Presented with a live demo at:
GridDemo: Live demonstrations of European and US Grid technology. Workshop at
HPCN Europe 2001, Amsterdam, June 25-27, 2001

Grid Services for Fast Retrieval on Large Multidimensional Databases

Peter Baumann[1]

Active Knowledge GmbH
Kirchenstr. 34, D-81675 Munich, Germany
Dial-up: voice +49-89-458677-30, fax -39, mobile +49-173-5644078
E-mail: peter.baumann@active-knowledge.com

Abstract. Fast, user-centric access to and evaluation of large supercomputing results is a well-recognised problem among compute and data service providers.

In the ESTEDI initiative, research and industry from Europe and beyond co-operate to overcome this obstacle. The basic approach is to augment the high-volume data generators with a database system for the management and extraction of spatio-temporal data. Led by ERCOFTAC (European Research Community on Flow, Turbulence and Combustion), a detailed requirements analysis has been carried out which forms the basis for the specification of a High-Performance Computing (HPC) database platform. Implementation of this platform relies on the multidimensional database system RasDaMan. Evaluation covers all major application fields by HPC centres with in-depth experience in their resp. field.

We introduce the ESTEDI project and give an overview of the data management platform under development.

1 Introduction

Satellites and other sensors, supercomputer simulations, and experiments in science and engineering all generate arrays of some dimensionality, spatial extent, and cell semantics. While such data differ in aspects such as data density (from dense 2-D images to highly sparse high-dimensional accelerator data) and data distribution, they usually share the property of extreme data volumes, both per data item and in quantity of data items. Usually, user access nowadays is accomplished in terms of files containing (part of) the information required, encoded in a sometimes more, sometimes less standardised data exchange format chosen from a rich variety of options. This implies several shortcomings.

[1] Research supported by the European Commission under grant no. IST-11009.

First, access is done on an inappropriate semantic level. Applications accessing HPC data have to deal with directories, file names, and data formats instead of accessing spatio-temporal data in terms of, say, simulation space-time and other user-oriented terms.

Second, data access is inefficient. Data are stored according to their generation process, for example in time slices, as opposed to a retrieval-driven organisation. All access pertaining to different criteria, such as spatial coordinates, requires data-intensive extraction processes and, hence, suffers from severe performance penalties.

Third, search across a multitude of data sets is hard to support. Evaluation of search criteria usually requires networks transfer of each (large) candidate data item to the client, implying a prohibitively immense amount of data to be shipped. Hence, many interesting and important evaluations currently are impossible.

All the aforementioned access efficiency problems are substantially intensified as the Grid community grows, as in the absence of optimisation methods obviously networks load grows linearly with the number of users.

In summary, a major bottleneck today is fast, user-centric access to and evaluation of supercomputing results.

Recently the ESTEDI project[2] (*European Spatio-Temporal Data Infrastructure for High-Performance Computing*) has set out to comprehensively collect requirements for HPC data management and to develop a standard architecture and best-practice knowledge for the efficient combination of HPC data processing and management, including fast and flexible retrieval, which is accepted as a broad consensus among the European HPC community. Among the various data structures on hand, ESTEDI focuses on n-dimensional raster data, so-called *Multidimensional Discrete Data* (MDD), as they comprise a central information category encountered in supercomputing.

The envisaged common data management platform is currently under implementation and will be thoroughly evaluated through in-practice application in all major HPC application fields. An existing multidimensional database management system (DBMS), RasDaMan[3] [1, 2, 3, 7], is used at the core, as RasDaMan already offers several important features.

In the remainder of this contribution, we will first outline the ESTEDI project (Section 2) and then give a brief overview of the RasDaMan DBMS in the context of HPC data (Section 3).

2 The ESTEDI Project

ESTEDI started in February 2000 and will be funded until January 2003; currently, the specification phase is completed and the first implementation phase is under way. The project addresses the delivery bottleneck of large HPC results to the users by augmenting

[2] see www.estedi.org
[3] see www.rasdaman.com

the high-volume data generators with a flexible data management and extraction tool for spatio-temporal data. The observation underlying this approach is that, whereas transfer of complete data sets to the client(s) is prohibitively time consuming, users actually do not always need the whole data set; in many cases they require either some subset (e.g., cut-outs in space and time), or some kind of summary data (such as thumbnails or statistical evaluations), or a combination thereof. Consequently, it is expected that an intelligent spatio-temporal database server can drastically reduce networks traffic and client processing load, leading to increased data availability. For the end user this ultimately means improved quality of service in terms of performance and functionality.

The project is organised as follows. Under guidance of ERCOFTAC[4] (European Research Community on Flow, Turbulence and Combustion), represented by University of Surrey, a critical mass of large European HPC centres plus the CFD package vendor Numeca s.a. perform a thorough requirements elicitation. In close cooperation with these partners and based on the requirements, the database experts of FORWISS (financial/administrative project coordination) and Active Knowledge GmbH (technical/scientific project management) specify the common data management platform.

The implementation platform RasDaMan (see Section 3) has been installed at all sites in the HPC partners' particular environment. In the current first implementation phase, data loaders and accessors are being implemented. Each partner addresses one specific area where the lab has special expertise and users:

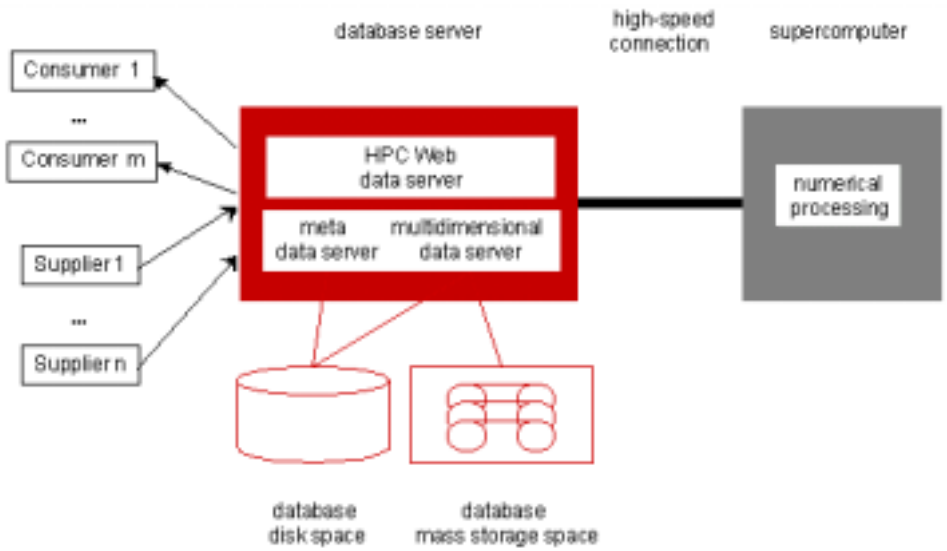

Fig. 1. The generic ESTEDI data management platform

[4] see http://imhefwww.epfl.ch/Imf/ERCOFTAC

- climate modelling by CCLRC (Council of the Central Laboratory of the Research Councils, UK) and MPIM (Max Planck Institute for Meteorology, DE);
- cosmological simulation by CINECA (Interuniversity Consortium of the North Eastern Italy for Automatic Computing, IT);
- flow modelling of chemical reactors by CSCS (Swiss Center for Scientific Computing, CH);
- satellite image retrieval and information extraction by DFD-DLR (German Remote Sensing Data Center, DE);
- simulation of the dynamics of gene expression by IHPC&DB (Institute for High-Performance Computing and Databases, RU);
- computational fluid dynamics (CFD) post-processing by Numeca International s.a. (BE).

In the next step starting in autumn 2001, the resulting application pilots will be operated under real-life conditions for evaluation.

In parallel to the HPC application development going on, the database developers extend RasDaMan with features determined necessary (see Section 3.2).

All development is in response to the user requirements crystalised by the User Interest Group (UIG) promoted by ERCOFTAC. Active promotion of the results, including regular meetings, is instrumental to raise awareness and ensure take-up among industry and academia, both in Europe and beyond.

The project outcome will be twofold: (i) a fully published comprehensive specification for flexible DBMS-based retrieval on multi-Terabyte data tailored to the HPC field and (ii) an open prototype platform implementing this specification, evaluated under real-life conditions in key HPC applications.

3 The RasDaMan Array DBMS

The goal of the RasDaMan DBMS is to provide database services on general MDD structures in a domain-independent way. To this end, RasDaMan offers an algebra-based query language which extends standard SQL92 with declarative MDD operators. Server-based query evaluation relies on algebraic optimisation and a specialised array storage manager.

Usually, research on array data management focuses on particular system components, such as multidimensional data storage [8] or data models [5, 6]. RasDaMan, conversely, is a complete array DBMS. It is generic in that functionality and architecture are not tied to some particular application area, but apply equally well, e.g., to 2-D satellite maps and 4-D climate simulations.

3.1 Modelling and Querying Array Data in RasDaMan

The conceptual model of RasDaMan centers around the notion of an n-D array (in the programming language sense) which can be of any dimension, spatial extent, and array cell type. As cell types, all valid C/C++ types and nested "structs" are admissible; the only exception is pointers, which are replaced by the database concept of persistent OID (object identifier) references. Following the classical relational database paradigm, RasDaMan also supports sets of arrays. Hence, a RasDaMan database can be conceived as a set of tables where each table contains a single column of array-valued attributes, together with a system-provided unique OID. These OIDs can be used to reference particular MDD objects from elsewhere in the database.

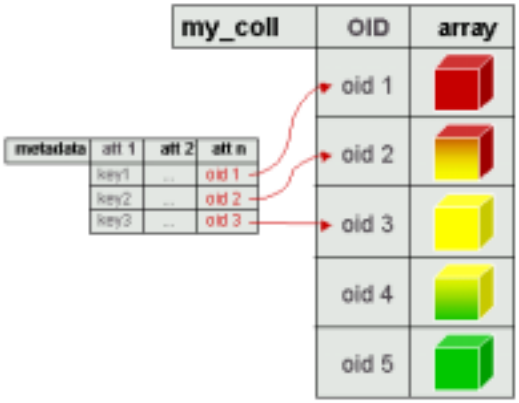

Fig. 2. The RasDaMan conceptual model: tables of n-D data cubes which can be referenced from elsewhere in the database

Based on a specifically designed array algebra [1], the RasDaMan query language, RasQL, offers MDD primitives embedded in the SQL query paradigm [2]. The expressiveness of RasQL enables a wide range of signal processing, imaging, and statistical operations. To give a flavour of the query language, we present some small examples.

Let us consider a table `ClimateModels` of 4-D climate models with dimensions `x`, `y`, `z`, and `t` containing model variables `T` (temperature) and `vx`, `vy`, `vz` for the wind components.

The following query retrieves, from all climate models stored, the T component:

```
select cs.T
from   ClimateSimulations as cs
```

The result is a set of 4-D cubes containing only the `T` variable. The principle behind is that expressions on cells are applied simultaneously to all cells, in RasQL called *induced operations*. For example, let us retrieve the absolute of the wind speed:

```
select sqrt( cs.vx*cs.vx + cs.vy*cs.vy + cs.vz*vz )
from   ClimateSimulations as cs
```

Spatial subsetting is done by so-called trim expressions where, for each dimension, the lower and upper bound of the desired result is specified; a wildcard "*" denotes the array's current bound. A section (projection) at a specific position along a dimension is specified by indicating not the lower/upper bound pair, but the section point. As an example, let us extract at time `t=42` the layers from ground up to 1,000 m (assuming meters as unit):

```
select cs[ *:*, *:*, *:1000, 42 ]
from   ClimateSimulations as cs
```

To select a subset from a table whose items fulfil a certain predicate, the `where` clause can be used in the same way as in SQL. The query below retrieves all those models where average temperature in 1,000 m over ground exceeds 5° C; the `avg_cells()` operation is a so-called condenser which computes the average value of all cells in the argument MDD:

```
select cs
from   ClimateSimulations as cs
where  avg_cells( cs[ *:*, *:*, 1000, *:* ] ) > 5.0
```

In the where clause, those data sets are determined which fulfil the average temperature criterion. Each element in the result set, then, undergoes a so-called trimming which reduces the data cube to that part which actually is desired.

3.2 Application Development and Client/Server Communication

Application programming interfaces supported are a Java and a C++ binding, both compliant to the object database standard ODMG 3.0[5]. The client libraries perform client/server communication, query preparation, and handling of MDD objects. For example, the following Java code piece creates a query object, prepares it, sends it to the server call, and receives the result set:

```
OQLQuery myQu = myApp.newOQLQuery();
myQu.create("select avg_cells(a) from " + coll + " as a");
DSet result = (DSet) myQu.execute();
```

[5] see www.odmg.org

MDD objects by default are returned in the main memory format which the client code expects, depending on CPU and compiler. Hence, arrays can immediately be processed with the means of the programming language. If desired, objects can be packaged by the server into particular data formats; for example, the following query would return JPEg images ready for usage in a browser-based Web application:

```
select jpeg( cs[ *:*, *:*, 1000, 42 ] )
from   ClimateSimulations as cs
```

The C++ interface uses RPC as the underlying communication protocol which allows the server a sign of life monitoring. The Java binding relies on http as its communication protocol. While this does not permit a sign of life technique, it has the advantage that it is compatible with firewalls. Hence, a Java client can safely be admitted in high-security environments.

3.3 Physical Array Storage and Processing

Internally, RasDaMan employs a storage structure based on the partitioning of an MDD object into tiles, i.e., sub-arrays. Subdivision does not have to be a regular grid, but can be defined to consist of arbitrary non-aligned tiles (see Fig. 5). To quickly determine the tiles affected by a query, a spatial index is employed [3]. Optionally tiles are compressed when stored; moreover, result data can be compressed for transfer to the client. Both tiling strategy and compression form tuning parameters invisible at the query level, but under control of the database developer and administrator.

Each tile is stored as a BLOB (Binary Large Object) in a relational database comprising a safe persistent store. As the resulting structure is very simple – an MDD object is mapped to a set of BLOBs –, any relational DBMS can serve as the underlying storage manager. Actually, even an object-oriented DBMS, O2, has been coupled with RasDaMan this way. An immediate advantage of this technique is the reduced administration overhead (only one database has to be maintained for both meta and array data) and the guaranteed consistency as opposed to the conventional mixture of relational meta data and file-based array management.

A series of optimisation rules is applied to a query prior to its execution to achieve an optimal access and processing pattern [7]. Of the 150 heuristic rewriting rules, 110 are actually optimising while the other 40 serve to transform the query into canonical form.

In the course of ESTEDI, a series of enhancements to RasDaMan will be done. To accommodate data volumes beyond disk capacity, tape archives will be coupled to RasDaMan, including staging techniques which take into account multidimensional spatial neighbourhood of tiles.

Tiles form not only the units of storage access, but obviously also form natural units for query parallelisation. Therefore, the RasDaMan server will be modified so as to evaluate queries by passing each tile to a separate processor. Among the problems to be addressed is overlapping tile borders in operations such as filters requiring access to the

neighbourhood of each cell. Techniques known in parallel computing, such as halo exchange [4], seem promising.

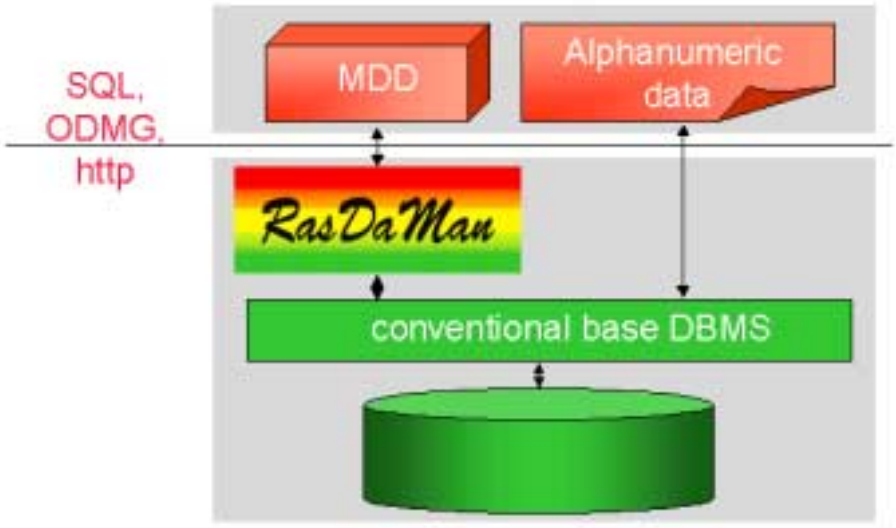

Fig. 3. Embedding of RasDaMan into overall data management

Optimisation of queries is already done extensively in RasDaMan to minimise disk access. The result is that on the average query processing is CPU-bound as opposed to I/O-bound which is usual with conventional database systems. The next step now is to optimise complex imaging and statistical queries with involved operation sequences per cell causing high processing workloads.

4 Conclusion

What is unique about the ESTEDI project is the combination of both HPC and database expertise. In contrast to other approaches also striving to overcome the HPC data delivery bottleneck, ESTEDI does so by providing database query support at a high semantic level, internally supported by transparent optimisation techniques. Two main advantages result from this: firstly, it enables the users to more concisely state the data they need, leading to a more focused result data set. Secondly, the query describes the whole task to the database server, and this opens up a wide field for internal optimisation.

Hence, we feel that such an interdisciplinary approach has considerable potential to overcome the current HPC data management bottleneck and, at the same time, provide a

substantially new quality of service to the users. In this sense, we see the ESTEDI initiative as an essential step towards the envisaged Grid community.

Acknowledgement

The author wishes to express his joy about the team of outstanding ingenuity and spirit that has met to form the ESTEDI consortium. It is such kind of active involvement which makes wheels rolling.

References

1. P. Baumann: A Database Array Algebra for Spatio-Temporal Data and Beyond. Proc. Next Generation Information Technology and Systems NGITS '99, Zikhron Yaakov, Israel, 1999, pp. 76 - 93.
2. P. Baumann, P. Furtado, R. Ritsch, N. Widmann: Geo/Environmental and Medical Data Management in the RasDaMan System. Proc. VLDB'97, Athens, Greece, 1997, pp. 548-552.
3. P. Furtado, P. Baumann: Storage of Multidimensional Arrays Based on Arbitrary Tiling. Proc. ICDE '99, Sydney, Australia 1999, pp. 480-489.
4. J. Hague: Halo Exchange in Mixed Shared and Distributed Memory Processors. 9th Workshop on the Use of High Performance Computing in Meteorology – Developments in Teracomputing, European Centre for Medium-Range Weather Forecast, Reading, UK, 2000.
5. L. Libkin, R. Machlin, and L. Wong: A Query Language for Multidimensional Arrays: Design, Implementation, and Optimization Techniques. Proc. ACM SIGMOD'96, Montreal, Canada, 1996, pp. 228 - 239.
6. A.P. Marathe, K. Salem: Query Processing Techniques for Arrays. Proc. ACM SIGMOD '99, Philadelphia, USA, 1999, pp. 323-334.
7. R. Ritsch: Optimization and Evaluation of Array Queries in Database Management Systems. PhD Thesis, Technische Universität München, 1999.
8. S. Sarawagi, M. Stonebraker: Efficient Organization of Large Multidimensional Arrays. Proc. ICDE'94, Houston, USA, 1994, pp. 328-336.

Appendix

The following images show sample HPC application data retrieved from RasDaMan databases. All visualisations have been done with rView, the RasDaMan visual frontend.

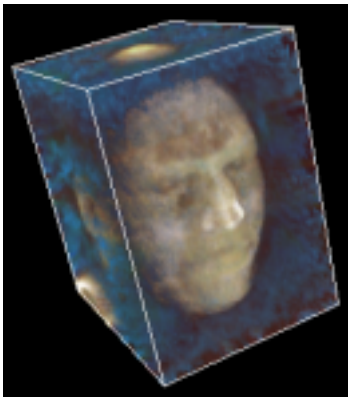

Fig. 4. Visible Human excerpt (3-D); data courtesy of Visible Human Project, see www.nlm.nih.gov/research/

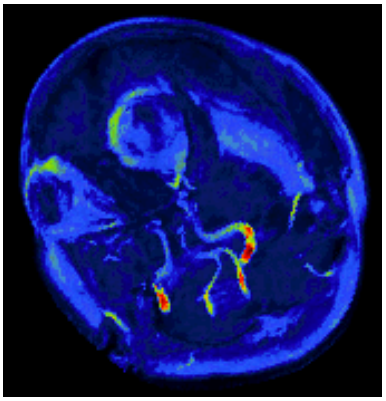

Fig. 5. Human brain activitation map (3-D); data courtesy Karolinska Institutet, Stockholm, Sweden

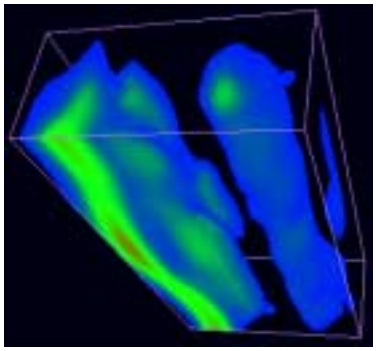

Fig. 6. Climate simulation (3-D retrieval result from 4-D climate model); data courtesy German Climate Research Centre

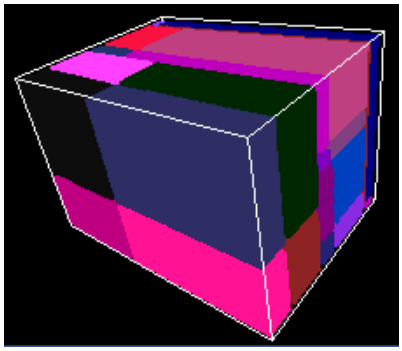

Fig. 7. Visualisation of the internal tiling structure of a 3-D object