J3.12 EFFECTIVE RETRIEVAL PERFORMED BY DIMES WITH THE APPLICATION OF LUCENE

Yujie Zhao, Ruixin Yang*, Menas Kafatos
Center for Earth Observing & Space Research (CEOSR)
George Mason University (GMU)

ABSTRACT

Earth science metadata search engines dealing with large volumes of Earth science metadata need an effective method to respond to queries and retrieve metadata very quickly. Basically, Earth science metadata includes three types of information: descriptive text, temporal range, and spatial range. So, except for the general text retrieval, Earth science metadata retrieval requires a temporal range query and a spatial range query, which are not widely needed by other types of search engines.

DIMES, a distributed metadata server, is being tested for use of Lucene to tackle the search problem of Earth science metadata. The vantage points of Lucene are that it is not only a high-performance text search index, but also has a special query type called range query which is especially fit for temporal range and spatial range queries of Earth science metadata once the temporal and spatial information are converted to strings with alphabetic orders. This paper introduces how DIMES applies the Lucene range query for applicable temporal and spatial range searches in the Earth sciences.

1. INTRODUCTION

As more and more data for Earth science research are available from Earth observing, in particular, satellite remote sensing, and from models, efficient mechanisms for finding and delivering distributed data become necessary. One well-known data delivery infrastructure is OPeNDAP (OPeNDAP, 2004). OPeNDAP was enhanced by being combined with the data analysis capability of GrADS (the Grid Analysis and Display System) (Doty *et al.,* 1997) to form GDS (GrADS Data Server) (Wielgosz *et al.,* 2001), which allows users to define operations performed on the server side and to obtain the resultant information (processed data) via the Internet.

The XML-based DIMES (Yang *et al.,* 2001) implemented a flexible metadata model and web-based metadata navigation interfaces to support various level metadata accesses. A Metadata Integrated Data Analysis Server (**MIDAS**) was developed by combining DIMES with GDS, initially named the Scientific Data and Information Super Server (SDISS) (Yang, Kafatos & Wang, 2002; Yang *et al.*, 2003). In the following, we will review the MIDAS architecture and its search interface, and then emphasize the recent implementation of effective temporal and spatial query of DIMES based on Lucene.

2. MIDAS ARCHITECTURE AND MAJOR COMPONENTS

Figure 1 is the high-level system architecture of MIDAS. Major components of the system are certainly GDS and DIMES. A GDS URL Generator is included in the architecture to help users to build the relatively complex GDS URLs through a GUI. The MIDAS is designed to be a distributed system, and therefore a register can be added to record all available MIDAS. Certainly, the MIDAS register itself can be distributed providing information on each server or even residing on the client side. However, the centralized register will make it easier to reach broader audiences by leveraging the existing centralized metadata search engines such as GCMD (Olsen & Major, 1996). On the server side, setting up a MIDAS starts both the data server and the metadata server. Metadata in a MIDAS will be

* Corresponding author address: Ruixin Yang, MS 5C3, School of Computational Sciences, George Mason University, Fairfax, VA 22030; e-mail: ryang@gmu.edu.

ingested and checked to reflect the changes on the data holdings served by the current data server via the ingest tool box.

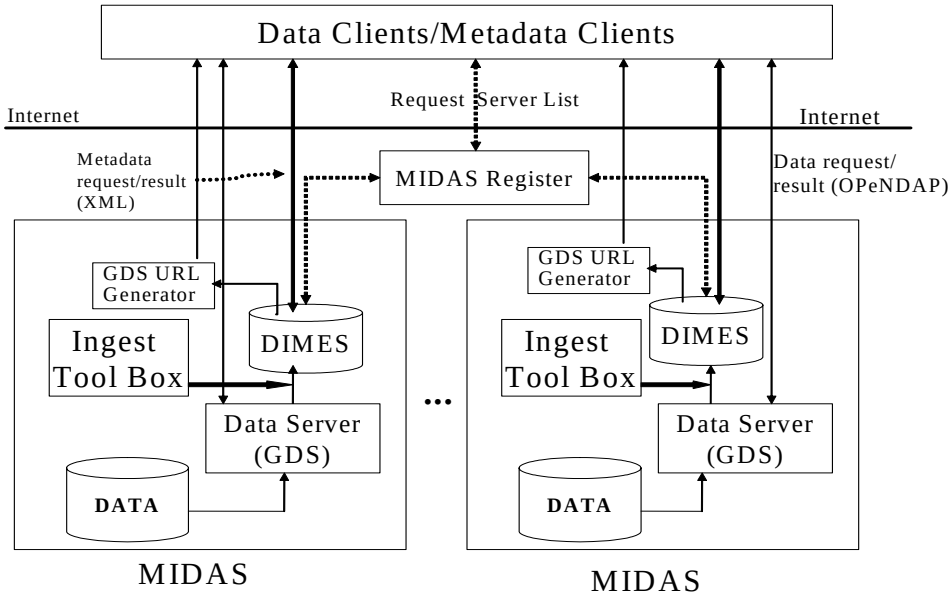

Figure 1. The high-level system architecture of MIDAS

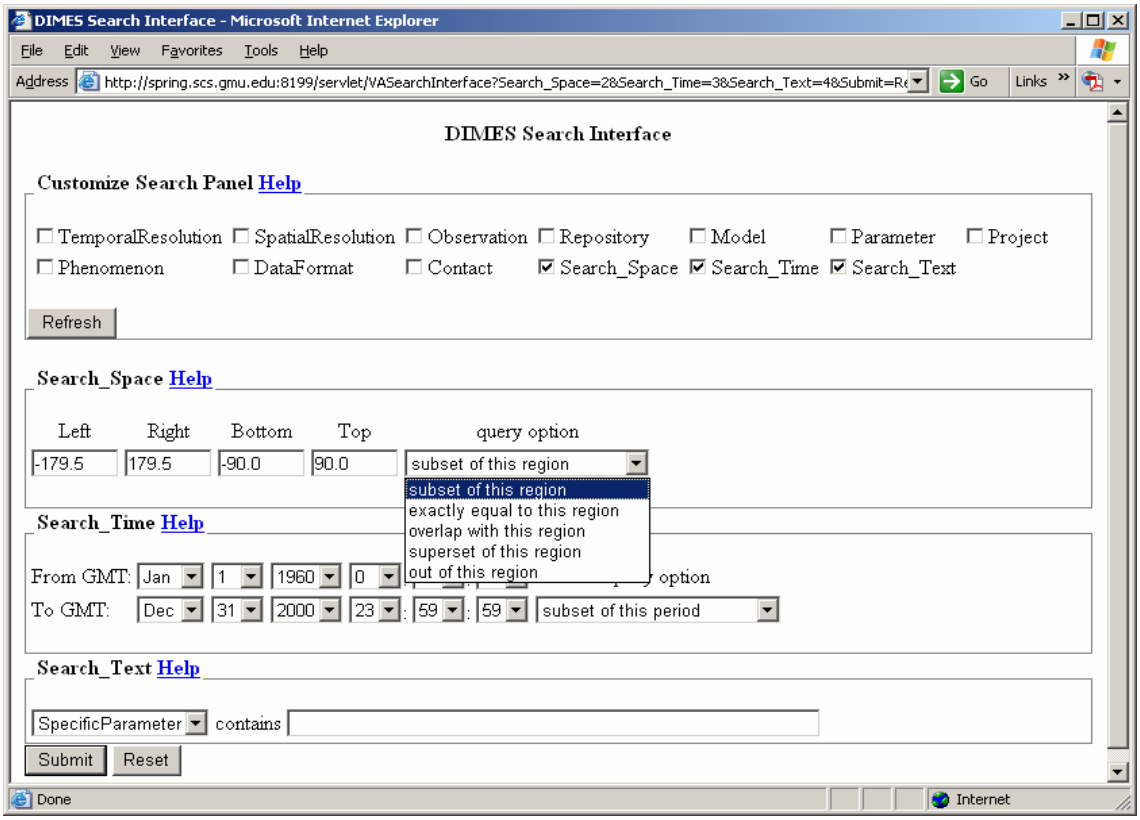

Figure 2. Metadata Search Interface of DIMES

Figure 2 is the search interface of DIMES, which primarily focuses on three basic types of search: temporal, spatial, and textual search, as well as other knowledge searches such as Observation, Project, etc. The knowledge is added by the DIMES users depending on the users' interest. Prior to submitting a query, the search panel should be customized by checking the related query conditions so that the conditions' input panels can be shown to the user. As in Figure 2, three query conditions, spatial, temporal, and textual, are selected, and the input panels of the three conditions are displayed. And then, the detailed query information can be set up in the conditions' panels.

Compared to the temporal and spatial ranges associated with datasets in the metadata, DIMES provides five types of queries given a temporal or spatial range: they are (1) the sub-range of the given, (2) the exact range as the given, (3) the overlap with the given range, (4) the super-range of the given, and (5) out of the given range. The "query option" can be chosen to set the preferred type, and the default type is sub-range of the given.

A spatial query input includes "left", "right", "bottom", and "top" to specify the longitude and latitude bounds. DIMES uses longitude representation -180^0 to 180^0. A longitude bound such as left = -120^0 and right = 90^0 means that it covers the range from -120^0 to 90^0. If left is greater than right, for instance left = 120^0 and right = -30^0, then it still goes from left to right, or alternatively from 120^0 to 330^0 under the 0^0 to 360^0 representation. Latitude uses -90^0 to 90^0 representation.

3. LUCENE AND RANGE QUERIES

Lucene, created by an experienced developer of text-search and retrieval tools, is a powerful text indexing and searching engine library written in Java, which gives Lucene cross-platform flexibility and straightforward integration with DIMES, written also in Java. Although it is open source and free software, Lucene is a high-performance software package and easy to use.

Basically, Lucene index is constructed by elementary units called **document**. Every document is corresponding to a unique textual ID, and it also contains a few fields, each of them having a name and a textual value. Figure 3 is an example of a document, which has three fields: the first field is as the unique ID of this document; the second field gives the start time of this data set; the third field gives the end time of this data set. To query a Lucene index, a field name and the expected textual value of this field are given in the query, and Lucene returns the unique document IDs whose documents have the specified field and textual value. In this example, if we give Lucene index the field name, StartTime, and the value, 20011203123030, then we will get the ID, DataSet5. Correspondingly, a simple Lucene query syntax requires two parts: a field and a term (the term is the textual value of the field) as **(StartTime: 20011203123030)**. And a compound query is the combination of this sort of simple queries by logic relations of **NOT**, **AND**, and **OR**.

Lucene supports a special type of query, Range Queries, which retrieve indices with strings' alphabetic order between the lower and upper bounds specified by a given range. And inclusive range queries use square brackets, and exclusive range queries use curly brackets. Range query **(StartTime: [19990301000000, 20000709235959])** retrieves the data sets whose start time is between March 1, 1999 and July 9, 2000 inclusively.

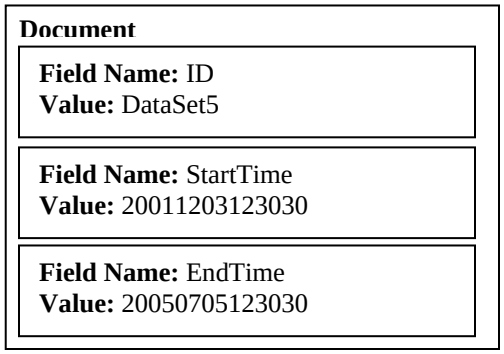

Figure 3. A Document of Lucene Index

The range queries are especially fit for the implementation of the temporal and spatial range search of Earth science data. As known, temporal and spatial queries are different from textual queries. Textual queries are often involved phrase matches, and very like point matches; temporal and spatial queries always deal with a range of matches, a set of points.

So the index supporting temporal and spatial queries must build the temporal and spatial orders in it. Lucene index keeps the orders of its indices, and in the meantime, considering its high performance, it is an efficient tool for temporal and spatial queries.

4. IMPLEMENTATION OF TEMPORAL QUERIES

In order to make time fit for range queries, time has to be converted to a string, and this conversion must keep the alphabetic order of the converted strings consistent with the order of their corresponding time. We use the string format "YYYYMMDDHHmmSS" in Lucene index to represent time, where YYYY is year, MM is month, DD is day, HH is hour, mm is minute, and SS is second, and this format keeps the order of its original time. For example, the beginning of May 6, 1999 is transformed to string "19990506000000." After the transformation, range queries become suitable for the retrieval of temporal queries.

A Lucene range query must have a lower bound and an upper bound to represent the relations of less than (lack of lower bound) and greater than (lack of upper bound). Therefore, we must define the earliest time and the latest time respectively prior to the implementation of range queries. We use "00000000000000" as the earliest time (because no practical Earth science metadata covers earlier time than this), and "99991231235959" as the latest time (because, currently, no practical Earth science metadata covers later time than year 9999).

Now, for the temporal query range given by **[start_time, end_time]**, the five types of temporal queries in Lucene query syntax are as the follows. Here, the "StartTime" and "EndTime" are the field names, and the second type, exact as the given range, is not range queries.

(1) sub-range of the given range: **(StartTime: [start_time, end_time]) AND (EndTime: [start_time, end_time])**, which means that both the start time and the end time of a metadata are between the given start time and end time;
(2) exact as the given range: **(StartTime: start_time) AND (EndTime: end_time)**, which means that both the start time and the end time of a metadata are the same as the givens;
(3) overlaps with the given range: **(StartTime: [start_time, end_time]) OR (EndTime: [start_time, end_time])**, which means that either the start time or the end time of a metadata is in the given temporal range;
(4) super-range of the given range: **(StartTime: [earlist_time, start_time]) AND (EndTime: [end_time, latest_time])**, which means that the start time of a metadata is less than the given start time and the end time of a metadata is more than the given end time;
(5) out of the given range: **(StartTime: {end_time, latest_time}) OR (EndTime: {earlist_time, start_time})**, which means that either the start time of a metadata is more than the given end time or the end time of a metadata is less than the given start time.

5. IMPLEMENTATION OF SPATIAL QUERIES

Longitude and latitude need to be converted to strings, too. For latitude, we use the transformation {X→Y=X+90}, and we represent Y with a 5-letter string, the first 3 letters for the integer part and the last 2 letters for the decimal part. For example, 33.19S (-33.19) degree is transformed to 56.81 and then represented by "05681," and 33.4N (33.40) degree is transformed to 123.40 and then represented by "12340." Now, the range bounded by "05681" and "12340" is the same as latitude interval between 33.19S degree and 33.40N degree. Similarly, for longitude, we use the transformation {X→Y=X+180} to convert a longitude to a 5-letter string by the same way as latitude.

Compared to temporal queries, spatial queries have two special characteristics. First, a spatial query is two dimensional, latitude and longitude; second, longitude has two formats of representation: one is from -180^{o} to 180^{o}; the other is from 0^{o} to 360^{o}. The expectation of DIMES is to retrieve the metadata of both representations given a longitude range from -180^{o} to 180^{o}.

To tackle the first problem, we separate a spatial query into two parts: a longitude query and a latitude query, and connect them with a logic relation. The relation between each of the first four types of spatial queries is **AND**, and the relation between the last type is **OR**.

Our strategy to tackle the second problem is, given a longitude range, to form a longitude query for each representation, and join them with a logic relation, either **AND** or **OR** depending on an actual query type. Regarding the fact that the discontinuity points of -180^0 ~ 180^0 representation and 0^0 ~ 360^0 are -180^0 and 0^0 respectively, we divide the longitude input to four categories: (a) [-180^0, 0^0], (b) (-180^0 ~ 180^0), (c) [0^0, 180^0], and (d) (0^0 ~ -0^0). A longitude range of type (a) or (c) utterly belongs to either representations, so given a range of type (a) or (c), it is easy to generate a longitude query for either representations. A longitude range of type (b) completely belongs to -180^0 ~ 180^0, but has to be separated into two ranges (180^0 ~ 360^0] and [0^0 ~ 180^0), so given a range of type (b), we need to generate a query of representation -180^0 ~ 180^0, and two queries, each for the two sub range, of representation 0^0 ~ 360^0. And type (d) is similar to type (c). From the DIMES search interface, the allowed longitude values range from -180^0 to 180^0.

6. CONCLUSIONS

In this paper, we give our approach to apply Lucene and its range queries to the temporal and spatial range queries of Earth Science metadata. The result shows that it is very good and efficient. The testing DIMES server has 2931 metadata, and by Lucene, it takes about one second to complete an individual temporal, spatial or textual query internally. But, Lucene index has its own capacity, which is about 10,000 per index field, so for very large amounts of data, this becomes a constraint. So we expect to develop more suitable search engine toolkits for Earth science data.

ACKNOWLEDGMENTS

The authors would like to thank Jordan Alpert and June Wang of NCEP for their suggestions on improving the system.

REFERENCES

Doty, B. E., J. L. Kinter III, M. Fiorino, D. Hooper, R. Budich, K. Winger, U. Schulzweide, L. Calori, T. Hol, and K. Meier, 1997: "The Grid Analysis and Display System (GrADS): An update for 1997: " 13th Conf. on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, pages 356-358 (American Meteorological Society, Boston).

Apache (The Apache Software Foundation), 2005, "Apache Lucene," http://lucene.apache.org/java/docs/ (last accessed on October 27, 2005).

Olsen, L.M., G. R. Major, "Global Change Master Directory Enhances Search for Earth Science Data," 1996, Transactions of the EOS, American Geophysical Union, http://www.agu.org/eos_elec/95127e.html (last accessed on October 27, 2005).

OPeNDAP, 2004: "What is OPeNDAP?" http://www.opendap.org/ (last accessed on October 27, 2005).

Wielgosz, J., B. E. Doty, J. Gallagher, and D. Holloway, 2001: "GrADS and DODS," 17th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, Jan. 2001.

Yang, R., X. Deng, M. Kafatos, C. Wang and X. Wang, 2001: "An XML-Based Distributed Metadata Server (DIMES) Supporting Earth Science Metadata," in Proceedings of the 13th International Conference on Scientific and Statistical Database Management (L. Kerschberg and M. Kafatos, eds.), pages 251-256, IEEE, Computer Society.

Yang, R., M. Kafatos, and X. Wang, 2002: "Managing Scientific Metadata Using XML," *IEEE Internet Computing,* v6, no.4, pp. 52-59

Yang, R., X. Wang, Y. Nie, Y. Zhao and M. Kafatos, 2003: "A Web-Based Scientific Data and Information Super Server with A Flexible XML Metadata Support," in Proceedings of the 19th International Conference on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, American Meteorological Society*, Long Beach, California, February 9-13, 2003 (CD-ROM).*