startree

GETTING STARTED WITH APACHE PINOT

Modern OLAP for Real-Time and User-Facing Analytics

Table of Contents

Introduction to Real-Time Analytics

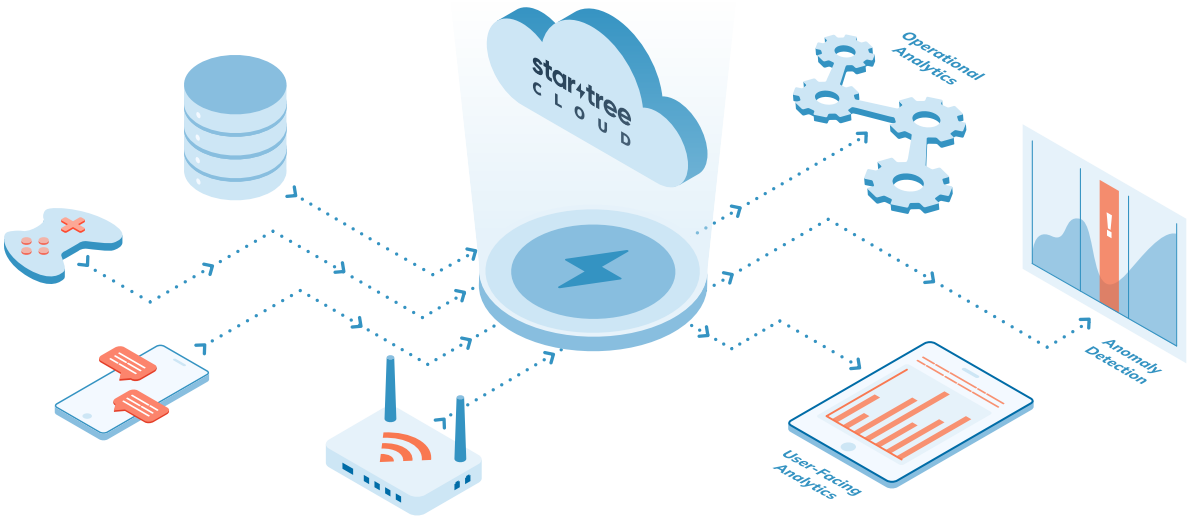
When one hears “decision maker,” it’s natural to think, “C-suite,” or “executive.” But thesedays, we’re all decision-makers. Yes, the CEO is a crucial decider… but so too is the franchise owner, the online shopper, the student.

Yet it has been traditional decision-makers who have had access to relevant data to guide their thinking. There simply has not been an easy way to scale real-time analytics tools for widespread use—let alone present them in an intuitive way that allows users to act on the information being shared. In other words, “Everyone is a Data Person but only a select few have access to the insights data” — this is about to change!

That situation is changing with the emergence of user-facing analytics: Giving users access to real-time data and analytics from within the product App or platform where they can take action. One of the best examples is LinkedIn’s “Who Viewed My Profile” application, which gives 700 million-plus LinkedIn users access to fresh insights.

But it’s not just access to data that users get. It is the ability to use that data to, for example, make a new connection just as they become aware of a profile. In other words, to really add value, it is not enough to provide reports or a dashboard to users. The data must be presented at just the right point in time to capture an opportunity for the user. Each LinkedIn member gets a personalized dashboard of all the profile views broken down by industry, location, job title, and week-over-week trend analysis. This enables the user to connect with the right people, leading to better opportunities, and accelerates LinkedIn’s mission of connecting the world’s professionals together.

This is a difficult challenge precisely because the value of data varies over time. If a handful of data analysts are looking at data in aggregate to pick out trends, it is a fairly straightforward task. But try to pick out those events that will be of value to thousands of users, at any time, and it becomes clear that we are charting new territory.

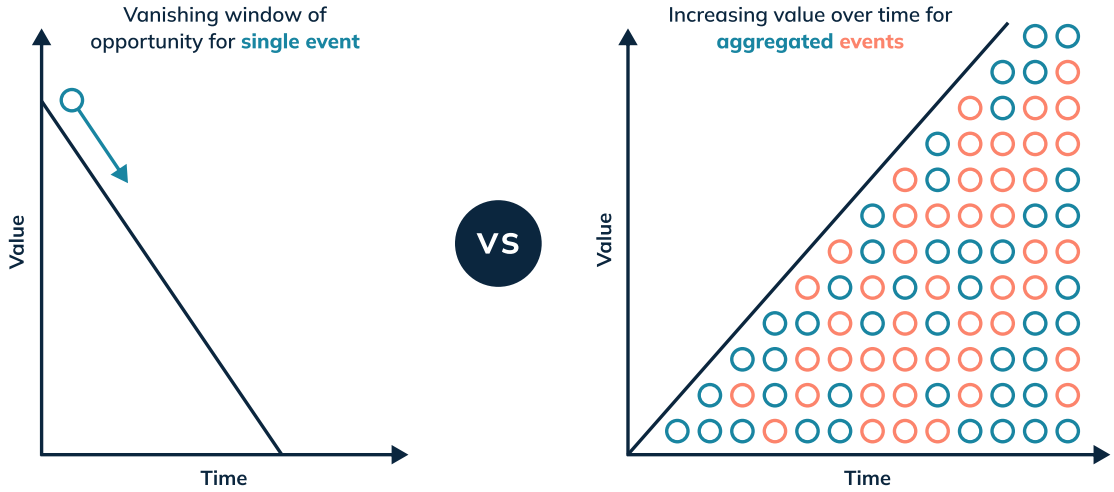


The Value of Data Over Time

Let's take a single event that gets captured digitally—for example, someone searches for the term "pizza" in a delivery app. What is the value of knowing everything about that single event?

The question makes sense only if we chart the value of that event over time. At the point in time when the search happens, the event has a high value: It is signaling that someone is actively searching to make a purchase of pizza. If the establishments in that area knew when that happened, they could offer those people a coupon. There is value because the user's intent is well-defined and immediate.

The value of this information drops off very quickly, however. It's no use knowing that someone wanted pizza yesterday. By the time you discover this fact today, chances are that the person already bought and enjoyed pizza from a competitor.



Now let's look not at a single event, but at events in aggregate. To extend our example, we are looking not at a single person searching for nearby pizza, but at all the people who searched for pizza within an area and over time. By looking at the aggregate data, we can learn new things. We can search for trends and do some forecasting.

Aggregate data shows an opposite pattern from single-event data: It might not be exceptionally valuable at the time of any single event, but its value increases dramatically over time as it allows for better and better forecasting.

There are times when you need both in order to make sense of even a single event—if you're doing fraud analytics, for example. And there is the challenge: Most systems are designed to handle one kind of data or the other, either single events or aggregate data. Many organizations end up using different systems, depending on specific use cases, which can get messy.

What would be ideal is to have both single events and aggregate data in one system, with both available quickly enough to be useful across a number of cases. This was one of the main goals of developing Pinot.

Who are Decision Makers?

Decision makers need information to make decisions. Traditionally, these have been people internal to organizations—operators and analysts. But people who are technically outside of an organization, but still interacting with it, might want data in order to make decisions as well.

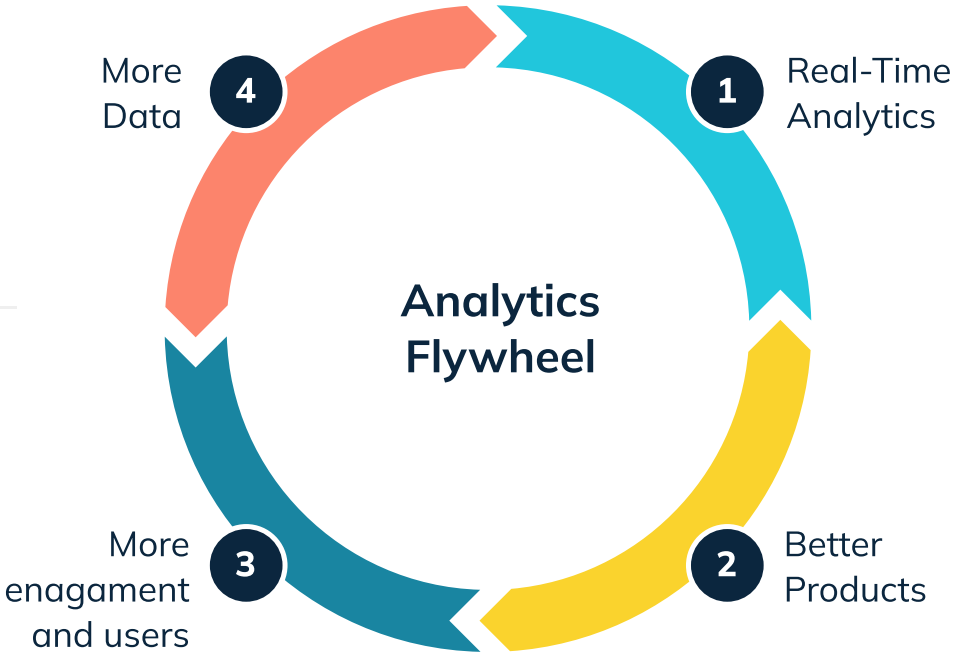
It should be no surprise that most analytics tools have been built with the first category (internal decision makers) in mind. Think about their profile: There are fewer of them, and they are not all using a system concurrently. Their tolerance for things like query latency is pretty high—if a query takes a few seconds, so be it: It is a part of their job.

The technical needs change, however, when we look at users and customers. (Sometimes users and customers are one and the same, and sometimes they are not; we provide some examples of each below.) Note how both of these categories differ from internal team members: There are many more of them, they are often connecting concurrently (to the tune of thousands or even millions of connections), and their tolerance for latency is much lower, with data analytics needing to be available in just milliseconds.

The requirements and scale of this second user base are very different, which is why there needs to be a special category of user-facing analytics. Companies that can master this new type of analytics will have a big point of differentiation.

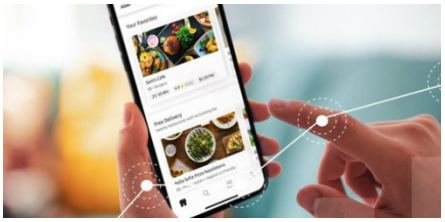
EXTERNAL
Application Users
Business Partners

INTERNAL
Business Analysts
Platform Operators



Food Delivery

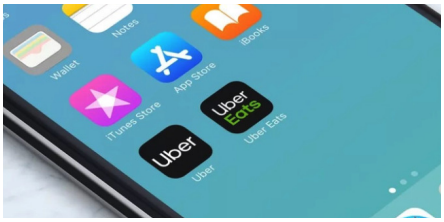
PLATFORM (INTERNAL)



Enabling Live Analysis with Real-Time Data

Long Requests - live orders that have been prepared by a restaurant, but have not yet been successfully assigned to a rider for at least 10 minutes.

EATERS



Improving End User Experience

Restaurants ordered from and dollar amount in users location in the last 5 minutes.

RESTAURANTS & RIDERS

Powering Service Performance Dashboards

Restaurants: Net payout, Daily and weekly items sold, Item ratings Riders: Orders delivered

Retail

PLATFORM (INTERNAL)

Enabling Live Analysis with Real-Time Data

Sales and user growth numbers categorized across different geographical regions, time of the year, retail channels, etc. Order fulfillment pipeline.

CONSUMER

Improving End User Experience

Popular items bought in the last hour. Item availability at stores near user's location.

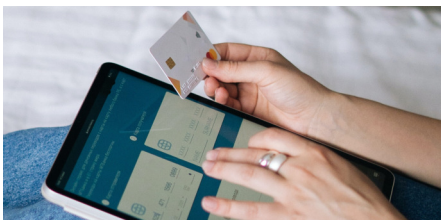
FRANCHISES/STORES

Powering Service Performance Dashboards

Inventory Management, Basket correlation, Cohort Analysis.

FinTech

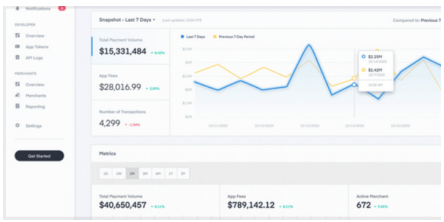
PLATFORM (INTERNAL)



Enabling Live Analysis with Real-Time Data

Tracking funds through ledger observability - condition of state machine for billions of transactions.

MERCHANTS (EXTERNAL)



Powering Service Performance Dashboards

Payment received. Total payment volume Number of transactions.

Key Requirements of Modern OLAP

Real-time analytics has transformed the way companies do business. It has unlocked the ability to make real-time decisions such as customer incentives, fraud detection, and operational intelligence. It also helps companies provide a personalized user experience which accelerates growth and improves retention. This is a complex problem and naturally, there are several OLAP (OnLine Analytics Processing) solutions out there, each focusing on a different aspect.

An ideal OLAP platform needs to be able to ingest data from a wide variety of sources. This data is then made available for querying across a wide spectrum of use cases. Each use case and corresponding ingestion method has unique requirements on the underlying platform which can often be competing in nature. For example, a real-time metrics use case requires higher data accuracy at the expense of performance. On the other hand, a user-facing analytical application will need to be optimized for speed.

In the chart below, you can see use cases and the associated query throughput, latency, consistency and accuracy, and complexity. As we can see from the query latency column, in the real-time analytics land, “fast” now means ultra-low latency in the order of milliseconds, even at extremely high throughput. As customer bases start shifting from analysts & data scientists to end-users who are interacting directly with data products, the ability to respond with ultra-low latency even at extremely high query throughput becomes very crucial for the underlying analytics system.

USE CASE	QUERY THROUGHPUT* (queries/second)	QUERY LATENCY* (p95th)	CONSISTENCY & ACCURACY	QUERY COMPLEXITY
User-facing Analytics	Very High	Very Low	Best Effort	Low
Personalization	Very High	Very Low	Best Effort	Low
Metrics	High	Very Low	Accurate	Low
Anomoly Detection	Moderate	Very Low	Best Effort	Low
Root-cause Analytics	High	Very Low	Best Effort	Low
Visualizations & Dashboarding	Moderate	Low	Best Effort	Medium
Ad-hoc Analytics	Low	High	Best Effort	High
Log Analytics & Text Search	Moderate	Moderate	Best Effort	Medium

Use Case Requirements

User Facing Analytics
This is a category of use cases where the analytical capabilities are exposed directly to the customer or end-user to provide a personalized experience. The key requirements are high throughput and low query latency since this directly impacts the user experience.

Personalization
This is a special type of user-facing analytics used to personalize the product experience for a given user. The key requirement here is to be able to extract real-time features based on the user's activity and product interaction which are then used to generate personalized recommendations or actions.

Metrics
Tracking business metrics in real-time is critical for operational intelligence, anomaly/fraud detection, financial planning, and more. This requires the OLAP platform to support high QPS, low latency queries. In addition, a lot of cases require a high degree of data accuracy. This means the OLAP platform must be able to handle duplicates or upserts in the data sources.

Anomaly Detection and Root Cause Analysis
It is important to be able to detect anomalies on large time-series datasets instantly in order to perform appropriate actions in time. It's also important to understand which dimensions were primarily responsible for causing the anomaly. Both of these use cases require the ability to perform high QPS temporal scans and 'group by' queries on the OLAP platform.

Visualization
Visualization can be as simple as a dashboard to plot metrics on different kinds of charts or as complex as geospatial visualization, clustering, trend analysis, and so on. Typically this requires the OLAP platform to integrate well with the existing visualization solutions such as Apache Superset and Grafana.

Ad Hoc Analytics
Oftentimes, users want to do real-time data exploration for debugging issues and detecting patterns as the events are being generated. This requires a way to support SQL (ANSI SQL compatible) queries on the underlying OLAP platform. Usually, the QPS is very low but the query complexity is high. This is a challenge since most of the common OLAP technologies don't support ANSI SQL out of the box.

Log Analytics / Text Search
The ability to perform text search queries on application log data in real-time is a less common but important use case. Logs can be enormous and often unstructured (eg: JSON format). The ability to do regex style text search on logs in real-time is vital for triaging production issues. Some applications will need the ability to do aggregation queries with text search predicates as part of their core business logic. The QPS is usually low for debugging use cases but can be high for user-facing applications.

Data Source Requirements

Common Requirements of Streaming, Batch/Offline, and OLTP System Sources:

Unstructured Data: The ability to handle unstructured data is very important for ease of adoption across a broad spectrum of users. Ideally, the OLAP platform should be able to handle such unstructured datasets without significant degradation in query performance.

High throughput: The OLAP platform must be able to keep up with the high rate of messages being produced across the different sources. This can easily range from a few MBs of data per day to TBs per day for larger use cases.

Data transformation support: The underlying OLAP platform must support basic transformations on the incoming records. This can include projection, filtering, column transforms (eg: string manipulation, numerical processing, UDFs), and so on.

Upserts: The ability to upsert data based on a primary key is crucial for certain use cases such as metrics. This can either be a full row upsert or partial (column subset) upsert. Without this capability, the resulting analytical queries will end up double-counting records or compute incorrect aggregations.

Unique Requirements for Each Data Source:

Streaming: In a streaming data source, data is produced and consumed in a streaming fashion, in an asynchronous manner. Common examples include Apache Kafka, Apache Pulsar, Amazon Kinesis, Google PubSub. This is extremely useful for capturing system and application logs as well as business events produced by different microservices. Streaming sources need low ingestion latency, which is the amount of time between data being produced and available for querying.

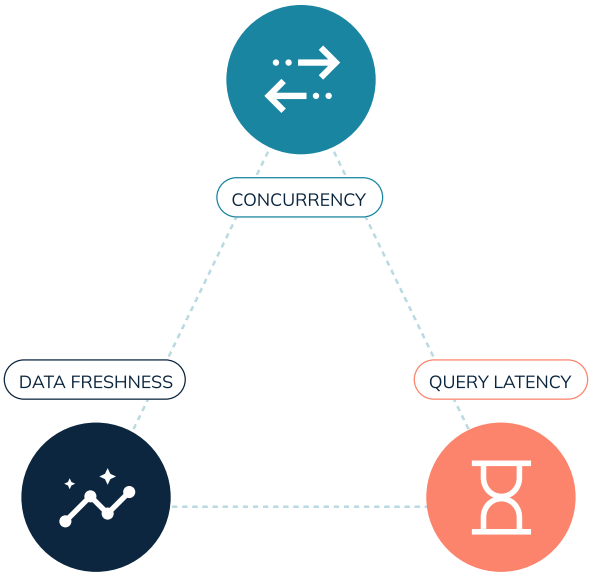
Batch (Offline): Batch sources consist of files or objects stored in a traditional storage system like Apache HDFS, Amazon S3, and Google GCS. This is typically used for bootstrapping a real-time analytics table with historical data, backfilling, and in scenarios where the raw data is not available in any real-time source. We must be able to ingest all the data in an efficient manner without compromising the SLA guarantees of the OLAP platform. Given that we will need to ingest the data multiple times (eg: daily jobs) it is preferable for the OLAP platform to support scheduling ingestion tasks in a periodic fashion.

OLTP: Oftentimes, we want to be able to analyze all the data being stored in a traditional OLTP (OnLine Transaction Processing) database like MySQL or Postgres. Data ingestion in this case is typically done in 2 modes. Bootstrap involves fetching historical data from data sources in a batch fashion. Real-time updates involve ingesting the changes happening to the source in real-time.

What is Apache Pinot?

Pinot is a real-time distributed OLAP datastore, purpose-built to provide ultra low-latency analytics, even at extremely high throughput. It can ingest directly from streaming data sources - such as Apache Kafka and Amazon Kinesis - and make the events available for querying instantly. It can also ingest from batch data sources such as Hadoop HDFS, Amazon S3, Azure ADLS, and Google Cloud Storage.

Pinot is a distributed system that is made of multiple components. Each component plays a unique role in a Pinot cluster which is distributed across many servers.



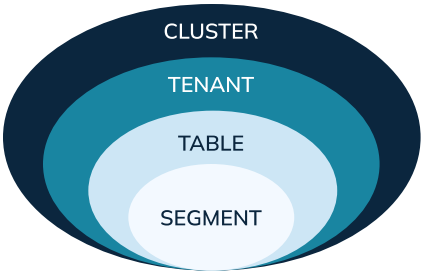
Understanding Apache Pinot Concepts

Table: A table is the logical component that represents a collection of related records. For example, we can store a collection of e-commerce orders in the orders table, sensors readings in the readings table. As the user, the table is the only logical component that you'll have to deal with directly.

Schema: Each table in Pinot is associated with a Schema. A schema defines what fields are present in the table along with the data types. As opposed to RDBMS schemas, multiple tables can be created in Pinot (real-time or batch) that inherit a single schema definition.

Segments: A table is expected to grow to an unlimited size. Pinot is designed to store a single table across multiple servers. To make that easier, we break a table into multiple chunks called segments. Each segment holds a subset of the records belonging to a table. A segment is the centerpiece in Pinot's architecture which controls data storage, replication, and scaling. Segments are physical structures that have a defined location and an address inside a Pinot cluster.

Tenants: A table is associated with a tenant. All tables belonging to a particular logical namespace are grouped under a single tenant name and isolated from other tenants. For example, marketing and engineering can be considered as two different tenants. Tables tagged with marketing will be isolated from engineering.



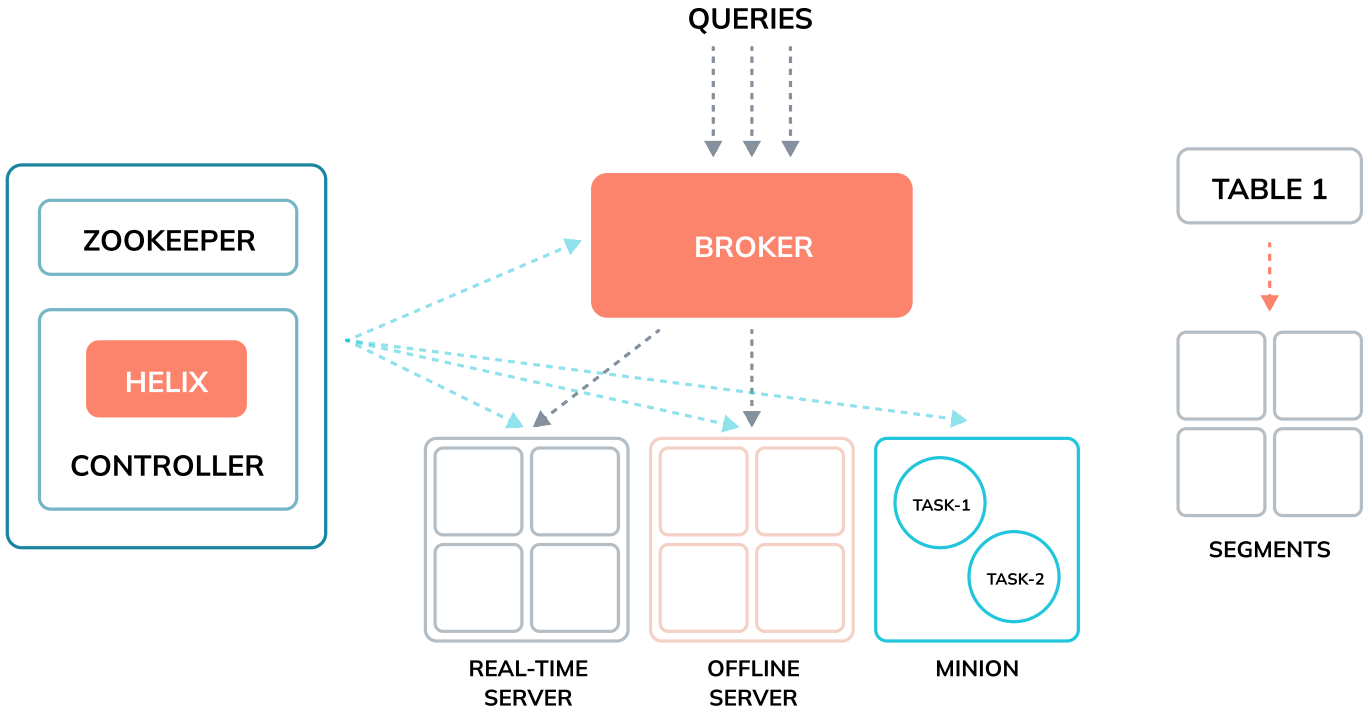
Components of a Pinot Cluster

Apache Zookeeper
Zookeeper is not a part of Pinot. But Pinot relies on Zookeeper for cluster management, metadata and configuration management, and coordination among different components.

Pinot Controller
You access a Pinot cluster through the Controller, which manages the cluster's overall state and health. The Controller provides RESTful APIs to perform administrative tasks such as defining schemas and tables. Also, it comes with a UI to query data in Pinot. The controller works closely with Zookeeper for cluster coordination.

Pinot Broker
Brokers are the components that handle Pinot queries. They accept queries from clients and forward them to the right servers. They collect results from the servers and consolidate them into a single response to send it back to the client.

Pinot Server
Servers host the data segments and serve queries off the data they host. There are two types of servers — offline and real-time. Offline servers typically host immutable segments. They ingest data from sources like HDFS and S3. Real-time servers ingest from streaming data sources like Kafka and Kinesis.



Apache Pinot Open Source

The Apache Pinot project started at LinkedIn in 2015, but in the past few years, we've seen many prominent companies across industries leverage Apache Pinot to provide analytics to their internal users and external customers.

At the start of 2021, there were 50,000 downloads. By 2022, the project reached over 1 million docker downloads! The slack community has grown to over 2100 members and the contributors have more than doubled over the course of a year.

COMPANIES	100+
CONTRIBUTORS	200+
SLACK MEMBERS	2,100+
DOCKER DOWNLOADS	1,000,000+
MEETUP GROUP	1,000+

Apache Pinot at Scale

Apache Pinot at LinkedIn - If you're using LinkedIn, you are accessing Pinot in some shape or form. Any number that you see on LinkedIn whether you go "Who Viewed My Profile?" or your LinkedIn feed, you are querying Pinot behind the scenes.

Apache Pinot at Uber - Many different systems in Uber are now powered by Apache Pinot. If you open an UberEats app and see orders near you, you are making a query to Pinot.

Apache Pinot at Stripe - Stripe only started using Pinot last year, but their scale has been enormous. Almost every transaction in Stripe is getting stored in Pinot, and they're able to achieve a sub-second latency at this scale.

LinkedIn	stripe	Uber
1M+ EVENTS/SEC	1PB+ DATA SIZE	200TB+ DATA SIZE
200K+ QUERIES/SEC	1T ROWS	30K+ QUERIES/SEC
1M+ QUERY LATENCY	< 1S QUERY LATENCY	<100MS QUERY LATENCY

Additional Resources

Wherever you are on your real-time analytics journey, we'd love to help you on your way.

Blogs & Videos

Ready to get your first cluster up and running? Check out this video or this blog to learn how to get started!

What makes Apache Pinot so fast?
The definition of real-time analytics has evolved drastically in the recent few years. Analytics is no longer just ad-hoc queries and dashboards, with lenient SLAs, for a handful of internal users. Companies now want to support complex real-time analytics use cases, such as user-facing analytics, personalization, anomaly detection, root cause analysis. This has dramatically changed the expectations of query latency, throughput, query accuracy, data freshness, and query flexibility from underlying analytics systems. Read on!

Apache Pinot & Real-Time Ad Analytics By UberEats
UberEats was one of the earlier adopters of Apache Pinot, and the company was able to leverage it across several interesting use cases. Here's one example of how UberEats relied on Apache Pinot (along with other open sources technologies) to deliver accurate, real-time ad analytics at scale, and reliably. Read on!

Connect with Us!

Follow us @startreedata

Watch us on YouTube

Connect with us on LinkedIn

Join our Meetup Group

Join our Community on Slack

Reach out to sales@startree.ai