

Cloud Platform and Infrastructures as a Code

Antonio Martino M63001644

Introduzione

Negli ultimi anni, la crescente domanda di scalabilità, disponibilità e prestazioni nei datacenter ha reso necessaria un'evoluzione significativa delle architetture di rete. Le topologie tradizionali, spesso basate su strutture gerarchiche, hanno evidenziato limiti importanti in termini di capacità e resilienza, soprattutto in presenza di traffico *Est-Ovest* intensivo, sempre più comune nei datacenter moderni.

In questo contesto, l'architettura **Leaf-Spine** si è affermata come una soluzione più scalabile ed efficiente, grazie alla sua struttura non bloccante dovuta alla presenza di percorsi multipli in parallelo, migliorando così la gestione del traffico e la tolleranza ai guasti.

In questo elaborato si propone un confronto tra una **rete Leaf-Spine** e una **rete bloccante**, implementate e simulate tramite **Containerlab**. L'obiettivo è analizzare le differenze tra le due architetture e dimostrare come la scelta della topologia possa influenzare le prestazioni e la resilienza in ambienti cloud, attraverso test di throughput, analisi della tolleranza ai guasti e valutazioni in scenari ad alto carico.

Repository Git-Hub: <https://github.com/antonio-martino/CPIAC-Project>

Architettura delle Reti

Topologia Leaf-Spine

La rete Leaf-Spine implementata è composta da:

- 2 switch **Spine**, ciascuno connesso a tutti gli switch Leaf;
- 4 switch **Leaf**, connessi sia agli Spine che agli host;
- 8 **Host**, distribuiti equamente sui Leaf.

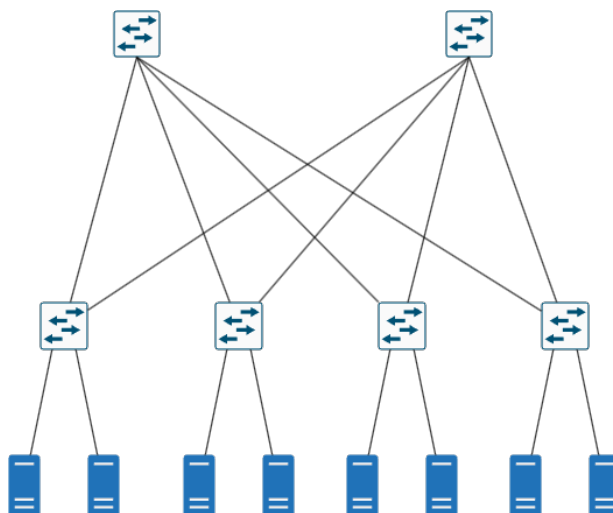


Figura 1: Schema logico della rete Leaf-Spine

Topologia Gerarchica Tradizionale

La rete gerarchica è strutturata come segue:

- 1 switch **Core**, al vertice della gerarchia;
- 2 switch di **Aggregazione**, connessi al Core e agli switch di accesso;
- 4 switch di **Accesso**, ciascuno collegato a due host;
- 8 **Host**, connessi agli switch di accesso.

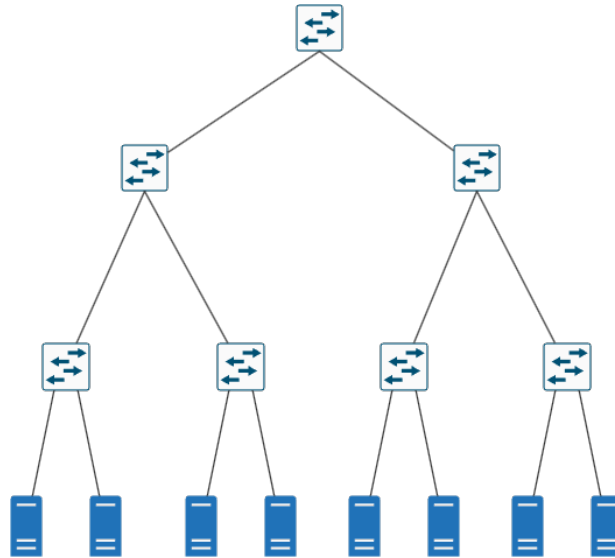


Figura 2: Schema logico della rete gerarchica tradizionale

Per l'implementazione delle due reti è stato utilizzato **Containerlab**, con switch virtuali basati su **FRRouting**.

Per rendere il confronto più realistico e significativo, è stato inoltre introdotto un limite di banda di **1000 Mbit/s** su ciascun collegamento.

Test

Misurazione Throughput

L'obiettivo di questo test è confrontare il throughput tra la rete gerarchica bloccante e la topologia Leaf-Spine, utilizzando lo strumento **iperf3** per simulare e analizzare il traffico generato.

Configurazione del test

- Su ciascun host con numero pari è stato avviato un server **iperf3** in ascolto.
- Gli host mittenti hanno avviato sessioni client **iperf3** simultanee verso i rispettivi destinatari, secondo le seguenti coppie:
 - host1 → host6
 - host3 → host8
 - host5 → host2
 - host7 → host4
- Ogni test ha avuto una durata di 10 secondi ed è stato eseguito con **5 stream TCP paralleli** (-P 5) per simulare un carico realistico.
- L'intera procedura è stata automatizzata tramite uno script **.sh**.

Risultati

Di seguito sono riportati i risultati ottenuti dal test.

Rete Bloccante

Coppie	Throughput [Mbit/s]	Ritrasmissioni
H1 - H6	581	151
H3 - H8	382	654
H5 - H2	417	1134
H7 - H4	459	1215

Tabella 1: Throughput e ritrasmissioni della rete bloccante

Rete Leaf-Spine

Coppie	Throughput [Mbit/s]	Ritrasmissioni
H1 - H6	949	11
H3 - H8	946	9
H5 - H2	947	15
H7 - H4	948	8

Tabella 2: Throughput e ritrasmissioni della rete Leaf-Spine

I risultati ottenuti evidenziano una differenza sostanziale nelle prestazioni tra le due architetture analizzate:

- La rete gerarchica ha fatto registrare un throughput medio complessivo di circa **459 Mbit/s**, accompagnato da un numero elevato di ritrasmissioni.
- La topologia Leaf-Spine ha raggiunto un throughput medio di circa **947 Mbit/s**, con un numero trascurabile di ritrasmissioni.

Tali differenze riflettono le caratteristiche strutturali delle due soluzioni.

La rete gerarchica, di tipo bloccante, soffre della presenza di colli di bottiglia dovuti alla mancanza di percorsi alternativi e alla concentrazione del traffico in pochi nodi intermedi. Queste limitazioni penalizzano le prestazioni complessive della rete.

La topologia Leaf-Spine, al contrario, è progettata per garantire ridondanza e bilanciamento del traffico grazie alla presenza di percorsi multipli. Questo approccio permette una distribuzione più efficiente del carico e si traduce in prestazioni significativamente superiori.

Misurazione della latenza

L'obiettivo di questo test è quello di confrontare i valori di latenza tra le due reti.

Configurazione del test

- Per simulare traffico di fondo, è stato eseguito un test **iperf3** analogo a quello del test di throughput.
- In parallelo, sono stati eseguiti test di latenza tra diverse coppie di host, utilizzando il comando **ping**.
- L'intera procedura è stata automatizzata tramite uno script **.sh**.

Risultati

Di seguito sono riportati i risultati ottenuti dal test.

Rete Bloccante

Test	RTT
Test 1	51.818
Test 2	51.479
Test 3	50.756
Test 4	51.274
Test 5	50.396
Test 6	50.012
Test 7	51.038
Test 8	51.338

Tabella 3: Risultati latenza rete bloccante

Rete Leaf-Spine

Test	RTT
Test 1	0.096
Test 2	10.527
Test 3	0.134
Test 4	10.329
Test 5	0.130
Test 6	5.675
Test 7	5.741
Test 8	10.467

Tabella 4: Risultati latenza rete leaf-spine

I risultati ottenuti evidenziano una differenza sostanziale nelle prestazioni tra le due architetture analizzate:

- La rete gerarchica ha fatto registrare una latenza media complessiva di circa **51 ms**.
- La topologia Leaf-Spine invece di circa **5 ms**.

Tali differenze riflettono ancora una volta le caratteristiche strutturali delle due soluzioni.

La rete gerarchica mostra una latenza significativamente più elevata, comportamento che è riconducibile ancora una volta alla presenza di colli di bottiglia e all'assenza di percorsi alternativi, che costringono il traffico a transitare attraverso un numero limitato di nodi centrali, aumentando i tempi di risposta e compromettendo l'efficienza complessiva.

Al contrario, la Leaf-Spine ha registrato una latenza media molto più bassa grazie alla presenza di più percorsi disponibili tra i nodi, che permette di distribuire il traffico in modo più efficiente e di ridurre i ritardi.

Simulazione Link Failure

In questa sezione analizziamo il comportamento delle due topologie al verificarsi di un'interruzione di un collegamento.

Configurazione del test

- Sono stati eseguiti comandi **ping** e **traceroute** da **host8** a **host4** prima dell'interruzione del collegamento.
- È stata simulata l'interruzione disabilitando una delle interfacce di rete coinvolte nel percorso.
- I test di **ping** e **traceroute** sono stati ripetuti per osservare l'effetto della disconnessione.
- L'intera procedura è stata automatizzata tramite uno script **.sh**.

Risultati

Di seguito sono riportati i risultati ottenuti dal test.

Rete Bloccante

Dall'analisi dei risultati ottenuti prima dell'interruzione del collegamento, si osserva che il comando **ping** funziona correttamente, come previsto. Inoltre, l'output di **traceroute** conferma il percorso atteso, che segue la sequenza: **host8** → **access4** → **agg2** → **core** → **agg1** → **access2** → **host4**.

Simuliamo quindi un'interruzione tra **agg1** e **core**.

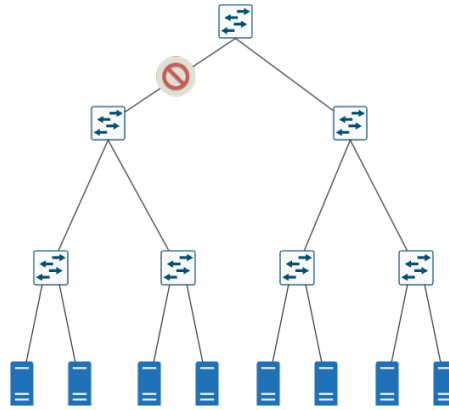


Figura 3: Interruzione del collegamento tra **agg1** e **core**

A seguito della disconnessione tra **agg1** e **core**, **host4** risulta completamente isolato dalla rete, in quanto non esistono percorsi disponibili per instradare il traffico verso questo nodo, rendendo impossibile qualsiasi comunicazione.

```
PING 10.0.0.14 (10.0.0.14) 56(84) bytes of data.
From 10.0.0.29 icmp_seq=1 Destination Net Unreachable
From 10.0.0.29 icmp_seq=2 Destination Net Unreachable
```

Figura 4: Risultato test ping post-interruzione

Rete Leaf-Spine

Anche in questo caso, prima dell'interruzione del collegamento, il comando **ping** funziona correttamente e l'analisi tramite **traceroute** conferma che il percorso seguito è **host8** → **leaf4** → **spine1** → **leaf2** → **host4**.

```
1 traceroute to 10.0.0.14 (10.0.0.14), 30 hops max, 46 byte packets
 2 1 10.0.0.29 (10.0.0.29) 0.009 ms 0.024 ms 0.006 ms
 3 2 10.0.1.38 (10.0.1.38) 0.005 ms 0.012 ms 0.011 ms
 4 3 10.0.1.13 (10.0.1.13) 0.006 ms 0.009 ms 0.008 ms
 5 4 10.0.0.14 (10.0.0.14) 0.005 ms 0.007 ms 0.006 ms
```

Figura 5: Percorso pre-interruzione nella rete Leaf-Spine

Successivamente, si procede a disabilitare il collegamento tra **spine1** e **leaf2**, portandoci alla seguente situazione:

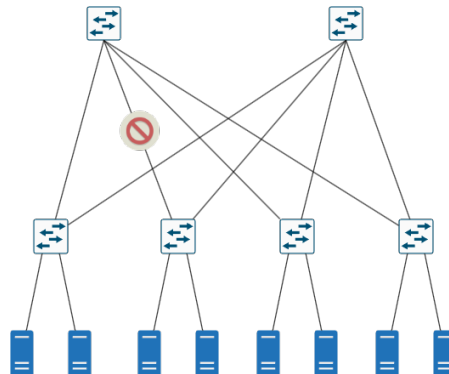


Figura 6: Interruzione del collegamento tra **spine1** e **leaf2**

Analizzando i risultati del test dopo l'interruzione, si osserva il comportamento atteso: la rete si adatta prontamente alla variazione. In particolare, dal nuovo **traceroute** si nota che il percorso seguito per raggiungere la destinazione è cambiato in **host8** → **leaf4** → **spine2** → **leaf2** → **host4**.


```

1 traceroute to 10.0.0.14 (10.0.0.14), 30 hops max, 46 byte packets
2 1  10.0.0.29 (10.0.0.29)  0.008 ms  0.008 ms  0.010 ms
3 2  10.0.1.42 (10.0.1.42)  0.002 ms  0.010 ms  0.006 ms
4 3  10.0.1.17 (10.0.1.17)  0.018 ms  0.006 ms  0.006 ms
5 4  10.0.0.14 (10.0.0.14)  0.003 ms  0.008 ms  0.013 ms

```

Figura 7: Percorso post-interruzione nella rete Leaf-Spine

I risultati della simulazione evidenziano come la rete Leaf-Spine dimostri una maggiore capacità di resilienza rispetto alla struttura gerarchica.

Nel caso della topologia bloccante, l'interruzione di un singolo collegamento comporta la perdita totale di connettività tra i nodi coinvolti, a causa dell'assenza di percorsi alternativi.

Al contrario, la rete Leaf-Spine, grazie alla presenza di molteplici percorsi ridondanti, è in grado di mantenere il flusso di traffico anche in caso di guasti, garantendo così la continuità del servizio.

Conclusioni

In questo elaborato sono state analizzate e messe a confronto due diverse architetture di rete comunemente impiegate nei datacenter: la tradizionale rete gerarchica a tre livelli e la più moderna topologia Leaf-Spine. Attraverso una serie di test automatizzati, sono state valutate le prestazioni di entrambe le soluzioni in termini di:

- **Throughput**
- **Latenza**
- **Tolleranza ai guasti**

I risultati ottenuti evidenziano come l'architettura Leaf-Spine rappresenti la scelta più adatta per i datacenter moderni, in cui scalabilità, bassa latenza e alta disponibilità costituiscono requisiti essenziali per gestire l'incremento del traffico e i carichi di lavoro distribuiti.

Pur comportando costi e complessità di implementazione superiori rispetto alla topologia tradizionale, i vantaggi offerti in termini di prestazioni e affidabilità ne giustificano pienamente l'adozione.