Abstract

The perception of quality service is often related to the timely and accurate response to user concerns. To help us assess how we are doing, we partnered an X window application with SAS to input and analyze data about user contacts. During the project, we ran into a problem with null bytes in C-generated binary files. This paper describes our solution to the handling of raw C binary files and how we hope to increase our quality of service by combining the use of an X application with SAS.

Introduction

Scientists rely on supercomputing tools to perform critical research computations in a timely manner. Without such tools, the same computations might take months or even years, crippling progress. The Scientific Computing Division of the National Center for Atmospheric Research provides supercomputing facilities to nearly 1,400 researchers at over 100 sites on a local, national, and international basis. Two-thirds of the computer users are from universities around the country, and for many of these users, NCAR is a critical resource for their supercomputing needs.

NCAR is one of five supercomputing centers in the United States funded by the National Science Foundation and is managed by the University Corporation for Atmospheric Research (UCAR), a consortium of 59 member universities. SCD's mission is to provide support for research in the atmospheric, oceanic, and related sciences, the development and execution of large models, and the archiving and manipulation of large datasets. As in many organizations, past experience has shown that the rating of our performance (and subsequent funding) is directly related to the quality of service that our users receive.

Why Analyze Our Tracking Data?

Helping people located over a wide geographical area use computing services that are distributed over a wide range of machines makes our support task challenging. We necessarily consult on a wide range of areas: the Cray computer operating system (UNICOS) and programming languages, code optimization and vectorization, the front-end computer, the Mass Storage System (containing over 17 trillion bytes of data), networking and data communications, software libraries, as well as other areas.

In an effort to continue to improve our quality of support in these areas, the SCD Consulting Office began tracking user questions approximately six years ago. Contacts were logged by hand, and data were primarily used as a reminder that a contact was still open; also for reference, should another user have the same problem.

As contacts increased to over 1,000 per month, consultants found the old method of logging contacts unacceptable in terms of overhead. Completion of the logs became a task relegated to one day at the end of the month. Additionally, management wanted more information about each contact logged, and they wanted it quickly. We knew we had to look for new ways to log and analyze our data.

The Front End: xconslog

The first step in analyzing our tracking data was to replace the archaic paper-and-pencil method with an electronic one. All consultants got diskless Sun workstations with the X Window System. Consultants gave input to a new X window application (also called an X client) and a few weeks later xconslog was born. With xconslog, what had taken minutes to log, was reduced to a few keystrokes and the click of a button. Data were now entered on a real-time basis (see Fig. 1). In addition to the expected speed, we found it was now easy to verify our database had the correct identifying information. It also provided us with the user's electronic mail address, so we could send a response at the fraction of the cost of a phone call. The only thing left to do was enter comments about the solution, and click "Done" and "Save." As a result of the new method, contact information became timely for the first time in many years and was more accurate.
The Back End: SAS

Since our facility was converting to a totally UNIX environment, SAS for UNIX became the instrument of choice to analyze the data collected by xconslog. SAS offered the flexibility of being able to manage a whole different viewpoint from our data with a minimum amount of time and effort. With SAS/GRAPH®, we could also look into visualizing the data. The most difficult part turned out to be discovering the tricks needed to read in data from a raw C binary file.

The Challenge: Reading A Raw C Binary File

The xconslog application was written in the C language. In C, integers can be declared as a long (4-bytes on our Sun workstations) or short (2-bytes). Binary integers were easy to handle with the recommended SAS ibx. informat. Character strings, however, were another matter.

All C character strings terminate with a null byte. The variable actually holds one less character than in the char declaration to leave room for the terminating null character. All C character strings terminate with a null byte. The valid length of each string and successfully read the string are undefined.

Depending on what informat you use, SAS either terminates the record (observation) when it hits a null byte, or ignores it and includes all the garbage. We circumvented this problem by using the array and do over statements and the index function to determine the valid length of each string and successfully read the data into a dataset. The code for doing this follows:

data xconslog.xconsdat;
libname xconslog 'sodinfo/sas/sqm/files';
infile 'sodinfo/sas/sqm/files/xconslog.92Mar'
recfm=f lrecl=276 firstobs=2;
input $1 offset ib4. /* offset to notes file */
  len ib4. /* length of note */
  del $charl. /* delete flag */
  contact $charl. /* type of contact */
  done $charl. /* done flag */
  rpt $charl. /* report flag */
  site $charl. /* site flag */
  refer $charl. /* referral flag */
  flag2 $charl. /* unused */
  area $char21. /* subject area */
  user $char9. /* xconslog user's name */
  seq $char7. /* contact sequence. */
  tmstmp $char12. /* Entry timestamp */
  scinum $char5. /* Scientist number */
  list_str $char23. /* List string */
  log $char9. /* Contact logon name */
  rem $char9. /* Contact name */
  tme $char5. /* Time spent on contact */
  phn $char20. /* Contact phone number */
  cm1 $char39. /* Email address */
  loc $char27. /* location of contact */
  div $char6. /* Contact division */
  dte $char9. /* Contact enter date */
  d_dte $char9. /* done date */
  group $char15. /* unused */
  buffer $char12. /* unused */
if x=index(fields, "T") then delete;
/* Put char fields to be processed into an array */
array fields area user seq tma
  scinum list_str log rem tme phn
  eml tct div dte d_dte;
/* Find position of null byte in string and parse data up to it */
do over fields;
x = index(fields, "00*");
  if x >= 2 then fields=substr(fields, 1, x-1);
  else fields = " ";
end;

Dates were another area that needed special attention. The SAS mmddyy8. informat would read them correctly unless the contact was still open or a date was missing for some other reason. In that case, the data contained a null byte as the first byte, followed by eight bytes of garbage. The mmddyy8. informat would try to interpret the nonsense bytes as a valid date. To work around this, we read the dates in first as character data. If the date string was blank, we set a new SAS date variable to zero (date becomes 1/1/60), otherwise we used the input function to set the new variable to a SAS date value.

/* Convert character dates into SAS date values */
if dte = " " then open_dte = 0;
  else open_dte = input(d_dte, mmddyy8.);
if d_dte = " " then cls_dte = 0;
  else cls_dte = input(d_dte, mmdyy8.);

Results

The xconslog application provides basic monthly statistics on a per-consultant basis, — one file per consultant per area (consulting topic) — listing the number of general consulting contacts, contact time (broken down by area and division), and percentage of contacts that were open over a range of days. We are currently writing SAS programs to combine the individual xconslog data into a single source, provide the same statistics for the group, and provide six-month-comparisons for the group.

Analyzing user contact data is an ongoing process, and we continue to find new ways to measure quality. Areas we are looking at now include

- Where should we be placing our priorities for training (both user and consultant), as shown by analyses of number of contacts per area and time required to complete contacts (breakdown by area and consultant),

- Is there need to produce additional documentation or revise existing documentation for a certain area,
• Where are the problem areas and are they hardware or software related,

• How are we doing with respect to the percentage of open contacts and the time required for closure, and

• Are the data showing us there is need for additional staffing at a new support office just set up at another site within NCAR.

Trademarks

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. In the USA and other countries. ® indicates USA registration.

Sun is a trademark of Sun Microsystems, Inc.

UNICOS is a registered trademark of Cray Research, Inc.

X and X Window System are trademarks of the Massachusetts Institute of Technology