ՇԻՐԱԿԻ Մ. ՆԱԼԲԱՆԴՅԱՆԻ ԱՆՎԱՆ ՊԵՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ
ШИРАКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М. НАЛБАНДЯНА
SHIRAK STATE UNIVERSITY AFTER M. NALBANDYAN
УЧЕНЫЕ ЗАПИСКИ ԳԻՏԱԿԱՆ ՏԵՂԵԿԱԳԻՐ SCIENTIFIC PROCEEDINGS

Պրակ Ա
Выпуск А
Issue A

2016 № 1

INFORMATION TECHNOLOGIES

УДК 004

T. A. Shahinyan

STORAGE AND PROCESSING OF LARGE SCALE RDF DATA

Keywords: parallel computing, databases, big data, semantic web.

Բանալի բառեր՝ զուգահեռ ծրագրավորում, տվյալների հենքեր, մեծ տվյալներ, սեմանտիկ վեբ:

Ключевые слова: параллельное программирование, базы данных, большие данные, семантик веб.

Storage and processing of huge amounts of data is one of major challenges during last years. Parallel relational and nonrelational database systems are usually used for these tasks. One of the growing sources of data is semantic web data in RDF representation. In this paper we discuss approaches for efficient storage and retrieval of RDF data using modern parallel data processing technologies.

1. Introduction

One of the challenges brought by the development of information technologies is the processing of huge amounts of data. The necessity for their efficient storage processing using the computing power of the Internet, computer grids and cloud system caused the emergence of non-relational data processing paradigms and technologies. MapReduce is a paradigm introduced by Google and implemented by several vendors [1]. One of the most widespread implementations of the MapReduce paradigm is the open source Apache Hadoop [2].

The Semantic Web is an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The most important technologies for making Semantic Web possible are Resource Description Framework(RDF), in which meaning is expressed in sets of subject-predicate-object triples, Universal Resource Identifier (URI) is used for identifying resources, and XML is used as one of possible ways of document serialization and exchange[3].

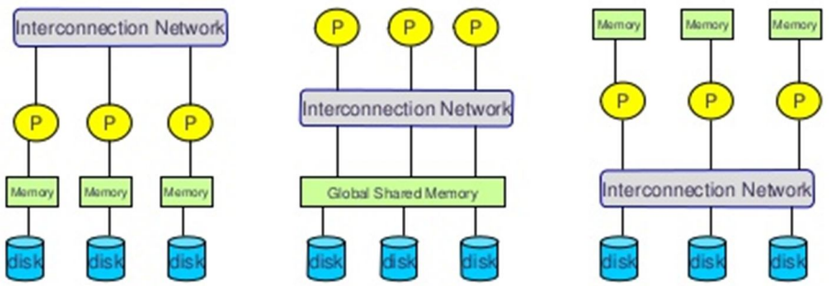
Our goal is the efficient usage of both technologies and approaches to achieve knowledge storage and processing in modern heterogenous computing environments.

2. DISTRIBUTED DATA PROCESSING APPROACHES

There are two main approaches for distributed data storage and processing. Parallel relational database systems seek to improve performance through parallelization of various operations, such as loading data, building indices and evaluating queries(involving selection, projection, aggregation and join operations).

Parallel databases are based on one of the following architectures:

- Shared memory architecture, where multiple processors share the main memory space, as well as mass storage (e.g. hard disk drives).
- Shared disk architecture, where each node has its own main memory, but all nodes share mass storage, usually a storage area network. In practice, each node usually also has multiple processors.
- Shared nothing architecture, where each node has its own mass storage as well as main memory [4].



Shared nothing architecture is now most often chosen due to the availability of cheaper and easier to maintain hardware of computer clusters, grids and clouds.

Although data may be stored in a distributed fashion, the distribution is governed solely by performance considerations. Parallel databases improve processing and input/output speeds by using multiple CPUs and disks in parallel. Centralized and client–server database systems are not powerful enough to handle many modern applications. In parallel processing many operations are performed simultaneously, as opposed to serial processing, in which the computational steps are performed sequentially.

Parallel databases are database systems that are implemented on parallel computing platforms. They are mainly focused on high performance query processing, including database queries and transactions that make use of parallelism techniques applied to an underlying parallel computing platform in order to achieve high performance [5].

There are two measures of parallel database performance:

$$\text{Speed up} = \frac{\text{elapsed time on a single processor}}{\text{elapsed time on multiple processors}},$$

$$\text{Scale up} = \frac{\text{volume of processed data in a time interval on a single processor}}{\text{volume of processed data in the same time interval on multiple processors}}$$

The volume of processed data may be either the number of transactions or the number of records in the database.

Scalability issues have lead to non-relational approaches of data modeling.

XML databases are an example of semi-structured databases. They use XML documents to store data and XPath, XQuery and other query languages to retrieve data. There are several commercial and open-source XML databases used in different areas.

AMGA is a distributed metadata catalog service in grid environment developed by EGEE user community.

Catalog Services play a vital role on Data Grids by allowing users and applications to discover and locate the data needed. On large Data Grids, with hundreds of geographically distributed sites, centralized Catalog Services do not provide the required scalability, performance or fault-tolerance. AMGA is based on gLite software stack [6].

MapReduce is a paradigm introduced by Google for processing huge datasets on certain kinds of distributable problems using a large number of computers (nodes), a cluster or a grid. It is based on two functions: Mapper and Reducer, which are used as arguments by higher-order functions Map and Reduce [1].

- On the "Map" step the master node takes the input, partitions it up into smaller sub-problems, and distributes those to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes that smaller problem, and passes the answer back to its master node.
- On the "Reduce" step the master node then takes the answers to all the sub-problems and combines them in some way to get the output – the answer to the problem it was originally trying to solve.

The advantage of MapReduce is that it allows for distributed processing of the map and reduction operations. Provided each mapping operation is independent of the others, all maps can be performed in parallel – though in practice it is limited by the data source and/or the number of CPUs near that data. Similarly, a set of 'reducers' can perform the reduction phase – all that is required is that all outputs of the map operation which share the same key are presented to the same reducer, at the same time. While this process can often appear inefficient compared to algorithms that are more sequential, MapReduce can be applied to significantly larger datasets than "commodity" servers can handle – a large server farm can use MapReduce to sort a petabyte of data in only a few hours. The parallelism also offers some possibility of recovering from partial failure of servers or storage during the operation: if one mapper or reducer fails, the work can be rescheduled – assuming the input data is still available.

The key feature of MapReduce is Scalability. That’s why we’ve chosen it as an alternative to relational database approach.

The most important drawbacks of the MapReduce paradigm is lack of indexes and query optimization.

We’ve chosen Hadoop as an implementation of MapReduce paradigm since it is one of the most widely used and supported open-source implementation of MapReduce[6].

Parallel relational database systems show high performance especially in homogeneous environments [7], whereas the most important advantages of MapReduce are fault tolerance and the ability to operate in heterogeneous environments [8].

3. DATA REPRESENTATION IN SEMANTIC WEB

The Resource Description Format (RDF) and OWL (Web Ontology Language) are used to represent information resources modeled as a directed labeled graphs, where edges represent the named link between two resources, represented by the graph nodes [9]. RDF uses URI references to identify resources and properties.

RDF graphs can be read and written by using the Jena software package, and queried using the SPARQL query language [10].

Semantic web is penetrating many areas of information processing and exchange activity. An example is UniProt, and effort to create a comprehensive catalog of protein data in RDF [11]. With queries written in SPARQL query language the UniProt knowledge base allows to retrieve any information about proteins, for example the following query will return the preferred gene name and disease annotation of all human entries that are known to be involved in a disease:

```
SELECT ?name ?text
WHERE
{       ?protein a up:Protein .
        ?protein up:organism taxon:9606 .
        ?protein up:encodedBy ?gene .
        ?gene skos:prefLabel ?name .
        ?protein up:annotation ?annotation .
        ?annotation a up:Disease_Annotation .
        ?annotation rdfs:comment ?text}
```

4. SEMANTIC WEB DATA PROCESSING

We have used Hadoop as an implementation of MapReduce to process protein data from UniProt catalog. The data was stored in HDFS distributed file system in line-based NTriples. For processing the data we have used Jena with its Elephas library, which provides Hadoop InputFormat and OutputFormat implementations for RDF[12]. It covers all RDF serializations that Jena supports and extensions by custom formats.

Elephas splits and parallelizes processing of input where the RDF serialization allows it.

We have used and extended various reusable basic Mapper and Reducer implementations covering the following common tasks: counting, filtering, grouping, splitting, transformation.

For example, filtering tasks allow rewriting SPARQL queries similar to the one above to filter resources satisfying given criteria by sequentially applying filters on different predicates of resources, to retrieve needed protein records. Using Elephas on large distributed networks show significant improvement of processing efficiency.

The future work will be development of automatic convertor of SPARQL queries into MapReduce tasks to be executed on Jena Elephas.

Տ. Ա. Շահինյան

Լայնածավալ RDF տվյալների պահպանում և մշակում

Հսկայական ծավալի տվյալների պահպանումը և մշակումը վերջին տարիների ընթացքում դարձել են կարևոր մարտահրավերներից մեկը: Զուգահեռ ռելացիոն և ոչ ռելացիոն տվյալների հենքերը սովորաբար օգտագործվում են այս խնդիրների լուծման համար: Տվյալների աճող աղբյուրներից են նաև սեմանտիկ վեբի RDF ներկայացմամբ տվյալները: Հոդվածում քննարկվում են RDF տվյալների արդյունավետ պահպանման և մշակման մոտեցումներ՝ հիմնված տվյալների զուգահեռ մշակման տեխնոլոգիաների վրա:

Т. А. Шагинян

Хранение и обработка крупномасштабных RDF данных

Хранение и обработка больших объемов данных является одним из основных проблем в последнее время. Параллельные реляционные и нереляционные базы данных обычно используются для решения этих задач. Одним из растущих источников данных являются данные семантик веб в представлении RDF. В этой статье обсуждаются подходы эффективного хранения и обработки RDF данных с использованием современных технологий параллельной обработки данных.

References

1. Jeffrey Dean, Sanjay Ghemawat, MapReduce: A Flexible Data Processing Tool, Communications of the ACM, Volume 53 Issue 1, January 2010.
2. Apache Hadoop, https://hadoop.apache.org/.
3. The Semantic Web, Berners-Lee, Tim; James Hendler; Ora Lassila Scientific American Magazine, May, 2001.
4. David DeWitt, Jim Gray, Parallel database systems: the future of high performance database systems, Communications of the ACM, Volume 35 Issue 6, June 1992.
5. David Taniar, Clement H.C. Leung, Wenny Rahayu, Sushant Goel, High Performance Parallel Database Processing and Grid Databases, John Wiley & Sons, 2008.
6. Nuno Santos, Birger Koblitz, Distributed Metadata with the AMGA Metadata Catalog, Workshop on Next-Generation Distributed Data Management, HPDC Conference, 2006.

7. A. Pavlo, A. Rasin, S. Madden, M. Stonebraker, D. DeWitt, E. Paulson, L. Shrinivas, and D. J. Abadi. A Comparison of Approaches to Large Scale Data Analysis. In Proc. Of SIGMOD, 2009.
8. A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi,A. Silberschatz, A. Rasin, HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads, VLDB 2009.
9. W3C Semantic Web Acivity, http://www.w3.org/2001/sw/ .
10. Michael Grobe, RDF, Jena, SparQL and the "Semantic Web" SIGUCCS'09, October 11–14, 2009, St. Louis, Missouri, USA.
11. Universal Protein Resource. http://www.uniprot.org/.
12. Apache Jena Elephas Documentation Page, https://jena.apache.org/documentation/hadoop/.

About the author

Tigran Shahinyan *- PhD of Technical Sciences, Institute for Informatics and Automation Problems, NAS RA, E-mail: tigran.shahinyan@gmail.com*

Received 15. 09. 2016.