Dynamic Reconfiguration and Oracle 9i Dynamically Resizable SGA

Erik Vanden Meersch, Global Sales Organization

Kristien Hens, Global Sales Organization

Sun BluePrints™ OnLine—January 2004

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300

Part No. 817-5209-11
Revision 1.0, 2/17/04
Edition: January 2004

Dynamic Reconfiguration and Oracle 9i Dynamically Resizable SGA

UNIX® has established its place in the data center, supporting the most mission-critical processes in any enterprise. Sun Microsystems has played a significant and pioneering role in this evolution by introducing support for server partitioning, on-line hardware servicing, and online partition resizing.

The Solaris™ 8 Operating Environment (Solaris™ 8 OE) and higher includes enhanced software support for these dynamic platform capabilities. Today, many major software vendors use this software in their products.

This Sun BluePrints™ OnLine article explains how Oracle 9i can operate in combination with Sun's dynamic reconfiguration (DR). It provides a brief overview of DR, intimate shared memory (ISM), dynamic intimate shared memory (DISM), and dynamically resizable system global area (SGA), and explains how these technologies fit together. In addition, this article provides step-by-step details for configuring Oracle relational databases on Sun Fire™ servers so that the DR capabilities of the Sun platform can maximized.

Note – The features described in this article should be used with the Solaris 9 Operating System Update 2 and newer versions.

This article presents the following topics:

Dynamically Reconfiguring UNIX Systems

Dynamic reconfiguration is the capability to remove from or add system resources (for example, CPUs, memory, and I/O cards) to a system without having to halt the operating system or the applications running on it. This feature was first introduced in 1997, and it is one of the factors that has contributed to Sun's success in the mission-critical arena.

In recent years, all major UNIX vendors have developed similar functionality. For example, IBM developed lpars for use on their p-series servers, and HP developed npars and vpars for use on the SuperDome.

In this article, we do not provide a detailed comparison of all UNIX server partitioning technologies that are available in the market today. Instead, we highlight two distinct advantages of Sun's server partitioning technology. (Note that on a Sun server, a partition is referred to as a domain.)

- First, Sun's server partitioning technology is based on the hardware modularity of the Sun Fire servers and it guarantees the highest possible isolation between domains. Because the failure of a component in a module (for example, the failure of CPU, a memory board, or an I/O board) has been proven to impact only the domain that is using the module, other domains are unaffected by the failure. They stay up and are in no way aware of the fault.
- Second, the technology is mature. While dynamic reconfiguration is a technology that can now be used in a production environment, it took several releases of the Solaris Operating System (Solaris OS) to reach this level of reliability. Companies that are just now announcing dynamic partitioning will lack this experience.

Using Dynamic Reconfiguration on Sun Fire Servers

Sun Fire servers are built with two types of boards: CPU/memory boards and I/O boards (with a number of PCI slots). Both of these boards connect to a central data and address switch. You can create a server partition, called a domain, by grouping at least one CPU/memory board and at least one I/O board. You can then reconfigure the interconnect to allow communication between the boards only in the same domain. Once this configuration is done, the boards are isolated from boards in any other domain.

A domain behaves in every respect as a standalone system. Once the domain is created, there is no software or hardware overhead on its operation because it is embedded in a larger chassis (unlike hypervisor-based systems).

On Sun Fire servers, every possible domain is referred to by a one-letter name (for example, “A, B, C, and D” on a Sun Fire 6800 server, which has four possible domains). A domain does not need to be created or removed. The only operations are adding a board to a domain and removing a board from a domain. Using dynamic reconfiguration, both of these tasks can be done while the Solaris software is running on the domain.

Dynamic Intimate Shared Memory

Most applications running on a domain will immediately benefit from dynamically added resources (CPU and memory). The Solaris OS immediately starts scheduling processes on new CPUs and distributes the new memory pages to processes that request them. No administrator action is required.

Applications that use a shared memory segment are an exception. These applications confine their memory usage to this segment. To enable applications to use a dynamically adjustable amount of shared memory, version 8 and later of the Solaris OS support dynamic intimate shared memory (DISM).

A DISM segment can be created with a size that is larger than the amount of physical memory in the system. If this is the case, the segment cannot be locked in memory at creation time. A process that makes use of a DISM segment can lock and unlock parts of the segment while the process runs. By doing so, the application can effectively react to the addition of physical memory to the system or prepare for removal of memory.

In the following section, we show how Oracle 9i can be configured to use DISM.

Dynamically Resizing the SGA in Oracle 9i

All Oracle processes that make up a database instance share one large memory segment called the system global area (SGA). Internally, chunks of this memory are used for distinct purposes, like the redo log buffer and the shared pool.

With the release of Oracle 9i, it is possible to change the size of these components without having to shut down the instance, as long as the sum of their sizes does not exceed the size of the initially created SGA. This initial size cannot be changed without shutting down the database instance.

With DISM, however, the SGA can be created larger than the amount of physical memory in the system, so that memory added to the system can later be used by the running database instance.

To make optimal use of the dynamic characteristics of the Sun Fire platform, we recommend that you configure the SGA to use DISM. As we will show in the next section, you can do this by configuring the SGA to be larger than the sum of its components, which triggers Oracle 9i to use DISM instead of ISM. The size of the SGA should be set to the maximum memory size that you expect the domain might grow to in the future.

Performance wise, it is best for the SGA to allocate one big DISM segment, rather than several smaller ones. You accomplish this by setting the Solaris kernel parameter `shmmax` to a very high value, for example, 4 gigabytes on a 32 bit system, possibly much higher on a 64bit system.

Even if the domain is already configured at its maximum number of system boards, we recommend that you configure the SGA to use DISM. This will enable you to later remove one or more system boards without having to stop database instances on the domain.

There is a small price to pay for the flexibility and availability gained by dynamically resizing the SGA. Every virtual memory segment that is not mapped to a file on the disk requires virtual memory reservation. This is also the case for a DISM segment, and it implies that the system needs to be configured with a swap device that is at least the size of this segment.

Configuration Example

In the example presented in this section, we use a Sun Fire 4800 server with two boards, each populated with two CPUs and one gigabyte of memory. The initial Oracle domain uses only one board. The maximum amount of physical memory a Sun Fire 4800 server can hold is 12 gigabytes; therefore, we will increase the maximum size of the SGA to 12 gigabytes.

The following code sample shows the default Oracle configuration.

```
SQL> startup
ORACLE instance started.

Total System Global Area  303531784 bytes
Fixed Size                   730888 bytes
Variable Size             285212672 bytes
Database Buffers           16777216 bytes
Redo Buffers                 811008 bytes
Database mounted.
Database opened.
```

When inspecting the address space of one of the database engine processes, as shown in the following sample, you can see that the SGA is created as an ISM segment.

```
$ pmap 386
386:    ora_pmon_SGA
0000000100000000     53880K r-x--  /unused/oracle/OraHome1/bin/oracle
000000010359C000      1048K rwx--  /unused/oracle/OraHome1/bin/oracle
00000001036A2000      1936K rwx--    [ heap ]
0000000380000000    315392K rwxsR    [ ism shmid=0x64 ]
FFFFFFFF7C400000         8K rw--R    [ anon ]
FFFFFFFF7C410000         8K rw--R    [ anon ]
```

As shown in the following sample, the SGA is increased to 12 gigabytes by setting the instance parameter `SGA_MAX_SIZE`. This static parameter needs to be set upon initial configuration of the instance because it requires you to shut down and restart the database. To do this, you need to configure a sufficiently large amount of swap space. (12 gigabytes plus the swap space reserved by all other processes on the system. Note that this is reported as having a status of "used" by the command `swap -s`.) If you do not configure a large enough swap space, the Oracle instance will fail to start.

The following list outlines the different steps we will take in the configuration example:

1. When you initially configure the instance, make sure that the instance parameter `SGA_MAX_SIZE` is as big as the size you eventually want your instance to be. This parameter has to be backed up by at least an equal amount of virtual memory (physical memory plus swap space).

2. Add a second system board using dynamic reconfiguration.

3. Now, grow the instance parameter `db_cache_size` while the instance is running.

4. To shrink the size of an SGA component, change the instance parameter on the running instance. If you want to remove the system board, you can do so at this time.

```
SQL> ALTER SYSTEM SET  sga_max_size=12g SCOPE=spfile;

System altered.

SQL>SHUTDOWN IMMEDIATE
SQL>STARTUP

ORACLE instance started.

Total System Global Area 1.2886E+10 bytes
Fixed Size                   748984 bytes
Variable Size            1.2868E+10 bytes
Database Buffers           16777216 bytes
Redo Buffers                 811008 bytes
Database mounted.
Database opened.
SQL>
```

As shown in the following sample, the address space now shows a 12 gigabyte DISM segment.

```
# pmap 368
368:    ora_pmon_SGA
0000000100000000     53880K r-x--  /unused/oracle/OraHome1/bin/oracle
000000010359C000      1048K rwx--  /unused/oracle/OraHome1/bin/oracle
00000001036A2000      1520K rwx--    [ heap ]
0000000380000000   12603392K rwxs-    [ dism shmid=0x12c ]
FFFFFFFF7C400000         8K rw--R    [ anon ]
FFFFFFFF7C410000         8K rw--R    [ anon ]
FFFFFFFF7C450000         8K rw--R    [ anon ]
```

After increasing the size of the SGA, you are ready to dynamically add the second system board.

The following sample shows the initial system configuration.

```
sf4800a[523]# prtconf | head
System Configuration:  Sun Microsystems  sun4u
Memory size: 1024 Megabytes
System Peripherals (Software Nodes):

sf4800a[524]# psrinfo
18      on-line   since 10/31/2003 16:54:12
19      on-line   since 10/31/2003 16:54:13
```

As shown in the next sample, the second board is added using the `cfgadm` command.

```
sf4800a[525]# cfgadm -c configure N0.SB0
{/N0/SB0/P2} Running CPU POR and Set Clocks
{/N0/SB0/P3} Running CPU POR and Set Clocks
{/N0/SB0/P2} @(#) lpost         5.15.0  2003/04/14 16:54
{/N0/SB0/P3} @(#) lpost         5.15.0  2003/04/14 16:54
{/N0/SB0/P2} Copyright 2001-2003 Sun Microsystems, Inc.  All rights
reserved.
{/N0/SB0/P3} Copyright 2001-2003 Sun Microsystems, Inc.  All rights
reserved.
{/N0/SB0/P2} Use is subject to license terms.
{/N0/SB0/P3} Use is subject to license terms.
{/N0/SB0/P2} Subtest: Setting Fireplane Config Registers for aid 0x2
{/N0/SB0/P2} Subtest: Display CPU Version, frequency
{/N0/SB0/P3} Subtest: Setting Fireplane Config Registers for aid 0x3
{/N0/SB0/P2} Version register = 003e0014.54000507
{/N0/SB0/P2} Cpu/System ratio = 5, cpu actual frequency = 750
{/N0/SB0/P3} Subtest: Display CPU Version, frequency
{/N0/SB0/P3} Version register = 003e0014.54000507
{/N0/SB0/P3} Cpu/System ratio = 5, cpu actual frequency = 750
{/N0/SB0/P2} Running Basic CPU
{/N0/SB0/P3} Running Basic CPU
{/N0/SB0/P2} @(#) lpost         5.15.0  2003/04/14 16:54
{/N0/SB0/P3} @(#) lpost         5.15.0  2003/04/14 16:54
{/N0/SB0/P2} Copyright 2001-2003 Sun Microsystems, Inc.  All rights
reserved.
{/N0/SB0/P3} Copyright 2001-2003 Sun Microsystems, Inc.  All rights
reserved.
{/N0/SB0/P2} Use is subject to license terms.
{/N0/SB0/P3} Use is subject to license terms.
{/N0/SB0/P2} Subtest: I-Cache Initialization
{/N0/SB0/P2} Subtest: D-Cache Initialization
{/N0/SB0/P2} Subtest: W-Cache Initialization
{/N0/SB0/P3} Subtest: I-Cache Initialization
```

After the dynamic configuration operation, the system configuration appears as follows.

```
sf4800a[528]# prtconf | head
System Configuration:  Sun Microsystems  sun4u
Memory size: 2048 Megabytes
System Peripherals (Software Nodes):

sf4800a[528]# psrinfo
2       on-line   since 10/31/2003 17:03:31
3       on-line   since 10/31/2003 17:03:36
18      on-line   since 10/31/2003 16:54:12
19      on-line   since 10/31/2003 16:54:13
```

As shown in the following sample, you can now change the size of the database buffer cache from 16 megabytes to 1.5 gigabytes without restarting the instance.

```
SQL> alter system set db_cache_size=1500M;

System altered.

SQL> show sga

Total System Global Area 1.2886E+10 bytes
Fixed Size                   748984 bytes
Variable Size            1.1308E+10 bytes
Database Buffers         1577058304 bytes
Redo Buffers                 811008 bytes
```

You can also dynamically shrink the size of the SGA components and then remove the system board that was just added. The following command shrinks the size of the database buffer cache back to 16 megabytes.

```
SQL> alter system set db_cache_size=16M;

System altered.

SQL> show sga

Total System Global Area 1.2886E+10 bytes
Fixed Size                   748984 bytes
Variable Size            1.2868E+10 bytes
Database Buffers           16777216 bytes
Redo Buffers                 811008 bytes
```

The following set of commands remove the system board and show the systems configuration after this operation.

```
sf4800a[545]# cfgadm -c unconfigure N0.SB0

sf4800a[547]# prtconf | grep -i mem
Memory size: 1024 Megabytes
```

About the Authors

Erik Vanden Meersch is a Technology Expert for Sun in Northern Europe, focusing on servers and the Solaris product. Erik is often involved in designing architectures for environments with high-availability requirements. With Sun for six years, Erik entered the company as a Solaris software instructor. He has a degree in electronics engineering, and his career, prior to joining Sun, includes research on algorithms for Very Large Scale Integration (VLSI) design, software engineering, and CAD support.

Kristien Hens is currently working as a Technology System Engineer, with focus on storage and cluster. For four years before joining GSO, Kristien worked for Sun Educational Services where she taught storage and Sun Cluster courses. In addition, she developed and taught the courses “Oracle 9i for Sun Engineers” and “Oracle 9i RAC on Sun Cluster 3.x Workshop.” Prior to joining Sun, Kristien worked as a developer in the field of computational linguistics.

Acknowledgements

The authors would like to recognize Dirk Demuynck for his continuous support of their efforts.

Ordering Sun Documents

The SunDocs^{SM} program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun Documentation Online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is `http://docs.sun.com/`

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: `http://www.sun.com/blueprints/online.html`