
Optimizing SAS Software Performance on Sun Fire 15K and 12K Servers



Who Should Care About This Paper:

SAS 8.2 or higher users with large workloads running on Sun Fire 15k & 12k Servers

Overview

Sun and SAS Engineering teams have spent many years working closely together to optimize the performance of SAS products on Sun Microsystems Enterprise Class Server line (Solaris / SPARC). During the last year the teams have spent a lot of time focusing on the largest Sun systems looking for ways to optimize the performance of SAS. Particular focus has been placed on the flagships of Sun's server line, the Sun Fire 15k (expandable to 106 CPUs) and Sun Fire 12K servers (expandable to 52 CPUs).

For IT shops that have a large SAS user base running on a multiple system board domain in a Sun Fire 15K or Sun Fire 12K there are some system tuning suggestions that could significantly enhance SAS performance.

Details of these suggestions are provided below. If you have questions on this document or need more background on the Sun / SAS relationship, please contact one of the technical staff in the contacts section of this document.

Details

The Sun Solaris Operating System does an excellent job of automatically utilizing all of the CPUs in the system as application workloads increase. Processes get scheduled on free CPUs as they are created. As processes run and get rescheduled an attempt is made to put them on a CPU close to their assigned physical memory location. However, due to other applications and reduced resource availability during execution, over time processes can be scheduled across multiple CPUs and system boards. On larger Sun servers this doesn't always allow for optimal physical placement of the process and its resources which can affect the overall application's performance.

There are two different toolsets that can be used on Sun Solaris to help optimize physical placement and prevent application "floating" which can reduce overall performance:

- (1) Processor Sets (covered in more detail below):

Allows the administrator to limit application execution to a predefined set of CPUs. Included in Solaris 8 and higher.

- (2) Solaris Resource Management (SRM):

A more comprehensive resource allocation tool that can be used to limit application execution and resource utilization across CPUs and memory. Under the covers it leverages the processor set

ability of Solaris along with other features to manage resources. SRM is a more complex resource management tool. Details of its implementation and features can be found at <http://docs.sun.com/> (Solaris 9 required for advanced processor set features of SRM)

Example: Running SAS with Solaris Processor Sets:

Use mpstat and psrset to create processor sets:

```
# mpstat                               ←-- Shows each CPU and their activity
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
 2  18  1  396    9  4   70  5  12  37   0  73  5  1  1  94
 3  14  1  328    5  0   67  4   9  39   0  59  3  1  1  96
 0  0  0  38     236 136   0  0  0  5   0  0  0  0  0 100
 1  0  0  25     2  1   0  0  0  0   0  0  0  0  0 100
 8  20  1  360    8  5   55  2  10  25   0 101  5  3  1  92
 9  15  1  268    4  1   48  2   9  24   0  68  3  5  1  92
10  15  1  309    5  1   51  2  10  29   0  75  3  6  1  90
11  14  1  263    9  6   49  2  10  29   0  70  2  6  0  91
#

# psrset -c 0 1 2 3                    ←-- This creates a new processor set
# psrset -c 8 9 10 11                  (be sure CPUs are on the same
                                        system board). The "-d" option
                                        can be used to delete sets.

# psrset -i                             ←-- This displays a list of processor sets
user processor set 1: processors 0 1 2 3
user processor set 2: processors 8 9 10 11
#

** for optimal performance it is best to allocate CPUs from
the same system board to a processor set.
```

Sample Script Used to Launch Batch Jobs on Different Processor Sets:

```
#!/bin/sh
echo "run five simultaneous sas batch scripts"
echo ""

time psrset -e 1 ./sas_batch_payroll.sh &
time psrset -e 1 ./sas_batch_accounts_receivable.sh &
time psrset -e 1 ./sas_batch_hr.sh &
time psrset -e 2 ./sas_batch_warehousebuild.sh
wait
```

* It is important to note that root access would be required to utilize and create processor sets unless a system utility such as "sudo" was put into place. An advantage of SRM (not discussed in detail here) is that it only requires root access for initial setup. However it is a more complex resource allocation tool and would require a potential system management philosophy change.

Useful Resources:

- See <http://www.sas.com/partners/directory/sun/papers.html> for other whitepapers on Sun / SAS Tuning
- <http://docs.sun.com/> online Sun Documentation and Man Pages (Solaris Resource Manager, psrset, mpstat, etc..)
- Information on Sudo can be found at: <http://www.courtesan.com/sudo/>

Author and Other Contact Information:

Tom Keefer (Author)
Sun Technical Platform Manager
SAS Institute
Email: Tom.Keefer@sas.com
Phone: 919-531-0850

Maureen Chew
Sun Market Development Engineer
Sun Microsystems Inc.
Email: Maureen.Chew@sun.com
Phone: 919-531-5852

Special thanks to Akiko Marti of the Sun Microsystems Oregon Benchmark Facility and Leigh Ihnen of the SAS Customer Technology Center for their assistance and knowledge we have leveraged during this effort.