Scholarly Very Large Data: Challenges For Digital Libraries

Jian Wu
Old Dominion University
Norfolk, VA, USA
jwu@cs.odu.edu

C. Lee Giles
Pennsylvania State University
University Park, PA, USA
giles@ist.psu.edu

ABSTRACT

The volume of scholarly data has been growing exponentially over the last 50 years. The total size of the open access documents is estimated to be 35 million by 2022. The total amount of data to be handled, including crawled documents, production repository, metadata, extracted content, and their replications, can be as high as 350TB. Academic digital library search engines face significant challenges in maintaining sustainable services. We discuss these challenges and propose feasible solutions to key modules in the digital library architecture including the document storage, data extraction, database and index. We use CiteSeerX as a case study.

KEYWORDS

scholarly big data, information extraction, digital library

1 INTRODUCTION

Since the use of the phrase “Scholarly Big Data” in a keynote of CIKM 2013, many open access (OA) scholarly datasets have been released. In 2014, the total number of scholarly documents in English online was estimated to be approximately 120 million [4], with a quarter OA. Another estimated the size of Google Scholar to be 160–165 million documents [6]. Recently, the Open Academic Society released the Open Academic Graph, unifying two big academic graphs: Microsoft Academic Graph (MAG) and AMiner. The MAG dataset contains about 210 million papers as of November 2018.

The number of scholarly papers increases at a rate of at least 1 million per year [1, 5]. All of this raises significant challenges for online digital libraries (DLs), such as CiteSeerX [3], which provides both search and download services. Besides the actual documents (usually in PDF), the size of related data extracted from PDF files, such as figures and tables is also substantial. Currently, CiteSeerX leverages big data techniques to extract, host, and analyze data. As such in the foreseeable future, “Scholarly Big Data” will soon be “Scholarly Very Large Data” (SVLD). How to get ready for this increase and make the service cost effective, durable, and sustainable is an open research question. In this paper, we present upcoming challenges of SVLD in the context of digital libraries, and use CiteSeerX as a case study. We also propose solutions that may be used for mitigating or solving other huge data problems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

LSN ’20, April 13–14, 2020, Chicago, IL
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

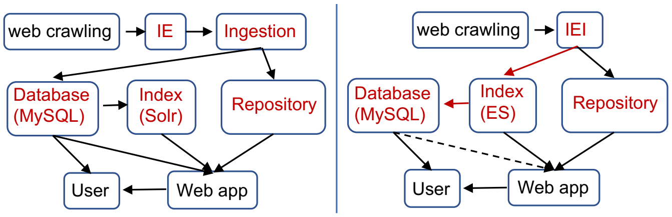

Figure 1: CiteSeerX architecture in 2018 (left) and the new proposed architecture (right). Red modules indicate SVLD bottlenecks. Red arrows mark changes. ES: ElasticSearch.

2 STATUS OF CITESEERX

CiteSeerX is a digital library (DL) search engine launched in 1998 [3]. As of 2018, it hosts more than 10 million OA academic PDF documents with full text. The top level system architecture is displayed in Figure 1. Heritrix, wget, and other customized crawling tools are used for web crawling. PDFMEF [8], which encapsulates GROBID, ParsCit, pdffigures2, and other open source content extractors are used for information extraction (IE). The ingestion module writes extracted metadata into a relational MySQL database and disambiguates near-duplicate documents. Ingestion also copies PDF, text, and xml files to the repository. A separate module pulls unindexed documents from the database and indexes them with Apache Solr.

Except for web crawling and the database, all servers run on a private cloud. The repository is a disk array mounted to a dedicated server and backed up to another. Solr runs on a dedicated server with one replicate. MySQL runs on dedicated servers with two replicates. The database is periodically backed up on Google Drive. The web application connects the database via JDBC. The search interfaces queries with SolrJ. The download service accesses the repository via a RESTful API. Files in the repository are organized in a hierarchical order such that a document with ID `10.1.1.1234.5678` is located under `10/1/1/1234/5678/`.

CiteSeerX’s goal is to crawl and index all OA academic documents online, estimated to be 27 million in 2014. A recent study[7] estimates that at least 28% of the scholarly literature is OA and the number of OA papers per annum is nearly half of all published. With an increase of 1 million annually, the number of OA papers will reach 35 million in 2022. The current and anticipated sizes of CiteSeerX in 2019 and 2022 is shown in Table 1. Because of its size and the increasing growth rate, its data will soon become SVLD, imposing many challenges to the infrastructure and software.

3 CHALLENGES IN MANAGING A DL

The **repository** in Table 1 shows that a cached download service for all OA scholarly papers must have at least 53-terabytes (TB). Adding the database (1.8TB), the index (1.2TB), and a backup, the total disk space needed is about 109 TB. This can be fulfilled by a rack server

Table 1: CiteSeerX size in 2019 and in 2022.

Type	2019	2022	Notes
Full text papers	10M	35M	
Database	550GB	1.8TB	2 replicates
Largest table (rows)	250M	875M	
Database dump	300GB	900GB	
Database buffer	38GB	128GB	10% cached
Indexed records (docs)	70M	245M	
Index size	360GB	1.2TB	1 replicate
Index heap	36GB	120GB	10% cached
Repository	15TB	53TB	1 backup
PDF	10TB	35TB	
TXT	900GB	3TB	
XML	400GB	1.4TB	
Figures	10TB	35TB	estimated
Crawl	43TB	150TB	linear

such as a Dell PowerEdge. However, backing up and sharing such a repository across servers is non-trivial. The current network used by CiteSeerX allows transfer of a single large file between servers at 20MB/s. With this bandwidth, the time to backup the entire repository will be at least 32 days. Each paper is associated with one `.txt` and one `.xml` file. The total number of files and folders to move in a backup is at least $35 \times 4 = 140$ million.

Another challenge comes from **automatic data extraction**. So far, CiteSeerX has focused on a *document-level* search service. Our goal is to provide *content-level* search. However, the size of extracted content from the PDF documents also takes storage space. In an experiment, we extracted 15+ million figures from about 6.7 million PDF documents [2], requiring 7TB. Scaling up to 35 million papers, the number of extracted figures would be 78 million, which uses about 35TB of storage, comparable to the storage of PDFs (Table 1).

With high-end servers with Solid State Disks (SSDs) and TB-level RAM, the **database** and the index can fit into a single server. However, when the number of papers reaches 35 million, the largest database table (`citations`) will contain about 1 billion rows. Although querying a single table can be fast, joins will be time consuming and be unacceptable to the user. Backup is also an issue with dumping or importing taking days to weeks.

The **index** with Apache Solr can easily scale up to 80+ million documents on a single server. But with about 245 million documents, simply scaling its memory heap will decrease performance. Currently, CiteSeerX runs on Solr 4.9 in a Master-Slave mode, which is not scalable. The SolrCloud+ZooKeeper solution requires at least 7 nodes to achieve 2 replicates and 2 shards each (3 for ZooKeeper and 4 for Solr); setting it up is nontrivial.

In the current architecture, MySQL and Solr have significant data overlap. Because the index is updated on Solr after MySQL is populated, data inconsistency may occur. With terabytes of data, MySQL in production requires high speed storage such as SSDs and more than 100GB of RAM to cache about 10% of the data.

4 POSSIBLE SOLUTIONS FOR SVLD

Table 1 shows that data generated by web crawling can be very large. One solution is to scan all documents in the crawl repository and remove the ones already ingested. The remaining documents are zipped in small batches and archived in a box.com, which provides *unlimited* storage through a contractual service. Another would be to expand the storage and store it as a whole in a storage-area network (SAN). The web servers access the storage via iSCSI, which means the repository is easily scaled up to more than 40TB.

The repository backup is mitigated by *parallel data movement*. The whole repository is distributed across 10 partitions, each hosting over 3 million documents. When backing up the repository, the 10 partitions are copied in parallel, effectively using bandwidth and significantly reducing transfer time. Integrity is maintained by a hash table in a key-value database, which records whether corresponding files are backed up or not. Because files in the repository are less volatile, we use SATA drives. Extracted content (e.g., figures) is stored with the same PDF document.

A MySQL cluster (Networked DataBase or NDB) is not designed for full text search. Our solution is to swap the positions of Index and Database (Figure 1, left) and replace Solr with ElasticSearch. In the new architecture (Figure 1, right), the IE and ingestion are integrated into one module called IEI. New data is ingested into ElasticSearch and the repository used for web application. Now metadata in the MySQL database only is only used for research. ElasticSearch scales horizontally using commodity rack servers and provides high availability by sharding data cross multiple nodes.

5 CONCLUSION

The sheer number of scholarly documents and extracted content generate *Scholarly Very Big Data*, imposing significant challenges to an academic digital library. The CiteSeerX solution is using a public cloud, SAN, and parallel data movement for the repository. ElasticSearch is employed as both a data store and a search platform. The database is taken out of production and only used for research.

REFERENCES

[1] Lutz Bornmann and Rüdiger Mutz. 2015. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *JASIST* 66, 11 (2015), 2215–2222.

[2] Sagnik Ray Choudhury, Shuting Wang, and C. Lee Giles. 2016. Scalable algorithms for scholarly figure mining and semantics. In *Proceedings of the International Workshop on Semantic Big Data*.

[3] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. [n.d.]. CiteSeer: An Automatic Citation Indexing System. In *Proceedings of JCDL 1998*.

[4] Madian Khabsa and C. Lee Giles. 2014. The number of scholarly documents on the public web. *PLoS ONE* 9, 5 (May 2014), e93949.

[5] PederOlesen Larsen and Markus von Ins. 2010. The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics* 84, 3 (2010), 575–603.

[6] Enrique Orduna-Malea, Juan M. Ayllón, Alberto Martín-Martín, and Emilio Delgado López-Cózar. 2015. Methods for estimating the size of Google Scholar. *Scientometrics* 104, 3 (Sep 2015), 931–949.

[7] Heather Piwowar, Jason Priem, Vincent Larivière, Juan Pablo Alperin, Lisa Matthias, Bree Norlander, Ashley Farley, Jevin West, and Stefanie Haustein. 2018. The state of OA: a large-scale analysis of the prevalence and impact of Open Access articles. *PeerJ* 6 (02 2018), e4375–e4375.

[8] Jian Wu, Jason Killian, Huaiyu Yang, Kyle Williams, Sagnik Ray Choudhury, Suppawong Tuarob, Cornelia Caragea, and C. Lee Giles. [n.d.]. PDFMEF: A Multi-Entity Knowledge Extraction Framework for Scholarly Documents and Semantic Search. In *Proceedings of K-CAP 2015*.