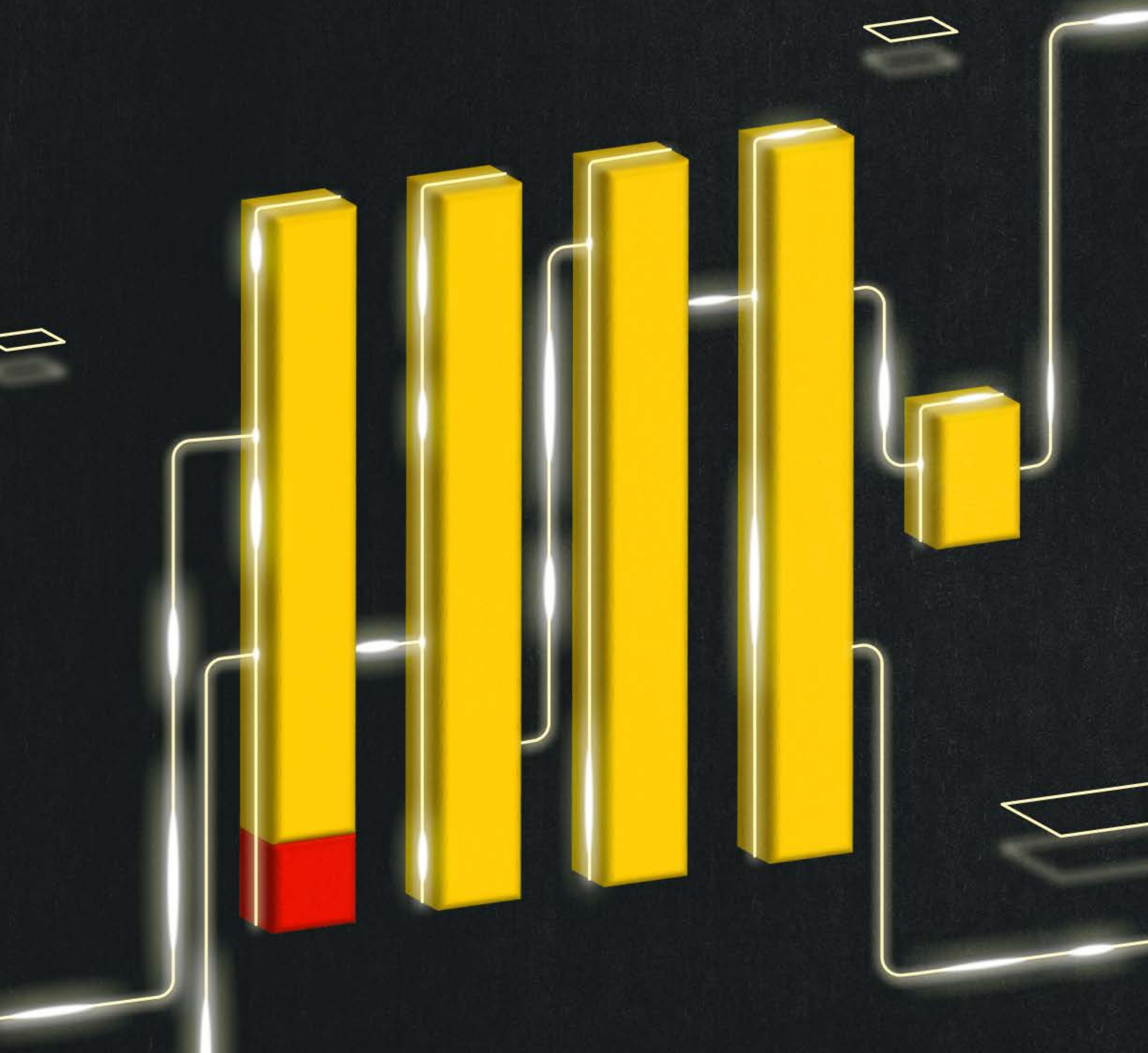
DoubleCloud

The complete ClickHouse® eBook



DoubleCloud's ClickHouse® eBook

Everything you ever needed to know about ClickHouse — from start to finish!

Read through DoubleCloud's complete ClickHouse eBook to begin your journey with ClickHouse. Here, we have gathered and organized all the essential resources you need to breeze your way to production.

We have developed this ultimate ebook for anyone looking to build a solution based on ClickHouse. Inside, you will find a plethora of exciting use cases, valuable tips, and tricks, as well as best practices from our solution architects.

And at the end of the guide, a **pleasant bonus** awaits you from DoubleCloud.

Table of contents

What is Clickhouse?

ClickHouse is one of the fastest online analytics processing **(OLAP)** database engines. It was initially developed by Yandex, Russia's largest technology company, in 2009 to power its Yandex.Metrica platform. With time, ClickHouse demand and user-basing grew, and it was finally open-sourced in 2016 under the Apache 2 license.

ClickHouse is a highly-scalable and fault-tolerant database management system **(DBMS)** built on a column-oriented structure. While traditional DBMSs store data row-wise, Clickhouse packs information in columns. The columnar structure allows ClickHouse to retrieve select columns with billions of rows **100x faster** than other systems. Moreover, the DBMS also uses data compression algorithms, parallel processing, and all available compute resources to process **billions of rows in milliseconds.**

ClickHouse is the one-stop solution for all analytical use cases requiring real-time performance. The open-source platform is supported by various cloud services allowing it to run on your favorite platform. It also has a vast community and third-party support offering useful extensions for language support and data ingestion from various platforms.

Where ClickHouse stands out

The IT space is filled with data-centric cloud platforms like Snowflake, AWS Redshift, and Google BigQuery, but ClickHouse sets itself apart with its distinct features and benefits.

1. Ease of implementation

Modern technology focuses on user experience. ClickHouse follows suit by offering an easy setup procedure, requiring only a few terminal commands for a fully functional cluster.

2. High performance

The ClickHouse platform is built for real-time analytical scenarios. The multi-node cluster setup and the column-oriented structure allow it to run most queries up to 100x quicker than traditional solutions. The high-performance capabilities make it ideal for real-time business reporting and analytics.

3. Linear scalability

ClickHouse offers a scalable environment for all workloads. The auto-scaling feature automatically increases or decreases the allocated resources depending on workload requirements. Moreover, users have the flexibility to set the **minimum** and **maximum** memory to utilize during operation.

4. Cost efficient

Having deployed on cloud services like AWS and GCP, ClickHouse follows the **"Pay as you go"** model. This means that users only pay for the resources that they utilize. Since ClickHouse auto-scales its resources, users only pay the bare minimum for small workloads. Furthermore, the flexible architecture scales compute and storage resources separately, making ClickHouse highly cost-effective.

5. Open-source

ClickHouse is an open-source platform that can be implemented on any cloud service. It does not bind users to a specific environment. Moreover, the project is backed by a large community of developers who continue to improve it and support those in need.

6. SQL-based language

The ClickHouse query language is based on SQL and is identical to the ANSI SQL standard. This is vital since most developers are familiar with SQL and can easily transition to ClickHouse. The query language includes the most common functions, including **`GROUP BY`, `ORDER BY`**, and **`JOIN`**, making executing complex commands possible.

ClickHouse scenarios and use cases

ClickHouse's amazing architecture and manifold benefits make it ideal for several practical scenarios. Industry leaders trust it for monitoring, logging, analytics, and general data processing tasks. Let's look at the industry-wide use cases.

Observability and monitoring

Many tech industry giants generate millions of data rows daily. A large chunk of this data consists of logs required to monitor running applications for smooth operations and error handling. Surfing through these log metrics requires a fast and scalable database engine like ClickHouse. ClickHouse's blazing-fast query performance allows engineers to spin up relevant data quickly and build dashboards with real-time updates for monitoring and analysis.

1. Uber

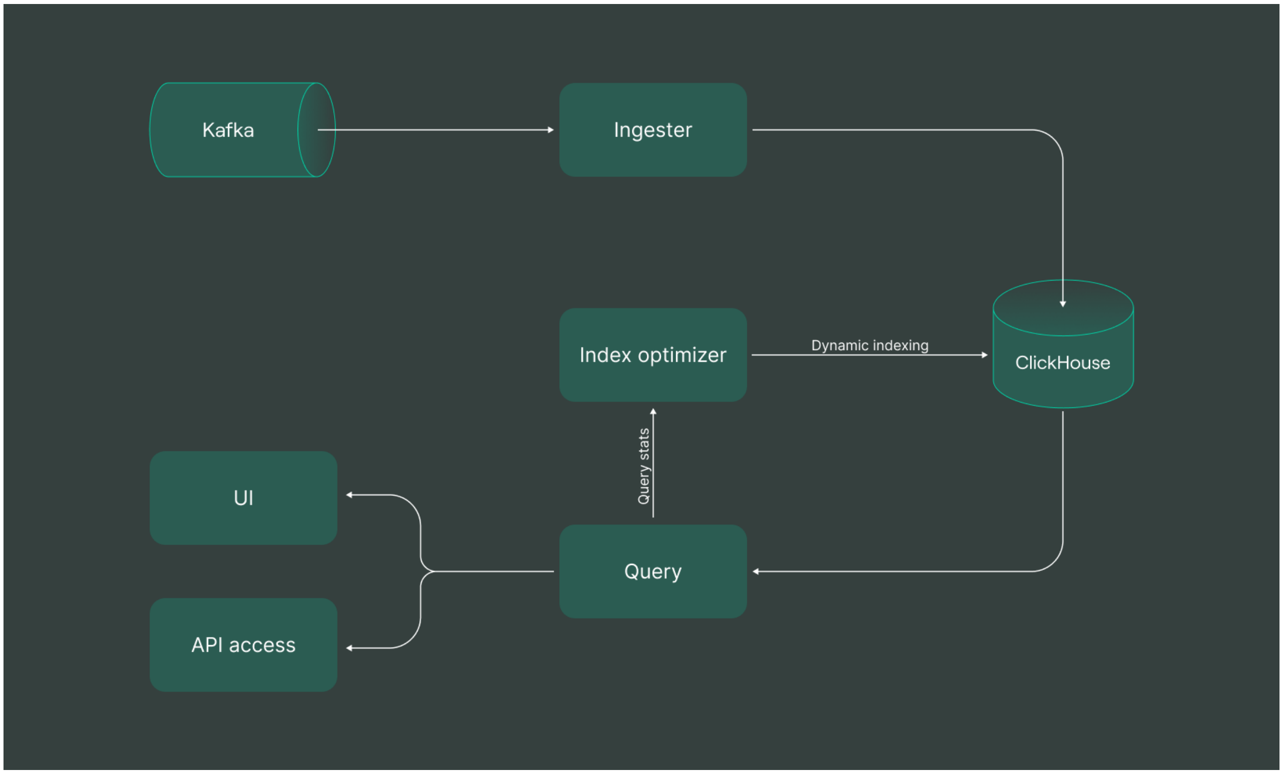
Background: Uber has operations all across the globe, and the platform comprises *"**Thousands of Services** emitting **hundreds TB** logs per **day**"* [1] . The logs are queried for real-time debugging, troubleshooting, and analytics. The procedure requires running

[1] Fast, Scalable and Reliable Logging at Uber with Clickhouse

hundreds of complex queries per second. The logs data is also schema-free, i.e. it can take multiple shapes and forms and requires a solution that can handle the dynamics.

Objectives: Uber requires a fast, scalable, and fault-tolerant database engine. This engine has a low-latency throughput, lowers query-running costs, and improves the engineer's productivity. The engine must also handle schema-free data and be scalable to handle large workloads, two primary aspects where solutions like ElasticSearch suffer.

Architecture: The overall architecture includes Apache Kafka and flattened JSON logs for capturing real-time events. These are then fed to appropriate ClickHouse tables.



High-level System Architecture

The tables are dynamically indexed i.e., only the most used columns (approx. 5% of the total) are indexed. This results in significantly fast querying while eliminating unnecessary indexing operations.

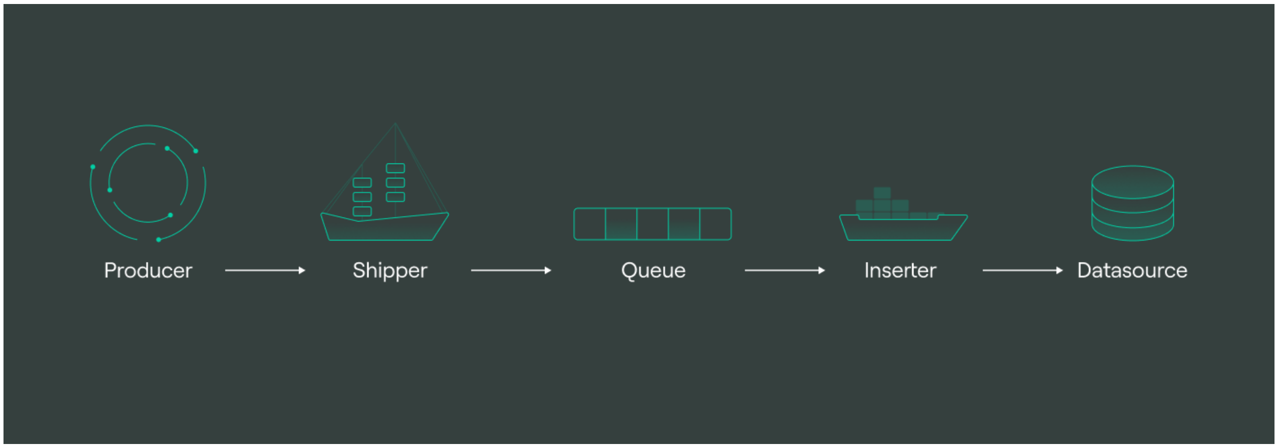
Outcome: The result is a robust architecture ready for all unforeseen challenges. It boasts a schema-free ingestion system and a single ClickHouse node can ingest **300,000 logs per second** (10x greater than an ElasticSearch node). Moreover, Uber managed to reduce its hardware costs by half[2] with the help of this efficient architecture, and all tasks are handled with a single ingestion pipeline for each region.

2. Cloudflare

Background: Cloudflare collects log information for all erroneous requests received. They used ElaticSearch containers to process analytics for all the log files for years. However, with growing volume, CloudFlare started experiencing slow query performance and high resource consumption, becoming a bottleneck in their day-to-day tasks.

Objectives: The aim is *to provide fast query support over this huge amount of data and to achieve all this without increasing the cost* [3]. Cloudflare required an alternative to ElasticSearch, which they found in the form of ClickHouse. On top of its fantastic query performance, ClickHouse also offered benefits like excellent multi-tenant support and a schema-free design.

Architecture: The overall architecture resembles a streaming pipeline consisting of producers, messages queue, and ending on a database engine. The diagram illustrates it better.



ClickHouse Architecture for Streaming Logs

The streaming pipeline is managed by Apache Kafka, which creates message queues for different logs. The queue then utilizes an ***Inserter*** to ingest logs into the ClickHouse platform. The inserters enable efficient data insertion using the **Cap N Proto** data exchange format while ingesting.

Outcome: By migrating to ClickHouse, Cloudflare could maintain a larger workload on a significantly smaller cluster. The CPU and memory consumption dropped by 8x while the

[2] Fast and Reliable Schema-Agnostic Log Analytics Platform

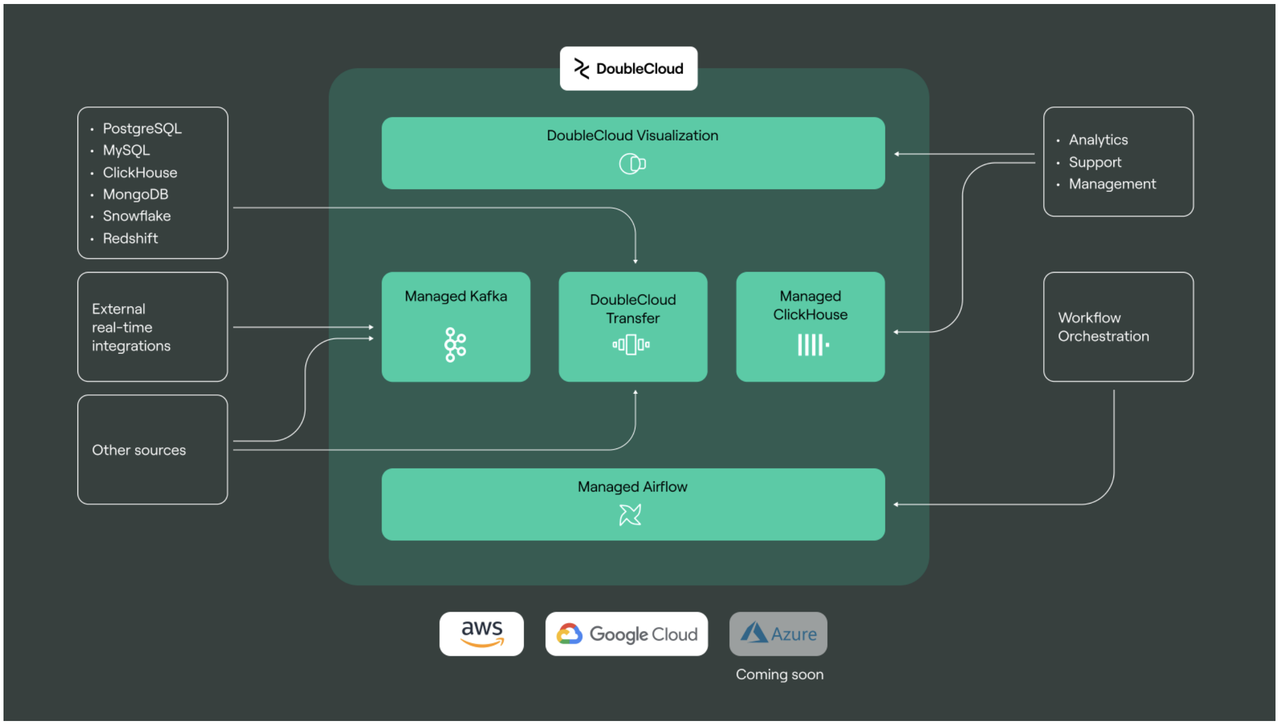
[3] Log Analytics Using ClickHouse

per document storage came down from 600 bytes to 60 bytes. The additional storage and system efficiency allow Cloudflare to run its operations faster and with lower costs.

3. DoubleCloud

Key Points: ClickHouse is a key part of DoubleCloud's data observability and analytics solutions. DoubleCloud Managed ClickHouse service offers a scalable, low-cost data solution that is up and ready within a few minutes. The service offers ClickHouses fast analytics capabilities and additional connectivity to external sources like MySQL, PostgreSQL, or Google Ad services.

Architecture: The overall infrastructure of the DoubleCloud solution includes an ingestion module using DoubleCloud Transfer and Apache Kafka data streaming. The data is then dumped into ClickHouse, which is used for Analytics, Support, and Management.



Due to the open-source nature of the platforms, the architecture can be deployed across various popular clouds.

Benefits: The overall solution comprises open-source, efficient, and scalable modules that make it ideal for large-scale deployments. It can handle 1 PB of uncompressed logs and 500k+ inserts per second. The infrastructure runs on 10x less hardware than other solutions making it a cost-effective option for all your analytical needs.

Real-time analytics

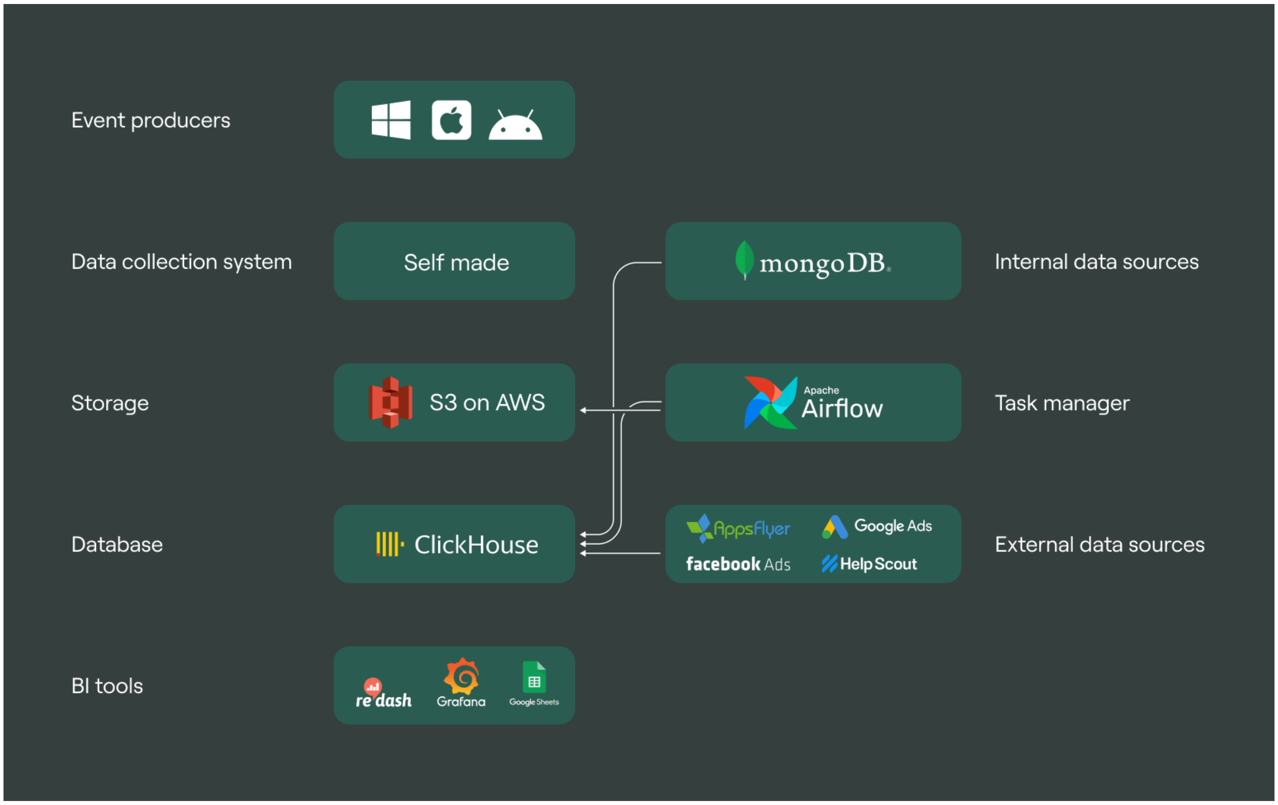
The top databases for end-user analytics in 2023 utilize large real-time data streams and terabytes or petabytes of current information, providing quick results and highlights when matched with modern embeddable visualization tools. ClickHouse stands out from other paid tools, as it can deliver results without significant costs. Even for the most demanding applications, it is an open-source database without hidden expenses or paid upgrades, giving complete access to everything.

Analyzing large data sets for business intelligence (BI) analytics can now be done quickly and affordably without time-consuming batch-driven processes. Key advantages of ClickHouse include scalability for real-time or static data volumes, production capabilities that surpass all traditional databases and stream processors, ease of use thanks to SQL, and integration with many commonly-used ingestion and reporting tools.

1. FunCorp

Objective: The FunCorp application records 3.5B in-app events every day. The requirement is to build a system that can process terabytes of data and obtain valuable information in real-time. These real-time analytics provide metrics that describe business operations and are vital for growth. The system architecture should be robust to process 700 GB of data daily, be fault-tolerant, and allow self-recovery in case of failure. The essential requirement is for the system to process all data with ultra-low latency (near real-time scenarios)

Architecture: The application architecture consists of end-user applications that are the main data producers. These applications are spread across various operating systems (OS) platforms, such as Android and Windows. The events are initially stored in an S3 bucket on AWS, from where they are ingested into ClickHouse. ClickHouse also ingests data from additional sources such as Google Ads and MongoDB.

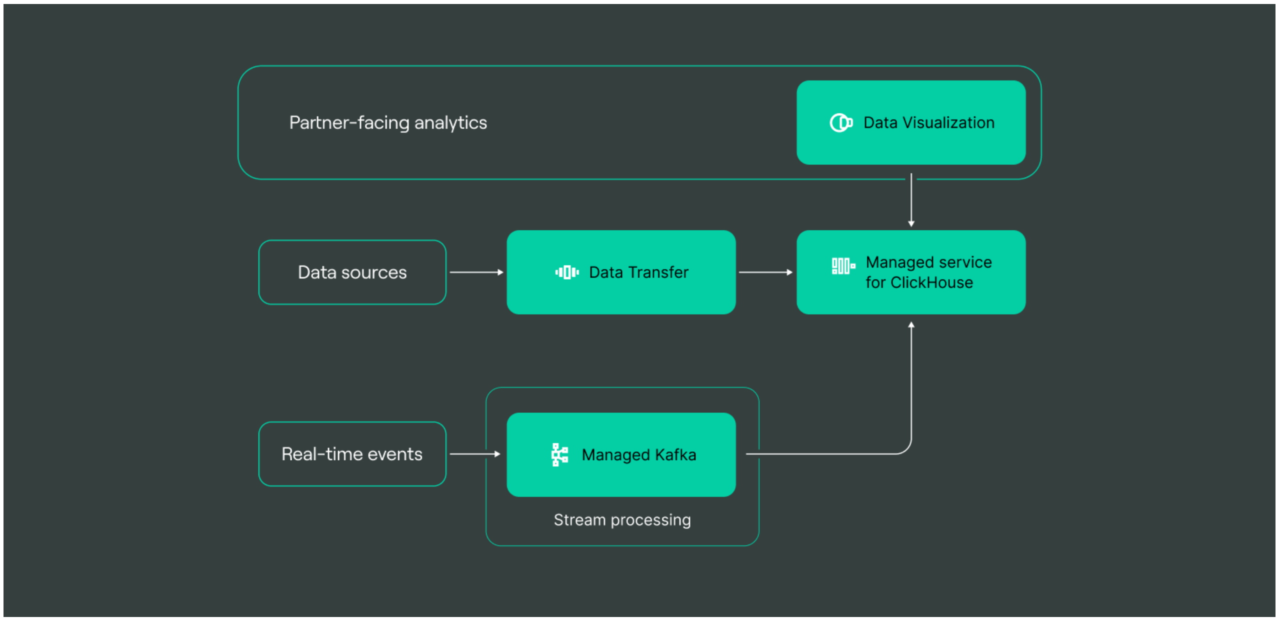


Outcome: Before opting for ClickHouse, FunCorp used RedShift to store and handle data events. The choice of ClickHouse was not easy and was made through extensive testing. It was contested against data processing engines like Vertica and Greenplum for query processing speeds. ClickHouse topped the charts with speeds up to 18x better than its competitors. Moreover, it proved a cost-effective alternative, using fewer resources to perform high-profile tasks.

2. Yango Tech

Objectives: Yango Tech has a wide network of 500 connected shops amassing GMV ARR over $900m. Their legacy solution was built around batch processing which limited updates to once a day. The delayed updates hinder business growth and client satisfaction. The updated architecture aims to deliver an aggregated solution for real-time business analytics.

Architecture: The final architecture consisted of a ClickHosue cluster hosted on the AWS platform. A data transfer solution is implemented to transfer information from ERP and WMS tables to the ClickHouse environment. The visualizations are implemented via Data Visualization tool.



Outcome: Yango Tech has recently launched an MVP partner analytics solution in just one week. The solution, which is based on DoubleCloud's ClickHouse technology, has proven to be 20% more cost-effective than its predecessor. The new architecture is both simple and powerful, making it ideal for processing large volumes of data in real-time scenarios. This makes it perfect for in-app analytics and creating reports that can help improve business models.

Data processing

A database system's primary judgment lies in its ability to process data. Large corporate organizations host data worth several terabytes and require efficient systems to process it. The processing involves complex query execution to apply transformations and extract vital insights. ClickHouse’s SQL-based language allows data engineers and scientists to perform all necessary operations within a few seconds.

1.Comcast

Comcast is one of the leaders of telecommunications in the United States, offering services such as the Internet, digital television, and telephone. They have developed a similar CDN management system within the open-source project Apache Traffic Control to handle their vast amounts of data.

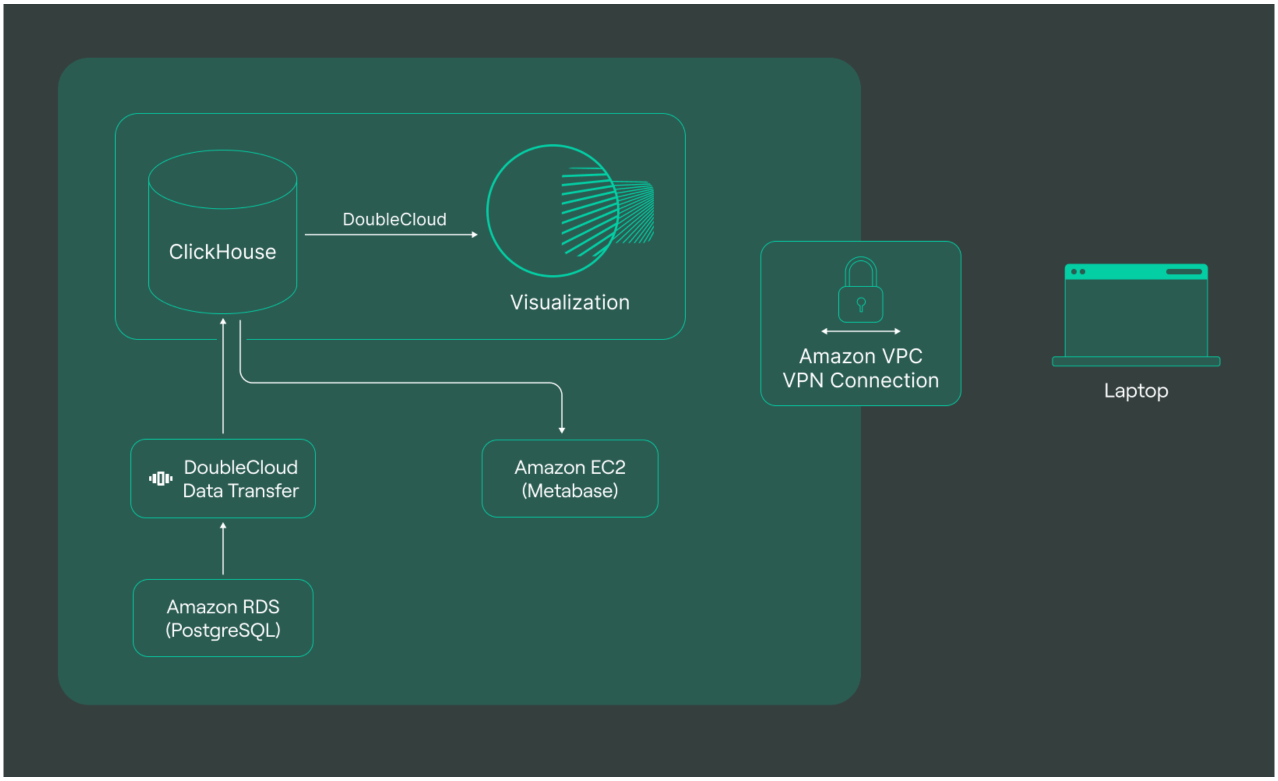
As a Fortune 100 telecommunications organization, Comcast hosts operations across the globe and has data pouring in from all ends. It uses ClickHouse to process several million log events per second, replacing its previous solutions using ElasticSearch and Splunk.

2. Loja Integrada & Pagali

The internet boom introduced us to the concepts of eCommerce and E-retail stores. With growing popularity, eCommerce platforms process several thousand orders per second. Loja Integrada was launched in 2013 and is Brazil's largest eCommerce platform. The platform is constantly growing, with 1300 agencies and partners in the ecosystem and 40,000 new stores created monthly.

Objective: Pagali is Loja Integrada's financial management solution with thousands of active users and monthly transactions worth Millions. It is responsible for 45%[4] of Loja Integrada's transactions and requires a fast storage and data processing solution. The previous solution with PostgreSQL could not match the data load and experienced frequent crashes. The team required an efficient DBMS that could instantly query logs for BI analytics.

Architecture: Loja Integrada opted for a DoubleCloud-managed ClickHouse-based solution. ClickHouse was the main storage for the entire data load, while a DoubleCloud Transfer pipeline is established to access data from auroraRDS. The dashboards were developed with the DoubleClouds visualization tool keeping ClickHouse as the source of all information.



Outcome: In addition to the massive gains brought to the eCommerce platform, the new solution has also increased the storage time of data required for investigating financial fraud crimes by six times, up to 90 days. ClickHouse's efficient system has not only increased performance six times and reduced the database size by half, but also cut the cost of operation and ownership by half due to a drop in resource and space utilization.

[4] Loja Integrada & Pagali switched to DoubleCloud for processing a massive amount of financial data

Data catalogue for National Science Data Fabric (NSDF)

The NSDF catalog contains Petabytes of information related to scientific research being conducted all around the globe. The purpose of building this repository is to maintain a centralized location for querying research-related information without any limitations or siloes. A database of this sort requires a lightning-fast scalable querying engine like ClickHouse. Moreover, since each query will have to scan across billions of records, the database engine must be cost-effective in storage and processing to make the overall infrastructure financially feasible.

The catalog is built in partnership with DoubleCloud, a National Science Data Democratization Consortium corporate partner, and is reliant on ClickHouse for all its major data processing operations. The column-based database checks all boxes for this specific use case, including

- Quick data insertions,
- Almost real-time aggregation scans,
- Native support for JSON documents for metadata storage

The architecture also outshined its competitors in extensive testing, completing insertion operation for 400,000 records in only 3 minutes and producing*the highest throughput for long queries and the lowest latency for short queries.*

Gaming

Over the past years, Gaming has emerged as a multi-million dollar industry with millions of daily concurrent users. Data analytics is an essential component of the gaming industry, with game developers utilizing data to enhance the quality of their storylines, provide customized in-game experiences, expand promotional activities, and optimize revenue generation.

With roughly 1 billion[5] active players in 2023, the gaming industry generates several Terabytes of data daily. This includes logs, player statistics, in-game information, financial transactions, etc. This kind of data requires a robust infrastructure that can handle the growing volume of information.

Due to the high demand for flexibility, speed, and scalability, many game development studios opt for ClickHouse as their primary data processing engine. ClickHouse's OLAP capabilities allow developers to capture real-time information and draw insightful analytics.

[5] Global digital gamers by segment 2027 | Statista

Beetested

Beetested is a start-up founded in 2022 that helps game development studios improve their products by analyzing player emotions. The platform records and analyzes players' emotions in real-time while engaged in the game. This approach helps them understand how players feel about the game, and this information is used by the development studio to enhance user experience.

Challenges and solution: However, data collection and analysis turned out to be a ***significant challenge*** . The application recorded 20 to 50 videos per game which amounted to ***millions of frames*** per game. The team opted for the ClickHouse engine (*DoubleCloud's managed service for ClickHouse, to be more specific*) for storing such a massive volume of information. The overall flow captures video frames which are then transformed by the Machine Learning team into representable information. The final form is stored in ClickHouse, from where it is accessed for analysis and representation.

Outcomes: Besides being the ideal infrastructure for such data volumes, DoubleCloud's solution came with additional benefits. Firstly, the managed service allowed the team to skip the hassle of the cluster configuration and jump straight into the analysis. Secondly, the infrastructure integrated DoubleCloud Visualization solution to generate analytical dashboards. This replaced the team's legacy solution involving PDF reports, which took hours to compile.

ClickHouse — benchmarks

We have discussed the query performance and robustness of the ClickHouse database, but all these claims are baseless without appropriate evidence. The true performance of the RDBMS is judged from the numbers resulting from benchmarks.
ClickHouse is popular for

- Handling large volumes of data (Big Data Processing)
- Efficiency in data analysis: queries, reading, summarization, and other actions
- Low resource utilization
- Low space complexity

The benchmarks aim to test the DBMS for all these claims and compare results against alternative solutions. Some of the popular benchmarks are discussed below.

Billion taxi rides dataset benchmark

Mark Litwintschik, a data professional with 15 years of experience, has tested the DoubleCloud Managed service for ClickHouse database with a dataset consisting of information for approximately 1 billion taxi rides. The benchmark[6] dataset contains 56

[6] 1.1 Billion Taxi Rides in ClickHouse on DoubleCloud (marksblogg.com)

compressed files with 104 GB of data, which spans to 500GB when decompressed. The benchmark run tests data import times and runtime for various complex queries.

The dataset is inserted into ClickHouse via S3, and a preliminary benchmark was performed to estimate the insertion query performance and the data size inside the database. Initially, a row-based table was created, and the insertion operation took around **29 minutes and 25 seconds,** with the final data size of **144.8 GB** on EBS. The table was then converted into an optimized column-based format. This operation took **50 minutes and 43 seconds** and reduced the data size to **98.3 GB.**

Afterward, Mark ran some query operations that involved group by clauses, aggregations, and count operations. The exact numbers are summarized in the table below.

Query	Processing Time / s
GROUP BY - COUNT ALL	0.347
GROUP BY - AVG	1.1
GROUP BY - COUNT - TYPE CASTING	1.389
Complex Query involving various functions	2.935

Most of the tasks were completed in approximately 1 second, as the author himself comments,*"The first query's time is the fastest I've seen among any of the Cloud-exclusive offerings I've run benchmarks on".*

MariaDB ColumnStore vs. Clickhouse vs. Apache Spark

The benchmark was performed by Percona to analyze the performance of these systems against each other. The benchmark is run for the Wikipedia page counts dataset (26 Billion rows), and the system specs are as follows:

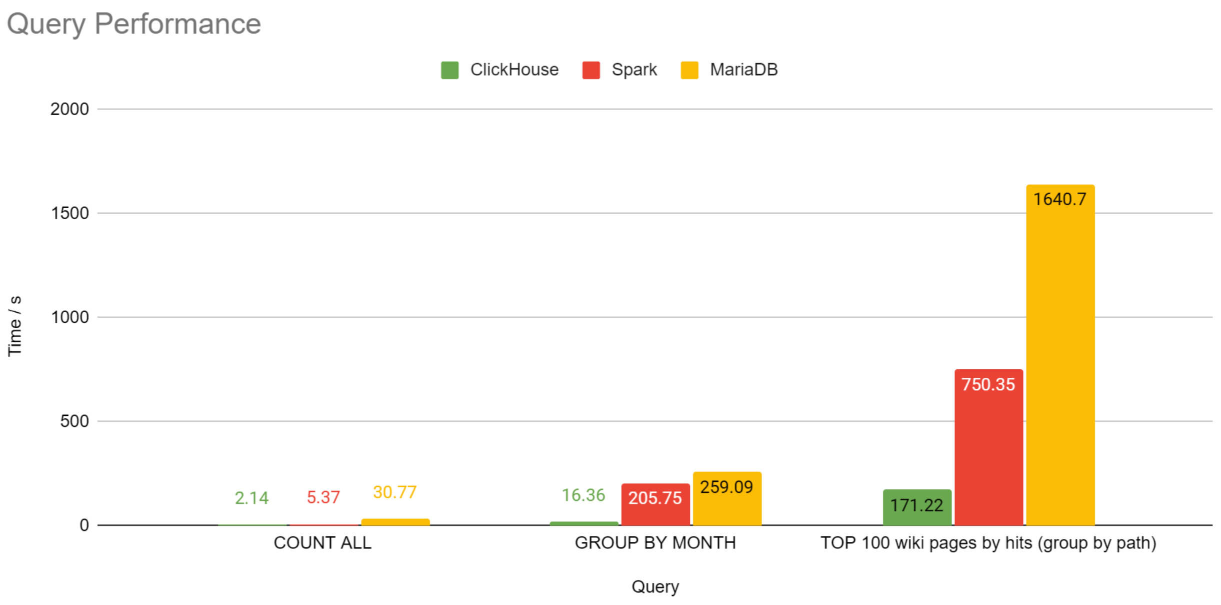
- **CPU:** physical = 2, cores = 32, virtual = 64, hyperthreading = yes
- **RAM:** 256Gb
- **Disk:** Samsung SSD 960 PRO 1TB, NVMe card

Each DBMS is run on a single-node setup with the most stable version of each program at the time of testing.

Query Performance

Multiple queries were tested across all the contenders and the results are summarized below.

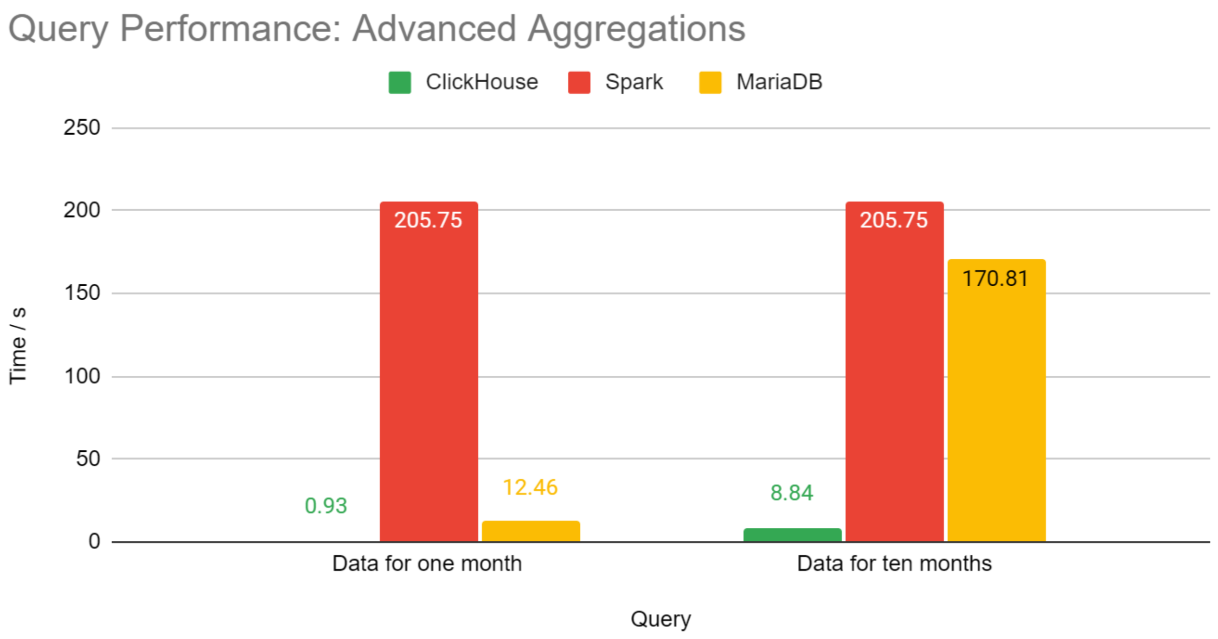
Query	ClickHouse / s	Spark / s	MariaDB / s
COUNT ALL	2.14	5.37	30.77
GROUP BY MONTH	16.36	205.75	259.09
TOP 100 wiki pages by hits (group by path)	171.22	750.35	1640.7



The above results demonstrate how ClickHouse outshines its opponents while processing some basic queries against billions of data points.

However, some additional tests were also conducted as part of the benchmark. These advanced queries grouped the data against **`month`** and applied a filter on the **`year`** column. The final result counted the total number of data rows. The query was run twice, once with returning data for 1 month and again for 10 months. The results are summarized below.

Query	ClickHouse / s	Spark / s	MariaDB / s
Data for one month	0.93	205.75	12.46
Data for ten months	8.84	205.75	170.81



ClickHouse once again proves its worth as a big data processing engine, with performances greater than 10x that of its competitors.

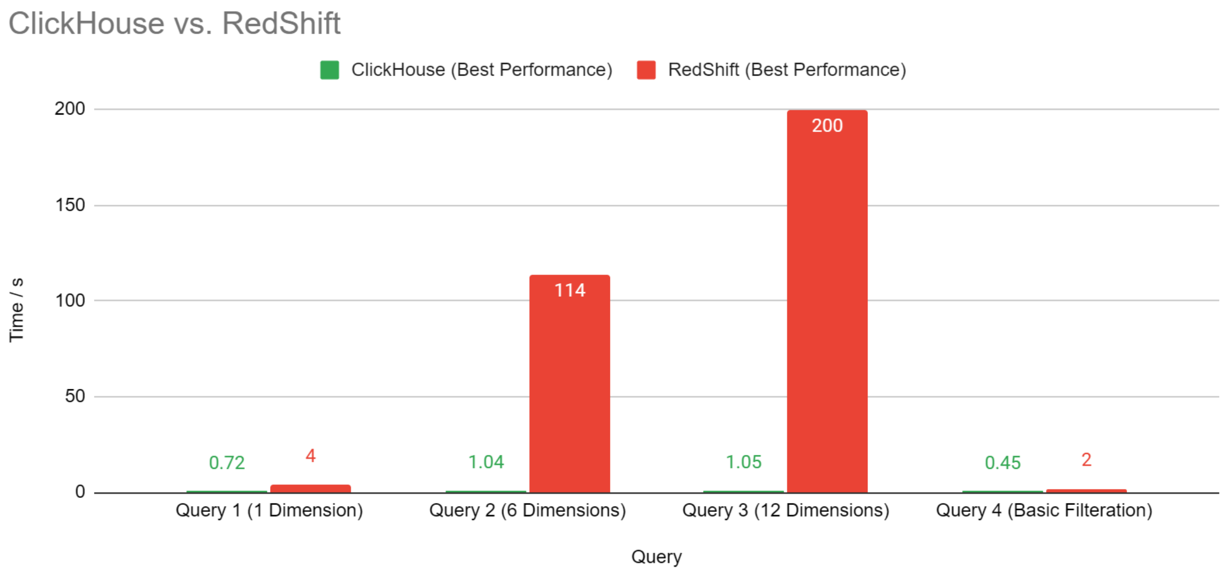
ClickHouse vs. RedShift: FinTech risk management

The benchmark[7] by Altinity aimed to compare the performance of ClickHouse DBMS against RedShift. The data for the run was custom-generated to mimic a typical FinTech dataset with 100 dimensions and P&L vectors with 1000 elements. It is a relatively small dataset and the benchmark aims to judge purely the computational capabilities of both our systems.

The tests conducted involved 4 different queries, each calculated the*maximum loss over a 95% confidence interval across an increasing number of dimensions* . The queries can be constructed using various functions but each gives different performance numbers. The original benchmark conducted experiments for different approaches but we will only discuss the most efficient ones from each system. The results are summarized below

Query	ClickHouse (Best Performance)	RedShift (Best Performance)
Query 1 (1 Dimension)	0.72	4
Query 2 (6 Dimensions)	1.04	114
Query 3 (12 Dimensions)	1.05	200
Query 4 (Basic Filteration)	0.45	2

[7] Database comparison [2020]: ClickHouse vs Redshift for FinTech | Altinity



Needless to say, ClickHouse obliterates the competition. Query 1 demonstrates an almost **5x** gain while Query 4 shows a **4x** better performance, but the true difference is seen in Query 2 and 3, where we see performance differences up to **200x.** While RedShift loses its performance exponentially with increasing data complexity, ClickHouse maintains its speed and emerges as the better engine.

ClickHouse vs. TimeScaleDB vs. InfluxDB (Time-Series Analysis)

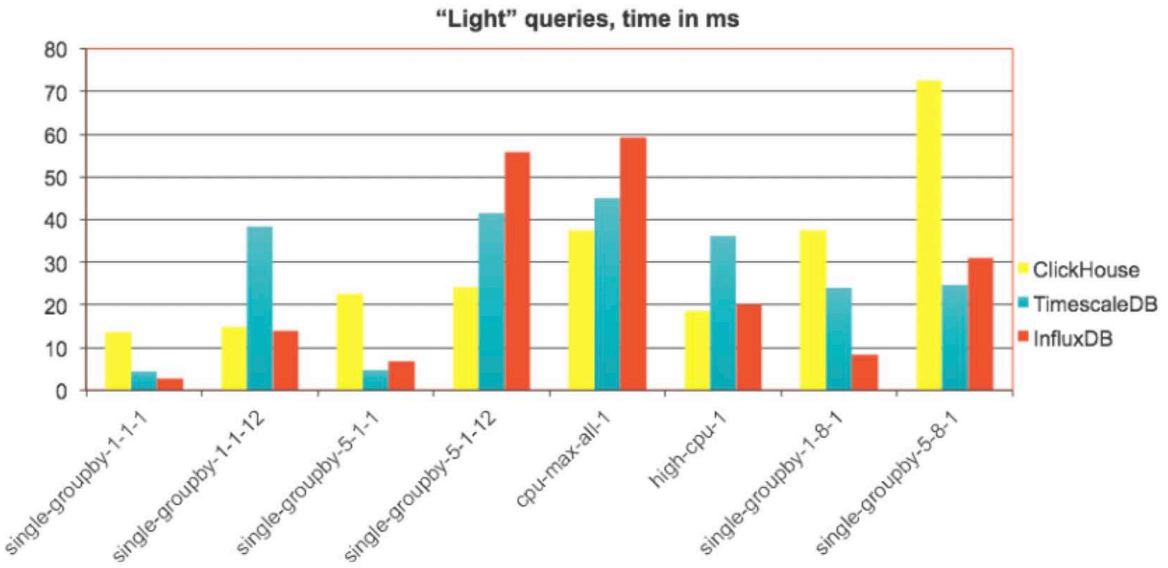
This benchmark[8] was conducted by Altinity to specifically quantify the performance of the DBMS against time-variate information. The test setup is summarised below

- Amazon r5.2xlarge instance, 8 vCPUs, 64GB RAM, EBS storage
- A dataset of 100M rows, 10 metrics per row, inserted in batches by 10K rows in 8 parallel workers
- 15 test queries, every query is run 1000 times in 8 parallel workers

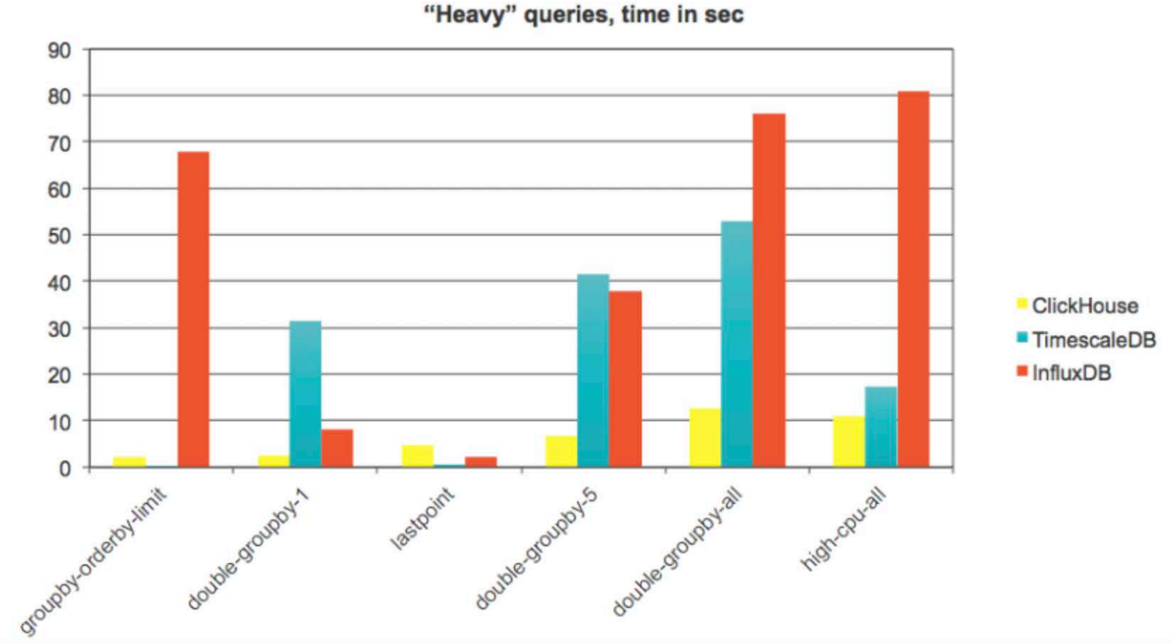
The data was generated using the Time-Series Benchmark Suite (TSBS). It contained 100M rows of CPU metrics data collected from 4000 devices. The 15 queries used for the testing contain various aggregations across different subsets of data.

The results are distributed into **'light queries'** and **'heavy queries'** where the former describes simple operations that take milliseconds (ms) to complete while the latter may take single-digit seconds or more. Results for both are shown in the charts below.

[8] ClickHouse Crushing Time Series – Altinity | The Real Time Data Company



Light Queries (source)



Heavy Queries (source)

The results here are quite interesting. We can see that in 7 out of the 15 scenarios, ClickHouse is the clear winner, but there is another analysis we can draw out of the charts. With lighter queries, ClickHouse is seen struggling against its competitors (sometimes by a large margin), but when we switch to heavy queries the tables turn. ClickHouse demonstrates consistent performance across the board for heavy queries, maintaining query times around 10 seconds.

All in all, DoubleCloud Managed ClickHouse demonstrated impressive performance against the specialized time-series DBMSs. It displayed comparable performance in most cases while outranking in other cases.

ClickHouse – how to start

Getting a DBMS up and running is a challenging task. It involves installing the framework, importing and maintaining data files, managing clusters and their scalability, and executing complex queries efficiently. Let's go through these challenges one by one.

Configuration and setup

Manual setup

The simplest way to get your ClickHouse cluster initialized is by downloading and installing the binary file on your OS. Natively ClickHouse runs on Linux, MacOS, and FreeBSD and on Windows using WSL. To download the binary file, run the following command in the terminal.

```
curl https://clickhouse.com/ | sh
```

The binary can be used to run the ClickHouse server, clickhouse-client, clickhouse-local, ClickHouse Keeper, and other tools. Now simply start the server with the following command

```
./clickhouse server
```

This will create all necessary files and folders in your working directory. The last step is to start the client.

```
.clickhouse client
```

The client will connect to the running server and will give you access to an interactive environment where you can access all your database files.

The quick install feature is an over-simplification of the actual setup procedure for those who wish to experiment with the infrastructure and remain in a development environment. For production-level use cases, the setup procedure is further complicated, with steps involving cluster setup and management and configuration files.

Advanced configuration

All configurations for ClickHouse are placed inside XML and YAML files. These files contain settings such as user profiles and access restrictions. By default, ClickHouse uses the `/etc/clickhouse-server/config.xml` file for loading all necessary configurations, but users can choose to specify their own files using the `--config-file=` or `-c` flag.

Stacking configurations

A good practice in ClickHouse is to stack various configuration files rather than create a single file for all your settings. The additional files are placed in the `config.d/` directory, created relative to your main configuration file. CllickHouse goes through all these configurations upon startup and adds them in a preprocessing step along with the main configuration. These stacked files are then read alphabetically and applied to the ClickHouse cluster.

Setting configuration this way preserves the original configuration file and groups similar configurations in a single file.

For practical scenarios, deploying ClickHouse can be complex, and DoubleCloud Managed Service becomes a much more viable option.

DoubleCloud Managed Service For ClickHouse

The managed service based on open-source sets up production-level data-ready clusters within 5 minutes. The following steps demonstrate the setup procedure.

The preliminary steps require you to have a DoubleCloud account for access to the console and the Docker daemon installed. When you have both these requirements fulfilled, you can go to your terminal to install the relevant software.

```
sudo service docker start
```

```
docker pull clickhouse/clickhouse-client
```

The above steps will start the docker daemon and download the latest `clickhouse-client` image. Next, you must log in to your ***DoubleCloud web portal*** and maneuver to the ***console***. From the console, follow the following simple steps

1. Move to the `Cluster Overview` page.
2. Click on **`Create cluster`** in the upper-right corner.
3. Select ClickHouse®.
4. Choose a provider and a region.
5. Select a suitable preset that fits your requirements.
    a. The present names define the CPU and RAM capacity and CPU chip.

   b. For example, `s1-c2-m4` means an AMD x86 chip (s1) with 2 CPU cores (c2) and 4 GB of RAM (m4).

6. Configure the amount of replicas and shards required for the data.
7. Under **Basic Settings:**
   a. Give the cluster a suitable **name.**
   b. Select the ClickHouse version the cluster will use.
8. With **Advanced settings**, you can
   a. Set the maintenance schedule for the cluster.
   b. Select the VPC to locate the cluster.
   c. Select the allocation of the ClickHouse Keeper Service (embedded or dedicated)
   d. Set the DBMS settings
9. Click **Submit.**

Notice how the entire procedure did not require maintaining config files or code execution. The Double Cloud Managed Service reduces the cluster setup to less than 5 minutes of button clicking. Next, you can generate the ***connection string*** for your running cluster and create a connection from your terminal using the following command.

```
docker run --network host --rm -it clickhouse/ Native interface connection string>
```

From here on, you can continue with database creation and querying.

Data migration

Migrating data is crucial when starting a new database system. It is vital to gain access to your existing databases quickly to get the system up and running. ClickHouse offers various routes and connectors to streamline the data migration process, reduce transition delays, and improve the time-to-market.

ClickHouse-local

`clickhouse-local` is a command line utility that runs the ClickHouse server in an isolated fashion. It is typically used to run ClickHouse in a lightweight environment for fast SQL processing, but it can also be used as a data migration utility.

The tool can be used if your existing database system is a ClickHouse-provided integration engine or Table Function (MySQL, PostgreSQL, MongoDB or SQLite) or has a JDBC or ODBC driver from the system vendor.

Third-Party ETL tools

ClickHouse supports various third-party connectors that facilitate the migration process. Some popular connectors include

- Airbyte
- dbt,
- Vector

These connectors can connect to almost any external data source and seamlessly transition data between the two systems.

DoubleCloud Transfer

Transfer is a service designed to easily migrate data between external databases and DoubleCloud's managed services. Its parallel execution capability enables Transfer to reach arbitrarily high throughput, limited primarily by the amount of resources assigned by the user and the capabilities of the source and target databases, providing an additional advantage over other migration services. The preliminary steps include:

1. Setting up a cluster using the managed service.
2. Initialize a database to store the migrated data.

Once the initial setup is complete, you will need to create a source and destination endpoint. Open the ***transfer*** page from the console and start by initializing the **source** endpoint.

1. Go to the **Endpoints** tab.
2. Click Create Endpoints and then select **Source.**
3. Select the **source** from the provided list (e.g., ElasticSearch, PostgreSQL, Snowflake, S3).
4. Configure settings for the endpoint. These would include
   a. Dataset.
   b. Path pattern.
   c. Data format (e.g. parquet).
   d. Additional configurations like bucket name in case of S3.
5. Click **Submit**

With this, the source endpoint is configured. Now you need to configure the destination.

1. Return to the **Endpoints** tab, and this time, click **Target** while creating the endpoint.
2. Give the endpoint a **name** and a **description.**
3. Configure endpoint parameters:
   a. Select the connection type as **Managed Cluster.**
   b. Specify the *username* and *password* for the cluster.
   c. Specify the destination database name (the one created in the initial steps).
4. Select **Drop** as the cleanup policy.
5. Leave the remaining fields blank and click **Submit.**

The last step is to activate a transfer. Go to the console, and in the list of services, click Transfer → Create Transfer. Next, select the source and destination endpoints set up in the previous steps. Select an appropriate name and description for the transfer process and select the transfer type from the list The `snapshot` type will be suitable in most cases) Click submit and the transfer will begin. Once the status is set to 'done', the database can be queried in ClickHouse.

Interested in learning more about commercial use cases for Data Transfer and implementation examples? Download our eBook 'Providing cross-system Data Transfer as a service' for free here.

DoubleCloud Bring Your Own Cloud (BYOC)

BYOC is DoubleCloud's solution for customers who wish to enjoy the luxury of managed services while retaining complete control of their data. Customers with existing AWS VPCs or Google Cloud projects can directly deploy DoubleCloud managed services in their private space. The solution uses external networks to transfer data between different cloud resources (clusters) and they are connected to the internet via secured connections. All configurations, network creations, and deployments are done under the customer's AWS VPC or project on GCP.

The BYOC solution is ideal for clients who require control of data due to GDPR compliances or simply just the flexibility to manage configurations themselves. Additionally, it's important to note that our BYOC solution can be used without any limitations in terms of the minimum monthly payment to your cloud provider.

Scalability

One of ClickHouse's key features is its flexible and scalable environment. This means that ClickHouse supports various clusters deployed across various nodes (servers). These multiple nodes contain replicated data files for fault tolerance and are used for load division during heavy processing.

DoubleCloud Hybrid Storage backed for cost efficiency

With Double Cloud, you can build data analytics at SSD storage speed and object storage prices, be it Amazon S3 or Google Cloud Storage. And using ClickHouse, you'll reduce your costs up to 5 times by decoupling the latest hot data to SSD, and the cold, less frequently used data to S3 or GCS. Learn more about →

Security and compliance

When switching to a new DBMS, data security is a primary concern. It is imperative for cloud providers to fulfill regulatory compliance and offer state-of-the-art security protocols for data protection.

Managed services for ClickHouse, such as that from DoubleCloud, are accompanied by industry-standard security infrastructure. DoubleCloud enables the fulfillment of the most advanced data security and reliability requirements. It holds various compliance certifications and boasts transparency regarding what data is collected and which is not. DoubleCloud holds customer information in the highest regard and ensures the highest security standard to maintain integrity and customer trust.

Monitoring and debugging

ClickHouse logs data for hardware resource utilization and server metrics. The hardware resource data is logged inside the `system.asynchronous_metric_log` table and contains utilization reports for CPU, RAM, network, and storage devices. These metrics can be used to debug hardware-related issues such as slow performance and database crashes. Additionally, ClickHouse also logs processor temperatures to diagnose overheating.

ClickHouse metrics are logged in the `system.metrics`, `system.events`, and `system.asynchronous_metrics` tables. These metrics include reports on how ClickHosue utilizes computational resources and common statistics for query processing, such as start and completion time. These metrics can be streamed to third-party visualization tools like Graphite and Prometheus. Log streaming is possible via an HTTP endpoint configured within ClickHouse configuration files.

Example configurations for Prometheus are as follows

```
<clickhouse>
   <listen_host>0.0.0.0</listen_host>
   <http_port>8123</http_port>
   <tcp_port>9000</tcp_port>
   <prometheus>
      <endpoint>/metrics</endpoint>
      <port>9363</port>
      <metrics>true</metrics>
      <events>true</events>
      <asynchronous_metrics>true</asynchronous_metrics>
   </prometheus>
</clickhouse>
```

The endpoint can be tested with the following terminal command

```
curl 127.0.0.1:9363/metrics
```

Complex queries and data modeling

As a ClickHouse user, your day-to-day tasks will revolve around data modeling and constructing analytical queries. It supports an SQL-like query language that is used for all analytical processing and supports the most popular functions like

- SELECT
- JOIN
- DELETE
- INSERT

However, even with the most optimized SQL structure, you may not get the desired query performance. This is because it is vital to optimally construct databases and table schemas when dealing with Clickhouse.

A good practice is to explore and understand your data before creating schemas. Additional knowledge of your data allows you to make technical schema-level decisions.

Data partitioning

ClickHouse allows users to partition data based on specific keys. Each partition is stored individually for easy access during execution. The partition key must be carefully selected, e.g., if you have a date column, you may choose it for partition since it is common practice to select data according to selective dates. Such decisions have a significant impact on performance.

Data compression

ClickHouse uses various algorithms for data compression, including LZ4 and ZSTD. It is important to pick out the optimal algorithm that balances the compression ratio and decompression speed trade-off. Applying appropriate compression saves space and helps the database engine perform better.

Interested in discovering more ClickHouse tips and tricks from industry experts? Check out our free on-demand webinar here.

Final words

ClickHouse is a blazing-fast database system that is specially constructed for analytical data processing and loTs of data. It is ideal for scenarios that require real-time analytics, and customer-facing and end-user analytics such as metric dashboards or business reports. ClickHouses column-based structure stores data in column partitions rather than rows. This design choice is perfect for analytical processing and allows the DBMS to pick out all the data for specified columns in milliseconds. Due to its amazing performance, it is used in various industries including telecommunication, gaming, and eCommerce platforms.

Getting started with the DBMS and formulating data architectures can also be a challenge sometimes. ClickHouse runs in a scalable environment and is deployed across a cluster of nodes. These nodes facilitate data replication for fault tolerance and load-balancing during heavy processing. An alternate method is to opt for ClickHouse cloud or DoubleCloud Managed Service for ClickHouse.