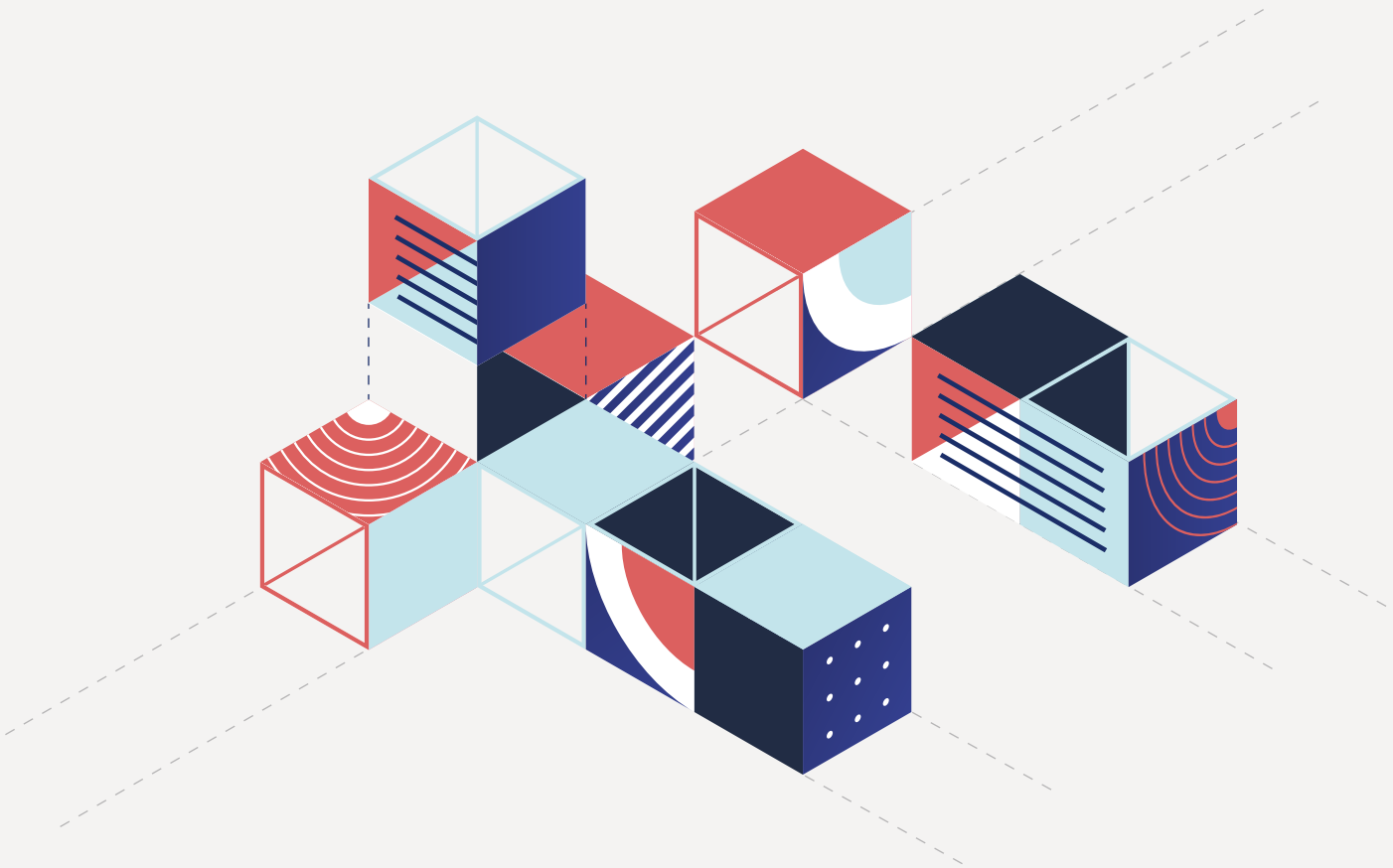
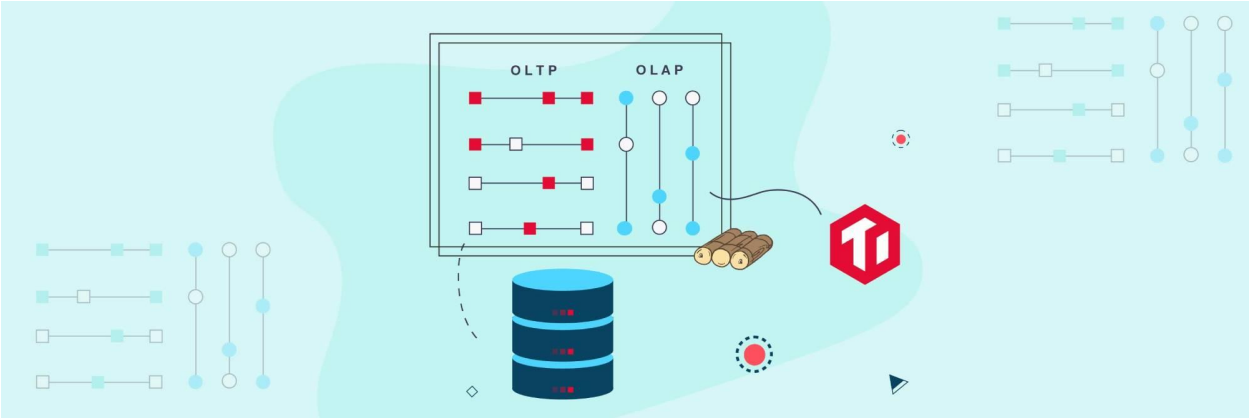
TiDB

TiDB: Making an HTAP Database a Reality



TiDB: Making an HTAP Database a Reality



Abstract

Every industry today aims to be data-driven. Businesses do not want to base their decisions on assumptions or intuition but on real data. They need to analyze large-scale data in real-time to identify potential risks, efficiently allocate their resources, and quickly serve their customers. However, the more data you hold in your DBMS, the longer it takes to retrieve and process it. As you amass more data over time, it's harder to process it in real-time.

Traditional IT systems are overwhelmed due to the vast amount of data volumes to be processed. Businesses need critical insights in minutes—not hours. In a typical architecture, businesses have both transactional and analytical workloads. Transactional systems are in the forefront of consuming fast data from sources generating gigabytes to terabytes of data, whereas analytical systems are used to perform advanced analytics on this massive incoming data, deriving value for the business.

However, a common challenge for such architectures is the separation of transactional and analytical systems which limits lots of business use cases. This is because the data for these workloads resides on separate clusters. An ideal system should be flexible enough to run both your transactional and analytical workloads. This is contrary to systems which load their transactional data and then move it to the analytical system via an extract, transform, load (ETL) layer. Even if data is copied to the analytical layer every 15 minutes, there is always a gap of at least 15 minutes before meaningful analytics can be performed.
This is where Hybrid Transactional/Analytical Processing (HTAP) becomes the most suitable processing architecture. TiDB, an HTAP database, empowers applications and business users to get real-time insights to underlying data, improve business agility, simplify underlying architecture, and reduce maintenance cost.

TiDB has helped businesses implement use cases through improved and real-time insights. Some of the popular use cases such as cybersecurity, credit and insurance fraud detection, logistics analytics, gaming analytics, and others run on TiDB. These use cases need instant analytics that are generated in real-time as opposed to batch analytics. TiDB is a perfect choice if you are considering how to replicate, migrate, or scale your database for extra capacity, looking for ways to optimize your existing storage capacity, getting concerned about slow query

performance, researching middleware scaling solutions or implementing manual sharding policy, looking for ways to perform real-time analytics, and more.

Most importantly, this white paper will show you how leading companies are using TiDB to solve their most pressing issues. If you have similar issues, TiDB might be a good choice for you.

Introduction to TiDB HTAP

TiDB ("Ti" stands for Titanium) is an open-source, distributed, NewSQL database that supports HTAP workloads. TiDB is fully compatible with the MySQL 5.7 protocol and has the common features and syntax of MySQL. TiDB's highlights include:

- **Data distribution:** TiDB is a distributed database designed for the cloud, providing flexible scalability, reliability, and security on the cloud platform. Users can elastically scale TiDB to meet their changing workloads. In TiDB, each piece of data has at least three replicas, which can be scheduled in different cloud availability zones to tolerate a data center wide outage. TiDB Operator helps manage TiDB on Kubernetes and automates tasks related to operating the TiDB cluster. This makes TiDB easier to deploy on any cloud that provides managed Kubernetes. TiDB Cloud (currently in beta), the fully-managed TiDB service, makes deploying, managing, and maintaining TiDB clusters even simpler with a fully-managed cloud instance that you control through an intuitive dashboard.
- **Scalability**: The TiDB architecture separates the computing and storage layers so you can scale them in or out independently as needed. Scaling is transparent to application operations and maintenance staff.
- **Financial-grade high availability:** The data is stored in multiple replicas. Data replicas obtain the transaction log using the Multi-Raft protocol. A transaction can be committed only when data has been successfully written to the majority of replicas. This guarantees strong consistency and availability when a minority of replicas go down. To meet the requirements of different disaster tolerance levels, you can configure the geographic location and number of replicas as needed.
- **Compatibility with the MySQL 5.7 protocol and MySQL ecosystem**: TiDB is compatible with the MySQL 5.7 protocol, common features of MySQL, and the MySQL ecosystem. To migrate your applications to TiDB, usually requires minimal (or no)

updates to your code.

- **ACID Compliance**: TiDB is fully ACID compliant and provides optimal transaction control by ensuring data integrity. This prevents users from seeing wrong or stale data.

TiDB architecture

The TiDB architecture comprises a set of TiDB servers, Placement Driver (PD) servers, TiKV servers, and TiFlash servers. These are explained in detail below.

TiDB server is a stateless SQL layer that exposes the connection endpoint of the MySQL protocol to the outside. TiDB server receives SQL requests, performs SQL parsing and optimization, and generates a distributed execution plan. It is horizontally scalable and provides a unified interface to the outside. It does not store data and is only for computation, SQL analysis, and transmitting data read requests to TiKV (or TiFlash) nodes.

PD server manages the cluster's metadata. It stores the metadata of real-time data distribution of every TiKV node and the topology structure of the TiDB cluster. It also provides the TiDB Dashboard management UI and allocates transaction IDs to distributed transactions. PD server is "the brain" of the entire TiDB cluster because it not only stores the cluster metadata, but also sends data scheduling commands in real time to specific TiKV nodes. This is according to the data distribution state reported by TiKV nodes in real time.

TiKV server stores data. TiKV is a distributed transactional key-value storage engine. A Region is the basic unit to store data. Each Region stores the data for a particular key range. Multiple Regions exist in each TiKV node. TiKV APIs provide native support to distributed transactions at the key-value pair level and support the snapshot isolation level by default. This is the core of how TiDB supports distributed transactions at the SQL level. After processing SQL statements, TiDB server converts the SQL execution plan to a TiKV API call. Therefore, data is stored in TiKV. TiKV data is automatically maintained in multiple replicas (three replicas by default), so TiKV has native high availability and supports automatic failover.

TiFlash server is what makes TiDB an HTAP database. It is a columnar storage extension of TiKV that provides a good isolation level and guarantees strong consistency.

How TiDB implements HTAP

The implementation of HTAP on TiDB is possible because of an intelligent architectural design (Figure 1) where TiDB creates an additional synchronous replica and stores it on TiFlash, a columnar-oriented storage.

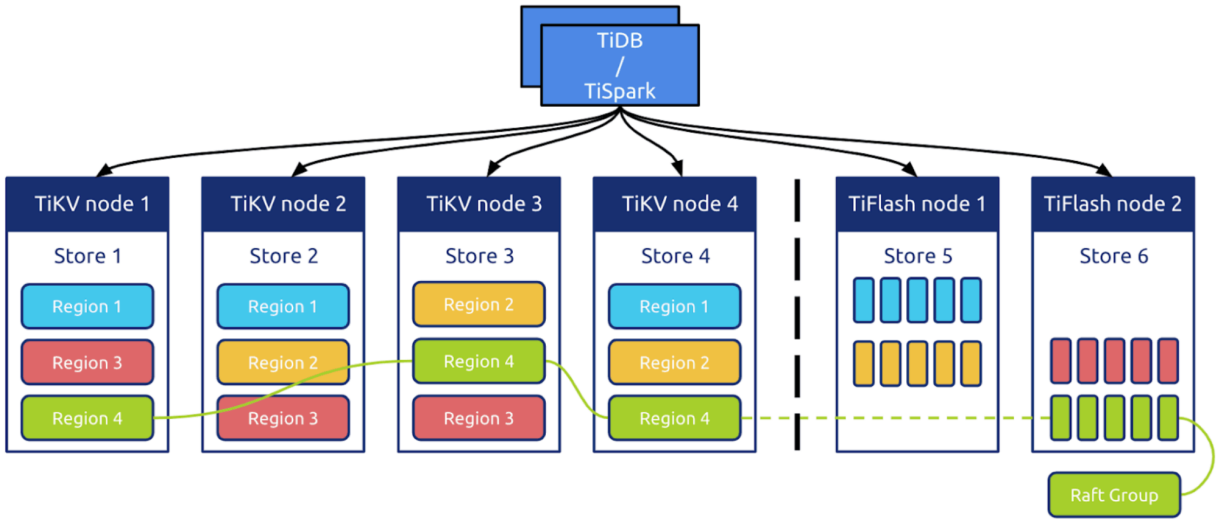


Figure 1: TiDB architecture with TiKV and TiFlash

Similar to TiKV, TiFlash also has a Multi-Raft system, which replicates and distributes data in the Regions. TiFlash replicates data in real-time in the TiKV nodes at a low cost that does not block writes in TiKV. It also provides the same read consistency as in TiKV and ensures that the latest data is read. The TiFlash Region replica is logically identical to the one in TiKV, and it is split and merged with the Leader replica in TiKV at the same time. TiFlash's highlights include the following.

Asynchronous replication: The TiFlash replica is asynchronously replicated as a special role, Raft Learner. This means when the TiFlash node is down or high network latency occurs,

applications in TiKV can still proceed normally. This replication mechanism inherits two advantages of TiKV: automatic load balancing and high availability.

Consistency: TiFlash provides the same snapshot isolation level of consistency as TiKV. This means you can read the data previously written in TiKV, ensuring that the latest data is read. Such consistency is achieved by validating the data replication progress. Every time TiFlash receives a read request, the Region replica sends a progress validation request (a lightweight RPC request) to the Leader replica. TiFlash performs the read operation only after the current replication progress includes the data covered by the read request's timestamp.

Optimizer choice: The column and row stores are one organic whole. But can the two stores coordinate? The trick is in the optimizer.

When the optimizer selects a query execution plan, it treats the column store as a special index. Among all the indexes in the row store and the special column store index, the optimizer uses statistics and cost-based optimization (CBO) to select the fastest index. Both the column and row stores are taken into consideration. You don't have to decide which storage engine to use in a complex query. The optimizer makes the best decision for you.

However, if you intend to completely isolate the column store and row store, you can manually specify that the query uses one of the two storage engines.

Computing acceleration: The columnar storage engine is more efficient in performing read operation. TiFlash pushes down some of the computation in the same way as the TiKV Coprocessor does. This reduces the data traffic to TiDB server and helps improve performance.

Real-time update: In TiDB, the replication between TiKV and TiFlash is from peer to peer. There's no in-between layer, so the data is replicated in real time. This reduces latency.

Scalable: The HTAP architecture balances replication and storage scalability. It uses the same replication and sharding mechanism as the previous Online Transactional Processing (OLTP) architecture. Therefore, the scheduling policy still applies to the HTAP architecture, and the cluster can still horizontally scale out or scale in. What's more, you can scale the column store and row store separately to meet the needs of your application.

Massively Parallel Processing (MPP)

Let’s discuss how TiDB HTAP achieves high performance for analytical queries.
The TiDB HTAP database uses the Massively Parallel (MPP) Processing engine shown in the following figure. MPP uses the shared-nothing architecture designed to handle multiple operations simultaneously by several processing units. Each processing unit works independently with its own operating system and dedicated memory

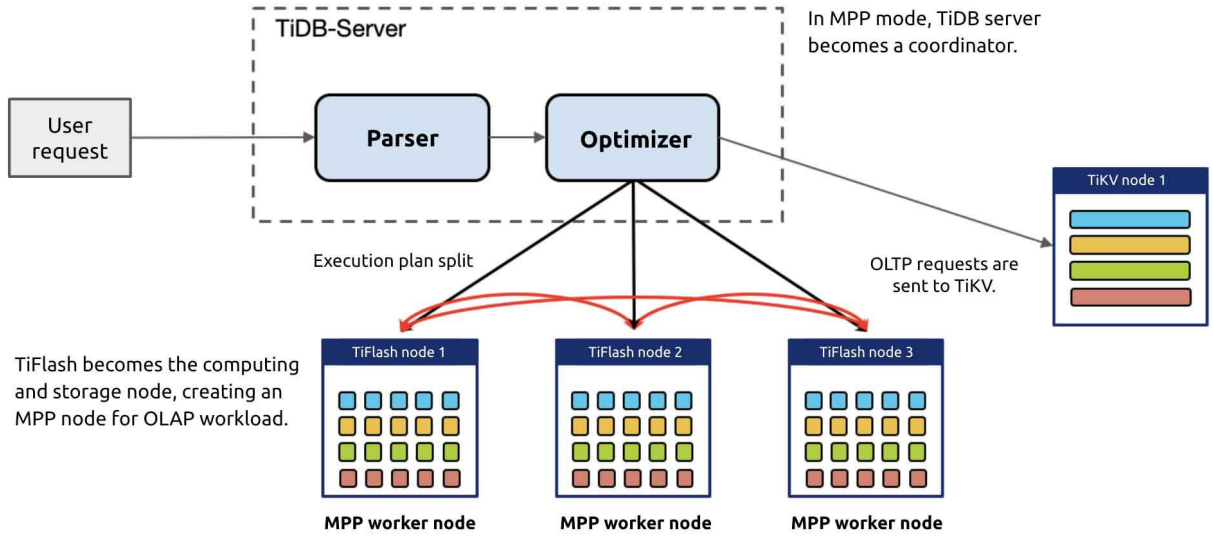


Figure 2: MPP architecture for TiFlash

In TiDB's MPP mode, TiFlash complements TiDB's computing capabilities. When dealing with Online Analytical Processing (OLAP) workloads, TiDB steps back to be a master node. The user sends a request to TiDB server, and all TiDB servers perform table joins and submit the results to the optimizer for decision making. The optimizer assesses all the possible execution plans (row-based, column-based, indexes, single-server engine, and MPP engine) and chooses the optimal one.

The following diagram shows how the analytical engine breaks down and processes the execution plan in TiDB's MPP mode. Each dotted box represents the physical border of a node.

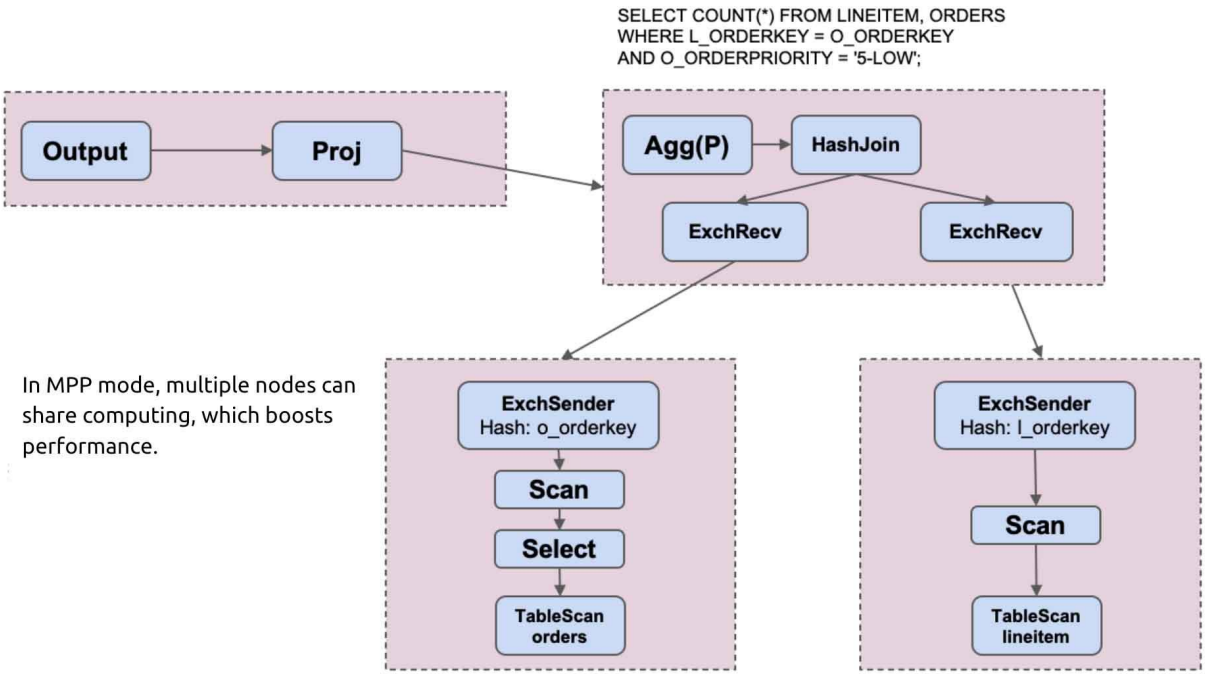


Figure 3: Execution plan

In the upper right corner, a query needs to join two tables and sort the results by a given order. This query is divided into an execution plan in which two nodes carry out the scan operations. One node scans the left table, and the other scans the right table. The two nodes do their join work independently according to the join condition, and the relevant data for the join are allocated together in each node. Data that belong to the same shard go to one node (or one group of nodes), and each node performs local computation. Finally, the results from each node are merged and returned to the client. This is the benefit of the MPP mode: large-scale queries like JOIN can be executed in parallel by multiple nodes.

HTAP performance

Performance expectations change based on the type of database used. For OLTP databases, the expectation is to run a high number of queries per second (QPS) with very low latency response times measured in milliseconds. For an OLAP database, the expectation is to run

highly complex queries with concurrency of 10-30 QPS and latency measured in seconds or minutes.

TiDB, being an HTAP database, merges these two workloads and provides high performance by querying data on TiKV and TiFlash in parallel. If a query demands point-select or minimal aggregation on the underlying data, the optimizer fetches the data from TiKV, the key value store, and offers milliseconds latency. It also processes more QPS.
However, if the query executed includes operations such as complex aggregations, multiple joins, historical data access, the optimizer fetches the data from TiFlash, the columnar data store. The optimizer may also combine results from both TiKV and TiFlash storage by fetching relevant data and boosting performance.

For us to prove a high performance TiDB HTAP database, we looked at a set of interesting data from the US Department of Transportation, including aircraft takeoffs and landings and on-time conditions from 1987 to the present. The data set contains more than 180 million lines of aircraft takeoff and landing records. To show comparative metrics, we ran the same workload on TiDB, MySQL 5.7, MariaDB columnar store 1.2.5, Spark 2.4.5 + Parquet, Oracle 12.2, and Greenplum 6.1.

Because some test objects do not support cluster mode, the test environment is a single machine. (However, with the help of TiDB's scalable system, TiFlash can also be expanded linearly.) The test machine had the following configuration:

- CPU: 40 vCores, Intel® Xeon® CPU E5-2630 v4 @2.20GHz
- Memory: 188 GB @ 2133 MHz
- Disk drive: One 3.6 TB NVMe SSD
- OS: centos-release-7-6.1810.2.el7.centos.x86_64
- Filesystem: ext4
- TiKV Region size: 512 MB
- Greenplum 6.1 segments (distributed randomly)
- Oracle 12.2

The following table compares the results:

Check for phrases	TiDB + TiFlash	MySQL 5.7.29	Greenplum 6.1	Mariadb Columnstore 1.2.5	Spark 2.4.5 + Parquet	Oracle 12.2.0.1
Q1	0.508	290.340	4.206	1.209	2.044	88.53
Q2	0.295	262.650	3.795	0.740	0.564	76.05
Q3	0.395	247.260	2.339	0.583	0.684	74.76
Q4	0.512	254.960	2.923	0.625	1.306	74.75
Q5	0.184	242.530	2.077	0.258	0.627	67.44
Q6	0.273	288.290	4.471	0.462	1.084	134.08
Q7	0.659	514.700	9.698	1.213	1.536	147.06
Q8	0.453	487.890	3.927	1.629	1.099	165.35
Q9	0.277	261.820	3.160	0.951	0.681	76.5
Q10	2.615	407.360	8.344	2.020	18.219	127.29

Figure 4: Performance comparison

The results show that TiDB + TiFlash performs more than 100 times better than MySQL. When you compare these results with MPP databases or the MariaDB ColumnStore and other analytical databases, TiDB delivers considerable performance gains for above mentioned scenarios. In addition, unlike TiDB, these databases are hard to update in real-time.

We also tested how long different queries took. We used a three-node, TPC-H 100 GB environment, with a table of over 100 million records. TiDB 5.0, Greenplum 6.15.0, and Apache Spark 3.1.1 were tested with the same hardware resources.
As shown in the chart below, TiDB was faster—on average, two to three times faster than Greenplum and Apache Spark. For some queries, TiDB is eight times faster.

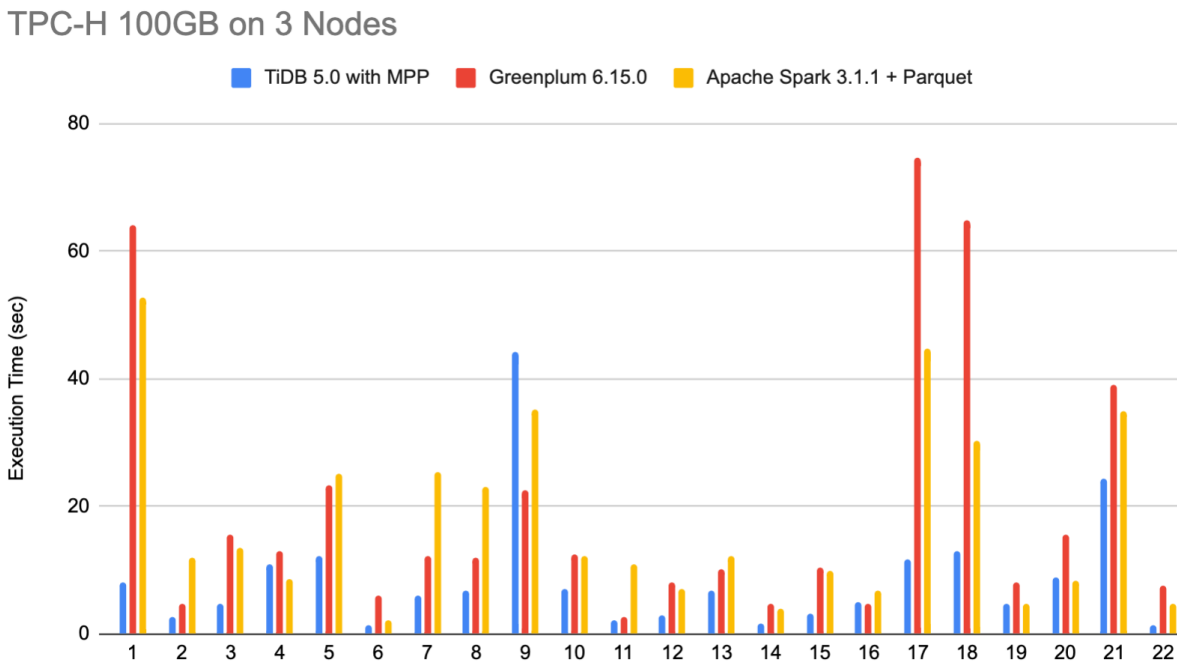


Figure 5: TPC-H performance comparison

Some popular TiDB scenarios

TiDB 5.0 is a complete HTAP database platform supporting both OLTP and OLAP workloads. Whether it's a read or write request, whether it's an OLTP or OLAP workload, TiDB processes it in the most efficient way possible and returns the result. Let’s explore some of the most popular scenarios.

Hybrid workloads

When faced with hybrid workloads, TiDB can efficiently handle both OLTP and OLAP requests (Figure 6). From a user perspective, all your data goes to one place. The app server receives all types of requests and sends them to the TiDB server, which dispatches the requests to different storage engines.

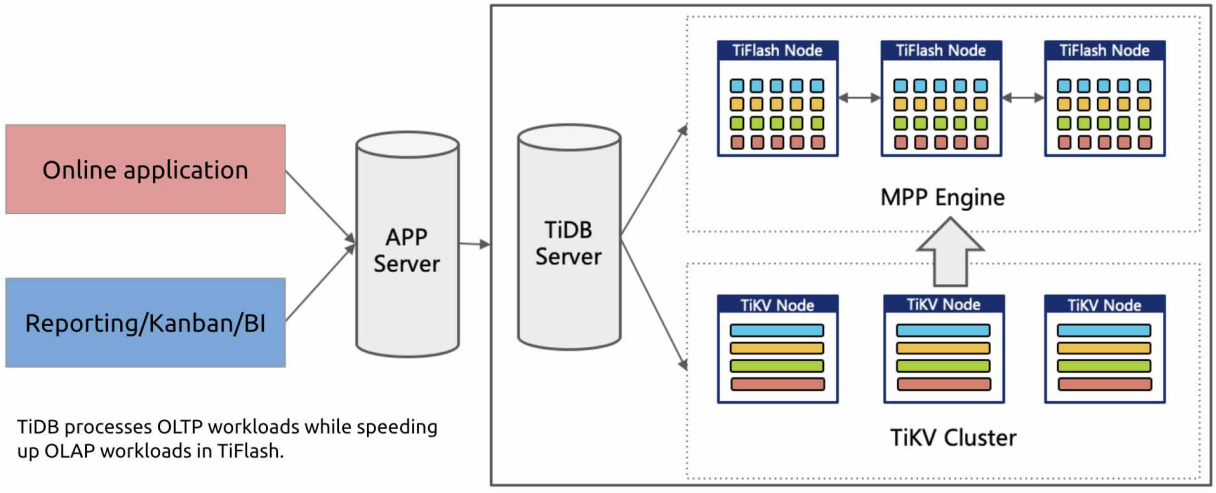


Figure 6: Simplicity of HTAP processing

Stream computing

Stream computing is also a pressing demand. Big data tools provide mature solutions for real-time log stream analytics. But if you need to delete or update data or need a table join, TiDB is an ideal choice.

First, TiDB is a true HTAP distributed database. You can connect it to Oracle as a replication destination or to MySQL as a replica using Kafka or other data pipelines. Also, if the application needs to process data, you can switch it back to the traditional database architecture.

Second, TiDB is also an OLTP database that responds to creates, reads, updates, and deletes (CRUD) in real time. With its hybrid storage engines, TiDB can process both point queries and aggregation queries.

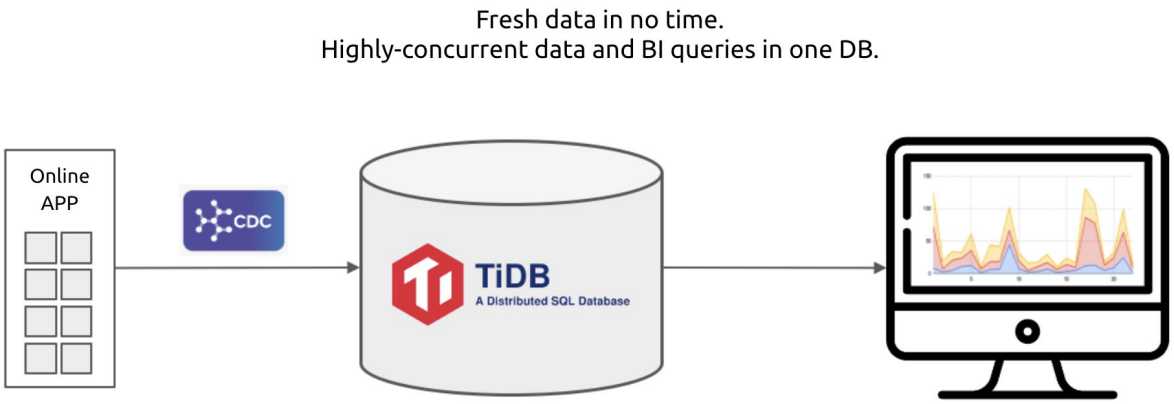


Figure 7: Stream computing using CDC

Data hub

If your company has more than one data source in the foreground—financing, ERP, sales, warehouses, click stream, user profile—each of them may be storing their data in their own databases. To achieve a real-time, single source of truth for hot data, you can integrate the data into TiDB via Change Data Capture (CDC) or Kafka, building a data hub layer (Figure 8).

A data hub is a layer between the application and the data warehouse. It stores data for only a limited time, while a data warehouse stores all historical data. A data hub tends to store hot data for real-time queries or processes highly-concurrent requests, while offline data warehouses and data lakes often provide older data for reporting and business intelligence (BI) queries.

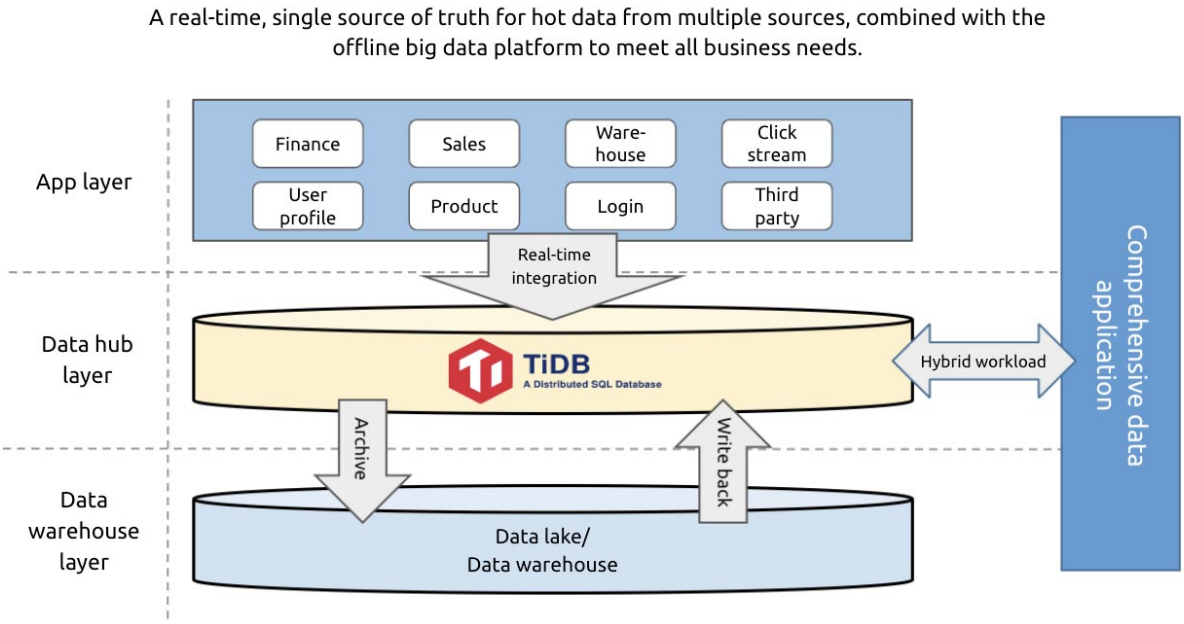


Figure 8: Data hub architecture

After you integrate TiDB into your data platform, it acts as a central hub for all your data. Despite an existing offline data warehouse or a Hadoop platform, you can position TiDB as an application layer to store and manage real-time data. As your business becomes more complicated and you enforce stricter data security standards, TiDB will also become a unified data central hub for data relations and lifecycle management to support your long-term growth.

Real-time analytics—some real world examples

Let's look at some of the most popular uses of real time analytics and see how TiDB can help.

Real-time fraud detection

Financial services companies closely monitor their transactions in real-time. They can't wait for a batch load. They need to make decisions instantly. For instance, if a debit card of a particular bank is swiped multiple times within a given time, a trigger is raised and a notification is sent to the customer, informing them about the activity. If the customer didn't make these transactions, they can inform the bank.

Healthcare analytics

The best fitness trackers help develop healthier habits by encouraging the user to take a few extra steps. These gadgets track everyday activity, sleep, heart rate, respiration and workouts. Companies producing these trackers have loads of data to analyze. One common use case is to monitor heart rate in real-time and alert users in case of unusual readings.

Move analytics to the edge

Companies use remote monitors and real-time data analytics to monitor trucks, planes, construction and manufacturing equipment, and other machines so they can spot maintenance issues prior to breakdowns. Predicting issues with real-time data analytics not only saves time and money, but also prevents catastrophic accidents.

Marketing campaigns

Real-time analytics empowers marketing teams to adjust their campaigns based on real-time data on clicks and conversions. This adjustment can lead to the appropriate target audience and attract more customers effectively.

How top companies use TiDB

Let's see how some of our customers use the TiDB HTAP solution to solve some of their most pressing issues.

Zalopay: a leading Internet technology company in Vietnam

Zalopay was the only startup company in Vietnam with a valuation of 1 billion USD. Their mobile payment application is the most widely used communication app in Vietnam, with over 100 million active users. The application allowed large number of users to transfer and collect payments with an online chat platform,
pay, recharge, book trips, order products, send red envelopes in family groups, and pay through the merchant's official account. The types of merchants cover multiple industries such as retail, catering, service, and e-commerce. Their business decision makers wanted to get insights into their various issues in real-time, understand trends (such as risk management and fraud detection), and to expand use of TiDB analytical services to conduct large scale data

mining for quick business recommendations. They found that TiFlash and integrated TiSpark was their optimal solution.

ZTO: China's largest logistics company

ZTO was processing more than 500,000 QPS. They decided to move out of Exadata to TiDB, which helped them to store three times more data with less cost, meet their workload requirements with an HTAP solution and improve performance by five times. After moving to TiDB, their live tracking "parcel" module was streamed to TiDB using Spark, and their mobile applications and real-time reporting could fetch data from TiDB in real-time. The mobile applications could show tracking status in near real time.

Shopee: Southeast Asia's leading e-commerce platform

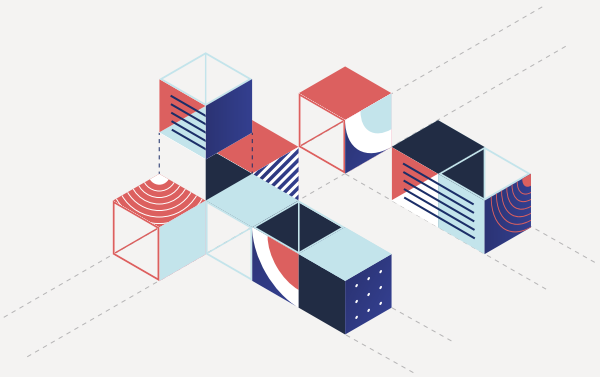
Shopee is the leading e-commerce platform in Southeast Asia and Taiwan. It provides customers with an easy, secure, and fast online shopping experience through strong payment and logistical support. As the business boomed, the team faced severe challenges in scaling their backend systems to meet the demand. Shopee decided to move their workloads to TiDB to provide better service and experience for users without worrying about the database capacity. The deployment includes over 60 nodes. One of their primary pain points was for their systems to detect abnormal behaviours and fraudulent transactions from the internal order logs and user behaviors.

Before TiDB, they started facing limitations when a major shopping season approached. In one of the promotional events, they received more than 11 million orders, 4.5 times more than the previous year. This followed by another massive surge in traffic where they received an all-time high of 12 million orders.

Now that Shopee has adopted TiDB, they can scale their databases on demand and maintain strong consistency while still maintaining strong compatibility with the MySQL protocol. Although their data size has grown eightfold over the past six months, the whole cluster still maintains a stable query response time. Generally, the 99th percentile response time is less than 60 ms.

Conclusion

TiDB, a distributed HTAP database is the industry-leading data platform to process heavy workloads without compromising on consistency, availability, and scalability. Its architecture replicates data between TiKV and TiFlash, thus providing real-time data consistency. Depending on the nature of the query, TiDB optimizer decides the best plan possible and fetches data from TiKV or TiFlash, hence ensuring the lowest response times. This is a true HTAP architecture and is already solving customer challenges pertaining to both transactional and analytical use cases.



Resource	URL
Try out TiDB	pingcap.com/download
Free trial with TiDB Cloud	tidbcloud.com/signup
Case Studies	pingcap.com/case-studies
Documentation	docs.pingcap.com/

To learn more and get started with your realtime HTAP journey with TiDB, visit us at https://pingcap.com/ or drop us a line at info@pingcap.com.