A Procedure for Rolling Reload of NonStop Kernel Processors

Rolling reloads can be used to avoid system loads in some situations such as processor upgrades and certain software product revision (SPR) installations. However, care must be exercised when performing a rolling reload to avoid causing an outage. To minimize the chance of problems during or after a rolling reload, follow the guidelines and procedures presented in this document.

1 Rolling reload considerations

1.1 Rolling reload vs. system load

Many HP NonStop system customers successfully use rolling reloads and other on-line software installation techniques. However, problems can potentially occur when halting and reloading processors because of unexpected system or application behavior, load balancing, and non-standard configurations. If the next processor is halted before the previously halted processor has fully recovered, an outage is probable. Rolling reloads should be used only when all of the following conditions are true:

- It is not possible to schedule an outage for an SPR installation.
- The SPR(s) being installed is very important - for example, an SPR recommended by an Outage Prevention Notice (OPN).
- The SPR documentation states that the SPR can be installed with a rolling reload.
- The application and system configuration are well understood and fault-tolerant.
- It is possible to load balance the application after repeated processor halts and reloads.

Note: The time needed to perform a rolling reload is usually longer than the time needed to perform a system load. If the application must be halted during the rolling reload, you can perform a system load instead for a potentially shorter application outage.

1.2 Application considerations

Successful application recovery from a rolling reload requires fault-tolerant application(s) that can be quickly rebalanced following a processor halt and reload.

If your application does not meet these criteria and it is possible to have an application interruption (but not a system load), an option is shutting down the application, performing the rolling reload, and then restarting the application. Note, however, that the resulting application interruption will probably be longer than that from a system load.

1.3 A few other tips

- Whenever possible, test the rolling reload on a development system using the production application and expected production workload.
- Perform the rolling reload when the system has its minimal utilization.
- Carefully follow the steps described in the remainder of this document.

2 Planning a rolling reload.

Rolling reloads are significant system events and require careful planning. The first step is to gather and evaluate all documentation pertaining to the planned update, such as SPR documentation or Hotstuff messages. The Global Customer Support Center (GCSC) is an excellent resource for rolling reload assistance. Contact the GCSC after completing a rolling reload plan. They can provide assistance with specific NonStop products that might be involved with or affected by the rolling reload. GCSC might also be able to provide information based on the experience of other customers who have installed the planned update using a rolling reload.

The remainder of this section identifies specific topics to consider when planning a rolling reload.

2.1 Verify fault-tolerant applications (including recovery)

Make sure that your application can tolerate and recover from sequential halts of different processors. Some applications can recover from a single processor halt but may have difficulty with repeated halts. Also make sure that your application can recover and rebalance when the processor is reloaded. Consider the recovery of cluster-based and Expand-based applications, and the potential impact on applications that use the Remote Duplicate Database Facility (RDF).

2.2 Choose the processor reload sequence

Unless there are specific reasons for choosing a different reload sequence, rolling reloads should update all even processor numbers first and then update all odd processors. This strategy provides extra time for even-odd process pairs to recover from processor halts and reloads. If the primary and backups for key operating system and application processes are not assigned to even-odd processor pairs, consider changing the order of the rolling reloads to separate the reloads of the processors or processes configured as pairs as much as practical.

2.3 Select rolling reload date and time

Choose a time in the day, week, or year when the system utilization is expected to be at its minimum. Even if your applications work in the presence of single processor halts and will fully recover after a processor halt, verify that no processor is utilized more than 45 percent before beginning the rolling reload.

2.4 Determine which processors need to be reloaded

Review previous software and firmware installations. The change being installed by the rolling reload might already be completed on some processors, for example, a processor that was reloaded using the new software due to a processor halt. Only reload the processors that must be updated with new software.

2.5 Determine how much time is required for a rolling reload

The time required for a rolling reload is the sum of the time to (1) halt each processor with a complete takeover by backup processes, (2) apply the change to each processor,

(3) reload each processor with a full recovery by each process, and (4) perform system and application recovery after all processors are reloaded. The following text describes how to determine the amount of time needed for each of these steps.

(1) **Determine how much time is required for backup processes to fully takeover**. The time needed for backup processes to fully takeover after a halt varies depending on processor performance, system load, processor loading, and the subsystems configured for a specific processor. Use an amount of time based on your operational experience, or, if no predetermined value is available, expect this time to be on the order of 2 minutes for a lightly loaded system.

(2) **Determine how much time is required to apply the change.** Consult the applicable documentation.

(3) **Determine how much time is required to complete a processor reload.** The time needed to fully reload a processor after a halt varies depending on processor performance, system load, processor loading, and the subsystems configured for a specific processor. Reload time is normally significantly longer than the time needed for the command-line RELOAD program "finished" response. Use an amount of time based on your operational experience, or, if no predetermined value is available, expect this time to be on the order of 3 minutes for a lightly loaded system. This time should include the time needed for each process to fully recover.

Determine how much time is required to rebalance the system. System rebalancing time depends primarily on the application and system configuration. Additional work might be required for systems involved in network transactions or RDF configurations, for example, allowing the RDF processing to catch up sending transactions to a backup system. You may also want to review the COMM access lists in order to anticipate any network routing issues. Accurately determining time durations might require some testing on a development system. For the best results, verify the amount of time for each activity and in total by rehearsing the rolling reload on a development system.

2.6 Create OBEY command files

In many cases, you can use OBEY command files to help verify that it is safe to continue with the next rolling reload step. After reading Section 3, Rolling reload preparation, determine if a set of OBEY command files can simplify and reduce typing errors during your rolling reload. If so, prepare the OBEY command files, testing them if possible on a development system. You cannot automate the entire procedure because some of the steps require manual intervention. For example, you must check the HP NonStop Open System Management (OSM) or Compaq TSM Service Application display to determine that there are no unexpected events. The OBEY command files must allow for the manual checking required, so they should not simply perform a set of operations, delay for some minutes, and repeat for the next processor. In particular, a single OBEY file should not contain both a halt and a reload since manual checking is required before the reload.

3 Rolling reload preparation

This section describes the activities that should be performed prior to starting the rolling reload.

3.1 Prepare the SPRs to be installed

Code to be updated should be present on the preferred load subvolume (often $SYSTEM.SYSnn, $SYSTEM.SYSTEM, or other $SYSTEM.Z*) depending on subsystem installation methods. For the best results, use DSM/SCM Build/Apply.

If FUP DUP must be used instead, there is a good chance the next version of code built with DSM/SCM will not include the changes, leading to an accidental future downgrade to the earlier version. If FUP DUP is used, you should prepare a DSM/SCM Build with the updates for the next "official" update with the new versions of the code. Use of FUP DUP can prevent a rollback from working, which is why it is not recommended.

If new code is placed on the system without using DSM/SCM, a compatible rollback strategy must be developed.

3.2 Update Firmware

The relevant firmware products should match for all currently running processor multifunction (PMF) CRUs. All embedded firmware products (such as boot millicode, service processor (SP), and controller firmware) must be consistent and must be the same as the latest revision of that product installed on the system. Use OSM/TSM to verify the consistency of the firmware releases prior to performing the rolling reload. Fix any discrepancies by using OSM/TSM before continuing.

Note: A NonStop system operation best practice is to keep boot millicode and SP firmware consistent and as current as possible. Because SP firmware can be updated without system disruption, NED recommends keeping the revision of SP firmware within one revision of the current SPR.

3.3 Verify system health

No OSM/TSM alarms should exist on the internal or external ServerNet fabrics. No disk path downs, disk media errors, or path switch alarms should exist. Examine and correct any problems found prior to the rolling reload. Correct any OSM/TSM alarms in power/environment. SCF STATUS SERVERNET $ZSNET must show that both X and Y fabric connectivity is available to all installed processors.

All processors except one at most (the first to reload) must be executing the NonStop Kernel operating system.

4 Rolling reload implementation

Before starting the rolling reload, complete the procedures described in Sections 2 and 3. Verify the fault-tolerance of the system, paying particular attention to the ServerNet Fabrics and processor-to-ServerNet addressable controller (SAC) relationships.

To implement a rolling reload, perform the pre-checks in Section 4.1, halt and reload each processor one at a time using the procedures in Sections 4.2-4.4, and perform the checks described in Section 4.5 after all processors are reloaded. The processors should be halted and reloaded in the order selected in Section 2.2, Choose the processor reload sequence.

If a processor upgrade is the reason for the rolling reload, use the guided processor replacement procedure instead of Sections 4.2-4.4, but be sure to verify that there are no new OSM/TSM alarms before each halt and before each reload. If necessary, the application checks described in Sections 4.2-4.4 should still be performed.

4.1 Pre-checks before halting the first processor

- Verify that *`SCF STATUS SERVERNET $ZSNET`* shows no problems.
- Verify that *`SCF STATUS SAC $ZZLAN.*, DETAIL`* shows no problems.
- Verify that all SLSA SACs are in the STARTED state, not STARTING. Check the access from both fabrics and all processors configured for each SAC.
- Verify that there are no new OSM/TSM alarms.
- Perform similar checks for applications and other subsystems such as ServerNet/FX and Telco-specific SACs.

4.2 Halt a processor

Use the OSM/TSM Low Level Link (LLL) or Service Connection to halt the processor. In general, do **not** perform a hard reset unless you require a very low-level re-initialization.

- Wait a minimum of one minute before proceeding with the next step.
- Verify that processor utilization (measured by ViewSys or other up system management application) stabilizes.
- Verify that the event flood triggered by halting a processor has receded by checking events in the two event logs ($0 and $ZLOG) using the “real-time” setting.

  Caution: Some subsystems might continue to generate events indicating that they have no backup, and these events should be ignored until after the reload.
- Verify that *`SCF STATUS SERVERNET $ZSNET`* shows no problems.
- Verify that *`SCF STATUS SAC $ZZLAN.*,DETAIL`* shows no problems
- Verify that all SLSA SACs are in the STARTED state, not STARTING. Check the access from both fabrics and all processors configured for each SAC.
- Verify that there are no new OSM/TSM alarms (other than the expected halted processor indication).

- Perform similar checks for applications and other subsystems such as ServerNet/FX and Telco-specific SACs.
- If necessary, verify the health of the applications.
- Verify that the time since the halt is about what was determined in the planning phase (Section 2.5, Determine how much time is required for a rolling reload).

4.3 Execute update step(s), if needed before reload.

You might perform a rolling reload to install new software or firmware on the processors. That update must now be applied to the halted processor. In many cases, the reload operation in Section 4.4 automatically installs the new software, so this step is often not needed.

4.4 Reload the halted processor.

Use *RELOAD <n>, PRIME,* or use the processor OSM Reload action. Use of the "PRIME" option forces a new copy of boot millicode to be loaded. This load is normally not needed, but takes less than one minute and ensures a consistent processor reset and copy of millicode.

Note: Do not specify the ServerNet fabric for the reload. (The X fabric will be tried first if no fabric is specified.)

Note: Sometimes, switching from the primary process to the backup process in a NonStop process pair is costly in terms of system resources or system response time, even though the backup process is "ready." For a system with heavy load and tight response-time requirements, the following steps might be needed to avoid causing an application outage while all the disks re-primary and the disk process preloads very large disk caches.

(1) Use the "*RELOAD <n>, PRIME, NOSWITCH*" option to reload the halted processor.
(2) Switch the disk processes one at a time, limiting the impact of the disk cache reload because only one cache at a time is reloaded.
(3) You might also need to pre-switch the disk processes one at a time for this type of system prior to halting a processor.

- Verify that no exceptions are listed in the command-line RELOAD response text. All system resources should be re-integrated as reported by the RELOAD response text.
- Wait a minimum of one minute before proceeding with the next step.
- Verify that processor utilization (measured by ViewSys or other up system management application) stabilizes.
- Verify that the event flood triggered by halting a processor has receded by checking events in the two event logs ($0 and $ZLOG) using the "real-time" setting.

  Caution: All subsystems should stop generating events indicating they have no backup. Wait until these events cease.
- Verify that *SCF STATUS SERVERNET $ZSNET* shows no problems.
- Verify that *SCF STATUS SAC $ZZLAN.*,DETAIL* shows no problems.

- Verify that all SLSA SACs are in the STARTED state, not STARTING. Check the access from both fabrics and all processors configured for each SAC.
- Verify that there are no new OSM/TSM alarms.
- Perform similar checks for applications and other subsystems such as ServerNet/FX and Telco-specific SACs.
- If necessary, verify the health of the applications.
- If necessary, verify that the updates have been installed and are active on this processor.
- Verify that the time since the reload is about what was determined in the planning phase (Section 2.5, Determine how much time is required for a rolling reload).

4.5 Verify and balance system after all processors are reloaded.

If necessary, verify that the updates have been installed and are active on all impacted processors. Check that remote application processes have correctly handled the DOWN and UP changes for the affected local processors. These steps are especially important if the application includes network transactions or is part of an RDF configuration. Rebalance the system and application as necessary.

5 Summary

You can use rolling reloads to avoid system loads in some situations, such as processor upgrades and certain SPR installations. However, follow this document's guidelines and procedures when performing a rolling reload to minimize the chance of an outage. Take your time, and if you have questions or concerns about rolling reloads, contact the GCSC for assistance.