CiteSeer$^{\mathcal{X}}$: an Architecture and Web Service Design for an Academic Document Search Engine

Huajing Li[1]
huali@cse.psu.edu

Isaac Councill[2]
icouncil@ist.psu.edu

Wang-Chien Lee[1]
wlee@cse.psu.edu

C. Lee Giles[1,2]
giles@ist.psu.edu

[1]Department of Computer Science and Engineering
[2]Information Sciences and Technology
Pennsylvania State University
State College, PA, 16802, USA

ABSTRACT

CiteSeer is a scientific literature digital library and search engine which automatically crawls and indexes scientific documents in the field of computer and information science. After serving as a public search engine for nearly ten years, CiteSeer is starting to have scaling problems for handling of more documents, adding new feature and more users. Its monolithic architecture design prevents it from effectively making use of new web technologies and providing new services. After analyzing the current system problems, we propose a new architecture and data model, CiteSeer$^{\mathcal{X}}$. CiteSeer$^{\mathcal{X}}$ that will overcome the existing problems as well as provide scalability and better performance plus new services and system features.

Categories and Subject Descriptors

H.3.6 [**INFORMATION SYSTEMS**]: Library Automation—*Large text archives*; H.3.4 [**INFORMATION SYSTEMS**]: Systems and Software—*Distributed systems*; C.5.5 [**COMPUTER SYSTEM IMPLEMENTATION**]: Servers

General Terms

Design, Management, Performance

Keywords

System architecture, Data model, Scalability

1. INTRODUCTION

CiteSeer has become a popular web-based scientific literature digital library and search engine that focuses primarily on the field of computer and information science. The hallmark feature of CiteSeer is Autonomous Citation Indexing (ACI) [3], which automatically extracts citation information from scholarly publications in electronic format. From its conception in 1997 until now, CiteSeer has grown into a collection of over 730,000 documents with over 8 million citations. The rising demands from system use and the increasing size of CiteSeer's archive have resulted in rising query latencies as well as significant degradation of system stability. The CiteSeer architecture design is monolithic, making the system difficult to administer, configure, and modify.

The introduction of a modular CiteSeer architecture offers a great opportunity to exploit newly emerged web technologies to improve the system's performance and configurability. Accordingly, the system will become more powerful and robust: more simultaneous transactions can be supported; and tasks like system monitoring and logging will be faciliated. The next generation CiteSeer is far more than a debugged version of CiteSeer. It is also a system with new elements: new services, new resources, and new data models, all working in a new framework. Our design goal is to setup a new architecture which can be scalable, flexible, self-adaptive and user-oriented.

2. CITESEER$^{\mathcal{X}}$ SYSTEM DESIGN

CiteSeer$^{\mathcal{X}}$ is built upon a new data model which has the following new features:

Extended Data Models: Expanding the old document-centric approach, CiteSeer$^{\mathcal{X}}$ introduces author and venue records into the system. These records are no longer considered as metadata *belonging to* a document, but as peer digital objects that are linked to documents as well as to each other.

Virtual Documents: Metadata of an existing paper can be inferred from various information sources, such as its citations and researcher publication lists. It is not rare to find the corresponding document has not been retrieved by the system. To address such cases, we propose the notion of *virtual documents*, which are built upon incomplete metadata and act as a *placeholder* of the document.

Digital Objects: To make CiteSeer$^{\mathcal{X}}$ more flexible and extensible in terms of storage types and service types, the notion of *digital objects*[2] is introduced into the system, separating physical storage from service access. As such, a level of abstraction can be defined on top of the physical storage. Actual physical storage can be distributed across multiple machines and sites.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-332-9/06/0005.

Based on the new data model, CiteSeer$^{\mathcal{X}}$ will have a new modular system architecture to overcome the limitations previously described. Figure 1 gives an overview of the new architecture. Basically, the system comprises three layers:

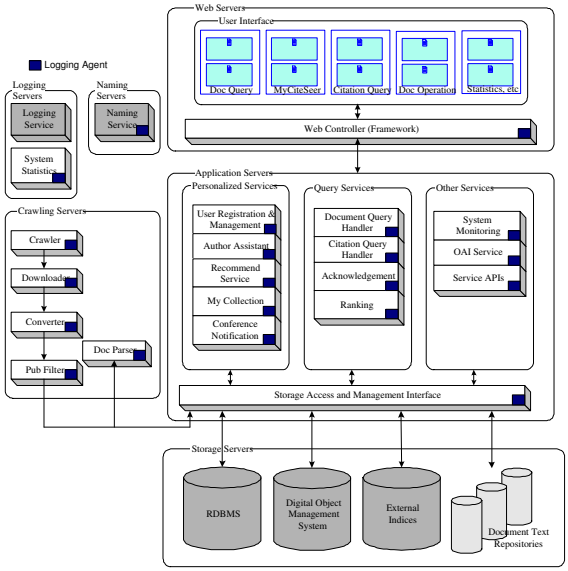

Figure 1: CiteSeer$^{\mathcal{X}}$ Architecture.

Storage Layer: The storage layer handles the management of and access to locally stored data objects of CiteSeer$^{\mathcal{X}}$. These objects are maintained by a digital object management system. Each digital object is accompanied by a description file that contains the metadata of the object.

Application Layer: The application layer is the collection of the function modules and services in CiteSeer$^{\mathcal{X}}$, which includes naming service, logging service, crawling service, query handling service, indexing service, personalized service, etc.

User Interface Layer: This layer provides an abstraction for the web interface of CiteSeer$^{\mathcal{X}}$ by acting as a gateway between the user interface and application modules and gives flexibility to update the application logic without worrying about the user interface as well as providing personalized services to users.

Under the proposed architecture, CiteSeer$^{\mathcal{X}}$ continues to support its existing services with more user-friendly interfaces and application-accessible APIs. New services and features are added into the system as well.

CiteSeer-Specific Services: These services are specific to CiteSeer and would provide value added if made available on the semantic web. These services enable the processing of citations and the navigation through those citations. This set of services includes the metadata extraction service, citation graph service, indexing service, metadata service, electronic repository service, electronic conversion service, duplicate identification service, etc. They are described in more detail in previous work[4].

Acknowledgement Extraction: We have developed an algorithm for automatic acknowledgement extraction in order to extend the native extraction capabilities of CiteSeer [1]. This initial algorithm identifies acknowledging text passages and extracts the names of acknowledged entities. This data is stored in an auxiliary index alongside CiteSeer's traditional indices with special bridges created to integrate the new acknowledgement data into the existing CiteSeer system. This integration represents a structural shift in entity relationship handling in CiteSeer, requiring an extension of the traditional author-document and document-document relationships to reified relations between entities and documents, flexibly modeling the roles of entities within the data.

Distributed Usage Logging Service: An XML based description language for information retrieval system usage logs is introduced in CiteSeer$^{\mathcal{X}}$ by modeling a user-system interaction ontology. The language encompasses rich semantic descriptions of the events being logged such as dependency between successive actions. The logging service architecture of CiteSeer$^{\mathcal{X}}$ reflects the idea of detaching the logging service from the target system such that is is no longer the duty of each module in the system to write document usages. Instead, an independently running logging service collects and manages logs from every module.

MyCiteSeer: For an ACI system like CiteSeer, it becomes increasingly difficult for users to find information that accurately matches their needs with the growth of the number of stored documents. In such scenarios, a user's query context, as well as his personal interests, can be taken into consideration in answering a user's query and effectively filter the results. To support personalized services, CiteSeer$^{\mathcal{X}}$ provides registration mechanism to profile users. The new logging framework and log schema are user-aware and session-aware, by which data mining algorithms and recommendation techniques can be applied.

3. CONCLUSIONS

This paper presents CiteSeer$^{\mathcal{X}}$, the next generation CiteSeer architecture that is designed to overcome the challenges of interoperability, extensibility and scalability of the current CiteSeer system. It is a completely new architecture that is based on a modular approach with web services, pluggable service components, distributed object repositories and transaction-safe processes, all utilizing an enriched data model.

4. ACKNOWLEDGMENTS

We gratefully acknowledge partial support from Microsoft Research, National Science Foundation and NASA.

5. REFERENCES

[1] C. L. Giles and I. G. Councill. Who gets acknowledged: measuring scientific contributions through automatic acknowledgement indexing. *Proceedings of the National Academy of Sciences*, 101(51):17599–17604, 2004.

[2] R. Kahn and R. Wilensky. A framework for distributed digital object services. *Technical Report, cnri.dlib/tn95-01*, 1995.

[3] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and Autonomous Citation Indexing. *IEEE Computer*, 32(6):67–71, 1999.

[4] Y. Petinot, C. L. Giles, V. Bhatnagar, P. B. Teregowda, H. Han, and I. Councill. A service-oriented architecture for digital libraries. In *ICSOC*, pages 263–268, 2004.