Exploring the applicability of Scientific Data Management Tools and Techniques on the Records Management Requirements for the National Archives and Records Administration- Phase 1 October 2002-December 2003

NCSA-NARA E-Mail Collections Research Summary Report

January 20, 2004

The NCSA-NARA project entitled, "*Exploring the applicability of Scientific Data Management Tools and Techniques on the Records Management Requirements for the National Archives and Records Administration*" was comprised of six separate but complimentary components. The research component focusing most thoroughly on the use of e-mail data collections was undertaken by the Automated Learning Group (ALG) team under the direction of Duane Searsmith, Senior Research Programmer. This work applied data mining and machine learning approaches to text analysis.

Upon procuring the ONDCP e-mail collection from NARA two other project teams directed the focus of their research efforts to include the use of the e-mail collection. The Time Series Characterization research team focused the final quarter work on data storage performance investigations using the e-mail collection to benchmark performance on PC clusters. The Pablo Performance Analysis group used the e-mail collection to facilitate the study of the performance of mySQL, a common software application used for data storage and retrieval.

As of the project meeting in early July 2003, the ALG team efforts shifted towards evaluation methods applied to email collections focusing on aspects of email specifically as a *genre* of text collections that may affect the performance of text mining techniques. The summary and details of this work can be found in the technical paper submitted as an attachment to this report.

Specific general comments on all three of these research efforts follow below:

ALG Team – Duane Searsmith

Discussion

Some objectives for this project changed over time as a result of events and discussions from the quarterly meetings. For the first two quarters the work was very much a collaboration working with Joanne Kaczmarek the University of Illinois archivist for electronic records. During this period, the focus was directed toward building a tool that could help to automate some aspects of the archival process applied to email collections. To this end, the OSBI email collection was acquired from the University of Illinois. Our working assumption during this period was that the university archivist could provide domain expertise (in lieu of direct interaction with NARA archivists) and the results of the research would be largely transferable to the university's own explorative needs in this area. During this period, the university archivist acquired not only the OSBI collection for our research, but also made available to us some listserv data sources. She also shepherded our requests through the university administration and negotiated with university counsel on details of confidentiality.

As of the meeting in early July 2003, the project focus shifted towards evaluation of methods applied to email collections and focusing on aspects of email specifically, as a genre of text collection, that affect the performance of text mining techniques. This shift in focus was due largely to feedback from NARA's Director of Research as to the agency's needs and expectations for this project. In this later period collaboration with the university archivist was greatly reduced because the domain expertise was no longer essential to the work being performed.

Objectives

- Extend the D2K/T2K (Data-to-Knowledge/Text-to-Knowledge) data mining toolkit and modules library to process large email and text collections.
- Extend the suite of algorithms available in D2K/T2K for text mining applications of this project.
- Acquire key data sets for evaluation.
- Evaluate supervised text classification methods against email test collections. Explore extensions to standard algorithms specifically for email collections.
- Apply unsupervised text classification methods against the test collections for demonstration of the quality of analyses possible with currently available research tools.

Activities

Activities for this project can be divided into three categories. The first category involves the creation of the necessary infrastructure in D2K/T2K for handling and analyzing email/text collections. The second category involves negotiations for acquisition of email collections as well as figuring out the correct mechanisms for security. The last category involves evaluation of various text classification methods for supervised, unsupervised, and query enhancement methods applied to email collections.

In the last category described, five areas were investigated. First area involved application of classification algorithms to email records with emphasis on weighting strategies across email header and body content. The second area looked into the potential for improving classification performance by including email thread information explicitly. The third area investigated the effects advanced term weighting strategies on classification performance. The fourth area focused on improving query performance through information retrieval techniques involving user feedback. This approach was intended for application against the ONDCP e-mail collection. The fifth and final activity involved the application of T2K clustering algorithms against the collection to illustrate the quality of the types of results achievable using such methods and tools. All of these activities were fully implemented in D2K/T2k and are available to NARA as tools for future research work and development.

Exploring the applicability of Scientific Data Management Tools and Techniques on the Records Management Requirements for the National Archives and Records Administration- Phase 1 October 2002-December 2003

Time Series Characterization Team – Nancy Tran

Email Archival Performance Benchmark on PC Clusters

Due to a lack of 'noteworthy' electronic large-scale archive applications for our I/O time series characterization study and responding to NARA's interest on the email collections released by NARA, we proposed building an email archival benchmark on PC clusters with special focus on data storage performance. Results on the first stage of building this benchmark indicate that memory-mapped I/O outperforms the common buffered I/O when archiving from CD drives to disks, provided there is adequate memory. More details can be found in the ensuing section.

Activities

During the 09-12/03 quarter, we built a preliminary prototype benchmark for archiving email collections supplied by NARA. Focusing on improving IO performance, we designed the prototype with two major capabilities used to support parallel archival from a single CD-RW drive to disks:

- Multiple *archival threads* can be launched in parallel for concurrent reads/writes, overlapping reads from CD drives with writes to disks, and allowing higher CPU utilization especially in multiprocessor PCs.
- Choice for *two access methods*: buffered IO and memory-mapped IO. Buffered I/O, a common approach for data copy, makes at least three memory copy operations for each read-write call – from the input drive to the operating system (OS) cache, from the cache to the archival application buffer, and finally, from the buffer to disks (assuming the same OS cache entry is re-used). In contrast, memory-mapped IO makes a single memory copy operation per read-write call, provided there is sufficient memory to accommodate the operation. We will compare the performance between these two methods. Incorporating them into the same benchmark allows archival experiments be conducted in both memory-deficient and memory-rich systems.

Experimental Environment and Data

We conducted experiments on a PC equipped with a 2.4 GHz Pentium IV uniprocessor, 1 GB of main memory, an 80 GB internal disk, and a CD-RW drive with theoretical peaks of 7200KB/sec for read and write, 3600KB/sec for rewrite, average transfer rate of 5.2 MB/sec, and random seek time of 97 milliseconds. The PC was installed with Red Hat Linux 7.3.

The email collection provided by NARA includes 2 CDs – one has 47 files, with a mix of small (less than 1 MB), medium (from 1 to 35 MB), and large files (from 60 to 132 MB), giving a total of 542 MB; the other CD has 19 files with medium to large file sizes, ranging from less than 3 MB to 89 MB, totaling 448 MB.

We experimented with both buffered IO and memory-mapped IO. For buffered IO, we varied the buffer size from 4 KB to 256 KB. The best size of 8 KB yielded the shortest execution time. Below are results obtained with the 8 KB buffer size.

Results

The workload of the archival benchmark is extremely I/O intensive. Commensurate with file sizes, CPU utilizations varied only from 1 to 4% for buffered IO, and 0.5 to 2.5% for memory mapped IO. Results presented here are averages from five runs for each category.

CD1 (542 MB)	Buffered IO (sec)	Memory-Mapped IO (sec)
1 thread	252	161
2 threads	476	1076

CD2 (448 MB)	Buffered IO (sec)	Memory-Mapped IO (sec)
1 thread	194	110
2 threads	395	646

Buffered IO throughput for *one-thread process* is 2.15 MB/sec for CD1, about 64% of the 3.36 MB/sec throughput achieved via memory-mapped IO. This memory-mapped performance gain is mainly due to a reduction in memory copy operations. One can attain higher throughputs with larger file sizes as shown by the memory-mapped IO performance of around 4 MB/sec on CD2, about twice the throughput of 2.3 MB/sec for buffered IO.

However, archival performance degrades rapidly with *multi-thread processes*. For both buffered and memory-mapped IO, archival times increase rapidly as the number of files increases and the file sizes are small. Here, the CD drive with a single RW head becomes the bottleneck. On the average, read response times increase by at least four folds, from an average of 4 milliseconds (1 thread) to 20 milliseconds (2 threads) -- the archival threads spent most time waiting for input data.

Summary

Our experimental results suggest that improving archival I/O performance for the NARA email collections requires:

- Sufficient available memory to take advantage of memory-mapped I/O. Preferably, the amount of memory should be able to cover the total amount of data on a single CD, minimizing cache eviction.
- A CD drive should be assigned only to a single archive process, minimizing contention to a slow device to obtain better response time for read accesses.

- Archival processes should be able to automatically choose the best access method based memory availability, I/O block size, and the size of the archival program.

As our ultimate goal is to alleviate the problem of bursty data accretion for archives at NARA, we can use the above insights to guide future designs of parallel archival benchmarks for two potential configurations: single PC, multiple CD drives and multiple PCs, multiple CD drives.

Pablo Performance Analysis Team – Dan Wells

The activity of the Pablo portion of the NARA project during the last quarter was the study of the performance of a software application used in data archival and retrieval and study using the e-mail collection as input. During the past quarter, we obtained the public domain database package, ***mySQL***. Pablo instrumentation was added to the code and it was ported to an IA-32 cluster belonging to the Pablo Research Group where it was tested. As a test case, code was written to extract relevant fields from the Unix-format e-mail and build database tables with that information.

Once the ONDCP e-mail was made available on the NCSA IA-32 cluster, the code was ported to that system. The format of the ONDCP e-mail differs from the Unix-format e-mail, so much of the code to extract the data fields had to be rewritten. The message fields of interest are the creator, creation date/time, recipient, subject and e-mail record type; the text of each message is extracted and stored in a regular file.

The tables in the database are the following:

- ***People*** – a list of the distinct people sending or receiving messages. It has the columns
  - *PersonID*, an integer sequence number;
  - *PersonName*, a character string in the message from a line beginning ‘CREATOR:’, ‘TO:’ or ‘CC:’.

- ***Subject*** - a list of the distinct e-mail topics. It has the columns
  - *SubjectID*, an integer sequence number;
  - *SubjectValue*, the character string extracted from the message from the line beginning ‘SUBJECT:’.

- ***RecordType*** - a list of the distinct e-mail record types. It has the columns
  - *RecordTypeID*, an integer sequence number;
  - RecordTypeValue, the character string extracted from the field beginning ‘RECORD TYPE:’ in the message.

- ***MessageTable*** - the main table coordinating the other tables. It contains the columns
  - *MsgID*, a sequence number identifying the message;

  - *CreatorID*, the value of *PersonID* taken from the ***People*** table where *PersonName* is the string extracted from the 'CREATOR:' field of the message;
  - *SubjectID*, the value of *SubjectID* extracted from the ***Subject*** table where *SubjectValue* is the string extracted from the 'SUBJECT:' field of the message;
  - *RecordTypeID*, the value of *RecordTypeID* extracted from the ***RecordType*** table where *RecordTypeValue* is the string extracted from the 'RECORD TYPE:' field of the message;
  - *Time*, the time the message was sent as extracted from the 'CREATION DATE/TIME:' field of the message converted into the format *yyyy-mm-dd hh:mm:ss*;
  - *TextFileName*, the character string containing the path name of the file where the full text of the message is stored.

- ***MsgRecipients*** – table associating the messages with its recipients listed in the 'TO:' or 'CC:' field of the messages. It contains the colums
  - *MsgID*, the value of the sequence number for this message. It is the same as *MsgID* taken from the table ***MessageTable***;
  - *RecipientID*, the value of *PersonID* taken from the ***People*** table where *PersonName* equals the string extracted from a 'TO:' or 'CC:' field of the message,
  - *SeqID*, a sequence number unique to the row entry.

At this time, all of the individual messages have been extracted and stored in a directory, the tables have all been created with the ***Subject***, ***People*** and ***RecordType*** tables fully populated. Data is still being inserted into the ***MessageTable*** and ***MsgRecipients*** table. When all of the data is entered we will begin experimenting with various queries with and without indexing, measuring the amount of I/O and time required to perform these operations.