The operational database system of the Deutscher Wetterdienst (DWD) based on a commercial RDBMS

R. Stanek
Deutscher Wetterdienst
Offenbach, Federal Republic of Germany

Summary

Starting a new forecast system in November 1999 at the Deutscher Wetterdienst a new data handling system has been implemented at the same time. The new data handling system is based on a commercial RDBMS (Relational Data Base System, ORACLE 8.x).
Accessing data from clients is performed either by socket communication or by SQLNET (here with restrictions). Data formats can be GRIB code, BUFR code or pure binary with heading format description. In principle data can be stored as database type BLOB (Binary Large Object) or BFILE (Binary File).
The connection of a hierachical storage mangement system (HSM) with the database is based on a metadata model of the database.

1. Introduction

The new forecast system of the Deutscher Wetterdienst has a higher resolution than the previous one. Therefore, the new forecast system implies increasing technical requirements, especially handling very high data quantities up to some terabytes a day and high performance according to database access.

Recently, a commercially supplied database (ORACLE) has been implemented. It is based on a relational data model which takes the information about and the relationships between the data. In contrast to an earlier data handling system which holds only data as binary format BUFR, GRIB or STRING (pure binary formats with a descriptive heading, e.g. ASCII texts may be stored as STRING) the new data handling system allows storing all kind of data in a transparent way, especially climate data in 'browser format' like NUMBER, CHAR, VARCHAR. It simplifies the accessing of data for using new technologies like internet. The main reasons for a commercial system are

- standardisation of database technology,
- portability,

but it requires also

- transaction rates for data access which are fast enough,
- fast and simple migration of existing database applications (batch mode),
- implementation of an array interface by calling from FORTRAN or C programs,
- unique command interface for retrieval data which are set offline or online.

The database is installed under ORACLE 8.x on SGI ORIGIN 2000, IRIX 6.5. The data access to binary data is performed by a user interface layer called 'CSOBANK'. The Interface 'CSOBANK' was developed by DWD. The Interface combines C programs with anonymous PL/SQL-blocks and OCI-Oracle Call Interfaces.
It guarantees high performance coupled with modern database management facilities.

2. Database configuration with respect to handling binary data

The following picture illustrates the connection between clients and data server for data access by using the interface layer 'CSOBANK':

Some server processes on the data server are connected to client processes by socket communication. Using a DWD written socket communication makes a SQLNET/client installation superfluous. Any request sent by the client process is received by the server process. The direction of transferring data from (to) data server to (from) the client depends on user requests (retrieving or storing data). There are two possibilities to store the binary data: Either as datatype BLOB (Binary Large Object) or BFILE (Binary file). The system is particularly suited to handle binary data exchange formats FM 92 GRIB and FM 94 BUFR code which are defined in the WMO Manual on Codes. On the other hand purely binary data (e.g. pictorial information, ASCII texts) can be handled as well.

For both BLOB and BFILE the metadata (free definition, for GRIBs mainly information of the product definition section) will be stored inline in ORACLE tables as rows identified by primary and foreign keys. There is all the information like prediction time, reference time, level, parameter that one needs to retrieve the data.

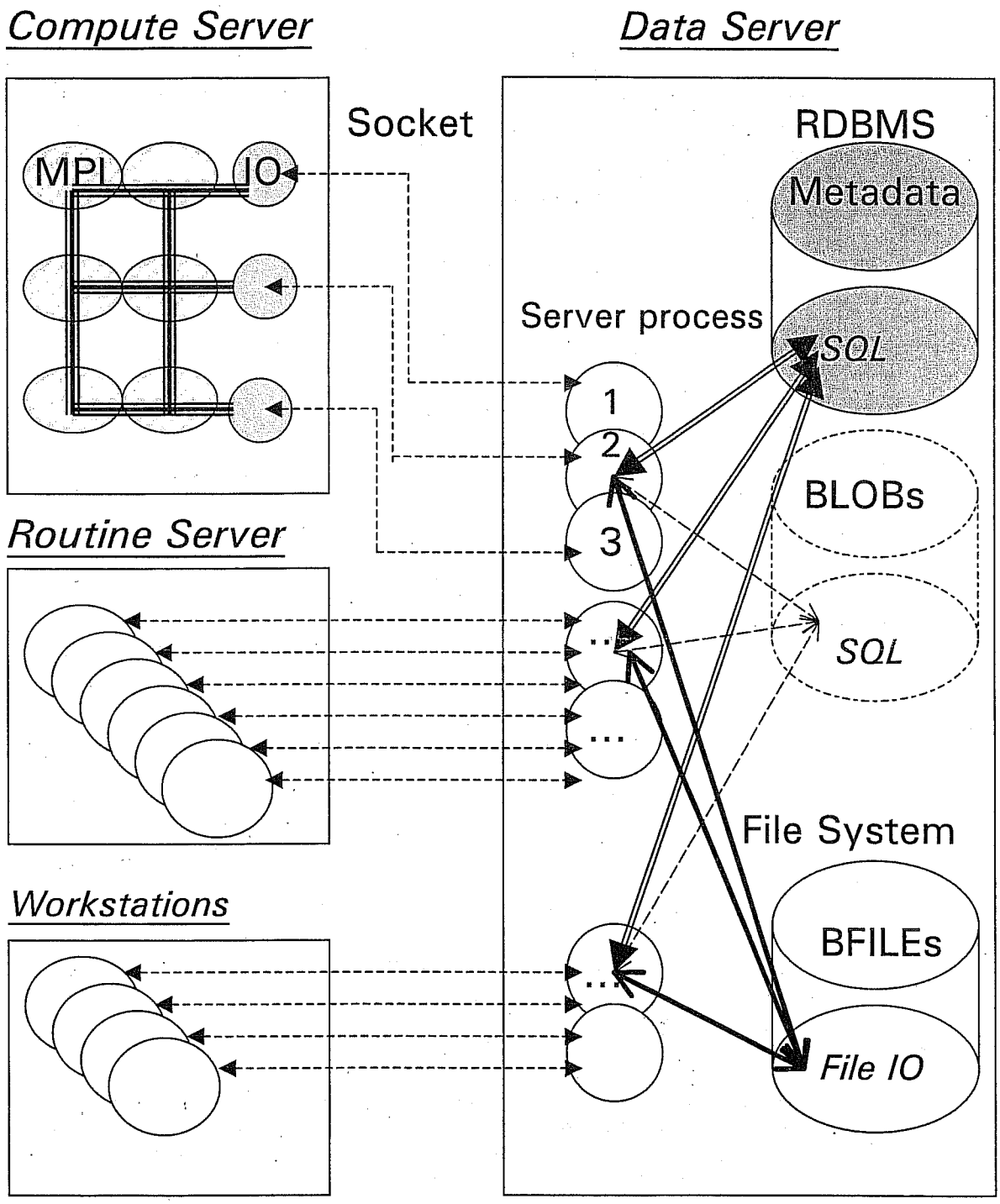
If datatype BLOB is selected the whole data transaction runs under control of ORACLE RDBMS. The binary data are stored as internal LOBs in the ORACLE 8 database. Writing and reading access can be done by 'CSOBANK' or other programs using SQLNET, e.g. OCI or extended PL/SQL.

On the other side, if BFILE is selected the data are stored in external host files. In addition to the metadata the reference to the postion in the filesystem is stored. ORACLE functionality allows only retrieving of data, whereas writing access for BFILEs is generally not supported. Therefore, writing data to data server is restricted to the use of 'CSOBANK'. Hereby the transaction control must be additionally implemented.
The performance is about 70 transaction/sec for metadata and 30 MByte / sec disk write, all measured directly on the data server for a single process. Including the net overhead the total transaction rate results to about 10 Mbyte / sec. It is remarkable that storing GRIBs as datatype BLOB has less performance. Here maximum transaction rates up to 1 MByte / sec were measured.

The use of asynchronous I/O with simultaneous parallel access during the NWP run will raise the performance of the database. In that case postprocessing routines can immediately receive the data from the database as soon as they are produced by the forecast model.

Generally all model fields (which size are presently of order 300 Kbytes) are stored as datatype BFILE. That means, every GRIB is stored as one external host file.

The connection HSM (hierarchical storage management, AMASS) to ORACLE is based on the metadata model of the database. According to special criteria (prediction time, reference time, run type of model...) the gribs are bundled into up to 1 GB files and are migrated to HSM. For receiving data from the archive 'CSOBANK' must be used.


Compute Server
Data Server
Socket
MPI
IO
RDBMS
Metadata
Server process
SQL
1
2
3
BLOBs
Routine Server
SQL
File System
Workstations
BFILEs
File IO

3. Description of the User Interface of the new DWD database

The new database may be accessed both by the interface layer of 'CSOBANK' or by direct access with SQL when SQLNET is installed. The database requests used by 'CSOBANK' have a simple layout. However, some of the abbreviations used are of German origin, thus migration of existing applications was done very quickly:

For example with the request:

```
csobank>    ak=re, ty=r128f, rty=m, rki=routi, dbase=gme, d=h00, lv=500,
            vv=00, vvmax=78, ee=11, sql="order by itime1 asc"
```

temperature fields (ee=11 from GRIB code table 2) of the new global model (ty=r128f,dbase=gme; rty=m: main run) are read. Prediction times (vv) are between 0 and 78 hours, level (lv) is 500 hPa and reference time is today 00 UTC. The SQL statement (sql=) is directly translated into a where clause sorting data by prediction time in ascending order.

Software utilities are written to access single grid points on demand, for example to produce time series, without retrieving whole grib data. This kind of data access is simply supported by OCI- Oracle Call Interface. But indeed for direct access knowledge of data structures of RDBMS is required.

File and array interface are supported, i.e. the results can be fetched directly into the program variables.

Security policy is established. Reading and writing access are ruled by determination of object privileges controlled by roles and application privilege management of the commercial database system. Besides additional passwords can be required.

Supported functionality of user interface are:

AKTION=	(ACTION=)
ak=ar	Explicit archiving data by specification of time boxes
ak=cr	Create global metadata which are necessary to build up a database
ak=dfset	Set defaults
ak=help	Access to online help documentation
ak=in	Print information of metadata
ak=lo	Delete single data
ak=lt	Delete whole time boxes
ak=pu	Purge whole database
ak=re	Retrieve data

ak = test	As ak = re, but only test, how many data are found
ak = st	Store data
ak = me	Select primary keys, necessary to access data with specific byte adress (single grid points)

4. Planned Activities

The migration of the old GRIB database to the new commercial system is successfully completed.
It has been proven that storing and retrieving data can be done in a efficient way. The perfomance is fast enough with respect to the new forecast system. In addition, some other binary data like satellite data, gridded products of precipitation data, radar data are presently stored as BFILEs in RDBMS.

In future all observations will be stored in the new system as a uniform dataype BUFR. Different to datatype GRIB, which is characterized by a big data amount (in order of MByte), BUFR is very short (some hundred Bytes) and, therefore, it is planned to store these as BLOB.

Simultaneously a complex data model is developed to store the same data with classical data structures which are typically for relational databases. The data are presented uniformly, easily accessible by commercial browser tools, especially suitable for internet applications.