The Census Reference System

Edward Cary Bean, Jr., U.S. Bureau of the Census, Washington, D.C.

ABSTRACT:

This paper describes how the advantages of both SAS® and UNIX were combined to create the 'Census Reference System' (CRS), a unique on-line database used by the U.S. Bureau of the Census for conducting its many household-based surveys. This system provides rapid access to over one hundred million data records without the use of a traditional relational database. The unique design of the CRS significantly reduced hardware and software requirements and their associated costs, as compared to other more traditional processing approaches.

BACKGROUND:

The U.S. Bureau of the Census regularly conducts surveys of households and housing across the United States. In order to collect survey information, the Census Bureau creates a master list called the 'Census Reference File' of all the housing units (homes) in the country. The file is based on information from the Decennial Census of Population and Housing and other current surveys. This reference information is used by the Census Bureau to locate and survey housing units for all of its household-based surveys. Some of the surveys using this file are the Current Population Survey, the American Housing Survey, and the Survey of Income and Program Participation. Prior to the implementation of the CRS only a small portion of the housing unit information was accessible via off-line computer tapes. This was due in part to the time required to search such a large volume of data, and in part, to the high cost of keeping the large volume of data in an on-line status on a mainframe computer. As a consequence, prior to the CRS, only a small portion of the information was easily accessible via computer with the remaining information only available by manually searching through vast amounts of microfiche of original Census forms. Therefore, accessing the reference information was time consuming and prone to errors.

Traditional relational database solutions to resolve the problems in accessing the Census Reference File information had been considered and tested. Unfortunately, due to the high volume of data and the number of fields (variables) involved, traditional relational database software proved to be extremely slow and required tens of gigabytes of storage.

The CRS, the topic of this paper, was developed out of a need to overcome the problems associated with traditional databases and provide Census Bureau field staff, analysts, and management with timely access to information contained in the Census Reference File in the most cost effective manner possible. The basic original requirements for the system are detailed below.

REQUIREMENTS:

Design a system that would:

1) Give non-programmers rapid access to over one hundred million data records in a simple to use environment

2) Permit on-line updating of some record level information

3) Store the data as compactly and cheaply as possible

4) Support multiple concurrent users with minimum contention for resources

5) Take advantage of inexpensive workstation hardware

6) Develop, test, and implement quickly.

The system must support access to four different types of information:

1) Non-alterable housing unit data (40 variables, 405 characters per record)

2) Alterable housing unit 'notes' data (4 variables, 200 characters per record)

3) Non-alterable block summary data (23 variables, 254 characters per record)

4) Alterable block summary data (4 variables, 200 characters per record)

While the above requirements would seem to be satisfied by a simple relational database application, the volume of the data prohibits this solution. Traditional relational database software products tend to be stricken with the equivalent of 'data traffic grid-lock' when processing a large volume of very large records stored in multiple tables. This is especially true if more than a very small percentage of records are being requested. As the amount of data and the size of records increase, performance does not degrade linearly, it tends to degrade exponentially. As an additional drawback, artificially splitting large logical records into different database tables and adding indexes to join them frequently increases the size of a data base several times over. This in turn increases disk storage, costs, and degrades processing due to the increased data management overhead.

SOLUTION:

The CRS solves all of the problems associated with a relational database solution and then some, by taking advantage of UNIX pipes, UNIX file compression, UNIX file system hierarchy, the ability of SAS to read data directly from UNIX pipes, and the many application tools available with SAS.

The Census Reference File data has a natural hierarchical structure. Housing units (1 housing unit = 1 record) are grouped by blocks. Blocks are grouped by address register areas (ARAs). Address register areas are grouped by district offices (DOs). The CRS capitalizes on this existing data hierarchy by mirroring it in its use of the UNIX file system. The top level CRS-UNIX file storage directory contains a sub-directory for each DO. Each DO has a sub-directory for each of its ARAs. Each ARA sub-directory contains 1 to 4 files containing all of the data for all of the blocks in the ARA.

In order to conserve disk space, all CRS data are stored as compressed ASCII text files. In most cases, standard UNIX compression of each data file enables a 6 to 1 decrease in data storage requirements whereas, a traditional relational database product could require a 3 to 1 increase in storage requirements. This results in a very significant 18 to 1 reduction in disk requirements when comparing the CRS with a traditional database solution. The data storage requirement for the CRS is 3.1 gigabytes of disk space which easily

fits on 2 high capacity Small Computer System Interface (SCSI) disk drives (formatted capacity of 2.8 gigabytes each) with enough remaining storage for work space and 'notes' data added by system users. The disk space required for a relational database to store and manipulate the same amount of data is estimated to exceed 60 gigabytes. When initially loaded the Census Reference File information consumed 3,118,443,000 bytes of disk storage (even when compressed).

DESIGN AND IMPLEMENTATION:

Several system designs for the CRS were rapidly prototyped using SAS and tested with a sample of the Census Reference File data. The first prototype design involved keeping the data in large compressed ASCII data files, one data file per district office. This approach worked, but it was rejected because it took several minutes and considerable amounts of additional temporary disk space to load the requested data into SAS data sets. Concurrent data requests slowed this prototype design dramatically due to competition for disk access to the very large files and data sets. A second prototype design placed the data in large SAS data sets (one per district office) and added indexes on key fields. While this approach increased access speed dramatically, it was also rejected because of the large amount of disk space required to store the large data sets.

The third and final prototype design for the CRS proves to be very efficient for both data access and data storage. This design stores data in thousands of compressed ASCII files (1 to 4 per ARA). Splitting the data up into relatively small compressed files (approximately 60 kilobytes compressed), based on the data's natural hierarchical structure, greatly reduces the time required for the system to load the desired data. Users supply the system via SAS/AF® screens, with the DOs and ARAs they wish to examine. The system then uncompresses the appropriate data files into UNIX pipes which are read directly by SAS and loaded into SAS data sets. The data sets are then merged, as necessary, and placed into SAS/FSVIEW® screens with custom pop-menus. The average time to load data is approximately one to two seconds of real time. This reduction in data loading time virtually eliminates disk access contention. The performance achieved by the CRS when compared to a relational database processing similar size data requests, was evaluated to be many times faster.

File System Hierarchy:

UNIX Directory/UNIX Sub-Directory/Compressed Text Files

District Office (Directories) *Count = 449*
- Address Register Area (Sub-directories) *Count = 153,272*
  - Housing Unit Data (Files)
    - *Count = 152,833*
    - *Variables Per File = 40*
    - *Length of File in Characters = 405*
    - *Av. Records Per File = from 500 to 2,000*
  - Block Summary Data (Files)
    - *Count = 123,690*
    - *Variables Per File = 23*
    - *Length of File in Characters = 254*
    - *Av. Records Per File = from 20 to 500*
  - Housing Unit Notes (Files)
    - *Count = 0 (built later by system users)*
    - *Variables Per File = 4*
    - *Length of File in Characters = 200*
    - *Av. Records Per File = from 500 to 2000*
  - Block Summary Notes (Files)
    - *Count = 0 (built later by system users)*
    - *Variables Per File = 4*
    - *Length of File in Characters = 200*
    - *Av. Records Per File = from 20 to 500*

The CRS also includes some other interesting innovations. In addition to viewing data, printing reports, and performing extractions for analysis, the system users requested that the system be able to add and modify note fields at the record level for both housing unit and block summary data. All of this was accomplished by means of compressed ASCII files for both source data and related note data. Note files contain only record identification information and the actual note data and only for those records having notes. When a selection of records is requested 1) the appropriate data files are loaded into SAS data sets, 2) their corresponding note files (if any) are loaded into SAS data sets, and 3) the appropriate SAS data sets are then merged for viewing and/or modification. Upon termination of a records request and if a modification has been made by an authorized user, the alterable notes data is written back out to the appropriate ASCII file and compressed. File locking is handled by SAS changing the UNIX file protections and creating a lock file so that only one user at a time can access a specific note file for modification.

Another interesting feature of CRS is that the system uses customized SAS pop-menus over SAS/FSVIEW screens under SAS Version 6.07.02 running on a SUN UNIX Version 4.1.3 operating system (see Figures 4 & 5). This feature was designed and implemented in the CRS even though, at the time, there was no documented way to accomplish this.

It took several days each to develop working prototypes for the 3 system designs on a SUN SPARC 1+ workstation. The adding of refinements and user requested features to the final design version took less than a week. Loading of the data into the CRS took over 2 months of near continuous processing across 3 major computer systems. First, the data was prepared from hundreds of tapes into separate DO files on the Census Bureau's Unisys mainframes. Next the data were downloaded from the mainframes to Digital Equipment Corporation (DEC) minicomputers where the files were compressed and directly accessible by the SUN-SPARC workstation. The compressed files on the DEC minicomputers were 'mounted' from the SUN workstation with the Network File System (NFS) and processed directly into the hundreds of thousands of directories, sub-directories, and data files on disks local to the SUN workstation. (The data files were compressed on the DEC computers to reduce network transfer traffic and because there was insufficient disk space to both store and process all the uncompressed data directly on the SUN workstation's disks.)

In summary, the primary features of SAS and UNIX which made this system possible were:

1. UNIX file compression
2. UNIX pipes
3. The UNIX file system and its hierarchy
4. The ability of UNIX to uncompress data as it flows into a pipe (the zcat command)
5. The ability of SAS under UNIX to read data directly from a pipe as if it were a file

CONCLUSION:

The CRS has now been in full service for nine months. The system hardware is located in a remote site and provides access for 25 to 100 users a day from both the Census Bureau's headquarters in Suitland, Maryland and its main processing site in Jeffersonville, Indiana via a secure wide area network. On occasion the system provides complete searches and large extracts from the complete collection of CRS data. This is accomplished by using simple custom UNIX and SAS programs.

By taking advantage of the best characteristics of both SAS and UNIX, the Census Reference System was developed

quickly and was able to meet, if not exceed, the specified design requirements.

This paper, and all of the software programs for this system are available via Internet File Transfer Protocol (FTP) connection to 'ftp.census.gov'. This should be more convenient than retyping partial examples from a publication. The paper, catalogs, programs (sorry, no data) will initially be located in a sub-directory under 'src/sas'.

If you have any questions about this paper, feel free to contact:

Edward 'Cary' Bean, Jr.
U.S. Department of Commerce
Bureau of the Census
Washington, D.C. 20233
phone - 301-763-4071
Internet - cbean@census.gov

Figure 1: Users pick a Census Bureau District Office to examine.

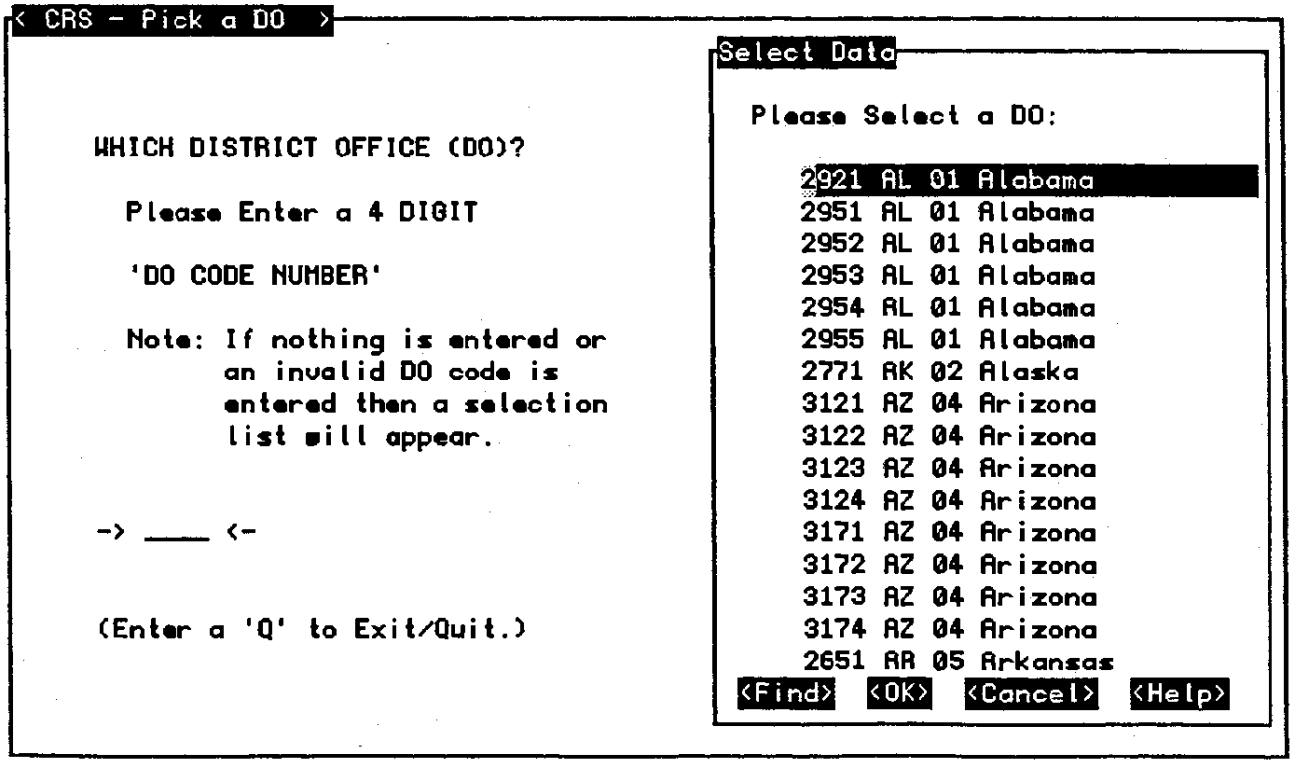

Figure 2: Users pick a Census Bureau Address Register Area to examine.

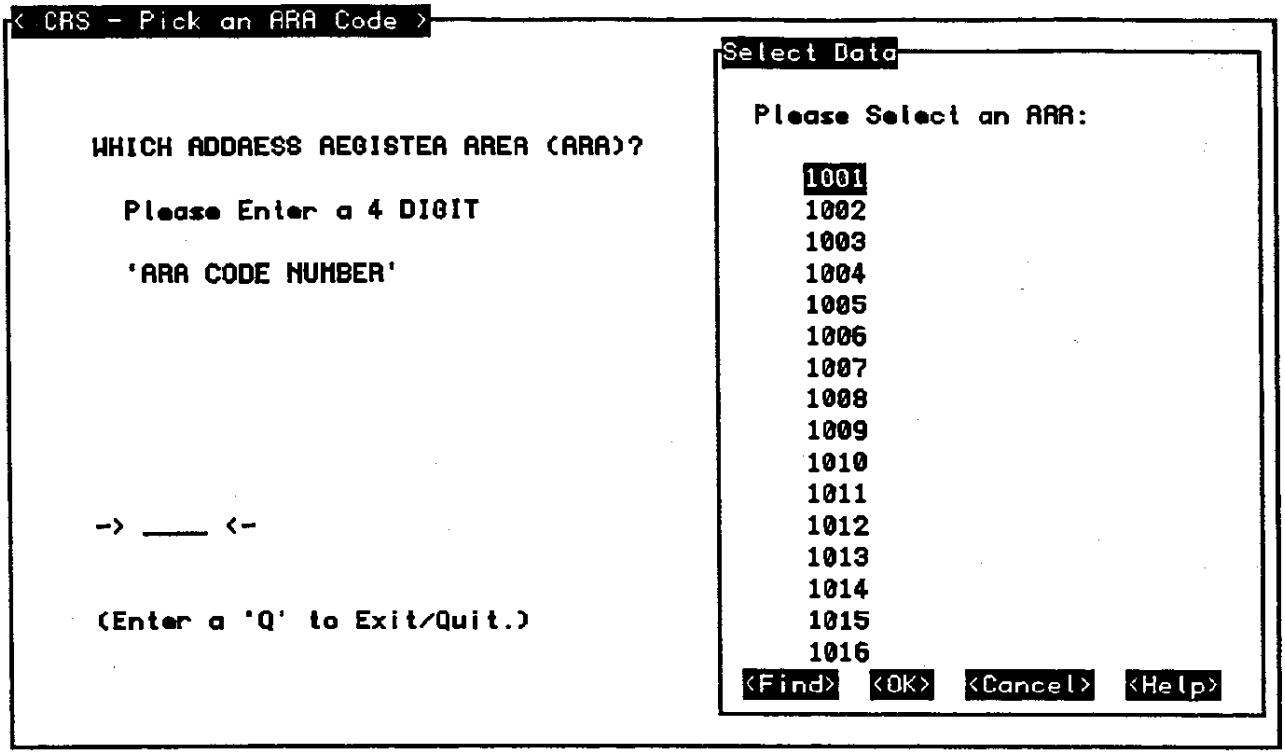

Figure 3: Users pick a method of viewing information, get help or get key functions etc.

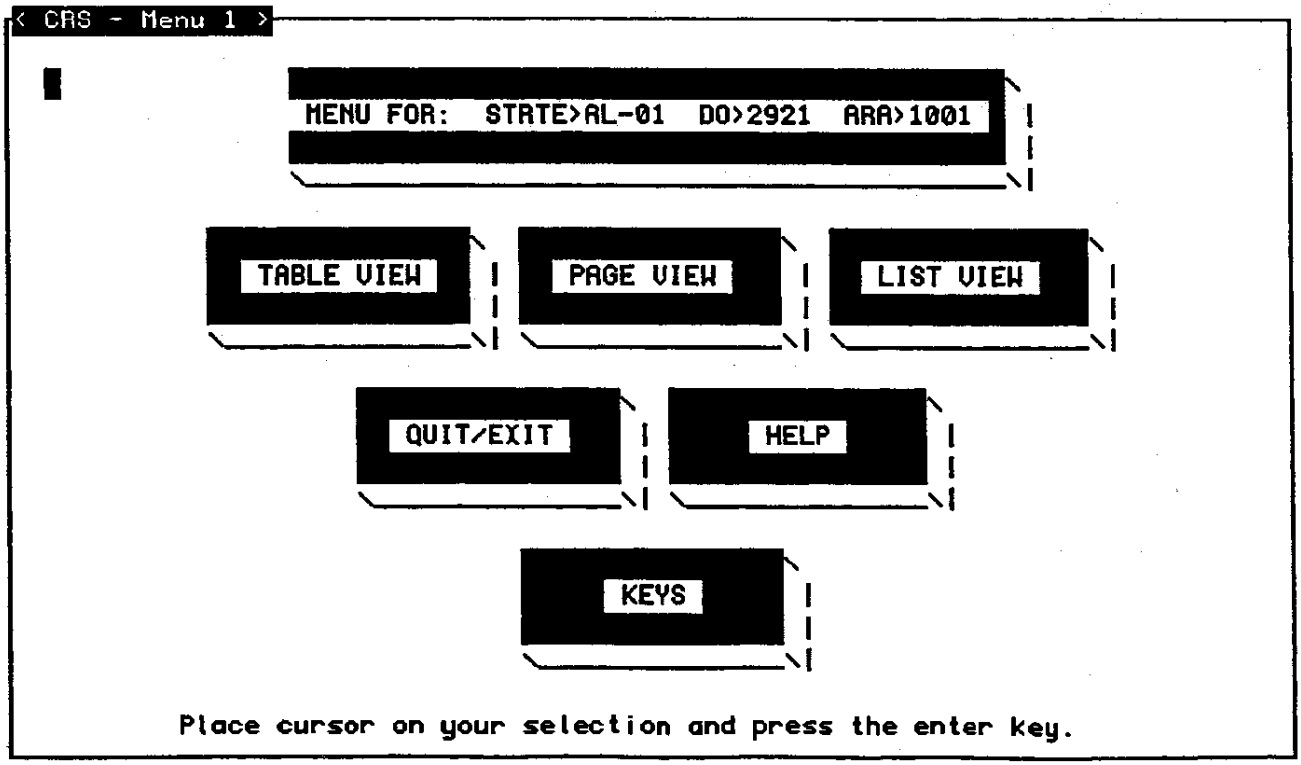

Figure 4: Users can view Block Summary information and Housing Unit information concurrently

FSVIEW: WORK.HU_GQ (E)
ACTIONS HELP QUIT CANCEL

S	CNT	N	COST	DO	ARA	BLK	BLKS	TEA	HUID	ADID	GQHI	PHDI
	1		01	2921	1001	101	A	2	3158149	0002	0	1
	2		01	2921	1001	101	A	2	3158156	0001	0	1
	3		01	2921	1001	101	A	2	3158164	0003	0	1
	4		01	2921	1001	102		2	3158172	0001	0	1
	5		01	2921	1001	103		2	3158180	0002	0	1
	6		01	2921	1001	103		2	3158198	0004	0	1
	7		01	2921	1001	103		2	3158206	0001	0	1
	8		01	2921	1001	103		2	3158214	0003	0	1
	9		01	2921	1001	104		2	3158222	0003	0	1
	10		01	2921	1001	104		2	3158230	0002	0	1

FSVIEW: WORK.BLOCK (E)
ACTIONS HELP QUIT CANCEL

S	CNT	N	ST	COST	CO	COCO	DO	ARA	BLK	BLKS	TRCT	COTR
	1	*	01		117		2921	1001	101	A	0301	
	2		01		117		2921	1001	101	B	0301	

Figure 5: Custom Pop-Menus over SAS/FSVIEW assist and control user activities without custom SAS/AF applications

FSVIEW: WORK.HU_GQ (E)
ACTIONS HELP QUIT CANCEL

SCROLL
GOTO VARIABLE...
HIDE (Variables)
SHOW (Variables)
BLOCK INFO
EXTRACT...
PRINT ON EXIT
SORT ASCENDING...
SORT DESCENDING...
WHERE...
WHERE ALSO...
WHERE UNDO (LAST)
WHERE UNDO (ALL)

DO	ARA	BLK	BLKS	TEA	HUID	ADID	GQHI	PHDI
2921	1001	101	A	2	3158149	0002	0	1
2921	1001	101	A	2	3158156	0001	0	1
2921	1001	101	A	2	3158164	0003	0	1
2921	1001	102		2	3158172	0001	0	1
2921	1001	103		2	3158180	0002	0	1
2921	1001	103		2	3158198	0004	0	1
2921	1001	103		2	3158206	0001	0	1
2921	1001	103		2	3158214	0003	0	1
2921	1001	104		2	3158222	0003	0	1
2921	1001	104		2	3158230	0002	0	1

E)
CEL

S	CNT	N	ST	COST	CO	COCO	DO	ARA	BLK	BLKS	TRCT	COTR
	1	*	01		117		2921	1001	101	A	0301	
	2		01		117		2921	1001	101	B	0301	