QUEST: Content-based Access to Geophysical Databases

E. Mesrobian, R. R. Muntz and E. C. Shek
UCLA Computer Science Dept.
Los Angeles, CA 90024
edmond@cs.ucla.edu
muntz@cs.ucla.edu
shek@cs.ucla.edu

C. R. Mechoso
J. D. Farrara
UCLA Atmospheric Sciences Dept.
Los Angeles, CA 90024
mechoso@atmos.ucla.edu
farrara@atmos.ucla.edu

P. Stolorz
Jet Propulsion Lab
CalTech
Pasadena, CA 91109
stolorz@jpl.nasa.gov

Abstract

A major challenge facing geophysical science today is the unavailability of high-level analysis tools with which to study the massive amount of data produced by sensors or long simulations of climate models. As part of a NASA HPCC Grand Challenge effort [Mun92], we have developed a prototype environment called QUEST to provide content-based query access to massive datasets used in geophysical applications. QUEST employs work stations as well as massively parallel processors to produce spatio-temporal features that are used as high-level indexes into terabyte datasets. This paper discusses our continued development of the QUEST environment.

1 Introduction

A critical challenge facing geophysical science today is the unavailability of high-level analysis tools with which to study the massive amount of information captured by sensors onboard orbiting satellites or produced by climate models. To address this challenge, we must develop a new generation of systems for scientific data management capable of coping with

- the highly complex multimodal queries and intelligent retrievals required for scientific investigations and knowledge discovery, and
- the staggering computational demands posed by the extraordinary size of the data set and the complexity of the tasks involved.

Both facets of this challenge are present in queries involving detection of changes and flow structures that are common in atmospheric and geophysical science. Thus, the recognition of phenomena such as monsoons, extratropical cyclones, cold and warm fronts, ocean currents and ocean eddies is rich both in data complexity (large multidimensional data sets) and logical complexity (difficulty of detecting and tracking patterns evolving in a multidimensional space). These applications must be supported by (1) large parallel systems capable of providing fast access to very large data sets on mass storage and (2) scientific information management systems capable of taming the complexities of the logical task involved.

Scientific data management systems capable of supporting successful geophysical investigations on data products generated by sensors onboard satellites or produced by models must address a number of key issues:

Conceptual abstraction Typically, scientists reason about geophysical variables in terms of trends, correlations with other variables, and/or relationships with phenomena such as extratropical cyclones. These abstractions are quite removed from the raw, low-level data products provided by model or satellite data (e.g., temperature). Traditionally, systems provide users with the ability to interact directly with the raw data via queries enabling users to specify spatio-temporal windows over their raw data. This approach is quite limited, when one considers that a dataset might be from 100 gigabytes to 5 terabytes. How does one locate the salient information? A system must also enable content-based access to data subsets based on high-level "features" extracted from the original raw data. These features are more amenable for local storage (typically less than 5% of the size of the raw data), and provide salient indexes to the large geophysical datasets.

Exploatory analysis The nature of exploratory data analysis for scientific hypothesis testing or phenomenon detection is basically an iterative, successive-refinement process. The scientist initially applies a coarse model on the data, and then uses the outcome of this first experiment to refine his/her model and methods; then the process is repeated until the hypothesis is dropped or it is refined into one that is fully corroborated by the collected data. For such investigations to be practical, the scientist must have at hand a powerful system that supports: (1) the easy formulation of powerful queries and discriminant decision rules against the database; (2) a natural representation of the relationships of the scientific domain of interest (e.g, in natural domains of the space and time, but, possibly, in the frequency domain as well); and (3) efficient execution of these queries without requiring the scientist to become cognizant of the storage structures and processing strategies involved.

Content based access to image databases has become an active area of research in recent years, but is still in its infancy. A sampling of recent work can be found in [1, 2]. The QUBIC project at IBM [3] is an example system that illustrates the state-of-the-art in image retrieval by content, while examples of work in the area of geoscience databases include VIMSYS [4] and Sequoia 2000 [5].

As part of a NASA HPCC Grand Challenge effort [Mun92], we have developed a prototype environment called QUEST to provide content-based query access to massive datasets used in geophysical applications. QUEST employs workstations as well as massively parallel processors to produce spatio-temporal features that are used as high-level indexes into terabyte datasets. Our first application area is the output of global change climate models (GCM). In the initial prototype, we extracted model-simulated trajectories of cyclones and anticyclones for content-based access to GCM data [6]. Through the use of query panels (forms), scientists select the subset of features to use as indexes into the database. The chosen features can then be displayed textually (via a browser) or visualized. The scientist can then form a request to retrieve the values of model variables based on the spatio-temporal characteristics of the chosen feature. The variables can be animated, and/or subsetted for further investigation.

This paper presents our continued development of the QUEST environment. To overcome the limitations imposed by the use of attribute-based queries, we have developed a novel data model and algebraic language to handle geoscientific data and have prototyped a parallel query execution engine for the language. In addition, we have started the initial design of the QUEST Scientist Workbench, a visual dataflow programming environment for specifying scientific queries.

2 Geophysical datasets

Geophysical datasets are generally produced by either observational systems (e.g., satellites) or models. Earth science phenomena, modeled or observed, typically contain features which can be extracted from their datasets. These spatio-temporal phenomena and their derived features can be managed, manipulated, and indexed by applications according to their spatial and temporal properties.

We chose the output of AGCMs as our application domain for two principal reasons: (1) it includes a challenging set of spatial-temporal patterns (e.g., cyclones, hurricanes, fronts, and blocking events); and (2) it is generally free of incomplete, noisy, or contradicting information. Hence it serves as an ideal testbed for validating our prototype environment.

The specific data set used in this study was generated by the The UCLA atmospheric general circulation model (AGCM) [7], a finite-difference model that includes sophisticated parameterizations of cumulus convection [8, 9], and planetary boundary layer processes [10, 11], as well as parameterizations of shortwave radiative transfer [12], and longwave radiative [13]. The horizontal structure of the model is typically represented by grid cells of various resolution; we are using a grid size of $5°$ longitude and $4°$ of latitude. The vertical component of the model is represented by a series of pressure layers; the version in this study has 9 layers in the vertical with the top at 50 millibars.

The prognostic variables of the AGCM are horizontal velocities, potential temperature, water vapor and ozone mixing ratio, surface pressure, ground temperature, and the depth of the planetary boundary layer. There are also diagnostic variables such as vertical velocities, precipitation, cloudiness, surface fluxes of sensible and latent heat, surface wind stress and radiative heating. Typically, the model's output is written out to the database at 12-hour (simulation time) intervals; however, this frequency can be modified depending on storage capacity of the database. the model can be run with different spatial resolutions (grid sizes) and temporal resolution (output frequency). At the lowest spatial resolution ($4° \times 5°$, 9 levels) with 12 hour output interval, the AGCM produces approximately 5 Gbytes of data per simulated year, while a 100-year simulation of a AGCM with a $1° \times 1.25°$, 57 levels) generates approximately 30 terabytes of output.

3 System Architecture

Figure 1 presents a functional description of QUEST environment for the efficient processing of scientific data analysis queries. The environment is composed of a Query Manager, a Query Execution Server, a a Query, the Quest Scientific Workbench, a Visualization Manager, and an Information repository.

A scientist interacts with the Quest Scientist workbench (Figure 2) to specify analysis queries of AGCM data stored on tertiary storage or of AGCM data products being produced by the AGCM model in real time [14]. The dataflow diagram is converted in a high-level, SQL-based query language by the Quest Scientist Workbench. The query is then sent to the Quest Query Manager, which first parses the query to produce a logical query tree in the Field Query Language, and then sends the query tree to the Quest Optimizer. The optimizer, possibly transforms the query tree operator sequence, and generates a query execution plan in the Field Query Execution Language. The Quest Query Execution Server, running either on a SUN workstation farm or on an IBM SP1, executes the query and produces results with are sent to the Visualization Manager for display.

3.1 Quest Field Data Model and Algebraic Language

We have developed a data model and an algebraic language designed to handle geoscientific data which is significantly different from traditional data models and their languages. Scientific data often come in the form of data fields in which data values are recorded at points of a multi-dimensional spatio-temporal index space. Generally, one may think of the coordinates of the index space as independent variables while the values associated with points in the index space are dependent variables.

A *field space* is formed by attaching a variable from a domain V called the *variable space* to each point in an independent space called the *index space*. The index space contains the set of points with which a variable value can be associated. A *field* is constructed by assigning a value in the variable space to each point in a subspace of the index space, called the *sampled index space*. The obvious reason that variable values are not assigned to all points in the index space is that the index space may be infinite and the finiteness of storage necessitates storing only values for a "sampled" set of points.

The basic operators supported in our initial prototype can be roughly separated into the following classes:

- Relational-like operators which mirror the nested relational algebra.
- Grouping operators which collect related tuples in a field for further processing.
- Resampling operators which allow the resampling and subsetting of fields.
- Component space conversion operators which derive new fields with different component space using complex scientific data processing algorithms (e.g., FFT, principal component analysis, tracking).
- Sequenced field manipulation operators.

3.2 Quest Scientific Workbench

We plan to prototype a visual dataflow programming environment (based on the field data model described below) that allows scientists to conveniently manipulate, analyze, and visualize different types of scientific data. The dataflow processing paradigm has been adopted by many popular scientific data analysis and visualization systems (e.g., AVS, IDL, Data Explorer, and Khoros). Why not use these systems? Existing systems lack a number of critical capabilities: (1) query (dataflow) optimization, (2) access to data objects residing in a local DBMS, (3) the ability to pipeline (stream) data products through the operator sequence; instead of allowing only a single operator is active at a given time, creating serious I/O bottleneck problems for large datasets, and (4) data flow processing over a network of workstations or on massively parallel computers. Our prototype will provide a data flow language and execution facility which addresses these deficiencies.

As an example scenario we offer the following. Consider a scientist whose current goal is to study the relationship between cloud formation, sea surface temperature, and atmospheric moisture by tracking the motion of a "warm pool" of water in the Pacific Ocean (see Figure 3). The clouds can act like a thermostat, so that when the water gets too warm, the clouds might increase and shut off the solar radiation. It may turn

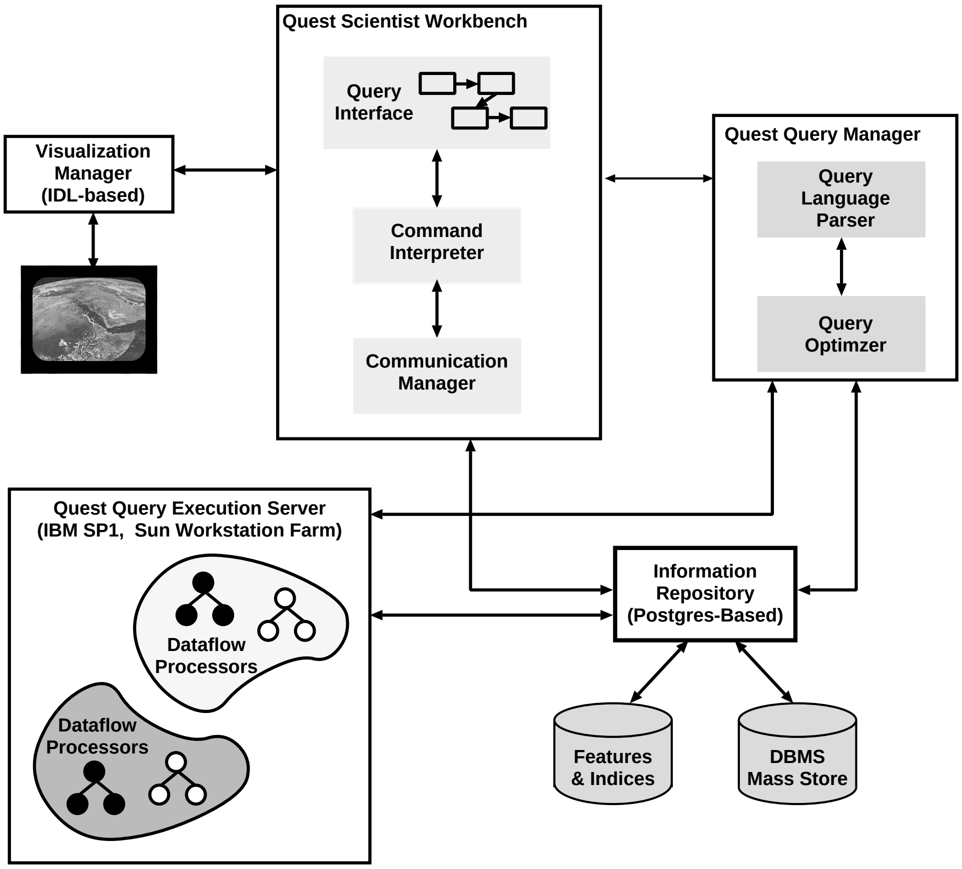

Figure 1: QUEST Environment

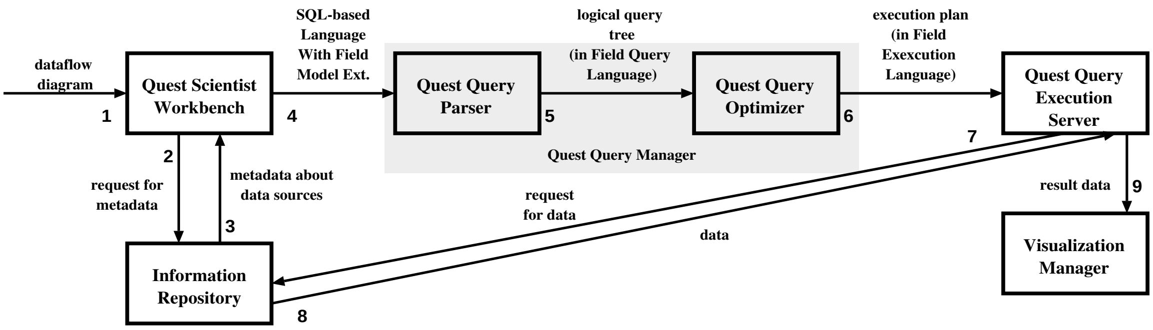

Figure 2: Data flow in the system in response to a user query.

out that the sea surface temperature gradients associated with the warm pool are important because this behavior does not seem to be apparent in other "warm pools" in other oceans where the major difference is in such a gradient. Tracks of warm pools are visualized, and data is examined by binning as a function of sea surface temperature. Statistics (e.g., average and standard deviation per bin) need to be compiled regarding cloud amount, cloud height, atmospheric temperature, and atmospheric water.

3.3 Quest Query Manager

Given the size and complexity of geoscientific data, query optimization plays an extremely important role in a scientific data analysis environment. We are developing a query optimizer for our environment. Algebraic query transformation techniques similar to those employed in relational database systems can be used for field query optimization. The field query language provides a macro level control structure that directs functions to be applied to chunks of the larger dataset. The language allows specification in terms of a set operations on chunks of data. Operations on those chunks are software modules that are atomic building blocks of the field query processor. However, properties of these functions can be important to the optimizer. For example, knowing commutativity and associativity properties of operations or characteristics, such as the appropriate degree of parallelism as a function of the input datasets.

Algebraic manipulation of expressions will be used to generate alternative evaluation plans. Using heuristic search rules, and cost estimates, the optimizer searches for the plan with the lowest cost and executes it. A key criterion is the organization of the physical data, since we desire to exploit advantages offered by stream processing techniques for complex queries involving large amounts of data. Furthermore, the query optimizer will have to take advantage of the availability of content-based indexes that have been extracted from the scientific datasets.

3.4 Quest Query Execution Server

We have prototyped a dataflow query processor for the field model on massively parallel processors (e.g. IBM SP1) for data arrays and simple data fields. The prototype execution engine is tailored for MPPs (operator suites are bundled into the engine), and supports the use of functions/operators provided by scientists. Query execution can be carried out on an MPP or a workstation farm depending on the computational complexity of the query and the availability of MPP time.

4 Discussion and Conclusions

Currently, scientists have access to large repositories of observational and model data, but lack high level analysis tools with which to investigate the data. Rather than being able to issue queries such as "Show me motion sequences that include hurricane (cyclone formation or temperature gradient patterns) that developed in the Southern Hemisphere", scientists are forced to issue queries like "Retrieve all temperature records produced between January 1, 1980 and December 25, 1991".

In this paper, we discussed our continued development of a prototype system called QUEST to provide content-based query access to massive datasets. To date, we have prototyped our novel Field data model and algebraic language for scientific queries. To test the expressive power of the language, we implemented a cyclone tracking algorithm. The algorithm was executed by the Quest Execution Server on both a network of SUNs and a 24-node IBM SP1. Preliminary results suggest that the performance of the execution server and the overhead introduced by the Field query language is quite small (10%) as compared to a standalone cyclone tracking program. Much work remains to be performed. We have started the design of Quest Scientist Workbench for expressing queries as visual dataflow programs. The SQL-based query language with Field data model extensions also remains to be developed.

Acknowledgements

We sincerely acknowledge support from NASA HPCC grants #NAG 5-2224 and #NAG 5-2225. We thank C.C. Ma for valuable discussions on cyclogenesis. We also thank Paul Hoffman of the UCLA Office of Academic Computing for his help and computer time on UCLA's IBM SP1.

References

[1] In *Visual Database Systems*. North Holland, 1992.

[2] S-K. Chang and A. Hsu. Image information systems: Where do we go from here? *IEEE*

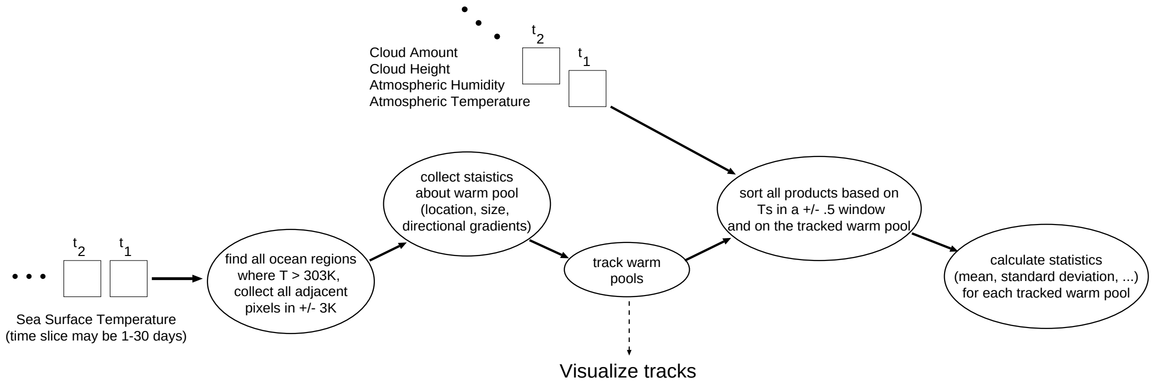

Figure 3: Dataflow diagram for recipe studying relationship between cloud formation, sea surface temperature, and atmospheric moisture.

Transactions on Knowledge and Data Engineering, 4(5):431–442, 1992.

[3] W. Niblack, R. Barber, and W. et. al. Equitz. The qubic project: Querying images by content using color, texture, and shape. Technical Report Research Report #RJ 9203, IBM Research Division, Feb. 1993.

[4] A. Gupta, T. Weymouth, and R. Jain. Semantic queries with pictures: The vymsys model. In *Proceedings of VLDB, Barcelona, Spain*, pages 69–79, Sept. 1991.

[5] A. Guptill and M. Stonebraker. The sequoia 2000 approach to managing large spatial object databases. In *Proc. 5th Int'l. Symposium on Spatial Data Handling, Charleston, S.C.*, pages 642–651, Aug. 1992.

[6] E. Mesrobian, R. R. Muntz, J. R. Santos, E. C. Shek, C. R. Mechoso, J. D. Farrara, and P. Stolorz. Extracting Spatio-Temporal Patterns from Geoscience Datasets. In *IEEE Workshop on Visualization and Machine Vision, Seattle, Washington*. IEEE Computer Society, June 1994.

[7] C. R. Mechoso, S. W. Lyons, and J. A. Spahr. The impact of sea surface temperature anomalies on the rainfall over northeast brazil. *J. Clim.*, 3:812–826, 1990.

[8] A. Arakawa and W. H. Schubert. Interaction of a cumulus cloud ensemble with the large-scale environment, part i. *J. Atmos. Sci.*, 31:674–701, 1974.

[9] S. J. Lord, W. C. Chao, and A. Arakawa. Interaction of a cumulus cloud ensemble with the large-scale environment. part iv: The discrete model. *J. Atmos. Sci.*, 39:104–113, 1982.

[10] J. W. Deardorff. Parameterization of the planetary boundary layer for use in general circulation models. *Mon. Wea. Rev.*, 100:93–106, 1972.

[11] M. Suarez, A. Arakawa, and D. Randall. *Mon. Wea. Rev.*, 111:2224–2243, 1983.

[12] A. Katayama. A simplified scheme for computing radiative transfer in the troposphere. numerical simulation of weather and climate. Technical Report 6, Dept. of Meteorology, University of California, Los Angeles, 1972.

[13] Harshvardhan, R. Davies, D. A. Randall, and T. G. Corsetti. A fast radiation parameterization for atmospheric circulation models. *J. Geophys. Res.*, 92:1009–1016, 1987.

[14] E. Mesrobian, R. R. Muntz, E. C. Shek, C. R. Mechoso, J. D. Farrara, and P. Stolorz. Real Time Data Mining, Management, and Visualization of GCM Output. In *Submitted to Supercomputing '94*, 1994.