

基準流程最佳實踐白皮書

目錄

[簡介](#)

[基線](#)

[什麼是基線？](#)

[為什麼選擇基線？](#)

[基線目標](#)

[核心基線流程圖](#)

[基線程式](#)

[第1步：編譯硬體、軟體和配置清單](#)

[第2步：驗證路由器是否支援SNMP MIB](#)

[第3步：輪詢和記錄來自路由器的特定SNMP MIB對象](#)

[步驟4：分析資料以確定閾值](#)

[第5步：解決已確定的立即問題](#)

[步驟6：測試臨界值監督](#)

[第7步：使用SNMP或RMON實施閾值監控](#)

[其他MIB](#)

[路由器MIB](#)

[Catalyst交換器MIB](#)

[序列連結MIB](#)

[RMON警報和事件配置命令](#)

[警報](#)

[活動](#)

[RMON警報和事件實施](#)

[相關資訊](#)

簡介

本檔案說明高可用性網路的基準概念和程式。它包括網路基準和閾值的重要成功因素，可幫助評估成功情況。它還提供了重要細節，說明遵循思科高可用性服務(HAS)團隊確定的最佳實踐指南的基線和閾值流程以及實施。

本文檔將逐步引導您完成基線建立過程。目前有些網路管理系統(NMS)產品可協助將此程式自動化，不過，無論您使用自動化或手動工具，基準線程式都保持不變。如果使用這些NMS產品，則必須調整您的唯一網路環境的預設閾值設定。必須有一個智慧選擇這些閾值的流程，以便它們有意義且正確。

基線

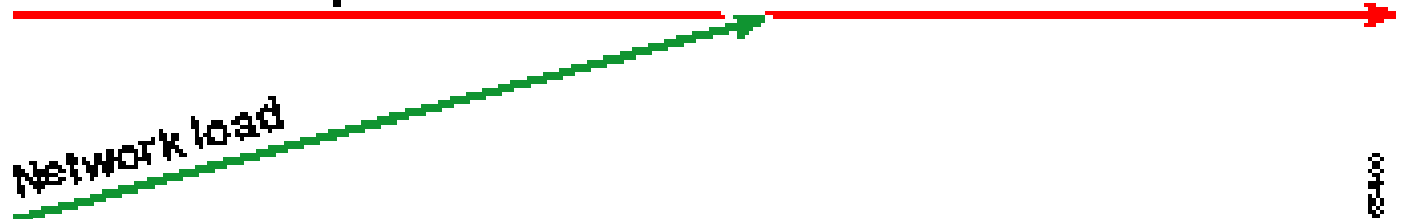
[什麼是基線？](#)

基線是一個定期研究網路以確保網路按設計運行的過程。它不只是一份報告來詳述網路在某一時間點的運行狀況。透過遵循基準線程式，您可以取得下列資訊：

- 獲取有關硬體和軟體運行狀況的寶貴資訊
- 確定網路資源的當前利用率
- 做出有關網路警報閾值的準確決策
- 確定當前的網路問題
- 預測未來問題

下圖說明了檢視基線的另一種方式。

Network breakpoint



紅線（網路中斷點）是網路中斷的點，它取決於硬體和軟體運行方式的知識。綠線，即網路負載，是隨著新應用的增加及其他此類因素在網路上的自然發展。

基準線的用途是決定：

- 您的網路處於綠線上的位置
- 網路負載增加的速度
- 希望預測兩者在什麼時候會相交

透過定期執行基線，您可以瞭解當前狀態並推斷故障何時發生並提前做好準備。這還可以幫助您更明智地決定何時、何處以及如何將預算資金用於網路升級。

為什麼選擇基線？

基線過程可幫助您確定網路中的關鍵資源限制問題並對其進行正確規劃。這些問題可描述為控制平面資源或資料平面資源。控制平面資源是裝置內特定平台和模組所特有的，可能會受到以下問題的影響：

- 資料利用率
- 啟用的功能
- 網路設計

控制平面資源包含引數，例如：

- CPU使用率
- 記憶體利用率
- 緩衝區利用率

資料平面資源僅受流量型別和數量的影響，包括鏈路利用率和背板利用率。透過對關鍵區域的資源利用率進行基準測試，您可以避免嚴重的效能問題，甚至避免網路崩潰。

隨著語音和影片等延遲敏感型應用的引入，基線處理現在比以往任何時候都更加重要。傳統的傳輸控制通訊協定/網際網路通訊協定(TCP/IP)應用程式可寬恕並允許一定的延遲。語音和影片基於使用者資料包協定(UDP)，不允許重新傳輸或網路擁塞。

由於新的應用程式組合，基線分析可幫助您瞭解控制層面和資料層面的資源利用率問題，並主動規劃更改和升級，以確保持續成功。

資料網路已經存在多年。直到最近，保持網路運行一直是一個相當寬宏大量的過程，其中有一些出錯的餘地。隨著IP語音(VoIP)等延遲敏感型應用的接受度不斷提高，網路運行工作變得越來越困難，對精度的要求也越來越高。為了更精確並給網路管理員管理網路奠定堅實的基礎，瞭解網路如何運行非常重要。為此，您必須完成一個名為基線的過程。

基線目標

基準線的目標是：

1. 確定網路裝置的當前狀態
2. 將該狀態與標準效能準則進行比較
3. 設定臨界值，以便在狀態超過這些準則時發出警示

由於資料量大，且分析資料的時間長，您必須先限制基準線的範圍，以便更輕鬆地瞭解該程式。最符合邏輯、有時也最有益的地方，就是從網路的核心開始。網路的此部分通常最小，且需要最穩定的裝置。

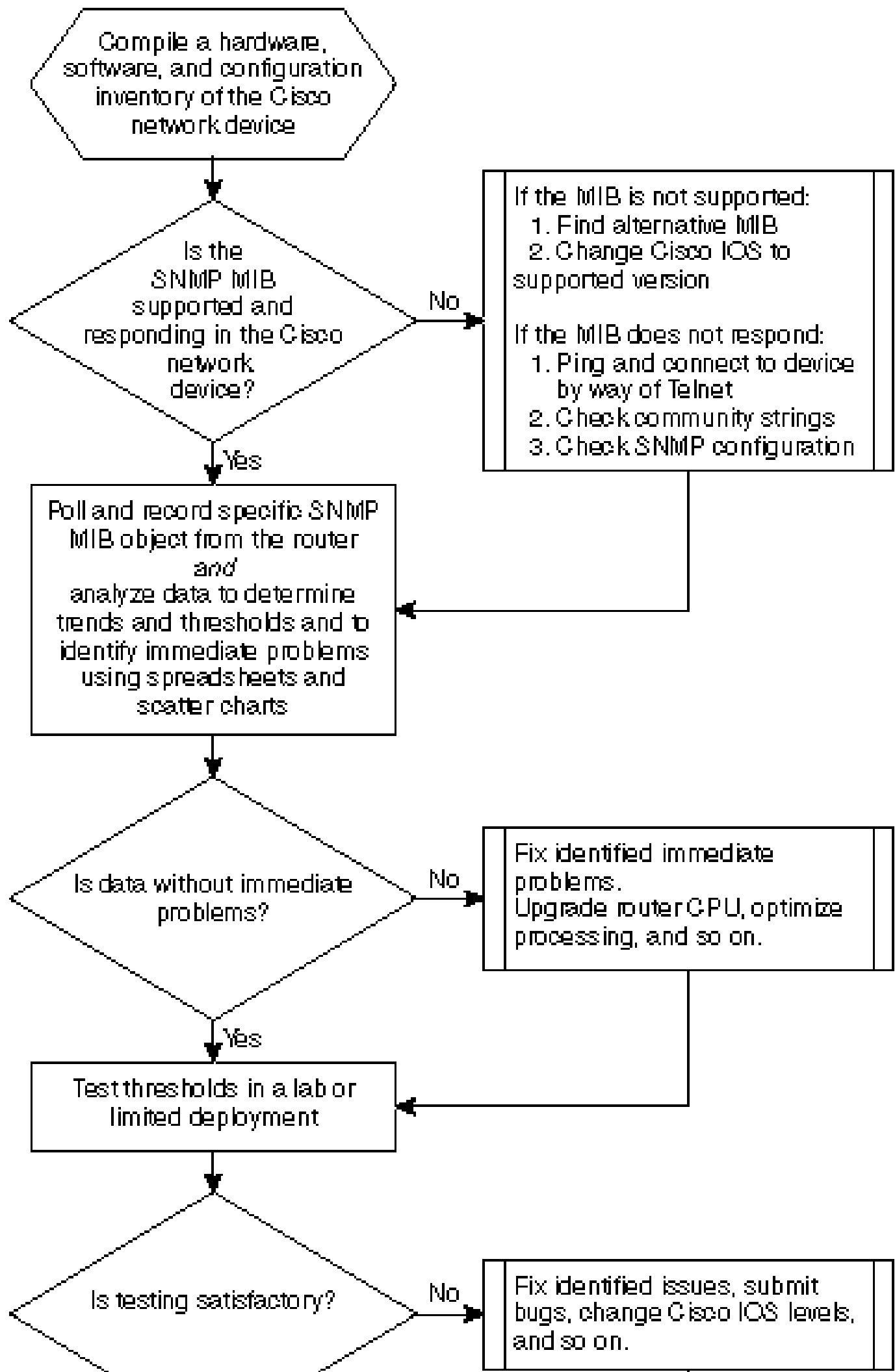
為簡單起見，本文檔說明如何確定一個非常重要的簡單網路管理協定管理資訊庫(SNMP MIB)的基線：cpmCPUTotal5min。cpmCPUTotal5min是思科路由器的中央處理器(CPU)的五分鐘衰減平均值，並且是控制平面效能指標。基線將在Cisco 7000系列路由器上執行。

瞭解該過程後，即可將其應用於大多數思科裝置上可用的海量SNMP資料庫中的任何可用資料，例如：

- 整合服務數位網路(ISDN)的用法
- 非同步傳輸模式(ATM)基地台遺失
- 可用系統記憶體

核心基線流程圖

下列流程圖顯示核心基準線處理作業的基本步驟。雖然有產品和工具可供您執行這些步驟中的某些步驟，但它們在靈活性和易用性方面往往存在差距。即使您計畫使用網路管理系統(NMS)工具執行基線劃分，但這仍是學習該過程並瞭解網路實際工作原理的一個好練習。此過程可能還會揭開一些NMS工具工作原理的神秘面紗，因為大多數工具基本上都做同樣的事情。



基線程式

第1步：編譯硬體、軟體和配置清單

編譯硬體、軟體和配置的清單非常重要，原因有幾個。首先，在某些情況下，Cisco SNMP MIB特定於您正在運行的Cisco IOS版本。有些MIB對象被替換為新對象或有時被完全刪除。收集資料後，硬體資產非常重要，因為您需要在初始基線後設定的閾值通常取決於思科裝置上的CPU型別、記憶體容量等。配置清單對於確保您瞭解當前配置也非常重要：您可能希望在基線之後更改裝置配置以調整緩衝區等。

對思科網路來說，這部分基準的最有效方法是使用CiscoWorks2000 Resource Manager Essentials (Essentials)。如果此軟體在網路中安裝正確，則Essentials應擁有其資料庫中所有裝置的當前清單。您只需要檢視庫存以瞭解是否存在任何問題。

下表是從Essentials導出並在Microsoft Excel中編輯的Cisco路由器類軟體清單報告的示例。在本清單中，請注意，您必須使用12.0x和12.1x Cisco IOS版本中的SNMP MIB資料和對象識別符號(OID)。

裝置名稱	路由器型別	版本	軟體版本
field-2500a.embu-mlab.cisco.com	Cisco 2511	M	12.1(1)
qdm-7200.embu-mlab.cisco.com	Cisco 7204	B	12.1(1)E
voip-3640.embu-mlab.cisco.com	Cisco 3640	0x00	12.0(3c)
wan-1700a.embu-mlab.cisco.com	Cisco 1720	0x101	12.1(4)
wan-2500a.embu-mlab.cisco.com	Cisco 2514	L	12.0(1)
wan-3600a.embu-mlab.cisco.com	Cisco 3640	0x00	12.1(3)
wan-7200a.embu-mlab.cisco.com	Cisco 7204	B	12.1(1)E
172.16.71.80	Cisco 7204	B	12.0(5T)

如果網路中未安裝Essentials，則可以從UNIX工作站使用UNIX命令列工具snmpwalk查詢IOS版本。如下例所示。如果不確定此命令如何工作，請在UNIX提示符處鍵入man snmpwalk，以瞭解更多資訊。當您開始選擇要比較基準的MIB OID時，IOS版本非常重要，因為MIB對象取決於IOS。另請注意，透過瞭解路由器型別，您可以稍後確定應該為CPU、緩衝區等設定什麼閾值。

```
nsahpov6% snmpwalk -v1 -c private 172.16.71.80 system
system.sysDescr.0 : DISPLAY STRING- (ascii): Cisco Internetwork Operating System Software
IOS (tm) 7200 Software (C7200-JS-M), Version 12.0(5)T, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Fri 23-Jul-2001 23:02 by kpma
system.sysObjectID.0 : OBJECT IDENTIFIER:
.iso.org.dod.internet.private.enterprises.cisco.ciscoProducts.cisco7204
```

第2步：驗證路由器是否支援SNMP MIB

現在，您已經擁有要輪詢基準的裝置，可以開始選擇要輪詢的特定OID。如果您提前確認您想要的資料確實存在，這將為您省去很多麻煩。cpmCPUTotal5min MIB對象位於CISCO-PROCESS-MIB中。

要查詢要輪詢的OID，您需要一個轉換表，該表可在思科的CCO網站上找到。要從Web瀏覽器訪問此網站，請轉至[Cisco MIBs頁](#)，然後按一下OIDs連結。

要從FTP伺服器訪問此網站，請鍵入ftp://ftp.cisco.com/pub/mibs/oid/。您可以從此站點下載已解碼並按OID編號排序的特定MIB。

以下示例摘自CISCO-PROCESS-MIB.oid表。本示例顯示cpmCPUTotal5min MIB的OID為1.3.6.1.4.1.9.9.109.1.1.1.5。

注意：不要忘記將「。」增加到OID的開頭，否則當您嘗試輪詢它時將出現錯誤。您還需要將「.1」增加到OID的末尾以例項化OID。這會告知裝置您要尋找的OID執行個體。在某些情況下，OID具有特定型別資料的多個例項，例如路由器有多個CPU時。

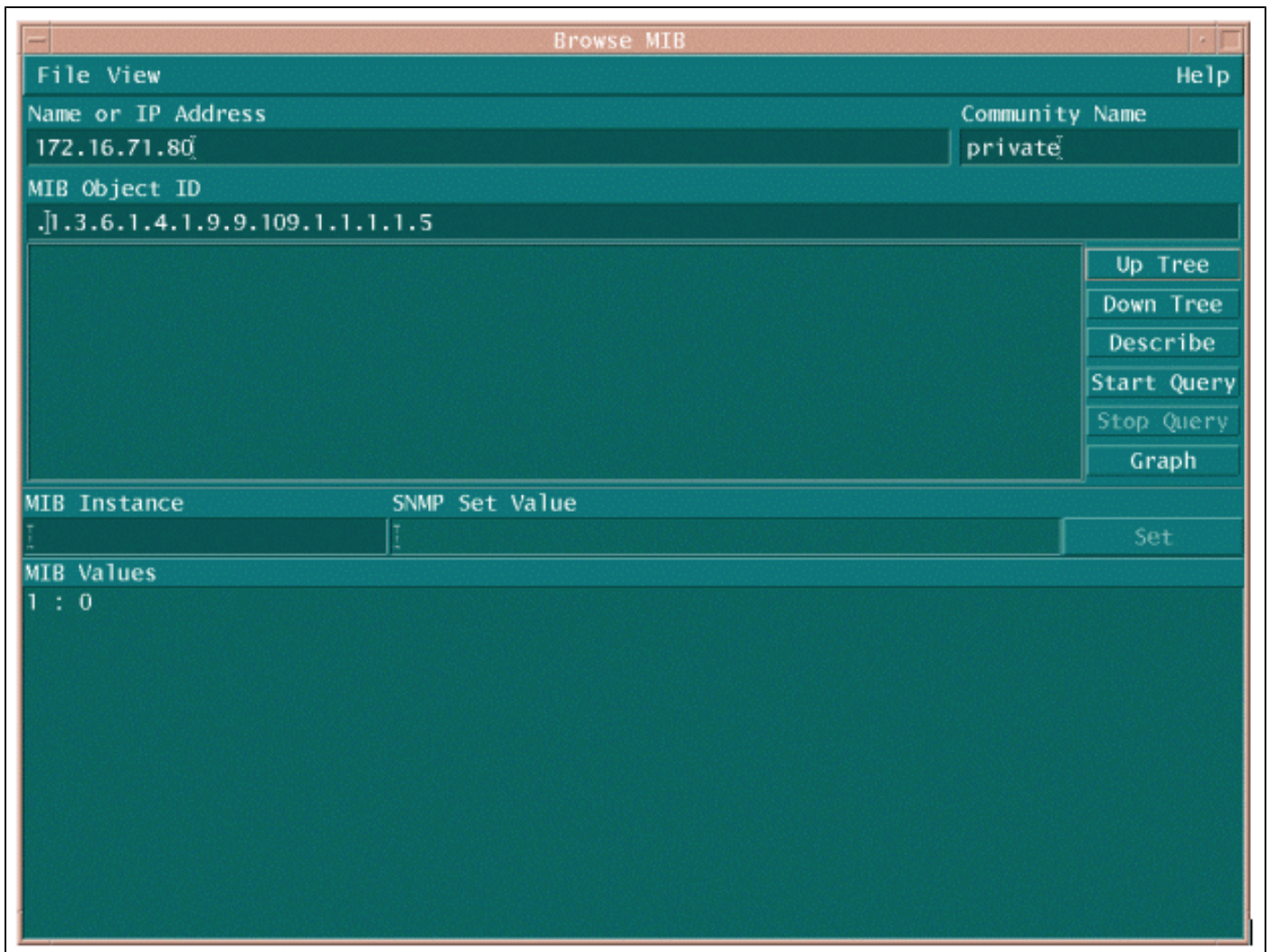
```
<#root>
```

```
ftp://ftp.cisco.com/pub/mibs/oid/CISCO-PROCESS-MIB.oid
### THIS FILE WAS GENERATED BY MIB2SCHEMA
"org" "1.3"
"dod" "1.3.6"
"internet" "1.3.6.1"
"directory" "1.3.6.1.1"
"mgmt" "1.3.6.1.2"
"experimental" "1.3.6.1.3"
"private" "1.3.6.1.4"
"enterprises" "1.3.6.1.4.1"
"cisco" "1.3.6.1.4.1.9"
"ciscoMgmt" "1.3.6.1.4.1.9.9"
"ciscoProcessMIB" "1.3.6.1.4.1.9.9.109"
"ciscoProcessMIBObjects" "1.3.6.1.4.1.9.9.109.1"
"ciscoProcessMIBNotifications" "1.3.6.1.4.1.9.9.109.2"
"ciscoProcessMIBConformance" "1.3.6.1.4.1.9.9.109.3"
"cpmCPU" "1.3.6.1.4.1.9.9.109.1.1"
"cpmProcess" "1.3.6.1.4.1.9.9.109.1.2"
"cpmCPUTotalTable" "1.3.6.1.4.1.9.9.109.1.1.1"
"cpmCPUTotalEntry" "1.3.6.1.4.1.9.9.109.1.1.1.1"
"cpmCPUTotalIndex" "1.3.6.1.4.1.9.9.109.1.1.1.1.1"
"cpmCPUTotalPhysicalIndex" "1.3.6.1.4.1.9.9.109.1.1.1.1.2"
"cpmCPUTotal5sec" "1.3.6.1.4.1.9.9.109.1.1.1.1.3"
"cpmCPUTotal1min" "1.3.6.1.4.1.9.9.109.1.1.1.1.4"
```

```
"cpmCPUTotal5min" "1.3.6.1.4.1.9.9.109.1.1.1.1.5"
```

有兩種常用方法可以輪詢MIB OID以確保其可用和正常工作。最好在開始批次資料收集之前執行此操作，這樣您就不必浪費時間輪詢不存在的對象，從而不會導致資料庫為空。其中一種方法是使用來自NMS平台(例如HP OpenView Network Node Manager (NNM)或CiscoWorks Windows)的MIB跟蹤器，並輸入要檢查的OID。

以下是HP OpenView SNMP MIB walker的一個示例。



輪詢MIB OID的另一種簡單方法是使用UNIX命令snmpwalk，如下例所示。

```
nsahpov6% cd /opt/OV/bin  
nsahpov6% snmpwalk -v1 -c private 172.16.71.80 .1.3.6.1.4.1.9.9.109.1.1.1.1.5.1
```

```
cisco.ciscoMgmt.ciscoProcessMIB.ciscoProcessMIBObjects.cpmCPU.cpmCPUTotalTable.cpmCPUTotalEntry.cpmCPUT
```

在這兩個示例中，MIB都返回值0，這意味著對於該輪詢週期，CPU平均使用率為0%。如果難以讓裝置使用正確的資料做出響應，請嘗試對裝置執行ping操作並透過Telnet訪問裝置。如果仍有問題

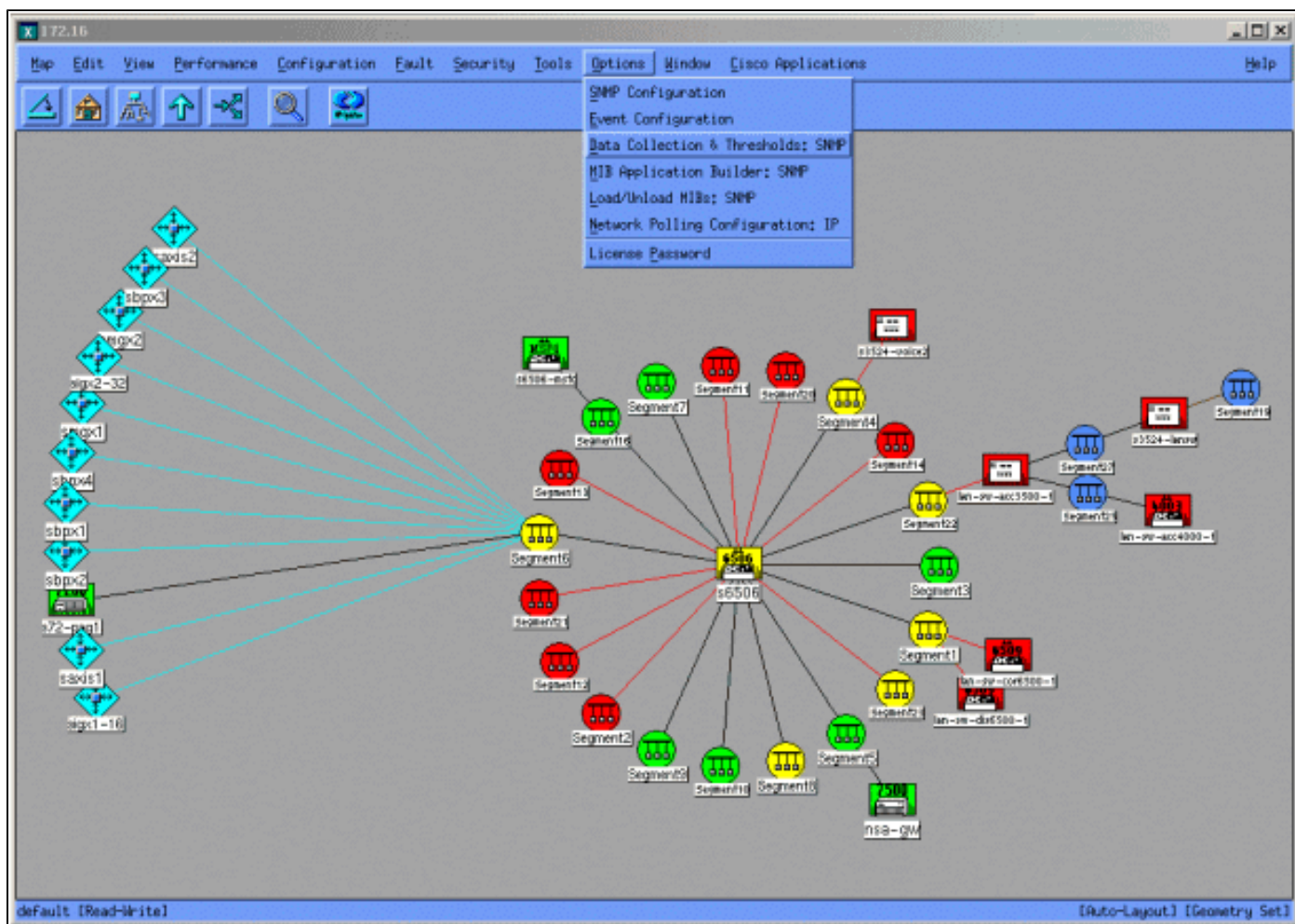
，請檢查SNMP配置和SNMP社群字串。您可能需要尋找替代的MIB或其他版本的IOS才能使之運作

第3步：輪詢和記錄來自路由器的特定SNMP MIB對象

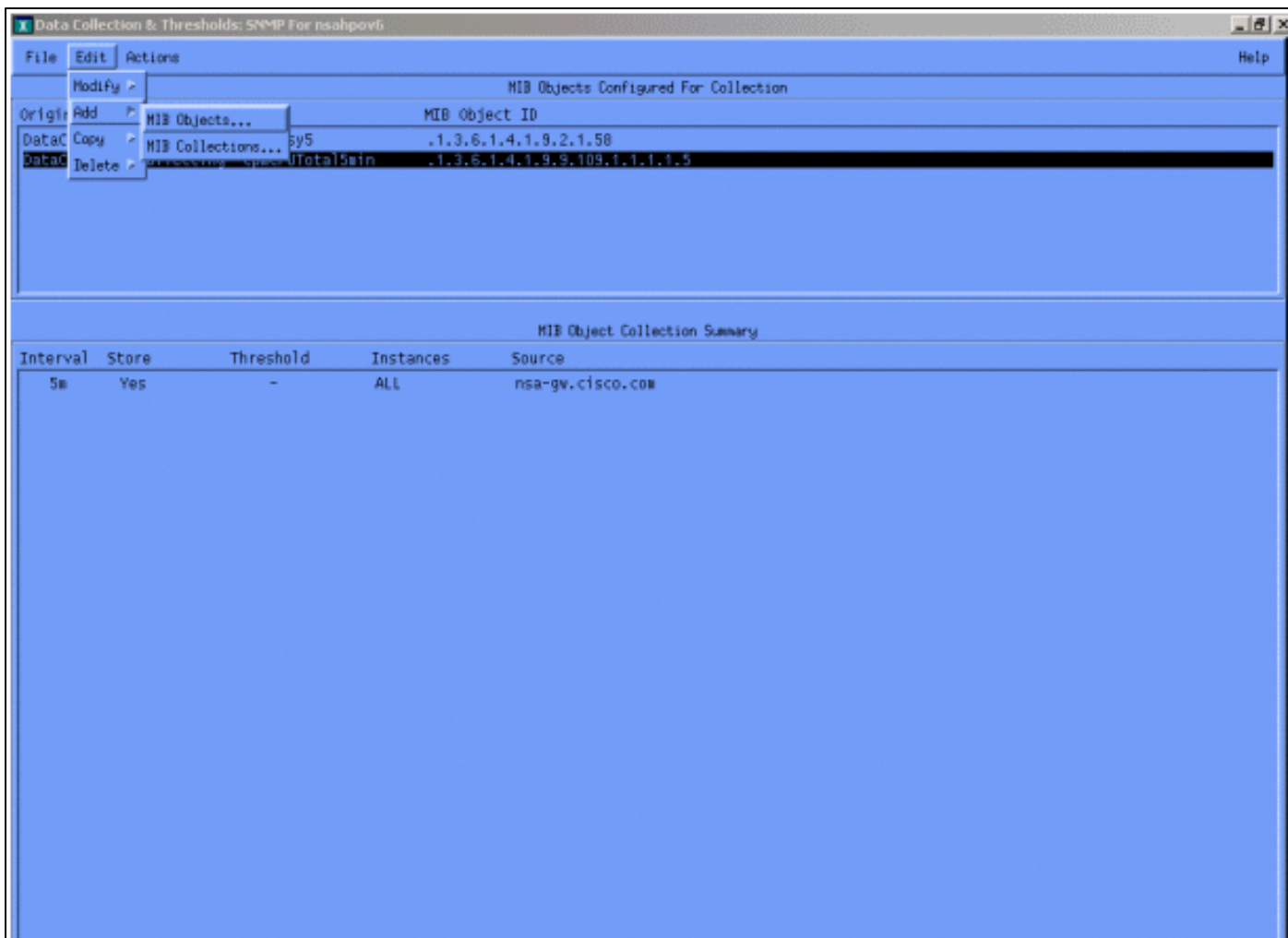
有多種方法可以輪詢MIB對象並記錄輸出。提供現成產品、共享軟體產品、指令碼和供應商工具。所有前端工具都使用SNMP get 進程獲取資訊。主要的差異在於組態的彈性以及在資料庫中記錄資料的方式。再次檢視處理器MIB，瞭解這些不同方法的運作方式。

現在您知道路由器支援OID，您需要決定輪詢它的頻率以及如何記錄。Cisco建議每隔五分鐘輪詢CPU MIB。更低的間隔會增加網路或裝置上的負載，而且由於MIB值是五分鐘平均值，因此輪詢它比平均值更頻繁是沒有用的。通常也建議基線輪詢至少要有兩個星期，以便您能夠分析網路上至少兩個每週的業務週期。

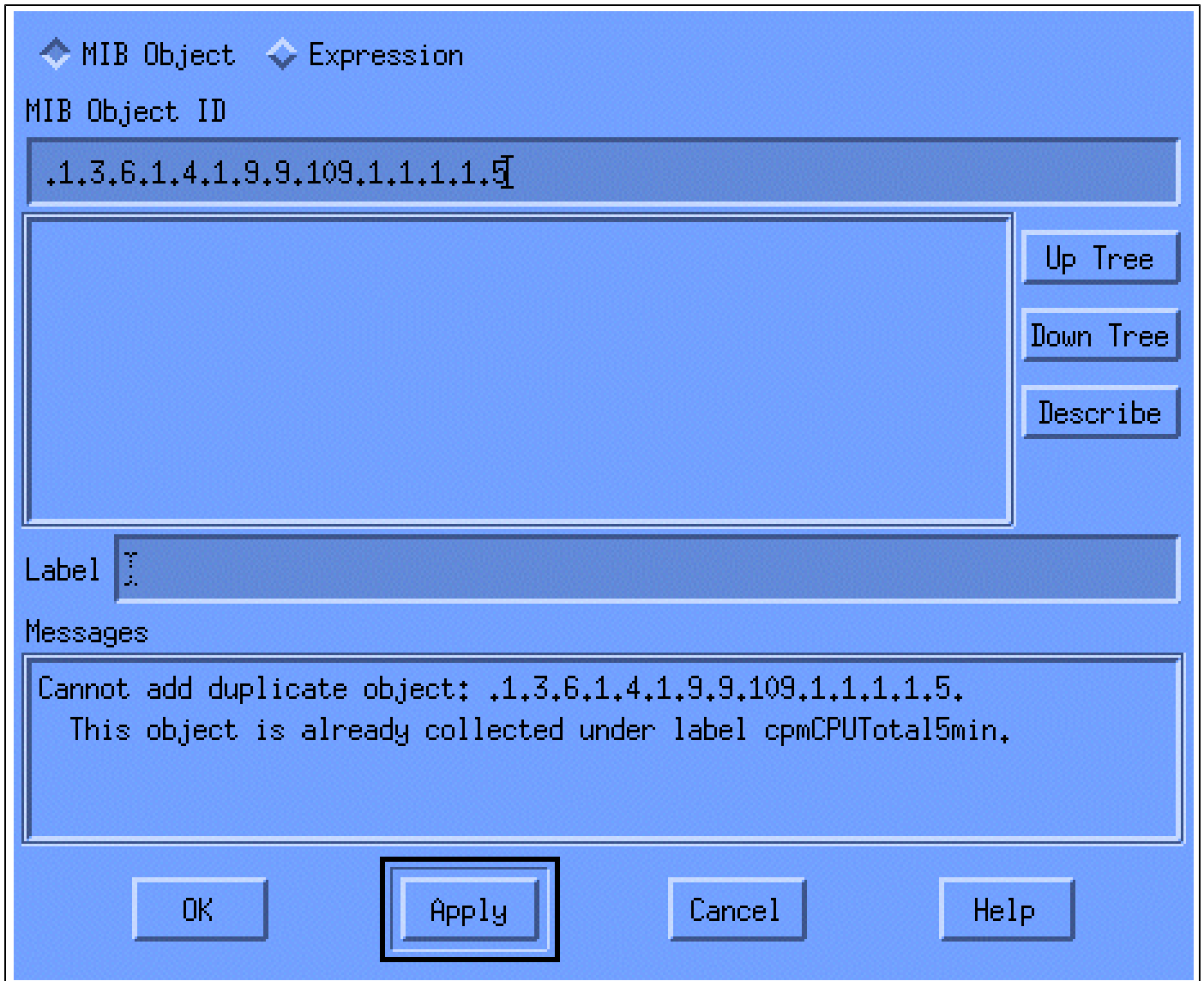
以下螢幕顯示如何使用HP OpenView Network Node Manager 6.1版增加MIB對象。從主螢幕中選擇選項>資料收集與閾值。



然後選擇Edit > Add > MIB Objects。

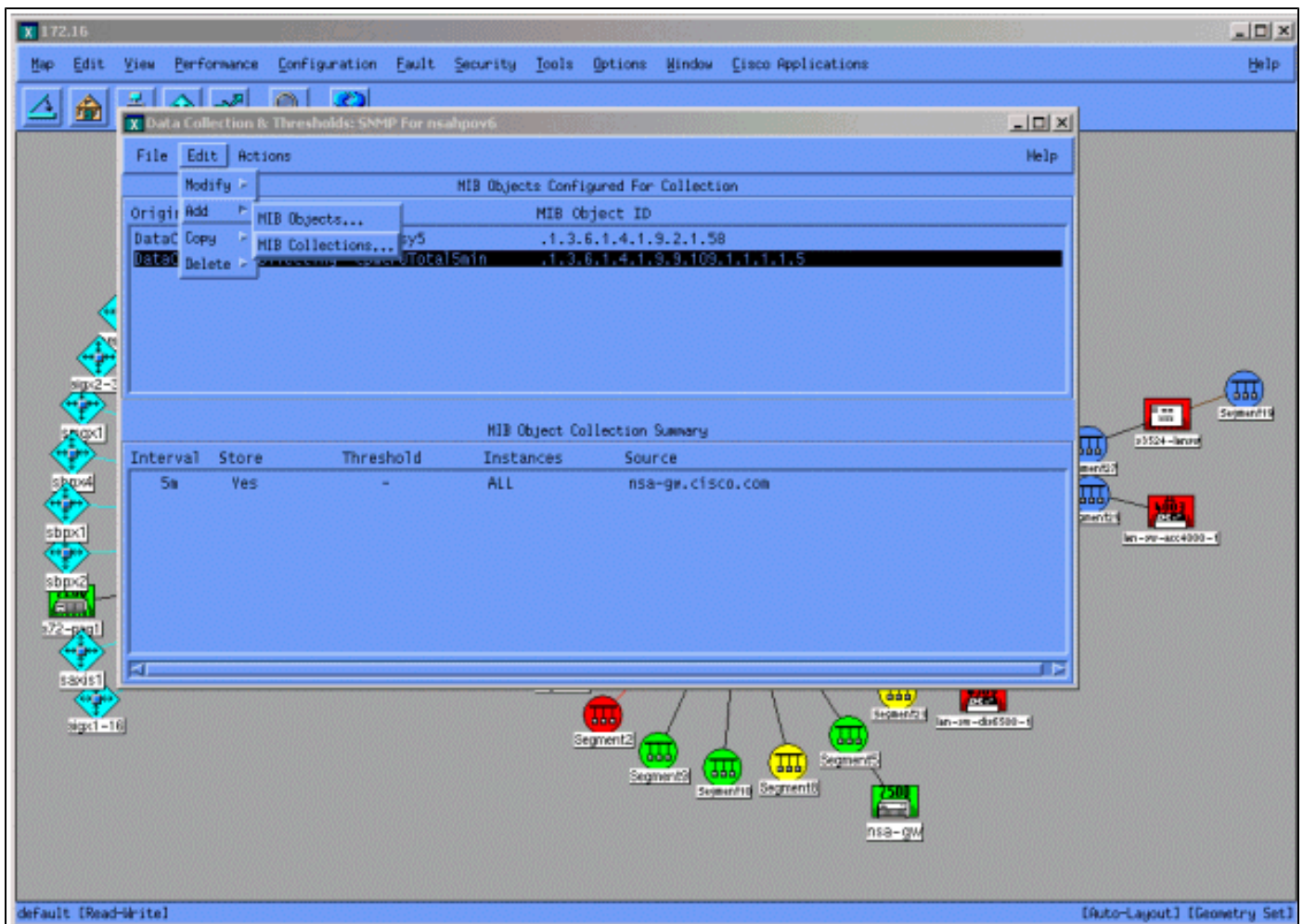


從選單中增加OID字串，然後按一下Apply。您現在已將MIB對象輸入到HP OpenView平台，以便對其進行輪詢。



接下來，您必須讓HP OpenView知道要為此OID輪詢的路由器。

從Data Collection選單中，選擇Edit > Add > MIB Collections。



在Source欄位中，輸入要輪詢之路由器的網域命名系統(DNS)名稱或IP位址。

從「Set Collection Mode」清單中選擇Store，No Thresholds。

將輪詢間隔設定為5m，持續5分鐘時間。

按一下「Apply」。

Set Collection Mode Store, No Thresholds

List Of Collection Sources

10.0.0.10 Add From Map

Delete

Delete All

Source Add

Instances: All

Only Collect On Sources With sysObjectIDs:

Create Event When SNMP Request Fails: 58720266

Polling Interval 5m

Threshold > 0 For 1 Consecutive Samples

Percent Of Threshold

Rearm = 0 absolute For 1 Consecutive Samples

Threshold Event Number 58720266

Configure Threshold Event... Configure Rearm Event...

OK Apply Cancel Help

要使更改生效，必須選擇檔案>儲存。

要驗證收集設定是否正確，請突出顯示路由器的收集彙總行，然後選擇Actions > Test SNMP。這將檢查社群字串是否正確並將輪詢OID的所有例項。

```
Starting SNMP test for all instances on nsa-gw.cisco.com.  
Checking MIB .1.3.6.1.4.1.9.9.109.1.1.1.1.5:
```

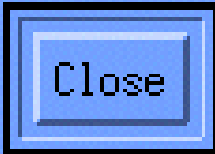
```
.1.3.6.1.4.1.9.9.109.1.1.1.1.5 (instance 1): 0  
.1.3.6.1.4.1.9.9.109.1.1.1.1.5 (instance 2): 1  
.1.3.6.1.4.1.9.9.109.1.1.1.1.5 (instance 3): 1
```

```
Tested all instances.
```

```
Instances which will be collected:
```

```
1 2 3
```

```
All instances will be collected.
```

A rectangular button with a double border and the word "Close" centered inside.

按一下Close，讓收集運行一週。在每週週期結束時，擷取資料進行分析。

如果您將資料傾印至ASCII檔案，並將其匯入試算表工具（例如Microsoft Excel），則更容易分析資料。要使用HP OpenView NNM執行此操作，您可以使用命令列工具snmpColDump。配置的每個集合都會寫入/var/opt/OV/share/databases/snmpCollect/目錄中的檔案。

使用以下命令將資料提取到名為testfile的ASCII檔案中：

```
<#root>
```

```
snmpColDump /var/opt/OV/share/databases/snmpCollect/cpmCPUTotal5min.1 >
```

```
testfile
```

注意：cpmCPUTotal5min.1是HP OpenView NNM在OID輪詢開始時建立的資料庫檔案。

生成的測試檔案類似於以下示例。

```
03/01/2001 14:09:10 nsa-gw.cisco.com 1  
03/01/2001 14:14:10 nsa-gw.cisco.com 1  
03/01/2001 14:19:10 nsa-gw.cisco.com 1  
03/01/2001 14:24:10 nsa-gw.cisco.com 1
```

```
03/01/2001 14:29:10 nsa-gw.cisco.com 1
03/01/2001 14:34:10 nsa-gw.cisco.com 1
03/01/2001 14:39:10 nsa-gw.cisco.com 1
03/01/2001 14:44:10 nsa-gw.cisco.com 1
03/01/2001 14:49:10 nsa-gw.cisco.com 1
03/01/2001 14:54:10 nsa-gw.cisco.com 1
03/01/2001 14:59:10 nsa-gw.cisco.com 1
03/.....
```

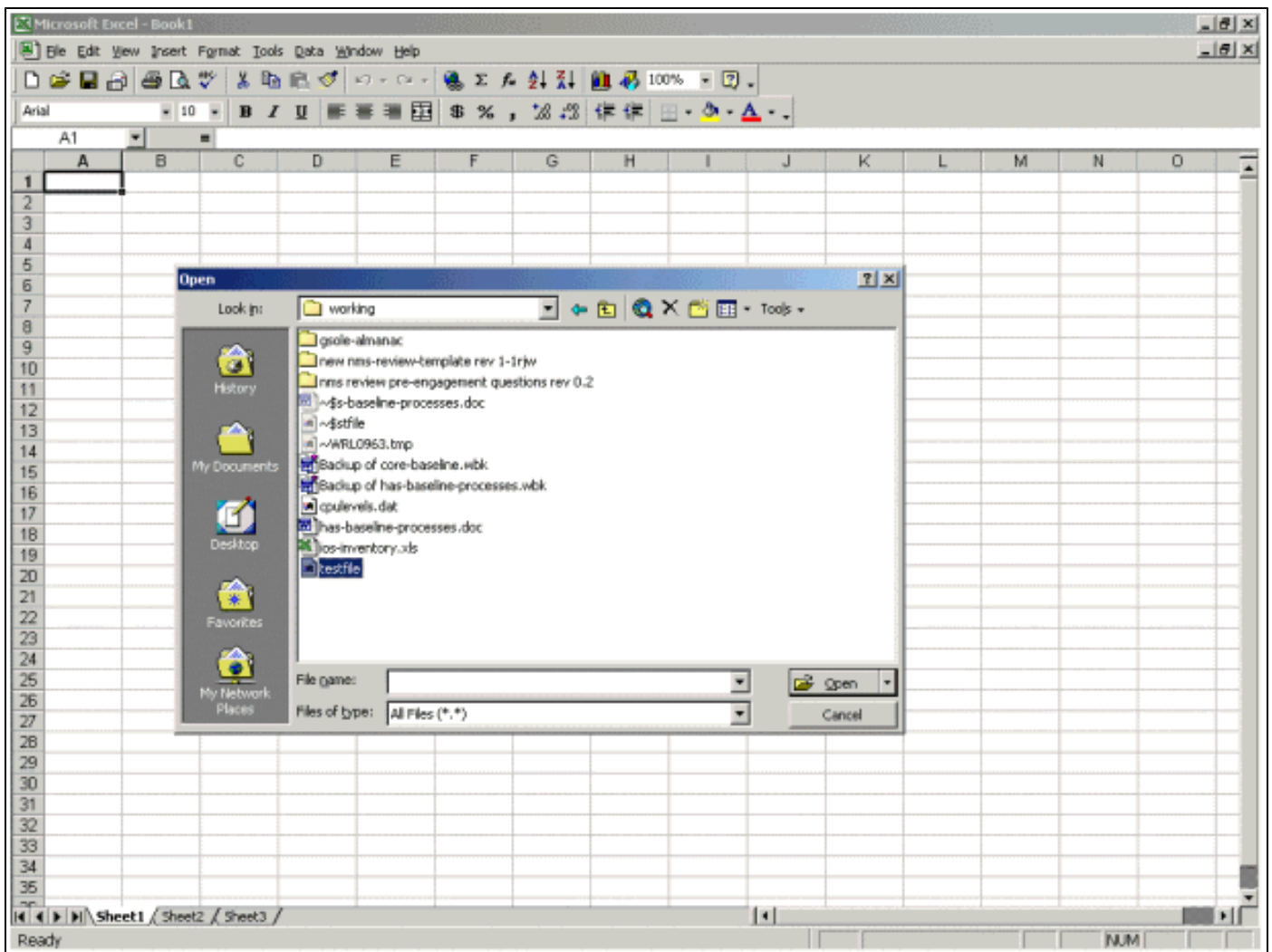
一旦測試檔案輸出位於您的UNIX月台上，您就可以使用檔案傳輸通訊協定(FTP)將它傳輸到PC上。

您也可以使用自己的指令碼收集資料。為此，請每五分鐘對CPU OID執行snmpget，並將結果轉儲到.csv檔案中。

步驟4：分析資料以確定閾值

現在有了些資料，就可以開始分析了。基準線的這個階段會決定您可以使用的臨界值設定，這些設定是效能或故障的精確量度，並且在您開啟臨界值監控時，不會觸發太多警示。最簡單的方法之一是將資料導入電子表格（如Microsoft Excel）並繪製散點圖。這種方法使您可以很容易地瞭解特定裝置在監控某個特定閾值的情況下會建立多少次異常警報。建議不要在未執行基線的情況下打開閾值，因為這樣可能會從超過所選閾值的裝置建立警報風暴。

若要將測試檔案匯入Excel試算表，請開啟Excel並選取檔案>開啟，然後選取您的資料檔案。



然後，Excel應用程式會提示您匯入檔案。

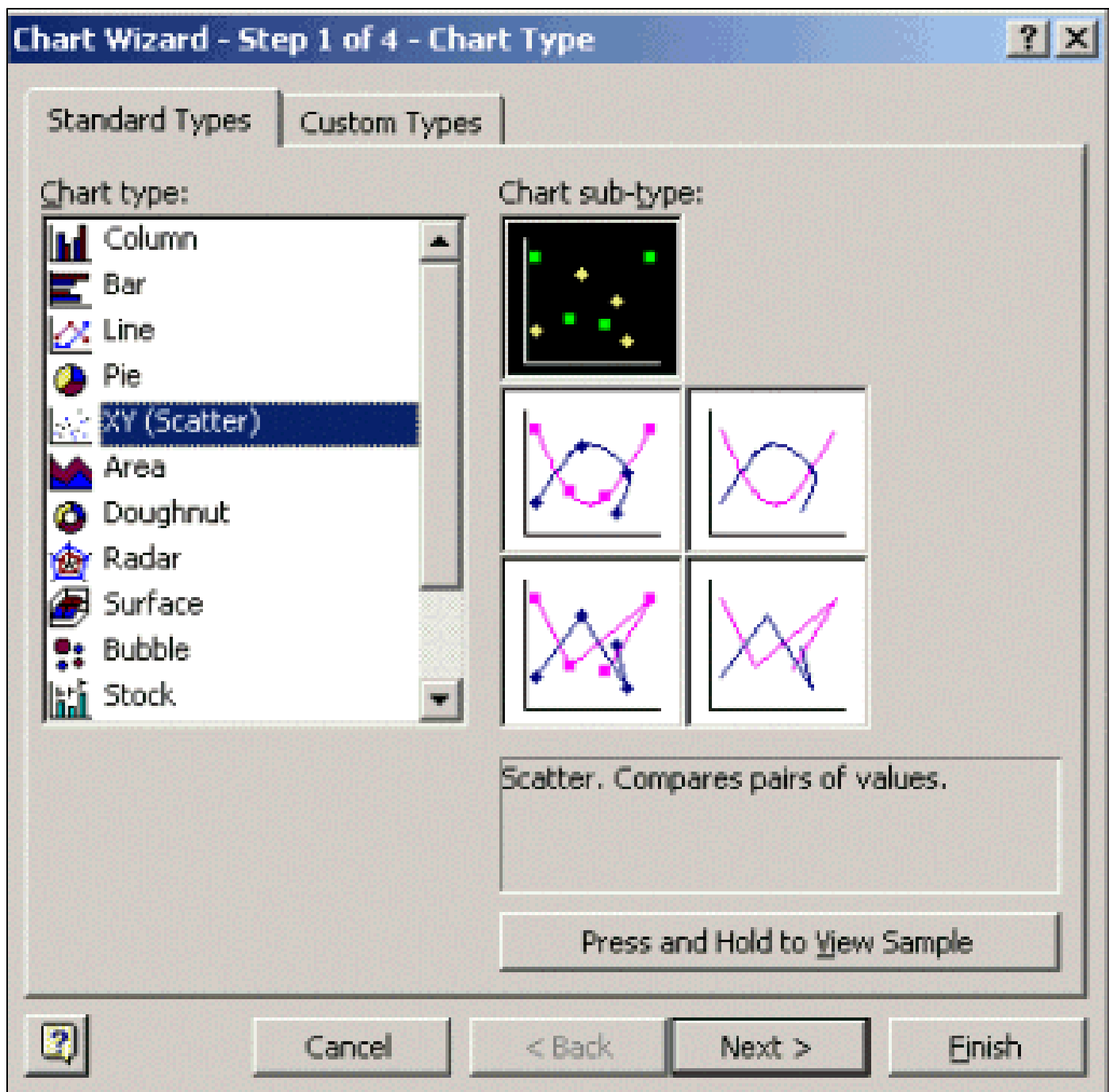
完成時，匯入的檔案看起來應該與下列螢幕類似。

	A	B	C	D	E	F	G	H	I	J	K	L
1	Wed Oct 11 12:52:23 PDT 2000	crflsbgb001	23									
2	Wed Oct 11 12:57:17 PDT 2000	crflsbgb001	22									
3	Wed Oct 11 13:00:05 PDT 2000	crflsbgb001	23									
4	Wed Oct 11 13:05:05 PDT 2000	crflsbgb001	24									
5	Wed Oct 11 13:10:04 PDT 2000	crflsbgb001	23									
6	Wed Oct 11 13:15:05 PDT 2000	crflsbgb001	23									
7	Wed Oct 11 13:20:04 PDT 2000	crflsbgb001	24									
8	Wed Oct 11 13:25:05 PDT 2000	crflsbgb001	25									
9	Wed Oct 11 13:30:05 PDT 2000	crflsbgb001	25									
10	Wed Oct 11 13:35:05 PDT 2000	crflsbgb001	23									
11	Wed Oct 11 13:40:04 PDT 2000	crflsbgb001	26									
12	Wed Oct 11 13:45:05 PDT 2000	crflsbgb001	23									
13	Wed Oct 11 13:50:05 PDT 2000	crflsbgb001	22									
14	Wed Oct 11 14:00:05 PDT 2000	crflsbgb001	21									
15	Wed Oct 11 14:05:05 PDT 2000	crflsbgb001	20									
16	Wed Oct 11 14:10:05 PDT 2000	crflsbgb001	20									
17	Wed Oct 11 14:15:04 PDT 2000	crflsbgb001	20									
18	Wed Oct 11 14:20:05 PDT 2000	crflsbgb001	20									
19	Wed Oct 11 14:25:04 PDT 2000	crflsbgb001	19									
20	Wed Oct 11 14:30:06 PDT 2000	crflsbgb001	18									
21	Wed Oct 11 14:35:04 PDT 2000	crflsbgb001	18									
22	Wed Oct 11 14:40:05 PDT 2000	crflsbgb001	17									
23	Wed Oct 11 14:45:05 PDT 2000	crflsbgb001	17									
24	Wed Oct 11 14:50:04 PDT 2000	crflsbgb001	17									
25	Wed Oct 11 15:00:04 PDT 2000	crflsbgb001	29									
26	Wed Oct 11 15:05:04 PDT 2000	crflsbgb001	36									
27	Wed Oct 11 15:10:05 PDT 2000	crflsbgb001	38									
28	Wed Oct 11 15:15:05 PDT 2000	crflsbgb001	41									
29	Wed Oct 11 15:20:05 PDT 2000	crflsbgb001	42									
30	Wed Oct 11 15:25:05 PDT 2000	crflsbgb001	39									
31	Wed Oct 11 15:30:05 PDT 2000	crflsbgb001	36									
32	Wed Oct 11 15:35:05 PDT 2000	crflsbgb001	31									
33	Wed Oct 11 15:40:05 PDT 2000	crflsbgb001	28									
34	Wed Oct 11 15:45:05 PDT 2000	crflsbgb001	27									
35	Wed Oct 11 15:50:06 PDT 2000	crflsbgb001	25									
36	Wed Oct 11 15:55:06 PDT 2000	crflsbgb001	25									

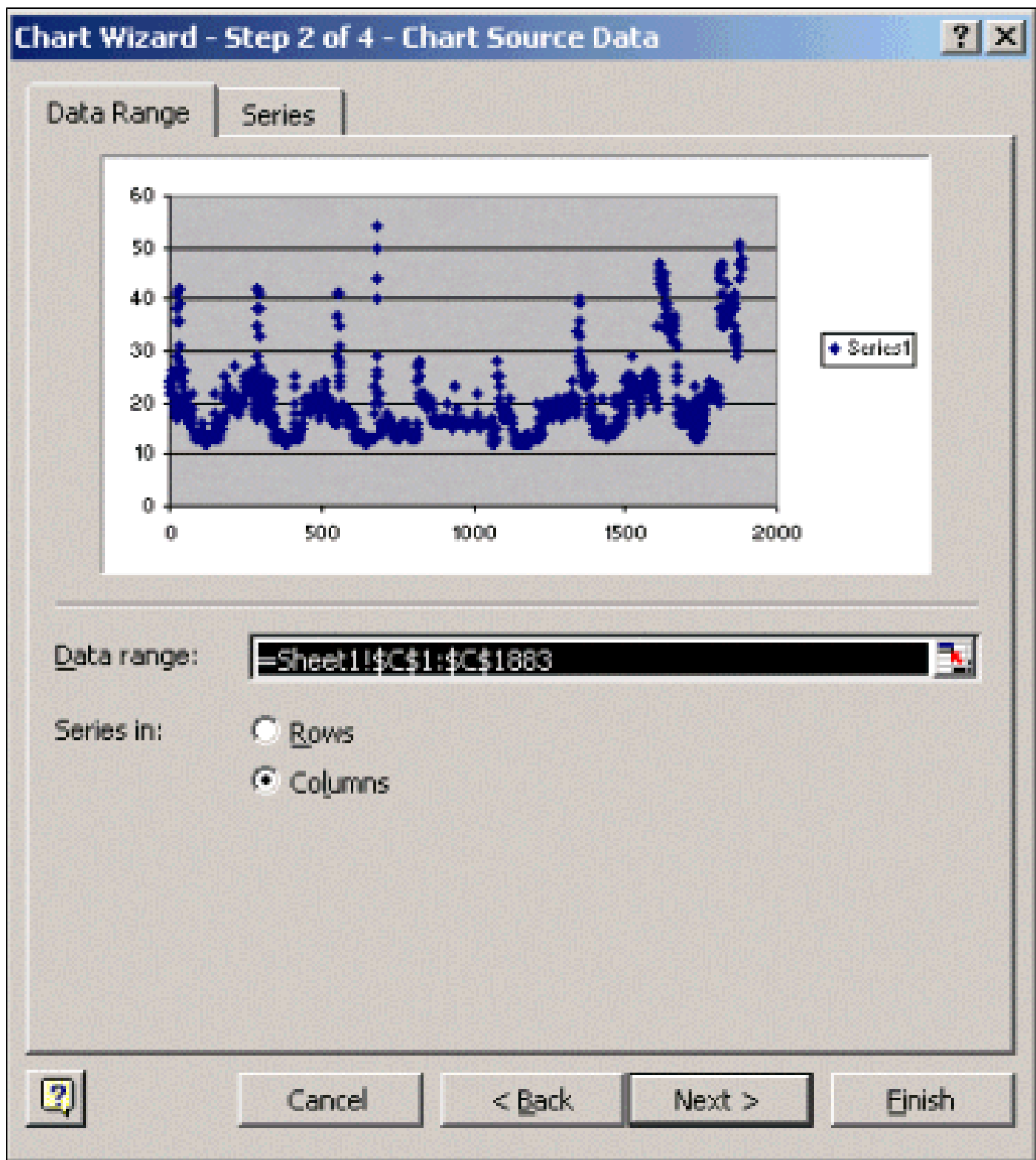
使用散點圖可以更輕鬆地檢視各種閾值設定在網路上的工作方式。

若要建立散佈圖，請反白匯入檔案中的C欄，然後按一下「圖表精靈」圖示。然後按照圖表嚮導中的步驟建立散點圖。

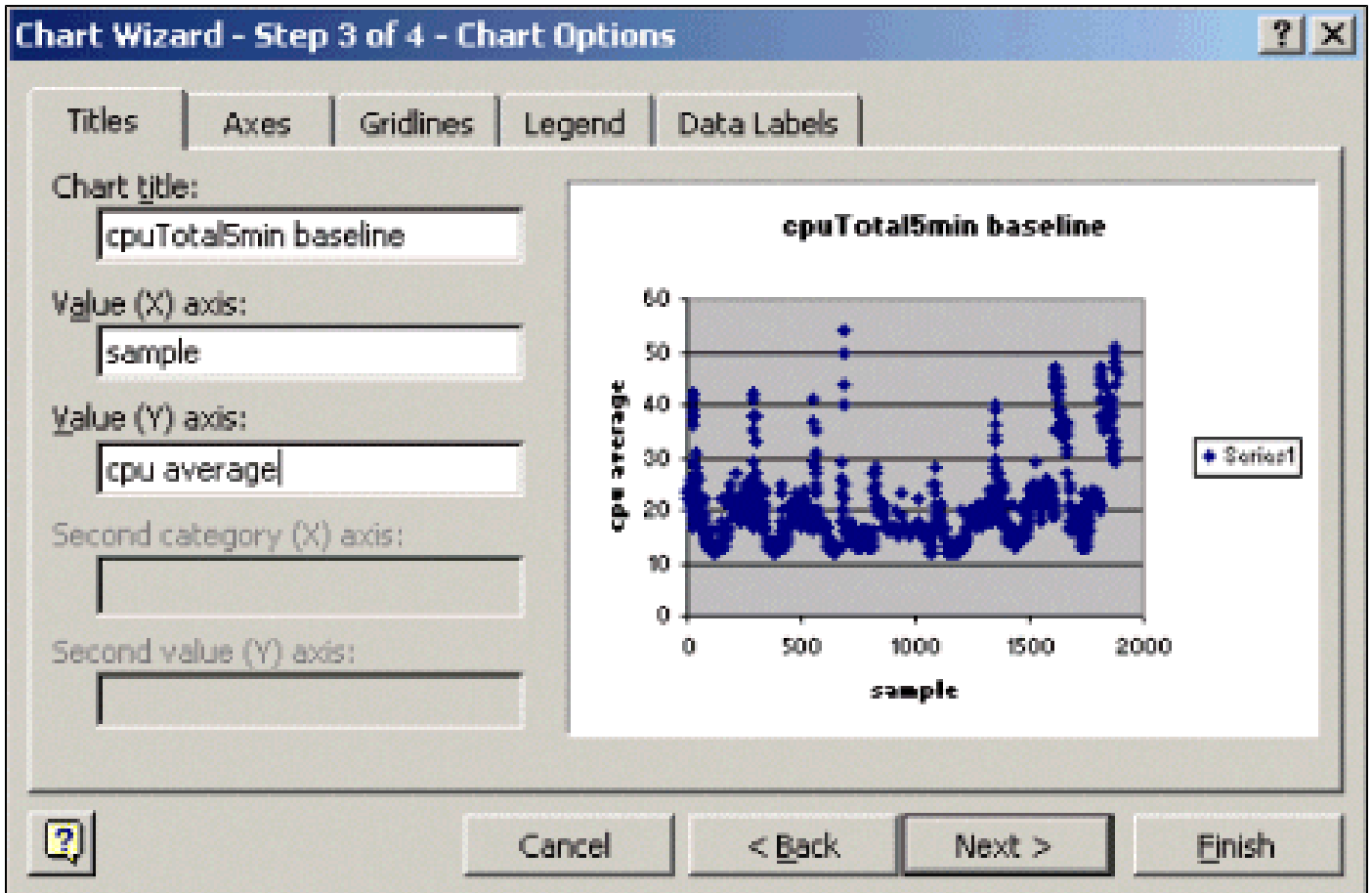
在「圖表精靈」步驟1中（如下所示），選取標準型別標籤，然後選取XY（散點圖）圖表型別。然後按一下Next。



在「圖表精靈」步驟2中（如下所示），選取資料範圍頁籤，然後選取資料範圍和欄選項。按「Next」（下一步）。



在圖表嚮導第3步中（如下所示），輸入圖表標題以及X和Y軸值，然後按一下Next。



在「圖表精靈」步驟4中，選取您要將散佈圖放在新頁面上，或是作為現有頁面中的物件。

按一下完成將圖表放置在所需的位置。

「如果？」分析

您現在可以使用散佈圖進行分析。不過，在繼續之前，您需要詢問以下問題：

- 供應商（在本例中供應商是思科）建議使用什麼作為此MIB變數的閾值？

一般而言，Cisco建議核心路由器不要超過60%的平均CPU使用率。之所以選擇60%，是因為路由器需要一些開銷，以防遇到問題或網路出現故障。思科估計，核心路由器需要大約40%的CPU開銷，以防路由協定必須重新計算或重新收斂。這些百分比視您使用的通訊協定，以及網路的拓撲和穩定性而有所不同。

- 如果使用60%作為閾值設定呢？

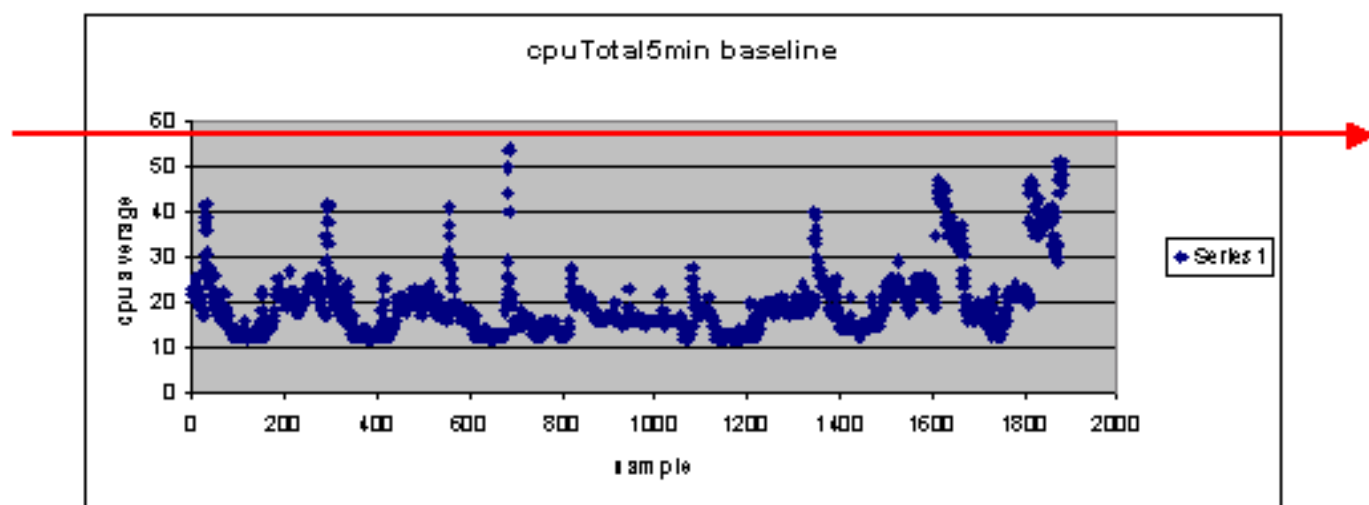
如果您在60的水準方向繪製一條橫跨散佈圖的線，您將會看到沒有任何資料點超過60%的CPU使用率。因此，在網路管理系統(NMS)站點上設定的閾值60不會在輪詢期間觸發閾值警報。此路由器可以接受百分比60。但是，請注意散佈圖中的部分資料點接近60。瞭解路由器何時接近60%的閾值是一件好事，這樣您就可以提前知道CPU接近60%，並且制定計畫來在到達該閾值時採取什麼措施。

- 如果我把門檻設為50%呢？

估計此路由器在此輪詢週期中四次達到50%的利用率，並且每次都會生成閾值警報。當您檢視

路由器組以瞭解不同的閾值設定會執行什麼操作時，此過程變得更加重要。例如，「如果我將整個核心網路的閾值設定為50%會怎樣？」要知道，只選擇一個數字是非常困難的。

CPU閾值「假設」分析



一個讓此過程更輕鬆的策略是「就緒」、「設定」、「開始」閾值方法。此方法連續使用三個閾值數字。

- 就緒—您設定的閾值，作為未來可能需要注意的裝置的預測指標
- Set -用作早期指示器的閾值，可提醒您開始計畫修復、重新配置或升級
- Go -您和/或供應商認為屬於故障狀況並需要一些操作才能修復的閾值；在本例中為60%

下表顯示「就緒」、「設定」、「執行」策略的策略。

閾值	動作	結果
45%	進一步調查	行動計畫選項清單
50%	制定行動計畫	行動計畫中的步驟清單
60%	實施行動計畫	路由器不再超過閾值。返回就緒模式

「就緒」、「設定」、「執行」方法會變更先前討論的原始基準線圖表。下圖顯示變更後的基準線圖表。如果您能辨識圖表上的其他交點，您現在有比以前更多的時間來規劃和反應。

Network break point

Network Warning

Network prediction

Network load

50
100

請注意，在此過程中，注意重點是網路中的例外，而不是其他裝置。我們假定，只要裝置低於閾值，它們就可以正常工作。

如果您從一開始就仔細思考這些步驟，您將為保持網路健康做好充分的準備。執行此型別的計畫對於預算計畫也非常有用。如果您知道您的前5個go路由器、中間集路由器和底部ready路由器是什麼，那麼您可以根據路由器的型別和您的行動計畫選項，輕鬆規劃升級所需的預算。相同的策略可用於廣域網(WAN)鏈路或任何其他MIB OID。

第5步：解決已確定的立即問題

這是基準線流程中比較容易的部份之一。一旦您確定了哪些裝置超出了go閾值，您應制定行動計畫，以使這些裝置重新回到閾值之下。

您可以向思科技術支援中心(TAC)提交支援請求，也可以聯絡您的系統工程師瞭解可用選項。你不應該想當然地認為，讓事情回到閾值之下會花掉你的錢。更改配置以確保所有進程以最有效的方式運行，可以解決某些CPU問題。例如，由於封包經過路由器的路徑，某些存取控制清單(ACL)可能會使路由器CPU執行率非常高。在某些情況下，您可以實施NetFlow交換來更改資料包交換路徑並減少ACL對CPU的影響。無論存在什麼問題，都有必要在此步驟中將所有路由器恢復到閾值之下，這樣您以後就可以實施閾值，而不會有使用過多的閾值警報泛洪NMS工作站的風險。

步驟6：測試臨界值監督

此步驟涉及使用將在生產網路中使用的工具在實驗室中測試閾值。監控閾值有兩種常用方法。您必須決定哪種方法最適合您的網路。

- 使用SNMP平台或其他SNMP監控工具的輪詢和比較方法

此方法會使用更多的網路頻寬來輪詢流量，並佔用您SNMP平台上的處理週期。

- 在路由器中使用遠端監控(RMON)警報和事件配置，以便只有在超出閾值時才會傳送警報

此方法可以減少網路頻寬使用，但也會增加路由器上的記憶體和CPU使用率。

使用SNMP實施閾值

要使用HP OpenView NNM設定SNMP方法，請像設定初始輪詢時那樣選擇Options > Data Collection & Thresholds。不過，這次請在收集選單中選擇儲存，檢查閾值而不是儲存，無閾值。設定閾值後，可以透過傳送多個ping和/或多個SNMP路徑來提高路由器上的CPU利用率。如果您無法強制CPU達到足夠高的臨界值，您可能必須降低臨界值。無論如何，您應確保閾值機制正常工作。

使用此方法的一個限制是無法同時實現多個閾值。您需要三個SNMP平台來設定三個不同的同時閾值。諸如[Concord Network Health](#) 和 [Trinagy TREND](#) 等工具可以為同一OID例項允許多個閾值。

如果您的系統一次只能處理一個臨界值，您可以考慮序列式的「就緒」、「設定」、「執行」策略。也就是說，在持續達到ready閾值時，應開始進行調查，並將該裝置的閾值提升至所設定的級別。持續達到設定層級時，開始制定行動計畫，並將該裝置的臨界值提升至執行層級。然後，當持續達到執行閾值時，實施您的行動計畫。這應該與三個同時閾值方法一樣有效。更改SNMP平台閾值設定只需要一點時間。

使用RMON警報和事件實現閾值

使用RMON警報和事件配置，您可以讓路由器自己監控多個閾值。當路由器檢測到閾值過高時，它會向SNMP平台傳送SNMP陷阱。您必須在路由器配置中設定SNMP陷阱接收器，才能轉發陷阱。警報和事件之間存在關聯。警報會檢查給定閾值的OID。如果達到閾值，警報進程將觸發事件進程，該進程可以傳送SNMP陷阱消息、建立RMON日誌條目或同時傳送這兩者。有關此命令的詳細資訊，請參閱[RMON警報和事件配置命令](#)。

以下路由器配置命令使路由器每300秒監控一次cpmCPUTotal5min。如果CPU超過60%，它將觸發事件1；當CPU回落到40%時，它將觸發事件2。在這兩種情況下，SNMP陷阱消息都將使用community private string傳送到NMS工作站。

若要使用Ready、Set、Go方法，請使用下列所有組態陳述式。

```
rmon event 1 trap private description "cpu hit60%" owner jharp
rmon event 2 trap private description "cpu recovered" owner jharp
rmon alarm 10 cpmCPUTotalTable.1.5.1 300 absolute rising 60 1 falling 40 2 owner jharp
```

```
rmon event 3 trap private description "cpu hit50%" owner jharp
rmon event 4 trap private description "cpu recovered" owner jharp
rmon alarm 20 cpmCPUTotalTable.1.5.1 300 absolute rising 50 3 falling 40 4 owner jharp
```

```
rmon event 5 trap private description "cpu hit 45%" owner jharp
rmon event 6 trap private description "cpu recovered" owner jharp
rmon alarm 30 cpmCPUTotalTable.1.5.1 300 absolute rising 45 5 falling 40 6 owner jharp
```

以下示例顯示以上語句配置的show rmon alarm命令的輸出。

```
<#root>
zack#
sh rmon alarm
```

```
Alarm 10 is active, owned by jharp
Monitors cpmCPUTotalTable.1.5.1 every 300 second(s)
Taking absolute samples, last value was 0
Rising threshold is 60, assigned to event
1
Falling threshold is 40, assigned to event
2
On startup enable rising or falling alarm
Alarm 20 is active, owned by jharp
Monitors cpmCPUTotalTable.1.5.1 every 300 second(s)
Taking absolute samples, last value was 0
Rising threshold is 50, assigned to event
3
Falling threshold is 40, assigned to event
4
On startup enable rising or falling alarm
Alarm 30 is active, owned by jharp
Monitors cpmCPUTotalTable.1.5.1 every 300 second(s)
Taking absolute samples, last value was 0
Rising threshold is 45, assigned to event
5
Falling threshold is 40, assigned to event
6
On startup enable rising or falling alarm
```

以下示例展示show rmon event命令的输出。

```
<#root>
```

```
zack#
```

```
sh rmon event
```

```
Event 1 is active, owned by jharp
Description is cpu hit60%
Event firing causes trap to community
private, last fired 00:00:00
Event 2 is active, owned by jharp
Description is cpu recovered
Event firing causes trap to community
private, last fired 02:40:29
Event 3 is active, owned by jharp
Description is cpu hit50%
Event firing causes trap to community
private, last fired 00:00:00
Event 4 is active, owned by jharp
Description is cpu recovered
Event firing causes trap to community
private, last fired 00:00:00
Event 5 is active, owned by jharp
Description is cpu hit 45%
Event firing causes trap to community
private, last fired 00:00:00
Event 6 is active, owned by jharp
Description is cpu recovered
Event firing causes trap to community
private, last fired 02:45:47
```


您可能想嘗試這兩種方法，以瞭解哪一種方法最適合您的環境。您甚至可能會發現方法組合非常有效。在任何情況下，都應在實驗室環境中進行測試，以確保一切工作正常。在實驗室中測試之後，只需在一小群路由器上進行有限部署，即可測試向運營中心傳送警報的過程。

在這種情況下，您必須降低閾值以測試進程：不建議人為提高生產路由器上的CPU。您還應確保當警報進入運營中心的NMS工作站時，有一個上報策略，以確保在裝置超出閾值時通知您。這些配置已在實驗室中使用Cisco IOS版本12.1(7)進行了測試。如果您遇到任何問題，應諮詢思科工程或系統工程師，瞭解您的IOS版本中是否存在Bug。

第7步：使用SNMP或RMON實施閾值監控

在實驗室和有限的部署中徹底測試過閾值監控後，您就可以跨核心網路實施閾值。現在，您可以系統地透過此基線過程瞭解網路上的其他重要MIB變數，例如緩衝區、空間記憶體、循環冗餘檢查(CRC)錯誤、AMT信元丟失等。

如果使用RMON警報和事件配置，則現在可停止從NMS站進行輪詢。這將減少NMS伺服器上的負載，並減少網路上的輪詢資料量。透過系統地執行此過程以獲取重要的網路運行狀況指標，您可以輕鬆達到網路裝置使用RMON警報和事件監控自己的程度。

其他MIB

瞭解此過程後，您可能需要調查其他MIB以做基線和監控。下列小節提供一些OID的簡短清單，以及您可能會覺得有用的描述。

路由器MIB

記憶體特性對於確定路由器的運行狀況非常有用。正常運行的路由器幾乎總是應該有可用的緩衝區空間。如果路由器開始耗盡緩衝空間，CPU將不得不更加努力地建立新的緩衝區，並嘗試為傳入和傳出資料包找到緩衝區。有關緩衝區的深入討論不在本檔案的範圍之內。但是，一般而言，正常路由器的緩衝區遺漏次數應該很少（如果有），而且不應有任何緩衝區故障或記憶體為零的情況。

物件	說明	OID
ciscoMemoryPoolFree	受管裝置上當前未使用的記憶	1.3.6.1.4.1.9.9.48.1.1.1.6

	體池位元組數	
ciscoMemoryPoolLargestFree	記憶體池中當前未使用的最大連續位元組數	1.3.6.1.4.1.9.9.48.1.1.1.7
緩衝未命中	緩衝區元素遺漏的數目	1.3.6.1.4.1.9.2.1.12
bufferFail	緩衝區配置失敗次數	1.3.6.1.4.1.9.2.1.46
bufferNoMem	因	1.3.6.1.4.1.9.2.1.47

	為沒有可用記憶體而造成緩衝區建立失敗的次數	
--	-----------------------	--

Catalyst交換器MIB

物件	說明	OID
cpmCPUTotal5分鐘	過去五分鐘期間的整體CPU忙碌百分比。此對象從OLD-CISCO-SYSTEM-MIB中刪除avgBusy5對象	1.3.6.1.4.1.9.9.109.1.1.1.5
cpmCPUTotal5sec	過去五秒期間的整體CPU忙碌百分比。此物件會從OLD-CISCO-SYSTEM-MIB取代busyPer物件	1.3.6.1.4.1.9.9.109.1.1.1.3

sysTraffic	上一個輪詢 間隔的頻寬 利用率百分 比	1.3.6.1.4.1.9.5.1.1.8
sysTrafficPeak	自上次清除 連線埠計數 器或系統啟 動後的流量 峰值計值	1.3.6.1.4.1.9.5.1.1.19
sysTrafficPeakttime	自峰值流量 計量器值出 現以來的時 間 (以百分 之一秒為單 位)	1.3.6.1.4.1.9.5.1.1.20
portTopNUtilization	系統中埠的 利用率	1.3.6.1.4.1.9.5.1.20.2.1.4
portTopNBufferOverFlow	系統中連線 埠的緩衝區 溢位數目	1.3.6.1.4.1.9.5.1.20.2.1.10

序列連結MIB

物件	說明	OID
loclfInputQueueDrops	因輸入 佇列已 滿而捨 棄的封 包數	1.3.6.1.4.1.9.2.2.1.1.26
loclfOutputQueueDrops	由於輸 出佇列 已滿而 丟棄的 資料包 數	1.3.6.1.4.1.9.2.2.1.1.27
loclfInCRC	發生循 環冗餘 校驗和 錯誤的 輸入資 料包的 數量	1.3.6.1.4.1.9.2.2.1.1.12

RMON警報和事件配置命令

警報

可以使用以下語法配置RMON警報：

<#root>

```
rmmon alarm number variable interval {delta | absolute} rising-threshold value  
      [event-number] falling-threshold value [event-number]  
      [owner string]
```

元素	說明
number	警示號碼，與RMON MIB中alarmTable中的alarmIndex相同。
變數	要監控的MIB對象，轉換為RMON MIB的alarmTable中使用的alarmVariable。
間隔	警報監視MIB變數的時間（秒），該變數與RMON MIB的alarmTable中使用的alarmInterval相同。
增量	測試MIB變數之間的更改，這會影響RMON MIB的alarmTable中的alarmSampleType。
絕對	直接測試每個MIB變數，這會影響RMON MIB的alarmTable中的alarmSampleType。
上升閾值	觸發警報的值。
event-number	（選擇性）當上升或下降臨界值超過其限制時，要觸發的事件編號。此值與RMON MIB的alarmTable中的alarmRisingEventIndex或alarmFallingEventIndex相同。
下降閾值	重設警示的值。
所有者字串	（可選）指定警報的所有者，該所有者與RMON MIB的alarmTable中的alarmOwner相同。

活動

可以使用以下語法配置RMON事件：

<#root>

```
rmon event number [log] [trap community] [description string]
           [owner string]
```

元素	說明
number	指派的事件編號，與RMON MIB中eventTable中的eventIndex相同。
log	(可選) 在觸發事件時生成RMON日誌條目，並將RMON MIB中的eventType設定為log或log-and-trap。
陷阱社群	(可選) 用於此陷阱的SNMP社群字串。將此行的RMON MIB中eventType的設定配置為snmp-trap或log-and-trap。此值與RMON MIB中eventTable中的eventCommunityValue相同。
說明字串	(可選) 指定事件的說明，該說明與RMON MIB的eventTable中的事件說明相同。
所有者字串	(可選) 此事件的所有者，它與RMON MIB的eventTable中的eventOwner相同。

RMON警報和事件實施

有關RMON報警和事件實施的詳細資訊，請閱讀網路管理系統最佳實踐白皮書的[RMON報警和事件實施](#)部分。

相關資訊

- [技術支援與檔案- Cisco Systems](#)

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。