

CRS-1和IOS XP可操作的最佳实践

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[Cisco IOS XR概述](#)

[进程和线索](#)

[进程和线索状态](#)

[同步消息通过](#)

[被阻塞的流程和进程状态](#)

[重要进程和他们的功能](#)

[Netio](#)

[组服务进程\(GSP\)](#)

[BCDL容量内容下载者](#)

[轻量级消息传送\(LWM\)](#)

[Envmon](#)

[CRS-1结构介绍](#)

[结构飞机](#)

[结构监听](#)

[控制层面概述](#)

[Catalyst 6500配置](#)

[多机箱控制层面管理](#)

[ROMMON和Monlib](#)

[升级指南](#)

[PLIM和MSC概述](#)

[PLIM超额预订](#)

[配置管理](#)

[安全](#)

[LPTS](#)

[一内部数据包如何转发？](#)

[带外](#)

[相关信息](#)

[简介](#)

本文帮助您了解这些：

- 进程和线索
- CRS-1结构
-
- Rommon和Monlib
- 物理层接口模块(PLIM)和模块化服务卡德(MSC)
- 配置管理
- 安全
- 带外
- 简单网络管理协议 (SNMP)

先决条件

要求

思科建议您有Cisco IOS XR知识。

使用的组件

本文档中的信息基于以下软件和硬件版本：

- Cisco IOS XR 软件
- CRS-1

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的详细信息，请参阅 [Cisco 技术提示规则](#)。

Cisco IOS XR概述

Cisco IOS XR设计扩展。内核是如此提供仅重要服务例如进程管理的微内核体系结构，日程安排、信号和计时器。所有其他服务例如文件系统、驱动程序、协议栈和应用程序考虑作为资源管理器和运行在内存受保护的用户空间。这些其他服务可以被添加或删除在运行时间，依靠目的而设计。微内核覆盖区只是12 kb。微内核和底层操作系统是从QNX软件系统和呼叫Neutrino。QNX专门化实时操作系统设计。微内核是抢先制人的，并且调度器是基于的优先级。这保证在进程之间的上下文转换程序非常快速，并且最高优先级的线索总是访问CPU，当要求。这些是Cisco IOS XR利用的一些好处。但是，最大的好处是进程间通信继承设计在操作系统核心内的。

中微子是通过操作系统的消息，并且消息是进程间通信的基本平均值在所有线索中的。当特定服务器要提供服务时，创建交换的消息一个信道。客户端附加到服务器信道通过对相关文件描述符为了使用服务的直接映射。客户端和服务端之间的所有通信是由同一机制。这是一台特级计算机的一个巨大的好处，CRS-1是。当一次本地读操作在标准UNIX内核时，被执行请考虑这些：

- 软件中断到内核里。
- 内核分派到文件系统里。
- 数据接收。

考虑这些在远程案件：

- 软件中断到内核里。
- 内核调度NFS。
- NFS呼叫网络组件。
- 远程调度网络组件。
- NFS呼叫。
- 内核调度文件系统。

读的本地的语义和读的远程不是相同的。变量和参数锁定和设置权限的文件的不同的。

认为QNX本地案件：

- 软件中断到内核里。
- 内核执行通过到文件系统的消息。

设想非本地案件：

- 软件中断到内核里。
- 内核进入QNET，是IPC传输机制。
- QNET进入内核。
- 内核调度文件系统。

关系到通过的参数的所有语义和文件系统系统参量是相同的。一切被分离了在允许客户端和服务端完全被分离的IPC接口。这意味着所有进程能这时任何地方运行。如果特定路由处理器太忙于服务请求，在DRP运作的您能容易地移植那些服务到不同的CPU。运行的一台特级计算机在不同的CPU的另外服务在能与其他节点容易地联络的多个节点间传播了。基础设施是到位为了提供机会扩展。思科利用此优点并且写入连接到通过内核的消息原理操作允许CRS路由器扩展到千位节点，节点，在这种情况下CPU，运行OS的实例的另外的软件，它是否是路由进程(RP)，一个分布式路由处理器(DRP)，模块化服务卡德(MSC)，或者交换机处理器(SP)。

进程和线索

在Cisco IOS XR内区域，进程是包含一个或更多线索的已保护内存区域。从程序员方面，线索完成工作，并且其中每一个完成一个逻辑执行路径为了执行一特定任务。他们运行内的线索要求在执行期间流的内存属于进程，保护从任何其他处理线索。线索是执行单元，与包括堆叠并且注册的执行上下文。进程是共享虚拟地址空间线索的一组，虽然进程能包含单个线索，但是经常包含更多。如果在一不同的进程的另一个线索尝试写入到在您的进程的内存，触犯的过程被结束。如果有超过结果在您的进程内运行的一个线索，则该线索访问在您的进程内的同一个内存和是有能力覆盖另一个线索数据。完成在步骤的步骤为了保持同步到资源为了防止在同一进程内的此线索。

线索使用呼叫互斥的一个对象(MUTEX)为了保证互斥到服务。有MUTEX的线索是能写入到内存特定区域为例的线索。没有MUTEX的其他线索不能。也有其他机制为了保证同步到资源，并且这些是信号量、条件变量或者Condvars，障碍和Sleepers。这些没有讨论在这儿，但是他们提供同步服务作为他们的责任的部分。如果等同于原理讨论此处对Cisco IOS，则Cisco IOS是操作许多线索，与访问同一个存储器空间的所有线索的单独进程。但是，Cisco IOS呼叫这些线索进程。

进程和线索状态

在Cisco IOS XR内有提供服务和服务端使用服务的服务器。一特定的进程能有提供同一服务的一定数量的线索。另一进程能有也许这时需要特定服务的很多个客户端。对服务器的访问总是不是可用的，并且，如果坐那里并且等待服务器是自由的对服务的一客户端的要求访问。在这种情况下客户端被认为阻止。这呼叫阻塞客户端服务器模型。客户端也许阻塞，因为它等待一种资源例如MUTEX，或者由于这样的事实服务器未回复。

发出ospf命令的show process为了检查线索的状况在OSPF程序的：

```
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 SHARED MEM MAIN MEM
      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
```

注意OSPF程序给工作ID (JID)，是250。这从未更改在运行路由器和通常在Cisco IOS XR特定版本。在OSPF程序内有与他们自己的线索ID (TID)的五个线索中的每一个。列出的是每个线索、每个线索优先级和其状态的堆栈空间。

同步消息通过

前面提到QNX是通过操作系统的消息。它实际上是通过操作系统的同步消息。很多操作系统的问题反射在同步消息传送。没有说同步消息通过引起所有问题，但是问题症状在同步消息通过相当反射。由于它同步，生命周期或状态信息是容易访问的对CRS-1操作员，在故障排除流程帮助。通过生命周期的消息类似于此：

- 服务器创建消息信道。
- 客户端连接到服务器的信道(类似于开放的posix)。
- 客户端传送信息到服务器(MsgSend)并且等待回复和块。
- 服务器收到(MsgReceive)从客户端的一个消息，处理消息和回复对客户端。
- 客户端疏导并且处理从服务器的回复。

此阻塞客户服务器模型是同步消息通过。这意味着客户端发送消息和块。服务器收到消息，处理它，回复回到客户端客户端然后疏导。这些是特定详细信息：

- 服务器在接收状态等待。
- 客户端传送信息到服务器并且变得阻止。
- 服务器收到消息并且疏导，如果等待在接收状态。
- 客户端迁移向回复状态。
- 服务器迁移向运行状态。
- 服务器进程消息。
- 对客户端的服务器回复。

• 客户端疏导。

发出show process命令为了在什么状态发现客户端和服务端是。

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

```

被阻塞的流程和进程状态

发出show process阻塞的命令为了看到什么进程在阻塞状态。

```

RP/0/RP1/CPU0:CWD CRS#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:CWD CRS#

```

同步的消息通过使您容易地跟踪生命周期区别线索之间的进程间通信。任何时间，线索可以在一特定状态。阻塞状态可以是问题的症状。这不意味着，如果线索在阻塞状态然后有问题，因此不发出show process阻塞的命令并且开有思科技术支持的一个Case。阻止线索也是非常正常。

注释上一个输出。如果查看在列表的第一个线索，请注释它是ksh，并且其回复在devc-conaux阻塞。客户端，在这种情况下ksh，传送了信息对devc-conaux进程，服务器，是devc-conaux，暂挂ksh回复阻塞，直到它回复。Ksh是某人在控制台或Aux端口使用的Unix shell。Ksh等待从控制台的输入，并且，如果有无，因为操作员不键入，然后保持阻止直到这样时刻处理若干输入。在处理以后，ksh返回应答阻止在devc-conaux。

这是正常，并且不说明问题。点是阻止线索是正常，并且依靠什么XR版本，系统种类您有，什么您配置，并且谁执行什么那修改show process阻塞的命令的输出。使用show process阻塞的命令是开始的好办法排除故障OS类型问题。如果有问题，例如CPU高，则使用前面的命令为了发现任何是否查找在正常外面。

了解什么为您作用的路由器是正常。当您排除故障进程生命周期时，这为您提供一个基准作为用用作比较。

任何时间，线索可以在特定状态。此表提供状态的列表：

如果状态是：	线索是：
停止	停止。内核等待发布线索资源。
运行	积极地运行在CPU
READY	不运行在CPU，但是准备好的
终止	暂停(SIGSTOP信号)
发送	等待服务器收到消息
接收	等待客户端传送信息
回复	等待服务器应答到消息
斯塔克	等待更多堆叠是请分配
WAITPAGE	等待进程管理器解决页错误

SIGSUSPEND	等待信号
SIGWAITINFO	等待信号
NANOSLEEP	一段时间休眠
MUTEX	获取MUTEX的等待
CONDVAR	等待一个条件变量发信号
加入	等待另一个线索的完成
INTR	等待中断
SEM	获取信号量的等待

重要进程和他们的功能

Cisco IOS XR有许多进程。这些是一些重要部分他们的功能解释此处。

监视器系统显示器(WDSysmon)

这是为进程的检测提供的服务暂停和低内存状况。由于内存泄漏或某个其他额外的情况，低内存能发生。暂停可以是一定数量的条件结果例如进程死锁、死环路、内核锁住或者日程安排错误。在所有多线程的环境系统能在叫作死锁情况或者死锁的状态获得。当一个或更多线索无法继续由于资源争用时，死锁能发生。例如，当线索B同时传送信息对线索A.时，线索A能传送信息到线索B。两个线索在彼此等待并且可以在发送阻塞状态，并且两个线索等待永久。这是介入两个线索的一个简单案件，但是，如果服务器对由许多线索使用的资源负责在另一个线索，然后请求的许多线索阻塞对该资源的访问可以是发送阻塞的等待在服务器。

结果死锁发生在一些个线索之间，但是能影响其他线索。死锁避免由好目的而设计，但是不考虑程序如何壮观地设计并且写入。有时一个特定的事件顺序是数据从属与的特定定时能导致死锁。死锁总是不确定的和通常是非常难再生产。WDSysmon有与运行在最高优先级中微子支持，63的一个的许多线索。运行在优先级63保证它线索获得在优先级基于先发制人的日程安排环境的CPU时间。WDSysmon与硬件监视器功能和看守一起使用在寻找暂停情况的软件进程。当这样情况检测时，WDSysmon在情况附近收集更多信息，能coredump进程或内核，写出到Syslog，运行脚本和结束被锁死的进程。从属在猛烈问题如何是，它可以启动路由处理器交换为了维护系统操作。

```
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

进程WDSysmon有九个线索。四运行在优先级10，人四在11，19，20和63。当进程设计时，程序员认真考虑优先级应该给在进程内的每个线索。如讨论以前，调度器是基于的，含义一个优先级更加高优先级的线索总是先占有一较低优先级。优先级63是线索能运行在，在这种情况下是线索7的最高优先级的。线索7是观察员线索，该的线索CPU HOG的跟踪。它比否则观看它必须运行在高优先级的其他线索也许没获得机会运行，防止它步骤设计执行。

Netio

在Cisco IOS，有快速交换和进程交换的概念。快速交换使用CEF代码并且发生在中断时间。处理交换用途ip_input，是IP交换代码，并且是定期的进程。在更加高端的平台上CEF交换在硬件方面完成，并且ip_input在CPU被安排。ip_input等同在Cisco IOS XR的是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

组服务进程(GSP)

其中每一运行其内核的自己的实例的有对通信的需要在所有级计算机与数千节点该。在互联网中

，一个对许多通信通过组播协议高效地执行。GSP是使用在CRS-1内的IPC的内部组播协议。GSP提供一给许多是无连接与异步语义的可靠组通信。这允许GSP扩展到一千节点。

```
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 SHARED MEM MAIN MEM
      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
```

[BCDL容量内容下载者](#)

BCDL可靠是使用的为了组播数据对多种节点例如RP和MSC。它使用GSP作为基础传输。在消息交付的BCDL保证。在BCDL内有代理程序、制造商和用户。代理程序是与制造商联络为了在其组播前获取和缓冲数据对消费者的进程。制造商是导致数据大家想要的进程，并且用户是进程感兴趣接收制造商提供的数据。在Cisco IOS XR软件升级期间，BCDL使用。

[轻量级消息传送\(LWM\)](#)

LWM是设计创建抽象层在应用程序之间的相互进程与彼此和中微子联络，与目标作为操作系统和传输层的独立消息传送的一Cisco创建的表。如果思科希望更改从QNX的OS供应商对别人，抽象层在底层操作系统帮助的基本功能的之间取消在操作系统和帮助的从属关系在移植到另一个操作系统。LWM提供同步保证消息发送，喜欢通过本地中微子的消息，造成发送方阻塞直到接收方回复。

LWM通过40位脉冲也提供异步消息发送。异步信息异步地传送，含义消息排队，并且发送方不由服务器阻塞，然而没有接收异步地，但是，当服务器为下个可用的消息时轮询。LWM被构造如客户端/服务器。服务器创建给它耳朵细听消息并且坐环路有一段时间执行侦听在信道的消息接收，创建的信道。当消息到达时疏导并且获得客户端标识符，有效是事和从消息的接收ID一样接收。服务器然

后执行处理的一些，并且以后执行对客户端标识符的一消息回复。

在客户端它执行消息连接。它被通过然后连接执行消息发送和阻塞的标识符。当服务器完成处理时，回复和客户端变得疏导。这实际上是相同的象通过中微子本地的消息，因此抽象层非常稀薄是。

LWM用系统呼叫和上下文交换机最小数量的设计高性能的，并且是IPC首选方法在Cisco IOS XR环境。

[Envmon](#)

至多级的基本，环境监控系统对警告，当物理参数，例如温度，电压，等等风扇速度，下跌在可操作的范围外面时和关闭达到关键级别也许损坏硬件的硬件负责。它监控每个可用的硬件传感器，对卡片细节的阈值周期地比较测量值，并且如所需要发出报警为了完成此任务。一不变进程，开始在系统初始化，周期地轮询所有硬件传感器，例如电压，温度和风扇速度，在机箱和提供此数据给外部管理客户端。另外，定期进程传感器读与告警门限值比较并且发布环境警报对随后的操作的系统数据库由Fault Manager。如果传感器读危险是超出范围，环境监控进程也许造成卡被关闭。

[CRS-1结构介绍](#)

- 多级结构— 3个阶段Benes拓扑
- 在最小化拥塞的结构内的动态路由
- 基本存储单元：136个字节信元，120个字节数据负载
- 改善数据流隔离和最小化在结构的缓冲需求的流量控制
- 演出加速传送的阶段
- 支持的(单播&组播)流量两个转换
- 每个转换支持的流量两优先级(高和低)
- 1M结构组播组的(FGIDs)支持
- 有效容错：N+1或N+k冗余使用与1+1相对的结构飞机在非常地增加的开销

当您在单机箱模式时运作，S1、S2和S3 asics在同样结构卡查找。此卡通常也指S123卡。在多机箱配置中，S2被分离，并且在结构卡机箱(FCC)。此配置要求两张结构卡形成飞机、S2卡和S13卡。每个MSC连接到八架结构飞机为了提供冗余，以便，如果疏松一个或更多飞机，您的结构仍然通过流量，虽然总流量，可以通过结构，更低。CRS只能仍然运行在多数数据包大小的线路速率用七架飞机。反压力在一架多架和均等飞机的结构发送。没有推荐运行系统以少于两架飞机，在一架多架和均等飞机。任何少于两架飞机不是支持的配置。

[结构飞机](#)

上一个图表代表一架飞机。您必须乘该图表以八。那意味着LC的喷雾器(ingressq) asic连接对8 S1s (每架飞机1个S1)。在每架结构飞机的S1连接到8台喷雾器：

- 机箱的8个顶部LCs
- 8个底下LCs

有每16个slot LC机箱的16 S1s：8顶部LCs的(1每架飞机) + 8底部LCs的。

在单个16个slot机箱上，S123结构卡有2 S1s、2 S2和4 S3s。那是结构加速计算的一部分。两倍有同等数量流量，能退出结构，象流量能输入。当前也有两块海绵(fabricq)每个LC与1台喷雾器比较。这允许缓冲在出口LC，当超过一入口LCs超载出口LC。出口LC能吸收从结构的该额外带宽。

[结构监听](#)

平面可用性和连接：

```
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 SHARED MEM MAIN MEM
    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
```

检查飞机是否是信元的接收/传送，并且一些错误增加：

```
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 SHARED MEM MAIN MEM
    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

在前面的命令的缩略语：

- CE —可校正错误
- UCE —无法修复的错误
- PE —奇偶校验错误

请勿担心，如果他们注意一些个错误，因为这在启动能发生。字段不应该增加在运行时间。如果他们，它可以是一问题的征兆在结构的。发出此命令为了获得错误的细分每架结构飞机：

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

控制层面概述

线路卡机箱和结构机箱之间的控制层面连接当前是通过RP (LCC)和SCGE的(FCC)千兆以太网端口。在端口之间的互连通过一个对Catalyst 6500交换机提供，可以通过两个或多个千兆以太网端口连接。

Catalyst 6500配置

这是用于多机箱控制层面的Catalyst交换机的推荐的配置：

- 单个VLAN在所有端口使用。
- 在接入模式运行的所有端口(没有中继)。
- 生成树802.1w/s使用环路预防。
- 两个或多个链路用于为了交叉连接两交换机，并且STP使用防止出现环路。没有推荐信道。
- 连接对CRS-1 RP和SCGE使用预标准模式的端口，因为IOS-XR不支持标准基于802.1s。
- 在连接在交换机之间和在交换机和RP/SCGE之间的端口应该启用UDLD。
- 默认情况下UDLD在CRS-1启用。

参考[启动在Multishelf系统的Cisco IOS XR软件](#)关于如何配置在Multishelf系统的一台Catalyst 6500的更多信息。

多机箱控制层面管理

Catalyst 6504-E机箱，为多机箱系统提供控制层面连接，为这些管理服务配置：

- 带内管理通过端口千兆位1/2，连接到LAN交换机在每Pop。访问为各种各样的子网和协议只允许。
- NTP用于为了设置系统时间。
- Syslogging执行到标准的主机。
- SNMP轮询和陷阱可以为重要功能启用。

注意： 不应该做变动对Catalyst运转中。在所有计划内更改应该执行前期测试在维护窗口期间，我们极力推荐，并且这执行。

这是管理配置示例：

```
#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和Monlib

思科monlib是在设备存储并且装载到执行的RAM由ROMMON的可执行程序。ROMMON使用monlib为了访问在设备的文件。ROMmon版本升级并且应该如此执行在思科技术支持的建议下。最新的ROMmon版本是1.40。

升级指南

完成这些步骤：

1. 下载从[思科CRS-1 ROMMON \(仅限注册用户\)](#)的ROMMON二进制。
2. 打开TAR文件并且复制6个Bin文件到Disk0 CRS根目录。

```
#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. 请使用show diag|inc ROM|NODE|PLIM命令为了看到当前ROMmon版本。

```
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. 进入ADMIN模式并且使用**upgrade rommon**所有**disk0**命令为了升级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. 退出ADMIN模式和回车**show log|好的Inc "ROMMON A"**和确保所有节点顺利地升级。如果其中任一节点发生故障，进入上一步步骤4并且重编程序。

```
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. 进入ADMIN模式并且使用**upgrade rommon b**所有**disk0**命令为了升级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. 退出ADMIN模式和回车**show log|好的Inc "ROMMON B"**和确保所有节点顺利地升级。如果其中任一节点发生故障，进入上一步步骤4并且重编程序。

```
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. **upgrade**命令烧录Bootflash的一个特殊保留部分与新的ROMMON的。但是新的ROMMON保持非激活，直到卡重新加载。因此，当您重新加载卡，新的ROMMON是活跃的。重置一次一个每个节点或请重置整个路由器为了执行此。

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. 请使用**show diag|inc ROM|NODE|PLIM**命令为了检查当前ROMmon版本。

```
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 : 40C192-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 : 160C48-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]
```

注意：在CRS-8和结构机箱上，ROMMON也设置风扇速率为默认速度4000个RPM。

PLIM和MSC概述

这代表在CRS-1路由器的数据包流，并且可互换使用这些术语：

Ingressq ASIC也呼叫喷雾器ASIC。

Fabricq ASIC也呼叫海绵ASIC。

EgressQ ASIC也呼叫Sharq ASIC。

SPP也呼叫PSE (分组交换机引擎) ASIC。

Rx PLIM > Rx SPP >入口问>结构>结构问> Tx SPP >出口问> Tx PLIM (喷雾器) (海绵) (Sharq)

数据包在物理层接口模块(PLIM)接收。

PLIM包含联接的MSC的物理接口。PLIM和MSC是通过机箱底板连接的独立的卡。结果特定MSC的接口类型由联接与PLIM的种类定义。从属在PLIM种类，卡包含为接口提供物理媒介和帧ASIC的一个多种编号。PLIM ASIC的目的将提供MSC和物理连接之间的接口。它中断光纤，执行灯对电子转换，终止成帧的媒体是SDH/Sonet/Ethernet/HDLC/PPP，检查CRC，添加一些控制信息呼叫缓冲报头并且转发依然是在MSC上的位。PLIM不来源/接收器HDLC或PPP Keepalive。这些由在MSC的CPU处理。

PLIM也提供这些功能：

- 过滤为1/10千兆以太网的MAC
- 1/10千兆以太网的入口/出口MAC记帐
- 1/10千兆以太网的VLAN过滤
- 1/10千兆以太网的VLAN记帐
- 入口缓冲和拥塞通知

PLIM超额预订

10GE PLIM

MSC的转发容量是40 Gbps，最大数量8个X10G PLIM提供功能近似终止流量80 Gbps。如果所有端口可用在PLIM填充，则超额预订发生，并且QoS模拟变得非常重要保证重要数据流没有疏忽地丢弃。对一些人来说，超额预订不是选项，并且必须避免。必须用于仅四八个端口为了执行此。另外，必须保重保证在MSC和PLIM内的最佳带宽对四个端口中的每一个是可用的。

注意： 端口映射从版本3.2.2向前更改。请参阅这些图表。

至版本3.2.1的端口映射 从版本3.2.2的端口映射向前

如前所提及，物理端口由两个Fabricq ASIC之一服务。端口的分配ASIC的静态定义并且不可能被修改。另外，8个X10G PLIM有两个PLA ASIC。第一个PLA服务端口0到3，第二种服务类别4到7。单个PLA的带宽容量在8个X10G PLIM的近似是24 Gbps。单个Fabricq ASIC的交换能力是大约62 Mpps。

如果填充端口0到3或者端口4到7，PLA (24 Gbps)的带宽容量共享在限制整体吞吐量的所有四端口之间。如果填充端口0,2,4 & 6 (3.2.1)或0,1,4 & 5 (3.2.2向前)和所有这些端口由一Fabricq ASIC，再

，服务，交换能力是62 Mpps限制吞吐量。

有些使用端口得到PLAs和Fabricq ASIC高效率为了完成最佳性能的是最佳的。

SIP-800/SPA

SIP-800 PLIM提供能力运行与叫作服务端口适配器的模块化接口界面卡(温泉)。SIP-800提供6个SPA海湾60 Gbps一理论上的接口产能。MSC的转发容量是40 Gbps最大数量。如果所有在SIP-800的海湾将填充，则，从属在SPA类型，很可能，超额预订发生，并且QoS模拟变得非常重要为了保证重要数据流没有疏忽地丢弃。

注意：超额预订不支持与POS接口。但是，10个Gb POS SPA的放置一定是适当为了保证正确吞吐量提供。IOS-XR版本3.4只支持10个Gb以太网SPA。此SPA提供超额预订功能。

对一些人来说，超额预订不是选项，并且必须避免。必须用于仅四sixbays为了执行此。另外，必须保重为了保证在MSC和PLIM内的最佳带宽对四个端口中的每一个是可用的。

SPA海湾映射

按照以前所述，物理端口由两个Fabricq ASIC之一服务。端口的分配ASIC的静态定义并且不可能被修改。另外，SIP-800 PLIM有两个PLA ASIC。第一个PLA服务端口0,1 & 3，第二种服务类别2，4 & 5。

单个PLA的带宽容量在SIP-800 PLIM的近似是24 Gbps。单个Fabricq ASIC的交换能力是大约62 Mpps。

如果填充端口0,1 & 3或者端口2，4 & 5，PLA (24 Gbps)的带宽容量共享在限制整体吞吐量的所有三端口之间。因为每单个的Fabricq服务那些端口组，端口组的最大信息包速率是62 Mpps。有些使用端口得到PLAs高效率为了达到最佳带宽的是最佳的。

建议的放置：

	SPA bay-	SPA bay-	SPA bay-	SPA bay-
第 1 项	0	1	4	5
第 2 项	1	2	3	4

如果要填充与超过四个SPA的卡，建议是完成其中一个选项以前列出，传播两端的组之间的接口(0,1 & 3 & 2,4 & 5)。您应该在其中一二者之一的开放端口中然后安置下个SPA模块0,1个& 3个& 2,4个& 5个端口组。

DWDM XENPACKs

从版本3.2.2向前，DWDM XENPACKs能安装和提供可调的光纤模块。冷却要求的这样XENPACK模块要求有在安装模块之间的一坏插槽。另外，如果单个DWDM XENPACK模块安装，可以使用最多四个端口，即使XENPACK模块不是DWDM设备。因此这有在Fabricq的一直接影响对PLA对端口映射。注意在此表里需要给予到此需求和考虑。

建议的放置：

	光学port-	光学port-	光学port-	光学port-
--	---------	---------	---------	---------

选项1或DWDM XENPACK	0	2	5	7
第2项	1	3	4	6

对于3.2.2或以后或3.3安装，请避免映射更改的Fabricq。因此一个更加简单的放置模式可能用于正常和DWDM XENPACK模块。

	光学port-	光学port-	光学port-	光学port-
第1项	0	2	4	6
第2项	1	3	5	7

如果要填充卡用超过四个非DWDM XENPACK端口，建议是完成其中一个选项列出，涂在两端的组之间的光接口接口模块(0-3 & 4-7)。您在0-3或4-7个端口组中需要然后安置下个光接口接口模块在其中一开放端口中。如果使用0-3端口组光接口接口模块#5，在4-7端口组中应该安置光接口接口模块#6。

参考的[DWDM XENPAK模块](#)欲了解更详细的信息。

配置管理

在IOS-XR的配置通过一两阶段配置被执行，配置由第一阶段的用户输入。这是仅配置语法由CLI检查的阶段。在此阶段输入的配置只为对座席进行配置进程所知，例如，CLI/XML。因为没有写入到sysdb服务器，配置没有验证。后端应用程序没有通知并且不能访问或有配置的任何知识在此阶段的。

在第二阶段，配置由用户明确地提交。在此阶段配置写入到sysdb服务器，后端应用程序验证配置，并且通知由sysdb生成。在您提交在第一阶段前，输入的配置您能中止配置会话。因此，假设是不安全的，在输入的所有配置阶段一在阶段两总是被提交。

另外，路由器的操作和运行的配置可以由多个用户修改在阶段一和阶段两。因此，治理配置和操作状态的阶段一路由器的任何测验也许不是有效在阶段两配置实际上被提交的地方。

配置文件系统

配置文件系统(CFS)是一套文件和目录用于为了存储路由器的配置。CFS存储在目录disk0:/config/下，是在RP使用的默认媒体。文件和目录在CFS是内部对路由器，并且应该由用户从未修改或删除。这能导致配置和影响服务的损耗或损坏。

CFS是检查点对standby-RP在每进行以后。这在故障切换以后帮助保留路由器的配置文件。

在路由器启动期间，最后活动配置从在CFS存储的配置进行数据库应用。因为这由路由器，自动地完成手工保存活动配置在每配置进行以后用户是不必要的。

在启动期间时，当配置应用做配置更改是不可行的。如果配置应用程序不完成，您看到此消息，当您登录到路由器时：

系统配置进程

此设备的启动配置目前装载。这能花费几分钟。您通知在完成。请勿尝试重新配置设备，直到此进程完成。少许罕见的情况，恢复从用户的路由器配置提供了ASCII配置文件也许是理想的而不是恢复从CFS的最后活动配置。

您能强制配置文件的应用程序：

```
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>
```

当您恢复路由器配置时，一个或更多配置项也许不能生效。所有失败的配置在CFS保存和维护直到下一次重新加载。

您能浏览失败的配置，寻址错误和重新应用配置。

在路由器的启动期间，这些是一些提示为了说明失败的配置。

在IOX，配置可以由于三个原因分类作为失败的配置：

1. 语法错误—分析程序生成语法错误，通常表明有一不兼容用CLI命令。您应该更正语法错误和重新应用配置。
2. 语义错误—在路由器的启动期间时，当配置管理器恢复配置语义错误由后端组件生成。请注意作为运行的配置一部分，cfgmgr对保证配置不负责接受。Cfgmgr仅仅是中间人和只报告后端组件生成的所有语义失败。它是至分析失败原因和确定失败的原因的每个后端组分所有者。用户能执行从配置模式的**描述<CLI commands>**为了容易地找到后端组分检验器的所有者。例如，如果**router bgp 217**出现作为失败的配置，**描述**命令显示组分检验器是ipv4-bgp组件。

```
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. 应用错误—配置顺利地验证并且接受作为运行的配置一部分由于某种原因，但是后端组件不能更新其操作状态。配置显示在两运行的配置，因为正确验证和作为失败的配置由于后端可操作的错误。**描述**命令在CLI可能再运行失败应用为了查找组件应用所有者。在起始操作员期间，完成这些步骤为了浏览和重新应用失败的配置：对于R3.2操作员能使用此步骤为了重新应用失败的配置：操作员能使用**show configuration失败的启动**命令为了浏览在路由器启动期间保存的失败的配置。操作员应该运行**show configuration失败的启动无错|文件myfailed.cfg**命令为了保存起始失败的配置到文件。操作员应该去**配置模式**和使用**负载/进行**命令为了重新应用此失败的配置：

```
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
```

对于R3.3镜像操作员能使用此更新步骤：操作员必须使用**show configuration失败的启动命令**和**负载配置失败的启动命令**为了浏览和重新应用所有失败的配置。

```
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
```

默认情况下IOS-XR写入core dump到光盘如果进程失败，但是没有，如果内核失败。注意对于多机箱系统此功能为线路卡机箱0只当前支持。软件以后的版本支持另一个机箱。

被建议内核为RP和MSC在标准和模式配置方面转存启用与使用这些配置：

```
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
```

内核转储配置

这导致内核失败的此出现：

1. RP失败，并且转储写入到在该RP的光盘在磁盘的根目录。
2. 如果MSC失败，转储写入到RP0光盘在磁盘的根目录的。

这对RP故障切换时间没有影响，因为Non-Stop Forwarding (NSF)为路由协议配置。它能花费失败的RP或线路卡的一些额外的分钟能再变得可用，在跟随失败后，当写入核心时。

此配置的新增内容的示例对标准和admin模式配置的显示此处。注意admin模式配置要求DRPs将使用的。

此输出显示内核转储配置示例：

```
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)#
```

安全

LPTS

本地信息包本地传输服务(LPTS)处理注定的数据包。LPTS由多种不同的组件制成。

1. 主要一个呼叫端口仲裁人进程。它听socket从不同的协议进程的请求，例如，BGP，IS-IS并且记录那些进程的所有约束信息。例如，如果BGP进程侦听在插口号179，PA从BGP进程得到该信息，然后分配约束到在IFIB的该进程。
2. IFIB，是LPTS进程的另一个组件。它帮助保持进程侦听到一个特定端口绑定的目录。IFIB由波尔特仲裁人进程生成和保持与端口仲裁人。它然后生成此信息的多子集。首先子集是片式IFIB。此片式可以关联到IPv4协议等等。片式然后发送合适流管理器，然后使用IFIB片式为了转发数据包到适当的进程。如果仅一进程存在或给一个适当的流管理器，第二子集是pre-IFIB，允许LC转发数据包到适当的进程。
3. 如果查看是重要的，例如，BGP的，多个进程流管理器进一步帮助分配数据包。每个流管理器有片式或多个片式IFIB和适当地转发数据包对适当的进程关联与片式IFIB。
4. 如果条目没有为目的地端口定义那么可能丢弃或转发对流管理器。如果有端口的，一项相关的策略数据包转发没有关联的端口。流管理器帮助然后生成个新会话条目。

内部数据包如何转发？

有流、Layer2 (HDLC，PPP)流和Layer4 ICMP/PING流和路由流的两种类型。

1. Layer2 HDLC/PPP —这些数据包由协议标识符识别和被发送直接地对喷雾器的CPU队列。第2层协议数据包获得高优先级和由CPU然后拾起(通过Squid)并且处理。因此Layer2的Keepalive直接地响应对通过LC通过CPU。这避免需要去答复的RP并且使用与分布式接口管理主题。
2. ICMP (层4)数据包在LC接收，并且他们通过查找发送通过IFBI到在喷雾器的CPU队列。这些数据包然后被发送对CPU (通过Squid)并且处理。答复通过喷雾器出口队列然后被发送为了通过结构转发。这是，万一另一应用程序也需要信息(复制通过结构)。一旦通过结构数据包被注定对适当的出口LC和通过适当的海绵和控制队列。
3. 路由流在IFIB查寻然后发送对整形队列(8000个队列)其中之一为控制数据包保留。在全双工时候，这是一个非整形过的队列和被服务。-高优先级。数据包通过在高优先级队列的结构到一套在海绵的CPU队列(类似于Squid排队在喷雾器)，然后进程然后发送由适当的进程、流管理器或者实际过程。答复被退还的通过出口线路卡海绵然后线卡。出口LC海绵有留出的一个特殊队列处理控制数据包。在海绵的队列拆分到高优先级、控制和低优先级信息包，每个输出端口基本类型。
4. PSE有为限制Layer4、Layer2和路由信息包的速率配置的一套策略。这些以后被预先设定并且更改是可配置的用户。

一个与LPTS的最常见的问题是丢弃的数据包，当您尝试ping路由器时。LPTS策略通常是限制这些数据包的速率。这是实际情形为了确认：

```
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

IP信息包是本能地不安全的。IPsec是使用的方法保护IP信息包。CRS-1 IPsec在软件转发路径实现，因此IPSec会话在RP/DRP终止。支持500 IPsec会话总数每CRS-1。编号取决于CPU速度和已分配资源。没有软件限制对此，只有在RP的来源于本地的和本地终止流量有资格IPsec处理。IPsec传输模式或隧道模式可以用于流量类型，虽则前面首选归结于在IPSec处理的较少开销。

R3.3.0支持BGP和OSPFv3的加密在IPsec。

关于如何的参考的[Cisco IOS XR系统安全配置指南](#)实现IPsec的更多信息。

注意： IPsec要求crypto饼，例如， hfr-k9sec-p.pie-3.3.1。

带外

控制台和AUX访问

CRS-1 RP/SCs有出于波段管理目的—控制台和AUX端口联机，以及带外的一个管理以太网端口通过IP。

每RP/SCGE控制台和AUX端口，两每机箱，可以连接到控制台服务器。这意味着单个机箱系统要求四个控制台端口，并且多机箱系统要求12个端口加上Supervisor引擎的两个进一步端口在Catalyst 6504-E。

Aux端口连接是重要，因为提供存取对于IOS-XR内核，并且能允许系统恢复，当这通过控制台端口时不是可能的。访问通过Aux端口只供给在系统本地定义的用户，并且，只有当用户得以进入根系统或支持级别时。另外用户必须安排**加密口令**定义。

虚拟终端访问

Telnet &安全壳SSH可以用于为了通过VTY端口到达CRS-1。默认情况下两个禁用，并且用户需要明确地启用他们。

注意： IPsec要求crypto饼，例如， hfr-k9sec-p.pie-3.3.1。

首先请生成RSA和DSA密钥如此示例所显示为了启用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
```

相关信息

- [路由器支援](#)
- [技术支持和文档 - Cisco Systems](#)