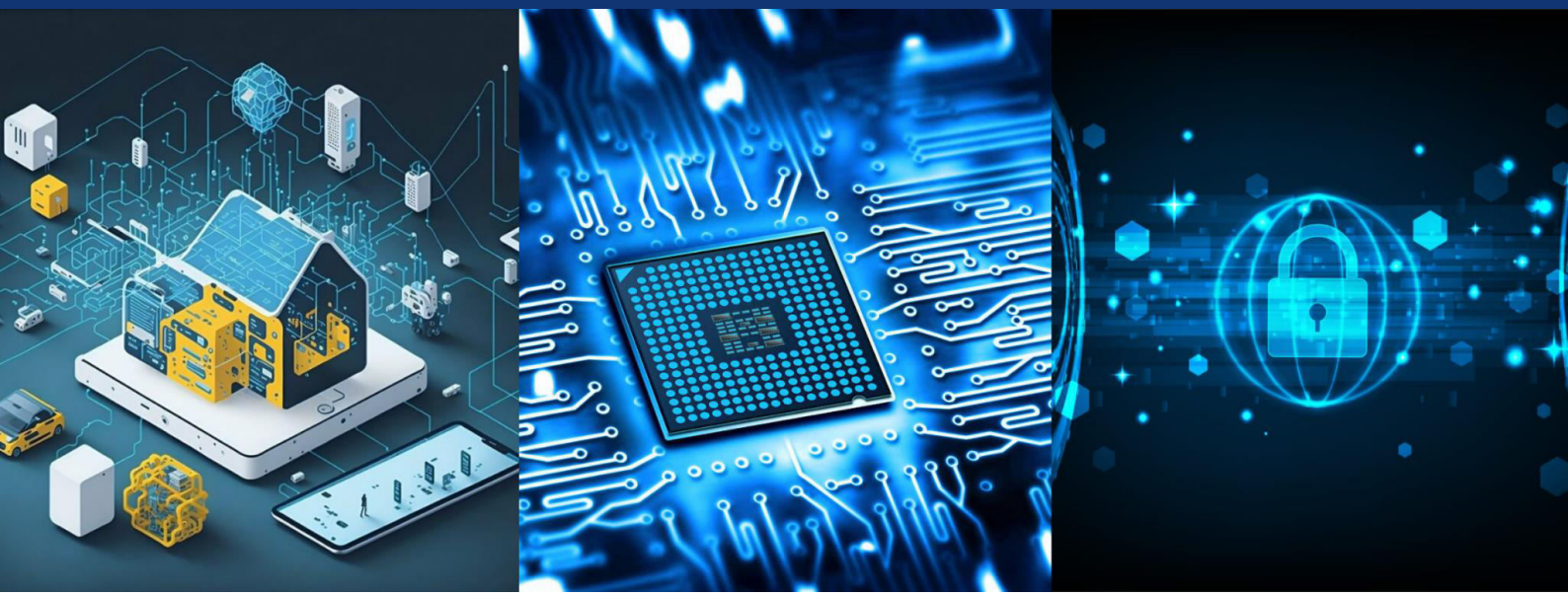
IJIRCCE

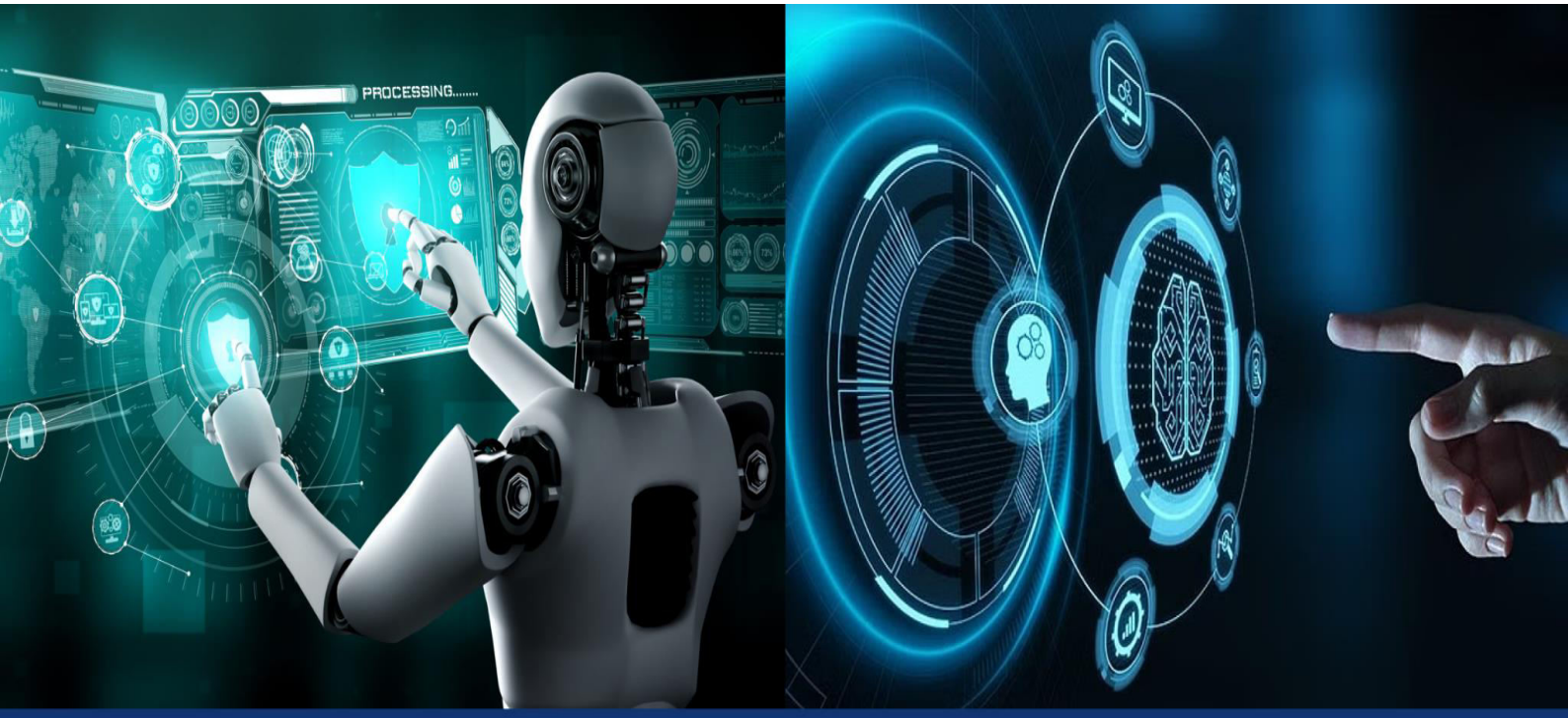
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 7, July 2026

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407049

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Predicate-Based RDF Indexing Using Hadoop MapReduce for Storage Optimization and Efficient Semantic Data Retrieval

Pintukumari Bera[1], Dr. Brij Bihari Dubey[2], Dr. Ashutosh Abhangi[3]

Research Scholar, ITMVU, Baroda, India[1]

Research Guide, ITMVU, Baroda, India [2]

Mentor, ITMSLS, Baroda, India [3]

ABSTRACT: Resource Description Framework (RDF) is widely used for Semantic Web applications and graph-based data management [1][10]. Traditional RDF storage generates redundancy due to repeated predicate representation, increasing storage requirements and affecting retrieval performance. This paper proposes a Predicate-Based RDF Indexing approach using Hadoop MapReduce to optimize RDF storage and improve retrieval efficiency [2][3]. The movie dataset was transformed into RDF format and processed using Hadoop Streaming with Python-based mapper and reducer implementations. Experimental results demonstrate a reduction in RDF storage from 7.09 MB to 5.02 MB, achieving approximately 29.26% storage optimization while maintaining accessibility and scalability.

KEYWORDS: RDF, Hadoop MapReduce, Predicate Indexing, Semantic Web, Storage Optimization

I. INTRODUCTION

The rapid growth of large-scale web data and Semantic Web applications has increased the need for efficient data representation and storage techniques. Resource Description Framework (RDF) is one of the most widely used models for representing semantic information in the form of subject-predicate-object triples [1][16]. RDF provides flexibility and interoperability for semantic applications; however, it generates large amounts of redundant data because predicates are repeatedly stored for multiple entities.

Traditional RDF storage methods may lead to increased storage requirements and reduced retrieval efficiency when handling large datasets [5]. Efficient indexing mechanisms are therefore necessary to improve RDF storage management and query performance [6].

Distributed computing frameworks such as Hadoop MapReduce provide scalability and parallel processing capabilities for handling large-scale datasets [2][3]. This research proposes a Predicate-Based RDF Indexing technique using Hadoop MapReduce for improving RDF organization and storage optimization.

II. LITERATURE SURVEY

Several researchers have investigated RDF storage optimization techniques and semantic data processing methods. Dean and Ghemawat introduced the MapReduce programming model for distributed processing of large datasets [2]. White discussed Hadoop architecture and distributed processing mechanisms for big data applications [3].

Weiss et al. proposed Hexastore indexing for RDF storage and demonstrated improved retrieval performance using semantic indexing techniques [5]. Wilkinson et al. developed Jena2 for efficient RDF storage and retrieval [6]. Harth and Decker presented optimized indexing structures for RDF storage and query processing to improve retrieval efficiency [13].

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407049

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

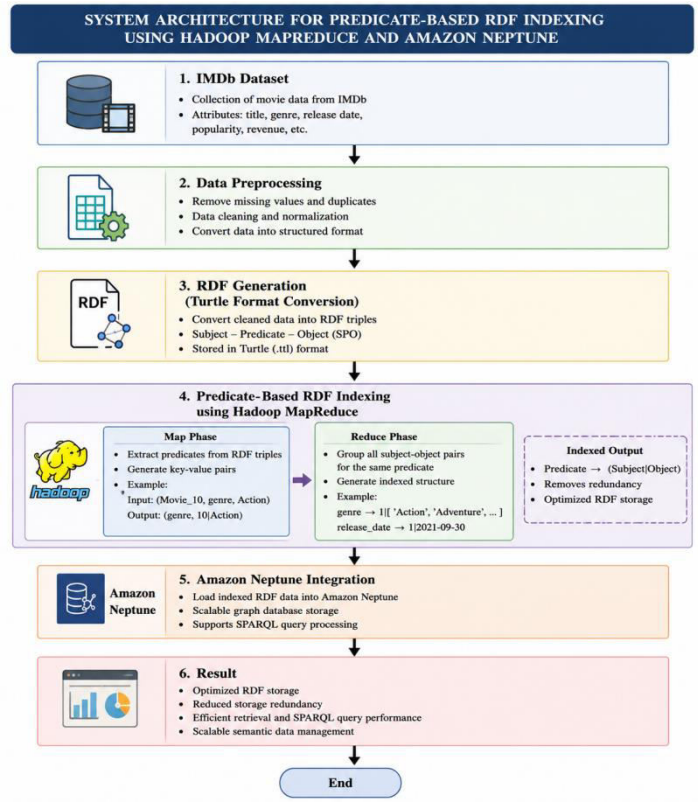
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Berners-Lee et al. introduced the Semantic Web framework and discussed semantic interoperability among heterogeneous systems [10]. Cyganiak et al. further explained RDF concepts and data representation models used in semantic systems [16].

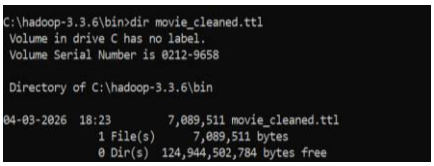
Urbani et al. proposed scalable distributed reasoning techniques using MapReduce for semantic data processing [19]. Zou et al. introduced graph-based query processing approaches for RDF datasets and SPARQL execution [20]. Recent graph database systems such as Amazon Neptune support RDF storage and SPARQL query execution [8]. The literature indicates that RDF optimization techniques focusing on indexing mechanisms can improve data organization, scalability, query efficiency, and distributed semantic data management [13][21][23].

III. SYSTEM ARCHITECTURE



IV. METHODOLOGY / APPROACH

The proposed methodology consists of multiple processing stages:
Step 1: RDF Dataset Generation
The movie dataset was cleaned and transformed into RDF format using Turtle (.ttl) representation [1][16].
Dataset:
movie_cleaned.ttl



Step 2: Mapper Phase
The mapper extracts RDF predicates and generates intermediate key-value pairs.

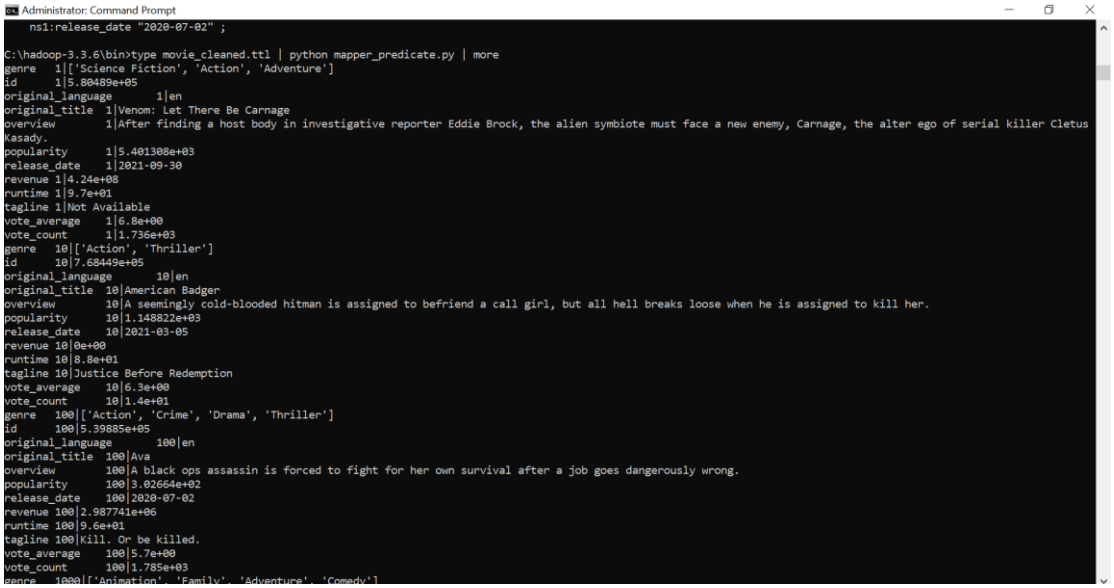
© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407049

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Input Triple:
(Movie_10, genre, Action)
Mapper Output:
(genre,10|Action)



Step 3: Reducer Phase
The reducer groups all subject-object pairs associated with the same predicate.
Reducer Output:
genre → 1|[‘Science Fiction’,‘Action’,‘Adventure’]
release_date → 1|2021-09-30
original_title → 1|Venom: Let There Be Carnage
Step 4: Predicate Index Generation
Final Structure:
Predicate → (Subject|Object)

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407049

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Step 5: Amazon Neptune Integration

The optimized indexed RDF output can further be integrated into Amazon Neptune for scalable graph storage and SPARQL query processing [8].

V. RESULTS & DISCUSSION

The proposed system was implemented using Hadoop Streaming with Python-based mapper and reducer components [9][21].

Experimental Environment

Operating System: Windows

Hadoop Version: 3.3.6

Programming Language: Python

Processing Framework: Hadoop Streaming

RDF Format: Turtle (.ttl)

Dataset Name:

movie_cleaned.ttl

Experimental Results

Map Progress: 100%

Reduce Progress: 100%

Job Status: Successfully Completed

Generated Output File:

predicate_index.txt

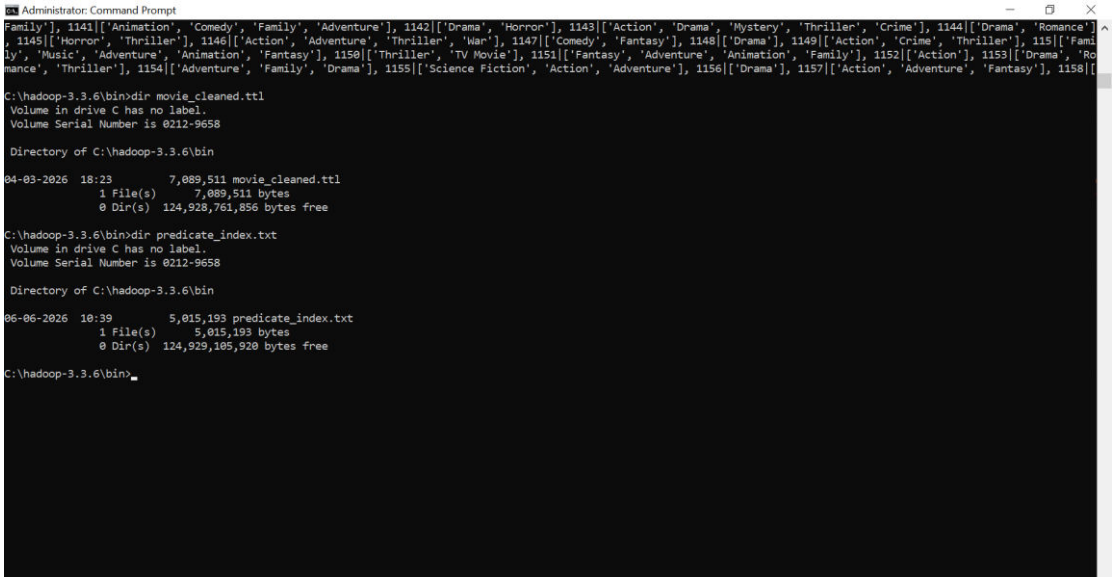
Sample Generated Output:

genre → 1|['Science Fiction','Action','Adventure']

release_date → 1|2021-09-30

original_title → 1|Venom: Let There Be Carnage

Storage Optimization Analysis



Parameter	Value
Original RDF Size	7.09 MB
Indexed Output Size	5.02 MB
Storage Saved	2.07 MB
Percentage Reduction	29.26%

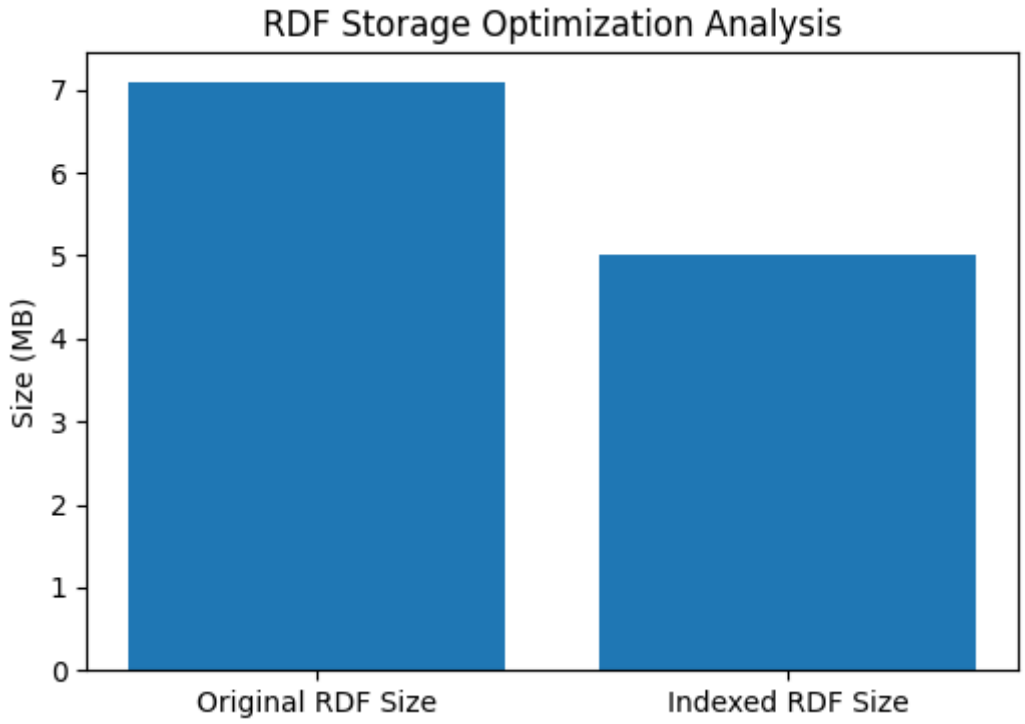
RDF Storage Optimization Analysis

© 2026 IJIRCCE | Volume 14, Issue 7, July 2026 DOI: 10.15680/IJIRCCE.2026.1407049

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Analysis:
The graph illustrates the comparison between the original RDF dataset size and the predicate-indexed RDF output size. The original RDF dataset occupied 7.09 MB, while the optimized indexed structure required only 5.02 MB. The proposed Predicate-Based RDF Indexing technique achieved approximately 29.26% storage optimization, indicating reduced redundancy and improved storage efficiency. The obtained results demonstrate that the proposed approach effectively organizes RDF entities and supports scalable semantic data management[8].

VI. CONCLUSION

This paper presented a Predicate-Based RDF Indexing technique using Hadoop MapReduce for RDF storage optimization and efficient semantic data management. The proposed approach grouped RDF entities according to predicates and generated an indexed structure that reduced repeated RDF storage. Experimental results demonstrated a reduction in storage requirements from 7.09 MB to 5.02 MB, achieving approximately 29.26% storage optimization. The obtained results indicate improved RDF organization, reduced redundancy, and enhanced storage efficiency. Furthermore, the proposed framework can support scalable semantic retrieval and graph-based processing through Amazon Neptune integration [8].

VII. FUTURE WORK

Future enhancements may include implementation of advanced RDF indexing techniques to further improve storage optimization and query performance. The proposed framework can also be extended for real-time RDF processing, larger datasets, and complete integration with Amazon Neptune for scalable semantic graph analysis.

REFERENCES

[1] G. Klyne and J. Carroll, “Resource Description Framework (RDF): Concepts and Abstract Syntax,” W3C Recommendation, 2004.
[2] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” Communications of the ACM, 2008.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[3] T. White, Hadoop: The Definitive Guide, O'Reilly Media, 2015.
[4] A. Harth et al., "YARS2: A Federated Repository for Querying Graph Structured Data from the Web," ISWC, 2007.
[5] C. Weiss et al., "Hexastore: Sextuple Indexing for Semantic Web Data Management," VLDB, 2008.
[6] K. Wilkinson et al., "Efficient RDF Storage and Retrieval in Jena2," SWDB Workshop, 2003.
[7] A. Hogan et al., "Scalable and Distributed Methods for Semantic Web Data," ACM Computing Surveys, 2021.
[8] Amazon Web Services, "Amazon Neptune Graph Database," Available: https://aws.amazon.com/neptune/, Accessed: June 2026.
[9] Apache Software Foundation, "Apache Hadoop Documentation."
[10] T. Berners-Lee et al., "The Semantic Web," Scientific American, 2001.
[11] D. Wood et al., "SPARQL Query Language for RDF," W3C Recommendation.
[12] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language," W3C Recommendation, 2013.
[13] A. Harth and S. Decker, "Optimized Index Structures for RDF Data Storage and Retrieval," Semantic Web Journal.
[14] M. Stonebraker and U. Çetintemel, "One Size Fits All: An Idea Whose Time Has Come and Gone," ICDE, 2005.
[15] L. Ding et al., "Swoogle: A Search and Metadata Engine for the Semantic Web," ACM CIKM.
[16] R. Cyganiak, D. Wood, and M. Lanthaler, "RDF 1.1 Concepts and Abstract Syntax," W3C Recommendation, 2014.
[17] S. Harris and N. Gibbins, "3store: Efficient Bulk RDF Storage," Proceedings of PSSS, 2003.
[18] K. Wilkinson, C. Sayers, H. Kuno, and D. Reynolds, "Efficient RDF Storage and Retrieval in Jena2," SWDB Workshop, 2003.
[19] J. Urbani, S. Kotoulas, E. Oren, and F. van Harmelen, "Scalable Distributed Reasoning using MapReduce," ISWC, 2009.
[20] L. Zou, M. Özsu, L. Chen, X. Shen, R. Huang, and D. Zhao, "gStore: A Graph-Based SPARQL Query Engine," VLDB Journal, 2014.
[21] D. Abadi, A. Marcus, S. Madden, and K. Hollenbach, "Scalable Semantic Web Data Management using Vertical Partitioning," VLDB, 2007.
[22] J. Broekstra, A. Kampman, and F. van Harmelen, "Sesame: A Generic Architecture for RDF Storage and Querying," ISWC, 2002.
[23] M. Schmidt, M. Meier, and G. Lausen, "Foundations of SPARQL Query Optimization," ICDT, 2010.
[24] Apache Jena Documentation, "Apache Jena Framework for RDF Applications."
[25] Oracle Corporation, "Big Data and Semantic Technologies Overview."

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details