

Easy Economic Extraction ex Enormous Entities or Big Becomes Beautiful

Prue Phillips, Australian National University, Canberra, Australia
Pippa Simpson, Wayne State University, Detroit, Michigan

ABSTRACT

Important quantitative economic analysis at the Australian National University (ANU) mainly relies on access to a large international trade dataset. Prior to 1986, this was achieved by programmers accessing data stored on magnetic tape, with turnaround of **up to 2 months!** This was reduced in 1991 to around 1 week, but now....what a difference SAS®, SAS/AF® and a dedicated server have made!

We discuss the design and implementation phases of the now popular STARS* system at ANU, giving the features that users liked and some of the reasons that the system was successful. The trials and tribulations encountered along the way are presented together with how we dealt with them.

Our presentation is suitable for applications managers and developers.

INTRODUCTION

The ANU had collected a large amount of machine readable trade and macroeconomic data for quantitative analysis. By far the largest dataset was the international bilateral trade data by commodity, consisting of approximately 30 million records each year.

Before 1984, trade data were stored on magnetic tapes -- one for each year from 1965. For each activity specialized programs had to be developed to extract data required. The data were accessed through a UNIVAC mainframe system, with university wide queuing to access the CPU. Turnaround on jobs requiring 10 minutes CPU was around 7-10 days! Thus for analysis of a bilateral trade matrix over several years one had to allow 2-3 months!!!

The data were not swapped to a FACOM mainframe system until 1986, and development of the systematic access systems began. These still required programmer intervention and due to staffing limitations turnaround was slow. International Monetary Fund, UN Industrial Development Organization, Food and Agriculture

Thus, the problem we faced was that there was a gigantic database which was:

- *difficult to access in a timely fashion,
- *resident on a mainframe, which had many other services to perform,
- *required programmer intervention for each type of activity.

Some of the ways to improve the situation are to:

- *keep data in disk storage and keep on buying disk storage as data expands. This would improve performance at least 100 fold.
- *use a low cost stand alone computer server dedicated to retrieval of this data only.
- *use a more friendly software/hardware environment; e.g. Dbase, RBase, Sybase, SAS, Oracle.

We were hampered by cost and resource constraints (as most are) but followed these paths in part. In 1991, a dedicated SUN sparc server was purchased, with the capacity to store the data on disc. The selection of appropriate software and development of an interactive user front end took into account the needs of the users, the data requirements and the possibly changeable and changing computer environment. Once SAS and SAS/AF had been chosen, the implementation stage involved adjustment to the software limitations and some trials and tribulations.

PEOPLE NEEDS

Users could be any staff at ANU with any level of computer literacy. The majority would have minimal knowledge of the operating system and no knowledge of SAS. We anticipated that the main users would be quantitative economists, but we have since found in addition extensive users are economic historians, human geographers, political scientists, demographers, ecologists and foresters.

--- A system was needed where data could be

Information Systems

on demand. Whilst the main emphasis was to be on the availability of consistent and reliable long-run time-series, it was also important to have the data available in a timely fashion for key variables such as Gross Domestic Product, interest and inflation rates, exchange rates and so on. Both the number of data sets and each data set could be increased or modified probably at least once a year.

Hence the system needs were:

- *ease of use without training and additional documentation,
- *ability to accommodate additional data systems if/when required.

We decided that essential features were:

- *extensive on-line help for all aspects of system - use, data, and so on,
- *self-explanatory menus,
- *different sub-systems for each data source, but consistency between sub-systems,
- *options for entering data requirements either through entering codes or using a menu,
- *aggregation of data and simple calculations,
- *ability to download data locally for analysis,
- *extensive tabulation facilities.

DATA REQUIREMENTS

Given that large data sets were repeatedly accessed we looked for a strategy which would limit storage requirements and also reduce access time .

Size of data

The largest dataset was the trade data comprising approximately 32 gigabytes. There were 12 other smaller datasets comprising approximately 3 gigabytes in decompressed form. First, the trade data were stored in a compressed format, with only key information accessed. This reduced the size of the trade dataset to 2 gigabytes. Other data were either stored as SAS datasets, or as compressed ASCII files.

Non-essential data and information were stripped from the files. Key information was stored as codes wherever possible and SAS formats and table look up were used.

Accessibility

A major problem was in large extraction times for seemingly simple requests.

Some of the solutions used were:

- *the SAS capacity to sequentially input data records, allowing for sequential processing of key information only, saving code, i/o operations and hence time,
- *separation of frequently used data to separate files to minimize search time for the majority of users.

Compatibility between data systems

Data coding was inconsistent for data from different sources, hence a first step was to make datasets compatible.

COMPUTER ENVIRONMENT

The computer environment was changeable only within the resource and funding limitations of the University. It was changing and likely to continue changing as technology (and resources) improve, so flexibility and portability were important. Although a PC/mainframe system was attractive because of user friendliness it was not possible.

Limited funding

In the academic community, PCs and PC software was available only on a very limited basis. Hence, the system had to be designed to allow access through a non-intelligent terminal connected to the ANU network.

Limited Software

There was to be no user software requirements, beyond communication software, so distributed processing was not possible.

Portability

We needed to develop a system that would be easy to convert to a PC system when required.

SAS on a UNIX central server was available and seemed to be the only system that would handle the database, statistical analysis and would fulfil our requirements.

IMPLEMENTATION

SAS/AF was chosen as applications development tool. The facilities available in SAS/AF allowed for the most efficient development of the required features for data handling and user friendly software.

Data Handling

Some aspects of SAS/AF ideally suited our situation:

- *separate sub-systems for each source, to allow for different data structures in each source, could be developed,
- *sequential implementation allowed a simple initial extraction system followed by enhancement of on-line help and analysis facilities,
- *ability to store information about requests through the SAS profile allows for verification and subsequent look-up of requests.

User friendliness

Many graphics capabilities were not available on a UNIX server remotely accessed, limiting the ability to design the menus and displays. However, given the limitations of tabbing, the interface was sufficiently friendly.

Liked Features

The user often found that interactive access to the database allowed them to gain more information than they had originally envisaged. The features that were particularly successful were:

- *extended lists and sized windows used for look-up and help allowed the user to feel in control ,
- *the user profile allowed storing of frequently produced reports, recall of last executed requests, and hence easy modification of last requests,
- *flexible report formatting minimized the work on reports for the user,
- *extracted data was stored in SAS datasets, allowing access to full power of SAS for subsequent analysis.

Despite the on-line help features, we soon found that extensive training available on continuing and flexible basis is invaluable.

TRIALS AND TRIBULATIONS

We did have some difficulties (calling them trials and tribulations may be a bit excessive!) with SAS/AF but our solutions although not sophisticated seem to have resolved our problems.

Permanent Formats

When you use a format in SCL (for instance attaching a format to a variable for display purposes in the attributes screen), SAS/AF appears to make its own copy of it. You can later update it, change the name, delete it and even delete the library it is stored in -- SAS/AF will continue to access the original!

This is a problem if you want to update formats while running the system, or if you want to use the same format name for different subsystems.

Possible solutions are:

- *there is little that can be done for displays, except to use different format names for different subsystems, which is very tedious,
- *for printed reports, you can maintain a separate format which is only accessed in SUBMIT blocks, as the problem only occurs when the format is used by SCL.

Nesting

There appears to be a limit for the number of nestings of CALL DISPLAY. If there are too many, SAS/AF loses its place! The only solution appears to be to substitute an occasional CALL GOTO when this problem is likely to occur.

The Log

When using a CALL GOTO, if the program has an empty screen, the log screen appears, which is confusing. This also has been a problem when initializing a system, but disappears when the initialization program has a screen.

A possible solution is to avoid CALL GOTO, as much as possible. However, when you must use it due to the nesting problem above, make sure you have something on the screen.

CONCLUSION

There is no doubt that the transfer of the data to a dedicated server was the turning point for access

Information Systems

step, allowing for rapid system development, and the liberation of scarce programmer resources for the development of further analysis tools.

The system development was successful because of careful specification of system requirements and structure before commencement. We designed a modular system with the basic system implemented first to improve researchers' productivity and to free programmers to implement more features. Addition of extra data systems is now possible in less than 2 weeks. SAS and SAS/AF do most of the work.

This system was developed and maintained by programmers who have also been users of the data to be accessed. This enabled us to ensure that system is responsive to users' requirements.

ACKNOWLEDGEMENTS

SAS and SAS/AF are registered trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Author contacts

Prue Phillips
International Economic Databank
Australian National University
Canberra 0200
Australia

Phone : 616 - 249 3065
Fax : 616 - 249 3941
email : Prue.Phillips@anu.edu.au

Pippa Simpson
Department of Pediatrics
Wayne State University
3901 Beaubien
Detroit, Michigan 48201-2196
USA

Phone : 313 - 745 5875
Fax : 313 - 745 5441
email : psimpso@cms.cc.wayne.edu