A Proposal of a Big Web Data Application and Archive for the Distributed Data Processing with Apache Hadoop

Martin Lnenicka, Jan Hovad, and Jitka Komarkova(✉)

University of Pardubice, Faculty of Economics and Administration, Pardubice, Czech Republic
martin.lnenicka@gmail.com, {jan.hovad,jitka.komarkova}@upce.cz

Abstract. In recent years, research on big data, data storage and other topics that represent innovations in the analytics field has become very popular. This paper describes a proposal of a big web data application and archive for the distributed data processing with Apache Hadoop, including the framework with selected methods, which can be used with this platform. It proposes a workflow to create a web content mining application and a big data archive, which uses modern technologies like Python, PHP, JavaScript, MySQL and cloud services. It also shows the overview about the architecture, methods and data structures used in the context of web mining, distributed processing and big data analytics.

Keywords: Big web data · Big data analytics · Web content mining · Distributed data processing · Python · Apache Hadoop

1 Introduction

During the past few years there has been a growing interest in the term big data, fostered by the broad availability of various data acquisition, storing and processing platforms. The concepts of linked and open data have led to the necessity to process these large amounts of data very quickly to retrieve valuable information. Types of data frequently associated with the big data analytics include the web data (e.g. web log data, e-commerce logs and social network interaction data), the industry specific transaction data (e.g. telecommunications call data records, geolocation data and retail transaction data), the machine generated / sensor data and the text (e.g. archived documents, external content sources or customer and supplier data). These data are not only needed to be stored, but also be prepared to provide tools for the support of the decision-making process. Big data applications may change the way of mining, storing, processing and analyzing of websites content. Big data applications do not replace traditional applications, but complement them. Businesses as well as public sector can mine this big web data to provide insight to support future decisions.

Thus, authors focus on design and coding a web content mining application, which provides the basic layer to build a big web data archive and then propose a framework for the processing of these data with the Apache Hadoop cluster.

M. Núñez et al. (Eds.): ICCCI 2015, Part II, LNCS 9330, pp. 285–294, 2015.
DOI: 10.1007/978-3-319-24306-1_28

2 Related Work and Background

The growth of data sources and the ease of access that information technology affords also bring new challenges on data acquisition, storage, management and analysis. The most recent survey about the term big data is well conducted in [1], where authors present the general background of big data and review related technologies, such as distributed approach, Internet of Things, data centers, and Apache Hadoop. They also introduce the terms big data generation and acquisition, big data storage and big data applications including web data analysis, which are closely related to the topic of this paper. Tien [2] then compares major differences between the big data approach and the traditional data management approach with four components (acquisition, access, analytics, application) and three elements (focus, emphasis, scope).

The work by Dean and Ghemawat [3] describes the file system implemented by Google called the Google File System (GFS), which handles the big data operations behind the Google services and it is the main part of the MapReduce framework. Also the findings of Vilas [4] lend support to the claim that high performance computing platforms are required, which impose systematic designs to unleash the full power of big data. A comparison of approaches to large-scale data analysis can be found e.g. in Pavlo et al. [5] or also Chen et al. [1]. The evidence supporting the use of big data for analytics and the improvement of the decision-making process may lie in the findings of Power [6], who proposes the use cases and user examples related to analyzing large volumes of semistructured and unstructured data.

There are three types of mining: data mining, web mining and text mining. Web mining lies in between data mining and text mining, and copes mostly with semi-structured data and/or unstructured data. It can be distinguished in the three different types of categories: web content mining, web usage mining and web structure mining. Web content mining is performed by extracting useful information from the content of a web page. It includes extraction of structured data from web pages, identification, match, and integration of semantically similar data, opinion extraction from online sources, and concept hierarchy or knowledge integration [7].

3 Problem Statement and Used Tools

The primary objective of this paper is to build a big web data archive and introduce a framework for the distributed processing of these data with Apache Hadoop. For the purpose of this objective the web content mining application is proposed.

The partial objectives are intended to meet the following objectives. The first one is to compose a suitable architecture and choose the right web technologies. Save and clear strings of each source by regular expressions. Transform the HTML structure into the easy accessible objects. Implement the processing phase by Python and in the case of the big web data, by the distributed approach represented by Apache Hadoop. Save necessary history data in a simple MySQL database and utilize the replicated commercial cloud storage Dropbox. Create a framework for the analysis with Apache Hadoop over the obtained big web data.

These points form the outline of this paper and they are shown in the following scheme, which is represented by the Fig. 1.

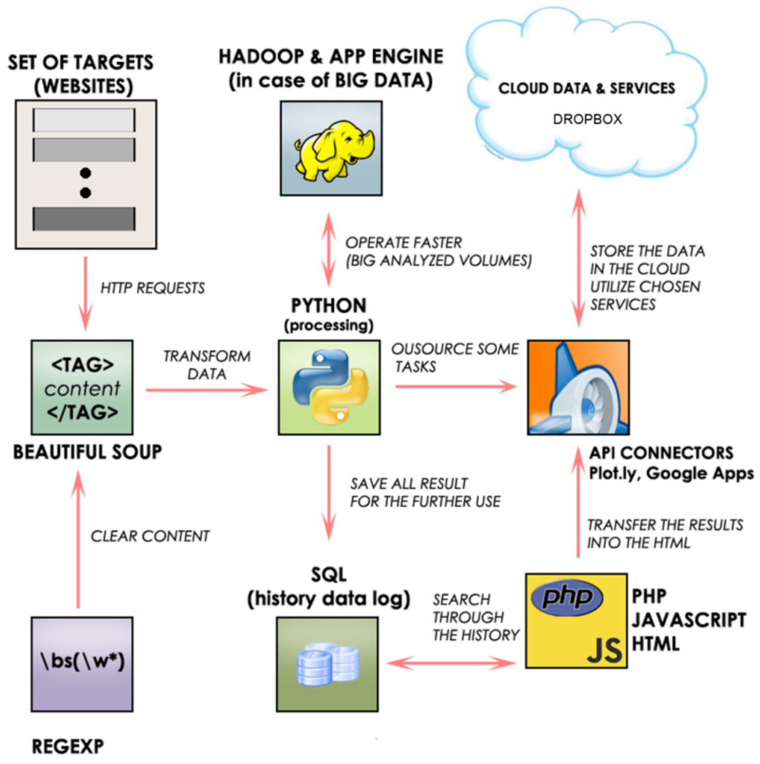
SET OF TARGETS (WEBSITES)
HADOOP & APP ENGINE (in case of BIG DATA)
CLOUD DATA & SERVICES
DROPBOX
OPERATE FASTER (BIG ANALYZED VOLUMES)
STORE THE DATA IN THE CLOUD UTILIZE CHOSEN SERVICES
HTTP REQUESTS
PYTHON (processing)
<TAG> content </TAG>
TRANSFORM DATA
OUSOURCE SOME TASKS
BEAUTIFUL SOUP
API CONNECTORS Plot.ly, Google Apps
SAVE ALL RESULT FOR THE FURTHER USE
CLEAR CONTENT
TRANSFER THE RESULTS INTO THE HTML
SQL (history data log)
SEARCH THROUGH THE HISTORY
php
JS
PHP JAVASCRIPT HTML
\bs(\w*)
REGEXP

Fig. 1. The solution scheme of this paper

The main tool used is Python in the version 2.7, because the version 3.3 has limited compatibility with some web services e.g. Google. Processing the HTML data (website raw string) response to the sent request is done by parsing the output tree, which is described by Jackson [8]. Uniform data structure as a XML is replaced by a JSON, which is more suitable for a Python and it is also used by many API's as a data structure to transfer the optional adjustment arguments [9]. An API creation is covered in the work of Henning [10]. The MySQLdb connector is used to save the big web data during the script runs. The beautiful soup (BS) library is used for accessing the data. The processing tasks use the MapReduce approach [3], which is in this paper represented by its open-source implementation Apache Hadoop. Ubuntu Server 12.04, Java 7 and Hadoop 2.2.0 are used for the deployment of the Apache Hadoop cluster.

The code is written in objective way and it can be extended to cover new situations and handle number of different tasks such as crisis management, social media monitoring, fast financial market decisions, customer segmentation, crime area identification based on the location based services, etc. The class diagram of the proposed application and its connection to the tools used by authors can be seen from the Fig. 2.

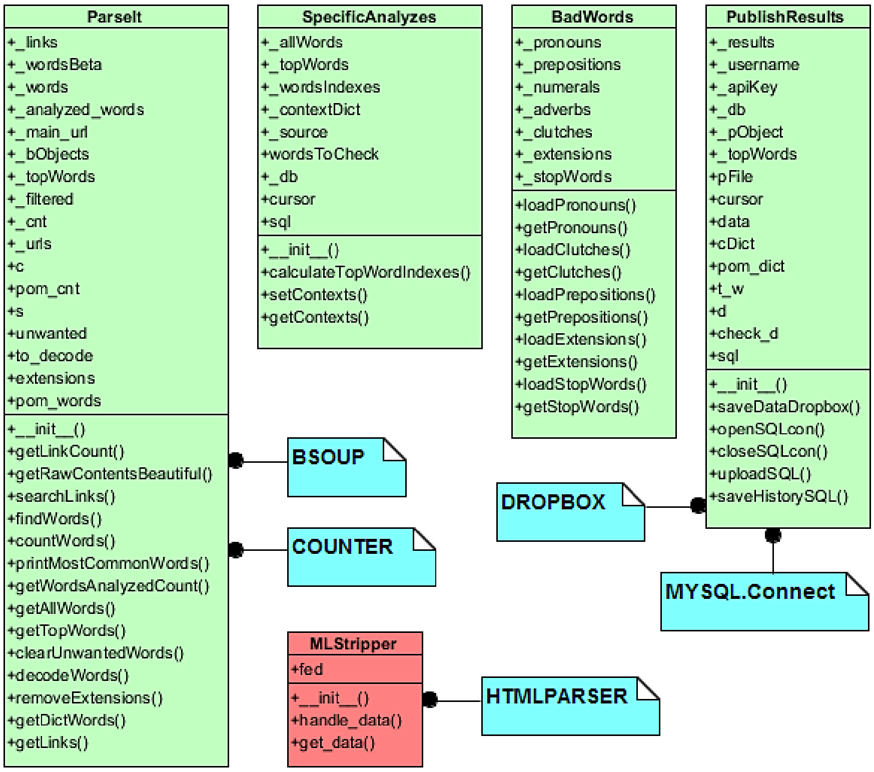

Fig. 2. Class diagram of the proposed application

4 A Description of the Big Web Data Application

4.1 Targets and the HTML Structure

Authors use the three biggest media servers in the Czech Republic. This fact is crucial, because the Czech language is characterized by hard inflection of individual words and their meanings. Targets are set to the www.novinky.cz, www.ct24.cz, www.m.idnes.cz. Desktop versions usually need more focus in the case of clearing the content, mobile versions are much clearer. The big web data mining application runs automatically in the specified time intervals on the server. This type of scan (over three servers with 20 000–30 000 words analysed) takes approx. 7 seconds.

The HTML request is made for each of these sources and the result is saved in the string variable. This variable contained everything, comments, JavaScript, tags and also a plain text of the articles. Then this text string is transformed by the BS library into the BS object (BO). Each source is then loaded by an object constructor and the title page is scanned for the relevant links that led to the individual articles. The only manual step before running the script is to explore the HTML structure for the identification of the targeted content.

4.2 Filtering and the Data Structure

Initially, the title page of the first mass media server is scanned for the all <h3> occurrences, because the new articles are characterized by this tag. All the content inside is scanned again for the <a> tag that contains links to the full text. Then, the content of this tag is returned and the string of the attribute "href" is saved into the associative array (dictionary).

Another iteration of this operation processes the founded links once again. They are converted into the BO one by one while the pure text of the article is extracted. Pure text is placed in the <div id="articleBody"> and this structure is global for the whole website. The content is returned and saved as a string ready for clearing. MLStripper class inherited from the HTMLParser and instantiated an object that cleares a string from the HTML tags and JavaScript. Punctuation is also removed by a simple regular expression. Words are stored in the JSON data structure, which is characterized as a key-value pair, where the value can be another associative array.

4.3 Removing Unwanted Content

Authors store all the words in the counter object. Short words (<=2 characters) are limited. Authors use a special class that operates with an access to the dictionaries of the unwanted words. These dictionaries are made during the tens of individual script runs and they are extended continuously.

In the Czech language, each word has many different shapes. For the purpose of this application, this problem is eliminated by using a JSON, where the key is the word in its default state and the value for this key is a list of alternatives.

4.4 Saving and Archiving of the Obtained Data

The obtained big web data are stored in the MySQL database on the daily basis since the deployment of the proposed application (January 2014) and then are saved to the Dropbox folder, as it is given in the following algorithm. Data from the MySQL database are then deleted daily. The plan is to build the big web data archive and save these big web data for the processing using the distributed approach and Apache Hadoop. Authors do not save or process these data in the real-time, so they did not use the technologies such as MySQL Binlog, Hadoop Applier, etc.

```
def saveHistorySQL(self):
  t_w = self._topWords
  d = self.openSQLcon()
  cursor = d.cursor()
  if (datetime.datetime.now().strftime("%H") == '00'):
      print("MIDNIGHT SCAN - CLEARING HISTORY DATA")
  cursor.execute("""TRUNCATE table realmining.HISTORY""")
      d.commit()
  else:
      print("ADDING HISTORY DATA")
      check_d = cursor.execute("""SELECT * FROM infor-
mation_schema.tables WHERE table_schema = 'realmining'
AND table_name = 'HISTORY' LIMIT 1;""")
      d.commit()
      if (check_d):
          print("History table is ready, inserting data")
      else:
          sql = """CREATE TABLE realmining.HISTORY
(WORD_NAME VARCHAR(100) NOT NULL,WORD_LENGTH
INT(3),WORD_COUNT INT(5),CONTEXT_STRING
VARCHAR(250),FULL_URL VARCHAR(250),START_TIME
TIME,START_DATE DATE) DEFAULT CHARACTER SET utf8 COLLATE
utf8_general_ci"""
          cursor.execute(sql)
          d.commit()
          print("History table doesn't exist")
      for top_w in self._topWords:
          wrdn = top_w[0].upper().encode("utf-8")
          wrdl = top_w[1]
          wrdc = top_w[2]
          cont = top_w[3]
          furl = top_w[4]
          cursor.execute("insert into realmining.HISTORY
(WORD_NAME,WORD_LENGTH,WORD_COUNT,CONTEXT_STRING,FULL_URL
,START_TIME,START_DATE) values
(%s,%s,%s,%s,%s,TIME(),DATE())",
(wrdn,wrdl,wrdc,cont,furl))
          d.commit()
      d.close()
def saveDataDropbox(self,path):
     pFile=open(path,"w+")
     for tpl in self._results:
pFile.write(str(tpl[0])+","+str(tpl[1])+","+str(tpl[2])+"
,"+str(tpl[3])+","+str(tpl[4])+"\n").encode("cp1250")
    pFile.close()
```

5 Distributed Approach and the Processing of Big Web Data

There are two main ways to process the big web data. The first one requires some hardware costs and investments or at least configuration of the computer cluster. This can be done by installing Apache Hadoop, an open-source platform written in Java for reliable, scalable and distributed computing, under the UNIX type operating system. The second option is to save all the hardware costs and use some of the commercial tools and sources (e.g. Google, Amazon, etc.). Both of these solutions have the similar base. The replicated File System (FS), which handles all the parallel and distributed processing operations automatically. It can be a job distribution, optimization, error handling, thread access and many more complicated tasks. In the case of the open-source solution, the file system is called the Hadoop File System (HDFS), in the case of the commercial solution, it is called the GFS. The descriptions of Master node and Slaves configuration together with the utilization of the commercial services are not included in this paper. Nevertheless, the topic of the big web archives and Apache Hadoop in practice has been clearly documented by Holmes [11].

5.1 A Big Web Data Archive and its Structure

Although there is a lot of technical information about Apache Hadoop, there is not much information about how to effectively structure data in a Hadoop environment. Even though the parallel processing systems provide an optimal environment for the processing big web data, the structure of the data itself plays a key role [12].

The MySQL relational data model of the proposed application is consisted of one database table, which is represented by the Fig. 3.

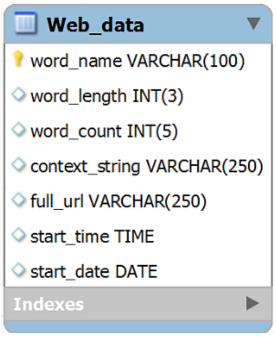

Fig. 3. Relational data model

Data stored in the database can be retrieved using Structured Query Language (SQL). On the daily basis, the data are moved from MySQL to the Dropbox folder as a single file. Through the run of the proposed application approx. 2–3 MB of the data were obtained daily. For the 15 months run it means 1.5 GB data file to process. This file was daily updated through incremental data transfer. The file was saved as a CSV formatted text file because in the CSV format a line break would start a new line. In order to use this type file, class CsvRecordInput in Apache Hadoop can be used.

5.2 Architecture of the Cluster for the Big Web Data Processing

Apache Hadoop is chosen especially for the advantages presented in Pavlo et al. [5] and Chen et al. [1], mostly because of much easier set up and use of this approach. Authors focus on a purely distributed solution to emulate a small-scale Apache Hadoop cluster for testing purposes with low costs and without utilization of a cloud service, not a large-scale production environment.

A simple architecture for the processing of the big web data is proposed. Only 5 PCs are used, one of them worked as a master, the others as slaves. The proposed cluster is based on Apache Hadoop 2.2.0 and Java 1.7. Ubuntu Server 12.04 is used as an operating system running on all PCs. All the computers are connected by 100 Mbps Ethernet to a dedicated Virtual Local Area Network (VLAN). A proposal for the speeding up the processing of the big web data by the proposed cluster and the distributed approach can be seen in the Fig. 4.

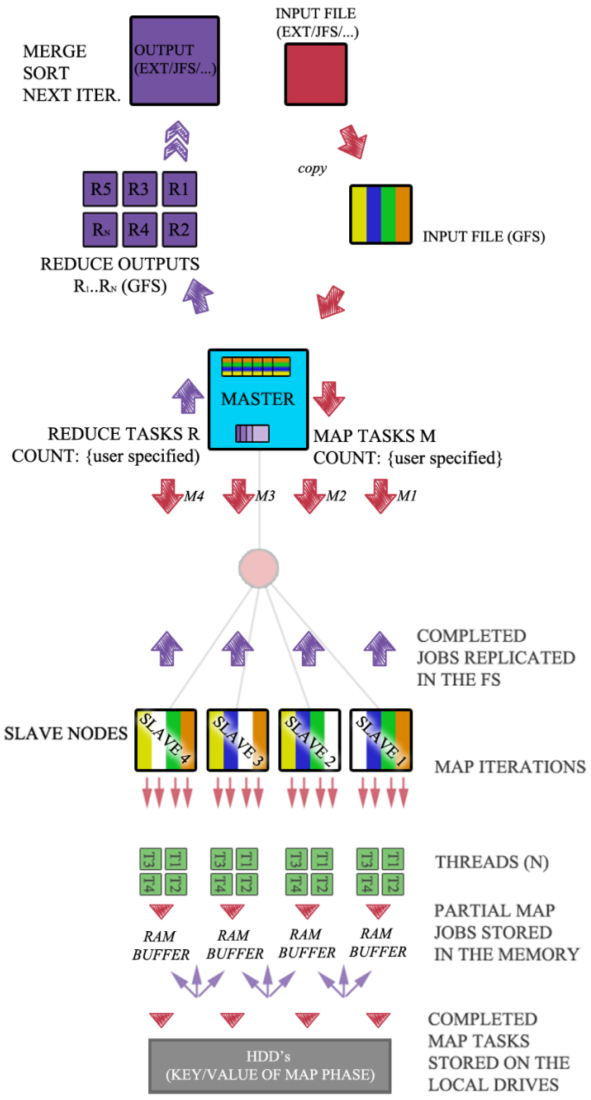
MERGE
SORT
NEXT ITER.
OUTPUT (EXT/JFS/...)
INPUT FILE (EXT/JFS/...)
copy
R5 R3 R1
RN R4 R2
INPUT FILE (GFS)
REDUCE OUTPUTS R1..RN (GFS)
MASTER
REDUCE TASKS R COUNT: {user specified)
MAP TASKS M COUNT: {user specified}
M4 M3 M2 M1
COMPLETED JOBS REPLICATED IN THE FS
SLAVE NODES
SLAVE 4 SLAVE 3 SLAVE 2 SLAVE 1
MAP ITERATIONS
T3 T1 T4 T2
THREADS (N)
PARTIAL MAP JOBS STORED IN THE MEMORY
RAM BUFFER
COMPLETED MAP TASKS STORED ON THE LOCAL DRIVES
HDD's (KEY/VALUE OF MAP PHASE)

Fig. 4. Speeding up the processing of the big web data using the distributed approach

5.3 A framework for the Big Web Data Processing with Apache Hadoop

The main challenge for this type of the big web data is to compute statistics for all the records on various combinations of reference dimensions. Table 1 then shows some selected methods based on the web content mining techniques and also trend analysis, which can be used with Apache Hadoop.

Table 1. The framework of the selected methods for the big web data processing

The type of the analysis	Key feature with Apache Hadoop
Trend analysis of the targeted words through the time, e.g. the use of the words on the daily basis, which words are used more frequently than the others, the comparison of the word between the mass media servers, etc.	e.g. WordCount example reads text files and counts how often words occur. This script can be also easily modified based on the selected attributes (see the chapter: A big web data archive and its structure). The barrierless version of WordCount algorithm is presented by Verma et al. [13].
Association	e. g. finding the rules associated with frequently co-occurring words.
Attribute importance	e. g. ranks words according to strength of relationship with target word, extracting a vector of meaningful content words.
Distance matrix	e. g. the Euclidean distance calculation between the two words.
Correlation analysis	e. g. the most correlated words based on the different distances (1, 2, 3), correlation matrices.
Clustering	e. g. groups words together into clusters based on the time or distance.
Classification	e. g. the problem of identifying to which of a set of predefined categories (art, sport, etc.) a word belongs.
Web content mining techniques Business intelligence techniques	A lot of scripts were already written for this area. More can be found e.g. in Holmes [11].

6 Conclusions and Future Research

In this paper, authors offer an effective approach to analyze publicly available and valuable content on the Web with the primary aim of building the big data archive. The proposed big web data application demonstrates the basic principles, which can be further broadly expanded to mine other relevant information resources over the Web. The second part of this paper is focused on the importance of the distributed approach and Apache Hadoop in the processing of the big web data. The deployment of this solution can save the resources, especially costs, and get the results in the reasonable time, which is an important part in the decision-making process.

There are three phases of the big web data value chain covered by this paper: a) big web data generation; b) big web data acquisition; c) big web data storage. The last phase, which is focused on the big web data analytics, is only mentioned through the framework of the selected analyses. It will be content of the future work and it will be presented together with the comparison and processing of the obtained big web data, the performance analysis of the proposed cluster solution and the implementation of the selected methods in Python or Java for the use with Apache Hadoop.

Acknowledgements. This paper was supported by the SGSFES_2015001 fund.

References

1. Chen, M., Mao, S., Liu, Y.: Big Data: A Survey. Mobile Networks and Applications **19**(2), 171–209 (2014)
2. Tien, J.M.: Big Data: Unleashing Information. Journal of Systems Science and Systems Engineering **22**(2), 127–151 (2013)
3. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Communications of the ACM **51**(1), 107–113 (2008)
4. Vilas, K.S.: Big Data Mining. International Journal of Computer Science and Management Research **1**(1), 12–17 (2012)
5. Pavlo, A., et al.: A comparison of approaches to large-scale data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pp. 165–178. ACM (2009)
6. Power, D.: Using 'Big Data' for analytics and decision support. Journal of Decision Systems **23**(2), 222–228 (2014)
7. Zhang, Q., Segall, R.S.: Web mining: A survey of current research, techniques, and software. International Journal of Information Technology & Decision Making **7**(4), 683–720 (2008)
8. Jackson, Q.T.: Efficient formalism-only parsing of XML/HTML using the § -calculus. ACM SIGPLAN Notices **38**(2), 29–35 (2003)
9. Peng, D., Cao, L., Xu, W.: Using JSON for Data Exchanging in Web Service Applications. Journal of Computational Information Systems **7**(16), 5883–5890 (2011)
10. Henning, M.: API design matters. Queue **5**(4), 24–36 (2007)
11. Holmes, A.: Hadoop in Practice. Manning, Shelter Island (2012)
12. Bradley, C., et al.: Data Modeling Considerations in Hadoop and Hive. Technical paper (2013)
13. Verma, A., et al.: Breaking the MapReduce stage barrier. Cluster computing **16**(1), 191–206 (2013)