

## Paper 290-28

## Using the SAS® v 9 Application Response Measurement system to provide metrics to HP-UX Workload Manager

Mark Coffey, Hewlett-Packard Corporation, Richardson, TX

### ABSTRACT

There are two main problems that a system administrator faces when configuring goal-based resource management software:

1. Determining the priority of each workload running on the system
2. Collecting useful performance data from each workload

Determining the priority of each workload can be resolved by investigating the importance of the workload to the organization and the expected performance of that workload. The next obstacle, collecting the performance data, is going to be different for each workload running on the system. For SAS jobs, the Application Response Measurement (ARM) specification can be used to provide the necessary data. Using the SAS ARM system and the HP-UX GlancePlus library, HP-UX Workload Manager (WLM) can extract the appropriate data for determining the performance of SAS software.

### INTRODUCTION

HP-UX Workload Manager is a goal-based resource management application that allocates resources based on the priorities and goals of the workloads running on the system. The organization's dependence on a workload determines its priority, while the goals vary from workload to workload. For SAS jobs, the goal is to have the job complete within a certain amount of time. The main obstacle to achieving this goal is collecting the right performance metric to feed to WLM. Because the goal for a SAS job is desired duration, the metric should be a value that indicates duration. Such a metric can be collected from applications that have implemented the Application Response Measurement standard.

### APPLICATION RESPONSE MEASUREMENT

Hewlett-Packard Company and Tivoli Systems released the ARM v 1.0 specification in 1996. This first release provided a set of routines that an application could use to report its performance. These routines are based on the ARM API unit of work called a *transaction*. Each transaction is measured by the elapsed time, or wall time, required to complete the work. A transaction has a definite starting and stopping point, and it can be anything from a database query to processing the sum of two numbers. Even though these two transactions are entirely different in their functionality, they can both be measured by the amount of time they took to complete.

The six routines defined in the ARM specification are:

- `arm_init` - Initializes the necessary environment and registers the application with the monitoring agent. Use this routine once per application.
- `arm_getid` - Initializes a new transaction. Each new transaction requires this routine. One application with multiple transactions will use this routine several times.

- `arm_start` - Signals the beginning of a transaction. The tracking of the elapsed time starts when this call is made.
- `arm_update` - Updates the monitoring agent on the transaction's performance while the transaction is still in progress.
- `arm_stop` - Signifies the end of the transaction. The tracking of the elapsed time is stopped when this call is made.
- `arm_end` - Tells the monitoring agent the application has ended. Use this routine only once per application.

For more information on the ARM specification, visit the ARM Working Group website at <http://regions.cmg.org/regions/cmgarmlw> or the Open Group Enterprise Management Forum website at <http://www.opengroup.org/management>.

### USING THE SAS v 9.0 ARM SYSTEM

SAS v 8.2 was the first release of SAS to support the ARM specification. Support of the specification was continued and improved with SAS v 9.0. With version 9.0, the ARM functionality has been divided into four subsystems, each measuring a different aspect of SAS jobs. The option used to enable the SAS ARM subsystems is `ARMSUBSYS`. This option can be set to include the input/output system, IOM Server, `PROC` and `DATA` steps, and OLAP server. Monitor all the subsystems by setting `ARMSUBSYS` to `ARM_ALL`:

```
ARMSUBSYS=(ARM_ALL)
```

### SPECIFYING THE ARM AGENT

By default, the SAS ARM library handles all the ARM transactions and logs the data to a file. Using the `SAS ARMAGENT` option, another ARM library can handle the ARM transactions. Hewlett-Packard provides an ARM library in GlancePlus for HP-UX that is capable of handling the SAS ARM transactions and making the performance data available for monitoring using the GlancePlus tools.

In order to allow GlancePlus to be used for the ARM calls, the environment must first be prepared. Because SAS v 9.0 for HP-UX is a 64-bit application, the 64-bit ARM library must be used to handle the ARM calls. This library resides at `/opt/perf/lib/pa20_64/libarm.sl`. SAS v 9.0 has a restriction on the length of the ARM library name; it can only be eight characters long. With `libarm.sl` being nine characters long, another file or link must be used to represent the library. Using the `ln` command, a link should be created in `/opt/perf/lib/pa20_64`:

```
$ ln -s /opt/perf/lib/pa20_64/libarm.sl \
/opt/perf/lib/pa20_64/armsas64
```

This creates a softlink, `/opt/perf/lib/pa20_64/armsas64`, that points to `libarm.s1`. The name `armsas64` was chosen to describe why the link is there and because it is only eight characters long.

SAS must be able to find this link by looking in the directories listed in the `$LD_LIBRARY_PATH` environment variable, so `/opt/perf/lib/pa20_64` needs to be in the listing. Set `$LD_LIBRARY_PATH` using the `export` command:

```
$ export LD_LIBRARY_PATH=/opt/perf/lib/pa20_64
```

If the `$LD_LIBRARY_PATH` variable is already set, just append the new directory:

```
$ export LD_LIBRARY_PATH=\
$LD_LIBRARY_PATH:/opt/perf/lib/pa20_64
```

Now `$LD_LIBRARY_PATH` contains the necessary path to allow 64-bit SAS to find the GlancePlus ARM library. The `ARMAGENT` option must now be set to direct the ARM calls to the GlancePlus library:

```
ARMAGENT=armsas64
```

With the SAS ARM subsystems enabled and the GlancePlus library as the ARM agent, all the transaction processing information will be sent to GlancePlus where the user can monitor the performance.

## IDENTIFYING STEPS AND TRANSACTIONS

Using the subsystems registers all the `PROC` steps under one transaction in GlancePlus. While this is useful for monitoring overall `PROC` step performance, it does not allow for identifying and monitoring individual steps. Providing a finer granularity of performance data at the step level can be useful in identifying the most resource-intensive parts of a job.

Each step in a SAS job can be represented by a unique transaction using macros provided by SAS. These ARM macros must be inserted in the SAS job around each step where the performance is to be measured. In doing this, an informal transaction group represents the performance of the SAS job.

If the user is going to insert ARM macros into the job, the `ARMSUBSYS` declaration is not needed. However, another variable, `_armexec`, must be set:

```
%let _armexec = 1;
```

The ARM macros available for use in a SAS job are:

- %arminit
- %armgtid
- %armstrt
- %armupdt
- %armstop
- %armend

These macros correspond in name and functionality to the six ARM routines described earlier. In the following example, a `DATA`

step and two `PROC` steps are each wrapped by an ARM transaction and will be available for individual monitoring in GlancePlus. The `%arminit` and `%armend` macros are each used once in the job. The `%armgtid`, `%armstrt`, and `%armstop` macros are used once for each transaction. The `%armupdt` macro is not used in this example, but it can be used inside transactions to provide updates on the performance.

```
/* Set the _armexec variable. */
%let _armexec = 1;

/* Initialize the ARM system and
start the first transaction. */
DATA _null_
  %arminit(appname="sas_job");
  %armgtid(txnname="sas_job_txn_1");
  %armstrt;
run;

/* DATA step definition. */
DATA sample;
.
.
run;

/* Stop the first transaction and
start the second transaction. */
DATA _null_
  %armstop;
  %armgtid(txnname="sas_job_txn_2");
  %armstrt;
run;

/* PROC step definition*/
PROC reg data=sample;
.
.
run;

/* Stop the second transaction and
start the third transaction. */
DATA _null_
  %armstop;
  %armgtid(txnname="sas_job_txn_3");
  %armstrt;
run;

/* PROC step definition*/
PROC logistic data=sample;
.
.
run;

/* Stop the third transaction
and end the ARM processing. */
DATA _null_
  %armstop;
  %armend;
run;
```

By default, all the SAS ARM macros must be inside a `DATA` statement. Parameters can be passed to the macros allowing them to be outside a `DATA` statement.

## SENDING ARM METRICS TO WLM

After all the appropriate steps have been followed, the GlancePlus tools are now capable of tracking the transactions. Using the GlancePlus `glance` tool and the `adviser_only`

option, metrics about a specific transaction or a group of transactions can be extracted and passed to WLM to be used in calculating the resources granted to a SAS job.

There are a couple of items that have to be determined to be able to send useful transaction performance data to WLM. The first item that must be determined is the desired duration for each transaction. This value is set by a user and is based on previous performance data and any service-level agreements that may be in place. For example, a particular PROC step may normally take 20 seconds to complete on an idle system. The goal for this PROC step on an active system may be 30 seconds because the step is not expected to complete as fast on a system that has other workloads running.

The second item is a WLM metric goal for the workload. The performance of the transaction group of a SAS job will be represented by a single metric reported to WLM. The `glance` tool can return the actual duration of a single transaction. Using this value, and the desired duration for each transaction, a metric can be calculated that is the actual transaction duration divided by the desired transaction duration. The resulting metric represents how well the transaction performed against its goal. A value greater than 1 means the transaction is exceeding the desired transaction duration time, and a value less than 1 means the transaction is completing faster than the desired duration. Based on this, the goal of the WLM metric is to be less than 1.

The table below represents a transaction group that consists of three transactions. The table lists the transaction goals, actual durations, and the resulting metrics that would be sent to WLM.

Transaction	Goal (seconds)	Actual (seconds)	WLM Metric
TXN_1	30	45	1.5
TXN_2	60	54	0.9
TXN_3	45	45	1.0

After TXN\_1 completes, the WLM metric is 1.5. This is greater than the WLM metric goal of 1. Because the metric goal is not being met, WLM will increase the amount of resources allocated to the workload. The next transaction, TXN\_2, completes and the metric goal was achieved. WLM may reduce the amount of resources granted to the workload if other workloads are not meeting their goals and therefore requesting more resources. The third transaction, TXN\_3, completes on time, and the metric goal is met. Once the SAS job ends, the transactions are idle until the job is run again.

## CONCLUSION

Using the ARM functionality in SAS v 9.0 is an effective method to provide performance metrics to HP-UX Workload Manager for SAS jobs. The metrics provide enough granularity to allow WLM to adjust the workload's resources enabling the job to complete within the desired duration.

## REFERENCES

- Application Response Measurement Working Group  
<http://regions.cmg.org/regions/cmgarw>
- Open Group Enterprise Management Forum  
<http://www.opengroup.org/management>

## ACKNOWLEDGEMENTS

The author would like to thank the SAS Institute for their help in developing this paper.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged.

### Contact the author at:

Author:	Mark Coffey
Company:	Hewlett-Packard
Street:	3000 Waterview Pkwy
City, State, Zip:	Richardson, TX 75080
Country:	USA
Work Phone:	+1 972 497 3088
Fax:	+1 972 497 3123
Email:	mark_coffey@hp.com

### HP Workload and Resource Management Division:

Email:	<a href="mailto:wlmfeedback@rsn.hp.com">wlmfeedback@rsn.hp.com</a>
Web:	<a href="http://www.hp.com/go/wlm">http://www.hp.com/go/wlm</a>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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