ALEPH 89-101
DATACQ 89-16
6 June 1989

Authors: A. Belk, D.R. Botterill, J. Harvey
J.Bourotte, J. Wotschack

Recommendations for the Use of Global Sections in Aleph Online Programming

Summary of the document

Recommendations are given of how to make use of the different methods of clusterwide updating of global sections.

Revision/Update Information: this is a new document

Contents

1 INTRODUCTION

Global sections are used to share data/constants/parameters between processes. This works fine between processes running on the same node. In our application there is a need to extend this facility over all nodes in the ONLINE CLUSTER. Unfortunately, VMS does not allow for that directly. Therefore code had to be written to provide a system which allows for clusterwide updating of global sections.

It looks as if there is no general and easy way of doing what we want. Each solution has some advantages and some disadvantages.

In general it seems that clusterwide updataing of global sections is not re-commended for quickly changing variables (e.g. scalers which count at a frequency of seconds or faster). These quantities should be organised such that no automatic clusterwide update is necessary.

This note is intended to give recommendations to the ALEPH ONLINE programmers on which method to use for which applications.

2 EXISTING CLUSTERWIDE UPDATING METHODS

Presently there exist two methods (and two sets of routines) to use global sections. These two solutions are complementary as we try to explain in the following.

2.1 AST Method

The clusterwide update is initiated through ASTs which force unmapping and remapping of all processes on the cluster which are CONNECTED to the global section. A more complete description of this method can be found in the ALEPH NOTE 88-205 / DATACQ 88-38.

This method is recommended for large global sections where the content does not need to be updated very frequently.

ADVANTAGES:

- Any changes of variables on one node are automatically available to processes on the other nodes. No particular action has to be taken by the user to make sure that he uses the latest values.
- Once the processes have been unmapped and remapped VMS will take care of swapping only thoses pages in/out which are addressed by the individual process.

DISADVANTAGES :

- Slow, uses quite a bit of CPU time.
- Each process in the cluster which is mapped to the section needs to be notified through an AST. The process which forces the update will be blocked until the AST are delivered to all processes.
- All processes which recieve the AST have to unmap and remap even if they do not make use of the variables in question. (At least at present) VMS does not deliver ASTs to processes which are suspended - if a process is suspended the access to this global section will get blocked.

- Frequent updates of global sections will result in large numbers of ASTs to be delivered.

2.2 QIO Method

The Clusterwide updating is done through writing the full global section to a disk file and then have it read back by the first program which uses the global section on each node, see ALEPH NOTE 89-97/ DATACQ 89-14.

The status of the global section for each node is kept in the lock-value-block of this global-section. Whenever the content of the section has changed (and this change is to be made clusterwide) a bit is set in the lock-value-block for all the nodes on which processes are mapped to the section. The first processes on each node which accesses data on the section will see the change, read the section from disk, and reset the bit.

This method is recommended for global sections up to a few hundred kbytes where the content needs to be updated more frequently.

ADVANTAGES:

- Fast, good performance.
- the global section in memory is only updated when needed and then only by the first process on the specific node which uses information from it. No ASTs are sent to all programs.

DISADVANTAGES:

- The change of variables is not automatically transmitted to the other nodes in the cluster but has to be done under user control (which is also an advantage in some cases).

3 APPLICATION SOFTWARE

Some software has been written to facilitate the use of the different methods.

3.1 AST Method

A software package exists to create INCLUDE FILES from DDL and the MAPPING ROUTINES. A system exists to store the global section content in ORACLE, to retrieve it from there, and to change table sizes. This has been described in the ALEPH NOTE 88-205 / DATACQ 88-38.

3.2 QIO Method

A software package has been written which is tailored to our partitioning scheme. It uses a small global section database into which global sections must be declared with their generic name, length and validity range (Detector-wide, Partition-wide, SubPartition-wide). A set of access-routines ensures that the process maps to the right global section running under ALEPH or in a Subdetector partition. This has been described in the note ALEPH NOTE 89-97/DATACQ 89-14.

4 RECOMMENDATIONS

Clusterwide updating of frequently changing variables should be avoided as much as possible. If it has to be done it needs to be done keeping the performance of the ONLINE VAX CLUSTER in mind.

The QIO Method is more performing for global sections up to a few hundred KByte size and tailored to the partitioning scheme. Therefore this method is recommended to be used to handle global sections which contain frequently changing variables and are not part of the aleph online database. These are PCT TABLES, PARTITION DEPENDENT GLOBAL SECTIONS, STATISTICS, etc...

The AST method has the advantage of distributing updated information automatically, it is, however, for most of the global sections which we use less performing than the QIO method. Since the automatic updating is a nice feature, it should be used (provided the problem of not delivered ASTs can be solved) for ONLINE DATABASE GLOBAL SECTIONS which are quasi static.