Seminar Report

History and Development of System Architectures

Humaira Azam

Matriculation No: 26210266

Supervisor: Narges Lux

Georg-August-Universität Göttingen
Institute of Computer Science

September 30, 2023

Abstract

This report talks about how different types of computer systems are important for doing data analysis efficiently. It explains how these systems have changed over time to handle today's data challenges. It also discusses different architectures including Shared-Nothing, Massively Parallel Processing Architectures (MPPA), In-Memory Computing, and Hybrid Architectures, characteristics and their applications. It also talks about new trends like better computer hardware (GPUs, FPGAs), cloud-based systems (microservices, containerization), and serverless computing and how they affect how well data analysis works.

This report also talks about the challenges in large-scale data handling, trends, and innovations in system architecture. Several challenges are also mentioned in the report, such as real-time performance analysis, system integration, and system security.

In addition, The case studies of Google's Big Table, Apache Kafka, and Amazon Redshift are described. Finally, it shows how these systems are used in different sectors like finance, healthcare, and online shopping to do things like fraud detection, monitoring data of patients, giving personalized recommendations, and making supply chains work better.

In conclusion, the choice of system architecture greatly influences speed, scalability, and cost-effectiveness of data analytics. With ongoing innovations, organizations can leverage data analytics by using modern system architectures.

List of Abbreviations

HPDA High-Performance Data Analytics

SNA Share Nothing Architectures

MPPA Massively Parallel Processing Architectures

AI Artificial Intelligence

AWS Amazon Web Services

GPUs Graphics Processing Units

FPGA Field Programmable Gate Array

Faas Function as a Service

Contents

1 Introduction

In today's world, where we have lots of data and fancy tools to analyze it, being able to analyze data really well is super important. Many different kinds of companies are using data to make intelligent decisions for customer satisfaction and stay ahead of the competition. But in the background, there is this high-powered computer system that enables us to process large amounts of data. These systems are like the foundation of business processing and they help us to process data, analyze it, and give us useful information.
Computer systems are good at analyzing data quickly but the things are really important to know about like how they've changed over time, why they're so important, what challenges they face, what new things are happening, how they're used in the real world, and what might happen in the future.
High-performance data analytics system architectures have found applications in various industries, enabling organizations to extract valuable insights, make informed decisions, and drive innovation. It's really important to understand what different industries need when it comes to data. For example, real-life stories show how these computer systems are making a real difference in fields like finance, healthcare, and online shopping. They're helping with things like giving personalized recommendations, fraud detection[11], patient tracking[3], and making supply chains work better.
In the fast-moving world of technology, it's crucial to know about the latest trends and new ideas. By using the right kind of computer system and making the most of data, companies can get better at analyzing things, get useful information, and make their businesses better[10].

2 Development in Data Analytics

2.1 Evolution of Data Analytics

The history of data analytics traces back to the early days of computing, where data processing was a relatively straightforward endeavour involving batch processing systems.[6] These systems were primarily used for simple calculations and reporting purposes, typically involving structured data in tabular formats. As the complexity and scale of data increased over time, so did the need for more advanced data analytics techniques and systems[6].

2.2 Limitations of Traditional Systems

Traditional data analytics systems, while revolutionary in their time, faced significant limitations[17] as data requirements became more demanding:

1. **Scalability:** Traditional systems struggled to efficiently scale to handle the growing volumes of data. The hardware and software architectures of these systems often hinder their ability to process large datasets[17].

2. **Speed and Efficiency:** As datasets grew larger, the processing speed of traditional systems slowed down due to limitations in hardware capabilities. Disk I/O latency,

caused by the need to fetch data from disk storage, became a bottleneck for data processing speed[17].

3. **Complexity and Data Variety:** Traditional systems were optimized for structured data, and handling unstructured or semi-structured data was a challenge. New data types and formats required more sophisticated processing techniques[17].

2.3 Improvements in Traditional Systems

To address the limitations of traditional systems[24], various improvements were introduced:

1. **Scalable Architectures:** The advent of distributed computing frameworks, such as Apache Hadoop and Spark, allowed data processing tasks to be distributed across clusters of machines, enabling horizontal scalability[24].

2. **In-Memory Computing:** In-memory computing techniques emerged, where data is stored in the system's main memory (RAM) instead of on disk. This dramatically reduces the data access latency associated with disk-based storage and speeds up data processing[24].

3. **Optimized Data Storage:** Columnar storage and compression techniques optimized the way data was stored on disk, allowing for faster retrieval and reducing the disk I/O bottleneck[24].

4. **Parallel Processing:** Modern systems adopted parallel processing paradigms, allowing multiple tasks or computations to be executed simultaneously, significantly improving overall processing speed[24].

5. **Fault Tolerance:** Traditional systems were often susceptible to failures that could interrupt data processing. New fault recovery mechanisms, such as data replication and distributed computing, were introduced to ensure continuous operations even in the presence of failures[24].

3 Shared-Nothing Architecture

Share Nothing Architectures (SNA) is a distributed computer architecture where individual computers(nodes) work together. In this setup, we have many individual nodes interconnected with each other using a network system. Each node works as an individual unit with its own memory and storage space. The nodes in this system are connected using a high powered reliable network connection and adding new nodes is relatively easy[22].

In a shared-nothing architecture, a large data set is split into smaller sets. Each segment goes to one of these computer (nodes) in different parts of the system. The data is partitioned horizontally, the incremental growth of the system is possible and new individual nodes can be added to increase the transmission capacity[21].

3.1 Characteristics of Shared-Nothing Architecture:

1. **Data Distribution:** In a shared-nothing architecture, data is distributed across multiple nodes in a distributed computing environment. Each node has its own local memory and storage, and there is no central shared storage or memory unit[22][21].

2. **Parallel Processing:** The architecture enables parallel processing by allowing each node to independently process its portion of the data. This leads to improved performance and faster data processing[22][21].

3. **Scalability:** Shared-nothing architectures are highly scalable as new nodes can be added to the system without requiring significant changes to the existing infrastructure. This is essential for handling growing data volumes[22].

4. **Fault Tolerance:** The architecture inherently provides fault tolerance since the failure of one node does not affect the functioning of other nodes. Data redundancy and replication can be employed to ensure data availability in case of node failures[21][22].

3.2 Example: Google's Bigtable

Google's Bigtable is a great example of a system that uses shared-nothing architecture. Bigtable is like a big computer storage system spread across many regular computer parts. It's made to handle lots and lots of organized information and is used by Google for things like Google Search, Google Maps, and YouTube. Shared-nothing architecture, which is what Bigtable uses, is a smart way to deal with huge amounts of data. It works by spreading the data out, doing many things at once, and being able to grow when needed. It's also good at handling problems and doing complicated tasks quickly. This makes it perfect for modern apps that need to work fast, use resources efficiently, and not break easily[19][4].

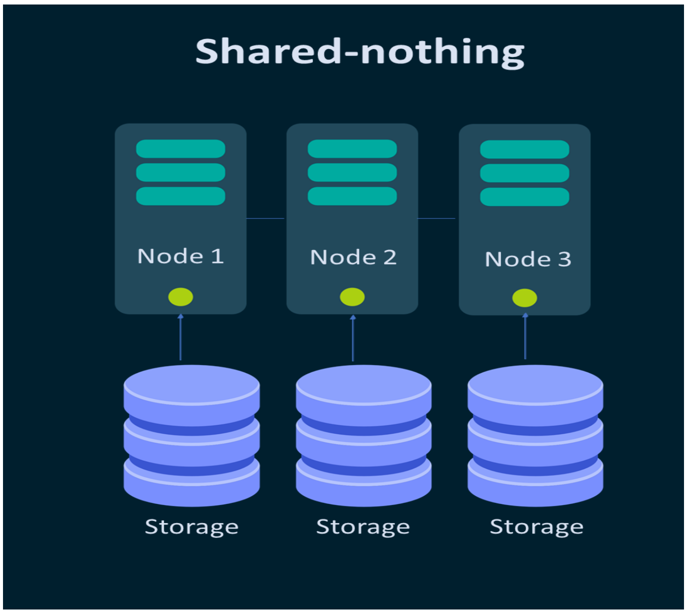

Figure 1: Shared Nothing Architecture[23]

4 Massively Parallel Processing (MPP) Architecture

MPPA is a coordinated way of processing a single task. It does this by using many processors, each with its own OS and memory. These processors talk to each other through messages. Sometimes, you can have as many as 200 or even more processors all working together on the same task. They connect through special data paths[7].
Setting up MPP is a bit tricky. You have to decide how to split up a big database among the processors and give each processor its part of the job. An MPP system is also called a "loosely coupled" or "shared nothing" system because the processors work more independently.
MPP is great for programs that need to search through lots of databases at the same time, like decision support systems and data warehouses. It's better at this than symmetrically parallel systems[7][25][8].

4.1 Characteristics of MPP Architecture:

1. **Parallelism:** MPP architecture divides data and processing tasks into smaller chunks that can be processed in parallel across multiple nodes or processing units. This parallelism significantly improves query performance and data processing speed[7].

2. **Scalability:** MPP systems are designed to scale horizontally by adding more nodes to the cluster. This scalability allows for handling ever-increasing data volumes and processing demands[8].

3. **Optimized Query Processing:** MPP systems often use query optimization techniques to distribute query workload across nodes efficiently, minimizing data movement and maximizing query performance[25].

4. **Complex Queries:** MPP architecture excels in handling complex queries involving joins, aggregations, and transformations. These queries can be divided into subtasks that are processed in parallel[7][8][25].

4.2 Example: Amazon Redshift

Amazon Redshift is a well-known computer system that uses a special way of working called MPP (massively parallel processing). It's made for looking at big amounts of data with SQL queries and is very good for things like business reports and data analysis.
MPP architecture, like what Amazon Redshift uses, is a strong method for handling and studying lots of data quickly. It works by doing many things at once, can get bigger when needed, and is really good at understanding complex questions. It also doesn't break easily and is great for jobs like storing and analyzing data, where we need fast answers and efficient work[1].

5 In-Memory Computing

In-memory computing is like using special software that lets you keep data in the computer's fast memory (RAM), spread out across many computers, and work on it all at once. Think of regular data that's usually stored in one big database. With in-memory computing, you can put that data in the super-fast RAM of lots of computers. RAM is about 5,000 times quicker than old-fashioned spinning disks. And when you add the ability to do many tasks at the same time, things become incredibly fast – like, really, really fast[12]. In-memory computing relies on two important things: storing data in RAM and doing tasks across many computers. While we expect in-memory tech to use RAM, it's also about spreading the work and data out for super-speedy processing. This way, we can get things done much faster[15].

5.1 Characteristics of In-Memory Computing:

1. **Data Storage:** In-memory computing involves storing data directly in the system's main memory (RAM) rather than on traditional disk storage. This eliminates the need to access data from slower disk storage, significantly reducing data retrieval latency[15].

2. **Data Processing Speed:** Accessing data from RAM is much faster than fetching it from disk, resulting in dramatically improved data processing speed and query performance[12].

3. **Real-Time Analysis:** In-memory computing enables real-time or near-real-time analysis of data. This is particularly advantageous for applications that require instant insights or rapid decision-making[12][15].

4. **Complex Analytics:** The speed of in-memory computing allows for the efficient execution of complex analytical operations, such as real-time aggregations, joins, and machine learning algorithms[12].

5.2 Example: SAP HANA

SAP HANA is a great example of a super-fast computer system that works with data differently. It's like a super quick database and computer program that lets companies deal with really big amounts of data instantly. In-memory computing, like what SAP HANA does, is a new way of working with data. It's all about storing data super fast, doing tasks really quickly, analyzing things in real-time (right away), and handling complicated questions. This is perfect for jobs that need speedy answers and efficient data work[34].

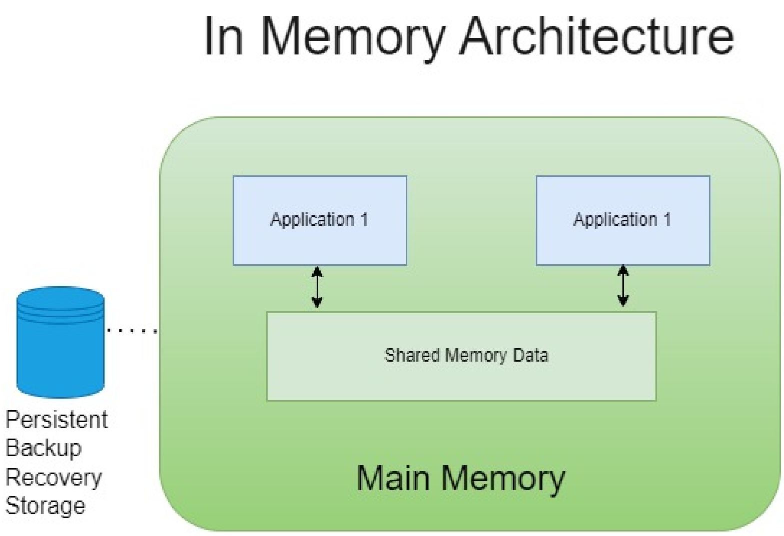

Figure 2: In Memory Architecture, With in-memory computing, computers pool their RAM together into a shared memory space[12].

6 Hybrid Architectures

In-memory computation means doing computer calculations mainly in a computer's fast memory, like RAM. This is usually for big and complicated calculations. It needs special computer software to work, and it often involves many computers working together as a group.
When they work together, these computers use all their memory (RAM) as one big memory space. This helps them do the calculations faster and handle really big tasks[32].
Lots of companies use the internet to get computer resources from public cloud services. They use these services to do some of their work. At the same time, they keep some of their work on their own private cloud, like a personal cloud just for them. They do this for different reasons, like saving money, following rules, or using specific technology. The most popular public cloud services for this kind of thing are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. These are like the big names that many companies trust for their cloud needs[32].

6.1 Characteristics of Hybrid Architectures:

1. **Combination of Approaches:** Hybrid architectures combine multiple architectural paradigms, such as shared-nothing, MPP, and in-memory computing, to leverage the strengths of each approach[32].

2. **Flexibility:** Hybrid architectures allow organizations to tailor their data processing strategies to the specific needs of different parts of their data ecosystem. This flexibility accommodates varying workloads and data characteristics[32].

3. **Optimized Performance:** By leveraging different architectures for different tasks, hybrid systems can optimize performance based on the nature of the workload. For example, using in-memory computing for real-time analytics and MPP for complex batch processing[32].

6.2 Example: Microsoft Azure Synapse Analytics (formerly SQL Data Warehouse)

Microsoft Azure Synapse Analytics is a type of computer service that mixes two things: a place to store and organize data (like a data warehouse) and the ability to handle really big data and connect different pieces of data.
Hybrid architectures, like what Azure Synapse Analytics uses, are good because they can do lots of different data tasks in a flexible way. They can use different methods to get the job done, make things work well, and grow when needed. This is helpful for businesses that have all kinds of data needs. Hybrid architectures let them use different ways of doing things and build smart and flexible data systems[29].

7 Trends and Innovations

7.1 Hardware Advancements

1. **Graphics Processing Units (GPUs):** GPUs have gained prominence in data analytics due to their parallel processing capabilities. Originally designed for rendering graphics, GPUs excel in performing mathematical computations in parallel. In data analytics, GPUs are leveraged for tasks such as machine learning training, deep learning, and simulations that require extensive parallelism. The massive parallel processing power of GPUs enables faster execution of data-intensive tasks, contributing to quicker insights and analysis[26].

2. **Field Programmable Gate Array (FPGA):** Field-Programmable Gate Arrays (FPGAs) are reconfigurable hardware devices that can be customized to perform specific tasks efficiently. FPGAs are being utilized in data analytics to accelerate specific algorithms and computations. They offer a balance between performance and power efficiency by allowing customization of hardware for specific workloads, making them suitable for real-time analytics and high-speed data processing[9].

7.2 Cloud-Native Architectures

1. **Microservices:** Microservices architecture involves breaking down applications into small, independent services that can be developed, deployed and scaled independently. This architecture enhances agility and allows for efficient development and maintenance of data analytics systems. In microservices-based systems, each component can be optimized for its specific task, leading to improved performance and scalability.

2. **Containerization:** Containerization technology, such as Docker and Kubernetes, has revolutionized how applications and services are deployed and managed. Containers encapsulate software and its dependencies, ensuring consistency across different environments. In data analytics, containerization facilitates seamless deployment and scaling of data processing pipelines, enabling better resource utilization and simplified management[18].

7.3 Serverless Computing

1. **Function as a Service (Faas):** Serverless computing, represented by Function as a Service (FaaS) platforms like Amazon Web Services (AWS) Lambda and Azure Functions, allows developers to execute code in response to events without managing server infrastructure. This paradigm shift eliminates the need for provisioning and managing servers, enabling developers to focus solely on writing code. In data analytics, FaaS can be utilized for event-driven processing, real-time analysis, and data transformation[31].

2. **Event-Driven Architecture:** Event-driven architecture is closely tied to serverless computing and microservices. In this architecture, components respond to events or messages, allowing for loosely coupled systems that can quickly adapt to changing data scenarios. Event-driven architecture is well-suited for real-time data processing, stream analytics, and handling asynchronous data updates[27].

Recent trends and innovations in hardware advancements, cloud-native architectures, and serverless computing have transformed the landscape of data analytics. These developments enhance performance, scalability, resource efficiency, and real-time analytics capabilities, allowing organizations to harness the power of data more effectively[27][31][26][9][18].

8 Challenges

8.1 Scalability

As data volumes and processing demands continue to grow exponentially, designing systems that can scale efficiently without compromising performance is a major challenge. Balancing the distribution of workloads across nodes while avoiding bottlenecks requires careful design and optimization[20]. In a distributed environment, load balancing becomes crucial to ensure that resources are evenly utilized across nodes. Uneven workloads can lead to degraded performance and inefficiencies, necessitating intelligent load distribution mechanisms[20].

8.2 Data Integration and Interoperability

Integrating data from diverse sources, formats, and systems while ensuring data consistency and accuracy is a complex task. Data normalization involves reconciling discrepancies in terminology, units, and schemas to create a unified view of the data. Overcoming data fragments, where data is fragmented across different departments or systems, requires creating a cohesive data architecture that enables seamless data sharing and integration[16].

8.3 Security and Privacy

Safeguarding sensitive data from unauthorized access, breaches, and cyberattacks is a critical challenge. Designing architectures with strong security measures, including encryption, access controls, and authentication mechanisms, is essential to ensure data protection. Adhering to data privacy regulations such as GDPR, HIPAA, and CCPA adds complexity to system architecture design. Systems must be designed to comply with these regulations

while still providing efficient and useful analytics.
The design of data analytics system architectures is accompanied by numerous challenges, each requiring careful consideration and innovative solutions. Scalability, data integration, security, and real-time analytics are among the most critical challenges organizations face. Addressing these challenges effectively is paramount to building robust and future-proof data analytics systems that can meet the demands of modern data environments[16].

9 Future Directions

9.1 Edge Computing

Edge computing involves processing data closer to the source, reducing the latency associated with sending data to centralized cloud servers. This is especially beneficial for real-time analytics and applications that require immediate responses. Edge architectures enable processing and analysis at the edge devices themselves. This enhances scalability and responsiveness, as edge devices can perform tasks locally without relying heavily on centralized data centres[30].

9.2 Quantum Computing

Quantum computing has the potential to revolutionize data analytics by leveraging quantum bits (qubits) to solve complex problems exponentially faster than classical computers. This could lead to breakthroughs in optimization, machine learning, and cryptography[33]. Quantum architectures could handle complex data analytics tasks that are currently computationally infeasible. This could have profound implications for large-scale simulations, pattern recognition, and data analysis involving vast datasets[33].

9.3 Artificial Intelligence(AI) Integration

Integrating Artificial Intelligence (AI) technologies like machine learning and deep learning into system architectures can enhance data analytics capabilities. AI-driven architectures can automate data preprocessing, feature extraction, and model training, leading to more accurate insights. AI-driven architectures can provide predictive insights by learning from historical data patterns. This enables organizations to make data-driven decisions based on future projections[2].

9.4 Blockchain Technology

Blockchain's decentralized and immutable nature can enhance the integrity and trustworthiness of data analytics processes. Transactions and changes are recorded in a tamper-proof manner, ensuring data quality. Blockchain-based architectures can enable secure and decentralized data sharing, governance, and collaboration among multiple parties. This is particularly relevant in industries requiring data collaboration while maintaining data ownership[5].
The future of data analytics system architecture holds exciting possibilities driven by emerging technologies. Edge computing, quantum computing, AI integration, and blockchain technology have the potential to reshape how data is processed, analyzed, and shared.

These innovations promise reduced latency, faster problem-solving, enhanced capabilities, and improved data integrity. As organizations strive to extract meaningful insights from vast and complex datasets, these future directions offer the potential to unlock new levels of efficiency, intelligence, and trust in data analytics processes[2][30][33][5].

10 Case Study: Google's Bigtable

Google Bigtable is a special computer system that can handle a massive amount of organized data, even as big as a petabyte. It's built to quickly and reliably access data, and it can grow really big if needed.Google uses Bigtable for some important services like Google Earth and keeping track of websites on the internet. But it's important to know that Bigtable is not like a regular database you might be familiar with.Google started working on Bigtable in 2004, and in 2016, they made it available for others to use in the cloud[19]. At its core, Bigtable is like a big map that can manage a huge amount of data and handle lots of actions like reading and writing data very quickly. It's great for organizing data in a way that's different from regular databases. Bigtable is a type of NoSQL database, and it's like others such as Cassandra and Hbase. It organizes data in a way that's different from regular databases. Instead of using columns like a table, it groups them into "families" and stores them together[19].

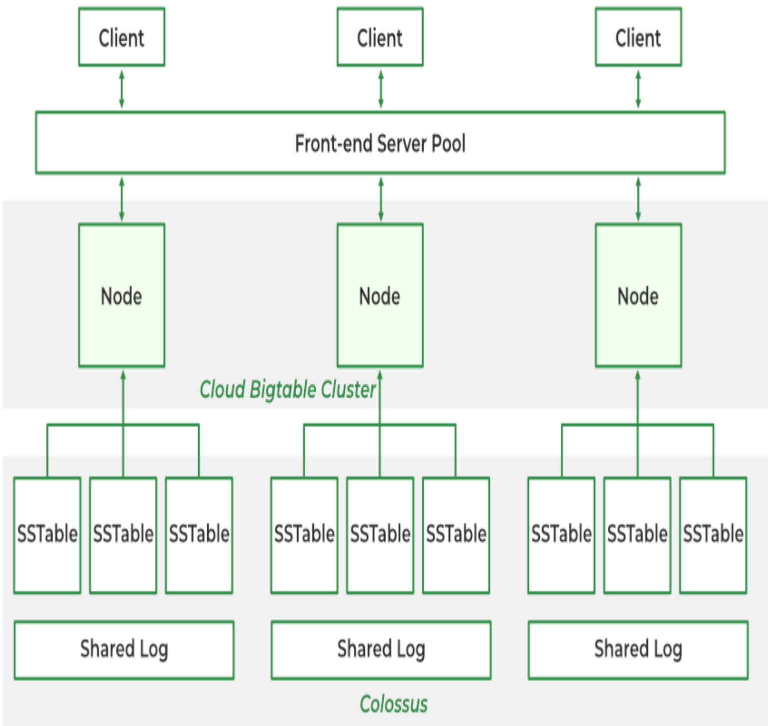

Figure 3: Google Big Table Architecture[13]

Google's Bigtable is a remarkable example of how an innovative system architecture can revolutionize data analytics capabilities at scale. Its ability to efficiently store, retrieve, and analyze massive datasets while ensuring scalability and reliability has enabled Google to maintain its position as a leader in the digital age[4][19].

11 Case Study: Apache Kafka

Apache Kafka, also known as Kafka, is like a computer tool that helps make applications that can respond to things happening in real time. In our world today, there are lots of things producing a continuous flow of data, like when you buy something online, pick a seat on a flight, or even when a thermostat tells you the temperature. Each of these things happening is called an "event," and it's like a digital record of what happened and when it happened. So, Kafka helps organize and handle all these events, making it easier for computer programs to react quickly when something important happens. It's not just about people; even machines can produce events, like that thermostat telling you the temperature[28].

Apache Kafka's architecture and features have made it a powerful tool for enabling high-performance data analytics through real-time data streaming. Its scalability, fault tolerance, and low-latency design have transformed the way organizations handle and analyze data, opening new possibilities for real-time insights and data-driven decision-making.

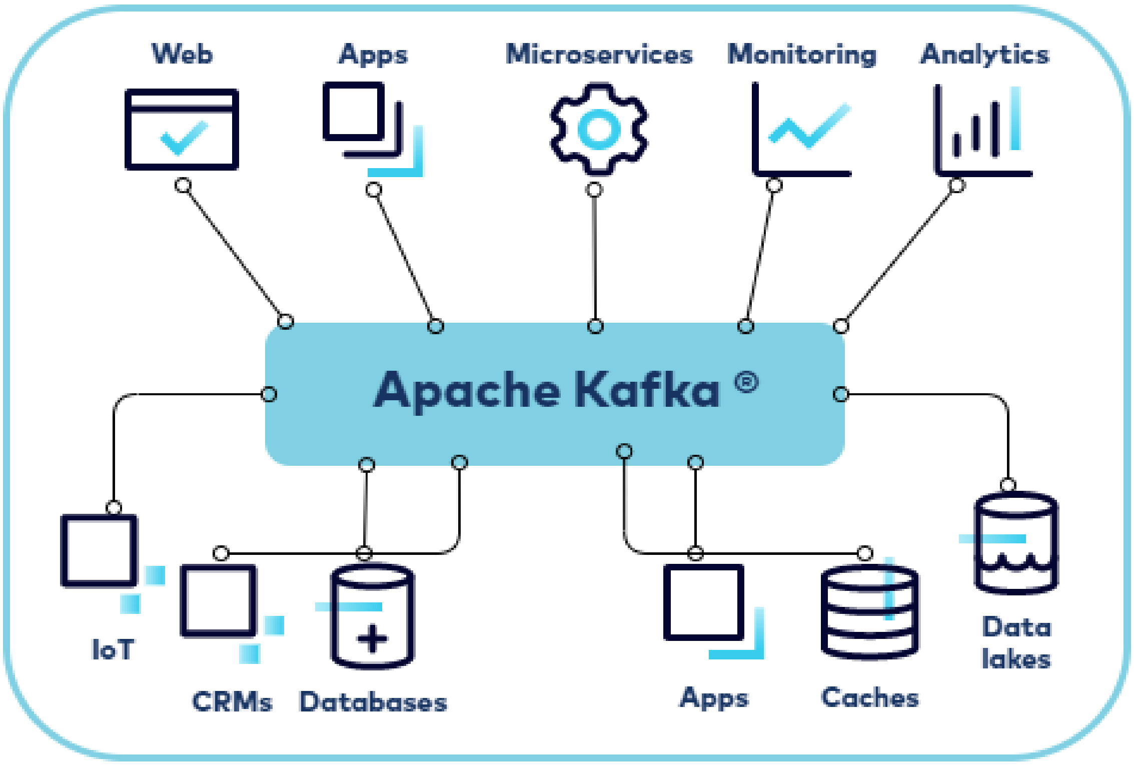

Figure 4: Apache Kafka Architecture[14]

12 Applications of HPDA Architectures

High-Performance Data Analytics (HPDA) architectures have found widespread application across various industries, transforming the way organizations process, analyze, and derive insights from large volumes of data. Let's delve into specific real-world applications in the fields of finance, healthcare, and e-commerce.

12.1 Finance: Detecting Fraudulent Activities

The financial industry faces a significant challenge due to the large number of daily transactions, which makes it vulnerable to potential fraudulent activities that must be identified in real-time[11].
The solution to this challenge lies in the adoption of high-performance data analytics architectures, such as real-time stream processing. These technologies empower financial institutions to examine transactions as they happen. By leveraging sophisticated algorithms and machine learning models, they can promptly identify patterns of fraudulent behavior. These advanced architectures enable organizations to react swiftly, thus averting financial losses.
The impact of implementing real-time fraud detection is substantial. It reduces the risk of financial losses stemming from fraudulent activities, builds greater trust among customers, and ensures compliance with industry regulations.

12.2 Healthcare: Patient Monitoring and Analytics

In the healthcare sector, there's a critical challenge: analyzing patient data quickly to ensure timely interventions and better patient outcomes.
The solution to this challenge lies in the adoption of high-performance data analytics architectures. These systems allow for real-time monitoring and analysis of patient data, including vital signs, electronic health records, and data from wearable devices. By processing this data in real-time, healthcare providers can spot irregularities, predict potential health issues, and take action promptly to prevent problems[11].
The impact of implementing real-time patient monitoring and analytics is significant. It enables the early detection of health issues, leading to improved patient care, fewer hospital readmissions, and a more efficient allocation of medical resources[11].

12.3 Healthcare: Early Warning Systems for Infectious Diseases

The challenge we face is the need for quick identification and response to infectious disease outbreaks to prevent them from spreading.
The solution to this challenge involves the use of high-performance data analytics architectures. These systems gather data from different sources, like medical records, lab results, and even social media, to spot the early signs of infectious disease outbreaks. Using advanced algorithms, they can detect unusual patterns, track disease trends, and send alerts to health authorities for immediate action[3].
The impact of these early warning systems is profound. They enable health authorities to respond swiftly to outbreaks, implement containment measures, and reduce the impact of infectious diseases on public health[3].

12.4 E-commerce: Personalized Product Recommendations

The challenge in the world of E-commerce is dealing with massive amounts of customer data that can be used to make the shopping experience better.
The solution to this challenge involves using high-performance data analytics systems[10]. These systems analyze customer data, which includes things like how people browse, what they buy, and their personal information. By studying this data, E-commerce platforms can give customers personalized suggestions for products they might like. This makes

customers more interested and leads to more sales.
The impact of these personalized recommendations is significant. They make customers happier, boost sales, and encourage loyalty by making the shopping experience fit each person's preferences[10].

High-performance data analytics architectures have transformed various industries by addressing specific challenges and enabling real-time analysis of large datasets. These applications showcase the power of data analytics in driving informed decision-making, enhancing customer experiences, and improving operational efficiency across diverse sectors.

13 Conclusion

The world of high-performance data analysis has come a long way, and the type of computer system you choose can make a big difference in how successful your data-driven projects are. This choice affects things like how fast you can process data, how well your system can grow, how it handles problems, and how much it costs. If you design your system well, it can help you get the most out of data analysis, giving you better information to make decisions.
Technology is changing fast, and it's giving us new ways to design computer systems. There are better computer parts, ways to use the internet, and ways to run programs without worrying about the hardware. These innovations let organizations handle bigger sets of data, work with data in real-time, and grow without using too many resources. Staying updated on these changes is very important. The world of data analysis keeps evolving, with new tools and methods popping up all the time. Being informed helps organizations make smart choices about their computer systems, making sure they can use data analysis to its fullest.
In the business world, being able to understand data and get useful insights from it is a big advantage. Picking the right computer system can be a game-changer. It lets organizations process data faster, grow efficiently, and get meaningful information to make important decisions. By embracing new ways of designing computer systems and staying updated on industry changes, organizations can put themselves ahead in the world of high-performance data analysis, getting valuable insights and growing steadily.

References

[1] *Amazon Redshift.* URL: https://aws.amazon.com/redshift/faqs/.

[2] *Artificial Intelligence Integration.* URL: https://en.wikipedia.org/wiki/Artificial_intelligence_systems_integration.

[3] Kornelia Batko and Andrzej Ślęzak. "The use of Big Data Analytics in healthcare". In: *Journal of big Data* 9.1 (2022), p. 3.

[4] *Bigtable overview.* URL: https://cloud.google.com/bigtable/docs/overview.

[5] *Block chain Technology.* URL: https://www.ibm.com/topics/blockchain.

[6] *Evolution of Data Analytics.* URL: https://ciosea.economictimes.indiatimes.com/news/big-data/the-evolution-of-data-analytics/93430964?redirect=1.

[7] *explaination of massively parallel processing.* URL: https://www.techtarget.com/whatis/definition/MPP-massively-parallel-processing.

[8] *figure of massively parallel processing.* URL: https://www.researchgate.net/figure/Architecture-of-MPP-databases-There-are-a-master-and-4-slaves-with-2-segments-on-each_fig7_338059218.

[9] *FPGA.* URL: https://en.wikipedia.org/wiki/Field-programmable_gate_array.

[10] Ahmad Ghandour. "Big Data Driven E-Commerce Architecture". In: *International Journal of Economics, Commerce Management (IJECM)* 3 (May 2015).

[11] Wishmy Ikhsan et al. "Fraud detection automation through data analytics and artificial intelligence". In: *Riset* 4 (Sept. 2022), pp. 103–119. DOI: 10.37641/riset.v4i2.166.

[12] *In memory computing.* URL: https://hazelcast.com/glossary/in-memory-computation/.

[13] *Introduction to Google Cloud Bigtable.* URL: https://www.geeksforgeeks.org/introduction-to-google-cloud-bigtable/.

[14] *Introduction to Kafka.* URL: https://docs.confluent.io/kafka/introduction.html.

[15] Nikita Ivanov. *What Is In-Memory Computing?* URL: https://www.gridgain.com/resources/blog/what-is-in-memory-computing.

[16] Anirudh Kadadi et al. "Challenges of data integration and interoperability in big data". In: *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014* (Jan. 2015), pp. 38–40. DOI: 10.1109/BigData.2014.7004486.

[17] *limitations of traditional systems of Data Analytics.* URL: https://www.purestorage.com/knowledge/why-traditional-storage-fails-big-data.html.

[18] Jon Mukaj. "Containerization: Revolutionizing Software Development and Deployment Through Microservices Architecture Using Docker and Kubernetes". PhD thesis. July 2023. DOI: 10.13140/RG.2.2.23804.51841.

[19] Matt Palmer. *What is Google Bigtable?* URL: https://www.zuar.com/blog/what-is-google-bigtable/#how-does-bigtable-work.

[20] *Scalability challenge*. URL: https://www.designgurus.io/blog/grokking-system-design-scalability.

[21] *Shared Nothing Architecture Explained*. URL: https://www.geeksforgeeks.org/difference-between-shared-nothing-architecture-and-shared-disk-architecture/.

[22] Sofija Simic. *Shared Nothing Architecture Explained*. URL: https://phoenixnap.com/kb/shared-nothing-architecture.

[23] *Snowflake, Redshift, BigQuery, and Others: Cloud Data Warehouse Tools Compared*. URL: https://www.altexsoft.com/blog/snowflake-redshift-bigquery-data-warehouse-tools/.

[24] *Traditional Data Systems*. URL: https://www.pearsonitcertification.com/articles/article.aspx?p=2427073&seqNum=2.

[25] *Transitioning from SMP to MPP, the why and the how*. URL: https://cloudblogs.microsoft.com/sqlserver/2014/07/30/transitioning-from-smp-to-mpp-the-why-and-the-how/.

[26] *What Is a GPU?* URL: https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html..

[27] *What is an Event-Driven Architecture?* URL: https://aws.amazon.com/event-driven-architecture.

[28] *What is ApacheKafka*. URL: https://www.ibm.com/topics/apache-kafka.

[29] *What is dedicated SQL pool (formerly SQL DW) in Azure Synapse Analytics?* URL: https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-overview-what-is.

[30] *What is edge computing?* URL: https://www.ericsson.com/en/edge-computing.

[31] *What is FaaS?* URL: https://www.ibm.com/topics/faas.

[32] *What is Hybrid cloud architecture*. URL: https://www.vmware.com/in/topics/glossary/content/hybrid-cloud-architecture.html.

[33] *What is quantum computing?* URL: https://aws.amazon.com/what-is/quantum-computing/.

[34] *what is SAP HANA*. URL: https://www.sap.com/products/technology-platform/hana/what-is-sap-hana.html.