

Risoluzione dei problemi di convergenza BGP lenta a causa di policy di route non ottimali su IOS-XR

Sommario

[Introduzione](#)

[Introduzione](#)

[Problema](#)

[Soluzione](#)

[Verifica](#)

[Conclusioni](#)

Introduzione

Questo documento descrive come diagnosticare i problemi di convergenza BGP lenta sui router Cisco IOS® XR che si verificano a causa di policy di routing non ottimali.

Introduzione

Il tempo di convergenza BGP è costituito da una serie di fattori. Una di queste è il tempo necessario per elaborare gli aggiornamenti BGP in entrata o in uscita tramite criteri di route configurati. Esistono diversi modi per scrivere un criterio di route per eseguire un'attività specifica. Un modo ottimale contribuisce a migliorare la convergenza BGP, ridurre al minimo le potenziali perdite di traffico ed evitare loop di routing temporanei. Cisco IOS® XR include uno strumento di profiling che misura il tempo impiegato da criteri di route specifici per stimarne il tempo di elaborazione.

Problema

La lenta convergenza BGP è a volte il risultato di policy sulle route scritte in modo non ottimale.

Soluzione

È disponibile uno strumento di analisi dei criteri per i criteri di route che può essere utilizzato senza alcun impatto sulle prestazioni per misurare il tempo impiegato in ogni istruzione di un criterio di route in un punto di connessione specifico. È possibile controllare il runtime del criterio di route in questo punto di connessione specifico. Per impostazione predefinita, la profilatura è abilitata solo per le statistiche aggregate dei criteri di route.

```
router bgp 65000
neighbor 10.0.54.6
remote-as 65000
update-source Loopback0
address-family ipv4 unicast
route-policy INGRESS-ROUTE-POLICY in
!
neighbor 10.0.54.11
remote-as 65001
ebgp-multihop 255
update-source Loopback0
address-family ipv4 unicast
route-policy EGRESS-ROUTE-POLICY out
```

<#root>

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile

```
Policy profiling data
Policy : INGRESS-ROUTE-POLICY
Pass : 1440233
Drop : 0
# of executions : 1440233
Total execution time : 57095msec      <=====
```

RP/0/RSP1/CPU0:XR1#

show bgp ipv4 unicast neighbors 10.0.54.11 | i Update group

```
Update group: 0.3 Filter-group: 0.5 No Refresh request being processed
RP/0/RSP1/CPU0:XR1#
```

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

```
Policy profiling data
Policy : EGRESS-ROUTE-POLICY
Pass : 726751
Drop : 0
# of executions : 726751
Total execution time : 108099msec <=====
RP/0/RSP1/CPU0:XR1#
```

Viene visualizzata la quantità cumulativa di tempo impiegata per elaborare le policy in entrata-ROUTE e in uscita-ROUTE-POLICY.

La creazione dei profili può essere applicata per i criteri di route in entrata o in uscita per qualsiasi punto di connessione.

<#root>

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 ?

debug-policy	Attachpoint name
permet	Attachpoint name
import	Attachpoint name
export	Attachpoint name
interafi-import	Attachpoint name
source-rt	Attachpoint name
interafi-export	Attachpoint name
retain-rt	Attachpoint name
addpath	Attachpoint name
neighbor-in-dflt	Attachpoint name
neighbor-in-vrf	Attachpoint name
neighbor-out-dflt	Attachpoint name
neighbor-out-vrf	Attachpoint name
orf-dflt	Attachpoint name
orf-vrf	Attachpoint name
dampening-dflt	Attachpoint name
dampening-vrf	Attachpoint name
default-originate-dflt	Attachpoint name
default-originate-vrf	Attachpoint name
clear-policy	Attachpoint name
show-policy-node0_RSP1_CPU0	Attachpoint name
aggregation-dflt	Attachpoint name
aggregation-vrf	Attachpoint name
nexthop	Attachpoint name
allocate-label	Attachpoint name
label-mode	Attachpoint name
l2vpn-import	Attachpoint name
l2vpn-export	Attachpoint name
redistribution-dflt	Attachpoint name
redistribution-vrf	Attachpoint name
rib-install-dflt	Attachpoint name
rib-install-vrf	Attachpoint name
network-dflt	Attachpoint name
network-vrf	Attachpoint name
redistribution-dflt	Attachpoint name
redistribution-vrf	Attachpoint name
rib-install-dflt	Attachpoint name
rib-install-vrf	Attachpoint name
network-dflt	Attachpoint name
network-vrf	Attachpoint name
l2vpn-export-mp2mp	Attachpoint name
l2vpn-export-vfi	Attachpoint name
l2vpn-export-evi	Attachpoint name
l2vpn-export-mspw	Attachpoint name
l2vpn-export-instance	Attachpoint name
WORD	Attachpoint name

RP/0/RSP1/CPU0:XR1#

È possibile cancellare lo stato in base alle esigenze.

<#root>

RP/0/RSP1/CPU0:XR1#

```
clear pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile
```

```
RP/0/RSP1/CPU0:XR1#
```

```
clear pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile
```

Se si abilita debug pc1 profile detail, si otterranno statistiche dettagliate per voce dei criteri di route.

```
<#root>
```

```
RP/0/RSP1/CPU0:XR1#
```

```
debug pc1 profile detail
```

Questi output sono stati raccolti dopo la completa scalabilità delle tabelle BGP Internet.

```
<#root>
```

```
RP/0/RSP1/CPU0:XR1#
```

```
show pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile
```

```
Policy profiling data
```

```
Policy : INGRESS-ROUTE-POLICY
```

```
Pass : 720100
```

```
Drop : 0
```

```
# of executions : 720100
```

```
Total execution time : 222788msec <===== about 3.7 minutes to process ingress updates
```

Node Id	Num visited	Exec time	Policy engine operation
PXL_0_1	720100	221796msec	if as-path aspath-match ... then <=====
			<truePath>
PXL_0_3	3525	3msec	set local-preference 150
	3525	0msec	<end-policy/>
			</truePath>
			<falsePath>
PXL_0_2	716575	225msec	set local-preference 50
	716575	82msec	<end-policy/>
			</falsePath>

```
RP/0/RSP1/CPU0:XR1#
```

```
show pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile
```

```
Policy profiling data
```

```
Policy : EGRESS-ROUTE-POLICY
```

```
Pass : 720105
```

```
Drop : 0
```

```
# of executions : 720105
```

```
Total execution time : 221975msec <===== about 3.7 minutes to process egress updates
```

Node Id	Num visited	Exec time	Policy engine operation
---------	-------------	-----------	-------------------------

PXL_0_1	720105	3005msec	if as-path aspath-match ... then	
			<truePath>	
PXL_0_5	0	0msec	set med 70	
	0	0msec	<end-policy/>	
			</truePath>	
			<falsePath>	
PXL_0_2	720105	218008msec	if as-path aspath-match ... then	<=====
			<truePath>	
PXL_0_3	25	0msec	set med 80	
	25	0msec	<end-policy/>	
			</truePath>	
			<falsePath>	
PXL_0_4	720080	145msec	set med 90	
	720080	76msec	<end-policy/>	
			</falsePath>	
			</falsePath>	

RP/0/RSP1/CPU0:XR1#

Come si può vedere, la riga PXL_0_1 per INGRESS-ROUTE-POLICY e PXL_0_2 per EGRESS-ROUTE-POLICY richiedono un tempo particolare e rallentano la convergenza.

Se le si correlano ai criteri di route, è possibile vedere che il problema è causato da AS-PATH-SET-11 in INGRESS-ROUTE-POLICY e AS-PATH-SET-22 in EGRESS-ROUTE-POLICY. Ogni set di percorsi AS è costituito da 100 righe di espressione regolare, il che è un modo piuttosto inefficiente per scrivere un criterio poiché non sfrutta alcuna possibile ottimizzazione o la potenza di regex.

```
route-policy INGRESS-ROUTE-POLICY
  if as-path in AS-PATH-SET-11 then
    set local-preference 150
  else
    set local-preference 50
  endif
end-policy
```

```
route-policy EGRESS-ROUTE-POLICY
  if as-path in AS-PATH-SET-21 then
    set med 70
  elseif as-path in AS-PATH-SET-22 then
    set med 80
  else
    set med 90
  endif
end-policy
```

```
as-path-set AS-PATH-SET-11
  ios-regex '^65101 65201_',
  ios-regex '^65102 65202_',
  ios-regex '^65103 65203_',
  ios-regex '^65104 65204_',
  ios-regex '^65105 65205_',
  --- removed 90 similar lines ---
  ios-regex '^65195 65295_',
```

```

ios-regex '^65196 65296_',
ios-regex '^65197 65297_',
ios-regex '^65198 65298_',
ios-regex '^65199 65299_'
end-set

as-path-set AS-PATH-SET-21
ios-regex '^$'
end-set

as-path-set AS-PATH-SET-22
ios-regex '^65169(_65169)*$',
ios-regex '^65392(_65392)*$',
ios-regex '^65133(_65133)*$',
ios-regex '^65231(_65231)*$',
ios-regex '^65161(_65161)*$',
--- removed 90 similar lines ---
ios-regex '^65281(_65281)*$',
ios-regex '^65336(_65336)*$',
ios-regex '^65238(_65238)*$',
ios-regex '^65381(_65381)*$',
ios-regex '^65103(_65103)*$'
end-set

```

Per migliorare le prestazioni dei criteri, è possibile valutare la configurazione dei set di percorsi AS con l'operazione di corrispondenza as-path nativa anziché con l'espressione regolare. In alternativa, è possibile utilizzare le espressioni regolari all'interno dei criteri di route in modo compresso, riducendo di conseguenza il numero di righe di espressione regolare utilizzate.

In questa tabella vengono elencati i criteri di corrispondenza del percorso AS offerti da Route Policy Language (RPL). Le funzioni di corrispondenza native utilizzano un algoritmo di corrispondenza binaria che offre prestazioni migliori rispetto al motore di corrispondenza delle espressioni regolari. La maggior parte degli scenari di corrispondenza ios-regex comuni potrebbe essere scritta con loro (o con le loro combinazioni).

Comando	Descrizione
locale	Determina se il router (o un altro router all'interno del sistema autonomo o della confederazione) ha originato il router
lunghezza	Esegue un controllo condizionale in base alla lunghezza del percorso AS
vicino-è	Verifica il numero o i numeri di sistema autonomi all'inizio del percorso AS confrontandoli con una sequenza di uno o più valori o parametri integrali.
originato-da	Verifica un percorso AS rispetto alla sequenza AS dall'inizio con il numero AS da cui proviene una route.

pass-through	Verifica se il valore integer o il parametro specificato viene visualizzato in un punto qualsiasi del percorso AS o se viene visualizzata la sequenza di valori integer e parametri.
lunghezza unica	Esegue controlli specifici in base alla lunghezza del percorso AS ignorando i duplicati

Verifica

Si tratta di criteri di route riorganizzati scritti con l'aiuto dei criteri di corrispondenza nativi. Ciò comporta una notevole riduzione dei tempi di elaborazione.

```
route-policy INGRESS-ROUTE-POLICY
  if as-path in AS-PATH-SET-11 then
    set local-preference 150
  else
    set local-preference 50
  endif
end-policy
```

```
route-policy EGRESS-ROUTE-POLICY
  if as-path is-local then
    set med 70
  elseif as-path in AS-PATH-SET-22 and as-path unique-length is 1 then
    set med 80
  else
    set med 90
  endif
end-policy
```

```
as-path-set AS-PATH-SET-11
  neighbor-is '65101 65201',
  neighbor-is '65102 65202',
  neighbor-is '65103 65203',
  neighbor-is '65104 65204',
  neighbor-is '65105 65205',
  --- removed 90 similar lines ---
  neighbor-is '65195 65295',
  neighbor-is '65196 65296',
  neighbor-is '65197 65297',
  neighbor-is '65198 65298',
  neighbor-is '65199 65299'
end-set
```

```
as-path-set AS-PATH-SET-22
  originates-from '65169',
  originates-from '65392',
  originates-from '65133',
  originates-from '65231',
  originates-from '65161',
  --- removed 90 similar lines ---
  originates-from '65281',
```

```

    originates-from '65336',
    originates-from '65238',
    originates-from '65381',
    originates-from '65103'
end-set

```

Questi output vengono raccolti dopo la ricezione e la propagazione successive della scala completa della tabella BGP Internet.

<#root>

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile

```

Policy profiling data
Policy : INGRESS-ROUTE-POLICY
Pass : 720100
Drop : 0
# of executions : 720100
Total execution time : 9612msec <===== about 10 seconds to process ingress updates

```

Node Id	Num visited	Exec time	Policy engine operation

PXL_0_1	720100	8540msec	if as-path aspath-match ... then
			<truePath>
PXL_0_3	7128	2msec	set local-preference 150
	7128	1msec	<end-policy/>
			</truePath>
			<falsePath>
PXL_0_2	712972	276msec	set local-preference 50
	712972	80msec	<end-policy/>
			</falsePath>

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

```

Policy profiling data
Policy : EGRESS-ROUTE-POLICY
Pass : 720126
Drop : 0
# of executions : 720126
Total execution time : 12399msec <===== about 12 seconds to process egress updates

```

Node Id	Num visited	Exec time	Policy engine operation

PXL_0_1	720126	190msec	if as-path is-local then
			<truePath>
PXL_0_7	0	0msec	set med 70
	0	0msec	<end-policy/>
			</truePath>
			<falsePath>
PXL_0_2	720126	11190msec	if as-path aspath-match ... then
			<truePath>
PXL_0_4	262734	65msec	if as-path unique-length is 1 then
			<truePath>
PXL_0_5	25	0msec	set med 80

	25	0msec	<end-policy/>
			</truePath>
			<falsePath>
PXL_0_6	720101	164msec	set med 90
	720101	57msec	<end-policy/>
			</falsePath>
			</truePath>
			<falsePath>
			<reference>
GOTO :			PXL_0_6
			</reference>
			</falsePath>
			</falsePath>

In alternativa, le linee ios-regex potrebbero essere compresse. Ciò consente inoltre di migliorare le prestazioni.

```
route-policy EGRESS-ROUTE-POLICY
  if as-path in (ios-regex '^$') then
    set med 70
  endif
  if as-path in (ios-regex '^65169(_65169)*$|^65392(_65392)*$|^65133(_65133)*$|^65231(_65231)*$|^65161(_65161)*$|^65162(_65162)*$|^65163(_65163)*$|^65164(_65164)*$|^65165(_65165)*$|^65166(_65166)*$|^65167(_65167)*$|^65168(_65168)*$|^65169(_65169)*$|^65170(_65170)*$|^65171(_65171)*$|^65172(_65172)*$|^65173(_65173)*$|^65174(_65174)*$|^65175(_65175)*$|^65176(_65176)*$|^65177(_65177)*$|^65178(_65178)*$|^65179(_65179)*$|^65180(_65180)*$|^65181(_65181)*$|^65182(_65182)*$|^65183(_65183)*$|^65184(_65184)*$|^65185(_65185)*$|^65186(_65186)*$|^65187(_65187)*$|^65188(_65188)*$|^65189(_65189)*$|^65190(_65190)*$|^65191(_65191)*$|^65192(_65192)*$|^65193(_65193)*$|^65194(_65194)*$|^65195(_65195)*$|^65196(_65196)*$|^65197(_65197)*$|^65198(_65198)*$|^65199(_65199)*$|^65200(_65200)*$|^65201(_65201)*$|^65202(_65202)*$|^65203(_65203)*$|^65204(_65204)*$|^65205(_65205)*$|^65206(_65206)*$|^65207(_65207)*$|^65208(_65208)*$|^65209(_65209)*$|^65210(_65210)*$|^65211(_65211)*$|^65212(_65212)*$|^65213(_65213)*$|^65214(_65214)*$|^65215(_65215)*$|^65216(_65216)*$|^65217(_65217)*$|^65218(_65218)*$|^65219(_65219)*$|^65220(_65220)*$|^65221(_65221)*$|^65222(_65222)*$|^65223(_65223)*$|^65224(_65224)*$|^65225(_65225)*$|^65226(_65226)*$|^65227(_65227)*$|^65228(_65228)*$|^65229(_65229)*$|^65230(_65230)*$|^65231(_65231)*$|^65232(_65232)*$|^65233(_65233)*$|^65234(_65234)*$|^65235(_65235)*$|^65236(_65236)*$|^65237(_65237)*$|^65238(_65238)*$|^65239(_65239)*$|^65240(_65240)*$|^65241(_65241)*$|^65242(_65242)*$|^65243(_65243)*$|^65244(_65244)*$|^65245(_65245)*$|^65246(_65246)*$|^65247(_65247)*$|^65248(_65248)*$|^65249(_65249)*$|^65250(_65250)*$|^65251(_65251)*$|^65252(_65252)*$|^65253(_65253)*$|^65254(_65254)*$|^65255(_65255)*$|^65256(_65256)*$|^65257(_65257)*$|^65258(_65258)*$|^65259(_65259)*$|^65260(_65260)*$|^65261(_65261)*$|^65262(_65262)*$|^65263(_65263)*$|^65264(_65264)*$|^65265(_65265)*$|^65266(_65266)*$|^65267(_65267)*$|^65268(_65268)*$|^65269(_65269)*$|^65270(_65270)*$|^65271(_65271)*$|^65272(_65272)*$|^65273(_65273)*$|^65274(_65274)*$|^65275(_65275)*$|^65276(_65276)*$|^65277(_65277)*$|^65278(_65278)*$|^65279(_65279)*$|^65280(_65280)*$|^65281(_65281)*$|^65282(_65282)*$|^65283(_65283)*$|^65284(_65284)*$|^65285(_65285)*$|^65286(_65286)*$|^65287(_65287)*$|^65288(_65288)*$|^65289(_65289)*$|^65290(_65290)*$|^65291(_65291)*$|^65292(_65292)*$|^65293(_65293)*$|^65294(_65294)*$|^65295(_65295)*$|^65296(_65296)*$|^65297(_65297)*$|^65298(_65298)*$|^65299(_65299)*$|^65300(_65300)*$|^65301(_65301)*$|^65302(_65302)*$|^65303(_65303)*$|^65304(_65304)*$|^65305(_65305)*$|^65306(_65306)*$|^65307(_65307)*$|^65308(_65308)*$|^65309(_65309)*$|^65310(_65310)*$|^65311(_65311)*$|^65312(_65312)*$|^65313(_65313)*$|^65314(_65314)*$|^65315(_65315)*$|^65316(_65316)*$|^65317(_65317)*$|^65318(_65318)*$|^65319(_65319)*$|^65320(_65320)*$|^65321(_65321)*$|^65322(_65322)*$|^65323(_65323)*$|^65324(_65324)*$|^65325(_65325)*$|^65326(_65326)*$|^65327(_65327)*$|^65328(_65328)*$|^65329(_65329)*$|^65330(_65330)*$|^65331(_65331)*$|^65332(_65332)*$|^65333(_65333)*$|^65334(_65334)*$|^65335(_65335)*$|^65336(_65336)*$|^65337(_65337)*$|^65338(_65338)*$|^65339(_65339)*$|^65340(_65340)*$|^65341(_65341)*$|^65342(_65342)*$|^65343(_65343)*$|^65344(_65344)*$|^65345(_65345)*$|^65346(_65346)*$|^65347(_65347)*$|^65348(_65348)*$|^65349(_65349)*$|^65350(_65350)*$|^65351(_65351)*$|^65352(_65352)*$|^65353(_65353)*$|^65354(_65354)*$|^65355(_65355)*$|^65356(_65356)*$|^65357(_65357)*$|^65358(_65358)*$|^65359(_65359)*$|^65360(_65360)*$|^65361(_65361)*$|^65362(_65362)*$|^65363(_65363)*$|^65364(_65364)*$|^65365(_65365)*$|^65366(_65366)*$|^65367(_65367)*$|^65368(_65368)*$|^65369(_65369)*$|^65370(_65370)*$|^65371(_65371)*$|^65372(_65372)*$|^65373(_65373)*$|^65374(_65374)*$|^65375(_65375)*$|^65376(_65376)*$|^65377(_65377)*$|^65378(_65378)*$|^65379(_65379)*$|^65380(_65380)*$|^65381(_65381)*$|^65382(_65382)*$|^65383(_65383)*$|^65384(_65384)*$|^65385(_65385)*$|^65386(_65386)*$|^65387(_65387)*$|^65388(_65388)*$|^65389(_65389)*$|^65390(_65390)*$|^65391(_65391)*$|^65392(_65392)*$|^65393(_65393)*$|^65394(_65394)*$|^65395(_65395)*$|^65396(_65396)*$|^65397(_65397)*$|^65398(_65398)*$|^65399(_65399)*$|^65400(_65400)*$|^65401(_65401)*$|^65402(_65402)*$|^65403(_65403)*$|^65404(_65404)*$|^65405(_65405)*$|^65406(_65406)*$|^65407(_65407)*$|^65408(_65408)*$|^65409(_65409)*$|^65410(_65410)*$|^65411(_65411)*$|^65412(_65412)*$|^65413(_65413)*$|^65414(_65414)*$|^65415(_65415)*$|^65416(_65416)*$|^65417(_65417)*$|^65418(_65418)*$|^65419(_65419)*$|^65420(_65420)*$|^65421(_65421)*$|^65422(_65422)*$|^65423(_65423)*$|^65424(_65424)*$|^65425(_65425)*$|^65426(_65426)*$|^65427(_65427)*$|^65428(_65428)*$|^65429(_65429)*$|^65430(_65430)*$|^65431(_65431)*$|^65432(_65432)*$|^65433(_65433)*$|^65434(_65434)*$|^65435(_65435)*$|^65436(_65436)*$|^65437(_65437)*$|^65438(_65438)*$|^65439(_65439)*$|^65440(_65440)*$|^65441(_65441)*$|^65442(_65442)*$|^65443(_65443)*$|^65444(_65444)*$|^65445(_65445)*$|^6544
```

```

set med 80
endif
if as-path in (ios-regex '^65354(_65354)*$|^65331(_65331)*$|^65342(_65342)*$|^65295(_65295)*$|^65208(_65208)*$|^65333(_65333)*$|^65332(_65332)*$|^65334(_65334)*$|^65335(_65335)*$|^65336(_65336)*$|^65337(_65337)*$|^65338(_65338)*$|^65339(_65339)*$|^65340(_65340)*$|^65341(_65341)*$|^65343(_65343)*$|^65344(_65344)*$|^65345(_65345)*$|^65346(_65346)*$|^65347(_65347)*$|^65348(_65348)*$|^65349(_65349)*$|^65350(_65350)*$|^65351(_65351)*$|^65352(_65352)*$|^65353(_65353)*$|^65354(_65354)*$|^65355(_65355)*$|^65356(_65356)*$|^65357(_65357)*$|^65358(_65358)*$|^65359(_65359)*$|^65360(_65360)*$|^65361(_65361)*$|^65362(_65362)*$|^65363(_65363)*$|^65364(_65364)*$|^65365(_65365)*$|^65366(_65366)*$|^65367(_65367)*$|^65368(_65368)*$|^65369(_65369)*$|^65370(_65370)*$|^65371(_65371)*$|^65372(_65372)*$|^65373(_65373)*$|^65374(_65374)*$|^65375(_65375)*$|^65376(_65376)*$|^65377(_65377)*$|^65378(_65378)*$|^65379(_65379)*$|^65380(_65380)*$|^65381(_65381)*$|^65382(_65382)*$|^65383(_65383)*$|^65384(_65384)*$|^65385(_65385)*$|^65386(_65386)*$|^65387(_65387)*$|^65388(_65388)*$|^65389(_65389)*$|^65390(_65390)*$|^65391(_65391)*$|^65392(_65392)*$|^65393(_65393)*$|^65394(_65394)*$|^65395(_65395)*$|^65396(_65396)*$|^65397(_65397)*$|^65398(_65398)*$|^65399(_65399)*$|^65400(_65400)*$|^65401(_65401)*$|^65402(_65402)*$|^65403(_65403)*$|^65404(_65404)*$|^65405(_65405)*$|^65406(_65406)*$|^65407(_65407)*$|^65408(_65408)*$|^65409(_65409)*$|^65410(_65410)*$|^65411(_65411)*$|^65412(_65412)*$|^65413(_65413)*$|^65414(_65414)*$|^65415(_65415)*$|^65416(_65416)*$|^65417(_65417)*$|^65418(_65418)*$|^65419(_65419)*$|^65420(_65420)*$|^65421(_65421)*$|^65422(_65422)*$|^65423(_65423)*$|^65424(_65424)*$|^65425(_65425)*$|^65426(_65426)*$|^65427(_65427)*$|^65428(_65428)*$|^65429(_65429)*$|^65430(_65430)*$|^65431(_65431)*$|^65432(_65432)*$|^65433(_65433)*$|^65434(_65434)*$|^65435(_65435)*$|^65436(_65436)*$|^65437(_65437)*$|^65438(_65438)*$|^65439(_65439)*$|^65440(_65440)*$|^65441(_65441)*$|^65442(_65442)*$|^65443(_65443)*$|^65444(_65444)*$|^65445(_65445)*$|^65446(_65446)*$|^65447(_65447)*$|^65448(_65448)*$|^65449(_65449)*$|^65450(_65450)*$|^65451(_65451)*$|^65452(_65452)*$|^65453(_65453)*$|^65454(_65454)*$|^65455(_65455)*$|^65456(_65456)*$|^65457(_65457)*$|^65458(_65458)*$|^65459(_65459)*$|^65460(_65460)*$|^65461(_65461)*$|^65462(_65462)*$|^65463(_65463)*$|^65464(_65464)*$|^65465(_65465)*$|^65466(_65466)*$|^65467(_65467)*$|^65468(_65468)*$|^65469(_65469)*$|^65470(_65470)*$|^65471(_65471)*$|^65472(_65472)*$|^65473(_65473)*$|^65474(_65474)*$|^65475(_65475)*$|^65476(_65476)*$|^65477(_65477)*$|^65478(_65478)*$|^65479(_65479)*$|^65480(_65480)*$|^65481(_65481)*$|^65482(_65482)*$|^65483(_65483)*$|^65484(_65484)*$|^65485(_65485)*$|^65486(_65486)*$|^65487(_65487)*$|^65488(_65488)*$|^65489(_65489)*$|^65490(_65490)*$|^65491(_65491)*$|^65492(_65492)*$|^65493(_65493)*$|^65494(_65494)*$|^65495(_65495)*$|^65496(_65496)*$|^65497(_65497)*$|^65498(_65498)*$|^65499(_65499)*$|^65500(_65500)*$|^65501(_65501)*$|^65502(_65502)*$|^65503(_65503)*$|^65504(_65504)*$|^65505(_65505)*$|^65506(_65506)*$|^65507(_65507)*$|^65508(_65508)*$|^65509(_65509)*$|^65510(_65510)*$|^65511(_65511)*$|^65512(_65512)*$|^65513(_65513)*$|^65514(_65514)*$|^65515(_65515)*$|^65516(_65516)*$|^65517(_65517)*$|^65518(_65518)*$|^65519(_65519)*$|^65520(_65520)*$|^65521(_65521)*$|^65522(_65522)*$|^65523(_65523)*$|^65524(_65524)*$|^65525(_65525)*$|^65526(_65526)*$|^65527(_65527)*$|^65528(_65528)*$|^65529(_65529)*$|^65530(_65530)*$|^65531(_65531)*$|^65532(_65532)*$|^65533(_65533)*$|^65534(_65534)*$|^65535(_65535)*$|^65536(_65536)*$|^65537(_65537)*$|^65538(_65538)*$|^65539(_65539)*$|^65540(_65540)*$|^65541(_65541)*$|^65542(_65542)*$|^65543(_65543)*$|^65544(_65544)*$|^65545(_65545)*$|^65546(_65546)*$|^65547(_65547)*$|^65548(_65548)*$|^65549(_65549)*$|^65550(_65550)*$|^65551(_65551)*$|^65552(_65552)*$|^65553(_65553)*$|^65554(_65554)*$|^65555(_65555)*$|^65556(_65556)*$|^65557(_65557)*$|^65558(_65558)*$|^65559(_65559)*$|^65560(_65560)*$|^65561(_65561)*$|^65562(_65562)*$|^65563(_65563)*$|^65564(_65564)*$|^65565(_65565)*$|^65566(_65566)*$|^65567(_65567)*$|^65568(_65568)*$|^65569(_65569)*$|^65570(_65570)*$|^65571(_65571)*$|^65572(_65572)*$|^65573(_65573)*$|^65574(_65574)*$|^65575(_65575)*$|^65576(_65576)*$|^65577(_65577)*$|^65578(_65578)*$|^65579(_65579)*$|^65580(_65580)*$|^65581(_65581)*$|^65582(_65582)*$|^65583(_65583)*$|^65584(_65584)*$|^65585(_65585)*$|^65586(_65586)*$|^65587(_65587)*$|^65588(_65588)*$|^65589(_65589)*$|^65590(_65590)*$|^65591(_65591)*$|^65592(_65592)*$|^65593(_65593)*$|^65594(_65594)*$|^65595(_65595)*$|^65596(_65596)*$|^65597(_65597)*$|^65598(_65598)*$|^65599(_65599)*$|^65600(_65600)*$|^65601(_65601)*$|^65602(_65602)*$|^65603(_65603)*$|^65604(_65604)*$|^65605(_65605)*$|^65606(_65606)*$|^65607(_65607)*$|^65608(_65608)*$|^65609(_65609)*$|^65610(_65610)*$|^65611(_65611)*$|^65612(_65612)*$|^65613(_65613)*$|^65614(_65614)*$|^65615(_65615)*$|^65616(_65616)*$|^65617(_65617)*$|^65618(_65618)*$|^65
```

Questi output sono stati raccolti dopo la ricezione e la propagazione della scala completa della tabella BGP Internet.

<#root>

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-in-dflt default-IPv4-Uni-10.0.54.6 policy profile

Policy profiling data

Policy : INGRESS-ROUTE-POLICY

Pass : 720100

Drop : 0

of executions : 720100

Total execution time : 30119msec <===== about 30 seconds to process ingress updates

Node Id	Num visited	Exec time	Policy engine operation
PXL_0_1	720100	4434msec	if as-path aspath-match ... then <truePath>
PXL_0_2	361	0msec	set local-preference 150
PXL_0_3	720100	3039msec	if as-path aspath-match ... then <truePath>
--- removed lines ---			
GOTO :			PXL_0_3 </reference> </falsePath>

RP/0/RSP1/CPU0:XR1#

show pc1 protocol bgp speaker-0 neighbor-out-dflt default-IPv4-Uni-UpdGrp-0.3-Out policy profile

Policy profiling data

Policy : EGRESS-ROUTE-POLICY

Pass : 720110

Drop : 0

of executions : 720110

Total execution time : 106566msec <===== about 1.8 minutes to process egress updates

Node Id	Num visited	Exec time	Policy engine operation
PXL_0_1	720110	2958msec	if as-path aspath-match ... then <truePath>
PXL_0_2	0	0msec	set med 70
PXL_0_3	720110	5222msec	if as-path aspath-match ... then <truePath>
PXL_0_4	3	0msec	set med 80
PXL_0_5	720110	4979msec	if as-path aspath-match ... then <truePath>
PXL_0_6			set med 80
--- removed lines ---			
GOTO :			PXL_0_3 </reference> </falsePath>

Conclusioni

Le prestazioni di un criterio di route possono essere migliorate con criteri di corrispondenza as-path nativi o espressioni regolari compresse.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).