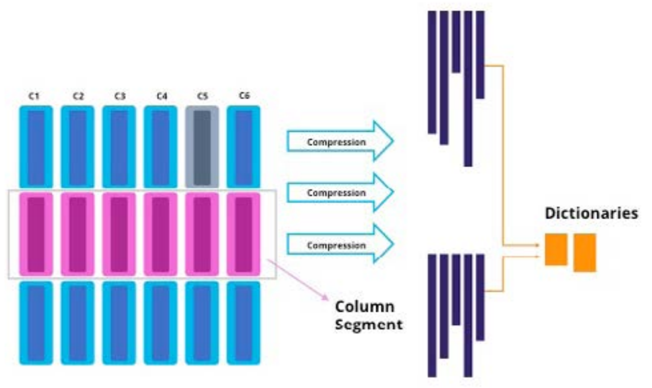


Figure 5 : Compressing column segments and dictionary encoding

In Kinetica, column data can be compressed in memory and storage. When needed, a copy of the column segment is uncompressed for use and then discarded when it is no longer being processed. If new data is updated into a column, the associated column segment is uncompressed, modified, and then recompressed immediately. For columns with string data values, dictionaries are built to achieve better compression ratios.

With Kinetica's column-oriented design, no row-to-column transformations or data reconstruction are required, resulting in blazing query performance. Queries that operate on columns are also vectorized to leverage multi-core and GPU architectures. This design enables multiple column values to process simultaneously in batches, rather than one at a time, boosting query throughput.

2 / Kinetica in your environment

2.1 Kinetica where you want it

Kinetica is a cloud-friendly engine that allows you to harness the power of GPU acceleration in scalable cloud environments, whether public, private, or virtualized. This offering gives companies of all sizes an easier pay-as-you-go model to solve extreme data problems. Kinetica continues in lock step with NVIDIA and other cloud platform providers in bringing the latest GPU insight engine technology to the cloud.

In the cloud, Kinetica can run on these platforms: with Amazon's AWS support for the latest enterprise-grade GPU instances from NVIDIA, and Google's second-generation instances with GPU support options. On the NVIDIA GPU cloud, Kinetica is also available as a container. On premises, Kinetica can run on several infrastructures, including NVIDIA's DGX supercomputer, or IBM, Dell, and HP Enterprise servers that include NVIDIA GPU and the CUDA driver.

2.2 High-Level Deployment Architecture

As shown in Figure 6, a typical Kinetica deployment in your ecosystem consists of multiple clusters, each comprised of multiple commodity servers, each equipped with GPUs.

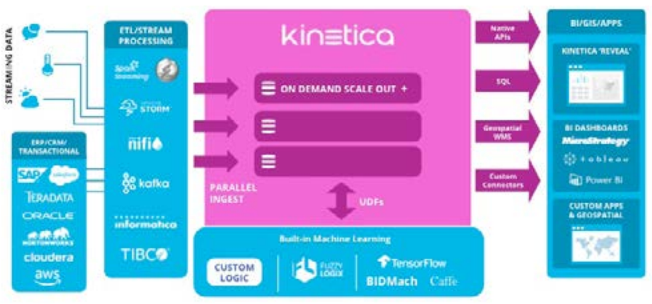


Figure 6 : Kinetica deployment in your ecosystem

Thanks to the symmetrical nature of all servers, scaling Kinetica up-and-out is easy and linear. To scale up, just add more GPU computing or storage resources to your servers. To scale out, just add more commodity servers or throw in additional GPU computing power to scale out your processing per server, without replacing the entire server.

With Kinetica, loading data into the cluster for analytics and reporting is designed to be simple and fast. Data records can be ingested into tables using distributed ingestion and extraction mechanisms. This makes it possible to load dimension and corresponding fact tables with billions of rows very quickly, without any indexing requirements and slowdowns.

Interacting with Kinetica is no different than interacting with any other data platform, even though it has many architectural differences under the hood.

Applications can send requests to any of the servers for processing via the SQL-92-compliant ODBC connector, or over native REST APIs that support JSON or avro serialization. This means that existing ETL and applications can stay, minimizing the time needed to get up and running with Kinetica.

2.3 Key use cases

Kinetica delivers low-latency, high performance analytics on large data sets, and makes running SQL queries on streaming and geospatial data easy. With Kinetica, you can continuously collect, analyze, and integrate streaming data with historical data. This makes it an extremely attractive insight engine for powering data science and business applications.

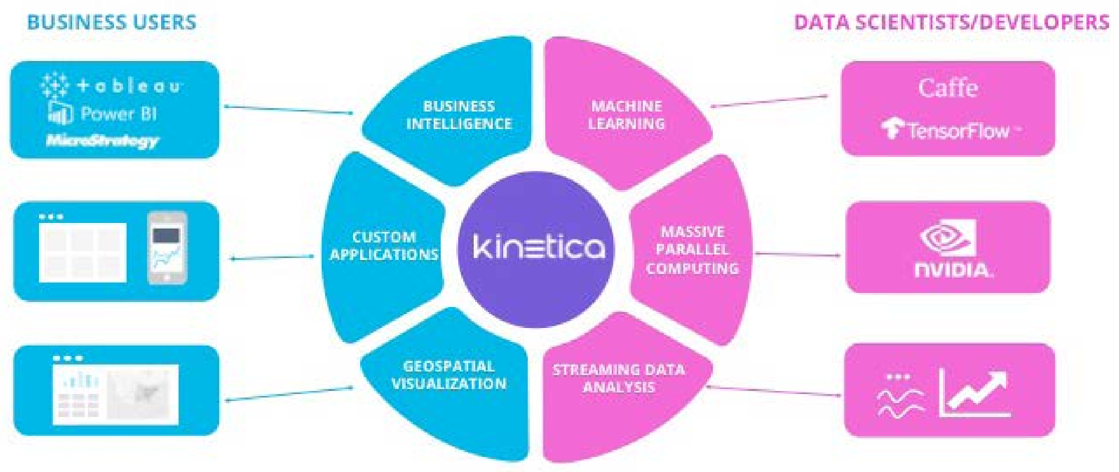


Figure 7 : Kinetica's core data science and business use cases

3 / Core Architecture

3.1 Data Management in Kinetica

Applications can access data in Kinetica through tables. Table data can be either sharded or replicated, for distribution across the cluster.

When sharding is used, the shard key (which could be the primary key) for each table record is hashed, and Kinetica determines which server the record will reside on. If there is no primary or shard key defined, Kinetica will distribute the table data in round-robin fashion, by picking a different server for every batch of records it ingests. Sharding uniformly spreads data and workload across all the servers, while maintaining a single copy of the record across the cluster. Sharding typically works well for large tables such as a fact table with billions of rows.

When replication is used, every Kinetica server within a cluster has the entire table dataset. Changes made to the table are applied to every table copy on the entire cluster. Replication works well when tables are relatively small and the workload requires joins with other tables. With replication, datasets are available locally on each server, and the joins can happen locally on each server.

Data in Kinetica is uniformly distributed across the cluster and stored in data containers on disk called TOMs. A rank can have one or more ***TOM***s, and by default, a rank has 1 TOM. A TOM is broken up into **chunks**. A chunk is the basic unit of operation for processing in Kinetica. Each chunk has a default size of 8 million records. Chunk size can be set per table to override the default value. As an example, Figure 8 shows server 0 with rank-0 process executing. The rank has 4 TOMs, (TOM 0..TOM3) in this case. Within each TOM, the tables are sharded.

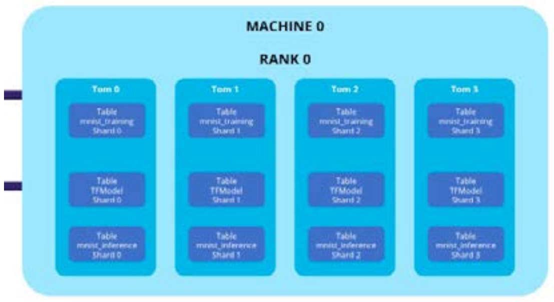

MACHINE 0
RANK 0
Tom 0
Table mnist_training Shard 0
Table TFModel Shard 0
Table mnist_inference Shard 0
Tom 1
Table mnist_training Shard 1
Table TFModel Shard 1
Table mnist_inference Shard 1
Tom 2
Table mnist_training Shard 2
Table TFModel Shard 2
Table mnist_inference Shard 2
Tom 3
Table mnist_training Shard 3
Table TFModel Shard 3
Table mnist_inference Shard 3

Figure 8 : Tables sharded across TOMs in a rank

3.2 Bootstrapping Kinetica

The **host manager** in Kinetica is a supervisor process that manages several install processes, and is at the heart of bootstrapping the system across the cluster. The host manager starts Kinetica's **global manager**, which is a parallel orchestration engine and controller for system configuration across the cluster.

Figure 9 : Kinetica web UI to viewing and controlling ranks

When the cluster starts up, on each server, several processes called **ranks** begin. Each rank is responsible for a single GPU. On a machine with multiple GPUs, one rank per GPU will launch. On startup, the ranks send their status to the global manager.

By using the Web UI (as shown in Figure 9), an administrator can control which rank process executes on which GPU of the server. Ranks can range from 0 to N, with rank-0 the head aggregator process, and ranks (1..N) the worker processes. When the **head aggregator process** starts up, it does not bring up the HTTP server but waits for the global manager to tell it that the workers are up and ready. After all the worker processes are ready, the head aggregator process starts up the HTTP server to allow requests to flow.

When the workers start up, they checkpoint with the global manager and the relevant TOMs start loading data. When starting a Kinetica cluster for the first time, necessary disk files are created. This is primarily for managing data persistence. However, if there is data already on disk (likely because of a server restart), the servers read the data from disk into memory. This is for data that was previously ingested and persisted in Kinetica. This process is called **warmup**. If full-text search is enabled, they initialize the full-text search components. If multi-head is enabled, worker processes start the HTTP server, and notify the global manager when they are ready to serve requests.

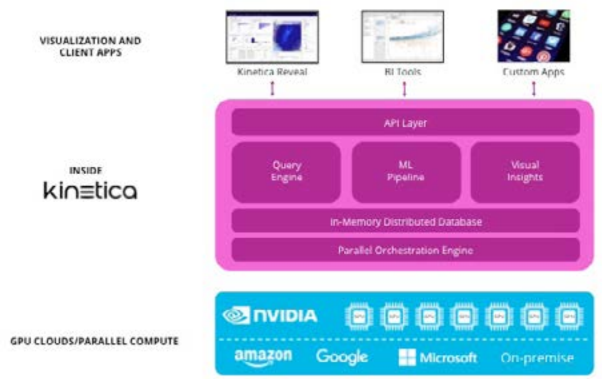


Figure 10 : Kinetica under the hood

For the purpose of high-availability and uniform workload distribution, each cluster has one optional HAProxy component that runs on the same hardware as the cluster itself. We will discuss this in more detail in the high-availability section of this paper. Other components such as the query engine, machine learning pipeline, and visual insight engine also start before Kinetica is ready to begin receiving and serving requests.

3.3 Client Architecture

Kinetica provides a rich set of APIs for querying & managing data. To talk to Kinetica, application SDKs are available in a variety of languages including C++, Java, JavaScript, NodeJS, Python and C#. Existing SQL applications can also easily migrate to Kinetica using the ANSI SQL-92-compliant ODBC connector. Additional language bindings can be constructed for any language capable of HTTP requests and JSON parsing.

So how does data flow from a client application to Kinetica? What happens within a Kinetica server when it receives a DDL statement or query? We will go into the details of the query engine in the next section, but before that, Kinetica authenticates and validates the request as described below.

If a high availability cluster isn't set up, requests from the client directly hit the Kinetica cluster and can be serviced by any node. The head aggregator processes and then manages the request.

In case of multiple clusters set up for high availability, requests from the client application go through the load balancer and hit the HAProxy component on each cluster. The HAProxy sends each query to the next available instance, allowing for heavier concurrent data accesses. Depending on whether the request needs to be serviced by all the clusters (for example, **synchronous requests** such as DDL statements or administrative tasks), HAProxy forwards these requests to the HAProxies residing on the other clusters. Additionally, no further requests are processed by the receiving HAProxy until the client's synchronous operation is complete. Typically, synchronous requests execute very quickly.

In the case of requests that don't need multiple clusters (for example, **asynchronous requests** such as DML operations including joins, filtering, and aggregation), the HAProxy continues to receive and service client requests without any waits, and these requests are round-robined to one of the available clusters.

Kinetica can be setup to run with or without client authentication. To learn more about **client authentication** in Kinetica, read the security section of this white paper. If client authentication succeeds, the request makes its way to the head aggregator process. If client authentication fails, an error is sent back to the requesting client.

The head aggregator process further validates the request, determining the tables and worker ranks. If the request can be completely serviced by the head aggregator process, the request is not sent to the worker ranks, and the head aggregator process directly processes the request and responds to the client. If the request cannot be completely serviced by the head aggregator process, it registers the request as a job to be scheduled by the global manager. Next, the global manager publishes the job to worker ranks for execution. Worker ranks have task receivers and thread queues running on the GPU that the rank is executing on. Based on whether the requests are synchronous or asynchronous, separate threads are used to process these requests and produce results.

3.4 In-Memory Distributed Database

Kinetica runs completely in-memory to optimize throughput and deliver fast query performance. During warmup, data is loaded from disk into memory. Kinetica tiered memory management system ensures that hot, warm, and cold table data is spread across RAM and VRAM resources (the memory available on the GPU card itself).

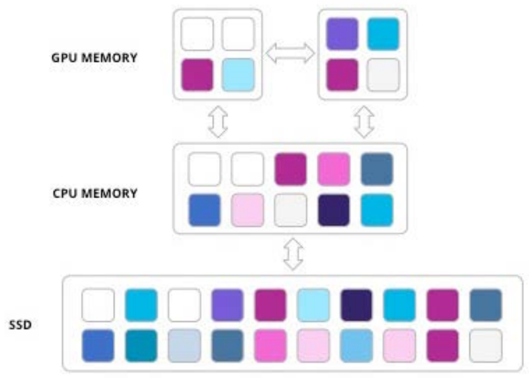


Figure 11 : Data distribution across Disk (SSDs), RAM, and GPU VRAM

This architecture enables Kinetica to deliver exceptional performance across different hardware configurations and dataset sizes.

Hot data is typically kept in VRAM. VRAM is generally small, but much faster than system RAM. By pinning data in VRAM using Kinetica's VRAM boost mode feature, the GPU has direct access to data in VRAM and there isn't any data movement between VRAM and RAM. Kinetica can thus deliver lightning-fast query performance in several use cases, while still being able to leverage cluster-wide system RAM to scale up and out to multi-terabyte in-memory processing.

In Kinetica, cold data is typically stored on disk. If there is no more space in memory, Kinetica depends on the operating system and swap space to manage memory by freeing up space to move the data required from disk. A richer tiered storage approach with advanced predicates and policies is planned for an upcoming release. This will allow Kinetica to manage data in all tiers, from GPU RAM to System RAM to SSD or disk.

3.5 Querying and Indexing

Kinetica supports ANSI SQL-92-compliant syntax, allowing SQL queries to contain multiple filters, complex predicates, joins, subqueries, and more like traditional relational databases.

When running this kind of query on large datasets with billions of rows, the number of numerical computations performed is a product of the complexity of the query predicates and the number of rows to be processed. Even when distributed, a conventional query engine using CPUs alone cannot deliver the result within an acceptable period. The query latency is huge, ranging from many minutes to hours. With Kinetica, SQL queries can process and analyze billions of rows in a matter of microseconds, delivering 50-100x faster results, without heavily relying on storage-bound indexing.

In Kinetica, queries can be executed by the application through the SDK, API, or directly via the ODBC connector. Once Kinetica receives a query, it is first parsed and converted into a relational algebra graph called the **query plan**. The query optimizer in Kinetica then applies various types of rules to rearrange query operations and functions to produce an optimal plan: the best approach to execute the query. In the case of queries with joins, Kinetica attempts to use the most efficient join option possible to run with a low-memory footprint. Additionally, if the user has supplied query hints, or if there is a column index, these are considered to further improve the query plan during optimization.

Finally, Kinetica's query execution runtime takes the optimized query plan, and runs it. The execution runtime leverages the CPU and GPU heavily to parallelize query execution whenever possible, like in the case of complex aggregations, group-bys, and joins. As the result set is generated, Kinetica streams the results back to the client while the query is still running. This enables applications to begin consuming data right away without fully waiting for the query to complete, and results in substantial performance gains. For example, for query requests that contain filtering conditions, the workers process the relevant table chunks and quickly return a subset of the data that correspond to the TOM, which is made available to the application.

3.6 High-Availability

Across clusters, the **HAProxy** component per cluster manages requests and quickly detects whether or not a cluster is available.

Within a cluster, the global manager monitors, starts, and stops ranks. It communicates with other ranks using a high-performance messaging library. It periodically sends heartbeat requests to rank processes to check whether or not they are online. The ranks respond back with their status, and this is stored in the global managers' registry. When network outages occur, ranks send updated status information about communication problems with others to the global manager. If a rank process is down, the global manager stops and restarts the rank. If the head aggregator process is down, another rank is selected to be the head aggregator rank. When the rank with an outage comes back online, it sends its new status to the global manager, and the global manager reinstates the rank along with its TOMs.

3.7 Location and time-based data management

Kinetica is designed from the ground up to harness the power of the GPU for large-scale analytics on streaming geospatial-temporal datasets. It does this by spreading spatial computations across thousands of GPU nodes, across multiple cards and machines.

Kinetica's architecture short-circuits and eliminates the middle tier, and thus there is no time wasted moving data between the database and visualization engine. Instead, Kinetica's design allows executing complex geospatial filters and analytics directly at the database tier, and data is rendered on the fly server-side, using the internal geospatial web server.

With over **80+ functions** for operating over geospatial data, including support for points, polygon shapes, WMS rendering, tracks, and labels, Kinetica makes it possible to process and visualize geospatial vector data at high speed.

3.8 Machine Learning Pipeline

For many organizations wrestling with the complexity of data generated by business apps and new data sources, machine learning (ML) has emerged as a viable strategy to find actionable data insights by automatically uncovering patterns, anomalies, and relationships. The applications are wide-ranging, from autonomous robots, to image recognition, drug discovery, and fraud detection. With Kinetica, you can simplify and accelerate the entire machine learning pipeline from data preparation, to model training, to model serving, all within a single GPU-accelerated platform that can simultaneously process and manage BI and AI workloads.

For data ingestion, Kinetica provides connectors that can seamlessly integrate with pre-existing systems and ingest millions of rows of data in parallel within seconds. Training is typically the most resource-intensive step in the machine learning pipeline, but due to the abundant computing power of GPUs, the models can be trained quite quickly. To start model training, Kinetica provides a user-defined function (UDF) framework that is highly extensible and flexible, to integrate with open source machine learning libraries such as Caffe, Torch, and MXNet. Additionally, data does not need to depend on intricate indexing and down-sampling, but can be prepared interactively to accelerate training.

.

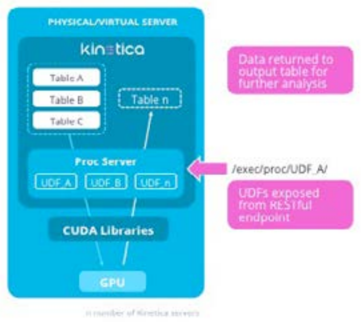


Figure 12 : User-defined functions for machine learning in Kinetica

UDFs have direct access to CUDA APIs, and can take full advantage of the distributed architecture of Kinetica. As shown in Figure 12, UDFs are able to receive filtered data, perform arbitrary computations, and then save output to a separate table.

Kinetica operationalizes machine learning by making the models available for scoring input data in the same database used for data generation and model training. With bundled TensorFlow, models can be scored in-line, optimizing for speed and better predictions.

3 /

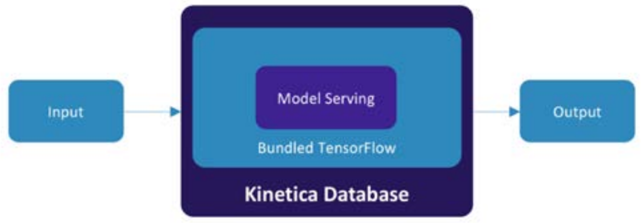


Figure 13 : User-defined functions for machine learning in Kinetica

The parallel computing power of the GPU in Kinetica delivers fast responses and is well-suited for the types of vector and matrix operations found in machine learning and deep learning systems.

3.9 Unstructured data

Building rich applications requires support for rich unstructured data. Kinetica's file system, KiFS, supports storing unstructured data, like, for example, images and video data required for model training or inference. Unstructured data stored in KiFS can get mounted, and is accessible via UDFs that can process this data for machine learning or analytical purposes. To learn more about KiFS in Kinetica, check out the **documentation**.

3.10 Visual insights with Kinetica Reveal

Business analysts need to visualize and interact with data elements to make faster decisions in real-time. Kinetica's integrated visualization dashboard, **Reveal**, enables interactive visual discovery optimized for temporal and geospatial analysis.

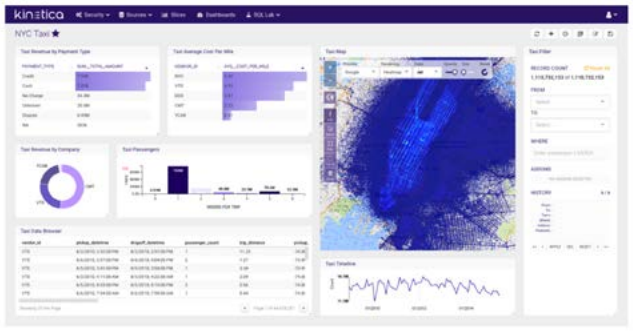


Figure 14 : Reveal dashboard with the NYC Taxi dataset

Reveal allows users to quickly build charts and graphs in a meaningful way to explore their datasets. With Reveal, you can simply drag and drop large data tables to slice and dice data and start producing on-the-fly analytics based on different filter criteria. If you still prefer writing SQL queries, Reveal also has a built-in SQL editor to support specific and custom queries.

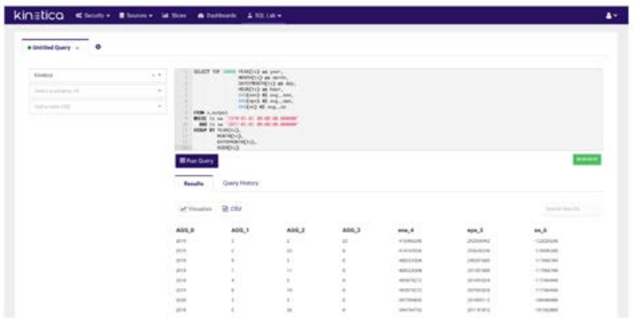


Figure 15 : Reveal dashboard with the NYC Taxi dataset

3 /

Out of the box, Reveal comes with a rich set of charts, diagrams, and map visualizations based on industry standards such as D3 and OpenLayers. For example, you can see your data in the form of tables, treemaps, heatmaps, directed-force diagrams, line-graphs, histograms, and more. Each slice or widget is connected to a single table and single chart type. You can mix and match multiple widgets in the dashboard, each providing different insight, and which can be added, positioned, sized, and arranged in any desired way. The dashboard can also be shared with other users that have access and refreshed automatically after a certain time interval.

3.11 Integrating with Visual Reporting Tools and GeoSpatial Tools

Kinetica supports a wide range of reporting, business intelligence and location-based analytical tools. The ODBC / JDBC drivers can be used to connect tools like Tableau, Tibco Spotfire, Microsoft PowerBI etc. or ETL tools like Informatica. The Geospatial WMS APIs can be used to build custom application using ESRI and MapBox APIs. In addition, the native REST API can be used to connect to any tools that consume JSON or can be used to build custom applications.

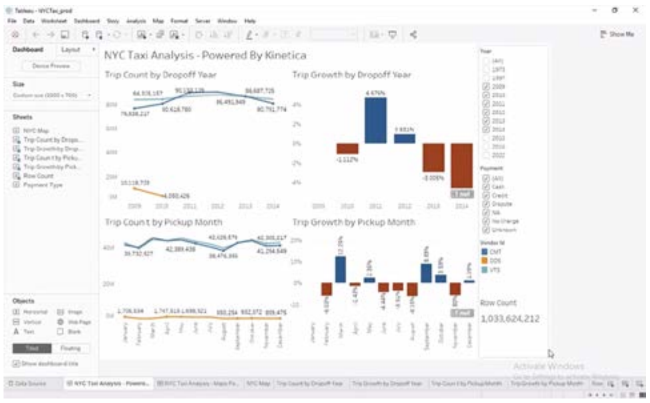

NYC Taxi Analysis - Powered By Kinetica
Trip Count by Dropoff Year
Trip Growth by Dropoff Year
Trip Count by Pickup Month
Trip Growth by Pickup Month
Row Count
1,033,624,212

Figure 16 : Business Intelligence using Tableau powered by Kinetica

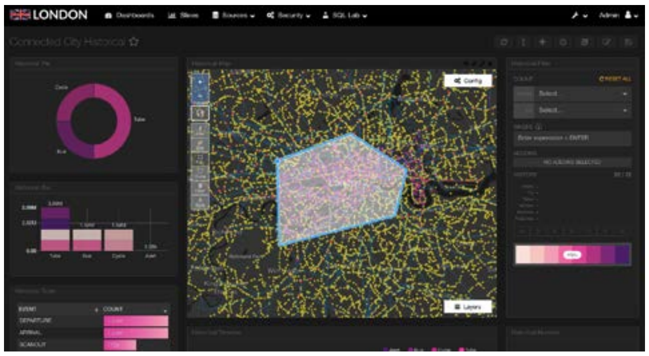


Figure 17 : Custom application using Kinetica and MapBox

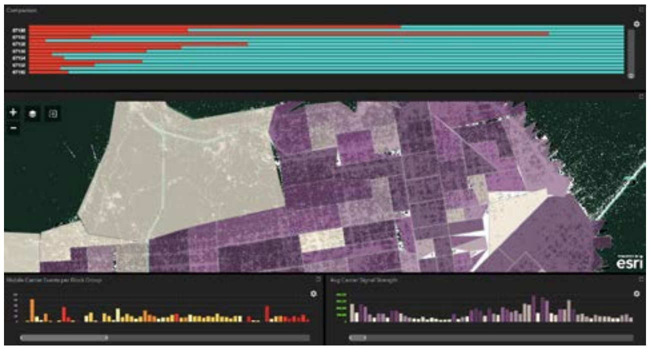


Figure 18 : Custom application using Kinetica and ESRI ArcGIS

4 / Administering Kinetica

4.1 Graphical administration application

Kinetica's administration application (called **Kinetica Admin**) is a dashboard that provides a centralized view of cluster functioning and any areas that need user attention. It displays cluster information such as status, number of connections, requests per second, number of tables and rows, and cluster uptime. It also highlights important information about resources across all the machines in the cluster, including CPU, memory, and disk usage.

Figure 19 : Kinetica's graphical administration application

Kinetica Admin helps you to drill down when needed, into several other areas such as cluster GPU ranks, administration and configuration settings, alerting, logging, and debug reports. By default, Kinetica admin runs on port 8080, accessed via the URL **http://<kinetica-host>:8080/**. To learn more about Kinetica Admin, check out the documentation **here**.

4.2 Securing Kinetica

Security is an important aspect of any modern data platform. To meet regulatory and compliance requirements, Kinetica supports several security features including authentication, authorization, auditing, and encryption.

Kinetica can be set up to run with or without client authentication. When client authentication is enabled through the cluster security configuration settings, user credentials are passed to the server for API, WMS, and ODBC connection requests. On the cluster, the HAProxy component receives these requests and passes them on to HTTPd, which is the preconfigured Apache HTTP daemon that runs on every server. HTTPd extracts the user credentials from the request, and validates them against the locally stored user credentials, or against the corresponding LDAP server. If authentication succeeds, the request is processed. If authentication fails, an error is sent back to the requesting client.

In terms of authorization, Kinetica's security model is similar to other SQL databases and has a role-based access control model. Users can be created internal to Kinetica, or stored externally in LDAP. Permissions are defined on resources such as individual tables, collections of tables, or the Kinetica instance itself. Roles can be created and assigned permissions, such that all the users who belong to a particular role have all the permissions assigned to the role. Roles can contain other roles, with the lower-level role inheriting the permissions of the higher-level role. The permissions a user has is the union of all permissions granted individually to the user and through roles. By default, Kinetica has pre-configured default users (admin, anonymous), and roles (public, authenticated) when it starts up.

In order to track "who does what, when, and how," Kinetica provides a complete 360-degree view of all database activity through full audit logging.

Audit activity is captured in JSON format files that can be set up to auto-rotate. To control the volume of audit records generated, the level of auditing can be configured to capture only http headers, http request body, or full audit data. After the audit records are written to file, third-party analysis tools can be used on the audit records to support information security tasks such as tracking user access, investigating suspicious activities, and validating access control policies.

Kinetica also supports on-the-wire data encryption over TLS, and encryption of user credentials at rest. To learn more about security in Kinetica, check out the documentation **here**.

5 / Conclusion

This paper describes the architecture of Kinetica, but the best way to get to know the technology is to try it out yourself. To test-drive Kinetica, visit **kinetica.com/trial/**

Kinetica easily integrates with your existing data infrastructure, and provides you with:

- A high-speed, in-memory insight engine
- Real-time analytics for streaming data
- Accelerated performance from your existing analytics tools and apps
- machine learning models applied to your data, so you can derive deeper insights

Kinetica drives your business from data-informed, to data-powered.

The journey to a fully operationalized AI capability begins with a single step toward the right infrastructure. Exploring the potential impacts that a next-gen analytical database could have on your business doesn't require a massive investment. Kinetica offers a **free 90-day trial** of its industry-leading solution, which can be deployed in the cloud or on commodity hardware for your POC.

Download free trial

About Kinetica

When extreme data requires companies to act with unprecedented agility, Kinetica powers business in motion. Kinetica is the instant insight engine for the Extreme Data Economy. Across healthcare, energy, telecommunications, retail, and financial services, enterprises utilizing new technologies like connected devices, wearables, mobility, robotics, and more can leverage Kinetica for machine learning, deep learning, and advanced location-based analytics that are powering new services. Kinetica's accelerated parallel computing brings thousands of GPU cores to address the unpredictability and complexity that result from extreme data. Kinetica has a rich partner ecosystem, including NVIDIA, Dell, HP, and IBM, and is privately held, backed by leading global venture capital firms Canvas Ventures, Citi Ventures, GreatPoint Ventures, and Meritech Capital Partners. For more information and trial downloads, visit **kinetica.com** or follow us on **LinkedIn** and **Twitter**.

kinetica

Headquarters

101 California St, Suite 4560
San Francisco, CA 94111

(888) 504-7832
+1 415 604-3444

To learn more

visit us at
kinetica.com

or
contact us