ORACLE
VM

An Oracle Technical White Paper
March 2014

Increasing Application Availability by Using the Oracle VM Server for SPARC Live Migration Feature: An Oracle Database Example

This paper describes how to configure an Oracle Solaris 11 server and a storage array so you can migrate a running Oracle database from one system to another system without interruption. This paper also includes an overview of the live migration process and instructions for performing a live migration.

Introduction

This paper describes how to take advantage of the live migration capability in the Oracle VM Server for SPARC software to migrate a running single instance of Oracle Database from one system to another system without interruption.

Note: Oracle Database 11*g* Release 2 that runs on an Oracle VM Server for SPARC system has been certified with the Oracle Solaris 10 OS and the Oracle Solaris 11. This paper used the Oracle Solaris 10 9/10 OS and the Oracle Solaris 11 OS to demonstrate the live migration feature of Oracle VM Server for SPARC. This was originally demonstrated using Oracle VM Server for SPARC 2.2 and is upwards compatible, including with the release current at this writing, 3.1.

Fundamentals

The following describes fundamental concepts, software, and features used to create the required system configuration in this paper.

- **Downtime.** When discussing high availability, you must distinguish between planned downtime and unplanned downtime.

  Planned downtime is a preventive or corrective maintenance task that interrupts day-to-day system operations. This type of downtime is the result of maintenance events, such as repair, backup, or upgrade operations. Be sure to schedule this downtime to minimally impact the availability of a system, because even a minor modification to subsystems can cause a disruption to availability and affect business needs.

  Unplanned downtime occurs as the result of hardware or software faults and environmental events. It is paramount to resolve such downtime immediately. Unplanned downtime is not addressed in this paper.

- **Oracle Database.** Oracle Database is the leading database application and is frequently used on servers. By using virtualization and consolidation, you can run Oracle Database in selected, certified virtual environments, such as Oracle VM Server for SPARC.

- **Oracle VM Server for SPARC.** Oracle VM Server for SPARC is supported on a broad range of SPARC platforms.

  Oracle VM Server for SPARC enables you to create multiple virtual systems on a single physical system. Each virtual system is called a *logical domain*, or domain. Each domain runs its own instance of the Oracle Solaris 10 OS or the Oracle Solaris 11 OS. Oracle VM Server for SPARC also enables you to make fine-grained assignments of CPU and memory resources to a workload. You can also use the Oracle VM Server for SPARC to migrate a running domain from one system to another system, which minimizes the disruption to running applications and provides a powerful platform for managing production environments.

The version of the Oracle Solaris OS software that runs on a guest domain is *independent* of the Oracle Solaris OS version that runs on the control domain. So, if you run the Oracle Solaris 10 OS in the control domain, you can still run the Oracle Solaris 11 OS in a guest domain, and if you run the Oracle Solaris 11 OS in the control domain, you can still run the Oracle Solaris 10 OS in a guest domain.

- **Live Migration.** Starting with the Oracle VM Server for SPARC 2.1 release, you can migrate an active domain from one system to another system while the domain continues to run. Live migration uses SSL to encrypt logical domain memory structures as they are securely transferred over the network. This protects sensitive information from unauthorized access while memory contents are in transit.

  The host where the migration is initiated is called the *source machine* and the host to which the domain is migrated is called the *target machine*. While the migration is in progress, the domain created on the target machine is referred to as the *migrated domain*.

  For information about the live migration requirements and restrictions for CPUs, memory, and I/O on the source and target machines, see "Live Domain Migration Requirements" in the .*Oracle VM Server for SPARC 3.1 Release Notes*

The live migration feature benefits the following areas:

System maintenance—Use live migration to minimize downtime. Note that Oracle VM Server for SPARC has availability features, such as multiple service domains, that permit systems to remain in operation during many maintenance procedures. However, if you must shut down equipment, such as when updating firmware, use live migration to move running applications to another server without interruption of service.

Hardware resource upgrade—Use live migration to move an active domain to a machine with more capacity: more physical memory, more or faster CPUs, or a better I/O subsystem. Such a change improves application performance by keeping in-memory structures warm, rather than having an application restart and reload its data structures in memory. For example, after a live migration, Oracle Database still has its System Global Area (SGA) and other memory structures laid out the same way as before the migration without having to rebuild them or reload them from disk. An alternative to moving important domains to a server with more capacity is to free up resources by live migrating less important domains. Another method is to dynamically reallocate resources to move CPU and memory resources from less to more critical domains. That can sometimes be faster and simpler than migrating domains.

Long computations—You can maximize application availability during a migration because there is no need to shut down an application while the migration is in process. For example, when migrating applications that perform long computations, such as financial data analysis or scientific computations, you do not need to restart from the beginning or perform intermediate data save and restore checkpoint procedures.

Better return on investment—You can use live migration to move applications from older to newer SPARC systems as part of a technology refresh or upgrade. Existing guest domains can be consolidated onto a single system to reduce power, space, and cooling, while improving performance. You can migrate a guest domain from a system with an older CPU architecture, such as an UltraSPARC T2 Plus or a SPARC T3, to a system with a SPARC T4 CPU by using the cross CPU migration feature. Domains on SPARC T4, SPARC T5, SPARC M5-32 and SPARC M6-32 can also be live migrated between each of those servers. See "Cross CPU Migration" for further details.

Hardware and Software Requirements

Before you begin, ensure that your software and hardware meet the following requirements.

Oracle Database Requirements

To run an Oracle Database single-instance configuration on an Oracle Solaris 10 Oracle VM Server for SPARC system, you must use Oracle Database 11*g* Release 2 (Oracle 11.2.0.1) or later. For the Oracle Solaris 11 OS, you must use Oracle Database 11*g* Release 2 (Oracle 11.2.0.3) or later. To verify that a particular version is supported and to check for additional requirements, see the Oracle Database documentation. Also see My Oracle Support.

Hardware Configuration

Any server that supports Oracle VM Server for SPARC can be used with Oracle Database. Additional hardware, such as external storage arrays or network switches, might be required based on the selected configuration and deployment variant.

Figure 1 shows a typical hardware configuration and cabling that are compliant with the live migration feature. This paper is based on the following hardware configuration:

- Two Oracle SPARC T4-1 servers that each have 64 Gbytes of memory and one SPARC T4 processor.
- Eight CPU cores per processor and eight CPU threads per core. Thus, each system has 64 CPU threads (8x8). One internal disk is used on each server as the system disk of the control domain.
- One Sun Storage 2540-M2 storage array from Oracle that has 16 300-Gbyte SAS disk drives. The storage array is connected to both servers, and each server is connected to the same array controller (A or B) to which LUNs are mapped. The storage array provides one LUN (LUN 0) that is used as the system disk of the guest domain and a second LUN to store the Oracle Database data (LUN 1).
- Two 4-Gbyte FC-AL host bus adapters (HBAs), one for each server. These HBAs are used to connect the storage array to both servers.
- Four on-board, 1-Gbit Ethernet network interfaces on each server, which are used for network communications.

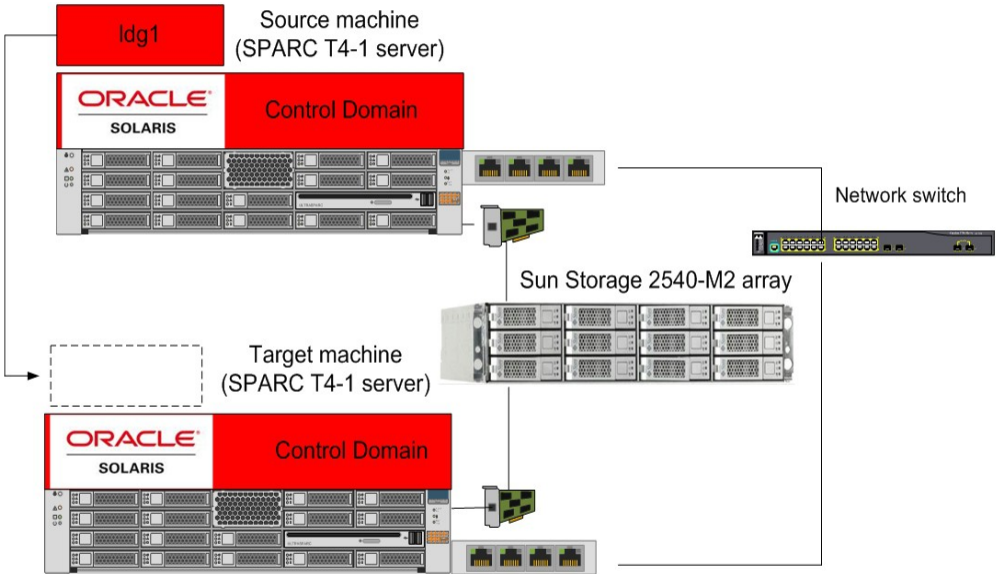

Figure 1--Hardware Configuration and Cabling

Running Oracle Database on Oracle VM Server for SPARC

Each guest domain that runs Oracle Database should have a minimum of 16 virtual CPUs (2 CPU cores) and 4 Gbytes of memory. These CPU and memory allocations are minimal for illustration purposes; your environment may require larger settings.

Each control domain that provides virtual devices (virtual disk and virtual network) to an Oracle single-instance guest domain should have a minimum of 8 virtual CPUs (1 CPU core) and 4 Gbytes of memory. These CPU and memory allocations are minimal for illustration purposes; your environment may require larger settings.

Configuration Details

In the example presented in this paper, the source server is configured with the control domain and one guest domain (named `ldg1`) that hosts the Oracle database. The control domain, named `primary`, is also an I/O domain. Furthermore, the control domain is used as a service domain that provides virtual device services (virtual disk and virtual network) to the guest domain.

To act as a service domain, the control domain provides the following virtual device services:

- One virtual disk service (`primary-vds0`) exports physical disks as virtual disks to the guest domain.
- One virtual console concentrator service (`primary-vcc0`) provides access to the virtual console of the guest domain. This service uses the port range 5000–5100 for access to virtual consoles.
- One virtual switch service (`primary-vsw0`) is associated with the primary network interface (`nxge0`).

Software Installation and Configuration

The following sections describe how to configure your environment for live migration.

Installing the Oracle Solaris OS

The Oracle Solaris OS comes pre-installed on new SPARC servers. For these systems, ensure that the appropriate release of Oracle Solaris is installed and verify that required patches have been applied.

To reinstall the Oracle Solaris OS on a SPARC system, see the Oracle Solaris 10 documentation or the Oracle Solaris 11 documentation. In this paper's example, the operating system is installed on the first internal disk of the server.

After the Oracle Solaris OS is installed, you can configure and enable the Oracle VM Server for SPARC software.

Installing the Oracle VM Server for SPARC Software

The Oracle VM Server for SPARC software comes pre-installed on new SPARC servers running Oracle Solaris 11. If the Oracle Solaris 10 OS has been installed on the server, you might have to install the Oracle VM Server for SPARC software. In either case, go to the Oracle VM Server for SPARC 3.1 Administration Guide - Installing and Enabling Software. That section details the steps needed to ensure the Oracle VM Server for SPARC software is installed, as well as necessary firmware levels for the server.

Configuring the Storage Array

The Sun Storage 2540-M2 storage array is connected to the control domain, which exports two LUNs (LUN 0 and LUN 1) as virtual disks to the guest domain.

The guest domain uses the first LUN (LUN 0) as the system disk for the operating system and the second LUN (LUN 1) as a disk to store Oracle Database data files. Figure 2 shows the storage layout of the source machine and target machine.

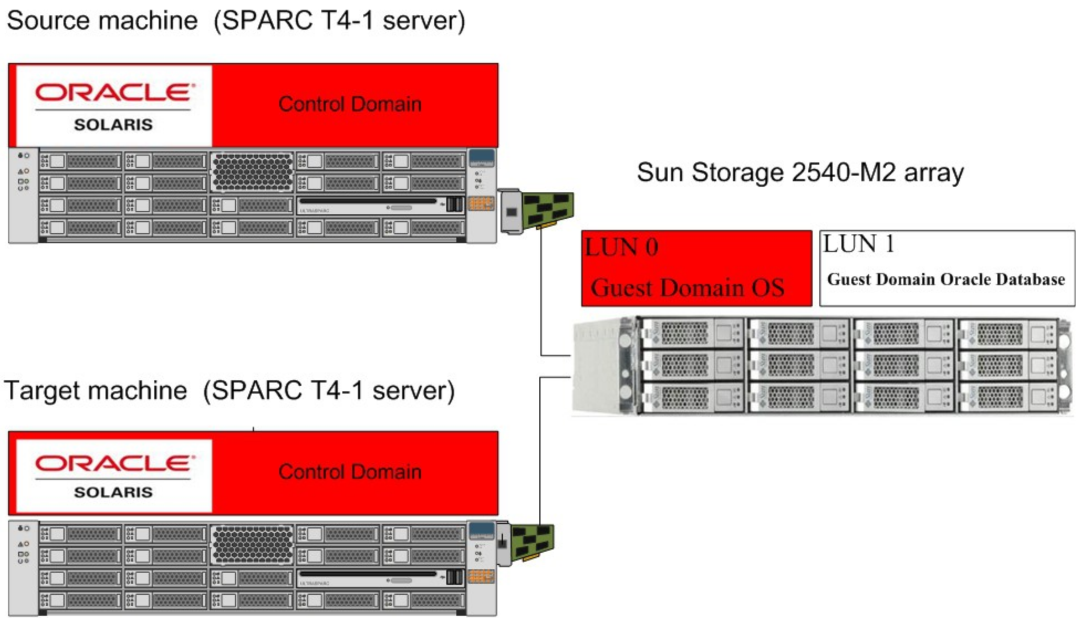
Source machine (SPARC T4-1 server)
ORACLE
SOLARIS
Control Domain
Sun Storage 2540-M2 array
LUN 0
Guest Domain OS
LUN 1
Guest Domain Oracle Database
Target machine (SPARC T4-1 server)
ORACLE
SOLARIS
Control Domain

Figure 2--Storage Configuration

Configuring the Control Domain

After installing the Oracle VM Server for SPARC software, configure the system to become the control domain as described in the Administration Guide in the section Setting Up Services and the Control Domain. These steps are outlined here.

Perform the following actions on each physical server:

1. Create a virtual console concentrator (`vcc`) service for use by the virtual network terminal server daemon (`vntsd`) as a concentrator for all logical domain consoles.

```
primary# ldm add-vcc port-range=5000-5100 primary-vcc0 primary
```

2. Create a virtual disk server (`vds`).

```
primary# ldm add-vds primary-vds0 primary
```

3. Create a virtual switch service (`vsw`) to enable networking for virtual network (`vnet`) devices in logical domains. In an Oracle Solaris 10 control domain, use the device driver name of the network interface you wish to use for the virtual switch.

```
primary# ldm add-vsw net-dev=igb0 primary-vsw0 primary
```

The following example shows the primary network interface on an Oracle Solaris 11 system. The system's primary network interface is `net0`, which is the vanity name of the data link on `igb0`. This network interface is used to create a virtual switch service (`vsw`).

```
primary# dladm show-phys

LINK   MEDIA      STATE   SPEED   DUPLEX   DEVICE
net0   Ethernet   up      1000    full     igb0
net1   Ethernet   up      1000    full     igb1
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

4. Verify that the services have been created by using the `ldm list-services` command.

```
primary# ldm list-services primary

VDS

   NAME         VOLUME OPTIONS DEVICE
   primary-vds0
VCC
   NAME         PORT-RANGE
   primary-vcc0 5000-5100
VSW
   NAME         MAC               NET-DEV  DEVICE   MODE
   primary-vsw0 02:04:4f:fb:9f:0d nxge0    switch@0 prog,promisc
VCC
   NAME          LDOM     PORT-RANGE
   primary-vcc0  primary  5000-5100
VSW
   NAME          LDOM    MAC               NET-DEV ID DEVICE
LINKPROP DEFAULT-VLAN-ID PVID VID ...
   primary-vsw0  primary 00:14:4f:f8:28:c2 igb0    0  switch@0  1
1               1500 on
VDS
   NAME          LDOM      VOLUME  OPTIONS  MPGROUP  DEVICE

   primary-vds0  primary
```

Oracle VM Server for SPARC uses SSL to encrypt migration traffic to protect sensitive data from exploitation without requiring additional hardware and dedicated networks.

The speed of the migration operation increases when both source and target machines use SPARC hardware cryptographic acceleration. This increases speed because CPU-intensive SSL operations can be offloaded to cryptographic algorithms implemented in silicon instead of software..

Note: SPARC T3 and earlier CPUs implement cryptographic acceleration with cryptographic units that must be assigned to the domain that uses them.

5. (T3 and earlier only) Determine whether you have cryptographic devices in the control domain.

```
primary#  list -o crypto primary
```

6. (T3 and earlier only) Assign cryptographic resources to the control domain. Assign one cryptographic unit for each core that will be assigned to the domain.

```
primary# ldm set-crypto 1 primary
```

7. Assign CPUs and memory to the control domain.

   You can reduce overall migration time by adding more CPUs to the primary domain on the source machine. It is best to assign CPUs in increments of entire SPARC cores, with a minimum of 1 CPU core. Alternatively, you can assign individual CPU threads, but you should have at least eight CPU threads assigned to the primary domain. For example, the following command assigns one core (eight CPUs) and 4 GB of memory to the control domain. These values are minimums for illustrative purposes.

```
primary# ldm set-core 1 primary    # or, ldm set-vcpu 8 primary

primary# ldm start-reconf primary
primary# ldm set-memory 4G primary
```

8. Reboot the system.

```
primary# init 6
```

   After the system reboots, Oracle VM Server for SPARC is enabled and the system is configured with one domain: the control domain, primary. From the control domain, you can create and configure additional domains.

9. Ensure that the Logical Domains Manager (`ldmd`) and the Virtual Network Terminal Server (`vntsd`) services are enabled.

```
primary# svcadm enable ldmd
primary# svcadm enable vntsd
```

10. Add a logical domain machine configuration to the service processor (SP) to make sure the configuration persists after a power-cycle,
    For example, the following command adds a configuration called `initial`.

```
primary# ldm add-config initial
```

11. Verify that the configuration is ready to be used.

```
primary# ldm list-config

factory-default

initial [current]
```

Configuring a Guest Domain

After the control domain has been configured, create the guest domain to be used as the Oracle Database node. Only one guest domain is created on the physical server that will be the source machine. The other physical server will eventually become the target machine.

The guest domain `ldg1` is initially created with the following resources:

- Eight Gbytes of memory
- 24 CPU threads (3 cores)
- One virtual network interface (`vnet0`) that is connected to the virtual switch `primary-vsw0`
- One virtual disk that appears as `c0d0` in the guest domain and is a LUN from the storage array. The domain `ldg1` uses LUN 0 of the storage array (`c2t6d0`).
- A second virtual disk that appears as `c0d1` in the guest domain and is a LUN from the storage array. The domain `ldg1` uses LUN 1 of the storage array (`c2t6d1`). This virtual disk is used to store Oracle Database files.

Perform the following steps to create each guest domain:

1. Create the `ldg1` guest domain from the control domain on the source machine..

   ```
   primary# ldm create ldg1

   primary# ldm set-core 3 ldg1
   primary# ldm set-memory 8G ldg1
   primary# ldm add-vnet vnet0 primary-vsw0 ldg1
   ```

   For the Oracle Solaris 11, run the following commands:

   ```
   primary# ldm add-vnet net0 primary-vsw0 ldg1
   primary# ldm add-vdsdev /dev/dsk/c2t6d0s2  vol1@primary-vds0
   primary# ldm add-vdisk vol1 vol1@primary-vds0 ldg1
   primary# ldm add-vdsdev /dev/dsk/c2t6d1s2 oradata@primary-vds0
   primary# ldm add-vdisk oradata oradata@primary-vds0 ldg1
   ```

2. After the domain is created, bind and start the guest domain from the control domain on the source machine by using the following commands.

   ```
   primary# ldm bind ldg1

   primary# ldm start ldg1
   ```

3. Determine the console port for the domain.

   ```
   primary# ldm ls ldg1

   NAME             STATE      FLAGS    CONS     VCPU  MEMORY    UTIL
   UPTIME
   ldg1             active     -n----   5000     24    8G        0.0%  1h
   1m
   ```

4. Access the console of the `ldg1` domain by using the `telnet` command.

   ```
   primary# telnet localhost 5000
   ```

After the guest domain is started, ensure that the appropriate Oracle Solaris OS and patches are installed in the guest domain. You can perform the installation over the network, from a DVD, or by using a DVD ISO image. Refer to the *Oracle VM Server for SPARC 3.1 Administration Guide*.

Oracle Solaris 11 only. You can use the Oracle Solaris 11 Automated Installer to install the packages as part of a network installation. See "How Do I Use the Automated Installer?" in *Installing Oracle Solaris 11 Systems*.

Note: LUNs often appear with different names on different servers. LUN names in control domains are not required to be identical. But the control domains on both systems must virtualize the same LUNs (LUN 0 and LUN 1) with the same volume names (vol1 and oradata).

Example 1–Configuring a Domain

The following example shows how a domain called ldg1 is configured with 24 CPUs, eight GB of memory, and two LUNs from the Sun Storage 2540-M2 storage array.

```
primary# ldm ls -l ldg1
NAME             STATE      FLAGS    CONS     VCPU  MEMORY    UTIL  UPTIME
ldg1             active     -n----   5000     24    8G        0.0%  1h 1m
SOFTSTATE
Solaris running
MAC
    00:14:4f:fb:96:89
HOSTID
```

```
    0x84fb9689
CONTROL
    failure-policy=ignore
DEPENDENCY
    master=
VCPU
    VID    PID    UTIL STRAND
    0      16     0.2%   100%
    1      17     0.0%   100%
    2      18     0.5%   100%
    3      19     0.0%   100%
    4      20     0.0%   100%
    5      21     0.0%   100%
    6      22     0.1%   100%
    7      23     0.0%   100%
    8      24     0.1%   100%
    9      25     0.0%   100%
    10     26     0.0%   100%
    11     27     0.0%   100%
    12     28     0.0%   100%
    13     29     0.0%   100%
    14     30     0.0%   100%
    15     31     0.0%   100%
    16     32     0.0%   100%
    17     33     0.0%   100%
    18     34     0.0%   100%
    19     35     0.0%   100%
    20     36     0.0%   100%
    21     37     0.0%   100%
    22     38     0.0%   100%
    23     39     0.1%   100%
MEMORY

    RA                PA                SIZE

    0x8000000         0x408000000       8G

VARIABLES

    boot-device=vdisk1
    keyboard-layout=US-English

NETWORK

    NAME      SERVICE                ID   DEVICE     MAC
MODE  PVID  VID   MTU   LINKPROP

    vnet1     primary-vsw0@primary   0    network@0  00:14:4f:f9:c0:62
1          1500

DISK

    NAME      VOLUME                 TOUT ID   DEVICE  SERVER
MPGROUP

    vdisk1    vol1@primary-vds0           0    disk@0  primary

    oradata   oradata@primary-vds0        1    disk@1  primary

VCONS

    NAME      SERVICE                     PORT

    ldg1      primary-vcc0@primary        5000
```

Example 2–Configuring a Control Domain

The following example shows the control domain's configuration. Use the `ldm list-services` command to verify that the services have been created.

```
primary# ldm ls-services primary
VCC
    NAME             LDOM             PORT-RANGE
    primary-vcc0     primary          5000-5100
VSW
    NAME             LDOM             MAC               NET-DEV   ID
DEVICE     LINKPROP   DEFAULT-VLAN-ID PVID VID        MTU   MODE
    primary-vsw0     primary          00:14:4f:f9:32:b0 nxgeg0   0
switch@0              1               1        1500

VDS
    NAME             LDOM       VOLUME      OPTIONS          MPGROUP
DEVICE
    primary-vds0     primary      vol1                  /dev/dsk/c2t6d0s2
                                  oradata               /dev/dsk/c2t6d1s2
```

Installing the Oracle Solaris OS in the Guest Domain

Ensure that the appropriate release of the Oracle Solaris OS is installed in the guest domain, and verify that the required patches are present.

After the Oracle Solaris OS is installed in the guest domain, the system can be configured and enabled to use the Oracle Database software.

Installing the Oracle Database

Installation of the Oracle Database software on an Oracle VM Server for SPARC system is similar to a standard Oracle Database installation. Start with the installation of the Oracle Database software and then apply the latest patch set. Use the second LUN to create a UFS or ZFS file system to store Oracle Database data files. Note that ZFS is the default root file system for the Oracle Solaris 11 OS.

For more information about using ZFS for an Oracle Database, see Configuring Oracle Solaris ZFS for an Oracle Database.

Preparing the Target Machine

1. Perform the steps described in the following sections to prepare the target machine:
   - "Installing the Oracle VM Server for SPARC Software"
   - "Configuring the Control Domain"
2. Ensure that all virtual I/O services that are used by the domain to be migrated are available on the target machine.
3. Ensure that the target machine has access to both LUN 0 and LUN 1 from the same storage.

   To add the first LUN (LUN 0) and second LUN (LUN 1) to the target machine, execute the following commands from the control domain of the target machine:

   ```
   primary# ldm add-vdsdev /dev/dsk/c2t6d0s2 vol1@primary-vds0
   ```

   ```
   primary# ldm add-vdsdev /dev/dsk/c2t6d1s2 oradata@primary-vds0
   ```

   Note: LUNs often appear with different names on different servers. LUN names in control domains are not required to be identical. But the control domains on both machines must virtualize the same LUNs (LUN 0 and LUN 1) with the same volume names (`vol1` and `oradata`).

4. Ensure that each virtual network device in the domain to be migrated has a corresponding virtual network switch on the target machine:

```
primary# ldm add-vsw net-dev=igb0 primary-vsw0 primary
```

For the Oracle Solaris 11 OS, run the following command:

```
primary# ldm add-vsw net-dev=net0 primary-vsw0 primary
```

Cross-CPU Migration

You can migrate a domain even if the source machine and the target machine have different processor types. For example migrate guest domain from a system with UltraSPARC T2 Plus or SPARC T3 CPU to a system that has a SPARC T4 CPU, or between SPARC T4, SPARC T5, M5-32 and M6-32 servers.

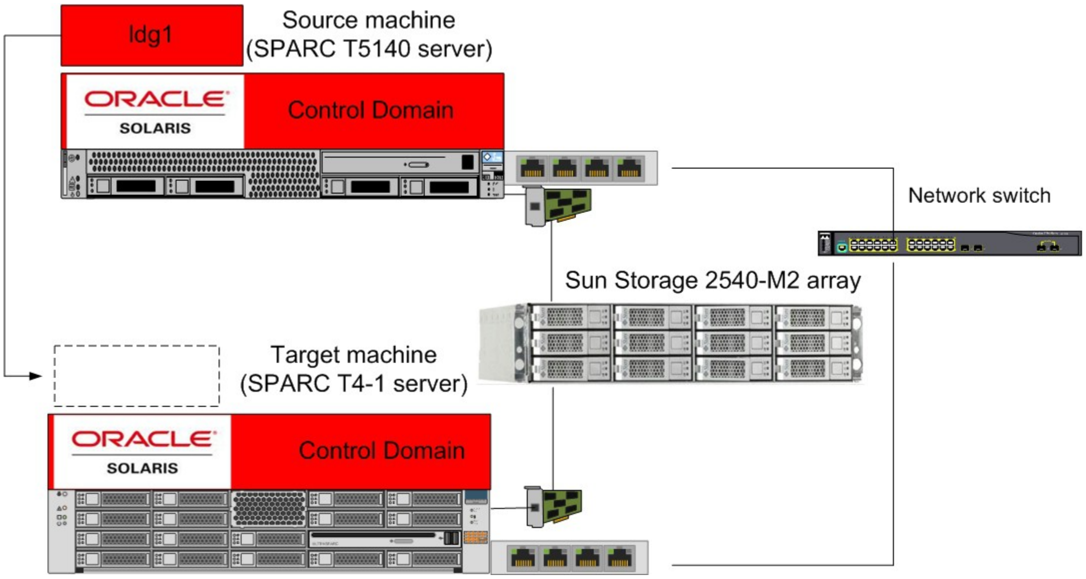

Figure 3--Storage Configuration

Note: To use cross-CPU migration feature you must set the `cpu-arch` property value on the domain while it is bound or inactive, specifying one of the following values:

- `generic` uses common CPU hardware features to enable a guest domain to perform a CPU-type-independent migration, and can be used for live migrating domains between UltraSPARC T2, UltraSPARC T2+, SPARC T3 and SPARC T4 servers or later.

- `migration-class1` uses CPU-specific hardware features common to recent SPARC servers. The `migration-class1` value balances performance and flexibility when migrating between platforms starting with the SPARC T4 platform. It has more CPU features, such as hardware
  cryptography instructions, and can be migrated to other CPU types as long as they are at least a SPARC T4.
- `sparc64-class1` is a cross-CPU migration family for SPARC64 platforms. This value is compatible only with Fujitsu M10 systems.
- `native` uses CPU-specific hardware features to enable a guest domain to migrate only between platforms that have the same CPU type. `native` is the default value, and always provides optimal CPU performance.

Using the `generic` value might result in reduced performance compared to the other values. This occurs because the guest domain does not use some features that are only present in newer CPU types. The `generic` setting enables the flexibility of migrating the domain between systems that use newer and older CPU types.

Use the `ldm set-domain` command to change the `cpu-arch` property value when the domain is in bound or inactive state:

```
primary# ldm stop ldg1
primary# ldm set-domain cpu-arch=migration-class1 ldg1
primary# ldm start ldg1
```

When the prerequisites are satisfied, you can continue with the live migration process.

Live Migration Process

The Logical Domains Manager on the source machine accepts the request to migrate a domain and establishes a secure network connection with the Logical Domains Manager that runs on the target machine. After this connection has been established, the migration occurs. The migration itself can be broken down into five different phases:

- **Phase 1**: After connecting with the Logical Domains Manager running on the target machine, information about the source machine and the domain to be migrated are transferred to the target machine. This information is used to perform a series of checks to determine whether a migration is possible.

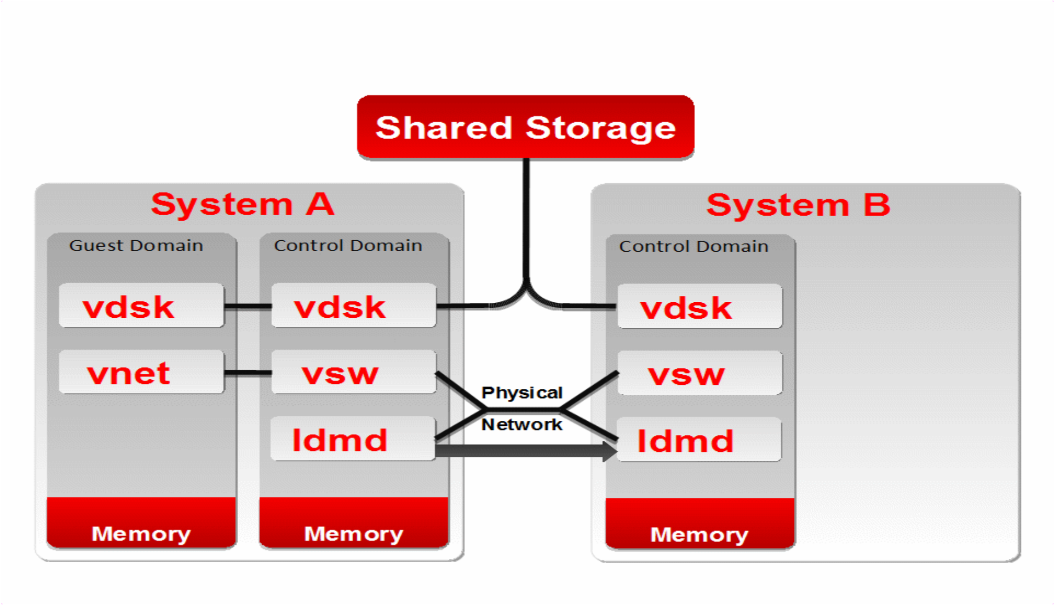

Figure 4--Live Migration Phase 1

- **Phase 2**: When all checks in Phase 1 have passed, the source and target machines prepare for the migration. An inactive domain is created on the target machine to receive the source domain.

  The source machine begins to track which memory blocks of the active guest domain change during the migration operation. If the domain to be migrated is inactive or bound, the migration operation proceeds to Phase 5.

- **Phase 3**: All the runtime state information for the domain is transferred to the target. This information includes the CPU states and the domain memory. The source machine sends all memory blocks to the target machine. When the guest domain memory is transferred to the target machine, the source machine begins to send blocks that are marked as being changed. This transfer operation occurs over several iterations as the guest domain continues to run.

- **Phase 4**: The domain to be migrated is suspended. At this time, the rest of the modified source domain memory blocks and state information is transferred to the target machine.

- **Phase 5:** After all state information is transferred, the handoff occurs when the target domain resumes execution and the source domain is destroyed. From this point on, the target domain is the sole version of the domain running.

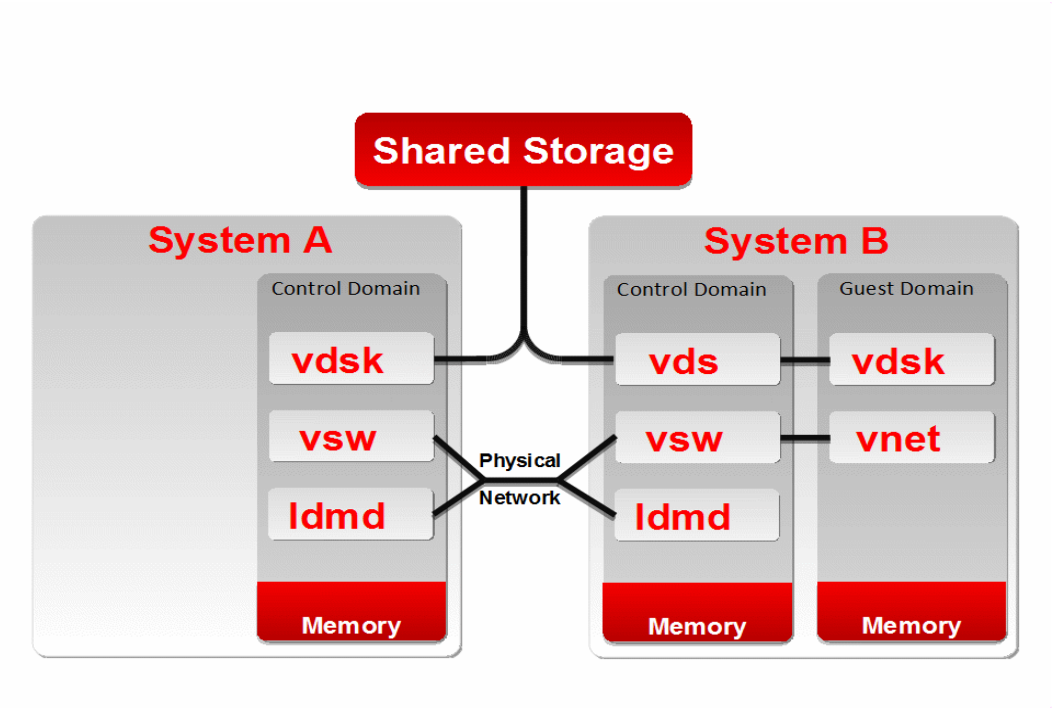

Figure 5--Live Migration Phase 5

Live Migration Examples

The following examples use the `ldm migrate-domain` command to migrate a guest domain from one system to another system.

You can optionally specify an alternate user for authentication on the target machine. See “Assign an Authorization to a User” in the *Oracle VM Server for SPARC Administration Guide* of the documentation.

The `ldm migrate-domain -n` command enables you to perform a dry-run migration, which performs migration checks but does not migrate the specified domain. Any requirement that is not satisfied is reported as an error, which enables you to correct any configuration errors before attempting an actual migration.

The following examples show various ways in which to perform domain migrations:

- Perform a dry-run migration of the `ldg1` domain to a target machine called `t4-1`.

```
primary# ldm migrate-domain -n ldg1 t4-1
Target Password: password
```

- Perform a migration of the `ldg1` domain to a target machine called `t4-1`.

  ```
  primary# ldm migrate-domain ldg1 t4-1
  Target Password: password
  ```

- Perform a non-interactive migration of the `ldg1` guest domain to a target machine called `t4-1`. The `-p` option specifies the name of a file, `pfile`, that contains the plain-text superuser password for the target machine, which enables you to access the `t4-1` target machine.

  ```
  primary# ldm migrate-domain -p pfile ldg1 t4-1
  ```

  Note: Ensure that you protect the file that contains the superuser password so that only the `root` owner, or a privileged user, can read it.

  ```
  primary# chmod 400 pfile
  ```

- Perform a migration of the `ldg1` domain to a target machine called `t4-1` and specify an alternate user, `tracy`, to be used for authentication on the `t4-1` machine.

  ```
  primary# ldm migrate-domain ldg1 tracy@t4-1
  ```

  For information about assigning an authorization to a user, see “Managing User Profiles” in the *Oracle VM Server for SPARC 3.1 Administration Guide*.

- Monitor the progress of the migration operation:

  ```
  primary# ldm list -o status ldg1
  ```

  Note: The migration time might vary depending on the amount of memory assigned and the network utilization.

Workload Description and Performance Results

The SwingBench kit contains a load generator and benchmarks that are designed to stress test an Oracle database. The code that ships with SwingBench includes the OrderEntry, SalesHistory, CallingCircle, and StressTest benchmarks.

This paper used the SwingBench OrderEntry workload to generate workload on the guest domain being migrated. OrderEntry is based on the `oe` schema that ships with Oracle Database 11*g*. The workload introduces heavy contention on a small number of tables, and it is designed to stress the system interconnects and memory.

The following describes the workload that is being generated on the guest domain:

- Size of the database on disk: 30 Gbytes
- System global area (`sga_target`): 18 Gbyte

  The SGA is a group of shared memory structures that contain data and control information for one Oracle Database instance.

- Number of CPUs assigned to the guest domain: 24 CPU threads (3 cores)
- Amount of memory assigned to the guest domain: 24 Gbytes
- Number of workload users (`users`) for workload: 50 users
- Time to think (time between actions taken by the user): 100 milliseconds
- Duration of the workload: 30 minutes

Table 1 shows the results of live migration operations that ran between two SPARC T4-1 machines with this particular workload:

Table 1. SPARC T4-1 Live Migration Results

NUMBER OF CPUS ON THE CONTROL DOMAIN	OVERALL MIGRATION TIME	SUSPENSION TIME	GUEST DOMAIN CPU USAGE
8 CPUs	8 min 12 sec	26 sec	70.00%
16 CPUs	4 min 2 sec	13 sec	80.00%
24 CPUs	2 min 3 sec	7 sec	85.00%

Before You Begin a Domain Migration

The following describes the live migration operation behavior that is typically seen when migrating a domain from one machine to another machine. Note that you might observe slightly different results because the migration time depends on the number of CPUs and the amount of memory assigned to the control domain, as well as the workload type.

- Oracle VM Server for SPARC live migration is designed to work over any type of network, including shared networks and insecure networks. The migrations that were run on the main NIC for this paper did not show any performance degradation when compared to those migrations that were run on the dedicated NIC. For best performance, use unsaturated, high-speed network interfaces, such as 10 Gbit Ethernet.

- Use the `mpstat` command to observe the effect of system memory tracking on the guest domain's CPU. See the `USR` column to view the usage for each CPU.
- The guest domain CPU load rises during live migration, as the guest domain CPU threads perform memory change-tracking in the hypervisor. This tracking is reported as CPU time for the application process.
- It is best to configure sufficient CPU resources on both the guest domain that runs the database and the control domain that orchestrates the live migration. For a database workload that is similar to the one used in this paper, assign 24 CPUs (3 cores) from a SPARC T4 system for the guest domain (database) and at least 1 core (8 CPU threads) for the control domain. Add more CPU cores to the control domain to reduce live migration times. It is best to perform your own benchmarks with your unique workload to determine the appropriate CPU resource allocations. See Table 1 and "Using CPU Dynamic Reconfiguration" for information about how to add more CPUs to the control domain.
- Migration time also depends on the following factors: the amount of memory that is assigned to the guest domain, the size of the database SGA, the number of database block changes, and the amount of read-write I/O to the disk.
- You can use the Automatic Workload Repository (AWR) tool to monitor the number of database block changes. Look for the `db_block_changes` information in the Instance Activity Stats section of the generated AWR report. See Section 5.2, "Overview of the Automatic Workload Repository" in *Oracle Database Performance Tuning Guide*.
- The guest domain should be configured to use the Network Time Synchronization Protocol (NTP) time synchronization service so the time on the guest domain is correctly reset after the migration operation completes. For an example of NTP setup on an Oracle Solaris 10 system or Oracle Solaris 11 system, see the "Configuring NTP" section.
- Do *not* use live migration as an emergency tool. However, if you must do so, ensure that your environment is ready by setting the storage paths in advance. Also, ensure that you perform live migration operations in an environment that is as close to the production environment as possible. By ensuring this readiness in advance, you will know how much time it takes to complete a migration.

Related Information

This section contains information that is relevant to the Oracle VM for SPARC live migration feature, including configuring NTP and using CPU dynamic reconfiguration.

Configuring NTP

Ensure that the system clock on the domain to be migrated is synchronized by using NTP.

In this example, the control domain on the source machine is used as a time source, and it configured as an NTP server. Note that it is best to select an NTP server that can be a dedicated time synchronization source, so that other services are not negatively affected if the machine is brought down for planned maintenance.

Example 3–Configuring an NTP Server

This example shows how to configure an NTP server.

```
# grep -v ^# /etc/inet/ntp.conf
server 127.127.1.0 prefer
broadcast 224.0.1.1 ttl 4
enable auth monitor
driftfile /var/ntp/ntp.drift
statsdir /var/ntp/ntpstats/
filegen peerstats file peerstats type day enable
filegen loopstats file loopstats type day enable
filegen clockstats file clockstats type day enable
keys /etc/inet/ntp.keys
trustedkey 0
requestkey 0
controlkey 0
# touch /var/ntp/ntp.drift
# svcadm enable ntp
```

The following example shows how to configure an NTP client:

```
# grep -v ^# /etc/inet/ntp.conf
server source prefer
slewalways yes
disable pll
# svcadm enable ntp
```

Using CPU Dynamic Reconfiguration

Oracle VM Server for SPARC supports dynamic reconfiguration of CPUs. CPUs can be dynamically added to or removed from any active domain, including the control domain, by using the `ldm add-core, ldm rm-core, ldm set-core, ldm add-vcpu,` `ldm rm-vcpu, ldm-set-vcpu` commands.

These commands can be issued from the control domain to increase or decrease the number of allocated CPUs. Increasing the number of CPUs on the control domain can reduce migration time, especially on the source system. For systems that have discrete cryptographic units, such as the UltraSPARC T2, UltraSPARC T2 Plus, and SPARC T3 platforms, you should add cryptographic units to the control domain as you dynamically add CPU resources.

The following example shows how to add one CPU core (eight CPU threads) to the control domain:

```
primary# ldm add-core 1 primary
```

After migration completes, you can remove the virtual CPUs from the control domain and allocate them to a different domain. This example removes a CPU core from the control domain and allocates it to the `ldg1` guest domain:

```
primary# ldm rm-core 1 primary
primary# ldm add-core 1 ldg1
```

Note: When reducing the number of CPUs, ensure that a sufficient number of CPUs are still available to efficiently handle the domain's workload. For instance, an Oracle Database single-instance guest domain should *always* have at least one CPU core.

Conclusion

An IT organization can reach tactical and strategic goals by choosing the right hardware and software to best manage application availability and to optimize data center resources.

This paper demonstrated the benefits of using the Oracle VM Server for SPARC live migration feature to manage the production lifecycle of an Oracle Database 11*g* Release 2 single-instance database. This paper showed the complete configuration process, including the creation and configuration of the domains, storage configuration, and software that were used to demonstrate these benefits. In addition to showing the benefits of the live migration feature, this paper described how to achieve better application availability during planned downtime and how to reduce hardware and software costs by means of dynamic hardware resource management in a production environment.

For more information about Oracle's virtualization solutions, visit oracle.com/virtualization.

About the Authors

Orgad Kimchi joined Sun/Oracle in September 2007. He currently works in the Independent Software Vendors (ISV) Engineering organization to assist software vendors in adopting Oracle technology that improves the performance of Oracle hardware and software. See Orgad's blog.

Roman Ivanov joined Sun/Oracle in January 2006. He currently works in the ISV Engineering organization to assist software vendors in adopting Oracle technology that improves the performance of Oracle hardware. See Roman's blog.

Jeff Savit joined Sun/Oracle in February 2000. He currently works in the Oracle VM Product Management organization. See Jeff's blog.

Resources

The following resources have been referenced in this paper:

- Oracle VM Server for SPARC web page
  http://www.oracle.com/us/technologies/virtualization/oraclevm/oracle-vm-server-for-sparc-068923.html
- Oracle VM Server for SPARC documentation
  http://www.oracle.com/technetwork/documentation/vm-sparc-194287.html
- Oracle Database documentation

  http://www.oracle.com/pls/db112/homepage
- My Oracle Support (registration required)

  https://support.oracle.com
- Oracle Solaris 10 documentation

  http://www.oracle.com/technetwork/documentation/solaris-10-192992.html
- Oracle Solaris 11 documentation

  http://www.oracle.com/technetwork/documentation/solaris-11-192991.html
- Oracle VM downloads

  http://www.oracle.com/technetwork/server-storage/vm/downloads/index.html

- "Configuring Oracle Solaris ZFS for an Oracle Database" http://www.oracle.com/technetwork/server-storage/solaris/config-solaris-zfs-wp-167894.pdf
- SwingBench web site

  http://dominicgiles.com/swingbench.html
- *Oracle Database Performance Tuning Guide*:

  http://www.oracle.com/pls/db112/to_toc?pathname=server.112/e16638/toc.htm
- Oracle's virtualization solutions web page

  http://oracle.com/virtualization
- Orgad's blog

  https://blogs.oracle.com/vreality
- Roman's blog

  https://blogs.oracle.com/pomah

Oracle is committed to developing practices and products that help protect the environment

Increasing Application Availability by Using the Oracle VM Server for SPARC: An Oracle Database Example
March 2014, Version 1.3
Authors: Orgad Kimchi, Roman Ivanov, Jeff Savit

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Hardware and Software, Engineered to Work Together