Scalable Exploratory Data Mining of Distributed Geoscientific Data

Eddie C. Shek[†‡], Richard R. Muntz[†], Edmond Mesrobian[†], and Kenneth Ng[†]

Computer Science Department[†]
University of California
Los Angeles, CA 90024

Information Sciences Laboratory[‡]
Hughes Research Laboratories
Malibu, CA 90265

Abstract

Geoscience studies produce data from various observations, experiments, and simulations at an enormous rate. Exploratory data mining extracts "content information" from massive geoscientific datasets to extract knowledge and provide a compact summary of the dataset. In this paper, we discuss how database query processing and distributed object management techniques can be used to facilitate geoscientific data mining and analysis. Some special requirements of large scale geoscientific data mining that are addressed include geoscientific data modeling, parallel query processing, and heterogeneous distributed data access.

Introduction

A tremendous amount of raw spatio-temporal data is generated as a result of various observations, experiments, and model simulations. For example, NASA EOS expects to produce over 1 TByte of raw data and scientific data products per day by the year 2000, and a 100-year UCLA AGCM simulation (Mechoso *et al.* 1991) running at a resolution of $1° \times 1.25°$ with 57 levels generates approximately 30 TBytes of data when the model's output is written out to the database every 12 simulated hours.

In geoscience studies, a scientist often wants to extract interesting geoscientific phenomena that are not directly observed from the raw datasets. The time-varying location of phenomena reduces the number of variables in the data space while their semantic interpretation makes it more natural for the scientist to hypothesize that there might be some meaning to the classification problem, for example, based on these variables. For example, cyclone tracks, which are the trajectories traveled by low-pressure areas over time, can be extracted from a sea-level pressure dataset by linking observed areas of local pressure minima at successive time steps. Modeled as time-series of polygonal cells on the earth surface, these tracks can be used as content-based indexes that allow efficient access to "interesting" regions in a large geophysical dataset.

There are obvious similarities between geoscientific feature extraction and data mining in business applications (Agrawal & Srikant 1995) (e.g., study stock market trends by correlating the price movements of selected stocks). They both involve sieving through large volumes of isolated events and data to locate salient (spatio-)temporal patterns in the data.

The patterns of interest for business data mining applications are generally simpler and are formed by lists or sets of alpha-numeric data items. On the other hand, a geoscientific feature such as a cyclone track is a complex spatio-temporal object that is derived from massive spatio-temporal datasets through a series of computationally expensive algorithms. Geoscientific feature extraction algorithms are often dependent on complex spatio-temporal definitions of the phenomenon of interest. Scientific data mining is further complicated by the fact that scientists often do not agree on the precise definition of a natural phenomenon, leading them to develop similar but incompatible mining routines.

We cast geoscientific data mining as a database query processing problem in order to take advantage of established automatic query optimization and parallelization techniques to deliver high performance to geoscientific data mining applications. In addition to supporting high performance parallel processing, a query processing system has to support an expressive spatio-temporal data model in order for it to properly handle the diversity and complexity of geoscientific data types.

Motivated by the requirements of geoscientific data mining applications, we are developing an extensible parallel geoscientific query processing system called *Conquest* (Shek, Mesrobian, & Muntz 1996). In this paper, we describe the design of Conquest, concentrating on the features that make it especially suitable for geoscientific data mining, specifically geoscientific data modeling, parallel query processing, and heterogeneous distributed data access. Then we present our experiences with using Conquest in a real-life geoscientific data mining application in which the upward propagation of planetary-scale waves affecting the formation of ozone holes are studied.

Geoscientific Data Modeling

A large scale geoscientific data analysis application often involves the processing and handling of a large variety of spatio-temporal geoscientific data, ranging from multi-dimensional arrays of floating point numbers (e.g., a sea-level pressure dataset) to time series of georeferenced points (e.g., cyclone tracks), and traditional alpha-numeric data.

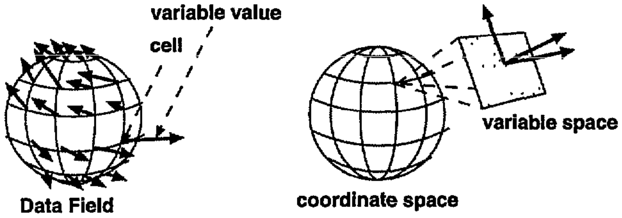

Figure 1: Example Geographic Data Field

A recurring characteristic of these data is that measurements of scientific parameters are recorded over a multi-dimensional, and often spatio-temporal, space. As a result, the central idea of the Conquest data model is that of the *field* (see Figure 1) which associates parameter values with *cells* in a multi-dimensional *coordinate space*. Cells can be of various geometric object types of different dimensionality including points, lines, polygons, and volumes. The type of the cells and hence the coordinate space they lie in are determined by *coordinate attributes*.

Values recorded for cells lie in a multi-dimensional *variable space*. The types of values that can be associated with a cell in the coordinate space of a field is dictated by *variable attributes*. The data type associated with a cell is not limited to simple data types; it can belong to a complex spatio-temporal data type or even be another field.

We refer to a cell and the variable value associated with it as a *cell record*. Not all cells in the coordinate space of a field are associated with variable values. We define a field's *cell coverage* (or simply coverage) to be the set of distinct (but not necessarily disjoint) cells in its coordinate space for which variable values are recorded. Given its cell coverage, the field's *cell mapping* maps each cell in its cell coverage to a value. The cell coverage and cell mapping logically define a field.

Some important semantic properties of data fields are captured in the Conquest data model. For example, the *extent* within the coordinate space in which cells lie and the *regularity* in which cells lie in the coverage strongly influence the choice of storage and index structures. Moreover, cell records in some fields (e.g., time series) have a natural *ordering* which dictate their access patterns.

Geoscientific Algebraic Operators

The Conquest data model defines an algebra which allows algebraic transformation techniques similar to that used in many relational DBMSs to be applied to geoscientific queries. The algebra contains a base set of general purpose logical field data manipulation operators, while users are allowed to introduce operators corresponding to application-specific algorithms. Conquest allows scientists to conveniently express their intentions by functionally combining complex scientific data manipulation operators within the algebra framework. The set of base logical operators supported can be roughly divided into the following classes:

- **Set-Oriented Relational Operators.** We define selection, projection, cartesian product, union, intersection, difference, and join operators similar to their counterparts in the relational algebra. While the logical schema for the result of these operators is well-defined, the resulting field often does not inherit the semantic properties of the input(s). For example, selecting cells in a field based on their variable values (e.g., cells in a regular sea-level pressure field which recorded parameter value greater than 980mb) in general returns a field whose cell coverage is unstructured.
- **Sequence-Oriented Operators.** Many geoscientific data mining applications involve studying the change of time-varying parameters. For example, given a set of cyclone track fields represented as time series of polygonal cyclone extents, we may want to find all cyclone trajectories whose spatial extent shrinks for 3 consecutive days. As a result, we introduce a number of sequence-oriented operators which generate fields by consuming cell records from input fields in sequence, modifying an internal state in the process, and output cell records of the output field.
- **Grouping Operators.** Data analysis applications often involve evaluating aggregate information on collections of related data from a field. We provide several cell record collection operators for collecting related cell records into subfields in preparation for aggregation. The grouping operator associates with each cell in a field's coverage a subfield containing all cells in a *neighborhood*. A nested field is defined as a field in which the values associated with cells in the coverage are fields. The nest operator moves selected coordinate attributes of a field into the variable space. Each cell in the coverage of the resulting field is associated with a field whose coordinate space is composed of the migrated attributes. Nesting a field causes "related" cell records in the original field to be grouped in a cell record, in the resulting field. The unnest operator has the inverse effect of the nest operator.
- **Space Conversion Operators.** We define operators that support the conversion of the format and

representation of field data so that differences between data fields from different sources can be reconciliated and then meaningfully compared and correlated. The sample operator derives variable values at a user-specified set of cells in the coordinate space of a field with an interpolation function. By imposing a regular grid on a field, sampling (or gridding) can present a structured view of the data by deriving variable values at regular grid points through interpolation. In addition, a field's cells and their variable values can be changed by applying a mapping function to each cell record. Coordinate attribute mapping can be used to convert one map projection to another, or ‘move" cells relative to their current positions by translation or rotation. One use of variable attribute conversion is to perform aggregation on related variable values, after they are collected by grouping operators.

Physical Data Model

A data field is structured in Conquest as a data stream. Conquest uses the cell record as the unit of data passing between physical operators, making it possible to take advantage of the Conquest grouping operators (group, nest and unnest) as a unique mechanism to control the granularity of data communication.

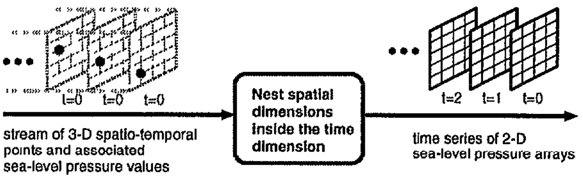
t=0 t=0 t=0
stream of 3-D spatio-temporal points and associated sea-level pressure values
Nest spatial dimensions inside the time dimension
t=2 t=1 t=0
time series of 2-D sea-level pressure arrays

Figure 2: Using the Nesting of a Data Field to Control the Granularity of Data Communication between Conquest Operators

For example, given a regular 3-D floating-point array measuring sea-level pressure on regularly spaced locations on the surface of the earth at regular time intervals, a cell record is a 4-tuple containing the spatio-temporal location and the floating point value recorded at the point. Nesting the spatial coordinate dimensions inside the time dimension causes the same array to be logically viewed as a time series of 2-D spatial array each storing the sea-level pressure values recorded on the earth surface at the corresponding time (see Figure 2). This allows cell records in the array to be implicitly referenced and hence significantly reduces the overhead required to explicitly represent the coordinate of each cell.

Extensible Parallel Query Execution

Parallelization techniques are commonly used to remove bottlenecks in I/O and computation and improve query performance. In particular, Conquest supports pipeline processing, partitioning, and multicasting to improve query performance.

Pipeline processing (or dataflow processing) supports "vertical" inter-operator parallelism in which two connecting operators in a query execution plans are assigned to different processors so that execution of the operators can overlap. Each operator consumes data arriving through a stream from its producer and feeds its output to an output stream until it blocks (e.g., when the stream buffer is full). In addition to its demonstrated effectiveness for traditional set-oriented queries, pipeline parallelism naturally supports stream query processing techniques which take advantage of data ordering to deliver excellent performance for many sequence- or set-oriented scientific queries. The benefit of stream processing is especially obvious when a scientific query is coupled with a visualization routine which consumes query results as they are being generated, allowing visualization to effectively overlap with query evaluation.

Intra-operator parallelization (or partitioned parallelism) is another form of parallelism. It provides opportunities for performance improvement by spreading I/O and computation across multiple processors or storage servers. It is achieved by dividing an input stream or dataset among a set of independent operators, each responsible for processing or retrieving a fragment of the data. In Conquest, a query execution plan fragment can be evaluated by a set of Conquest processes in a process group, each responsible for evaluating the query execution plan fragment on a portion of cell records in the input data stream.

Conquest also allows a data stream to multicast to multiple consumer process groups to provide additional opportunities for I/O and computation to be optimized. In addition, the multicast operator explicitly controls data flow to avoid data being sent too fast from a producer operator and flooding the system when the consumer operators fails to keep pace.

Automatic Query Parallelization

Extensibility is an important requirement of a geoscientific information system. One of the major implications of extensibility to query optimization is that the search space of query execution plans has to be extended as user-defined operators are introduced. As a result, to perform automatic parallelization, the optimizer in an extensible query processing environment has to be able to answer the questions of *whether* an operator in a query execution plan can be parallelized, and if so, *how* it can be parallelized.

The basic approach to achieve intra-operator parallelism for a unary stream operator is to partition the input stream into substreams, each of which is assigned to a copy of the stream operator. To simplify the discussion, we assume that the partitioning is based on time ranges. In order words, each processor is assigned a fragment of the logical input stream and is responwhile convenient, accessing data from distributed objects eliminates opportunities to take advantage of the query capability of data repositories to optimize query evaluation. Some database servers and scientific data format libraries efficiently support some data manipulation and filtering operations. Most notably, indexes can be defined to provide alternative access paths to data and to filtered out unnecessary data internally without having them to be translated for external consumption. In addition, many problems do not fit the stream paradigm (e.g., slab multi-dimensional subarray extraction), and fit better into the storage management subsystem rather than the query execution engine (Graefe 1993). As a result, it is often advantageous to optimize extraction of data from external data sources by pushing operations and filters into the data source to take advantage of efficient processing and reduce the amount of data that needs to be extracted out of the data source.

A description of our proposed approach to optimize access to heterogeneous datasets by taking advantage of the fact that some repositories can efficiently execute some operations can be found in (Shek, Mesrobian, & Muntz 1996). In short, by consulting the data dictionary, a reference to a distributed object in a query execution plan may be mapped to a collection of scan operators to the underlying data repositories for the object. A set of *operator ingestion rules* guide how operators in a query expression can be "pushed" into logical scan operators for execution by the corresponding repository.

Data sources supported by Conquest include files in popular scientific data formats such as HDF (Nat 1993) containing multi-dimensional raster datasets, and extended relational DBMS Postgres which is used as both a storage and an external content-based index server.

Implementations and Experiences

Conquest has been ported to run on massively parallel processor supercomputers (IBM SP1, SP2 and Intel Paragon) as well as workstation farms using the portable message passing library PVM as the inter-process communication mechanism. It has been in use for the past two years at UCLA and JPL for exploratory data analysis and data mining of spatio-temporal phenomena produced by the UCLA and ECMWF Atmospheric General Circulation Models (AGCMs) and satellite-based sensor data such as NCAR's ECMWF Global Basic Surface and Upper Air Advanced Analyses. Previously reported geoscientific data mining activities include the extraction and analysis of cyclonic activity, blocking features, and oceanic warm pools (Stolorz *et al.* 1995).

Upward Energy Propagation

The upward propagation of planetary-scale waves from the troposphere into the stratosphere has a profound effect on the structure of the stratospheric circulation. Occasionally, the rapid growth and upward propagation of waves during winter in the northern hemisphere can lead to a reversal of the high-latitude stratospheric wind from westerly (i.e., west to east) to easterly. On longer time scales, the weaker upward propagation of the planetary waves in the southern hemisphere leads to a stronger westerly winds than in the northern hemisphere. This results in the formation of a well-defined "ozone hole" each spring over Antartica, while no such ozone hole develops in the Arctic.

To detect upward propagation of wave energy into the stratosphere, we might first compute a measure of the phase difference of a particular component (e.g., zonal wave number 1, the wave with the longest wavelength), at a given latitude, between two pressure levels in the upper troposphere (e.g., 50mb and 500mb levels). Next we locate waves of sufficient strength (amplitude) at the two neighboring pressure levels by computing the first Fourier coefficient of the geopotential height data values measured at these pressure levels.

We implemented the query as a series of Conquest operators which computation can be partitioned along the time dimension for parallel evaluation, i.e., the input datasets can be divided into (equal-size) pieces and processed in parallel. This partitioning is driven by the fact that the window of relevance of the query is instantaneous because no information from an earlier period is needed in order to extract upward wave propagation event at a particular time.

We have performed the study on 3 HDF-based geopotential height datasets on a 4-node Sun workstation network (consisting of SparcStation 10s and SparcStation 20s): a NCAR ECMWF Upper Air Advanced Analyses dataset (14 geopotential levels, 2.5° lat. × 2.5° lon. × 12 hours resolution, from 1985-1994, 2Gbyte), a CSIRO AMIP dataset (6 levels, 3.184° lat. × 5.625° lon. × 6 hours resolution, from 1979-1986, 370Mbyte), and a UCLA AGCM dataset (6 levels, 4° lat. × 5° lon. × 12 hours resolution, from 1980-1989, 330Mbyte). 7304 instances of upward wave propagation events are extracted from the largest NCAR ECMWF Analyses dataset in 8610 seconds with 1 node and in 2430 seconds on 4 nodes. The speedup is not perfect mainly because of the non-even distribution of upward wave propagation events over time (see Figure 3) and that of the computing resources on the heterogeneous collection of computing nodes.

After independent upward wave energy propagation events are extracted, trajectories of such events that persisted for more than 1 day are located. Figure 4 shows the number of upward wave propagation trajectories between 500mb and 50mb levels from the CSIRO AMIP dataset at different latitudes, demonstrating that the frequency of upward wave propagation trajectories decreases as it approaches the equator.

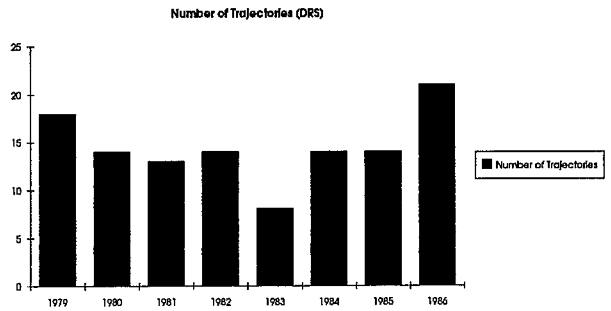

Figure 3: Number of upward wave propagation trajectories between 500mb and 50mb levels extracted from the CSIRO AMIP dataset per year

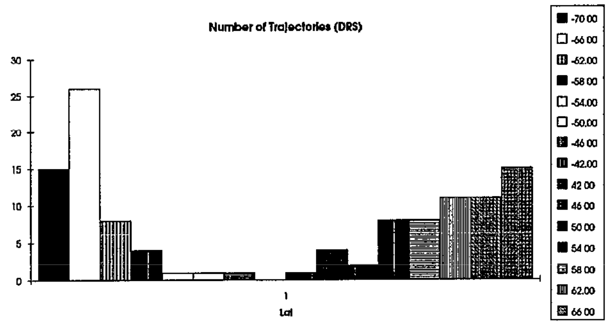

Figure 4: Number of upward wave propagation trajectories between 500mb and 50mb levels at different latitudes extracted from the CSIRO AMIP dataset

Conclusions

Conquest defines a geoscientific data model, and applies distributed and parallel database query processing techniques to handle computationally expensive data mining queries on massive distributed geoscientific datasets. The usefulness of Conquest as a data mining system is demonstrated in a upward energy propagation study in which Gbytes of raw data are digested and summarized into less than 1 Mbyte of energy propagation trajectory information (representing a size reduction of 4 orders of magnitude) which help scientists gain insight into the process of energy propagation and ozone hole formation.

Query optimization in Conquest emphasizes parallelization and optimized data access. This is becasue we realized that the benefit of algebraic transformation (Wolniewicz & Graefe 1993) is limited due to the application-specific nature of scientific operators. Furthermore, it is unclear what the effects of algebraic query expression transformations are on the accuracy of query results since many scientific operators are very sensitive to the accuracy and precision of its inputs; small round-off errors introduced at one point in a query execution plan may snowball as data flows through multiple operators and cause significant error in the result.

OASIS (Mesrobian *et al.* 1996) is a complementary effort to Conquest at UCLA that aims to develop a flexible environment for scientific data analysis, knowledge discovery, visualization, and collaboration. It provides application developers, as well as end-users, the logical abstraction that the environment is simply a set of objects. While the core OASIS services, implemented in Sunsoft's CORBA-compliant NEO, provide users with transparent access to heterogeneous distributed objects without regards for their underlying storage and representation, they do not immediately support parallel processing of data retrieved from these objects. As a result, we are currently reimplementing Conquest as the OASIS distributed query service to exploit distributed object computing technologies to support complex geoscientific query processing.

References

Agrawal, R., and Srikant, R. 1995. Mining sequential patterns. In *Proc. 11th Int'l Conf. on Data Engineering*.

Graefe, G. 1993. Query evaluation techniques for large databases. *ACM Computing Surveys* 25(2):73–170.

Mechoso, C.; Ma, C.; Farrara, J.; and Spahr, J. 1991. Simulations of interannual variability with a coupled atmosphere-ocean general circulation model. In *Proceedings of 5th Conference on Climate Variations.* American Meteorology Society.

Mesrobian, E.; Muntz, R. R.; Shek, E. C.; Nittel, S.; LaRouche, M.; and Kriguer, M. 1996. OASIS: An open architecture scientific information system. In *Proc. of Sixth International Workshop on Research Issues in Data Engineering: Interoperability of Nontraditional Database Systems*, 107–116.

National Center for Supercomputing Applications. 1993. *HDF User's Guide, Version 3.2.*

Shek, E. C.; Mesrobian, E.; and Muntz, R. R. 1996. On heterogeneous distributed geoscientific query processing. In *Proc. of Sixth International Workshop on Research Issues in Data Engineering: Interoperability of Nontraditional Database Systems*, 98–106.

Soley, R. M., ed. 1992. *Object Management Architecture Guide (2nd Edition).* Object Management Group.

Stolorz, P.; Mesrobian, E.; Muntz, R. R.; Shek, E. C.; Santos, J. R.; Yi, J.; Ng, K.; Chien, S. Y.; Nakamura, H.; Mechoso, C. R.; and Farrara, J. D. 1995. Fast spatio-temporal data mining of large geophysical datasets. In *Proc. of First International Conference on Knowledge Discovery and Data Mining*, 300–305.

Wolniewicz, R. H., and Graefe, G. 1993. Algebraic optimization of computations over scientific databases. In *Proc. 19th Int'l Conf. on VLDB.*