
apache
Ignite
TM

Distributed Database For High-Performance Applications With In-Memory Speed

Scale Across Memory And Disk Without Compromise

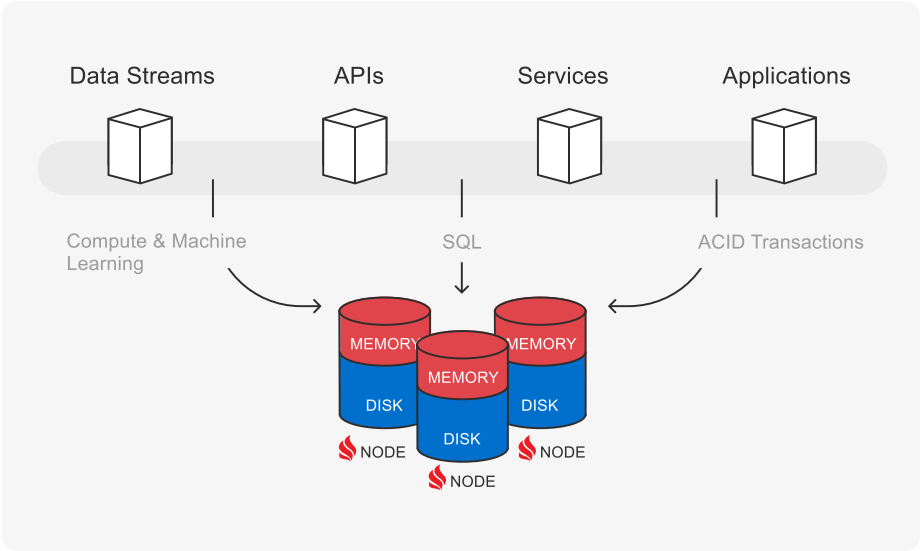
IN-MEMORY DATABASE OVERVIEW

What is an in-memory database?

An in-memory database (IMDB) is a data management system that stores data primarily in the computer's main memory.

How does an in-memory database work?

In-memory databases rely on spinning disks for data storage. IMDBs allow mission-critical applications to benefit from faster response times than disk-based databases.



Apache Ignite as a distributed in-memory database scales horizontally across memory and disk without compromise

Apache Ignite works with memory, disk, and Intel Optane as active storage tiers.

This multi-tier architecture combines the advantages of in-memory computing with disk durability and strong consistency, all in one system.

- **Speed of memory**
- **Strong consistency**
- **Durability of disk**

ADVANTAGES OF IGNITE MULTI-TIERED ARCHITECTURE

- **Instantaneous cluster restarts**

  Ignite becomes fully operational from disk upon a cluster startup or restarts without requiring a preload or a warm-up the memory tier.

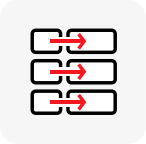
- **Multi-tiered storage**

  Ignite treats disk as an active storage layer, allowing it to cache a subset of the data in memory and query both in-memory and disk-only records with SQL and all other available APIs.

Apache Ignite as an in-memory database supports a variety of developer APIs

Essential Developer APIs

SQL



Key-value

ACID transactions

Enable you to request, join, and group distributed datasets.

High-Performance Computing APIs

Compute

Machine learning

Services

Execute logic close to the data, thus eliminating expensive data shuffling over the network.

Real-Time Streaming APIs

Streaming

Continuous Queries

Messaging

Allow the seamless implementation of event-driven architectures.

APACHE IGNITE FAQS

What Is In-Memory Computing?

In-memory computing is a software and data-processing technique that stores data sets in memory across a cluster of interconnected nodes. An average speed performance is 10-1000x faster than in disk-based systems.

In-memory computing software includes a distributed in-memory store with APIs and libraries optimized for high-performance data processing. Each cluster node (physical or virtual machine) contributes its available memory space with CPU cores to the total capacity of the cluster.

An application interacts with the cluster as a single unit, letting the in-memory computing software shield and manage all the internals related to inter-node communications, data distribution, and queries processing. The cluster scales linearly and horizontally to meet the data volume and throughput goals of the applications.

Apache Ignite Belongs To The In-Memory Computing Category:

Build real-time and event-driven solutions that process data with in-memory speed

Scale up and out across available memory and disk capacity

Take advantage of built-in SQL, high-performance computing and real-time processing APIs

Is Ignite A Distributed Cache?

Yes

When Ignite native persistence is disabled, Ignite can function as a distributed in-memory cache with support distributed ACID transactions, SQL queries, high-performance computing APIs, and more.

Is Ignite A Distributed Database?

Yes

Ignite is a distributed database for high-performance computing with in-memory speed.

Data in Ignite is stored in-memory and/or on-disk, and is either partitioned or replicated across a cluster of multiple nodes. This provides scalability, performance, and resiliency.

Is Ignite An In-Memory Database?

Yes

Ignite multi-tier storage supports both in-memory and disk tiers. You can always disable the native persistence and use Ignite as a distributed in-memory database, with support for SQL, transactions and other APIs.

Is Ignite An In-Memory Data Grid?

Yes

Ignite is a full-featured distributed data grid. As a grid, Ignite can automatically integrate with and accelerate any 3rd party databases, including any RDBMS or NoSQL stores.

Is Ignite An SQL Database?

Not fully

Although Ignite supports SQL natively, there are differences in how Ignite handles constraints and indexes.

Ignite supports primary and secondary indexes, however the uniqueness can only be enforced for the primary indexes. Ignite also does not support foreign key constraints at the moment.

Is Ignite A Disk- Or Memory-Only Storage?

Both

Native persistence in Ignite can be turned on and off. This allows Ignite to store data sets bigger than can fit in the available memory.

Essentially, smaller operational data sets can be stored in-memory only, and larger data sets that do not fit in memory can be stored on disk, using memory as a caching layer for better performance.

Is Ignite A NoSQL Database?

Not exactly

Just like other NoSQL databases, Ignite is highly available and horizontally scalable.

However, unlike other NoSQL databases, Ignite supports SQL and ACID transactions across multiple cluster nodes.

Is Ignite A Transactional Database?

Not fully

ACID Transactions are supported, but only at key-value API level. Ignite also supports cross-partition transactions, which means that transactions can span keys residing in different partitions on different servers.

At SQL level, Ignite supports atomic but not yet transactional consistency. A SQL transactions implementation is already in the works and will be released in Ignite 3.

Is Ignite A Multi-Model Database?

Yes

Ignite supports both key-value and SQL for modelling and accessing data.

In addition, Ignite provides strong processing APIs for computing on distributed data.

Is Ignite A Key-Value Store?

Yes

Ignite provides a feature-rich key-value API that is JCache (JSR-107) compliant and supports Java, C++, .NET, and other programming languages.