USE OF BINARY DATA REPRESENTATIONS AT FLEET NUMERICAL OCEANOGRAPHY CENTER

W. THORPE
FLEET NUMERICAL OCEANOGRAPHY CENTER
MONTEREY, CALIFORNIA, USA

1. INTRODUCTION

The Fleet Numerical Oceanography Center (FNOC) began a project several years ago for a replacement of the on-line storage, user access and archival of the reports data base. The data base to be replaced consists of a permanent file of index sequential records, where each record is identified by a system label containing a 3 character type-of-data identifier and a 4 character calculated date-time. Data within each record is in packed binary form, where each report is assigned a fixed number of 60-bit words. Each parameter within a report is assigned a fixed number of bits, with no individual parameter crossing word boundaries.

User access to the data is via direct access to the data records through a multi-user subroutine. Isolating individual parameters within a report required shift/mask procedures in each application program. The storage format, plus users having direct access to the data, resulted in duplicate code among application programs and also required coordination with all users before any changes could be made to the format of the data base.

2. BUFR AS A STORAGE FORMAT

While it was immediately obvious that BUFR, or a variation of BUFR, was an ideal choice for the data base replacement, FNOC performed extensive testing of BUFR to be sure it would satisfy the requirements for use as the on-line storge format for the

reports data base. Also to be considered was whether BUFR could be effectively employed in providing an interface program between the user and the data base. Areas of prime concern were access speed and storage requirements.

2.1 DBMS Testing

A test data set consisting of 16332 SYNOP reports was used to perform timing tests on two commercially available Data Base Management Systems (DBMS) to determine if a commercial system could replace the user direct access and shift/mask procedures to identify individual parameters within reports. Timing for the user direct access method under the old format of the data base, and shifting/masking individual parameters produced a CP time of .3 seconds, 10 wall clock seconds in a non-dedicated machine mode using CDC 855 computers equipped with 885 disks under the NOS/BE operating system.

Data was then put into BUFR format and tests were performed with the two commercial DBMSs under the NOS/VE operating system. The CP time for accessing the 16332 reports with one DBMS was 36 seconds, 175 wall clock seconds in a dedicated machine mode. The other commercial DBMS performed better, using 17 CP seconds, but an equally excessing wall time. In both tests the times for the DBMSs included only accessing of the BUFR records with no individual parameters isolated.

A third set of testing was done where the same BUFR data set was accessed using a prototype in-house file management system including execution of a BUFR decoder. The timing was 2.6 CP, 10 wall clock seconds for both accessing and isolating individual parameters within reports; the BUFR decoder did not apply reference values or scaling to individual values.

From these tests it was concluded that the advantages of a commercial DBMS, such as, increased control over data security and integrity, increased portability of applications programs which access data, and increased flexibility of data and software designs is not worth the sacrifice in performance.

2.2 Data storage using BUFR

The continuous binary stream which characterizes BUFR is similar to the packed binary method of data storage that BUFR was intended to replace. The major differences are that the BUFR version contains the Section 3 description of data, and individual parameters may cross word boundaries. The BUFR on-line storage of data will contain both Sections 3 and 4 (there is no need to include Sections 0, 1, 2 or 5). Since there is a need for quick access to complete individual reports contained within a BUFR "message", and to check for duplicate reports when adding new data to the existing data base, it was determined through timing tests that considerable CP time is saved by forcing each report to begin on a computer word boundary. This is accomplished by filling the last computer word of a report with binary zeroes, if the report does not end on a word boundary. It is easily determined from the Section 3 data descriptors how many bits are necessary for each report of a given type, which is then readily converted to the number of computer words required to contain that report. For those reports with possible delayed replication, eg TEMP reports, a local descriptor is included in the Section 3 description of the data that indicates the number of words required for an individual report.

Even though this method of storage is machine dependent, and there is some loss of space efficiency of the continuous bit stream, only trivial changes are necessary to the BUFR encoding/decoding routines for machines with different word lengths. The possible zero fill added to each report to ensure starting on a word boundary is a small price to pay for the gain in speed of access.

2.3 BUFR-stored data at the record level

BUFR data is stored at the record level in the same manner as the non-BUFR storage; the file consists of index sequential records identified by a system label. The label contains a 3 character type-of-data identifier followed by a 4 character calculated date-time. The size of each record varies according to data types, generally around 3000 60-bit words. For data

sets larger than what will fit into one record, separate records are generated as continuations. The continuation records contain only Section 4 data, as the Section 3 description covers the first and all subsequent records for the same type and date-time of data.

3. USER ACCESS OF BUFR-STORED DATA

Because of the poor performance of the commercial DBMSs, FNOC developed an in-house file management system for accessing data stored in BUFR. FNOC is currently using the NOS/BE operating system which used fixed-size field length to control main memory. The extraction of data from the BUFR data base is accomplished by a separate program which is part of the user's job stream and runs before the execution of the application program. This pre-processing program is controlled by a set of directives which describe the user's data request and will be contained in an input record located within the job stream.

3.1 Directives

The input directives are defined by the mandatory key phrase EXTRACT, and two optional key clauses, PARAMETER and WITH. The mandatory phrase EXTRACT is used to describe which report type is to be returned. Report type names are in plain language such as surface-land, pibal, bathy, etc. In order to specify which report type is being requested, the user will use the EXTRACT phrase. This phrase will consist of the key word EXTRACT, followed by one or more report-type names. An exmple of this clause is:

```
EXTRACT surface-land
```

When a user requires elements contained within a BUFR record in other than standard BUFR units, the user can specify conversion to the units desired through the optional key clause PARAMETER. This request is initiated by the key word PARAMETER, followed by a list of parameter unit requests. These requests, separated by commas will be made up of a parameter name, followed by the keyword IN, followed by a units identifier.

```
PARAMETER ttdb IN celsius
```

The extent of the search can be limited by the use of the third optional key clause, WITH. This clause can use either a range group or a comparison group. The range group is used to indicate the upper and lower limits within which the extracted data must fall. It consists of the keyword WITH, followed by one or more range groups separated by commas, where the group is a parameter name, the key words RANGING FROM, a lower value, the key word TO, and an upper value. A comparison group is used to describe the relationship between the requested parameters and a specified value. It consists of one or more comparison groups, separated by commas, where each group is a parameter name, a comparison operator, and a value. The comparison operator can be one of the following: < , > , =, <>, <=, >=.

Example: extract laca (latitude-course accuracy) between 30N and 30S and lonca (longitude-course accuracy) between 20W and 20E.

```
WITH laca RANGING FROM -30 TO 30, lonca RANGING FROM -20
TO 20
```

Example: extract dry bulb temperatures <= 10 and dewpoints >15.

```
WITH ttdb <= 10, tdtd > 15
```

The above requests can be combined, intermingled, and written in free form:

```
WITH laca RANGING FROM -30 to 30,
ttdb <= 10,
lonca RANGING FROM -20 TO 20,
tdtd > 15
```

Example of complete directive:

Extract dry bulb temperatures from surface land reports where the time is between 06Z and 12Z, latitude between 30N and 30S, longitude between 20E and 30E:

```
EXTRACT surface-land
PARAMETER ttdb IN celsius
WITH gg RANGING FROM 6 TO 12,
laca RANGING FROM -30 TO 30,
lonca RANGING FROM 20 TO 30
```

4. USER ACCESS SUBROUTINES

Once data has been extracted in the pre-processing program, users have access to the data from within applications programs through four subroutines, CHPARM, BUFFRD, OPENFIL, and CLOSEFIL. CHPARM enables the users to specify the parameters they require. CHPARM will inform them if the parameters are available at the time the routine is executed, and will initialize the routine for subsequent calls to BUFFRD. When the user call CHPARM, the user will pass to it a character array which contains the list of requested parameters. The order of the parameters in this array will determine the order the values are stored when they are returned by BUFFRD. CHPARM will also return an array of logical variables, with one variable for each requested parameter. These logical variables will indicate whether or not the corresponding parameter is available. The format of the call is:

```
CALL CHPARM (CNAME, CPARMS, LINDIC, ILENTH)
```

where

- CNAME - name of the report type requested in the input directives
- CPARMS - character array which contains the names of the requested parameters
- LINDIC - logical array which returns, for each parameter requested, a value of true if it is available, and false if not available
- ILENTH - integer variable indicating the length of both

The second subroutine, BUFFRD, is used to return specific values for the parameters originally requested with the call to CHPARM. Each call to BUFFRD will return the values from one report in a real array, with one value per word, and in the same order as the parameters listed in the call to CHPARM. BUFFRD will also return a logical variable that will indicate the end of the data in the temporary file. The format of the call is:

```
CALL BUFFRD (VALUES, LEOD)
```

where

VALUES - real array used to return the values requested
LEOD - logical variable that returns the value TRUE when the end of the local file is reached.

The remaining subroutines OPENFIL and CLOSFIL are used to open and close the local files for the particular data types the user wants to read. In each of these subroutines, the user passes the type of data requested and the routine will open or close the appropriate data file.

```
CALL OPENFIL (CNAME)
CALL CLOSFIL (CNAME)
```

where

CNAME is the name of a report type requested in the input directives.

5. DATA ARCHIVE

The modifications to BUFR for efficient retrieval of data in the on-line storage does not apply to archived data. In the process of moving data from on-line into the archives, the binary zero fill that each report may contain is removed to

form standard BUFR Sections 3 and 4. Sections 0, 1, and 5 are also added to create standard BUFR "messages".

There does lie ahead the rather large task of converting archived data from the previous formats into BUFR format. At this point no schedule for that conversion has been set.

6. GRIDDED BINNARY (GRIB)

FNOC has developed software to receive and transmit GRIB, but the full capabilities will not be possible for both transmitting and receiving GRIB (and BUFR) unitl the communications lines are upgraded to support X.25. The upgrading of the communications lines are scheduled for 1990.

7. PLANNED FUTURE USE OF BINARY REPRESENTATIONS

FNOC is working toward full capabilities of the use of BUFR in terms of transmission, receipt, on-line storage and archival of data. Still unscheduled is the conversion of data within the archives to BUFR.