Intelligence Community Massive Digital Data Systems Initiative

Tatu Ylonen 23-Nov-1995 03:00

Posted in group: **mail.cypherpunks**

Below is some information about the Intelligence Community Massive Digital Data Systems Initiative.

Summary:
- new data 2 - 5 terabytes (10^12 bytes) per day
- total size about 20 petabytes (20 * 10^15 bytes)
- 300 terabytes on-line, the rest accessible in a few minutes
- funding (for the research initiative, not for the final system):
  3-5 million USD per year estimated for investments

Now, how much is 2 - 5 terabytes per day?
- 20 - 50.000.000 jpeg images (100kB/image, relatively high-quality) per day
- 20 - 50.000.000 minutes of GSM-quality phone intercepts per day
- 1.000.000 - 2.500.000 minutes of compressed (256kbit/sec) video per day
- 1.000.000.000 - 3.000.000.000 e-mail messages per day
- you can continue the list; most available data sets turn out to be much smaller

How much is 20 petabytes? Assuming you want to collect information about 100.000.000 people worldwide, this makes 200 megabytes per person (on the average for each of those 100 million people).

200 megabytes per person on the average is quite a lot, since for many of those people you probably don't have all that much data. Maybe 90% of the data for 10% of the people?

(Of course, in a database like this you might also have a lot of data like aerial imaginery, satellite imaginery, economical information, etc., so it is a little exaggarated to talk about all of it being on individual people.)

The full text is below.

Crypto relevance? Makes you think whether you should protect your data.

Tatu

From: dbo...@cs.wisc.edu (Dbowner)
To: b...@mitre.org, mi...@nobozo.CS.Berkeley.EDU, sho...@csr.lbl.gov, gr...@sfbay.enet.dec.com, li...@cs.wisc.edu, ragr...@almaden.ibm.com, man...@gte.com, hei...@gte.com, da...@hplabs.hpl.hp.com, sh...@hplabs.hpl.hp.com, to...@almaden.ibm.com, rei...@ksr.com, j...@allegra.att.com, ra...@allspice.berkeley.edu, mcl...@vaxa.isi.edu, ni...@MIMSY.CS.UMD.EDU, a...@purdue.edu, la...@ccr-p.ida.org, dar...@watson.ibm.com, gros...@math.uic.edu, db...@cs.wisc.edu, meta...@llnl.gov, jma...@mosaic.uncc.edu, w...@thumper.bellcore.com

Cc: su...@mitre.org, con...@mitre.org
Subject: Call For Papers MDDS
Date: Thu, 18 Nov 93 11:08:03 EST
Resent-To: dbworld...@cs.wisc.edu
Comments: IF YOU REPLY TO THIS MESSAGE, BE SURE TO EDIT THE to: AND cc: LISTS.
 The dbworld alias reaches many people, and should only be used for
 messages of general interest to the database community. Mail sent
 to dbedu goes to the subset of addresses with a .edu suffix; mail
 sent to dbusa goes to the subset of US addresses. Please use
 the smaller lists when appropriate. Requests to
 get on or off dbworld should go to dbworld...@cs.wisc.edu.
Reply-To: (Susan L. Hanlon) <su...@linus.mitre.org>
Resent-Reply-To: (Susan L. Hanlon) <su...@linus.mitre.org>

3 November 1993

Dear Colleague:

Subject: Call for Abstracts for Massive Digital Data Systems

Future intelligence systems must effectively manage massive amounts of digital data (i.e., multi-terabytes or greater). Issues such as scalability, design, and integration need to be addressed to realize a wide spectrum of intelligence systems ranging from centralized terabyte and petabyte systems comprised of many large objects (e.g., images) to distributed heterogeneous databases that contain many small and large objects (e.g., text). The Community Management Staff's Massive Digital Data Systems (MDDS) Working Group on behalf of the intelligence community, is sponsoring a two day invitation-only unclassified workshop on the data management of massive digital data systems with government, industry, and academia.

The workshop will be held on the 1st and 2nd of February 1994 in Reston, Virginia. The objective of the workshop is to make industry and academia aware of intelligence community needs, stimulate discussion of the technical issues and possible solutions, and identify potential research efforts that warrant further investigation for possible government funding. The amount of funding estimated for investments is three to five million dollars per year over the next 2-3 years.

Last July, a one-day, classified, government-only workshop was held to characterize the magnitude of the problem and identify the major challenges. The needs, issues, and in some cases, lessons learned, were presented for different data types including Imagery, Text, Voice, Video, and Multi-media. Enclosure 1, "Massive Digital Data System Issues", is an unclassified description of the consolidated challenges.

The Massive Digital Data Systems Working Group is soliciting one-page abstracts related to the issues of the data management of massive digital systems including (but not limited to) scalability, architecture and data models, and database management functions. The focus of the abstract should be on potential solutions for the longer term research challenges (i.e., 5-10 years out) that must be addressed today in order to effectively manage data of massive proportions in the future. The solutions need not be limited to proven approaches today but can foster new approaches and paradigms. Issues relating

to the storage media and analysis tools, while important to the intelligence community, are not within the scope of the workshop. Selection for attendance will be based upon technical relevance, clarity, and quality of the proposed solution.

Each one-page abstract should follow the abstract format enclosed (Enclosure 2). All submissions must be UNCLASSIFIED. To allow enough time for proper evaluation of each abstract, the deadline for submission is 01 December 1993. You will be notified of acceptance to attend by 17 December 1993. Abstracts should be forwarded to one of the following:

Jackie Booth, P.O. Box 9146, Rosslyn Station, Arlington, VA 22219
Jackie Booth, ORD/SETA, fax number (703) 351-2629
boo...@mcl.saic.com (Internet)

Please pass this call for abstracts on to other colleagues that are working on solutions in this area.

Sincerely,

Dr. David Charvonia
Director, Advanced Technology Office
Community Management Staff

Enclosures:
1. Massive Digital Data Systems Issues
2. Abstract Format

Enclosure 2
ABSTRACT FORMAT

Title:

Author(s):

Organization/Affiliation:

Address:

Phone: FAX:

Description:

Status: (Research, Prototype, Operational)

Scope: (Size of effort in terms of dollars and/or staff months; Size of system in terms of amount of data, number of databases, nodes, users, etc.)

Customer: (if applicable)

Operational Use: (if applicable)

****

Forward to one of the following:

Jackie Booth, P.O. Box 9146, Rosslyn Station, Arlington, VA 22219
Jackie Booth, ORD/SETA, fax number (703) 351-2629
boo...@mcl.saic.com (Internet)

MASSIVE DIGITAL DATA SYSTEMS ISSUES

EXECUTIVE SUMMARY

Future intelligence systems must effectively manage massive amounts of digital data (i.e., multi-terabytes or greater). Issues such as scalability, design, and integration need to be addressed to realize a wide spectrum of intelligence systems ranging from centralized terabyte and petabyte systems comprising many large objects (e.g. images) to distributed heterogeneous databases that contain many small and large objects (e.g. text). Consequently, Massive Digital Data Systems (MDDS) are needed to store, retrieve, and manage this data for the intelligence community (IC). While several advances have been made in database management technology, the complexity and the size of the database as well as the unique needs of the IC require the development of novel approaches. This paper identifies a set of data management issues for MDDS. In particular, discussions of the scalability issues, architectural and data modeling issues, and functional issues are given. The architectures for MDDS could be centralized, distributed, parallel, or federated. The functions of MDDS include query processing, browsing, transaction management, metadata management, multimedia data processing, integrity maintenance, and realtime data processing. Representing complex data structures, developing appropriate architectures, indexing multimedia data, optimizing queries, maintaining caches, minimizing secondary storage access and communications costs, enforcing integrity constraints, meeting realtime constraints, enforcing concurrency control, recovery, and backup mechanisms, and integrating heterogeneous schemas, are some of the complex tasks for massive database management. The issues identified in this paper will provide the basis for stimulating efforts in massive database management for the IC.

1.0 INTRODUCTION

1.1 The Challenge

The IC is challenged to store, retrieve, and manage massive amounts of digital information. Massive Digital Data Systems (MDDS), which range from centralized terabyte and petabyte systems containing many large objects (e.g., images) to distributed heterogeneous databases that contain many small and large objects (e.g., open source), are needed to manage this information. Although technologies for storage, processing, and transmission are rapidly advancing to support centralized and distributed database applications, more research is still needed to handle massive databases efficiently. This paper describes issues on data management for MDDS including scalability, architecture, data models, and database management functions. Issues related to storage media, analysis tools, and security while important to the IC are not within the scope

of this paper.

The key set of data management issues for MDDS include:

- Developing architectures for managing massive databases
- Utilizing data models for representing the complex data structures
- Formulating and optimizing queries
- Developing techniques for concurrency control and recovery
- Integrating heterogeneous schemas
- Meeting timing constraints for queries and transactions
- Indexing multimedia data
- Maintaining caches and minimizing secondary storage access and communications costs
- Enforcing integrity constraints.

1.2 Background

The IC provides analysis on current intelligence priorities for policy makers based upon new and historical data collected from intelligence sources and open sources (e.g., news wire services, magazines). Not only are activities becoming more complex, but changing demands require that the IC process different types as well as larger volumes of data. Factors contributing to the increase in volume include continuing improvements in collection capabilities, more worldwide information, and open sources. At the same time, the IC is faced with decreasing resources, less time to respond, shifting priorities, and wider variety of interests. Consequently, the IC is taking a proactive role in stimulating research in the efficient management of massive databases and ensuring that IC requirements can be incorporated or adapted into commercial products. Because the challenges are not unique to any one agency, the Community Management Staff (CMS) has commissioned a Massive Digital Data Systems Working Group to address the needs and to identify and evaluate possible solutions.

1.3 Assumptions and Project Requirements

Future intelligence systems must provide a full suite of services for gathering, storing, processing, integrating, retrieving, distributing, manipulating, sharing and presenting intelligence data. The information to be shared is massive including multimedia data such as documents, graphics, video, and audio. It is desired that the systems be adapted to handle new data types.

The goal is to be able to retain the data for potential future analysis in a cost effective manner. The more relevant data would remain on-line, say for 5 years, organized with the most relevant data accessible in the least amount of time. It is expected that 2 to 5 terabytes of new data has to be processed each day. Thus, the total size of the database (both on-line and off-line) could be as large as 20 petabyes with about 300 terabytes of data stored on-line. It is assumed that storage devices (primary, secondary, and even tertiary) for the large multimedia databases as well as data pathways with the required capacity will exist. The access times are about 5 seconds for the data less than a week old, about 30 seconds for data under two months old, and on the order of minutes for data up to 10 years old.

2.0 SCALABILITY ISSUES

A particular data management approach can be scaled to manage larger and larger databases. That is, a database can often sustain a certain amount of growth before it becomes too large for a particular approach. For example, more memory,

storage, and processors could be added, a new hardware platform or an operating system could be adopted, or a different microprocessor could be used (e.g. using a 32 bit microprocessor instead of a 16 bit microprocessor). Once the size of the database has achieved its limit with a particular approach, then a new approach is required. This new approach could be a new architecture, a new data model, or new algorithms to implement one or more of the functions of the database management system (DBMS), or a combination of these features. Discussions of these three features are given below.

Architectures: The type of architecture impacts the size and response time of the DBMS.

Centralized approaches are being migrated to distributed and parallel approaches to handle large databases. Some architectures such as a the shared nothing parallel architectures are scalable to thousands of processors, but will have multiprocessor communication issues. Current approaches need to be assessed to determine their scalability limits. New approaches may be required for handling massive databases.

Data Models: Data models which support a rich set of constructs are desired for next generation database applications. However, the search and access time of the DBMS would depend on the data model used. For example DBMSs which support complex data structures use large caches, access data through pointers, and work well with large main memory in general, while DBMSs based on simpler data models maintain index files and provide associative access to the secondary storage. The limits of these models within the context of massive databases need to be understood. New or modified approaches may be required.

DBMS Functions: The techniques to implement the DBMS functions have to be modified to handle massive databases. For example, as the size of the database increases, new approaches for query optimization, concurrency control, recovery, and backup, access methods and indexing, and metadata management will be required.

The architectural, data modeling, and functional issues that need to be addressed for MDDS will be elaborated in sections 3 and 4.

3.0 ARCHITECTURAL AND DATA MODELING ISSUES

3.1 Architectural Issues

This section describes some of the architectural issues that need to be addressed for an MDDS. In the case of the centralized approach, a major issue is managing the data transfer between the main memory and secondary storage. One could expect the data that is a week old to be cached in main memory, the data that is less than two months old to be in secondary memory, and data that is a few years old to be in tertiary storage. In designing the data management techniques (such as those for querying, updating, and transaction processing), data transfer between the main and secondary memories needs to be minimized. There is also a need to reflect patterns of use (e.g., in migrating items to lower/higher levels of storage hierarchy). Another issue is the relationship between the size of the cache and the size of the database.

When one migrates to distributed and parallel architectures, a goal is to maintain a larger number of smaller databases. It is assumed that processors and storage devices are available. A major issue is the communication between the

processors. In designing the data management mechanisms, an objective would be to minimize the communication between the different processors. For example, in the case of a join operation between several relations in a relational DBMS, each fragmented across multiple sites, an issue is whether to merge all of the fragments of a relation and then perform the join operation or whether to do several join operations between the fragments and then merge the results to form the final result. Different configurations of the distributed and parallel architectures also need to be examined. For example, there could be point-to-point communication between every processor, or the processors could be arranged in clusters and communication between clusters is carried out by designated processors. Another issue in migrating to a distributed architecture is handling data distribution. For example, if the data model is relational, then how could one fragment the various relations across the different sites? If the relations are to be replicated for availability, then how could consistency of the replicated copies be maintained? Another issue is what data could be cached within the distributed system, how could data be cached, and for how long could the cache be maintained.

While distributed and parallel architectures are being investigated for managing massive databases, federated architectures are needed to integrate the existing different and disparate databases. The existing databases could be massive centralized databases or they could be distributed databases. Furthermore, they could be relational, object-oriented and even legacy systems. An issue in heterogeneous database integration is developing standard uniform interfaces which can be accessed via an integration backplane. If the environment is a federated one, where the nodes have some autonomy, then a major issue is the ability to share each other's data while maintaining the autonomy of the individual DBMSs. This is hard because cooperation and autonomy are conflicting goals. The techniques to implement the DBMS functions for data retrieval, updates, and maintaining integrity have to be adapted or new approaches have to be developed for federated architectures.

Extensible architectures are also being investigated for massive databases. With such architectures, DBMSs are extended with inferencing modules which make deductions from data already in the database. This way, one need not store all of the data in the database explicitly. Instead, appropriate inference rules are used to make deductions and derive new data. This way the size of the database is reduced. The issues include determining what data is to be stored in the database and what data is to be stored in the knowledge base manipulated by the inferencing module, effective management of the knowledge base, and adapting the functions of the DBMS to handle extensible architectures.

3.2 Data Modeling Issues

In selecting an appropriate data model for massive databases, several issues must be considered. Providing a data model powerful enough to support the representation of complex data must be addressed. For example, with a multimedia document, one may need to devise a scheme to represent the entire document in such a way to facilitate browsing and updating. Since the age of a document could be used to move it between different storage media, it is desirable for the data model to support the representation of temporal constructs. The representation of different types of multimedia devices and grouping of documents are also important considerations in selecting a data

model. The data model chosen has an impact on the techniques to implement the functions of a DBMS. For example, DBMSs based on some models use associative access while those based on some other models use pointer traversal.

In migrating to a distributed/parallel architecture, if it is assumed that the data model is the same for all databases, then a major issue is whether it is feasible to provide a conceptual view of the entire massive database to the user. However, in the case of a federated architecture, since it is generally assumed that the individual data models are different, several additional issues need to be considered. For example, could the users have a global view of the massive database or could they have their own individual views? In either case, it would be desirable for the users to access the distributed databases in a transparent manner. If a global view is enforced, the query processor could transform the queries on the global view to the views of the individual databases. If each user has his own view, then the query processor could transform the users view into the views of the individual databases. Other issues for a federated architecture include the representation of the individual schemas (which describe the data in the databases), determining which schemas to be exported to the federation, filtering appropriate information from the schemas at different echelons, integrating the schemas to provide a global view, and generating the external schemas for the users. In integrating the different schemas, the semantic and syntactic inconsistencies between the different representations need to be resolved. For example, the address in database A could include the house number and the street name while in database B it could just be the city and the state.

4.0 FUNCTIONAL ISSUES

The techniques to implement the functions of MDDS will be impacted by the architectures and data models as well as requirements such as integrity and multimedia data processing. Therefore some of the functional issues have already been addressed in section 3. This section provides a more detailed overview of the functional issues. First the basic functional issues for MDDS (such as issues on query processing and transaction management) will be discussed and then the impact of maintaining integrity, realtime processing, and multimedia data processing will be given.

4.1 Querying, Browsing, and Filtering

The query operation is a means by which users can retrieve data from the database. Closely related is the browsing operation where users traverse various links and subsequently scan multiple documents either sequentially or concurrently. To determine if the new information warrants viewing by the analyst and/or to enforce access control, automatic filtering of the data is needed. Some issues in query management for massive databases are using an appropriate language for specifying queries and developing optimization techniques for the various operations involved in a query. The goals are to make it easier for users to formulate queries and also to minimize data transfer between primary and secondary storage.

Query management in a federated environment must provide the means for formulating and processing queries seamlessly and efficiently. This involves designing an interface for formulating queries over multiple sources. There is a need for query optimization, in order to prevent degradation in performance in the distributed system. In addition to determining the execution strategy for a query, query optimization techniques could also determine which portion of the

query processing is to remain under direct and unshared control at the analyst's workstation. Methods need to be developed for browsing the integrated information space and for displaying results obtained from multiple sources. Finally, data from local databases have to be filtered according to the various constraints (such as security constraints) and enforced before sending it to the remote sites.

Query processing algorithms in an extensible architecture need to incorporate inferencing techniques. The usefulness of inferencing techniques for intelligence applications can best be illustrated with a simple example. Suppose parts A, B, C and D are needed to build a nuclear weapon, and also suppose that the following constraint is enforced: " if three of the four parts are shipped to country X, then the fourth part should not be shipped to X." Therefore, if parts A, B, and C are already shipped to X and there is a request from X for part D, then the inferencing module will determine that this part cannot be shipped. An issue in developing an inference module is determining the deduction strategies to be implemented. These strategies could be just logical deduction or could include more sophisticated techniques such as reasoning under uncertainty and inductive inference. With most inference strategies one runs into the problem of an infinite loop; therefore appropriate time limits must be enforced to control the computation.

In general, the issues to be addressed in query management will include:

- Query optimization.
- Handling data distribution
- Making intelligent deductions
- Uniform vs. user-tailored query language

4.2 Update Transaction Processing

Multi-user updates are supported in general to improve performance. The goal is for multiple users to be able to update the database concurrently. A major issue here is ensuring that the consistency of the database is maintained. The techniques that ensure consistency are concurrency control techniques. Often update requests are issued as part of transactions. A transaction is a program unit that must be executed in its entirety or not executed at all. Therefore, if the transaction aborts due to some error, such as system failure, then the database is recovered to a consistent state.

Several concurrency control algorithms have been designed and developed for different environments. Some algorithms are suitable for short transactions in business processing applications and some others are suitable for long transactions which often involve multimedia data. To handle long transactions efficiently, weaker forms of consistency conditions have been formulated. Several recovery techniques have also been developed to maintain the consistency of the database. If the transaction is long, then the log files that record the actions of the transaction may be quite large. Efficient management of log files becomes an issue. As the size of the database increases, a transaction would take a longer time for execution. Adapting the concurrency control and recovery algorithms or developing new algorithms to work with the massive databases becomes an issue.

Update transaction processing gets more complicated in distributed and federated environments. For example, if replicated copies are to be maintained, then

making them consistent will have an impact on the performance. Therefore, an issue here is whether to maintain strict consistency or select a subset of the copies and make them consistent immediately so that the remaining copies could be updated at a later time. One of the problems with a federated environment is the different concurrency control and recovery algorithms used by the individual DBMSs. In such a situation synchronizing the different techniques becomes a major issue.

4.3 Access Methods and Index Strategies

To enhance the performance of query and update algorithms, efficient access methods and index strategies have to be enforced. That is, in generating strategies for executing query and update requests, the access methods and index strategies that are used need to be taken into consideration. The access methods used to access the database would depend on the indexing methods. Therefore creating and maintaining appropriate index files is a major issue in a DBMS. Usually, the size of the index file grows with the size of the database. In some cases, the index file could be larger than the database itself. Some of the issues include determining what type of indexes are to be maintained for massive databases. Is it feasible to have dense indexing where there is an entry in the index file for every entry in the database? If so, the index file could have as many entries as there are in the database. Is it better to have sparse indexing so that the size of the index file could be reduced? If so, is there a strategy to determine which entries in the database are to be indexed? For multimedia data, indexing could be done not only by content but by type, language, context (i.e., where, how, when it was collected), author (i.e., for documents), and speaker (i.e., for voice). The challenge is how to index and to provide improved mechanisms for extraction of the information used for indexing. For example, the ability to automatically index voice is desired. Additionally, the ability to index voice and video (with associated voice) with their transcriptions (i.e., time alignment) is necessary.

Various storage structures have been proposed. These include B-Trees and Parent-Child links. The question is, are these methods suitable for massive databases? Voice and video data require segmentation into logical units for storage and access. Additionally, the ability for automatic segmentation within documents of embedded drawings and figures and their interpretation (via seamless integration with image handling tools) is needed. Other challenges include providing user transparent hierarchical storage management (i.e., store the most relevant or most recent information on the fastest media) and the ability to reposition data in the storage hierarchy based upon changing importance, migration mechanisms for transferring information to newer storage media or a new architecture (failure to do so can lead to exorbitant costs to maintain discontinued storage media drives or inaccessible data), archival technology/policies for older/less important information, and synchronization of information distributed across multiple repositories

Compression can decrease the costs of storage and transmission especially for the larger objects such as vector and raster spatial data types, voice, imagery, and video. Real-time conversion of heterogeneous voice and video compression and file formats in network broadcasts/multicasts is an issue. For imagery, a capability such as pyramidal decomposition for providing reduced resolution images is needed for browsing purposes.

4.4 Managing the Metadata

The metadata includes a description of the data in the database (also referred to as the schemas), the index strategies and access methods used, the integrity mechanisms enforced, and other information for administrative purposes. Metadata management functions include representing, querying, and updating, the metadata. In massive databases, if the metadatabase is much smaller than the database, then the traditional techniques could be applied to manage the metadata. If the metadatabase becomes massive, then new techniques need to be developed. An issue here is whether the techniques for massive databases could be applied for massive metadatabases also. Support for schema evolution is desired in many new generation applications. For example, the structures of the entities in the database could change with time. An entity could acquire new attributes or existing attributes could be deleted. The metadata needs to be represented in a manner that would facilitate schema evolution. That is, appropriate models to represent the metadata are desired. Since the metadata has to be accessed for all of the functions of a DBMS, the module that is responsible for accessing the metadata needs to communicate with all the other modules. Efficient implementation of this module is necessary to avoid performance bottlenecks.

Certain types of metadata, such as the schemas, are usually accessible to the external users. An issue here is whether to provide a view to the users that is different from the system's view of the metadata. For example, a different representation of the metadata could be sued for the users. Also, if the metadatabase is massive, then subsets of it could be presented to the users.

4.5 Integrity

Concurrency control and recovery issues discussed in section 4.3 are some of the issues that need to be dealt with in order to maintain the integrity (i.e. consistency) of the database. Other types of integrity include maintaining the referential integrity of entities and enforcing application dependent integrity constraints. Referential integrity mechanisms must ensure that the entities referenced exist. The question is, how could the references to an entity be deleted when an entity itself is deleted? If the databases are massive, then there will probably be more references to the deleted entity. Deleting all these references in a timely manner is an issue.

In the case of application specific integrity constraints, they could trigger a series of updates when one or more items in the database gets updated. Again, as the size of the database increases, the number of updates that are triggered could also increase. The issue here is ensuring that the updates are carried out in a timely manner.

4.6 Realtime / Near Realtime Processing

Within a massive digital data system, the challenges of realtime or near realtime processing will be compounded. For realtime or near realtime applications, timing constraints may be enforced on the transactions and/or the queries. In the case of a hard realtime environment, meeting the timing constraints may cause the integrity of the data to suffer. In the case of soft realtime constraints (also referred to as near real-time), there is greater flexibility in meeting the deadlines. The issues for real-time processing include:

If a transaction misses its deadline, then what are the actions that could be taken?

Could a value function be associated with a transaction which can be used to determine whether the transaction should continue after it misses its deadline?

Could the transaction be aborted if the value of the data approaches zero?

What is the impact on the scheduling algorithms when timing constraints are present?

How can the techniques be extended for a distributed/federated architecture?

In the case of realtime updates in a distributed replicated environment, is it possible to maintain the consistency of the replicated copies and still meet the timing constraints?

What is the impact on the techniques for multimedia data processing?

4.7 Multimedia Data Processing

By nature, multimedia data management has to deal with many of the requirements for indexing, browsing, retrieving, and updating of the individual media types. Implementing multimedia data types will require new paradigms for representing, storing, processing, accessing, manipulating, visualizing, and displaying data from various sources in different media. One of the major issues here is synchronizing the display of different media types such as voice and video. Other issues include selecting/developing appropriate data models for representing the multimedia data and developing appropriate indexing techniques such as maintaining indexes on textual, voice, and video patterns. For example, the ability to index voice and video simultaneously may be desired.

In addition to the manipulation of multimedia data, frameworks for the integration of multimedia objects as well as handling different granularity of multimedia objects (i.e., 1 hour video clip versus a spreadsheet cell) need to be considered. A flexible environment has to be provided so that the linked and embedded distributed multimedia objects can accommodate geographic/network changes. Finally, the data manipulation techniques as well as the frameworks need to be extensible to support new and diverse data types.

4.8 Backup and Recovery

On-line backup procedures are being used for massive databases. This is because off-line procedures will consume too much time for massive databases. Even if the backup procedures are carried out on-line, the system could be slowed down and therefore the performance of other data management functions would suffer. The issue here is to develop improved techniques for backup so that it will not impact functions such as querying, browsing, and updating.

Recovery issues for transaction management were discussed in section 4.3. Other recovery issues include whether to maintain multiple copies of the database, and if so, the number of copies to be maintained, and whether the checkpointing, roll-back and recovery procedures proposed for traditional databases could be used for massive databases or is there a need to develop special mechanisms?

5.0 SUMMARY

Massive digital data systems will require effective management, retrieval, and integration of databases which are possibly heterogeneous in nature. Achieving this concept of massive intelligence information systems will require new technologies and novel approaches for data management. While hardware is rapidly advancing to provide massive data storage, processing, and transmission, the software necessary for the retrieval, integration, and management of data remains an enormous challenge.

This paper has identified a set of issues for managing the data in massive digital data systems with a focus on intelligence applications. First, an overview of the current approaches to data management and the scalability of the current approaches were discussed Then some architectural and data modeling issues were given. Finally, a discussion of the issues for the various functions of MDDS were given. The set of issues identified is by no means considered a complete list. As the progression of research, prototyping, and deployments continue, new or hidden challenges will arise.

From owner-...@cs.wisc.edu Thu May 11 12:35:05 1995
Date: Thu, 11 May 95 17:35:05 -0500
Reply-To: nwo...@harpo.wh.att.com
Sender: owner-...@cs.wisc.edu
Precedence: bulk
From: nwo...@harpo.wh.att.com (Kingsley Nwosu)
To: Multiple recipients of list <dbw...@cs.wisc.edu>
Subject: (DBWORLD) Massive Digital Data Systems
X-To: dbw...@quarg.cs.wisc.edu
X-Listprocessor-Version: 7.1 -- ListProcessor by CREN

Birds of a Feather Session on the
Intelligence Community Initiative in
Massive Digital Data Systems

To be held in Conjunction with the 1995 ACM SIGMOD International Conference on Management of Data

in the Crystal Room at The Fairmont Hotel, San Jose, California
on Tuesday, 23 May 1995 from 7 pm to 9 pm

****

Massive Digital Data Systems (MDDS) will require effective management, retrieval, and integration of databases that may be heterogeneous. Achieving this concept of massive intelligence information systems will require new technologies and novel approaches for data management. While several advances have been made in database management technology, the complexity and the size of the database coupled with the unique needs of the Intelligence Community (IC) require the development of novel approaches. The MDDS Initiative has been started by the Community Management Staff (CMS) of the IC to identify the data management issues and challenges as well as to develop possible solutions for managing massive databases.,

The purpose of the Birds of a Feather Session is to provide an overview of the MDDS program, solicit input toward developing a testbed /framework for the initiative, and to have technical discussions on novel methods for managing massive databases. Tentative agenda for the session is as follows:

1. Overview of MDDS Initiative
2. Overview of MDDS Research Projects
3. Discussion on testbed/framework
4. Discussion on techniques for massive database management

Information on the MDDS Initiative can be obtained from :

http://www.nml.org/other_programs/mdds/mdds.html

For more information contact:
Dr. Bhavani Thuraisingham
K329, The MITRE Corporation, Burlington Road, Bedford, MA 01730
email: th...@mitre.org; Phone: 617-271-8873; Fax: 617-271-2352

From owner-...@cs.wisc.edu Tue May 16 13:51:40 1995
Date: Tue, 16 May 95 18:51:40 -0500
Reply-To: Bhavani_Th...@qmgateib.mitre.org
Sender: owner-...@cs.wisc.edu
Precedence: bulk
From: "Bhavani Thuraisingham" <Bhavani_Th...@qmgateib.mitre.org>
To: Multiple recipients of list <dbw...@cs.wisc.edu>
Subject: (DBWORLD) MDDS Session Time Change
X-To: dbw...@quarg.cs.wisc.edu
X-Listprocessor-Version: 7.1 -- ListProcessor by CREN

5/15/95 5:17 PM
MDDS Session Time Change

Birds of a Feather Session on the
Intelligence Community Initiative in
Massive Digital Data Systems

Held in Conjunction with the 1995 ACM SIGMOD International Conference on
Management of Data

in the Crystal Room at The Fairmont Hotel, San Jose, California
on Tuesday, 23 May 1995 from 7:30 pm to 9:30 pm

****

Massive Digital Data Systems (MDDS) will require effective management,
retrieval, and integration of databases that may be heterogeneous. Achieving

this concept of massive intelligence information systems will require new technologies and novel approaches for data management. While several advances have been made in database management technology, the complexity and the size of the database coupled with the unique needs of the Intelligence Community (IC) require the development of novel approaches. The MDDS Initiative has been started by the Community Management Staff (CMS) of the IC to identify the data management issues and challenges as well as to develop possible solutions for managing massive databases.,

The purpose of the Birds of a Feather Session is to provide an overview of the MDDS program, solicit input toward developing a testbed /framework for the initiative, and to have technical discussions on novel methods for managing massive databases. Tentative agenda for the session is as follows:

1. Overview of MDDS Initiative
2. Overview of MDDS Research Projects
3. Discussion on testbed/framework
4. Discussion on techniques for massive database management

Information on the MDDS Initiative can be obtained from :
http://www.nml.org/other_programs/mdds/mdds.html

For more information contact:
Dr. Bhavani Thuraisingham
K329, The MITRE Corporation, Burlington Road, Bedford, MA 01730
email: th...@mitre.org; Phone: 617-271-8873; Fax: 617-271-2352

----------FOOTER-