Hazelcast

by Atena Jafari Parsa
Software Intern at İ2İ Systems

Time is money. 

Time is more valuable than money. ✓

Time Complexity ⏳ > Space Complexity 💽

“ If I’d asked my customers what they wanted, they would have said a faster horse.”

-Henry Ford,
Founder of the Ford Motor Company

Hazelcast stores most frequently used data in RAM, it is expensive but fast.

Hazelcast

A fast, smart system that helps multiple parts of a software communicate instantly whilst running on different machines or devices.

- ☕ Java-based in-memory data grid
- 🔑 Distributed, scalable, and super-fast key-value pairs store.
- 🚀 Work faster without asking the main database every time
- 💥 Handle millions of users using the system simultaneously

Why is Hazelcast fast?

1️⃣ It Stores Data in RAM (Memory)

Hazelcast keeps data in RAM, not on a hard drive — which means access is 100x faster than traditional databases.

2️⃣ It Shares Data Across Machines

Hazelcast runs on multiple servers at once (a cluster).
All machines work together and share the load — no waiting in line.

3️⃣ It Doesn’t Need to Ask the Database Every Time

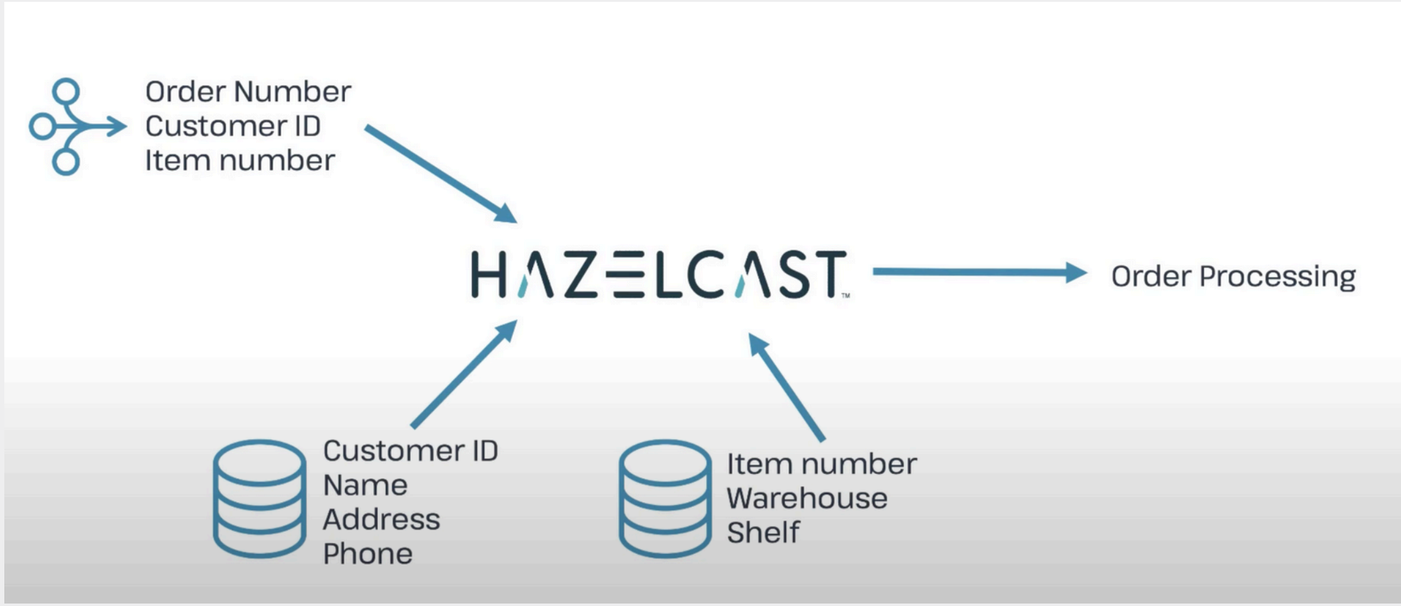
Hazelcast can remember frequently used data.

4️⃣ It’s Always Running in the Background

Hazelcast is always ready, storing and reacting to data instantly — even while things are moving (streaming, real-time events).

Hazelcast connects:

- **Mobile/Web/Desktop/SMS apps indirectly to AOM.**
- **AOM (Account & Order Management) to TGF (Traffic Generator).**
- **Fast, synchronized state sharing and distributed cache (e.g., session state, live user balance).**



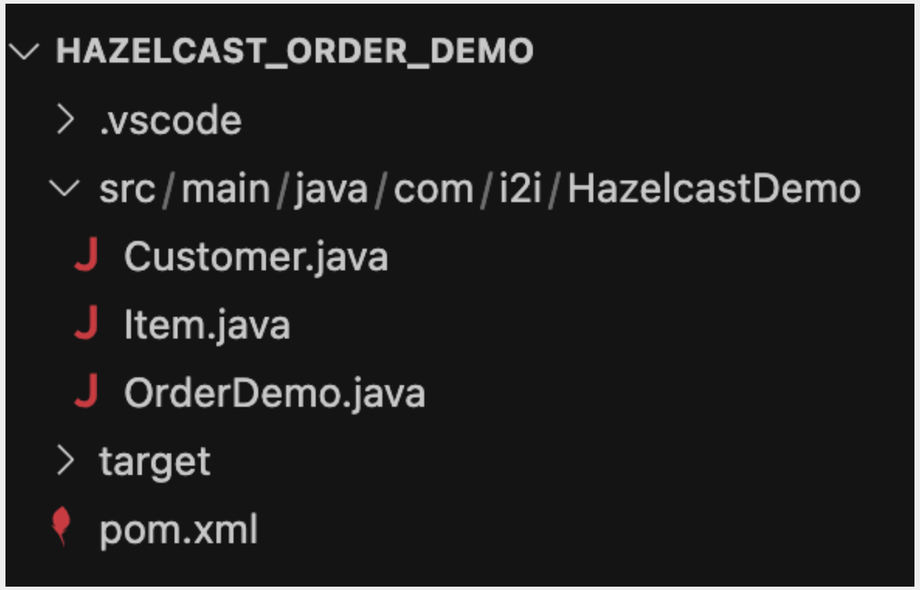
Demo Sales Project with Hazelcast

- Customer Class and Item Class
- Order class with simulated live order

Requirements:

On Visual Studio Code, add these extensions:

- Java
- Maven for Java (You can also use Gradle)
- Language Support for Java(TM) by Red hat
- Extension Pack for Java



Demo Sales Project with Hazelcast Cont.

- Customer Class with name, address and phone number
- Item Class with Warehouse letter and shelf number

```java
public class Customer {
    public String name;
    public String address;
    public String phone;

    public Customer(String name, String address, String phone) {
        this.name = name;
        this.address = address;
        this.phone = phone;
    }
}
```

```java
// Item.java
public class Item {
    public String warehouse;
    public String shelf;

    public Item(String warehouse, String shelf) {
        this.warehouse = warehouse;
        this.shelf = shelf;
    }
}
```

Demo Sales Project with Hazelcast cont.

- Order class, hazelcast instance and creates distributed IMaps for customers and items
- Adds sample data:

```
public class OrderDemo {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        HazelcastInstance hz = Hazelcast.newHazelcastInstance();

        // Create distributed maps
        IMap<Integer, Customer> customerMap = hz.getMap(name:"customers");
        IMap<Integer, Item> itemMap = hz.getMap(name:"items");

        // Add sample data
        customerMap.put(key:1, new Customer(name:"Atena", address:"Istanbul", phone:"555-123"));
        itemMap.put(key:101, new Item(warehouse:"Warehouse-A", shelf:"Shelf-3"));
```

Demo Sales Project with Hazelcast Cont.

- Incoming data stream is simulated
- Then, customer and item data is looked up from the created maps
- Finally, the simulated “live” order processing is printed neatly

```
    // Simulate incoming order
    int orderNumber = 5001;
    int customerId = 1;
    int itemNumber = 101;

    // Look up data from maps
    Customer customer = customerMap.get(customerId);
    Item item = itemMap.get(itemNumber);

    // Print final enriched order (simulate Order Processing)
    System.out.println("Order #" + orderNumber);
    System.out.println("Customer: " + customer.name + ", " + customer.address + ", " + customer.phone);
    System.out.println("Item: " + item.warehouse + ", " + item.shelf);

    hz.shutdown();
}
```

Demo Sales Project with Hazelcast Cont.

- mvn clean compile exec:java (to run)
- Hazelcast started up, formed a local cluster, and shut down cleanly after the job.

```
Jul 01, 2025 1:00:59 AM com.hazelcast.core.LifecycleService
INFO: [192.168.1.154]:5701 [dev] [5.3.6] [192.168.1.154]:5701 is STARTED
Jul 01, 2025 1:00:59 AM com.hazelcast.internal.partition.impl.PartitionStateManager
INFO: [192.168.1.154]:5701 [dev] [5.3.6] Initializing cluster partition table arrangement...
Order #5001
Customer: Atena, Istanbul, 555-123
Item: Warehouse-A, Shelf-3
```

Resources

- https://youtu.be/AJU22XPZaOo?si=BRedzvI84BjFZCi3
- https://youtu.be/hi6U3IgCfI8?si=hOPYmtKx9mqtO9kk
- Henry Ford Picture Retrieved from: https://tr.wikipedia.org/wiki/Henry_Fordt

Thank You