



# Hazelcast Jet

A stream processing engine accelerated with in-memory speed

[hazelcast.com/jet](https://hazelcast.com/jet)

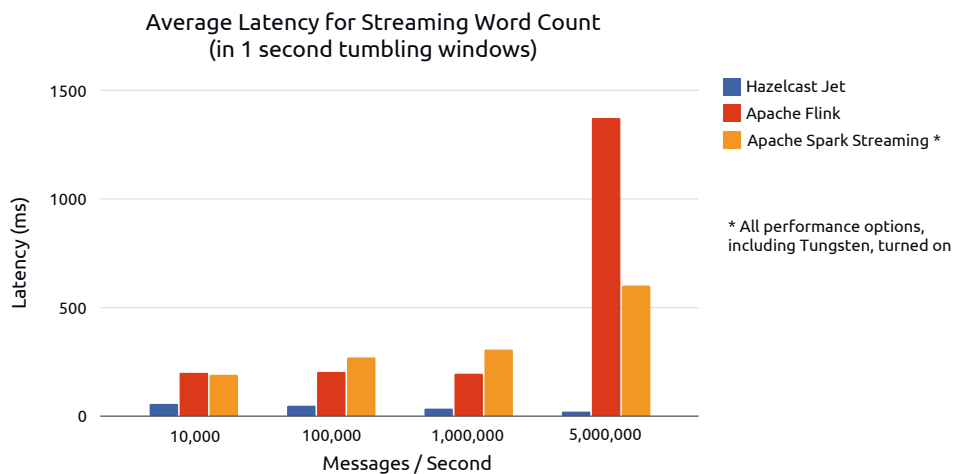
## Feature Summary

- **Programming languages:** Java (plus Python for user-defined functions).
- **Connectors:** Kafka, MongoDB, IMDG, and more; see official documentation for the up-to-date list; can use any Kafka Connect module.
- **JVM versions:** JDK 8, 9, 10, 11
- Supported JVM vendors: Oracle JDK, OpenJDK, IBM JDK, Azul Zing & Zulu.
- **Management Center:** Administration UI for per-node statistics, JMX APIs, REST APIs.
- **Streaming operations:** Time window aggregations (tumbling, sliding, session, custom windows defined by user code), unordered and late data handling.
- **Apache Beam support,** can be plugged in as a Beam runner.
- **Exactly-once processing,** including two-phase commit support.
- **Change data capture** for MySQL, Oracle, PostgreSQL, SQL Server, Cassandra, MongoDB.
- **Job Upgrade:** Replace running job with new version of code with no interruption.
- **Fault tolerance:** Distributed state snapshots for recovery, automatic re-execution of part of a job upon worker failure, Lossless Cluster Restart.
- **Security:** A Full suite of security capabilities to ensure your data is protected.
- **PaaS support:** OpenShift, VMware Tanzu (formerly Pivotal Cloud Foundry).

Hazelcast Jet® is a stream processing engine designed for high performance, easy maintenance, and resilience. Its architecture entails a distributed engine that leverages in-memory computing to run applications at near real-time speeds. Jet is used to develop both stream and batch processing applications using a directed acyclic graph (DAG) for job planning to optimally leverage resources when parallelizing subtasks.

## Why Hazelcast Jet

Jet is extremely fast (see chart below) and scales well. The cooperative multi-threading architecture leverages your entire cluster for optimal parallel processing with low context-switching overhead. “Lite” jobs run short-lived tasks with lower startup latency. Tight integration with Hazelcast IMDG provides ultra-fast, in-memory lookups and stream enrichment.



### Average latency in streaming engines, lower is better.

Jet is easy to use. It is delivered as a small JAR file, so its core can be embedded in lightweight, self-contained applications (ideal for edge computing). Jet also runs in client-server mode to support many client apps. Its declarative API simplifies the coding of large-scale jobs. And Jet includes many pre-built connectors to easily integrate with your data platforms.

Jet is very reliable. It uses in-memory snapshots to track state for recovery. It gracefully handles node failures (no data loss/duplication). Lossless Cluster Restart enables fast recovery after a total cluster failure or maintenance shutdown. Job Upgrade lets you easily replace a job with a newer version, without interruption.

## Advantages

- **Fast.** Jet shows higher throughput and lower latency versus other leading stream processing engines in head-to-head benchmarks.
- **Scalable.** Simply add more nodes to add more processing power to Jet.
- **Easy.** Jet can be deployed in embedded mode in apps or as a client-server installation, and offers an easy API for building sophisticated stream processing apps.
- **Reliable.** Jet runs 24/7 deployments with safeguards against data loss and data duplication, even upon node failure.
- **Secure.** Jet leverages the security suite in Hazelcast IMDG to protect your data from unauthorized access.

## Example Use Cases

- **Internet of Things and edge processing.** Operations optimization, fleet management, predictive maintenance, equipment/systems monitoring.
- **Transaction processing.** Retail, online payments.
- **Real-time machine learning inference.** Fraud detection, recommendations, classification/categorization.
- **Large-scale analytics.** Continuous/drill-down query, on-demand ETL.

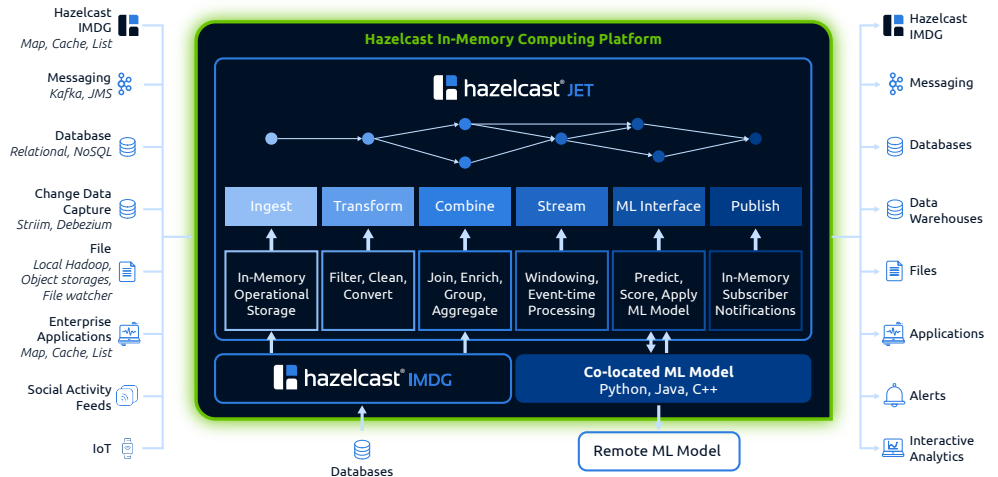


2 West 5th Ave., San Mateo CA  
94402 USA  
Email: sales@hazelcast.com  
Phone: +1 (650) 521-5453  
Visit us at www.hazelcast.com

All rights reserved.

## Architecture

Hazelcast Jet is built on top of Hazelcast IMDG to leverage the memory management, clustering, high availability, and security capabilities of IMDG. A basic Jet deployment consists of data sources, Jet connectors, the core Jet engine, IMDG, systems of record, and data sinks.



**Integrated in-memory store.** IMDG speeds up frequently run streaming tasks such as taking snapshots (for recovery), tracking state, and enriching streaming data.

**User defined functions.** Jet lets you easily plug in external Python or Java code as part of the processing pipeline. Jet starts up the Python VMs in the same nodes as the processing tasks to reduce network latency, and maintains the VMs to ensure high availability.

**Exactly-once processing.** Jet lets you correctly handle streaming transactions with exactly-once processing, even upon failures. It supports any replayable data source (e.g., Kafka, Hazelcast IMap journal) and any idempotent data sink (i.e., with upsert capabilities). It also supports two-phase commit logic to allow more sources (e.g., JMS) and sinks (e.g., Kafka, files) that are not replayable/idempotent.

**Apache Beam integration.** Jet supports Apache Beam, a portable API layer for building sophisticated parallel processing pipelines. Use Jet as an execution engine (“runner”) to get the highest throughput and lowest latency Beam pipelines.

### Hazelcast Jet Pro

- All capabilities in open source Hazelcast Jet
- Lossless Cluster Restart
- MC Packs: Notification, Security, Support, Integration
- PaaS support (OpenShift, VMware Tanzu)
- IMDG Pro features

### Hazelcast Jet Enterprise

- All capabilities in Hazelcast Jet Pro
- Job Upgrade
- Security
- MC Pack: Config
- IMDG Enterprise features (except WAN Replication)

To get started with Hazelcast Jet, visit <https://hazelcast.com/products/jet/#learn>.