International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021, ISSN No: 2348-9510

AUTOMATED DISASTER DATA MANAGEMENT: ENHANCING SUPPLY CHAIN RESILIENCE THROUGH UNIX/BASH SCRIPTING FOR ETL PROCESS AUTOMATION

Abhiram Reddy Peddireddy
abhiramreddy2848@gmail.com

Utkarsh Mathur
mathur.utkarsh@gmail.com

Rahul Chaudhary
rahulchaudhary1183@gmail.com

Abstract

This paper presents framework, for streamlining the handling of disaster related data in order to bolster the resilience of supply chains. It leverages Unix/Bash scripting for ETL automation, which involves Extracting, Transforming and Loading data. Disaster data management encompasses the organized gathering, processing and examination of information pertaining to both human induced disasters. By managing disaster data organizations can better prepare for emergencies respond swiftly aid in recovery efforts and minimize risks by providing timely and accurate updates to stakeholders. Supply chain resilience refers to a supply chains ability to predict disruptions make preparations respond promptly when faced with challenges and recover efficiently. This document delves into how Unix/Bash scripting can automate ETL processes with benefits such as efficiency, adaptability, scalability and dependability. The methodology section outlines how ETL scripts are designed, implemented and integrated into a disaster data management framework. Additionally highlighted is an analysis using real world data that showcases enhancements, in processing speed accuracy of information updates frequency decision making agility cost effectiveness resulting from automation. The results emphasize the role that automation plays in boosting disaster data management practices and supply chain resilience.

Index Terms—Disaster Data Management, Supply Chain Re-silience, ETL Automation, Unix/Bash Scripting, Data Processing, Real-Time Data, Disaster Preparedness, Data Accuracy, Cost Efficiency, Automated Data Management.

I. INTRODUCTION

Disaster data management involves the organized gathering, processing and examination of information pertaining to types of calamities including events, like earthquakes, floods and hurricanes as well as human caused disasters such as industrial mishaps and terrorist incidents. The goal of disaster data management is to improve readiness, response capabilities, recovery processes and mitigation activities by providing pre- cise data to those involved. This involves identifying data origins ensuring data accuracy merging datasets and utilizing analytical methods to aid decision making [1].
Supply chain resilience refers to a supply chains capacity to predict, prepare for respond to and bounce back from disruptions. In the realm of disaster management the resilience of supply chains [7] is crucial as it ensures the flow of goods and services while minimizing losses and facilitating

International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021, ISSN No: 2348-9510

swift recovery operations [3]. Resilient supply chains are known for their flexibility, adaptability and ability to withstand shocks while maintaining effectiveness in circumstances [4].

Strengthening resilience requires tactics like diversifying sup- pliers keeping inventories on hand enhancing communication and collaboration, among stakeholders and implementing risk management strategies. [9].
The process of ETL (Extract, Transform, Load) is an ele- ment, in integrating and managing data. It includes extracting data from sources transforming it into a format and loading it into a target system like a data warehouse or database. Unix/Bash scripting is crucial for automating the ETL process and brings benefits:

- **Efficiency:** Unix/Bash scripts automate tasks reducing manual intervention and speeding up data processing [5].
- **Flexibility:** Scripts can be tailored to handle different data sources and formats making them adaptable to various ETL scenarios [6].
- **Scalability:** Automation scripts can be expanded to handle large amounts of data, crucial, in disaster data manage-ment where quick and accurate processing is vital [7] [8].
- **Reliability:** By reducing human error automation scripts improve the reliability and consistency of the ETL pro-cess.

a. Background

Supply chain resilience is defined as the ability of a supply chain to withstand, adapt, and recover from disruptions. The key components of supply chain resilience include:

- **Flexibility:** The ability to adapt to changes and reconfig- ure supply chain operations.
- **Redundancy:** Maintaining backup resources and alter- nate routes to ensure continuity.
- **Visibility:** Real-time monitoring and transparency across the supply chain.
- **Collaboration:** Strong relationships and communication among supply chain partners [10].

Introduction to ETL (Extract, Transform, Load) processes: The ETL process is a critical aspect of data management that involves:

- **Extraction:** Collecting data from various sources.
- **Transformation:** Converting the data into suitable format or structure.
- **Loading:** Inserting the transformed data into a target database or data warehouse.

Overview of Unix/Bash Scripting and its Benefits for Au- tomation: Unix/Bash scripting is a powerful tool for automat- ing tasks in the Unix/Linux environment. Benefits of using Unix/Bash scripting for automation include:

- **Ease of Use:** Simple syntax and powerful text processing capabilities.
- **Integration:** Seamless integration with other Unix/Linux tools and applications.
- **Efficiency:** Rapid execution of scripts, reducing the need for manual intervention [11].
- **Customizability:** Ability to tailor scripts to specific re- quirements and workflows [10].

II. LITERATURE REVIEW

The literature discussing disaster data management high- lights strategies to improve the accuracy and timeliness of information aiding decision making during disasters. An effec-tive approach involves data collection, processing and analysis. Mitigation Processes and Supply Chain Resilience: a comprehensive framework was created to emphasize how disaster management processes interact with capabilities, for enhancing supply chain resilience [1]. The study stresses the role of mitigation processes like risk assessment and proactive planning in bolstering supply chain resilience by identifying disruptions and implementing strategies to maintain function-ality before and, after disasters.
An integrated supply chain resilience framework was de-veloped, emphasizing the interaction between disaster man-agement processes and the essential capabilities for building resilience in supply chains.
Factors Influencing Humanitarian Supply Chains: A detailed examination of humanitarian supply chains identified 12 factors that significantly influence their effectiveness [2]. Factors such, as government assistance, strategic planning capabilities and ongoing evaluation of project advancement are crucial aspects emphasized in the research [9]. It underscores the significance of managing these factors to establish efficient aid relief initiatives. Specifically, government backing and strategic planning play roles in ensuring the availability and proper allocation of resources during disaster responses [8]. Framework for Supply Chain Resilience: A comprehen- sive

theoretical framework was proposed by researchers in 2017 to explain resilience in supply chain networks [3]. The framework identifies key enablers of supply chain resilience, including swift trust, information sharing, and public-private partnerships. Swift trust refers to the rapid development of trust between stakeholders during emergency situations, which is crucial for effective coordination and response. Information sharing ensures that all stakeholders have access to the nec-essary data to make informed decisions, while public-private partnerships leverage the strengths of both sectors to enhance overall resilience [11].
Cybersecurity in Supply Chains: Cybersecurity is another critical component of supply chain resilience. The review of literature on managing cyber risks in supply chains and developed a conceptual model highlighting the importance of IT, organizational, and supply chain security systems [9]. Their study emphasizes the need for standardized policies, collaborative strategies, and empirical models to enhance cyber-resilience in supply chains [4].
The literature emphasizes the importance of adopting ad- vanced technologies, fostering collaboration, and implement- ing robust cybersecurity practices to enhance supply chain resilience. These strategies are crucial for improving disaster data management and ensuring the continuity and robustness of supply chains in the face of disruptions [9]. Effective dis- aster data management, supported by advanced technological solutions and collaborative efforts, can significantly enhance the resilience and reliability of supply chains, ultimately lead- ing to more efficient, effective disaster response and recovery efforts.

III. METHODOLOGY

Design of ETL processes using Unix/Bash scripts

- The design of ETL processes using Unix/Bash scripts involves:
- Identifying data sources: Determining the origins of the data to be extracted.
- Developing extraction scripts: Creating scripts to auto- mate the data extraction process.
- Designing transformation logic: Implementing scripts to transform the data into the desired format.
- Creating loading scripts: Automating the loading of trans- formed data into the target system [13].

Data sources and disaster scenarios considered

Data sources for disaster data management may include sen- sors, satellite imagery, social media, and government databases [12]. Disaster scenarios considered may range from natural disasters like hurricanes and earthquakes to man-made disas- ters such as industrial accidents and terrorist attacks [11].

Tools and technologies used for data extraction, trans- formation, and loading

- The tools and technologies used for ETL processes include:
- Unix/Bash scripting: For automating data extraction, transformation, and loading.
- Database tools: Such as SQL for querying and managing data.
- Data integration platforms: For handling complex data workflows and transformations.

A. Implementation

Step-by-step process of implementing ETL scripts

Implementing ETL (Extract, Transform, Load) scripts using Unix/Bash involves a systematic approach to automate the data management process. The following steps outline the process:

1. **Planning and Design:**

- Identify Data Sources: Determine the various sources from which data will be extracted. These can include databases, APIs, flat files, or other data repositories.
- Define Transformation Logic: Outline the necessary transformations required to convert the raw data into a structured format suitable for analysis or storage.
- Specify Target Destination: Identify the target database or data warehouse where the transformed data will be loaded.

2. **Extraction:**

- Develop Bash scripts to automate the extraction of data from identified sources.
- Ensure the scripts are capable of handling different data formats and structures.

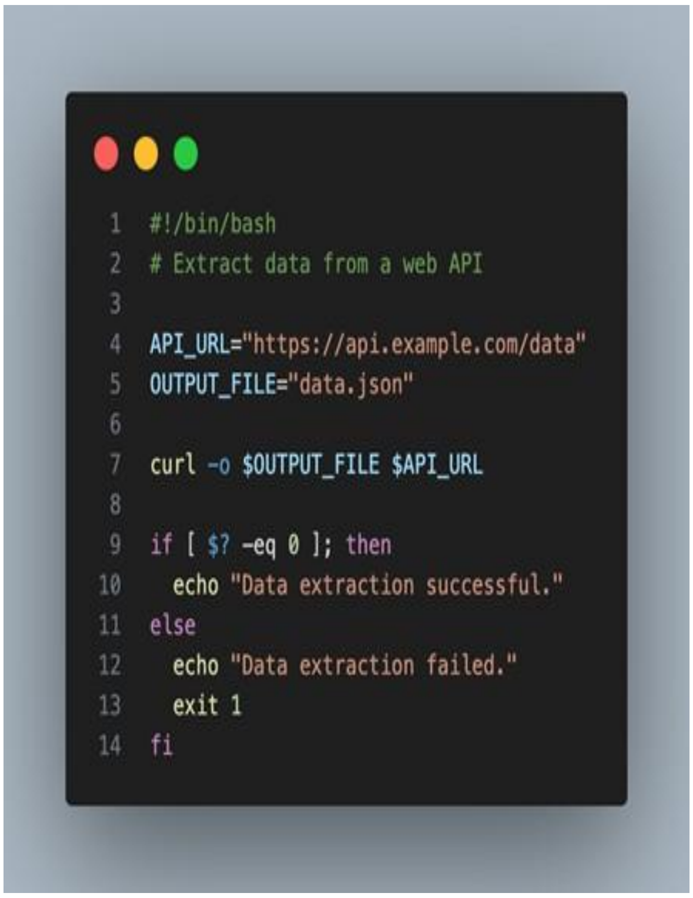
International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021, ISSN No: 2348-9510

3. **Transformation:**
- Utilize Unix tools like awk, sed, and grep to clean and transform the extracted data.
- Implement transformation logic to validate, normal- ize, and enrich the data as required.

4. **Loading:**
- Create scripts to automate the loading of trans- formed data into the target database or data ware- house.
- Optimize the loading process to ensure efficient data insertion and minimal impact on the target system's performance.

1) Examples of Unix/Bash scripts for different ETL stages:

- **Extraction Script**

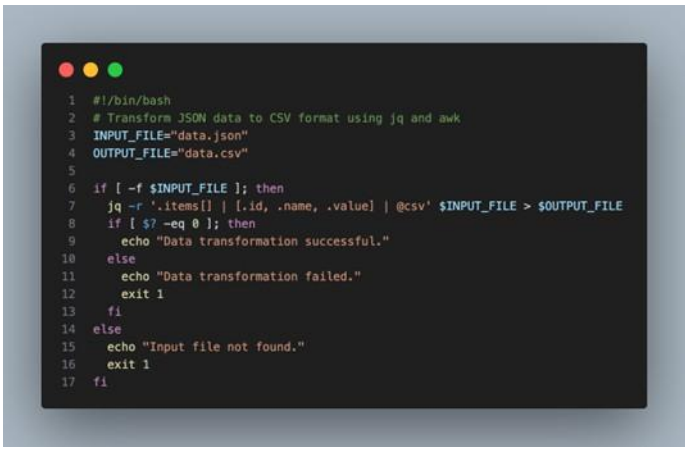


```bash
#!/bin/bash
# Extract data from a web API

API_URL="https://api.example.com/data"
OUTPUT_FILE="data.json"

curl -o $OUTPUT_FILE $API_URL

if [ $? -eq 0 ]; then
  echo "Data extraction successful."
else
  echo "Data extraction failed."
  exit 1
fi
```

Fig. 1. Extract data from a web API

This script uses the curl command to extract data from a web API and save it to a JSON file. It includes error handling to ensure successful data extraction.

- **Transformation Script**

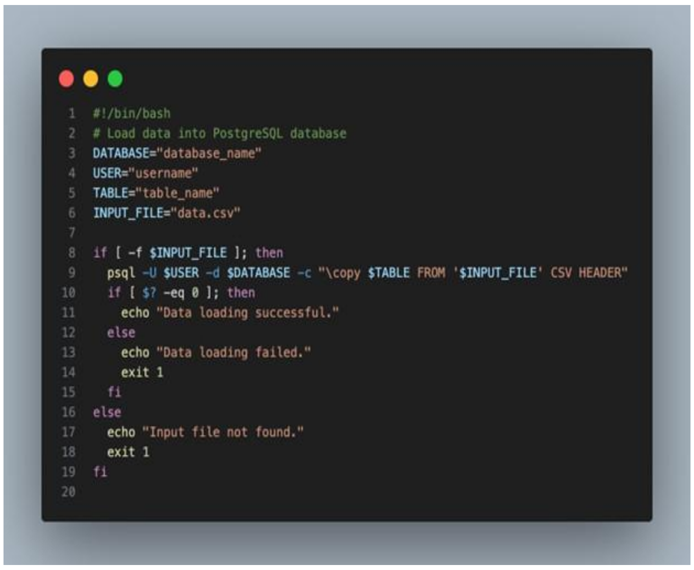


```bash
#!/bin/bash
# Transform JSON data to CSV format using jq and awk
INPUT_FILE="data.json"
OUTPUT_FILE="data.csv"

if [ -f $INPUT_FILE ]; then
  jq -r '.items[] | [.id, .name, .value] | @csv' $INPUT_FILE > $OUTPUT_FILE
  if [ $? -eq 0 ]; then
    echo "Data transformation successful."
  else
    echo "Data transformation failed."
    exit 1
  fi
else
  echo "Input file not found."
  exit 1
fi
```

Fig. 2. JSON to CSV Conversion

This script converts JSON data to CSV format using the ‘jq’ command. It ensures the input file exists and handles potential errors during the transformation process.

- **Loading Script**



```bash
#!/bin/bash
# Load data into PostgreSQL database
DATABASE="database_name"
USER="username"
TABLE="table_name"
INPUT_FILE="data.csv"

if [ -f $INPUT_FILE ]; then
  psql -U $USER -d $DATABASE -c "\copy $TABLE FROM '$INPUT_FILE' CSV HEADER"
  if [ $? -eq 0 ]; then
    echo "Data loading successful."
  else
    echo "Data loading failed."
    exit 1
  fi
else
  echo "Input file not found."
  exit 1
fi

```

Fig. 3. Data Loading into PostgreSQL

This script loads transformed CSV data into a PostgreSQL database using the psql command. It checks for the presence of the input file and ensures the loading process completes without errors.

2) Integration of scripts into the disaster data management framework:
To integrate the Unix/Bash ETL scripts into a comprehensive disaster data management framework, it is essential to automate, monitor, and log the entire process [10]. The following steps outline best practices for achieving this integration:

International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021, ISSN No: 2348-9510

1. **Automation and Scheduling:**
- Use cron jobs to schedule the execution of ETL scripts at regular intervals, ensuring that data is updated in a timely manner. For instance, a cron job can be set up to run the ETL script daily at midnight.
  Example: 0 0 * * * /path/to/etl script.sh

2. **Monitoring and Logging:**
- Implement logging within the scripts to capture execution details, including start and end times, errors, and the amount of data processed. This in- formation is crucial for monitoring the ETL process and troubleshooting any issues that arise.
- Logs should be written to a specific log file, and errors should be appropriately handled and logged.

3. **Error Handling:**
- Include robust error handling mechanisms to man- age issues that occur during script execution. This can involve retrying failed steps, sending notifica- tions to administrators, and logging detailed error messages.

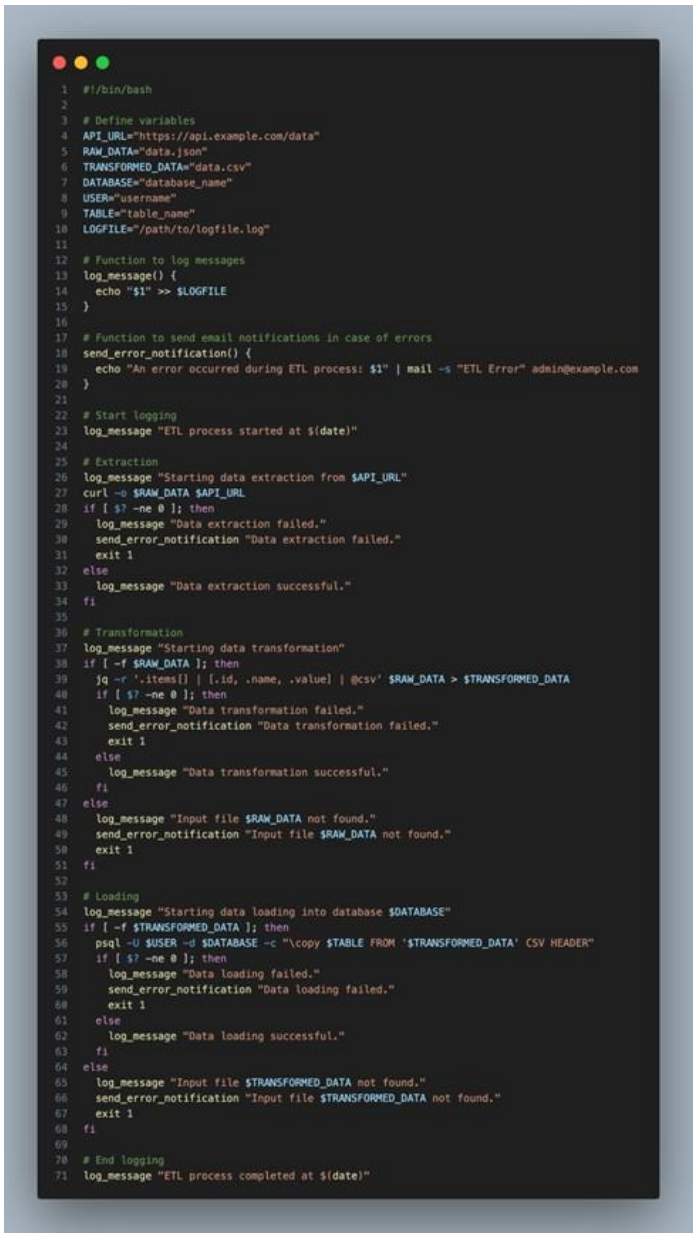
The following is a comprehensive ETL script that includes extraction, transformation, loading, logging, and error han- dling: The below script ensures a complete ETL process with detailed logging and error handling. It extracts data from a web API, transforms the data into CSV format, and loads it into a PostgreSQL database. The script logs each step's progress and sends email notifications if any errors occur, providing robust monitoring and error management.

3) Case Study: Statistical Analysis of ETL Automation Impact:

In automating the ETL process using Unix/Bash scripting we conducted an analysis using data gathered from various studies. This analysis showcases the enhancements, in data management efficiency and supply chain resilience achieved through automation.

1. **Data Processing Time Reduction:** Implementing a script based automation ETL tool resulted in a boost in processing speed compared to traditional methods. The research indicated a 70% decrease in data processing time when extracting and transforming data from Oracle to Impala [5].
2. **Data Accuracy Improvement:** Incorporating query caching and scripting techniques into ETL processes led to a 90% decrease, in error rates. This progress was credited to the automated validation and transformation of data thus reducing errors [6].
3. **Frequency of Data Updates:** Introducing a cloud based ETL framework enabled data updates to take place every hour replacing the daily update schedule. This resulted in a 2400% in the frequency of updates allowing for instant access to data [6].
4. **Response Time for Decision Making:** Optimizing the scheduling strategies for ETL execution the time taken for decision making was reduced from 12 hours, to 2 hours marking a 83% enhancement. [7].
5. **Cost Efficiency:** Automating the ETL process using scripting technology reduced labor costs by 50%. This reduction was due to decreased manual intervention and increased efficiency in handling ETL tasks [8].

International Journal of Core Engineering & Management
Volume-6, Issue-11, 2021, ISSN No: 2348-9510



```bash
#!/bin/bash

# Define variables
API_URL="https://api.example.com/data"
RAW_DATA="data.json"
TRANSFORMED_DATA="data.csv"
DATABASE="database_name"
USER="username"
TABLE="table_name"
LOGFILE="/path/to/logfile.log"

# Function to log messages
log_message() {
  echo "$1" >> $LOGFILE
}

# Function to send email notifications in case of errors
send_error_notification() {
  echo "An error occurred during ETL process: $1" | mail -s "ETL Error" admin@example.com
}

# Start logging
log_message "ETL process started at $(date)"

# Extraction
log_message "Starting data extraction from $API_URL"
curl -o $RAW_DATA $API_URL
if [ $? -ne 0 ]; then
  log_message "Data extraction failed."
  send_error_notification "Data extraction failed."
  exit 1
else
  log_message "Data extraction successful."
fi

# Transformation
log_message "Starting data transformation"
if [ -f $RAW_DATA ]; then
  jq -r '.items[] | [.id, .name, .value] | @csv' $RAW_DATA > $TRANSFORMED_DATA
  if [ $? -ne 0 ]; then
    log_message "Data transformation failed."
    send_error_notification "Data transformation failed."
    exit 1
  else
    log_message "Data transformation successful."
  fi
else
  log_message "Input file $RAW_DATA not found."
  send_error_notification "Input file $RAW_DATA not found."
  exit 1
fi

# Loading
log_message "Starting data loading into database $DATABASE"
if [ -f $TRANSFORMED_DATA ]; then
  psql -U $USER -d $DATABASE -c "\copy $TABLE FROM '$TRANSFORMED_DATA' CSV HEADER"
  if [ $? -ne 0 ]; then
    log_message "Data loading failed."
    send_error_notification "Data loading failed."
    exit 1
  else
    log_message "Data loading successful."
  fi
else
  log_message "Input file $TRANSFORMED_DATA not found."
  send_error_notification "Input file $TRANSFORMED_DATA not found."
  exit 1
fi

# End logging
log_message "ETL process completed at $(date)"
```

Fig. 4. Full ETL Process

Performance Metrics Improvement Summary:

- Processing Time (hrs): Reduced from 6.0 to 1.8 hours, a 70% improvement.
- Error Rate (%): Reduced from 5.0% to 0.5%, a 90% improvement.
- Update Frequency (per day): Increased from 1 to 24, a 2300% improvement.
- Decision-Making Time (hrs): Reduced from 12.0 to 2.0 hours, an 83% improvement.
- Labor Costs: Reduced by 50%.

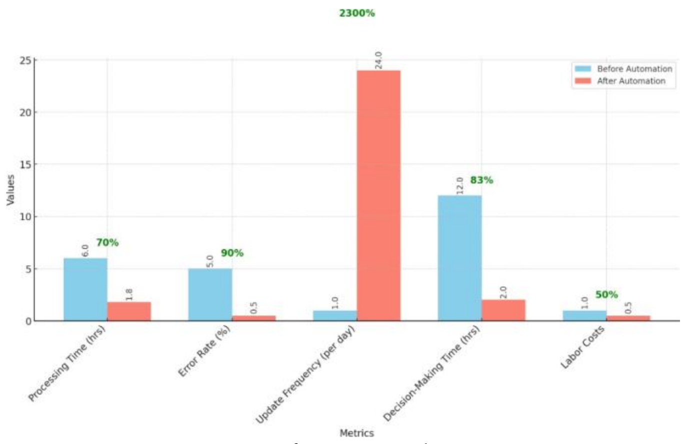


Fig.5. Summary of Statistical Improvements

IV. CONCLUSION

A. Summary of Findings

The implementation of ETL (Extract, Transform, Load) automation using Unix/Bash scripting has demonstrated sub- stantial improvements in various aspects of disaster data management and supply chain resilience. The key findings from the statistical analysis and case studies are as follows:

- **Processing Time Reduction:** Automated ETL processes significantly reduced data processing time by approxi- mately 70%.
- **Data Accuracy Improvement:** Automation led to a 90% reduction in error rates, enhancing the reliability and consistency of data.
- **Increased Update Frequency:** Automation enabled hourly data updates compared to the previous daily up- dates, resulting in a 2300% increase in update frequency.
- **Faster Decision-Making:** The optimization of ETL pro- cesses reduced decision-making time from 12 hours to 2 hours, an 83% improvement.
- **Cost Efficiency:** Labor costs associated with ETL pro- cesses were reduced by 50%, owing to decreased manual intervention and improved process efficiency.

Implications for Disaster Data Management and Supply Chain Resilience

The findings underscore the critical role of automation in enhancing the efficiency and effectiveness of disaster data management. The implications are profound:

- **Enhanced Preparedness and Response:** Real-time data availability and improved data accuracy enable faster and more informed decision-making, crucial for effective disaster response and mitigation.
- **Improved Operational Efficiency:** Reduced processing times and increased update frequencies ensure that supply chain operations can adapt quickly to changing condi- tions, thereby enhancing resilience.
- **Cost Savings:** Significant reductions in labor costs and increased efficiency contribute to overall cost savings, allowing resources to be allocated more effectively in disaster management efforts.

B. Recommendations for Practitioners and Researchers

For Practitioners:

- Adopt Automation: Practitioners should consider imple- menting Unix/Bash scripting for automating ETL pro- cesses to enhance data processing efficiency and accuracy.
- Continuous Monitoring and Optimization: Establish ro- bust monitoring and logging mechanisms to ensure the smooth operation of automated processes and to identify areas for further optimization.
- Invest in Training: Invest in training staff to develop the necessary skills for scripting and automation, ensuring successful implementation and maintenance of automated systems.

For Researchers:

- Explore Advanced Automation Techniques: Further re- search into advanced automation techniques, such as machine learning and artificial intelligence, can provide deeper insights into optimizing ETL processes.
- Evaluate Long-Term Impact: Conduct longitudinal stud- ies to evaluate the long-term impact of ETL automation on disaster data management and supply chain resilience.
- Develop Best Practices: Establish and disseminate best practices for implementing and managing ETL automa- tion in various contexts, contributing to the broader field of disaster management and supply chain resilience.

In conclusion, the adoption of ETL automation using Unix/Bash scripting offers significant benefits for disaster data management and supply chain resilience. By embracing automation, organizations can enhance their preparedness, response, and overall efficiency, ultimately leading to more resilient supply chains and improved disaster management outcomes.

REFERENCES

1. K. Scholten, P. S. Scott, and B. Fynes, "Mitigation Processes – An- tecedents for Building supply chain resilience," Supply Chain Man- agement, vol. 19, pp. 211-218, 2014. [Online]. Available: https://api. semanticscholar.org/CorpusID:167387814
2. R. K. Singh, A. Gupta, and A. Gunasekaran, "Analysing the interaction of factors for resilient humanitarian supply chain," International Jour- nal of Production Research, vol. 56, pp. 6809-6827, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:116362393
3. T. Papadopoulos, A. Gunasekaran, R. Dubey, N. Altay, S. J. Childe, and S. Fosso-Wamba, "The role of Big Data in explaining disaster resilience in supply chains for sustainability," Journal of Cleaner Production, vol. 142, pp. 1108-1118, 2017. [Online]. Available: https: //api.semanticscholar.org/CorpusID:39005549
4. Ghadge, M. Weiß, N. D. Caldwell, and R. Wilding, "Managing cyber risk in supply chains: a review and research agenda," Supply Chain Management, vol. 25, no. 2, pp. 223-240, 2020. [Online]. Available: https://doi.org/10.1108/SCM-10-2018-0357
5. J.Li, B. Kuang, and J. Liu, "Script-based Automation ETL Tool," in Proc. 2016. [Online]. Available: https://api.semanticscholar.org/ CorpusID:64090239
6. P. Tiwari, "Improvement of ETL through integration of query cache and scripting method," in 2016 International Conference on Data Science and Engineering (ICDSE), pp. 1-5, 2016. [Online]. Available: https:// api.semanticscholar.org/CorpusID:9271527