hazelcast®

Maintaining an Online Edge in Today's Highly Competitive Banking World

How BNP Paribas was able to leverage in-memory speed to meet strict customer-facing SLAs.

BNP Paribas Bank Polska S.A. Case Study

Introduction

Banking has always been a complicated business for many reasons. Consider the external forces of strict regulations, tough global competition, continually rising customer expectations, and the growing dependence of online systems and the associated risks of breaches. Also, consider common yet comprehensive business activities such as ongoing mergers and acquisitions. The industry continues to evolve, in which increased concern over data privacy, and new strategic initiatives around digital transformation, add to the complexity.

More than ever, a priority for banks today is to find the right balance between the proven reliability of legacy systems and new innovative technologies that will carry the institutions into the future. The former maintains a stable business operation that helps retain customer trust, but the latter introduces new opportunities to gain a competitive advantage. The search for new competitive advantage is especially important considering the entrants in the market, particularly the newer fintech companies that offer digital payment processing and personal finance tracking that have traditionally been in the domain of large institutions.

Fortunately, many technology vendors are recognizing the stringent requirements of banks and the financial services industry as a whole. As one example, open source software is increasingly relevant as prospective users get to investigate hands-on with no up-front commitment, while the commercial vendors who back the open source products provide premium functionality (around business continuity and security) that demanding environments need.

Background

BNP Paribas S.A. is a French international banking group and one of the largest banks in the world by total assets. With roots of its first founding in 1848, the current institution was formed by the merger of Banque Nationale de Paris (BNP) and Paribas in 2000, and is one of the three major international French banks.

It has both retail and investment banking operations, with the former business unit serving more than 30 million customers in its three domestic markets of France, Belgium, and Italy. It has a presence in 72 countries and operates in the western United States as Bank of the West, and in Poland as BNP Paribas Bank Polska S.A.

In the last few years, the IT team of BNP Paribas Bank Polska turned to the use of in-memory technologies to solve critical business issues that involved systems integration across merged companies, as well as architectural modernization to expand new opportunities. They are also exploring how stream processing technologies can be leveraged with hot initiatives like machine learning to further modernize their architecture.

The comprehensive IT department in the Poland division is largely self-sufficient from the rest of the parent company. The team consists of many technical experts with a variety of skill sets and responsibilities such as architecture definition, technology recommendation, standards definition, research, development, systems integration, and DevOps. They operate as an agile department to support several specific business areas in the bank: retail, corporate investment, central (core) banking, and personal finance.

In their Internet banking initiative, they were looking to leverage microservices, containers, and more DevOps resources to boost the team's agility. At the same time, they owned very stable core systems and wanted to make sure they could integrate modern technologies for newer initiatives without impacting the legacy systems. The aim was to reconcile both core stability and new high velocity initiatives.

The team was familiar with how in-memory technologies could be used to address the bank's needs. They were new to Hazelcast and had deeper familiarity with products from other vendors such as Redis Labs, GigaSpaces, and Oracle (Coherence). Due to limitations and pricing of the other technologies, team members experimented with Hazelcast, and the results of a proof-of-concept (POC) led to additional use of Hazelcast in their systems.

Challenge

The team's use of Hazelcast IMDG came in two distinct stages. The first entailed migrating customer data from a newly acquired bank's systems into the BNP Paribas infrastructure within strict regulatory and time constraints. The second entailed developing a new architecture that could support the Internet banking initiative without disrupting the already-reliable systems in the core banking group.

Data Migration and Reconciliation to Support a Business Merger

In the first Hazelcast deployment, the team was responsible for merging customer data from a newly acquired bank into the BNP Paribas central core banking system. With so much data and the constantly changing nature of the data, migration was not a simple process. As data was being moved to the core system, new data was being created in the source system, which meant the team had to recheck the integrity of the data after the initial bulk migration and make adjustments to fully update the data in the core systems. This "technical reconciliation" process required tracking new data after the initial migration, finding the differences such as new accounts, how much money was deposited/withdrawn, etc., and then applying those differences. Since this was customer financial information, which was obviously critically important to get right, there was no margin for error.

The challenge was not necessarily the mechanics and logistics of migrating the data, but rather, getting all the data moved over and reconciled within a short time frame. Performance was a critical requirement and using traditional migration processes (ETL/ELT) was too slow and therefore too risky. They turned to the open source version of Hazelcast to provide the processing acceleration they felt was required to make the migration successful. The ability to use open source Hazelcast with no up-front commitment meant that they could quickly and easily validate its capabilities in their environments.

Architecture Modernization

The second major deployment of Hazelcast entailed their architecture modernization to support online/ Internet banking initiatives. The IT team was well aware of the competition in the Polish online banking landscape, as well as the raised consumer expectations as a result of innovations in online applications from vendors in all industries. So they needed a system that delivered high performance and scale to deliver a strong customer experience across all online interfaces.

The team was focused on maintaining agility for the retail banking business but wanted to minimally impact the stable core systems that drove core banking operations. This meant that they wanted to create an architectural strategy that would keep the core systems intact while building an online architecture that relied upon data in the core systems without overwhelming the core systems.

The plan was to create an abstraction layer via REST services to let the front-end developers for Internet banking access information in core banking systems as quickly as possible with low latency and workload overhead. Certainly, they could have continued with a service-oriented architecture (SOA) using a request-response paradigm on existing technologies, but they faced two main problems with that approach. First, the cost of integration would be high because it required expertise in the underlying systems. Second, the growing number of customers and requests would likely overwhelm the operation of the core banking systems and the SOA bus, as well as exposing only a subset of the data in the core systems for the sake of mitigating load was not an option.

On existing systems, average response time for querying core banking systems was around one second, with some results at three seconds. This was unacceptable to the front-end teams focused on delivering the best customer experience. They were asking for responses in the 100 millisecond range. To the infrastructure team, this did not seem possible, but they were prepared to do further research with Hazelcast.

While any caching system could seemingly solve their problems, the team identified a couple of major problems with using a typical caching pattern. As a start, the cost of storing so much data could make the system infeasible. Also, data synchronization would be a problem as the bank needed to minimize the discrepancies between the data that is sent to the customer and the data that actually exists in their records. The chance for stale data in conventional caching systems was too risky. The system needed to be real-time so that everyone had access to the same information. It also had to be reusable across the bank, and thus very robust and easy to set up, to handle many different applications and use cases.

Solution

The IT team was able to implement Hazelcast successfully to build the solutions they needed. Both of the aforementioned initiatives took advantage of in-memory speeds to deliver on stakeholder requirements.

For the migration project, Hazelcast was used as a very fast operational data store in a cluster of 16 virtual machines, each with 8 cores and 20 GB of RAM for a total of 130 GB allocated to the migration data. The POC was created in two weeks, which allowed them to assess whether they would be able to perform the migration in the expected time window.

For the Internet banking project, the team built an architecture that leveraged Hazelcast as a modern caching solution that not only provided all the data available in the core banking systems to the front-end developers but also met the requirements around performance in terms of responsiveness and around minimal workload impact to the core banking systems. Data synchronization with the core systems (i.e., the systems of record) was done through Hazelcast, which made it easy to ensure the latest data was delivered to customers.

Results

For the migration project, the team was able to complete all the work over the weekend, thus minimizing the impact on daily operations. By meeting their service-level agreement (SLA), they not only accomplished the challenging task, but also avoided disastrous consequences of burdening operational systems during peak business hours. The fact that they were able to build their POC in two weeks also contributed to the success of this business-critical project.

	Response Time (milliseconds)	Throughput (requests/second)
Required Figures	Max: 200 ms	1500
SOA Test Results	Avg: 500 ms Min: 250 ms	200
Hazelcast POC Results	Avg: 50 ms	17,000
Hazelcast Production Results (on 4 bare metal servers, 8 CPU, 128 GB RAM)	Avg: 0.8 ms	84,000

This significant performance boost (200x faster!) provided all stakeholders, including the front-end developers, the performance necessary to continually build advanced and innovative applications for BNP Paribas customers. At the same time, the core systems were minimally impacted, allowing them to continue handling their daily operational load without disruption from the workloads on the Internet banking systems.

Overall, Hazelcast was easy to deploy and very stable with no issues in production. Data is replicated in real time not only between nodes for high availability but also across data centers to deliver an extra level of availability should a data center go down. With the two clusters, they were able to achieve zero downtime. At one point in the two years in production, they had a major network issue, but that did not affect Hazelcast at all, as it properly handled the network partition and the resulting split-brain state.

More interest grew throughout the bank to use Hazelcast, including the need for a repository for card information and a cache for identity management. Though Hazelcast was not perceived yet to be a mainstream technology, the justification to move forward ultimately was an easy argument. The team soon built out more than 10 repositories. Hazelcast is very low maintenance in production, so they had a framework for easy future deployments. Technical support was great, and they felt like all their questions were being answered. Now, whenever they create new microservices, they use Hazelcast for high-speed data storage.

Opportunities for Open Banking?

While the BNP Paribas IT team was mostly focused on the bank's Internet banking systems, the scalable and future-proofed foundation they created is applicable to other banking requirements. Most notably, open banking is an emerging trend and it relies on architectures that are technically similar to what the BNP Paribas IT team built. Since open banking largely entails exposing application programming interfaces (APIs) with appropriate authorization controls to third-party applications, performance and scale will be critical characteristics for handling the added load. Overall, the relatively small investment the team made on this project will most certainly pay off with even bigger outcomes in the months and years ahead.

> *"In addition to the performance gain, Hazelcast was super easy to deploy and very stable with no issues. We had the out-of-the box replication between nodes and between data centers, so we had a highly available architecture."*
>
> ***Szymon Domagala***
> *Enterprise Architect, BNP Paribas*

The required maximum response time was set at 200 ms, and Hazelcast delivered average response times of 0.8 ms. The 84,000 requests per second from Hazelcast was well above the SLA-defined 1500 requests per second. Additionally, the performance scales linearly with the cluster size.

Additional Success

The team uses Hazelcast Jet for their real-time streaming architecture. They use Jet in conjunction with IMDG as the fast repository for stream data enrichment. With tens of millions of transactional and behavioral events per day, their system needs to be fast and resilient. More information will be provided in an upcoming case study on this project entailing their real-time marketing platform that sends offers to customers when specific conditions are met.

hazelcast®

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

@hazelcast
https://www.linkedin.com/company/hazelcast
https://www.facebook.com/hazelcast