Chapter 3
A Survey on JSON Data Stores

Lubna Irshad
Nanjing University of Aeronautics and Astronautics, China

Zongmin Ma
https://orcid.org/0000-0001-7780-6473
Nanjing University of Aeronautics and Astronautics, China

Li Yan
Nanjing University of Aeronautics and Astronautics, China

ABSTRACT

The World Wide Web is a bunch of interlinked documents full of information and knowledge where people search sources to interpret relevant information. As the modern web is becoming more interactive and data-centric, it is necessary to focus on how to exchange data efficiently, easily, and quickly. There are several ways to exchange data among web services like XML and JSON. JSON is one of its kind of emerging data-interchange format. Its data model maps the type system of many programming languages is recommended for web development, especially Java scripting language. Previously, JSON was the focus by NoSQL community, but presently, the relational database community also claims to manage both relational and JSON data in one platform efficiently. This chapter focuses on various approaches to use JSON stores with NoSQL and relational database systems. The study reveals work accomplished so far to identify the working of JSON Stores as NoSQL stores and role relational system plays to enhance the power of JSON stores.

INTRODUCTION

The formal international standard for JSON is RFC-4627 (Request for Comments). Douglas Crockford, the discoverer of the JSON (JavaScript Object Notation) specification, describes JSON as "a lightweight data-interchange format" (Introducing JSON, n.d.). As a part of well-defined ECMA (European Computer Manufacturers Association)-262 that is widely followed, makes it flexible to use. It consists of attributes (name/value) pairs, list of values and objects that is easy to understand, develop and integrate. Initially JSON was the focus of the NoSQL (Not only SQL) community providing native JSON Stores like

DOI: 10.4018/978-1-5225-8446-9.ch003

MongoDB (MongoDB, n.d.), CouchDB (Apache CouchDB, n.d.) and many more (Beyer et al., 2005). However, due to its swift acceptance by Web developers, it has gained attention in RDBMS (Relational Database Management System) to embed it as a part. JSON's "No-schema" nature helps the developer to start working with data in no time without the hassle of schema design (Liu, Hammerschmidt, & McMahon, 2016). It is self-describing and has independent text format that exists in almost every significant programming language (Introducing JSON, n.d.). JSON is a format that fills a particular niche to integrate multiple services across many platforms and require no prior knowledge. There are several ways to exchange data among Web Services; JSON is one of its kind of emerging data-interchange format as compared to XML and many other NoSQL document stores.

XML (Extensible Markup Language) and JSON both are hierarchical, human readable and have the support of many programming languages. JSON has replaced XML for data serialization (Yan Betts, 2014) as XML is heavy, more complex and requires more bytes for transfer even for a smaller task. XML required parser while JavaScript function can parse JSON easily without any additional parser. Format of JSON is concise as compared to XML that has lengthy tags and namespaces (Haq, Khan, & Hussain, 2015). JSON is semi-structured hierarchal NoSQL data model as described in Figure 1. It is among one of the most popularly used categories of NoSQL Database System.

NoSQL database is an alternative approach as compared to traditional Relational Database Management Systems (RDBMS). In traditional RDBMS, it is mandatory to design schema carefully before storing/adding data in the tables and creating relations (Liu, Hammerschmidt, & McMahon, 2016). NoSQL databases is a technology that helps to design a database that encompasses a variety of data models like document databases, key-value pair stores, columnar stores and graph formats stores (Chandra, 2015). NoSQL databases deal with a large volume of structured, semi-structured, and unstructured data that is scalable and vary from time to time (MongoDB, n.d.). Data can be validated using validation rules (MongoDB, n.d.). In today's development where time to market is crucial, agile development is gaining much popularity and NoSQL databases are fulfilling these challenges very well as compared to RDBMS.

RDBMS is a packet full of features that include native query language, efficient query processing techniques, indexes for quicker data retrieval, virtual tables to store and query sensitive data (views), triggers for predefined actions, multilevel user access control permissions, data recoveries in case of disaster, ACID (Atomicity, Consistency, Isolation, Durability) safety guaranteed transactions and many more. RDBMS provides immediate data access, stiff data consistency and quick data retrieval for transactional enterprise applications but in many applications quick developments, scalability prevails over everything.

JSON Document Stores besides pulses have some minuses too as compared to relational databases. Major limitations of these NoSQL Document Stores are lacking powerful native query like SQL, missing ACID transaction capability and challenging for data analysis/processing (Chandra, 2015). This trend leads to increase complexity of the management of data with polyglot persistence (Fowler & Sadalage, 2012). On one hand, by using JSON developers gains the benefits of reducing development overhead, costs and improves the quality of service. On the other hand, loses the power of rich native query processing constructs, indexes for quicker data retrieval, safety guaranteed transactions and many more.

This paper focuses on JSON document Storage and up-to-date review of various approaches to store JSON document in NoSQL and Relational Database System. In addition, we have discussed the advantages of using JSON in NoSQL and Relational Databases.

The main theme of this paper has three folds. The first section discusses the background that apprehends JSON data model, its comparison with XML, JSON schema that describes and validate JON document, support of JSON in NoSQL databases and relational databases along with their structural differences. It