Dynamic Load Balancing of SAS® Batch Programs in a VMS® Cluster Environment

Rich Levin, Syntex Corporation, Palo Alto, CA

ABSTRACT

One of the perennial problems in SAS system management is distribution of workload among various hosts. Programmers typically are either assigned to only one of the hosts, or must decide on their own which host to use. This paper describes a method for dynamically balancing the load of SAS batch work among VAX® hosts in a cluster environment.

The technique is to run a representative SAS program on a regular basis, and to post a resulting performance metric on the host as a system-wide logical. Then, each SAS job references a SAS macro that finds the least heavily loaded host based on the performance metric, and submits the job to that host for execution using SAS/CONNECT® software. With some modification, the technique could be extended to environments based on other platforms, or to a heterogeneous environment.

INTRODUCTION

At Syntex, we have licenses for SAS Software on several VAX hosts. Balancing the SAS batch load on these hosts has traditionally been done on a project basis, and with only limited success. The problem is that the load changes constantly as programmers do their work. To compensate for these changes, a technique was developed that allows batch jobs to be run on the host with the lightest load regardless of the host on which the program was submitted. Further, this balancing occurs invisibly, and requires very little overhead. As of this writing, the technique has been fully developed and tested, but not put into production.

REQUIREMENTS

To implement the technique, the following items are required:

- Multiple VAX hosts set up in a cluster arrangement.
- The SAS System installed on more than one of these hosts, with the exact same product mix.
- SAS/CONNECT software.
- A DECNET® proxy for each person using SAS (proxies allow people to access other VAX hosts without the need to log in again).
- Batch queues on each host with sufficient depth to avoid excessive queue waiting time.
- A fairly representative SAS batch program that uses both CPU and I/O resources, but sparingly. This program serves as a benchmark, and must include the STIMER or FULLSTIMER options.
- SAS macro autocall library must be available on all hosts, and must contain FND_BEST.SAS.

FILES USED

File Name	Description
BENCHMK.COM	Control file that submits itself, runs the benchmark, runs the parser, and posts the results.
BENCH.SAS	The actual benchmark program (must include the FULLSTIMER option).
CALC.SAS	Program that parses the log of the benchmark for the total resource statistics.
PARMS.TXT	A list of VAX hosts to use for balancing.
FND_BEST.SAS	SAS autocall MACRO that finds the fastest VAX, and submits code to that host.
AUTO_BAL.SAS	An AUTOEXEC program that calls FND_BEST.SAS.

DETAILS OF THE TECHNIQUE

First, the Digital Control Language (DCL®) program BENCHMK.COM resubmits itself every 1/2 hour on each VAX host listed in PARMS.TXT. The program checks for the existence of a logical named SAS_VAXEN, that contains the list of hosts to use for balancing. If this logical does not exist, it is created by reading PARMS.TXT. The program is resubmitted only when the current VAX matches the first VAX in the list, and is submitted on all the other hosts (each lagged by 5 minutes).

Next, the program runs BENCH.SAS, and CALC.SAS. CALC.SAS creates a process symbol called SAS_ETIME (the value being the elapsed time for the entire BENCH.SAS job run) that is read by BENCHMK.COM and passed as a system wide logical called SAS_ETIME_hostname (where hostname is the name of this particular host). CALC.SAS also appends the results of this run to a SAS dataset that can later be used for performance measurement and capacity planning.

BENCHMK.COM then passes SAS_ETIME_hostname on to all the other hosts. A safety valve exists that checks the status of the command that passes the result for each of the other hosts. If this status is in error, the host is considered unavailable, and SAS_ETIME_hostname for that host is set arbitrarily high on all the other hosts.

Now, when programmers submit batch their jobs, they use the /AUTOEXEC=AUTO_BAL.SAS switch that forces FND_BEST to run when SAS is started.
At Syntex, we have a special DCL setup routine that is used to automatically create a SAS symbol that contains all the default switches that we want people to use when running SAS. When load balancing is implemented, the /AUTOEXEC=AUTO_BAL.SAS switch will be added to this setup routine for batch jobs. Thus, no changes are required by the person submitting the job.

Finally, the macro FND_BEST simply checks the current host for all the SAS_ETIME_hostname values, and checks to see which host has the fastest time. Once this is determined SAS/CONNECT is used to signon to this host and remote submit the rest of the program. Since we set up proxies for everyone and are using the DECNET communications method in SAS/CONNECT, no user ID or password is needed to perform this operation. Since batch processes terminate automatically, no ENDRSUBMIT or SIGNOFF statements are necessary.

LIMITATIONS

- The technique can only be used invisibly for batch jobs, since SAS/CONNECT will only remote submit the first step in an interactive environment.
- The benchmark program must be run often enough so that the performance information is fairly accurate at all times. The 1/2 hour time interval was chosen for Syntex arbitrarily for testing purposes. If the load changes based on the time of day, BENCHMK.COM could be changed to allow for smaller intervals during peak hours, and longer intervals during off hours.
- The BENCH.SAS program must be representative but nonintrusive. Remember, this program will be run often, and on every SAS host.
- Since load balancing is handled through SAS/CONNECT, SAS will run on two hosts for any given job most of the time (whenever the current host is not the best). This overhead should be minimal, but would prevent allowing everyone to submit their jobs on one controlling host.
- Batch programs that utilize SAS/CONNECT might not work properly.
- Batch SAS programs must not specify a different macro autocall library.
- Batch SAS programs must not turn off macro autocall checking.

HETEROGENEOUS ENVIRONMENTS

This technique could be extended to a heterogeneous environment, but only if the following hurdles are overcome:

- All hosts must use the same communications method (most likely TCP/IP).
- All hosts must have the ability to store values in memory that can be read system wide.
- All hosts must share common storage areas (probably NFS).
- All batch SAS programs must not use any host specific code.
- A method for writing FILENAME and LIBNAME statements in a general way must be developed.
- CALC.SAS and BENCHMK.COM will have to be written specifically for each type of host.
- Since DECNET will not be used, a way of passing the USER ID, and PASSWORD to the connect script must be found.

These hurdles are not trivial, but could be overcome. The more similar each of the hosts are, the less trouble the conversion becomes.

PERFORMANCE MEASUREMENT

The CALC.SAS program appends to a SAS dataset the elapsed time, the CPU time, and the ratio of elapsed time to CPU time for each run of BENCH.SAS. In other words, this SAS dataset contains information about the performance of SAS batch jobs at all times. This information could be used to find peak SAS hours, perform capacity planning, and to monitor the effectiveness of load balancing.

CONCLUSION

As we have seen this technique holds the promise to greatly improve the way SAS batch traffic is handled in a VAX cluster environment, and may be extended to a heterogeneous environment. The technique can be applied without changing existing SAS batch programs, and can be made virtually invisible. Although some overhead is required to implement the technique, in the long run it should be well worth the cost.

ACKNOWLEDGEMENTS

The following people contributed to the preparation of this paper:

Andy Cary
Jed Malitz

SAS, and SAS/CONNECT 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.