Pratiques opérationnelles recommandées relatives à CRS-1 et IOS XR

Contenu

Introduction
Conditions préalables
Conditions requises
Composants utilisés
Conventions
Aperçu de Cisco IOS XR
Processus et thread
États de processus et de thread
Dépassement synchrone de message
États de processus et de processus bloqués
Importants processus et leurs fonctions
Netio
Le groupe entretient le processus (le système de préférences généralisées)
Téléchargeur de contenu en vrac BCDL
Messagerie légère (LWM)
Envmon
Introduction de la matrice CRS-1
L'avion de matrice
Surveillance de matrice
Contrôlez l'aperçu plat
Configuration de Catalyst 6500
Gestion d'avion de contrôle de Multi-châssis
ROMMON et Monlib
Instructions de mise à jour
Aperçu PLIM et de MSC
Surabonnement PLIM
Gestion de la configuration
Sécurité
LPTS
Comment est-ce qu'un paquet interne est expédié ?
Hors de la bande
Informations connexes

Introduction

Ce document vous aide à comprendre ces derniers :

- Processus et thread
- Matrice CRS-1
- Plan de contrôle
- Rommon et Monlib
- Module d'interface de couche physique (PLIM) et carte modulaire de service (MSC)
- Gestion de la configuration
- Sécurité
- Hors de la bande
- Protocole de gestion de réseau simple (SNMP)

Conditions préalables

Conditions requises

Cisco recommande que vous ayez la connaissance du Cisco IOS® XR.

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Logiciel Cisco IOS XR
- CRS-1

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Aperçu de Cisco IOS XR

Le Cisco IOS XR est conçu pour mesurer. Le noyau est une architecture de Microkernel ainsi il fournit seulement des services essentiels tels que la gestion de processus, l'établissement du programme, les signaux, et les temporisateurs. Tous autres services tels que des systèmes de fichiers, gestionnaires, piles de protocoles et demandes sont examinés comme gestionnaires de ressources et passage dans l'espace d'utilisateur protégé par mémoire. Ces autres services peuvent être ajoutés ou lors de l'exécution retiré, qui dépend de la conception de programme. L'empreinte de pas de Microkernel est le kb seulement 12. Le Microkernel et le système d'exploitation sous-jacent est des logiciels qnx, et s'appelle Neutrino. QNX se spécialise dans la conception de système d'exploitation en temps réel. Le Microkernel est de préemption, et le programmateur est priorité basée. Ceci s'assure que la commutation de contexte entre les processus est très rapide, et les thread les plus prioritaires ont toujours accès à la CPU en cas de besoin. Ce sont certains d'avantages dont le Cisco IOS XR profite. Mais, le plus grand avantage est la conception d'héritage des transmissions de processus inter dans le noyau de systèmes

d'exploitation.

Le Neutrino est un message passant le système d'exploitation, et les messages sont les moyens de base des communications entre processus parmi tous les thread. Quand un serveur particulier veut fournir un service, il crée un canal pour les messages permutants. Les clients relient aux serveurs le canal en traçant directement au descripteur de fichier approprié afin d'utiliser le service. Toutes les transmissions entre le client et serveur est par le même mécanisme. C'est un avantage énorme pour un ordinateur superbe, qui CRS-1 est. Considérez ces derniers quand une opération "lecture" locale est exécutée sur un noyau standard UNIX :

- Interruption de logiciel dans le noyau.
- Répartitions de noyau dans le système de fichiers.
- Des données sont reçues.

Considérez ces derniers dans le cas distant :

- Interruption de logiciel dans le noyau.
- Le noyau achemine le NFS.
- Le NFS appelle le composant réseau.
- Le distant achemine le composant réseau.
- Le NFS s'appelle.
- Le noyau achemine le système de fichiers.

La sémantique pour les gens du pays lus et le distant lu ne sont pas identique. Les arguments et les paramètres pour le verrouillement de fichier et placer des autorisations sont différents.

Considérez le QNX cas local :

- Interruption de logiciel dans le noyau.
- Le noyau exécute le message passant dans le système de fichiers.

Considérez le cas non local :

- Interruption de logiciel dans le noyau.
- Le noyau entre dans QNET, qui est le mécanisme de transport IPC.
- QNET entre dans le noyau.
- Le noyau achemine le système de fichiers.

Toute la sémantique qui concernent l'argument passant et les paramètres de système de fichiers sont identiques. Tout a été découplé à l'interface IPC qui permet le client et serveur à isoler complètement. Ceci signifie que n'importe quel processus peut fonctionner n'importe où à tout moment. Si un processeur particulier d'artère est des demandes de service trop occupées, vous pouvez facilement migrer ces services vers une CPU différente qui fonctionne sur un DRP. Un ordinateur superbe qui dirige différents services sur différentes CPU se propage à travers les plusieurs noeuds qui peuvent facilement communiquer avec n'importe quel autre noeud. L'infrastructure est en place afin de fournir l'occasion de mesurer. Cisco a utilisé cet avantage et a écrit le logiciel supplémentaire qui s'accroche dans les exécutions de principe du message passant le noyau qui permet au routeur CRS pour mesurer aux milliers de Noeuds, où un noeud, dans ce cas une CPU, exécute un exemple du SYSTÈME D'EXPLOITATION, si c'est un processus d'artère (RP), un processeur distribué d'artère (DRP), une carte modulaire de services (MSC), ou un processeur de commutateur (fournisseur de services).

Processus et thread

Dans les limites du Cisco IOS XR, un processus est une zone mémoire protégée qui contient un ou plusieurs thread. Du point de vue de programmeurs, les thread effectuent le travail, et chacun se termine un chemin logique d'exécution afin d'effectuer une tâche spécifique. La mémoire que les thread exigent pendant l'écoulement de l'exécution appartient au processus ils fonctionnent en dedans, protégé contre tous les autres thread de processus. Un thread est une unité d'exécution, avec un contexte d'exécution qui inclut une pile et s'enregistre. Un processus est un groupe de thread qui partagent un espace d'adressage virtuel, bien qu'un processus puisse contenir un thread simple mais contient plus souvent plus. Si un autre thread dans différentes tentatives d'un processus d'écrire à la mémoire dans votre processus, le processus offensant est détruit. S'il y a plus d'un thread qui fonctionne dans votre processus, alors ce thread a accès à la même mémoire dans votre processus, et en conséquence est capable pour remplacer les données d'un autre thread. Terminez-vous les étapes dans une procédure afin de mettre à jour la synchronisation aux ressources afin d'empêcher ce thread dans le même processus.

Un thread emploie un objet appelé une exclusion mutuelle (MUTEX) afin d'assurer l'exclusion mutuelle aux services. Le thread qui a le MUTEX est le thread qui peut écrire à une zone mémoire particulière comme exemple. D'autres thread qui n'ont pas le MUTEX ne peuvent pas. Il y a également d'autres mécanismes afin d'assurer la synchronisation aux ressources, et ce sont des sémaphores, des variables conditionnelles ou Condvars, des barrières, et Sleepons. Ceux-ci ne sont pas discutés ici, mais ils fournissent des services de synchronisation comme partie de leurs fonctions. Si vous égalisez les principes discutés ici au Cisco IOS, alors le Cisco IOS est un processus simple actionnant beaucoup de thread, avec tous les thread qui ont accès au même espace mémoire. Mais, le Cisco IOS appelle ces processus de thread.

États de processus et de thread

Dans le Cisco IOS XR il y a des serveurs qui fournissent les services et les clients qui utilisent les services. Un processus particulier peut avoir un certain nombre de thread qui fournissent le même service. Un autre processus peut avoir un certain nombre de clients qui pourraient avoir besoin d'un service particulier à tout moment. Access aux serveurs n'est pas toujours disponible, et si un accès de demande de client à un service qu'il se repose là et attend le serveur pour être libre. Dans ce cas le client est dit bloqué. Ceci s'appelle un client-server model de blocage. Le client pourrait être bloqué parce qu'il attend une ressource telle qu'un MUTEX, ou étant donné que le serveur n'a pas encore répondu.

Émettez une commande **OSPF de processus d'exposition** afin de vérifier le statut des thread dans le processus OSPF :

```
RP/0/RP1/CPU0:CWDCRS#show process ospf
                  Job Id: 250
                     PID: 110795
         Executable path: /disk0/hfr-rout-3.2.3/bin/ospf
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Tue Jul 18 13:10:06 2006
           Process state: Run
           Package state: Normal
      Started on config: cfg/gl/ipv4-ospf/proc/101/ord_a/routerid
                    core: TEXT SHAREDMEM MAINMEM
              Max. core: 0
              Placement: ON
```

```
            startup_path: /pkg/startup/ospf.startup
                   Ready: 1.591s
               Available: 5.595s
        Process cpu time: 89.051 user, 0.254 kernel, 89.305 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
250    1      40K  10 Receive       0:00:11:0509 ospf
250    2      40K  10 Receive       0:01:08:0937 ospf
250    3      40K  10 Receive       0:00:03:0380 ospf
250    4      40K  10 Condvar       0:00:00:0003 ospf
250    5      40K  10 Receive       0:00:05:0222 ospf
```

Notez que le processus OSPF est donné une identification des tâches (JID), qui est 250. Ceci ne change jamais sur un routeur courant et généralement sur une version particulière de Cisco IOS XR. Dans le processus OSPF il y a de cinq filète chacun avec leur propre ID de thread (TID). Énuméré est l'espace de pile pour chaque thread, la priorité de chaque thread et son état.

Dépassement synchrone de message

On lui cite précédemment que QNX est un message passant le système d'exploitation. C'est réellement un message synchrone passant le système d'exploitation. Les beaucoup de les questions du système d'exploitation sont reflétées à la Messagerie synchrone. On ne lui dit pas que le dépassement synchrone de message pose tous les problèmes, mais plutôt le symptôme du problème est reflété dans le dépassement synchrone de message. Puisqu'il est synchrone, le cycle de vie ou les informations d'état est facilement accessible à l'opérateur CRS-1, qui facilite le processus de dépannage. Le message passant le cycle de vie est semblable à ceci :

- Un serveur crée un canal de message.
- Un client se connecte au canal d'un serveur (analogue au posix ouvert).
- Un client envoie un message à un serveur (MsgSend) et des attentes une réponse et des blocs.
- Le serveur reçoit (MsgReceive) un message d'un client, traite le message, et répond au client.
- Le client débloque et traite la réponse du serveur.

Ce client-server model de blocage est le dépassement synchrone de message. Ceci signifie que le client envoie un message et des blocs. Le serveur reçoit le message, le traite, répond de nouveau au client et alors le client débloque. Ce sont les détails spécifiques :

- Le serveur attend REÇOIVENT dedans l'état.
- Le client envoie un message au serveur et devient BLOQUÉ.
- Le serveur reçoit le message et débloque, si attendant dedans recevez l'état.
- Le client se déplace à l'état de RÉPONSE.
- Le serveur se déplace à l'état COURANT.
- Processus de serveur le message.
- Le serveur répond au client.
- Le client débloque.

Émettez la commande de **processus d'exposition** afin de voir dans quels états le client et serveur sont.

```
RP/0/RP1/CPU0:CWDCRS#show processes
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
1      1       0K  0  Ready       320:04:04:0649 procnto-600-smp-cisco-instr
1      3       0K  10 Nanosleep     0:00:00:0043 procnto-600-smp-cisco-instr
1      5       0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      7       0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
```

```
1      8       0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      11      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      12      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      13      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      14      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      15      0K  19 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      16      0K  10 Receive       0:02:01:0207 procnto-600-smp-cisco-instr
1      17      0K  10 Receive       0:00:00:0015 procnto-600-smp-cisco-instr
1      21      0K  10 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      23      0K  10 Running       0:07:34:0799 procnto-600-smp-cisco-instr
1      26      0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      31      0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      33      0K  10 Receive       0:00:00:0000 procnto-600-smp-cisco-instr
1      39      0K  10 Receive       0:13:36:0166 procnto-600-smp-cisco-instr
1      46      0K  10 Receive       0:06:32:0015 procnto-600-smp-cisco-instr
1      47      0K  56 Receive       0:00:00:0029 procnto-600-smp-cisco-instr
1      48      0K  10 Receive       0:00:00:0001 procnto-600-smp-cisco-instr
1      72      0K  10 Receive       0:00:00:0691 procnto-600-smp-cisco-instr
1      73      0K  10 Receive       0:00:00:0016 procnto-600-smp-cisco-instr
1      78      0K  10 Receive       0:09:18:0334 procnto-600-smp-cisco-instr
1      91      0K  10 Receive       0:09:42:0972 procnto-600-smp-cisco-instr
1      95      0K  10 Receive       0:00:00:0011 procnto-600-smp-cisco-instr
1      103     0K  10 Receive       0:00:00:0008 procnto-600-smp-cisco-instr
74     1       8K  63 Nanosleep     0:00:00:0001 wd-mbi
53     1      28K  10 Receive       0:00:08:0904 dllmgr
53     2      28K  10 Nanosleep     0:00:00:0155 dllmgr
53     3      28K  10 Receive       0:00:03:0026 dllmgr
53     4      28K  10 Receive       0:00:09:0066 dllmgr
53     5      28K  10 Receive       0:00:01:0199 dllmgr
270    1      36K  10 Receive       0:00:36:0091 qsm
270    2      36K  10 Receive       0:00:13:0533 qsm
270    5      36K  10 Receive       0:01:01:0619 qsm
270    7      36K  10 Nanosleep     0:00:22:0439 qsm
270    8      36K  10 Receive       0:00:32:0577 qsm
67     1      52K  19 Receive       0:00:35:0047 pkgfs
67     2      52K  10 Sigwaitinfo   0:00:00:0000 pkgfs
67     3      52K  19 Receive       0:00:30:0526 pkgfs
67     4      52K  10 Receive       0:00:30:0161 pkgfs
67     5      52K  10 Receive       0:00:25:0976 pkgfs
68     1       8K  10 Receive       0:00:00:0003 devc-pty
52     1      40K  16 Receive       0:00:00:0844 devc-conaux
52     2      40K  16 Sigwaitinfo   0:00:00:0000 devc-conaux
52     3      40K  16 Receive       0:00:02:0981 devc-conaux
52     4      40K  16 Sigwaitinfo   0:00:00:0000 devc-conaux
52     5      40K  21 Receive       0:00:03:0159 devc-conaux
65545  2      24K  10 Receive       0:00:00:0487 pkgfs
65546  1      12K  16 Reply         0:00:00:0008 ksh
66     1       8K  10 Sigwaitinfo   0:00:00:0005 pipe
66     3       8K  10 Receive       0:00:00:0000 pipe
66     4       8K  16 Receive       0:00:00:0059 pipe
66     5       8K  10 Receive       0:00:00:0149 pipe
66     6       8K  10 Receive       0:00:00:0136 pipe
71     1      16K  10 Receive       0:00:09:0250 shmwin_svr
71     2      16K  10 Receive       0:00:09:0940 shmwin_svr
61     1       8K  10 Receive       0:00:00:0006 mqueue
```

États de processus et de processus bloqués

Émettez la commande **bloquée par processus d'exposition** afin de voir quel processus sont dans l'état bloqué.

```
RP/0/RP1/CPU0:CWDCRS#show processes blocked
  Jid       Pid Tid                 Name State  Blocked-on
65546      4106   1                  ksh Reply    4104  devc-conaux
  105     61495   2              attachd Reply   24597  eth_server
  105     61495   3              attachd Reply    8205  mqueue
  316     65606   1           tftp_server Reply    8205  mqueue
  233     90269   2               lpts_fm Reply   90223  lpts_pa
  325    110790   1             udp_snmpd Reply   90257  udp
  253    110797   4                ospfv3 Reply   90254  raw_ip
  337    245977   2                fdiagd Reply   24597  eth_server
  337    245977   3                fdiagd Reply    8205  mqueue
65762   5996770   1                  exec Reply       1  kernel
65774   6029550   1                  more Reply    8203  pipe
65778   6029554   1       show_processes Reply       1  kernel
RP/0/RP1/CPU0:CWDCRS#
```

Le dépassement synchronisé de message te permet de dépister facilement le cycle de vie de transmission d'interprocessus entre les différents thread. À tout moment, un thread peut être dans un état spécifique. Un état bloqué peut être un symptôme d'un problème. Ceci ne signifie pas que si un thread est dans l'état bloqué puis il y a un problème, ainsi n'émet pas la commande **bloquée par processus d'exposition** et ouvre une valise avec le support technique de Cisco. Les thread bloqués sont également très normaux.

Notez la sortie précédente. Si vous regardez le premier thread dans la liste, notez-la est le ksh, et sa réponse est bloquée sur le devc-conaux. Le client, le ksh dans ce cas, a envoyé un message au processus de devc-conaux, le serveur, qui est devc-conaux, réponse ksh d'attentes bloquée jusqu'à ce qu'elle réponde. Le Ksh est le shell unix que quelqu'un utilise sur la console ou le port auxiliaire. Le Ksh attend l'entrée de la console, et s'il n'y en a aucun parce que l'opérateur ne tape pas, alors il reste bloqué jusqu'à une telle heure qu'il traite une certaine entrée. Après le traitement, le ksh revient à la réponse bloquée sur le devc-conaux.

C'est normal et n'illustre pas un problème. Le point est que les thread bloqués sont normaux, et il dépend quelle version XR, le type de système vous ont, de ce que vous avez configuré et qui fait ce que cela modifie la sortie de la commande **bloquée par processus d'exposition**. L'utilisation de la commande **bloquée par processus d'exposition** est une bonne manière de commencer à dépanner des problèmes de type de SYSTÈME D'EXPLOITATION. S'il y a un problème, par exemple la CPU est élevée, alors emploie la commande précédente afin de voir si quelque chose regarde en dehors de la normale.

Comprenez ce qui est normal pour votre routeur de fonctionnement. Ceci fournit une spécification de base pour que vous utilisiez comme comparaison quand vous dépannez les cycles de vie de processus.

Àtout moment, un thread peut être dans un état particulier. Cette table fournit une liste des états :

Si l'état est :	Le thread est :
MORT	Mort. Le noyau attend de libérer les ressources en thread.
S'EXÉCUTER	Activement s'exécutant sur une CPU
PRÊT	Ne pas s'exécuter sur une CPU mais est prêt à fonctionner
ARRÊTÉ	Interrompu (signal SIGSTOP)
ENVOYEZ	Attendre un serveur pour recevoir un message
RECEVEZ	Attendre un client pour envoyer un

	message
RÉPONSE	Attendre un serveur pour répondre à un message
PILE	En attendant plus de pile pour être allouez
WAITPAGE	Attendre le gestionnaire de processus pour résoudre un défaut de page
SIGSUSPEND	Attendre un signal
SIGWAITINFO	Attendre un signal
NANOSLEEP	Sommeil pendant une période
MUTEX	Attendre pour saisir un MUTEX
CONDVAR	Attendant une variable conditionnelle à signaler
JOIGNEZ	Attendre la fin d'un autre thread
INTR	Attendre une interruption
SEM	Attendre pour saisir une sémaphore

Importants processus et leurs fonctions

Le Cisco IOS XR a beaucoup de processus. Ce sont des quelques importants avec leurs fonctions expliquées ici.

Moniteur système de surveillance (WDSysmon)

C'est un service donné pour la détection du processus s'arrête et des états de taille mémoire basse. La mémoire basse peut se produire en raison d'une fuite de mémoire ou d'une autre circonstance étrangère. Un coup peut être le résultat d'un certain nombre de conditions telles que des blocages de processus, des boucles infinies, des calages de noyau ou des erreurs de établissement du programme. Dans n'importe quel environnement multifil le système peut obtenir dans un état connu sous le nom d'état de blocage, ou juste simplement blocage. Un blocage peut se produire quand un ou plusieurs thread ne peuvent pas continuer en raison du conflit de ressource. Par exemple, filetez A peut envoyer un message pour fileter B tandis que simultanément le thread B envoie un message pour fileter le R. Les deux thread attendent sur l'un l'autre et peuvent être dedans envoient l'état bloqué, et les deux thread attendent pour toujours. C'est un cas simple qui implique deux thread, mais si un serveur est responsable d'une ressource qui est utilisée par beaucoup de thread est bloqué dans un autre thread, puis les nombreux thread qui demandent accès à cette ressource peut être envoient attendre bloqué sur le serveur.

Les blocages peuvent se produire entre quelques thread, mais peuvent affecter d'autres thread en conséquence. Des blocages sont évités par bonne conception de programme, mais indépendamment de la façon dont un programme est conçu et magnifiquement écrit. Parfois une séquence d'opérations particulière qui sont personne à charge de données avec des synchronisations spécifiques peut entraîner un blocage. Les blocages ne sont pas toujours déterministes et sont généralement très difficiles à se reproduire. WDSysmon a beaucoup de thread avec un qui fonctionnent que le Neutrino prend en charge, aux 63 les plus prioritaires. S'exécuter à la priorité 63 l'assure que le thread obtient le temps- CPU dans un environnement de établissement du programme de préemption basé par priorité. WDSysmon fonctionne avec la capacité et les montres de surveillance de matériel au-dessus des processus de logiciel qui

recherchent des états de coup. Quand de telles conditions sont détectées, WDSysmon collecte les informations supplémentaires autour de la condition, peut coredump le processus ou le noyau, écrire aux Syslog, exécutent des scripts, et détruisent les processus aboutis à une impasse. Personne à charge sur la façon dont radical le problème est, il peut initier un commutateur de processeur d'artère plus d'afin de mettre à jour l'exploitation du système.

```
RP/0/RP1/CPU0:CWDCRS#show processes wdsysmon
                  Job Id: 331
                     PID: 36908
         Executable path: /disk0/hfr-base-3.2.3/sbin/wdsysmon
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Tue Jul 18 13:07:36 2006
           Process state: Run
           Package state: Normal
                    core: SPARSE
                Max. core: 0
                   Level: 40
                Mandatory: ON
             startup_path: /pkg/startup/wdsysmon.startup
             memory limit: 10240
                   Ready: 0.705s
        Process cpu time: 4988.295 user, 991.503 kernel, 5979.798 total

JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
331    1      84K  19 Receive       0:00:00:0029 wdsysmon
331    2      84K  10 Receive       0:17:34:0212 wdsysmon
331    3      84K  10 Receive       0:00:00:0110 wdsysmon
331    4      84K  10 Receive       1:05:26:0803 wdsysmon
331    5      84K  19 Receive       0:00:06:0722 wdsysmon
331    6      84K  10 Receive       0:00:00:0110 wdsysmon
331    7      84K  63 Receive       0:00:00:0002 wdsysmon
331    8      84K  11 Receive       0:00:00:0305 wdsysmon
331    9      84K  20 Sem           0:00:00:0000 wdsysmon
```

Le WDSysmon de processus a neuf thread. Quatre exécutés à la priorité 10, les autres quatre sont à 11, à 19, à 20 et à 63. Quand un processus est conçu, le programmeur considère soigneusement la priorité que chaque thread dans le processus devrait être donné. Comme discuté précédemment, le programmateur est la priorité basée, qui signifie qu'un thread plus prioritaire acquiert toujours un d'une priorité plus basse. La priorité 63 est la plus prioritaire qu'un thread peut exécuter à, qui est le thread 7 dans ce cas. Le thread 7 est le thread de watcher, le thread des porcs cette CPU de pistes. Il doit fonctionner à une haute priorité que les autres thread qu'elle les observe autrement ne pourraient pas obtenir l'occasion de s'exécuter du tout, ce qui l'empêche des étapes qu'elle a été conçue pour exécuter.

Netio

Dans le Cisco IOS, il y a le concept du changement de commutation rapide et de processus. La commutation rapide utilise code CEF et se produit au temps d'interruption. La commutation de processus utilise l'ip_input, qui est le code de Commutation IP, et est un processus programmé. Sur des Plateformes plus à extrémité élevé la commutation de CEF est faite dans le matériel, et l'ip_input est programmé sur la CPU. L'équivalent de l'ip_input dans le Cisco IOS XR est Netio.

```
P/0/RP1/CPU0:CWDCRS#show processes netio
```

```
                  Job Id: 241
                     PID: 65602
          Executable path: /disk0/hfr-base-3.2.3/sbin/netio
               Instance #: 1
                     Args: d
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
  Max. spawns per minute: 12
             Last started: Tue Jul 18 13:07:53 2006
            Process state: Run
            Package state: Normal
                     core: DUMPFALLBACK COPY SPARSE
                Max. core: 0
                    Level: 56
                Mandatory: ON
             startup_path: /pkg/startup/netio.startup
                    Ready: 17.094s
         Process cpu time: 188.659 user, 5.436 kernel, 194.095 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
241    1     152K  10 Receive       0:00:13:0757 netio
241    2     152K  10 Receive       0:00:10:0756 netio
241    3     152K  10 Condvar       0:00:08:0094 netio
241    4     152K  10 Receive       0:00:22:0016 netio
241    5     152K  10 Receive       0:00:00:0001 netio
241    6     152K  10 Receive       0:00:04:0920 netio
241    7     152K  10 Receive       0:00:03:0507 netio
241    8     152K  10 Receive       0:00:02:0139 netio
241    9     152K  10 Receive       0:01:44:0654 netio
241    10    152K  10 Receive       0:00:00:0310 netio
241    11    152K  10 Receive       0:00:13:0241 netio
241    12    152K  10 Receive       0:00:05:0258 netio
```

Le groupe entretient le processus (le système de préférences généralisées)

Il y a un besoin de transmission dans n'importe quel superordinateur avec plusieurs milliers de Noeuds ce chaque exécuté son propre exemple du noyau. En Internet, un aux beaucoup transmission est efficacement fait par l'intermédiaire des protocoles de multifusion. Le système de préférences généralisées est le protocole interne de multifusion qui est utilisé pour l'IPC dans CRS-1. Le système de préférences généralisées fournit un aux beaucoup la transmission fiable de groupe qui est sans connexion avec la sémantique asynchrone. Ceci permet au système de préférences généralisées pour mesurer au millier de Noeuds.

```
RP/0/RP1/CPU0:CWDCRS#show processes gsp
                  Job Id: 171
                     PID: 65604
          Executable path: /disk0/hfr-base-3.2.3/bin/gsp
               Instance #: 1
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
  Max. spawns per minute: 12
             Last started: Tue Jul 18 13:07:53 2006
            Process state: Run
            Package state: Normal
                     core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                    Level: 80
                Mandatory: ON
             startup_path: /pkg/startup/gsp-rp.startup
                    Ready: 5.259s
```

```
               Available: 16.613s
        Process cpu time: 988.265 user, 0.792 kernel, 989.057 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
171    1     152K  30 Receive       0:00:51:0815 gsp
171    3     152K  10 Condvar       0:00:00:0025 gsp
171    4     152K  10 Receive       0:00:08:0594 gsp
171    5     152K  10 Condvar       0:01:33:0274 gsp
171    6     152K  10 Condvar       0:00:55:0051 gsp
171    7     152K  10 Receive       0:02:24:0894 gsp
171    8     152K  10 Receive       0:00:09:0561 gsp
171    9     152K  10 Condvar       0:02:33:0815 gsp
171    10    152K  10 Condvar       0:02:20:0794 gsp
171    11    152K  10 Condvar       0:02:27:0880 gsp
171    12    152K  30 Receive       0:00:46:0276 gsp
171    13    152K  30 Receive       0:00:45:0727 gsp
171    14    152K  30 Receive       0:00:49:0596 gsp
171    15    152K  30 Receive       0:00:38:0276 gsp
171    16    152K  10 Receive       0:00:02:0774 gsp
```

Téléchargeur de contenu en vrac BCDL

BCDL est sûrement des données multicast utilisées à de divers Noeuds tels que la RPS et les MSCs. Il utilise le système de préférences généralisées comme transport sous-jacent. Garanties BCDL **dans la** livraison de **commande des** messages. Dans BCDL il y a un agent, un producteur et un consommateur. L'agent est le processus qui communique avec le producteur afin de récupérer et mettre en mémoire tampon les données avant ses Multidiffusions aux consommateurs. Le producteur est le processus qui produit les données que chacun veut, et le consommateur est le processus intéressé recevoir les données fournies par le producteur. BCDL est utilisé pendant les mises à jour de Logiciel Cisco IOS XR.

Messagerie légère (LWM)

LWM est une forme Cisco-créée de la Messagerie qui a été conçue pour créer une couche d'abstraction entre les applications que le processus inter communiquent les uns avec les autres et Neutrino, avec le but comme indépendance du système d'exploitation et de la couche transport. Si Cisco désire changer le constructeur de SYSTÈME D'EXPLOITATION de QNX à quelqu'un d'autre, une couche d'abstraction entre les fonctions rudimentaires des aides de système d'exploitation sous-jacent retirent la dépendance sur le système d'exploitation et les aides dans la mise en communication sur un autre système d'exploitation. LWM fournit la livraison garantie synchrone de message, qui aime le message indigène de Neutrino passant, fait bloquer l'expéditeur jusqu'à ce que le récepteur réponde.

LWM fournit également la livraison asynchrone de message par l'intermédiaire de 40 impulsions de bit. Des messages asynchrones sont asynchrone envoyés, qui signifie que le message est aligné et l'expéditeur ne bloque pas, mais n'est pas reçu par le serveur asynchrone, mais quand le serveur vote pour le prochain message disponible. LWM est structuré comme client/serveur. Le serveur crée un canal qui lui donne une **oreille** pour écouter dedans des messages et se repose dans un moment où la boucle fait un message reçoivent l'écoute sur le canal, qu'il a juste créé. Quand un message arrive il débloque et obtient un identificateur client, qui est efficacement la même chose que l'ID de réception du message a reçue. Le serveur en exécute alors qui traite et fait plus tard une réponse de message à l'identificateur client.

Sur le côté client il fait un message se connectent. Il obtient passé un identifiant à qui il connecte et puis fait un message envoient et est bloqué. Quand le serveur finit le traitement, il répond et le client devient débloqué. C'est pratiquement identique que le message indigène de Neutrinos passant, ainsi la couche d'abstraction est très légèrement.

LWM est conçu avec un nombre minimal d'appels système et de Commutateurs de contexte pour des hautes performances, et est la méthode préférée de l'IPC dans l'environnement de Cisco IOS XR.

Envmon

Tout au plus le niveau fondamental, le système de superviseur d'environnement est responsable de l'avertissement quand des paramètres physiques, par exemple la température, tension, vitesse des ventilateurs et ainsi de suite, chute en dehors des plages opérationnelles, et d'arrêter le matériel qui approche les niveaux essentiels où le matériel pourrait être endommagé. Il périodiquement surveille chaque capteur disponible de matériel, compare la valeur mesurée contre des seuils de carte-particularité, et donne des alarmes selon les besoins afin d'accomplir cette tâche. Un processus persistant, commencé à l'initialisation de système, qui vote périodiquement tous les capteurs de matériel, par exemple tension, la température, et vitesse des ventilateurs, dans le châssis et fournit ces données aux clients externes de Gestion. En outre, le processus périodique compare des lectures de capteur aux seuils d'alarme et édite des alertes environnementales à la base de données du système pour l'action ultérieure par le gestionnaire de défaut. Si les lectures de capteur sont dangereusement hors de plage, le procédé de surveillance de l'environnement pourrait entraîner la carte être arrêt.

Introduction de la matrice CRS-1

- Matrice à plusieurs étages — topologie de Benes de 3 étapes
- Routage dynamique dans la matrice pour réduire l'encombrement
- Cellulaire : 136 cellules d'octet, charge utile de 120 données d'octet
- Contrôle de flux pour améliorer la localisation du trafic et pour réduire des conditions requises de mise en mémoire tampon dans la matrice
- Étape pour présenter la livraison de Speedup
- Deux fontes du trafic prises en charge (Unicast et Multidiffusion)
- Deux priorités du trafic prises en charge par fonte (ciel et terre)
- Soutien des groupes de multidiffusion de matrice de 1M (FGIDs)
- Tolérance aux pannes rentable : La Redondance N+1 ou N+k utilisant la matrice surface par opposition à 1+1 au coût considérablement accru

Quand vous vous exécutez en mode de châssis unique, l'asics S1, S2 et S3 se trouvent sur les mêmes cartes de matrice. Cette carte est également généralement mentionnée comme la **carte S123**. Dans une configuration de Multi-châssis, le S2 est séparé et il est sur le châssis de carte de matrice (FCC). Cette configuration exige de deux cartes de matrice de former un avion, une carte S2, et une carte S13. Chaque MSC se connecte à huit avions de matrice afin de fournir la Redondance de sorte que si vous desserrez un ou plusieurs avions, vos passages de matrice trafiquent toujours bien que l'ensemble du trafic, qui peut passer par la matrice, soit inférieur. Les CRS peuvent encore fonctionner au linerate pour la plupart des longueurs de paquet avec seulement sept avions. La contre-pression est envoyée au-dessus de la matrice au-dessus d'un impair et même plat. Il n'est pas recommandé pour exploiter un système à moins de deux avions, dans un impair et même plat. Quelque chose moins de deux avions n'est pas une configuration prise en charge.

L'avion de matrice

Le diagramme précédent représente un avion. Vous devez multiplier ce diagramme par huit. Cela signifie que le pulvérisateur (ingressq) asic d'un LC se connecte à 8 S1s (1 S1 par avion). Le S1

dans chaque avion de matrice se connecte à 8 pulvérisateurs :

- les 8 LCS supérieurs du châssis
- les 8 LCS inférieurs

Il y a 16 S1s par 16 châssis de l'emplacement LC : 8 pour le LCS de dessus (1 par avion) + 8 pour le LCS de bas.

Sur des 16 châssis à emplacements simple, une carte de la matrice S123 a 2 S1s, 2 S2 et 4 S3s. Ce fait partie du calcul de speedup de matrice. Il y a deux fois autant le trafic, qui peut quitter la matrice que le trafic peut entrer. Il y a également actuellement deux éponges (fabricq) par LC comparé à 1 pulvérisateur. Ceci tient compte de bufferiser sur le de sortie LC quand plus d'une surcharge LCS d'entrée un de sortie LC. Le de sortie LC peut absorber cette bande passante supplémentaire de la matrice.

Surveillance de matrice

Disponibilité et Connectivité plates :

```
RP/0/RP1/CPU0:CWDCRS#show processes gsp
                  Job Id: 171
                     PID: 65604
          Executable path: /disk0/hfr-base-3.2.3/bin/gsp
               Instance #: 1
               Version ID: 00.00.0000
                  Respawn: ON
            Respawn count: 1
  Max. spawns per minute: 12
             Last started: Tue Jul 18 13:07:53 2006
            Process state: Run
            Package state: Normal
                     core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                    Level: 80
                Mandatory: ON
             startup_path: /pkg/startup/gsp-rp.startup
                    Ready: 5.259s
                Available: 16.613s
         Process cpu time: 988.265 user, 0.792 kernel, 989.057 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
171    1     152K  30 Receive       0:00:51:0815 gsp
171    3     152K  10 Condvar       0:00:00:0025 gsp
171    4     152K  10 Receive       0:00:08:0594 gsp
171    5     152K  10 Condvar       0:01:33:0274 gsp
171    6     152K  10 Condvar       0:00:55:0051 gsp
171    7     152K  10 Receive       0:02:24:0894 gsp
171    8     152K  10 Receive       0:00:09:0561 gsp
171    9     152K  10 Condvar       0:02:33:0815 gsp
171    10    152K  10 Condvar       0:02:20:0794 gsp
171    11    152K  10 Condvar       0:02:27:0880 gsp
171    12    152K  30 Receive       0:00:46:0276 gsp
171    13    152K  30 Receive       0:00:45:0727 gsp
171    14    152K  30 Receive       0:00:49:0596 gsp
171    15    152K  30 Receive       0:00:38:0276 gsp
171    16    152K  10 Receive       0:00:02:0774 gsp
```

Vérifiez si avions cellules reçoivent/de transmissions et quelques erreurs incrémentent :

```
RP/0/RP1/CPU0:CWDCRS#show processes gsp
                  Job Id: 171
                     PID: 65604
         Executable path: /disk0/hfr-base-3.2.3/bin/gsp
              Instance #: 1
              Version ID: 00.00.0000
                 Respawn: ON
           Respawn count: 1
  Max. spawns per minute: 12
            Last started: Tue Jul 18 13:07:53 2006
           Process state: Run
           Package state: Normal
                    core: TEXT SHAREDMEM MAINMEM
                Max. core: 0
                   Level: 80
                Mandatory: ON
             startup_path: /pkg/startup/gsp-rp.startup
                   Ready: 5.259s
               Available: 16.613s
        Process cpu time: 988.265 user, 0.792 kernel, 989.057 total
JID    TID  Stack pri state        HR:MM:SS:MSEC NAME
171    1     152K  30 Receive       0:00:51:0815 gsp
171    3     152K  10 Condvar       0:00:00:0025 gsp
171    4     152K  10 Receive       0:00:08:0594 gsp
171    5     152K  10 Condvar       0:01:33:0274 gsp
171    6     152K  10 Condvar       0:00:55:0051 gsp
171    7     152K  10 Receive       0:02:24:0894 gsp
171    8     152K  10 Receive       0:00:09:0561 gsp
171    9     152K  10 Condvar       0:02:33:0815 gsp
171    10    152K  10 Condvar       0:02:20:0794 gsp
171    11    152K  10 Condvar       0:02:27:0880 gsp
171    12    152K  30 Receive       0:00:46:0276 gsp
171    13    152K  30 Receive       0:00:45:0727 gsp
171    14    152K  30 Receive       0:00:49:0596 gsp
171    15    152K  30 Receive       0:00:38:0276 gsp
171    16    152K  10 Receive       0:00:02:0774 gsp
```

Les acronymes dans la commande précédente :

- CE — Erreur corrigible
- Écu — Erreur non corrigible
- PE — Erreur de parité

Ne vous inquiétez pas s'ils notent quelques erreurs, car ceci peut se produire sur le démarrage. Les champs ne devraient pas incrémenter le lors de l'exécution. S'ils sont, ce peut être une indication d'un problème dans la matrice. Émettez cette commande afin d'obtenir une répartition des erreurs par avion de matrice :

```
admin show controllers fabric plane <0-7> statistics detail
```

Contrôlez l'aperçu plat

La Connectivité d'avion de contrôle entre le châssis de linecard et le châssis de matrice est actuellement par l'intermédiaire des ports Gigabit Ethernet sur la RPS (LCC) et SCGE (FCC). L'interconnexion entre les ports sont fournies par l'intermédiaire d'une paire de Commutateurs de Catalyst 6500, qui peuvent être connectés par l'intermédiaire de deux ports Gigabit Ethernet ou plus.

Configuration de Catalyst 6500

C'est configuration recommandée pour des Commutateurs de Catalyst utilisés pour l'avion de contrôle de multi-châssis :

- Un VLAN simple est utilisé sur tous les ports.
- Tous les ports exécutés dans le mode d'accès (aucune jonction).
- Le spanning-tree 802.1w/s est utilisé pour la prévention de boucle.
- Deux liens ou plus sont utilisés afin de croix-connecter les deux Commutateurs et STP est utilisé pour boucle-empêche. La Manche n'est pas recommandée.
- Ports qui se connectent mode pré-standard CRS-1 RP et SCGE à utilisation puisqu'IOS-XR ne prend en charge pas le 802.1s basé par normes.
- UDLD devrait être activé sur les ports qui se connectent entre les Commutateurs et entre les Commutateurs et le RP/SCGE.
- UDLD est activé par défaut sur le CRS-1.

Référez-vous à apporter le Logiciel Cisco IOS XR sur un système de Multishelf pour plus d'informations sur la façon configurer un Catalyst 6500 dans un système de Multishelf.

Gestion d'avion de contrôle de Multi-châssis

Le châssis du Catalyst 6504-E, qui fournit la Connectivité d'avion de contrôle pour le système de multi-châssis, est configuré pour ces services de supervision :

- Administration intrabande par l'intermédiaire du gigabit 1/2 de port, qui se connecte à un commutateur de RÉSEAU LOCAL à chaque bruit. On permet seulement Access pour un petit choix de sous-réseaux et de protocoles.
- Le NTP est utilisé afin de placer l'heure système.
- Syslogging est exécuté aux hôtes standard.
- L'interrogation et les déroutements SNMP peuvent être activés pour des fonctions essentielles.

Note: Aucune modification ne devrait être apportée au Catalyst en fonction. Le test antérieur devrait être fait sur n'importe quelle modification prévue et on le recommande fortement que ceci soit fait pendant une fenêtre de maintenance.

C'est un échantillon de configuration de gestion :

```
#In-band management connectivity
interface GigabitEthernet2/1
 description *CRS Multi-chassis Management Ethernet - DO NOT TOUCH*
 ip address [ip address] [netmask]
 ip access-group control_only in
!
!
ip access-list extended control_only
 permit udp [ip address] [netmask] any eq snmp
 permit udp [ip address] [netmask] eq ntp any
 permit tcp [ip address] [netmask] any eq telnet

#NTP

ntp update-calendar
ntp server [ip address]
```

```
#Syslog
logging source-interface Loopback0
logging [ip address]
logging buffered 4096000 debugging
no logging console

#RADIUS
aaa new-model
aaa authentication login default radius enable
enable password {password}
radius-server host [ip address] auth-port 1645 acct-port 1646
radius-server key {key}

#Telnet and console access
!
access-list 3 permit [ip address]
!
line con 0
 exec-timeout 30 0
 password {password}
line vty 0 4
 access-class 3 in
 exec-timeout 0 0
password {password}
```

ROMMON et Monlib

Le monlib de Cisco est un programme exécutable qui est enregistré sur le périphérique et chargé dans la RAM pour l'exécution par ROMMON. ROMMON emploie le monlib afin d'accéder à des fichiers sur le périphérique. Des versions de ROMmon peuvent être mises à jour et devraient être faites ainsi sous la recommandation du support technique de Cisco. La plus défunte version de ROMmon est 1.40.

Instructions de mise à jour

Procédez comme suit :

1. Téléchargez les binaires ROMMON de Cisco CRS-1 ROMMON (clients enregistrés seulement).
2. Éclatez le fichier tar et copiez les 6 fichiers de COFFRE dans le répertoire racine CRS de Disk0.

```
#In-band management connectivity
interface GigabitEthernet2/1
 description *CRS Multi-chassis Management Ethernet - DO NOT TOUCH*
 ip address [ip address] [netmask]
 ip access-group control_only in
!
!
ip access-list extended control_only
 permit udp [ip address] [netmask] any eq snmp
 permit udp [ip address] [netmask] eq ntp any
 permit tcp [ip address] [netmask] any eq telnet

#NTP

ntp update-calendar
ntp server [ip address]
```

```
#Syslog
logging source-interface Loopback0
logging [ip address]
logging buffered 4096000 debugging
no logging console

#RADIUS
aaa new-model
aaa authentication login default radius enable
enable password {password}
radius-server host [ip address] auth-port 1645 acct-port 1646
radius-server key {key}

#Telnet and console access
!
access-list 3 permit [ip address]
!
line con 0
 exec-timeout 30 0
 password {password}
line vty 0 4
 access-class 3 in
 exec-timeout 0 0
password {password}
```

3. Utilisez le **show diag | ROM inc. |NOEUD|**Commande **PLIM** afin de voir la version de ROMmon en cours.

```
RP/0/RP0/CPU0:ROUTER(admin)#show diag | inc ROM|NODE|PLIM
NODE 0/0/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/0/CPU0 : 4OC192-POS/DPT
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/2/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/2/CPU0 : 8-10GbE
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/4/SP : Unknown Card Type
NODE 0/6/SP : MSC(SP)
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
PLIM 0/6/CPU0 : 16OC48-POS/DPT
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/RP0/CPU0 : RP
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/RP1/CPU0 : RP
  ROMMON: Version 1.19b(20050216:033559) [CRS-1 ROMMON]
NODE 0/SM0/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM1/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM2/SP : FC/S
  ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
NODE 0/SM3/SP : FC/S
ROMMON: Version 1.19b(20050216:033352) [CRS-1 ROMMON]
```

4. Entrez dans le mode d'ADMIN et employez l'**upgrade rommon une toute la** commande **disk0** afin d'améliorer le ROMMON.

```
RP/0/RP0/CPU0:ROUTER#admin
RP/0/RP0/CPU0:ROUTER(admin)#upgrade rommon a all disk0
Please do not power cycle, reload the router or reset any nodes until
 all upgrades are completed.
Please check the syslog to make sure that all nodes are upgraded successfully.
If you need to perform multiple upgrades, please wait for current upgrade
 to be completed before proceeding to another upgrade.
Failure to do so may render the cards under upgrade to be unusable.
```

5. Annulez le mode d'ADMIN et écrivez le **show log | l'inc. « CORRECT, les ROMMON A »** et s'assurent tous les Noeuds avec succès mis à jour. Si les Noeuds l'uns des échouent, passez de retour à l'étape 4 et reprogrammez.

```
RP/0/RP0/CPU0:ROUTER#show logging | inc "OK, ROMMON A"
RP/0/RP0/CPU0:Oct 28 14:40:57.223 PST8: upgrade_daemon[380][360]: OK, ROMMON A is
programmed successfully. SP/0/0/SP:Oct 28 14:40:58.249 PST8: upgrade_daemon[125][121]: OK,
ROMMON A is programmed successfully. SP/0/2/SP:Oct 28 14:40:58.251 PST8:
upgrade_daemon[125][121]: OK, ROMMON A is programmed successfully. LC/0/6/CPU0:Oct 28
14:40:58.336 PST8: upgrade_daemon[244][233]: OK, ROMMON A is programmed successfully.
LC/0/2/CPU0:Oct 28 14:40:58.365 PST8: upgrade_daemon[244][233]: OK, ROMMON A is programmed
successfully. SP/0/SM0/SP:Oct 28 14:40:58.439 PST8: upgrade_daemon[125][121]: OK, ROMMON A
is programmed successfully. SP/0/SM1/SP:Oct 28 14:40:58.524 PST8: upgrade_daemon[125][121]:
OK, ROMMON A is programmed successfully. LC/0/0/CPU0:Oct 28 14:40:58.530 PST8:
upgrade_daemon[244][233]: OK, ROMMON A is programmed successfully. RP/0/RP1/CPU0:Oct 28
14:40:58.593 PST8: upgrade_daemon[380][360]: OK, ROMMON A is programmed successfully.
SP/0/6/SP:Oct 28 14:40:58.822 PST8: upgrade_daemon[125][121]: OK, ROMMON A is programmed
successfully. SP/0/SM2/SP:Oct 28 14:40:58.890 PST8: upgrade_daemon[125][121]: OK, ROMMON A
is programmed successfully. SP/0/SM3/SP:Oct 28 14:40:59.519 PST8: upgrade_daemon[125][121]:
OK, ROMMON A is programmed successfully.
```

6. Entrez dans le mode d'ADMIN et employez l'**upgrade rommon b toute la** commande **disk0** afin d'améliorer le ROMMON.

```
RP/0/RP0/CPU0:ROUTER#admin
RP/0/RP0/CPU0:ROUTER(admin)#upgrade rommon b all disk0
Please do not power cycle, reload the router or reset any nodes until
 all upgrades are completed.
Please check the syslog to make sure that all nodes are upgraded successfully.
If you need to perform multiple upgrades, please wait for current upgrade
 to be completed before proceeding to another upgrade.
Failure to do so may render the cards under upgrade to be unusable.
```

7. Annulez le mode d'ADMIN et écrivez le **show log | l'inc. « CORRECT, les ROMMON B »** et s'assurent tous les Noeuds avec succès mis à jour. Si les Noeuds l'uns des échouent, passez de retour à l'étape 4 et reprogrammez.

```
RP/0/RP0/CPU0:Router#show logging | inc "OK, ROMMON B"
RP/0/RP0/CPU0:Oct 28 13:27:00.783 PST8: upgrade_daemon[380][360]: OK,
ROMMON B is programmed successfully.
LC/0/6/CPU0:Oct 28 13:27:01.720 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/2/SP:Oct 28 13:27:01.755 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
LC/0/2/CPU0:Oct 28 13:27:01.775 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/0/SP:Oct 28 13:27:01.792 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM0/SP:Oct 28 13:27:01.955 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
LC/0/0/CPU0:Oct 28 13:27:01.975 PST8: upgrade_daemon[244][233]: OK,
ROMMON B is programmed successfully.
SP/0/6/SP:Oct 28 13:27:01.989 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM1/SP:Oct 28 13:27:02.087 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
RP/0/RP1/CPU0:Oct 28 13:27:02.106 PST8: upgrade_daemon[380][360]: OK,
ROMMON B is programmed successfully.
SP/0/SM3/SP:Oct 28 13:27:02.695 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
SP/0/SM2/SP:Oct 28 13:27:02.821 PST8: upgrade_daemon[125][121]: OK,
ROMMON B is programmed successfully.
```

8. La commande de **mise à jour** grave juste une section de bootflash réservée par offre spéciale avec le nouveau ROMMON. Mais le nouveau ROMMON reste inactif jusqu'à ce que la carte soit rechargée. Ainsi quand vous la recharge la carte, le nouveau ROMMON est en

activité. Remettez à l'état initial chaque noeud un par un ou remettez à l'état initial juste le routeur entier afin de faire ceci.

```
Reload Router:
RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or 0/RP1/CPU0 reload (depends on which on is
in Standby Mode.
RP/0/RP0/CPU0:ROUTER#reload
!--- Issue right after the first command. Updating Commit Database. Please wait...[OK]
Proceed with reload? [confirm] !--- Reload each Node. For Fan Controllers (FCx), !--- Alarm
Modules (AMx), Fabric Cards (SMx), and RPs (RPx), !--- you must wait until the reloaded
node is fully reloaded !--- before you reset the next node of the pair. But non-pairs !---
can be reloaded without waiting. RP/0/RP0/CPU0:ROUTER#hw-module node 0/RP0/CPU0 or
0/RP1/CPU0 reload
!--- This depends on which on is in Standby Mode. RP/0/RP0/CPU0:ROUTER#hw-module node
0/FC0/SP
RP/0/RP0/CPU0:ROUTER#hw-module node 0/AM0/SP
RP/0/RP0/CPU0:ROUTER#hw-module node 0/SM0/SP
!--- Do not reset the MSC and Fabric Cards at the same time. RP/0/RP0/CPU0:ROUTER#hw-module
node 0/0/CPU
```

9. Utilisez le **show diag | ROM inc. |NOEUD|**Commande **PLIM** afin de vérifier la version de ROMmon en cours.

```
RP/0/RP1/CPU0:CRS-B(admin)#show diag | inc ROM|NODE|PLIM
NODE 0/0/SP : MSC(SP)
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
PLIM 0/0/CPU0 : 4OC192-POS/DPT
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/2/SP : MSC(SP)
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
PLIM 0/2/CPU0 : 8-10GbE
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/6/SP : MSC(SP)
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
PLIM 0/6/CPU0 : 16OC48-POS/DPT
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/RP0/CPU0 : RP
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]
NODE 0/RP1/CPU0 : RP
  ROMMON: Version 1.32(20050525:193559)  [CRS-1 ROMMON]]
NODE 0/SM0/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
NODE 0/SM1/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
NODE 0/SM2/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
NODE 0/SM3/SP : FC/S
  ROMMON: Version 1.32(20050525:193402)  [CRS-1 ROMMON]
```

Note: Sur CRS-8 et châssis de matrice, ROMMON fixe également les vitesses des ventilateurs à la vitesse par défaut de 4000 t/mn.

Aperçu PLIM et de MSC

Ceci représente l'écoulement de paquet sur le routeur CRS-1, et ces termes sont utilisés l'un pour l'autre :

IngressQ ASIC s'appelle également le pulvérisateur ASIC.

FabricQ ASIC s'appelle également l'éponge ASIC.

EgressQ ASIC s'appelle également le Sharq ASIC.

Des espèces s'appelle également le PSE (engine de commutateur de paquets) ASIC.

Rx PLIM > Rx espèces > d'entrée Q > matrice > matrice Q > Tx espèces > de sortie Q > Tx PLIM (pulvérisateur) (éponge) (Sharq)

Des paquets sont reçus sur le module d'interface de couche physique (PLIM).

Le PLIM contient les interfaces physiques pour le MSC auquel il accouple. Les PLIM et le MSC sont les cartes distinctes connectées par l'intermédiaire du fond de panier du châssis. En conséquence les types d'interface pour un MSC particulier sont définis par le type du PLIM au lequel il a accouplé. La personne à charge sur le type de PLIM, la carte contient un divers nombre d'ASIC qui fournissent les medias et le tramage physiques pour les interfaces. Le but du PLIM ASIC sont de fournir l'interface entre le MSC et les connexions physiques. Il termine la fibre, fait la lumière à la conversion électrique, termine le tramage de medias étant SDH/Sonet/Ethernet/HDLC/PPP, vérifie le CRC, ajoute une certaine information de contrôle appelée l'en-tête de mémoire tampon et en avant les bits qui restent sur le MSC. Le PLIM ne fait pas source/évier le Keepalives HDLC ou de PPP. Ceux-ci sont manipulés par la CPU sur le MSC.

Le PLIM fournit également ces fonctions :

- Filtrage MAC pour 1/10 Gigabit Ethernet
- MAC d'entrée/de sortie expliquant 1/10 Gigabit Ethernet
- VLAN filtrant pour 1/10 Gigabit Ethernet
- VLAN expliquant 1/10 Gigabit Ethernet
- Mise en mémoire tampon et notification d'encombrement d'entrée

Surabonnement PLIM

10GE PLIM

Les 8 X 10G PLIM offre la capacité pour terminer approximativement 80 GBP du trafic tandis que la capacité de transmission du MSC est un maximum de 40 GBP. Si tous les ports disponibles sur le PLIM sont remplis, alors le surabonnement a lieu et la modélisation de QoS devient extrêmement importante pour s'assurer que le trafic de la meilleure qualité n'est pas par distraction abandonné. Pour certains, le surabonnement n'est pas une option et doit être évité. Seulement quatre des huit ports doivent être utilisés afin de faire ceci. En outre, le soin doit être pris pour s'assurer que la bande passante optimale dans le MSC et le PLIM est disponible à chacun des quatre ports.

Note: Le mappage de ports change de la version 3.2.2 en avant. Voir les ces diagrammes.

Mappage de ports jusqu'à la version 3.2.1 Mappage de ports de la version 3.2.2 en avant

Comme précédemment mentionné, les ports physiques sont entretenus par l'un des deux FabricQ ASIC. L'attribution des ports à l'ASIC est statiquement définie et ne peut pas être modifiée. En outre, les 8 X 10G PLIM a deux PLA ASIC. Le premier PLA entretient les ports 0 3, les deuxièmes services 4 7. La capacité de bande passante d'un PLA simple sur les 8 X 10G PLIM est approximativement 24 GBP. La capacité de commutation d'un FabricQ simple ASIC est approximativement 62 Mpps.

Si vous remplissez port 0 3 ou ports 4 7, la capacité de bande passante du PLA (24 GBP) sont partagées entre chacun des quatre des ports qui limitent le débit global. Si vous remplissez ports 0,2,4 et 6 (jusqu'à 3.2.1) ou 0,1,4 et 5 (3.2.2 en avant) comme tous ces ports êtes entretenu par l'un FabricQ ASIC, dont la capacité de commutation est 62 Mpps, de nouveau, qui limite la capacité de débit.

Il est le meilleur d'utiliser les ports en quelque sorte qui obtient le rendement le plus élevé des PLAs et du FabricQ ASIC afin de réaliser la performance optimale.

SIP-800/SPA

Le SIP-800 PLIM offre la capacité de fonctionner avec les cartes d'interface modulaires connues sous le nom d'adaptateurs de port de service (stations thermales). Le SIP-800 fournit à 6 baies de STATION THERMALE une capacité théorique d'interface de 60 GBP. La capacité de transmission du MSC est un maximum de 40 GBP. Si toutes les baies sur le SIP-800 devaient être remplies, alors, dépendant sur le type de STATION THERMALE, il est possible que le surabonnement ait lieu et modélisation de QoS devient extrêmement important afin de s'assurer que le trafic de la meilleure qualité n'est pas par distraction abandonné.

Note: Le surabonnement n'est pas pris en charge avec des interfaces de POS. Mais, le placement de la STATION THERMALE de POS 10 gigaoctets doit être approprié afin de s'assurer que la capacité de débit correcte est fournie. La STATION THERMALE de 10 Ethernets gigaoctet est seulement prise en charge dans la version 3.4 IOS-XR. Cette STATION THERMALE offre des capacités de surabonnement.

Pour certains, le surabonnement n'est pas une option et doit être évité. Seulement quatre des sixbays doivent être utilisés afin de faire ceci. En outre, le soin doit être commande rentrée pour s'assurer que la bande passante optimale dans le MSC et le PLIM est disponible à chacun des quatre ports.

Mappage de baie de STATION THERMALE

Comme mentionné dedans précédemment, les ports physiques sont entretenus par l'un des deux FabricQ ASIC. L'attribution des ports à l'ASIC est statiquement définie et ne peut pas être modifiée. En outre, le SIP-800 PLIM a deux PLA ASIC. Le premier PLA entretient les ports 0,1 et 3, les deuxièmes services 2, 4 et 5.

La capacité de bande passante d'un PLA simple sur le SIP-800 PLIM est approximativement 24 GBP. La capacité de commutation d'un FabricQ simple ASIC est approximativement 62 Mpps.

Si vous remplissez port 0,1 et 3 ou ports 2, 4 et 5, la capacité de bande passante du PLA (24 GBP) sont partagées entre chacun des trois des ports qui limitent le débit global. Puisqu'un FabricQ simple chaque services ceux mettent en communication des groupes, le débit de paquets maximum du groupe de port a 62 ans Mpps. Il est le meilleur d'utiliser les ports en quelque sorte qui obtient le rendement le plus élevé des PLAs afin de réaliser la bande passante optimale.

Placement suggéré :

	Bay# de STATION THERMALE	Bay# de STATION THERMALE	Bay# de STATION THERMALE	Bay# de STATIO N

				THERMALE
Option 1	0	1	4	5
Option 2	1	2	3	4

Si vous voulez remplir carte avec plus la STATION THERMALE de quatre, la recommandation est de se terminer une des options précédemment répertoriées, qui se propagent les interfaces entre les groupes à deux orifices (0,1 et 3 et 2,4 et 5). Vous devriez alors placer les prochains modules de STATION THERMALE dans un des ports ouverts dans l'un ou l'autre les 0,1 et 3 et 2,4 et 5 groupes de port.

DWDM XENPACKs

De la version 3.2.2 en avant, DWDM XENPACKs peut être installé et fournir les modules **réglables d'**optique. Les conditions requises de refroidissement de tels modules XENPACK exige qu'il y ait un emplacement vide entre les modules installés. En outre, si un module simple DWDM XENPACK est installé, un maximum de quatre ports peut être utilisé, même si les les modules XENPACK ne sont pas des périphériques DWDM. Ceci a donc une incidence directe sur le FabricQ au PLA au mappage de ports. L'attention doit être prêtée à cette condition requise et est considérée dans cette table.

Placement suggéré :

	Port d'optique #	**Port d'optique #**	**Port d'optique #**	**Port d'optique #**
Option 1 ou DWDM XENPACK	0	2	5	7
Option 2	1	3	4	6

Pour des 3.2.2 ou plus tard ou 3.3 installation, évitez la modification de mappage de FabricQ. Un modèle plus simple de placement peut donc être utilisé pour le militaire de carrière et les modules DWDM XENPACK.

	Port d'optique #	**Port d'optique #**	**Port d'optique #**	**Port d'optique #**
Option 1	0	2	4	6
Option 2	1	3	5	7

Si vous voulez remplir carte avec plus de quatre ports non-DWDM XENPACK, la recommandation est de se terminer une des options répertoriées, qui répand les modules d'interface Optiques entre les groupes à deux orifices (0-3 et 4-7). Vous devez placer alors les prochains modules d'interface Optiques dans un des ports ouverts dans les 0-3 ou 4-7 groupes de port. Si vous utilisez le groupe du port 0-3 pour le module d'interface Optique #5, les modules d'interface Optiques #6 devraient être placés dans le groupe du port 4-7.

Référez-vous aux modules DWDM XENPAK pour plus de détails.

Gestion de la configuration

La configuration dans IOS-XR est faite par une configuration à deux étages, la configuration est entrée par l'utilisateur dans la première phase. C'est l'étape où seulement la syntaxe de configuration est vérifiée par le CLI. La configuration écrite dans cette étape est seulement connue au processus d'agent de configuration, par exemple, CLI/XML. La configuration n'est pas vérifiée puisqu'on ne lui écrit pas au serveur de sysdb. L'application principale ne sont pas annoncées et ne peuvent accéder à ou avoir aucune connaissance de la configuration dans cette étape.

Dans la seconde étape, la configuration est explicitement commise par l'utilisateur. Dans cette étape la configuration est écrite au serveur de sysdb, les applications principales vérifient les configurations et des notifications sont générées par le sysdb. Vous pouvez abandonner une session de configuration avant que vous commettiez la configuration écrite dans la première phase. Ainsi, il n'est pas sûr de supposer que toute la configuration écrite dans le stage premier est toujours commise dans l'étape deux.

En outre, le fonctionnement et/ou la configuration en cours du routeur peuvent être modifiés par des plusieurs utilisateurs pendant le stage premier et l'étape deux. Ainsi, aucun test du routeur qui exécute la configuration et/ou l'état opérationnel dans le stage premier ne pourrait être valide dans l'étape deux où la configuration est commise réellement.

Systèmes de fichier de configuration

Le système de fichier de configuration (CFS) sont un ensemble de fichiers et des répertoires utilisés afin d'enregistrer la configuration du routeur. Le CFS est enregistré sous le répertoire disk0:/config/, qui est le support par défaut utilisé sur le RP. Les fichiers et les répertoires dans le CFS sont internes au routeur et devraient ne jamais être modifiés ou retirés par l'utilisateur. Ceci peut avoir comme conséquence la perte ou la corruption de la configuration et affecte le service.

Le CFS est checkpointed au standby-RP après chaque validation. Ceci aide la conserve le fichier de configuration du routeur après un basculer.

Pendant le démarrage du routeur, la dernière configuration active est appliquée de la base de données de validation de configuration enregistrée dans le CFS. Il n'est pas que l'utilisateur sauvegarde manuellement la configuration active après chaque validation de configuration, puisque ceci est fait automatiquement par le routeur.

Il n'est pas recommandé d'apporter des modifications de configuration tandis que la configuration est appliquée pendant le démarrage. Si l'application de configuration n'est pas complète, vous voyez ce message quand vous ouvrez une session au routeur :

Processus de configuration système

La configuration de démarrage pour ce périphérique charge actuellement. Ceci peut prendre quelques minutes. On vous annonce sur la fin. S'il vous plaît ne tentez pas de modifier le périphérique jusqu'à ce que ce processus soit complet. Dans des quelques rares cas, il pourrait être desirable de restaurer la configuration de routeur à partir d'un fichier de configuration fourni par utilisateur ASCII au lieu de restaurer la dernière configuration active du CFS.

Vous pouvez forcer l'application d'un fichier de configuration par :

```
using the “-a” option with the boot command. This option forces
    the use of the specified file only for this boot.

    rommon>boot <image> –a <config-file-path>

    setting the value of “IOX_CONFIG_FILE” boot variable to the
    path of configuration file. This forces the use of the specified file
    for all boots while this variable is set.

    rommon>IOX_CONFIG_FILE=<config-file-path>
    rommon>boot <image>
```

Tandis que vous restaurez la configuration de routeur, un ou plusieurs éléments de configuration pourraient pour les prendre effet. Toute la configuration défectueuse est enregistrée dans le CFS et est mise à jour jusqu'à la prochaine recharge.

Vous pouvez parcourir la configuration défectueuse, adresser les erreurs et réappliquer la configuration.

Ce sont quelques conseils afin d'adresser la configuration défectueuse pendant le startup du routeur.

Dans IOX, la configuration peut être classifiée en tant que configuration défectueuse pour trois raisons :

1. Erreurs de syntaxe — Le programme d'analyse syntaxique génère les erreurs de syntaxe, qui indiquent habituellement qu'il y a une incompatibilité avec des commandes CLI. Vous devriez corriger les erreurs de syntaxe et réappliquer la configuration.
2. Erreurs sémantiques — Des erreurs sémantiques sont générées par les composants principaux quand le gestionnaire de configuration restaure la configuration pendant le startup du routeur. Il est important de noter que le cfgmgr n'est pas responsable d'assurer la configuration est reçu en tant qu'élément de la configuration en cours. Cfgmgr est simplement un **intermédiaire** et signale seulement toutes les pannes sémantiques que les composants principaux génèrent. Il incombe à chaque propriétaire composant principal pour analyser la raison de panne et pour déterminer la raison pour la panne. Les utilisateurs peuvent exécuter le **commands> de la description <CLI du** mode de configuration afin de trouver facilement le propriétaire du vérificateur composant principal. Par exemple, si le **BGP 217 de routeur** apparaît en tant que configuration défectueuse, la commande de **description** prouve que le vérificateur composant est le composant ipv4-bgp.

```
RP/0/0/CPU0:router#configure terminal
RP/0/0/CPU0:router(config)#describe router bgp 217
The command is defined in bgpv4_cmds.parser

Node 0/0/CPU0 has file bgpv4_cmds.parser for boot package /gsr-os-mbi-3.3.87/mbi12000-rp.vm
from gsr-rout
Package:
    gsr-rout
        gsr-rout V3.3.87[Default]  Routing Package
        Vendor : Cisco Systems
        Desc   : Routing Package
        Build  : Built on Mon Apr  3 16:17:28 UTC 2006
        Source : By ena-view3 in /vws/vpr/mletchwo/cfgmgr_33_bugfix for c2.95.3-p8
        Card(s): RP, DRP, DRPSC
        Restart information:
```

```
          Default:
            parallel impacted processes restart
Component:
    ipv4-bgp V[fwd-33/66]  IPv4 Border Gateway Protocol (BGP)

File: bgpv4_cmds.parser


User needs ALL of the following taskids:

        bgp (READ WRITE)

It will take the following actions:
  Create/Set the configuration item:
         Path: gl/ip-bgp/0xd9/gbl/edm/ord_a/running
        Value: 0x1
Enter the submode:
        bgp
RP/0/0/CPU0:router(config)#
```

3. Appliquez les erreurs — La configuration a été avec succès vérifiée et reçue en tant qu'élément de la configuration en cours mais le composant principal ne peut pas mettre à jour son état opérationnel pour quelque raison. La configuration affiche dans les les deux la configuration en cours, puisqu'elle a été correctement vérifiée, et en tant que configuration défectueuse en raison de l'erreur opérationnelle principale. La commande de **description** peut de nouveau être exécutée sur le CLI qui ne s'est pas appliqué afin de trouver le composant appliquent le propriétaire.Terminez-vous ces étapes afin de parcourir et réappliquer la configuration défectueuse pendant les opérateurs de démarrage :Pour R3.2 les opérateurs peuvent employer cette procédure afin de réappliquer la configuration défectueuse :Les opérateurs peuvent employer la commande de **show configuration failed startup** afin de parcourir la configuration défectueuse enregistrée pendant le démarrage du routeur.Les opérateurs devraient exécuter le **noerror de show configuration failed startup | classez la commande myfailed.cfg** afin de sauvegarder le startup a manqué configuration à un fichier.Les opérateurs devraient aller aux commandes de **mode de configuration** et de **chargement/validation d'**utilisation afin de réappliquer cette configuration défectueuse :

```
RP/0/0/CPU0:router(config)#load myfailed.cfg
Loading.
197 bytes parsed in 1 sec (191)bytes/sec
RP/0/0/CPU0:router(config)#commit
```

Pour les images R3.3 les opérateurs peuvent utiliser cette procédure mise à jour :Les opérateurs doivent employer la commande de **show configuration failed startup** et la commande de **startup de load configuration failed** afin de parcourir et réappliquer n'importe quelle configuration défectueuse.

```
RP/0/0/CPU0:router#show configuration failed startup
!! CONFIGURATION FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
telnet vrf default ipv4
server max-servers 5 interface POS0/7/0/3 router static
address-family ipv4 unicast
 0.0.0.0/0 172.18.189.1

!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
router bgp 217 !!%
Process did not respond to sysmgr !
RP/0/0/CPU0:router#

RP/0/0/CPU0:router(config)#load configuration failed startup noerror
Loading.
```

```
263 bytes parsed in 1 sec (259)bytes/sec
RP/0/0/CPU0:mike3(config-bgp)#show configuration
Building configuration...
telnet vrf default ipv4 server max-servers 5 router static
address-family ipv4 unicast
  0.0.0.0/0 172.18.189.1
 !
!
router bgp 217
!
end

RP/0/0/CPU0:router(config-bgp)#commit
```

Déchargeur de noyau

Par défaut IOS-XR écrit un vidage de mémoire au disque dur si un crash de processus, mais pas si le noyau lui-même tombe en panne. Notez que pour un système de multi-châssis cette fonctionnalité est actuellement seulement prise en charge pour le châssis 0 de linecard. L'autre châssis est pris en charge dans une version future de logiciel.

On lui suggère que des vidages mémoire de noyau pour la RPS et des MSCs soient activés avec l'utilisation des ces configuration dans les configurations de norme et d'admin-mode :

```
RP/0/0/CPU0:router#show configuration failed startup
!! CONFIGURATION FAILED DUE TO SYNTAX/AUTHORIZATION ERRORS
telnet vrf default ipv4
server max-servers 5 interface POS0/7/0/3 router static
address-family ipv4 unicast
 0.0.0.0/0 172.18.189.1

!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
router bgp 217 !!%
Process did not respond to sysmgr !
RP/0/0/CPU0:router#

RP/0/0/CPU0:router(config)#load configuration failed startup noerror
Loading.
263 bytes parsed in 1 sec (259)bytes/sec
RP/0/0/CPU0:mike3(config-bgp)#show configuration
Building configuration...
telnet vrf default ipv4 server max-servers 5 router static
address-family ipv4 unicast
  0.0.0.0/0 172.18.189.1
 !
!
router bgp 217
!
end

RP/0/0/CPU0:router(config-bgp)#commit
```

Configuration de vidage mémoire de noyau

Ceci a comme conséquence cette occurrence pour un crash de noyau :

1. Un RP tombe en panne et un vidage mémoire est écrit au disque dur sur ce RP dans le

répertoire racine du disque.
2. Si un MSC tombe en panne, un vidage mémoire est écrit au disque dur de RP0 dans le répertoire racine du disque.

Ceci n'a aucune incidence des temps de Basculement RP puisque l'expédition direct (NSF) est configuré pour les protocoles de routage. Il peut prendre quelques minutes supplémentaires pour que le RP ou le linecard tombé en panne devienne disponible de nouveau après qu'il suive un crash tandis qu'il écrit le noyau.

Un exemple de l'ajout de cette configuration à la norme et à la configuration de mode d'admin est affiché ici. Notez que la configuration de mode d'admin exige des DRP d'être utilisés.

Cette sortie affiche un exemple de configuration de vidage mémoire de noyau :

```
RP/0/RP0/CPU0:crs1#configure
RP/0/RP0/CPU0:crs1(config)#exception kernel memory kernel filepat$
RP/0/RP0/CPU0:crs1(config)#exception dump-tftp-route port 0 host-$
RP/0/RP0/CPU0:crs1(config)#commit
RP/0/RP0/CPU0:crs1(config)#
RP/0/RP0/CPU0:crs1#admin
RP/0/RP0/CPU0:crs1(admin)#configure
Session                   Line        User      Date                       Lock
00000201-000bb0db-00000000  snmp        hfr-owne  Wed Apr  5 10:14:44 2006
RP/0/RP0/CPU0:crs1(admin-config)#exception kernel memory kernel f$
RP/0/RP0/CPU0:crs1(admin-config)#exception dump-tftp-route port 0$
RP/0/RP0/CPU0:crs1(admin-config)#commit
RP/0/RP0/CPU0:crs1(admin-config)#
RP/0/RP0/CPU0:crs1(admin)#
```

Sécurité

LPTS

Les services de transport locaux de paquet (LPTS) manipulent les paquets localement destinés. LPTS est fait de divers différents composants.

1. Le principal s'appelle le processus d'arbitre de port. Il écoute des demandes de socket de différents processus du protocole de routage, par exemple, BGP, IS-IS et maintient toutes les informations obligatoires pour ces processus. Par exemple, si un processus BGP écoute au numéro de prise 179, la PA obtient ces informations des processus BGP, et puis assigne une attache à ce processus dans un IFIB.
2. L'IFIB, est un autre composant du processus LPTS. Il aide à garder un répertoire d'où un processus est qui écoute une attache spécifique de port. L'IFIB est généré par le processus d'arbitre de port et est gardé avec l'arbitre de port. Il génère alors de plusieurs sous-ensembles de ces informations.Le premier sous-ensemble est une part de l'IFIB. Cette part peut être associée au protocole d'ipv4 et ainsi de suite. Des parts sont alors envoyées pour s'approprier les gestionnaires d'écoulement, qui emploient alors la part IFIB afin d'expédier le paquet au processus approprié.Le deuxième sous-ensemble est un pre-IFIB, permet au LC pour expédier le paquet au processus approprié si seulement un processus existe ou à un gestionnaire approprié d'écoulement.
3. Les gestionnaires d'écoulement aident plus loin à distribuer les paquets si la consultation est non triviale, par exemple, des processus multiples pour le BGP. Chaque gestionnaire

d'écoulement a une part ou des parts de multiple de l'IFIB et correctement en avant des paquets aux processus appropriés associés avec la part de l'IFIB.
4. Si une entrée n'est pas définie pour la destination port puis elle peut être abandonnée ou expédiée au gestionnaire d'écoulement. Un paquet est expédié sans le port associé s'il y a une stratégie associée pour le port. Les aides de gestionnaire d'écoulement alors génèrent une nouvelle entrée de session.

Comment est-ce qu'un paquet interne est expédié ?

Il y a deux types d'écoulements, posent 2 (HDLC, PPP) écoulements et posent 4 écoulements ICMP/PING et le routage circule.

1. Couche 2 HDLC/PPP — Ces paquets sont identifiés par l'identificateur de protocole et sont envoyés directement aux files d'attente CPU dans le pulvérisateur. Les paquets de protocole de la couche 2 obtiennent la haute priorité et sont puis pris par la CPU (par l'intermédiaire du calmar) et traités. Par conséquent le Keepalives pour la couche 2 est directement répondu à par l'intermédiaire du LC par l'intermédiaire de la CPU. Ceci évite la nécessité d'aller au RP pour des réponses et lit dedans avec le thème de la Gestion d'interface distribuée.
2. ICMP (paquets de couche 4) sont reçus dans le LC et ils sont envoyés par l'intermédiaire de la consultation par l'IFBI dans les files d'attente CPU sur le pulvérisateur. Ces paquets sont alors envoyés à la CPU (par l'intermédiaire du calmar) et traités. La réponse est alors envoyée par les files d'attente de sortie de pulvérisateur afin de pour être expédiée par la matrice. C'est au cas où une autre application aurait besoin également d'informations (répliquées par la matrice). Une fois par la matrice le paquet est destiné au de sortie approprié LC et par la file d'attente appropriée d'éponge et de contrôle.
3. Conduisant des écoulements sont recherchés dans l'IFIB et alors envoyé à la sortie formant des files d'attente (8000 files d'attente) l'un d'entre eux est réservé pour des paquets de contrôle. C'est une file d'attente non formée et est simplement entretenue chaque fois qu'il est plein. – haute priorité. Le paquet est alors envoyé par la matrice sur des files d'attente prioritaire en jeu de files d'attente CPU sur l'éponge (semblable au calmar s'aligne sur le pulvérisateur), et puis des processus par le processus approprié, le gestionnaire d'écoulement ou le processus réel. Une réponse est renvoyée par l'éponge de linecard de sortie et puis le linecard. L'éponge du de sortie LC a une file d'attente spéciale mise de côté pour manipuler des paquets de contrôle. Les files d'attente dans l'éponge sont coupées en paquets de haute priorité, de contrôle et de faible priorité, par base de port de sortie.
4. Le PSE a un ensemble de régulateurs qui sont configurés pour la couche 4 de limitation de débit, la couche 2 et les paquets de routage. Ceux-ci sont pré-établis et changent pour être utilisateur configurable à une date ultérieure.

Un de la plupart de problème courant avec LPTS est des paquets qui sont lâchés, quand vous tentez de cingler le routeur. Les régulateurs LPTS sont habituellement limitation de débit ces paquets. C'est le cas afin de confirmer :

```
RP/0/RP0/CPU0:ss01-crs-1_P1#ping 192.168.3.14 size 8000 count 100
Type escape sequence to abort.
Sending 100, 8000-byte ICMP Echos to 192.168.3.14, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!!!!!!!!!!!!!!!!!!!.!!!!!!!!!!
!!!!!!!!!!!!!!!.!!!!!!!!!!!!
Success rate is 97 percent (97/100), round-trip min/avg/max = 1/2/5 ms
```

```
RP/0/RP0/CPU0:ss01-crs-1_P1#show lpts pifib hardware entry statistics location 0/5/CPU0 | excl
0/0

* - Vital; L4 - Layer4 Protocol; Intf - Interface;
DestAddr - Destination Fabric Address;
na - Not Applicable or Not Available

Local, Remote Address.Port          L4    Intf         DestAddr     Pkts/Drops
---------------------------------- ----- ------------ ------------ --- any
any   any          Punt       100/3
224.0.0.5 any                        any   PO0/5/1/0    0x3e         4/0
224.0.0.5 any                        any   PO0/5/1/1    0x3e         4/0
<further output elided>
```

IPsec

Les paquets IP sont en soi non sécurisés. IPsec est une méthode utilisée pour protéger les paquets IP. CRS-1 IPsec est mis en application dans le chemin de transfert de logiciel, donc la session d'IPsec est terminée sur le RP/DRP. Un nombre total de 500 sessions d'IPsec par CRS-1 sont pris en charge. Le nombre dépend de la vitesse du CPU et des ressources allouées. Il n'y a aucune limite de logiciel à ceci, seulement le trafic local-originaire et local-terminé sur le RP sont habilité à la manipulation d'IPsec. Le mode ou le tunnel mode de transport d'IPsec peut être utilisé pour le type de trafic, cependant l'ancien est dû préféré à moins de temps système dans le traitement d'IPsec.

R3.3.0 prend en charge le cryptage du BGP et de l'OSPFv3 au-dessus d'IPsec.

Référez-vous au guide de configuration de sécurité des systèmes de Cisco IOS XR pour plus d'informations sur la façon implémenter IPsec.

Note: IPsec a besoin du crypto secteur, par exemple, hfr-k9sec-p.pie-3.3.1.

Hors de la bande

Console et Access AUX.

Les CRS-1 RP/SCs ont une console et le port auxiliaire disponibles pour hors de la Gestion de bande, aussi bien qu'un port Ethernet de Gestion pour hors bande par l'intermédiaire de l'IP.

La console et le port auxiliaire de chaque RP/SCGE, deux par châssis, peuvent être connectés à un serveur de console. Ceci signifie que le système de châssis unique exige quatre ports de console, et les systèmes de multi-châssis exigent 12 ports plus encore deux ports pour les engines de superviseur sur le Catalyst 6504-E.

La connexion de port auxiliaire est importante puisqu'elle permet d'accéder au noyau IOS-XR et peut permettre la restauration du système quand ce n'est pas possible par l'intermédiaire du port de console. Access par l'intermédiaire du port auxiliaire est seulement à la disposition des utilisateurs localement définis sur le système, et seulement quand l'utilisateur a accès de racine-système ou de niveau de Cisco-support. En outre l'utilisateur doit faire définir un **mot de passe secret**.

Accès au terminal virtuel

Le telnet et le Protocole Secure Shell (SSH) peuvent être utilisés afin d'atteindre le CRS-1 par

l'intermédiaire des ports vty. Par défaut chacun des deux sont désactivés, et les besoins de l'utilisateur de les activer explicitement.

Note: IPsec a besoin du crypto secteur, par exemple, hfr-k9sec-p.pie-3.3.1.

Générez d'abord les clés RSA et DSA suivant les indications de cet exemple afin d'activer le SSH :

```
RP/0/RP1/CPU0:Crs-1#crypto key zeroize dsa
% Found no keys in configuration.
RP/0/RP1/CPU0:Crs-1#crypto key zeroize rsa
% Found no keys in configuration.

RP/0/RP1/CPU0:Crs-1#crypto key generate rsa general-keys
The name for the keys will be: the_default
  Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keypair.
  Choosing a key modulus greater than 512 may take a few minutes.

How many bits in the modulus [1024]:
Generating RSA keys ...
Done w/ crypto generate keypair
[OK]

RP/0/RP1/CPU0:Crs-1#crypto key generate dsa
The name for the keys will be: the_default
  Choose the size of your DSA key modulus. Modulus size can be 512, 768, or 1024 bits. Choosing
a key modulus
How many bits in the modulus [1024]:
Generating DSA keys ...
Done w/ crypto generate keypair
[OK]

!--- VTY access via SSH & telnet can be configured as shown here. vty-pool default 0 4 ssh
server ! line default secret cisco users group root-system users group cisco-support exec-
timeout 30 0 transport input telnet ssh ! ! telnet ipv4 server
```

Informations connexes

- Support de Routeurs
- Support et documentation techniques - Cisco Systems