Cluster Platform 220/1000 Architecture -- A Product from the SunTone™ Platforms Portfolio

---

By Enrique Vargas - Enterprise Engineering

Sun BluePrints™ OnLine - August 2001

http://www.sun.com/blueprints

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 816-1383-10
Revision 01, 08/17/01
Edition: August 2001

Please
Recycle

Cluster Platform 220/1000 Architecture--A Product from the SunTone™ Platforms Portfolio

Sun customers interested in the quick deployment of highly available, distributed, or non-distributed applications using the Sun™ Cluster 3.0 technology have a new choice—Cluster Platforms from the SunTone™ Platforms portfolio. The SunTone Cluster Platform line provides a qualified (Quality Assurance) integration of Sun hardware and software products (integration based on engineering and field best practices), which enables the quick and safe deployment of an Sun Cluster 3.0 cluster environment.

Note – The SunTone^{SM} program has been designed to provide customers with a thorough evaluation of their business operations and information technology (IT) infrastructure, which helps improve their quality of service and reliability. Currently, the SunTone Certification and Branding program has more than 1,400 applicants for certification —including service providers, independent software vendors, and integrators— and provides more than 90 SunTone Certified Solutions. The SunTone Platforms portfolio is a new addition to the program, providing application-ready hardware and software stacks to help improve IT infrastructure deployment time and reliability. For more information, see `http://www.sun.com/integratedplatforms`.

This article discusses the function of individual hardware and software components, as well as the connectivity details involved in developing the Cluster Platform 220/1000 product. Because the Cluster Platform 220/1000 is built using Sun hardware and software components, the reader could make use of the best practices and principles set forth in this article to either duplicate the specific product configuration, or create similar configurations using alternate components. The end result will provide a more integrated Sun Cluster 3.0 software solution which minimizes problems with component interactions, while improving overall system availability.

This article is intended for systems administrators that have previous Sun Cluster 3.0 software experience, as well as system architects and technologists interested in the adoption of the Sun Cluster 3.0 product. Since the JumpStart™ technology is a key element of the Cluster Platform 220/1000 architecture, this article can be used by cluster customers with previous JumpStart server experience interested in leveraging such technology to help achieve higher levels of availability.

To better understand the application of the Cluster Platform 220/1000, there will be a follow up Sun BluePrints™ Online article describing the implementation of a highly-available NFS Server.

Product Philosophy

The Cluster Platform 220/1000 integrates the hardware and software component stack using best practices learned from engineering and field experience. The hardware stack components are properly labeled and placed in a Sun StorEdge™ Expansion Cabinet in compliance with existing power, cooling, and Electro Magnetic Interference (EMI) requirements. System components are cabled to provide redundant cluster interconnect between nodes, access to mirrored shared storage, and access to a highly available production network.

For the software stack and patch integration, the Cluster Platform 220/1000 includes a Netra™ T1 management server, which is used as a JumpStart server. The JumpStart server automates the installation of *all* software modules and patches required by two Sun Enterprise™ 220R servers to form an Sun Cluster 3.0 environment. When the management server is first turned on, it customizes its JumpStart environment by requesting information specific to the customer locality (i.e. cluster name, cluster node names, ethernet and IP addresses, timezone, name services, etc.). After the management server customization is performed, the two Sun Enterprise 220R servers will automatically boot as Sun Cluster 3.0 nodes, after the JumpStart software installation is complete.

In addition to providing the JumpStart function, the management server is intended to manage the cluster environment. Currently the management server includes the SUNWccon package (extracted from the Sun Cluster 3.0 client software) to manage the cluster node consoles through the Sun Cluster Control Panel GUI (ccp) function. If required by customers, the management server has been allocated enough CPU power and memory resources to implement a Sun™ Management Center (Sun MC) software and AnswerBook™ servers.

The Cluster Platform 220/1000 provides customers with the integrated hardware and software stack required to implement a *basic* Sun Cluster 3.0 environment. A *basic* Sun Cluster 3.0 environment means that before being deployed into production, it will still require a service representative (or a service-qualified customer) to perform the following tasks:

- Configure the Sun Cluster 3.0 quorum disk. Quorum devices are resources used by the cluster infrastructure to establish a majority vote when determining cluster membership.
- Install a specific Sun Cluster 3.0 data service based on application needs. Any supported Sun Cluster 3.0 data services can be installed. A data service includes an application and cluster routines (start, stop, monitoring, etc.) required to integrate the application into the cluster environment.
- Configure disk data. Veritas Volume Manager and Solstice DiskSuite™ software can be used.

Hardware Components

As depicted in FIGURE 1, the Cluster Platform 220/1000 system includes a two-node cluster having access to mirrored SCSI storage, a terminal concentrator, and a management server. All components are discussed in detail in the following paragraphs.

- The management server is a Netra T1 AC200 server, configured with one 500 MHz UltraSPARC™ II CPU, 256 Mbytes of memory, one DVD drive, and two 18.2 Gbytes internal SCSI disks. The management server provides access to the cluster console, and functions as a JumpStart server (automated software-install server) for the cluster nodes. Using JumpStart technology to load the required software and patches on the cluster nodes is considered a best practice, since it automates the software install and can reduce operator errors during a catastrophic failure recovery. In addition, the JumpStart technology provides a manageable environment where the latest versions of required software packages and associated patches are collected.
- Each of the cluster nodes is a Sun Enterprise 220R server configured with two 450-MHz UltraSPARC II CPUs, two Gbytes of memory, and two 18.2 Gbytes internal SCSI disks. Each cluster node is loaded with two Ultra-SCSI, differential, PCI cards, and two Sun Quad FastEthernet™ PCI cards. The first cluster node is factory-configured to have the scsi-initiator-id OpenBoot™ Prom (OBP) environment variable set to a value of 6 to remove bus conflicts when sharing external SCSI disk between the two cluster nodes.

- Shared storage is implemented using two Sun StorEdge D1000 Disk Arrays, each configured with a single (non-segmented) Ultra-SCSI bus backplane which hosts 12- 18.2 Gbytes, 10,000 RPM, Ultra-SCSI drives. One disk array provides a total storage capacity of 200 Gbytes for production data; the other is used to mirror the 200 Gbytes of production data.

Note – There are currently two spare Ultra-SCSI ports on each cluster node. Four additional Ultra-SCSI cables are supplied with the product to either implement a segmented Ultra-SCSI bus on each Sun StorEdge D1000 Disk Array (to increase the maximum bandwidth by splitting the bus) or to attach an additional Sun StorEdge D1000 Disk Array and its mirror.

- The 8-port terminal concentrator provides serial access to the management server and cluster node consoles. The terminal concentrator is only accessible over 10BASE-T ethernet.
- The cluster interconnect is implemented with ethernet patch cables to increase availability by reducing network components (hubs or switches). The redundant Sun Quad FastEthernet 100BASE-T ports (qfe1 and qfe5) are placed on two separate PCI controllers to avoid a controller single point of failure (SPOF).
- Public network access for each cluster node is implemented using the onboard hme0 (100BASE-T port) with qfe0 (100BASE-T port on a separate controller) as a failover interface (dashed line in FIGURE 1). Five qfe ports are still available to supply additional production network requirements.

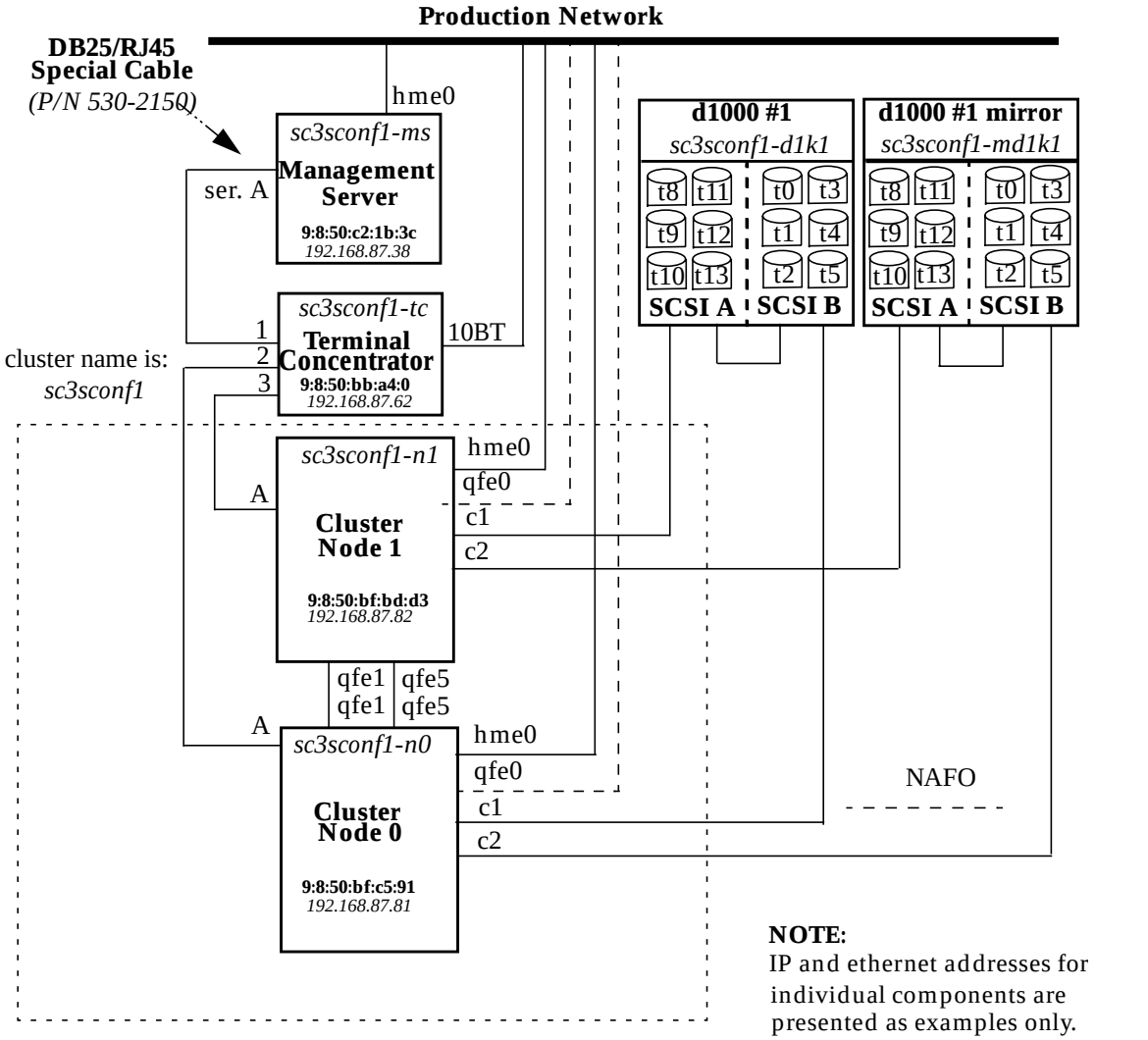
FIGURE 1 Cluster Platform 220/1000 Logical Connectivity

The hardware stack is placed in a Sun StorEdge Expansion Cabinet in the order depicted in FIGURE 2 to comply with existing power, cooling, and Electro Magnetic Interference (EMI) requirements. Power is distributed between two power sequencers to avoid a SPOF. The terminal concentrator and management server do not require redundancy, since these components are used for managing the cluster (any failures to these particular components does not impact the cluster health).

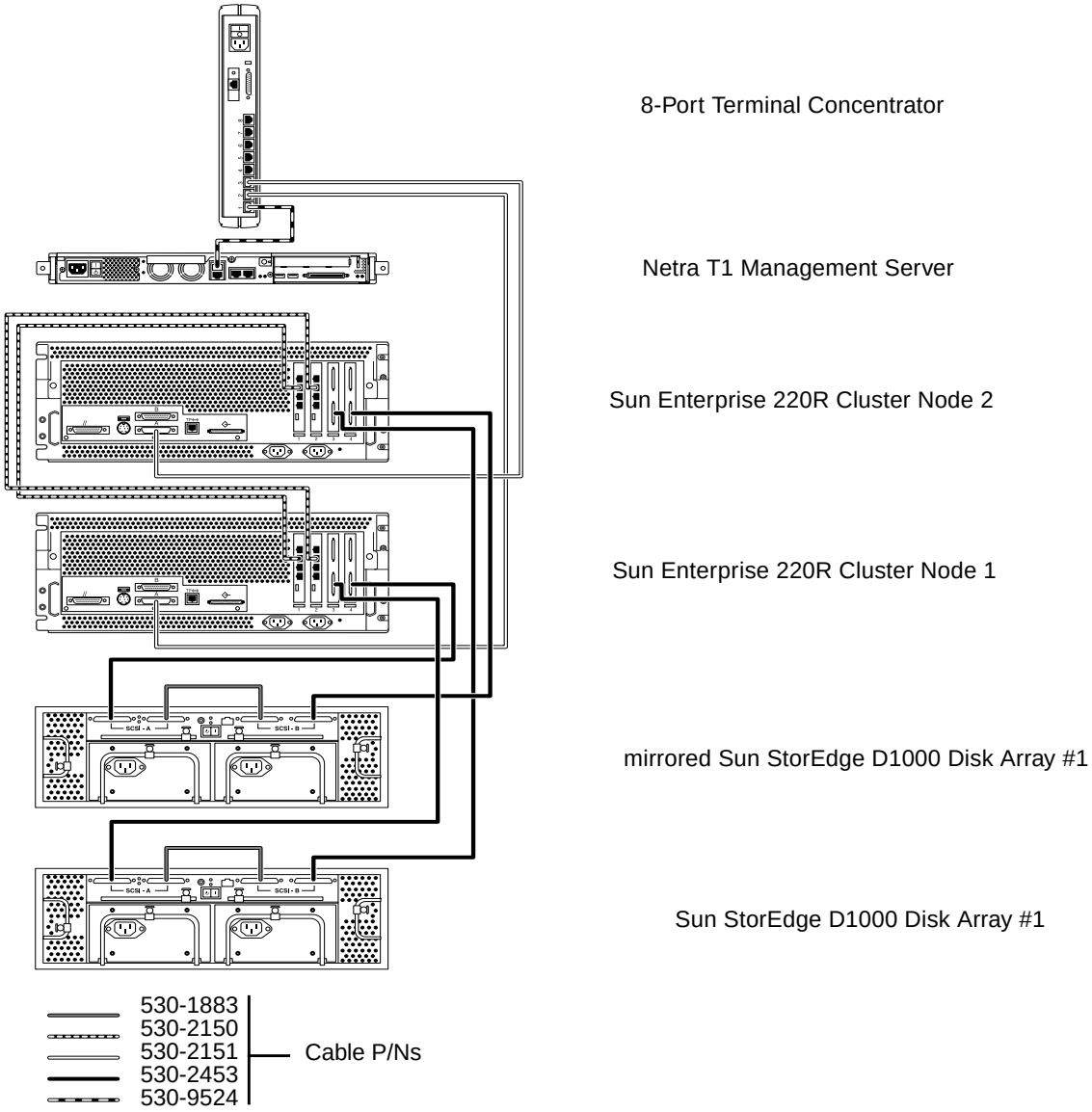
FIGURE 2 Cluster Platform 220/1000 Rack Placement

Software Components

All the software components required for the integration of the Cluster Platform 220/1000 are included in the Netra T1 management server's JumpStart environment. The management server executes the Solaris™ Operating Environment (Solaris OE) 8 01/01 Operating Environment and the following software packages:



- Solstice DiskSuite version 4.2.1 software, to mirror the root and swap partitions and help increase system availability.
- The SUNWccon package, to provide the Sun Cluster Control Panel GUI (ccp) function and manage the cluster node consoles.



Note – All software packages installed on the management server include recommended patches. The Netra T1 server requires Solaris OE 8 01/01 as a minimum to support its hardware architecture, while the Sun Cluster 3.0 software requires Solaris OE 10/00.

All the JumpStart software components are placed under the `/SOFTWARE` tree, which contains the following directories:

- `SC3.0-Build92-Product`: Includes all the files that make up the Sun Cluster 3.0 software (FCS version).
- `SC3.0-Build92DataServices`: Includes all data services released with the Sun Cluster 3.0 software (FCS version). Because the Cluster Platform 220/1000 does not include configured Sun Cluster 3.0 Data Services, this directory is provided to support the data service installation.
- `Solaris8_10-00`: Includes the collection of files which support the JumpStart environment and make up the Solaris OE 8 10/00 software.

Note – A set of three CDs included with the Cluster Platform 220/1000 product (Part Number 724-9002-02) can be used to recover the entire contents of the management server software in the event of a catastrophic failure: *CD0* contains a bootable copy of the Solaris 8 OE 01/01 configured to execute recovery software; *CD1* contains the Solaris OE (root); and *CD2* contains the JumpStart server support files (the `/SOFTWARE` directory). Full recovery procedures are included with the product *Cluster Platform 220/1000 User's Guide, A product from the SunTone™ Platforms Portfolio* (Part Number 806-7408-11).

JumpStart Software Components

The JumpStart software components include the Solaris OE 8 10/00, Solstice DiskSuite version 4.2.1 software, Sun Cluster 3.0 software (and data services), and all recommended patches.

Note – To learn more about the architectural details of a JumpStart environment, review the Sun BluePrints Online article *Building a JumpStart Infrastructure* by Alex Noordergraaf. This article can be found at:
`http://www.sun.com/blueprints/0401/BuildInf.pdf`.

Even though the reference of the Sun BluePrints OnLine article recommends a certain JumpStart directory structure as a best practice, such best practice was published after the Cluster Platform 220/1000 was developed. JumpStart directory structure best practices will be adopted by future SunTone Cluster Platform products.

The JumpStart components are placed under the `/SOFTWARE/Solaris8_10-00` tree, which contains the following files:

- `sysidcfg`: A script used by the JumpStart server to identify the machine locality (timezone, name services, netmask, etc.) of all nodes to be served.
- `rules`: A script used by the JumpStart server to identify the operating system install profile, as well as the post-install script.

- `check`: A script which generates the `rules.ok` file after verifying the syntax within the `rules` file.
- `rules.ok`: A file generated by the check script. The JumpStart process aborts if this file is not present.
- `noask_pkgadd`: An administration script used by `pkgadd(1M)` to avoid user interaction when installing software packages.
- `response/sunwmd`: An empty response file expected by all Solstice DiskSuite software packages when performing unassisted `pkgadd(1M)`. If this file does not exist, the `pkgadd(1M)` process will fail.

The `/SOFTWARE/Solaris8_10-00` tree contains the following directories:

- `Profiles`: Includes the `sc3sconf1-n0.profile` and `sc3sconf1-n1.profile` scripts used by the JumpStart server to identify internal disk partitioning and the content of the Solaris OE 8 10/00 software packages on each node.
- `Solaris_8`: Includes the entire CD-ROM image of the Solaris OE 8 10/00 software.
- `Drivers`: Includes the `sc3sconf1-n0.driver` and `sc3sconf1-n1.driver` scripts, which identify the JumpStart serve finish scripts to be executed after loading the Solaris OE 8 10/00 packages on each cluster node. The `driver.init` file initializes the execution environment and each cluster node (i.e. disable routing, establish a default router, build the `/etc/hosts` file, build the `.profile` file, etc.). The `driver.run` file executes the scripts included in the `sc3sconf1-n0.driver` and `sc3sconf1-n1.driver` scripts.
- `Finish`: Includes all the JumpStart server finish scripts invoked through the `sc3sconf1-n0.driver` and `sc3sconf1-n1.driver` files.
- `Packages`: Includes all packages to be added after loading the Solaris OE 8 10/00 packages (currently, the Solstice DiskSuite software packages only).
- `Patches/8_Recommended`: Includes all patches required by the Solaris OE, Sun Cluster 3.0Sun Cluster 3.0 software, and Solstice DiskSuite software. These patches are installed in the order established by the `Patches/8_Recommended/patch_order` file.

Note – This product currently includes all patches required by the Sun Cluster 3.0 first customer ship (FCS) version. A local Sun Service representative must be contacted to acquire the full list of required patches.

The `Files` directory includes the following files:

- `JumpStart.conf`: Contains all variables collected during system configuration of the management server. When the management server is first booted, it will prompt for the cluster name, terminal concentrator and cluster nodes names, ethernet and ip addresses, etc.

- `JumpStartConfig.init`: This script is invoked by the `/etc/rc2.d/S91JumpStart` script included with the management server. The `S91JumpStartConfig.init` script executes during the first reboot and builds the `JumpStart.conf` and the `/etc/rc2.d/S91sconf-attach` files. The `S91JumpStartConfig.init` script removes the `/etc/rc2.d/S91JumpStart` script to avoid execution during the next reboot.
- `vfstab.ms:` This file replaces the `/etc/vfstab` file in the management server after the first reboot to include the newly mirrored root and swap Solstice DiskSuite metadevices. During the second reboot of the management server, the `/etc/rc2.d/S91sconf-attach` script finishes up the Solstice DiskSuite configuration. The `/etc/rc2.d/S91sconf-attach` creates an Solstice DiskSuite database, mirrors the root and swap partitions, and initializes the `/etc/lvm/md.tab` file.
- `JumpStartHosts.conf`: This file is built using data from the `JumpStart.conf` file and contains the names and IP addresses of all cluster elements (management server, terminal concentrator, first cluster node, second cluster node). The contents of this file are appended to the `default/etc/hosts` file allocated to each cluster node.
- `S94n0-sds-mirror and S94n1-sds-mirror`: One of these files is moved into the `/etc/rc2.d` directory of each cluster node before the first reboot. During the first reboot, this script finishes up the Solstice DiskSuite configuration and creates the Solstice DiskSuite database. This script also modifies the `/etc/vfstab` to include newly created Solstice DiskSuite metadevices and it attaches mirrors to the root, swap, and `/globaldevices` metadevices. This script also installs the Sun Cluster 3.0 software on each node and replaces the `default /etc/inet/ntp.conf` file included with the Sun Cluster 3.0 software. This script removes itself to avoid execution during the next reboot.
- `md.conf`: This file replaces the default `/kernel/drv/md.conf` file included with the Solstice DiskSuite software.
- `ntp.conf`: This file replaces the default `/etc/inet/ntp.conf` file included with the Sun Cluster 3.0 software.

Automated Software Installation

The automated software installation on each cluster node takes approximately one hour, maybe longer in a saturated network, and is executed in parallel by typing the following command at the OBP prompt on each cluster node:

```
ok> boot net - install
```

The software install process makes use of the JumpStart environment at the beginning, and then makes use of `/etc/rc2.d` scripts to continue with the installation process after a cluster node reboots. This section makes use of the files introduced in the previous section, and is intended to further clarify the JumpStart server process.

JumpStart Server Install Sequence

During the JumpStart server process on the management server (*ms*), the following steps represent the time-sequenced events involved during the automated software install on the first cluster node (*node*).

1. The *node* sends a RARP packet on the network to be able to map its ethernet address to a name and IP address:
   - The *ms* maps the *node's* ethernet address to a machine name using the `/etc/ethers` file. The *node's* IP address is resolved using the `/etc/hosts` file (or a name service) and the *ms* replies with an ARP response. The ARP response supplies the *node* with a name, IP address, and install server IP (and ethernet) address.
   - The *node* uses the TFTP protocol to bring in the Solaris OE miniroot from the *ms*, and, once executed, begins the bootstrap and software install process. The *node* uses the NFS protocol to mount the `/SOFTWARE/Solaris8_10-00` directory hosted by the *ms*.
2. The *node* opens the `/SOFTWARE/Solaris8_10-00/sysidcfg` file on the *ms* to identify the *node's* locality (timezone, name services, netmask, etc.).
3. The *node* opens the `/SOFTWARE/Solaris8_10-00/rules.ok` on the *ms* to identify the operating system install profile, as well as the post-install script.
4. The *node* opens the `/SOFTWARE/Solaris8_10-00/Profiles/sc3sconf1-n0.profile` file on the *ms* to identify internal disk partitioning and the content of the Solaris OE 8 10/00 software packages to be installed.
5. The *node* opens the `/SOFTWARE/Solaris8_10-00/Drivers/sc3sconf1-n0.driver` file on the *ms*, to identify the JumpStart server finish scripts to be executed after loading the Solaris OE 8 10/00 packages on each cluster node.
6. The *node* begins processing of the `/SOFTWARE/Solaris8_10-00/Profiles/sc3sconf1-n0.profile` on the *ms* and the Solaris OE 8 10/00 software packages (selected and deselected packages):
   - The internal boot disk and mirror are partitioned to include the root, swap, and `/globaldevices` filesystems, and to allocate space for the Solstice DiskSuite database.

- The Solaris OE disk label (VTOC) is created on the boot disk and its mirror.
- The root and `/globaldevices` filesystems are created.

7. The *node* begins installation of the selected software packages included in the `/SOFTWARE/Solaris8_10-00/Solaris_8/Product` directory on the *ms*.
8. Package installation is completed on the *node*, and the `/etc/vfstab` is updated with the new mount points:
   - The `/etc/hosts` file includes the *node* name and IP address.
   - The physical and logical devices (`/devices` and `/dev` are updated).
   - The boot block is installed on the boot disk.
9. The *node* starts execution of the `/SOFTWARE/Solaris8_10-00/Drivers/sc3sconf1-n0.driver` file, which invokes the `/SOFTWARE/Solaris8_10-00/Drivers/driver.init` file, which triggers the following events:
   - The `/.profile` file is updated with `$MANPATH` and `$PATH` variables to support the cluster environment.
   - The `/etc/motd` file is updated to include cluster post-installation instructions (see login message produced in the codebox below).

CODE EXAMPLE 1 `/etc/motd` file content for each cluster node

```
Apr 27 11:25:51 sc3sconf1-n1 login: ROOT LOGIN /dev/console
Sun Microsystems Inc.   SunOS 5.8       Generic February 2000

Execute "/usr/cluster/bin/scstat -p" to query the cluster installation.
Execute "/usr/cluster/bin/scconf -p" to query the cluster state.
Execute "/usr/cluster/bin/scsetup (once on any cluster node) to start
the cluster configuration.

Make sure the "default-root-password" is changed to preserve security.
```

   - The `/etc/hosts` file is updated to include the cluster elements.
   - The `/etc/defaultrouter` file is created with a gateway entry.
   - The `/etc/notrouter` file is created to prevent network routing.

10. The `/SOFTWARE/Solaris8_10-00/Drivers/sc3sconf1-n0.driver` file invokes the following JumpStart server finish scripts on the *ms*:
    - `/SOFTWARE/Solaris8_10-00/Finish/add_sds.fin`: Installs all Solstice DiskSuite software packages.
    - `/SOFTWARE/Solaris8_10-00/Finish/set_root_pw.fin`: Sets the node root password to abc (root password is changed during the first login).

- /SOFTWARE/Solaris8_10-00/Finish/remote_root_enable.fin: Modifies the /etc/default/login file to allow root login over the network.
- /SOFTWARE/Solaris8_10-00/Finish/ sds_file.fin: Copies the file /SOFTWARE/Solaris8_10-00/Files/metacvt script (Solstice DiskSuite tool) from the *ms* to the *node's* /usr/sbin directory.
- /SOFTWARE/Solaris8_10-00/Finish/set_n0-mdtab_file.fin: Copies the /SOFTWARE/Solaris8_10-00/Files/n0-md.tab on the *ms* to the *node's* /etc/lvm/md.tab directory.
- /SOFTWARE/Solaris8_10-00/Finish/set_n0-mirror_file.fin: Copies the /SOFTWARE/Solaris8_10-00/Files/S94n0-sds-mirror on the *ms* to the *node's* /etc/rc2.d directory.
- /SOFTWARE/Solaris8_10-00/Finish/add_patch.fin: Detects the Solaris 8 OE on the *node* and installs all patches included in the *ms's* /SOFTWARE/Solaris8_10-00/Patches/8_Recommended/patch_order file.
- The JumpStart server process finishes and invokes a system reboot.

Software Install Sequence During First Reboot

During the first reboot of the cluster node, the /etc/rc2.d/S94n0-sds-mirror script is executed, and the following steps represent the time-sequenced events it triggers:

1. The Solstice DiskSuite database is created.
2. The /etc/vfstab is modified to include Solstice DiskSuite metadevices.
3. All Solstice DiskSuite metadevices included in the /etc/lvm/md.tab are created and initialized.
4. The root, swap, and /globaldevices Solstice DiskSuite metadevices are mirrored.
5. The /etc/rc2.d/S91sconf-attach file is created to be executed during the next reboot.
6. The cluster node is identified to be a first node or second node and the Sun Cluster 3.0 software is installed. The first node is considered the cluster master and a special command line option needs to be used for the software install.
7. The default /etc/inet/ntp.conf is replaced with a copy of the /SOFTWARE/Solaris8_10-00/Files/ntp.conf file on the management server.

8. The `/etc/rc2.d/S94n0-sds-mirror` file is removed to prevent execution during the next reboot.

9. The root filesystem is locked using the `lockfs(1M)` command (after execution of the `metaroot(1M)` command to mirror the root partition) and a system reboot is invoked.

Note – The Solstice DiskSuite 4.1 User's Guide recommends using the `lockfs(1M) -fa` command option after execution of the `metaroot(1M)` command only. Application of the `lockfs(1M)` command is discouraged anywhere else.

Software Install Sequence During Second Reboot

During the second reboot of the first cluster node, the `/etc/rc2.d/S91sconf-attach` script is executed and attaches mirrors to the root, swap, and `/globaldevices` Solstice DiskSuite metadevices. The `/etc/rc2.d/S91sconf-attach` script removes itself to avoid execution during the next reboot. The Sun Cluster 3.0 software is now part of the kernel software and the cluster node automatically joins the cluster environment after each reboot.

Conclusions

The hardware and software stacks used to implement the Cluster Platform 220/1000 are integrated using best practices which maximize system availability. The hardware and software architecture presented in this article can be used by existing customers to create an equivalent configuration which minimizes problems with component interactions while improving system availability.

The Cluster Platform 220/1000 makes use of the JumpStart technology to automate the software installation of all cluster nodes and enables customers to quickly deploy highly available applications. Using the JumpStart technology is a best practice, since it provides a manageable environment which collects the latest software versions and patches, and reduces operator errors when recovering from catastrophic failures.

Author's Bio: Enrique Vargas

Enrique Vargas brings a wealth of large systems experience to Sun Microsystems, Inc. He specializes in high-end UNIX® offerings, including the Sun Enterprise 10000 server. Enrique joined the Sun's Enterprise Engineering group in 1997 from Amdahl Corporation, where he also focused on high-end Solaris Operating Environment systems and UTS (mainframe UNIX).

Enrique is a co-author of the "Sun Cluster Environment: Sun Cluster 2.2" and the "Resource Management" Sun BluePrints books. He has 20+ years of experience in the UNIX computer industry, which includes hardware design, diagnostics/drivers software development, system architecture and integration, field support, technical marketing, and sales.