Real-time data analytics for Enhanced Decision–Making for the Entertainment Industry

A Comprehensive Review

Sai Ram Oduri, A20522183

soduri@hawk.iit.edu

ABSTRACT

This study investigates the integration of real-time data warehousing (DWH), big data streaming, and Explainable Artificial Intelligence (XAI) to help commercial organizations make better decisions in a competitive landscape. Recognizing the widespread usage of these technologies for competitive advantage, the research focuses on the effective management of big data to enable real-time stream processing inside data warehousing frameworks. The study performs a comprehensive analysis of the literature, thoroughly investigating recent breakthroughs and problems in real-time stream processing systems. The study addresses the need for data engineering in addressing problems and creating solutions for real-time DWH and big data stream processing. This study adds to a more comprehensive knowledge of the symbiotic link between data engineering, big data, and XAI, and provides insights for the application of real-time stream processing frameworks across varied data streams in modern corporate contexts. We will primarily take an example of Netflix to make use of deep research done in the platform.

INTRODUCTION

In big media businesses where decision-making for the consumer is extremely important, real-time data analytics are key leverages for businesses to utilize. Processing the continuous stream of data for real-time analysis is a challenge when we are adopting the ETL stage for Data warehousing due to the nature of big data applications concerning volume, variety, velocity, volatility, variability, veracity, and value.

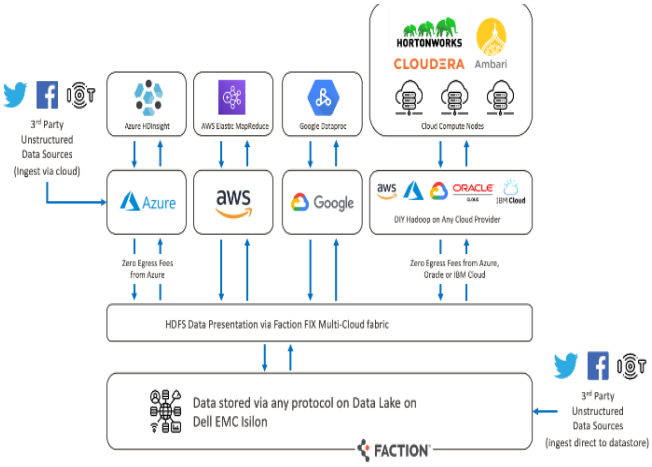


Fig a. Hadoop Architecture and tools

The benefits of real-time data warehousing in the entertainment industry include:

1. **Improved business agility:** Real-time data warehousing reduces time lag in business processes, allowing organizations to respond to changes and take advantage of opportunities faster.
2. **Enhanced decision-making:** Real-time data warehousing enables immediate access to current and historical data, allowing companies to spot opportunities and adjust their strategy across different aspects of the business, thus improving decision-making processes.

3. **Personalized marketing**: Real-time analytics, enabled by data warehousing, allows for the processing and analysis of data in extremely short periods of time, leading to personalized marketing, customer sentiment analysis, and recommendation engines in the entertainment industry.
4. **Content optimization:** Big data analytics, facilitated by real-time data warehousing, helps media companies interact directly with their audience through pre-scheduled video streaming, and assists in pinpointing the precise content that customers would like to connect with regularly, thus improving customer interaction and content optimization.
5. **Real-time data processing and analysis**: Modernizing a data warehouse to support real-time data processing and analysis enables businesses to make faster and more informed decisions, as it allows for low-latency data processing and analysis and provides immediate insights to decision-makers and reporting purposes.[1]

Defining streaming data is necessary, in this case, it is continuous supply of big data to the streaming database. There is continuous generation of massive data from the consumers when watching content, playing games anything related to streaming. The metadata where there is heterogeneous data gathered with a semi-structured, unstructured, processing or analysis of value of data when considered in its availability or the time-taken to arrive/ freshness of data generated. To address the challenges various approaches have been developed so far from the join algorithms such as stream processing, Streamhouse technologies for data latency, and distributed streaming with ETL, DWH, processing framework, and multi-join query processing in cloud DWH.

Many XAI approaches concentrate on answering certain questions or qualities of the explaining ability of a specific type of model. Following that, different XAI approaches may give different interpretations for the same occurrences. Without examination, it is difficult to determine whether the explanation is trustworthy. With the variety of XAI techniques, the choice to pick an XAI method and the subsequent development of the XAI method with model assessment is more than a single activity, but a comprehensive process.

Methodology

I. DATA WAREHOUSING

We will explore what is known as – 'Data Mesh'[2] used in Netflix, which is used by the data platform team where infrastructure is used company-wide to process data at scale. A data mesh refers to a comprehensively managed, streaming data pipeline solution designed to facilitate Change Data Capture (CDC) applications. General use cases include – studying and extracting events from generic applications, using DB connectors like Cassandra, using processing algorithms such as projection, unions, joins other aggregation procedures.

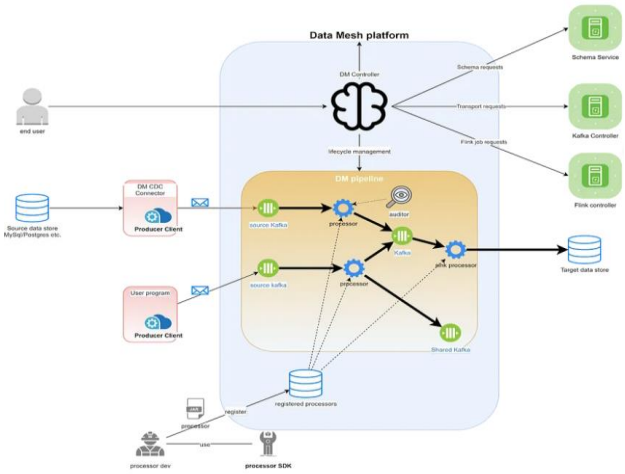


Fig Data Mesh System[b]

In this above figure we can see that - data mesh CDC connector which connects to Kafka, user

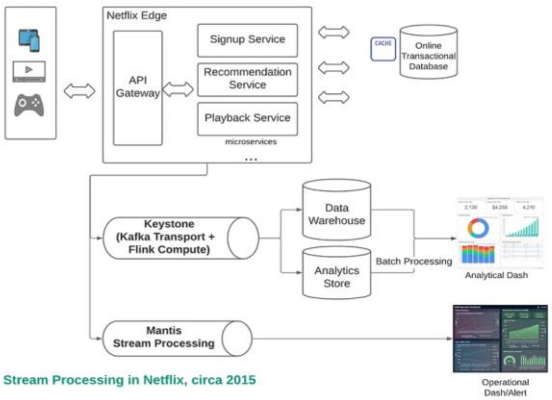
program connects to source Kafka. The module receives data such as events, logs, across the application into the Kafka then the pipeline performs the actual heavy lifting data processing work and the corresponding microservices manage the life cycle. We know from the above architecture that various sources' data is flowing into the pipeline. There is need for transformations in the data to make the data fit for analysis and meaningful insights. Then it eventually goes into the destination data store. This is all the primary data warehousing. If we were using Google Cloud, we can leverage the data into a Dataflow pipeline and use BigQuery database to perform meaningful analysis for data engineering tasks, where the controllers figure out the resources and adjust the configurations.

What are processors here? Any Flink job here is a processor which is utilized for reading events from upstream transports and does transformations on the data. Parallely there are several intermediate processors writing data to another transports. Finally, the data is subscribed to a Kafka topic, or ElasticSearch.

In 2022, Netflix data pipeline processed approximately 500 billion events per day, which is equivalent to 1.3 petabytes of data per day[3].

Ensuring the reliability and efficiency of data ingestion pipelines is paramount, as they play a crucial role in handling streams of events. The necessity arises for frameworks that can deliver these events in a distributed manner, ensuring scalability and load distribution across cluster nodes. The ingestion framework must persist incoming data into both memory and disk for future consumption, resembling a producer-consumer system. These tools operate as message buses in the integration pipeline without altering the data itself. Spark serves as our chosen stream processing framework, while Cassandra, a columnar NoSQL database, functions as the data storage system. Below are brief descriptions of each tool:

1. **Apache Kafka:** Renowned for real-time data retrieval, Kafka excels at ingesting large volumes of high-velocity data, requiring fast, fault-tolerant, and distributed pipelines. Acting as a central hub for real-time processing, Kafka replaces traditional message queue systems like Rabbit MQ and IBM MQ due to its superior throughput, reliability, and replication capabilities.
2. **Apache Spark and Storm:** Spark and Storm stand out as popular distributed stream processing computation frameworks.
3. **Cassandra:** Chosen as a columnar NoSQL database, Cassandra efficiently stores vast amounts of data across multiple commodity servers, ensuring high availability with no single point of failure.
4. **Flume:** Functioning as an agent-based framework, Flume facilitates information gathering from diverse sources, integrating them to form an enterprise data lake. Highly adept at collecting and aggregating substantial amounts of data from various sources into a centralized data store, Flume employs JVM processes as agents.



(Figure: How does stream processing help with operational and analytical data)

Stream Processing in Netflix(circa 2015)

In a streaming data pipeline, issues such as component failures or data loss can occur, leading to the need for backfills. Backfilling

involves re-ingesting or reprocessing data to fill gaps in the dataset caused by issues in the pipeline. **In Cockroach DB**, **a change feed is a feature that allows you to track changes to a database in real-time**. By creating a new change feed with the initial scan turned on, Cockroach DB enables a more efficient way to perform ad hoc backfills. The change feed captures changes to the data, including the initial state, effectively reconstructing the dataset. The change feed mechanism simplifies the process of backfilling, making it more manageable. Cockroach DB's change feeds provide real-time visibility into data changes, facilitating prompt responses to issues.[4]

II. DATA INGESTION IN NETFLIX

Then we step into the data ingestion pipelines for building the recommendation system in Netflix. This is the biggest cash grab for Netflix as it hooks up customers to the content platform, builds continuous engagement with the platform.

Marken Architecture

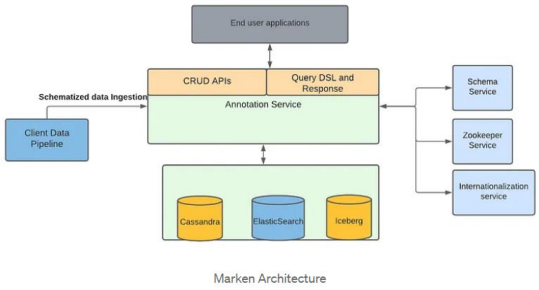


Fig C Marken architecture

Designing data pipelines involves the separation of producer flows from client teams, ensuring clients are not burdened with concerns about data availability or writing procedures. This **decoupling strategy** simplifies the interaction between producers and consumers of data.

The **lifecycle of a movie encompasses various creative stages**, with multiple temporary files delivered before reaching the final movie file. Additionally, movies often involve different languages, each with its set of delivered files. Teams aim to run algorithms and create annotations using these diverse media files.

```
{
  "annotationOperationKeys": [
    {
      "annotationType": "string",
      "annotationTypeVersion": "integer",
      "pivotId": "string",
      "operationNumber": "integer"
    }
  ],
  "id": "UUID",
  "operationStatus": "STARTED",
  "isActive": true
}
```

Sample Code[2]

To facilitate, the algorithm runs on different permutations of media files, we can streamline the process using the following parameters:

- **Annotation Schema Type:** Identifies the schema for annotations generated by the algorithm.
- **Annotation Schema Version:** Specifies the schema version of the generated annotations.
- **PivotId:** A unique identifier for the file or method used to generate annotations, such as the SHA hash of the file or the movie identifier number.
- **OperationNumber:** An auto-incremented number for each new operation.

1. In **StartAnnotationOperation**, the operation is stored with its OperationKey in a STARTED state, and the corresponding OperationIDs in this state are efficiently cached in EVCache for quick searches. Users then utilize **UpsertAnnotationsInOperation** to update annotations in the operation, passing both annotations and the OperationID.

2. During this phase, operations are in an ACTIVE

and STARTED state, accommodating the creation of 2K to 5K annotations in a single run.

3. The **FinishAnnotationOperation** call concludes the process by marking the current operation as FINISHED and ACTIVE, removing its ID from Memcache. Any prior ACTIVE operation is then marked as not active in Cassandra.

4. Finally, the **updateByQuery** API in **ElasticSearch** is invoked to mark associated documents as not active. This organized sequence ensures a smooth, efficient, and scalable execution of annotation operations.

Search API:

- Excludes annotations from operations with isAnnotationOperationActive = FALSE or operations in the STARTED state.
- Implements filters in ElasticSearch queries and queries EVCache to exclude relevant annotations.

Error Handling:

- If an error occurs in Cassandra, the client call fails.
- Elasticsearch errors are handled with a retry logic for update Query calls. Failed calls trigger messages to SQS for automated retries after intervals.

This approach ensures a robust and efficient data pipeline for managing movie annotations.

III. DEPLOYMENT ORCHESTRATION USING FLINK

The declarative reconciliation protocol, integral across the architectural stack, ensures a singular user-declared goal state stored in a durable Source of Truth Store, currently utilizing AWS RDS. In case of state conflicts or disruptions, this source is treated as authoritative, facilitating eventual system-wide reconciliation. This design allows for seamless recreation of clusters, like

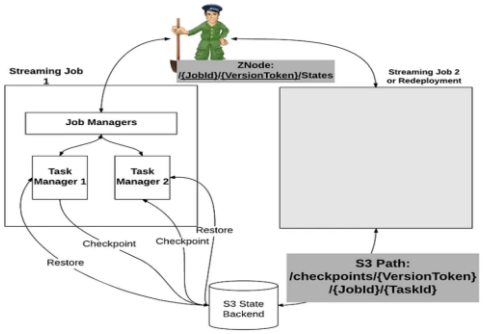


Fig d. Streaming Job Flows

Kafka, and ensures continuous self-healing and automated operations. Idempotent operations enhance resilience, enabling services to autonomously reconcile, fostering operational agility.

Flink cluster has job managers and task managers. There is a independent Flink cluster for each instance level isolation, with only zookeeper for shared service coordination and S3 for storage of checkpoint states**. Ververica is a data artisan** that makes use of Stream house. It provides a near real-time latency solution for high-latency batch jobs in stream house. This enables Netflix to evaluate its compatibility by executing operations and measuring outcomes.

The advantages of using Streamhouse reduces the latency from days to minutes [5] and **leveraging Flink SQL** offers unparalleled flexibility in navigating the dynamic landscape between Stream housing and real-time stream-based evolving requirements. Flink is apt technology due to its batch and stream unified architecture. Developers have made a new component called **Flink CDC** which is for batch and stream data ingestion. Flink CDC makes use of a wide range of connectors from MySQL to Amazon Aurora, where data operations are performed and ingested into Kafka, Iceberg etc. It supports a wide range of APIs such that large data gets ingested into sharded databases for availability. This also involves optimization where checkpoints are involved such as buffer

debloating and GIC allowing for faster checkpointing where file sizes are reduced significantly by 95%[5] .

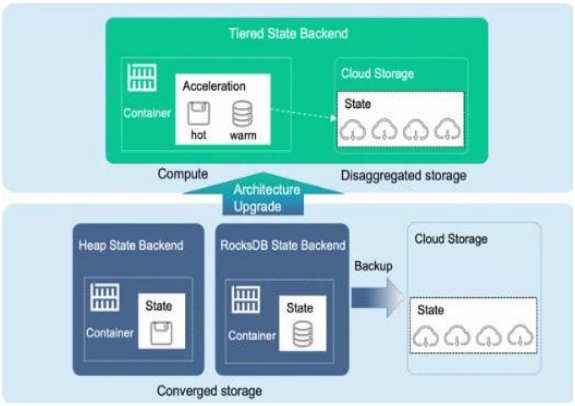


Fig Apache Flink 2.0 – Tiered State Backed Architecture (ref. Ververica Cloud)

In Flink 2.0, the primary goal is API Evolution, involving restructuring into four categories: Low-Level, High-Level, Connector, and Operation APIs. The evolution occurs in two phases. Phase one includes removing legacy elements and cleaning up deprecated classes. Phase two involves redesigning the API with a new query able state, refactoring the DataStream API, and introducing a streamlined configuration layer.

V. Challenges & Solutions

To address the small-file problem at Netflix, engineering efforts focused on the Keystone Router, a crucial component distributing 3 trillion daily events across 2,000 routing jobs and 200,000 parallel operators to various data sinks in Netflix's S3 repository. This includes Hive, Elasticsearch, and a Kafka consumer. Storm's record delivery system, which ensures messages are delivered at least once, does not provide a guarantee of maintaining the correct order of message transmission. Trident, on the other hand, requires adherence to a specific transaction order for its proper functioning.[4]The challenge arose as numerous Flink operators were generating small S3 files after each checkpoint operation, causing suboptimal performance for S3. A solution emerged with the use of **Flink's memory threshold feature**. By adjusting the default from **1KB to 1MB**, Netflix directed the job manager to consolidate state acknowledgments from operators into one substantial checkpoint file. This approach eliminated the need for operators to write to state, with only the job manager handling the S3 writes.

Facing a challenge where half of the jobs in the pipeline ended without writing S3 files due to path issues, Netflix encountered an inefficiency with 13,000 unnecessary metadata requests being sent without file writes. To address this, Netflix collaborated with Flink to implement a change in version 1.2.1. The modification involved calling the Checkpoint Stream Factory only once during the operator's initialization, rather than every 30 seconds, reducing wasteful metadata requests and enhancing efficiency.[6]

VI. Comments for XAI for feeding data into recommendation systems

Now that all the data is ingested into Netflix, and users are targeted for advertising and recommendations. This is how Netflix is maintaining success in the entertainment industry.[9] There are certain rules Netflix will have to follow to abide regarding the users content viewership. There will be strict content filtering employed if there is a kid's profile to no such filtering if a user is watching R rated or explicit content. However, every time a user is engaged, we have to make decisions to personalize his homepage, the content he has viewed. The recommendation system is like a decision-making system where AI is capable of taking decisions to leverage content to the users.

XAI Criterion – General effect in social settings

Teams at Netflix(or any organization) employ a artifact-specific approach that allows to target various elements such as content filtering, personalization, sizzles(short previews) which involves a decentralized infrastructure allowing for transparency. For such an organization to produce shows that maintain big capital over how youngsters mainly teenagers view their relationships, academics, career. To address such challenges, Netflix uses Digital Marketing to target users' interests and by involving with the current trends in the world, it shapes the thinking of a generation.

First the organization to maintain transparency, it has to allow third parties to allow auditing data. All the data ingested in the pipeline which goes into the recommendation can be moderated by XAI, where **content moderation teams feed data why they have chosen to show such data only to users.** Now adding transparency allows for a user to know how such artwork has appeared on his page. By collection of users history and using feedback loops into the recommendation pipeline, where again the data is fed into the algorithm.

But auditing of content should also be done outside the chain of command, where third parties can voice their opinion. EBA ensures high-risk systems surpass regulatory benchmarks and encourages inclusivity for low-risk vendors. Additionally, an EBA framework aligning processes with ethical standards draws from data regulatory bodies, exemplified by CNIL's privacy impact assessment. Nonetheless, these frameworks may focus on specific ethical concerns, warranting caution due to diverse normative standards.

1. Employ algorithms that automatically flag content using predefined rules for every user before the data is fed into recommendation pipeline, based on his watch history.

2. Employing meta data analysis using tags, user interactions can provide content moderation.

These algorithms, often operating as black-box models, benefit from XAI methods like feature masking, feature mutation, and simplification, making their decision-making more interpretable and accountable.[8]

XAI Process for feature contribution

1. **Hyperparameter Tuning:**

- **Use Case:** In the context of the recommendation system, hyperparameter tuning can involve optimizing parameters in the XAI methods that influence the interpretability and accuracy of explanations. For example, adjusting parameters related to the sensitivity of feature impact calculations in ALE or the granularity of feature variation in PDP.

2. **Explanation Consistency:**

- **Use Case:** Consistency in explanations refers to the stability of results across different runs or scenarios. In the recommendation system, this means that the XAI methods should provide similar explanations for similar situations. For instance, if a user is repeatedly recommended similar content, the explanation for these recommendations should remain consistent.

3. **User-Friendly Presentation:**

- **Use Case**: In the recommendation system, presenting XAI results in a user-friendly manner involves:

- **Providing explanations in plain language:** Translate complex XAI insights into understandable terms for users who may not have a technical background.

- **Visual aids:** Use charts, graphs, or other visualizations to represent how different

features influence content suggestions, making it easier for users to comprehend.

- **Personalization:** Tailor the presentation of explanations to align with the user's preferences, ensuring that the information is relevant and meaningful to them.

4. Iterative Improvement with User Feedback:

- Use Case: Actively seek user feedback on the presented explanations in the recommendation system. If users express confusion or dissatisfaction with certain explanations, iterate on the XAI methods to address these concerns. This could involve refining the algorithms, adjusting parameters, or incorporating new interpretability techniques based on user input. [5]

EBA Process for developing 'Good' recommendations.

The inherent complexity of normative concepts, such as fairness and justice, poses significant conceptual challenges in deciding what data flows into recommendation systems / Netflix's user home page. Balancing ethical alignment with respect for pluralism and addressing trade-offs in interpreting ethical principles are imperative. The autonomous and complex nature of data ingestion and recommendation introduces technical constraints. Transparency in machine learning models remains a challenge, making responsibility assignments during content filtering audits difficult. Adaptive approaches and **'living traceability' methodologies attempt to reconcile these challenges.** Economic and social constraints underscore the need to align incentives with a normative vision for ethically sound decisions and recommendations. Since **'The Netflix Effect'**[7] is now a global phenomenon where users' behavior patterns follow a new show, it has to be kept in mind that content moderation and rigorous personalization is not the only solution. But geographical interests play a major role. Questions about the distribution of benefits and burdens, potential impacts on smaller companies, and adversarial behavior during audits require nuanced solutions. Netflix's **sole** content moderation constraints highlight the necessity for a transparent and well-recognized EBA process where third parties and auditors are allowed to make decisions about what shows stay, what can be banned. But objectivity concerns, regulatory bodies, and the tension between national jurisdictions and global technologies complicate the institutional framework.

VII. CONCLUSION

In conclusion, this comprehensive study delves into the intricate integration of real-time data warehousing, big data streaming, and Explainable Artificial Intelligence (XAI) to empower commercial organizations in making informed decisions within a competitive landscape. The research focuses on the efficient management of big data for real-time stream processing within data warehousing frameworks, addressing key research topics related to publication channels, implementation issues, tools/approaches for ETL stage challenges, and evidence provided in overcoming distinct challenges.

The study, using Netflix as a prime example, explores the sophisticated data mesh architecture employed for **Change Data Capture (CDC) applications**. Netflix's data pipeline, processing an astounding 500 billion events per day in 2022, showcases the significance of robust frameworks such as Apache Kafka, Apache Spark, Storm, Cassandra, and Flume. The recommendation system, a pivotal aspect for user engagement, involves intricate data pipelines and algorithms, demonstrating the

strategic importance of data engineering in content optimization.

Deployment orchestration using Flink, and the challenges encountered in Netflix's streaming data pipeline, such as the small-file problem, highlight the continuous efforts to enhance efficiency and reliability. The study also emphasizes the pivotal role of Explainable AI in the recommendation system, ensuring transparency, content moderation, and ethical decision-making. Recommendations for XAI implementation include employing algorithms to flag content, metadata analysis using tags, and leveraging interpretability methods for black-box models.

In addressing challenges, Netflix's commitment to transparency and external auditing is underscored. The paper suggests employing third-party audits for content moderation, ensuring adherence to ethical standards, and navigating complex normative concepts. The study recognizes the global impact of 'The Netflix Effect' and emphasizes the need for nuanced solutions considering geographical interests, economic and social constraints, and the balance between national jurisdictions and global technologies.

In essence, this study contributes to a deeper understanding of the symbiotic relationship between data engineering, big data, and XAI in modern corporate landscapes. The insights provided can guide organizations in leveraging real-time stream processing frameworks for enhanced decision-making, personalized marketing, and content optimization, ultimately fostering agility and innovation in the face of evolving business challenges.

VIII. CITATIONS AND ARTICLE SOURCES

1. Modernizing a data warehouse for real-time decisions by Stephen Catanzano https://www.techtarget.com/searchdatamanagement/opinion/Modernizing-a-data-warehouse-for-real-time-decisions
2. Data Mesh — A Data Movement and Processing Platform @ Netflix by Bo Lei, Guilherme Pires, James Shao, Kasturi Chatterjee, Sujay Jain, Vlad Sydorenko - https://netflixtechblog.com/data-mesh-a-data-movement-and-processing-platform-netflix-1288bcab2873
3. How Netflix Built Their Data Pipeline with Amazon Redshift by Donal Tobin https://www.integrate.io/blog/how-netflix-built-their-data-pipeline-with-amazon-redshift
4. Data Mesh: How Netflix moves and processes data from CockroachDB by Dan Kelly https://www.cockroachlabs.com/blog/netflix-data-mesh
5. Streamhouse Unveiled by Jing Ge https://www.ververica.com/blog/streamhouse-unveiled
6. How Netflix Optimized Flink for Massive Scale on AWS by Alex Woodie https://www.datanami.com/2018/04/30/how-netflix-optimized-flink-for-massive-scale-on-aws/
7. The Netflix effect: The most powerful content marketing channel in the world, by Fernando Amaral https://fernandoamaral.org/netflix-effect-content-marketing/
8. https://thinkml.ai/intelligence-on-netflix-how-netflix-is-using-ai-and-bigdata/
9. The Netflix Decision Making Model Is Why They're So Successful, Dan Pontefract https://www.forbes.com/sites/danpontefract/2019/02/04/the-netflix-decision-making-model-is-why-theyre-so-successful/?sh=599428a673bc

IX. REFERENCES

[1] N. Mohamed, J. Al-jaroodi, Real-Time Big Data Analytics: Applications and Challenges.

International Conference on High Performance Computing & Simulation (HPCS), 2014.

[2] https://netflixtechblog.com/data-ingestion-pipeline-with-operation-management-3c5c638740a8

[3] Seong-Eun Hong and Hwa-Jong Kim, "A comparative study of video recommender systems in big data era," *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria, 2016, pp. 125-127, doi: 10.1109/ICUFN.2016.7536999

[4] S. M. Nabeel Mustafa, M. Umer Farooque, M. Tahir, S. M. Khan and R. Qamar, "Frameworks, Applications and Challenges in Streaming Big Data Analytics: A Review," pp. 1-6, doi: 10.1109/ICONICS56716.2022.10100410.

[5] J. Huang, Z. Wang, D. Li and Y. Liu, "The Analysis and Development of an XAI Process on Feature Contribution Explanation," 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 2022, pp. 5039-5048, doi: 10.1109/BigData55660.2022.10020313.

X. IMAGE SOURCES

a. https://dta0yqvfnusiq.cloudfront.net/facti15983775/2020/06/Hadoop-multi-cloud-architecture-5ef3aa04313ec.png
b. https://miro.medium.com/v2/resize:fit:828/format:webp/0*-gA7_aFh4okv6Qp9
c. https://miro.medium.com/v2/resize:fit:828/format:webp/0*YNagRtOKICtNNMg2
d. https://netflixtechblog.com/keystone-real-time-stream-processing-platform-a3ee651812a