

Cloud Platform and Infrastructures as a Code

Antonio Martino M63001644

Introduzione

Negli ultimi anni, la crescente richiesta di scalabilità, disponibilità e prestazioni nei datacenter ha portato a un'evoluzione significativa nelle architetture di rete. Le topologie tradizionali, spesso basate su strutture gerarchiche hanno mostrato limiti evidenti in termini di capacità e resilienza, specialmente in scenari con traffico *East-West* intensivo, tipico dei moderni datacenter.

In questo contesto, l'architettura **Leaf-Spine** si è affermata come una soluzione più scalabile e performante, grazie alla sua natura non bloccante e alla possibilità di sfruttare percorsi multipli in parallelo.

In questo elaborato si propone un confronto tra una **rete Leaf-Spine** e una **rete bloccante di tipo gerarchico**, implementate e simulate tramite *Containerlab*, un framework moderno per la creazione di laboratori di rete containerizzati. L'obiettivo è valutare le differenze tra le due architetture e dimostrare come la scelta della topologia di rete possa influenzare le prestazioni e la resilienza di ambienti cloud distribuiti, attraverso test di throughput, tolleranza ai guasti e analisi del comportamento in scenari di carico.

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

Architettura delle Reti

In questa sezione vengono descritte le due architetture di rete progettate e implementate al fine di condurre un'analisi comparativa delle loro prestazioni in scenari tipici di datacenter.

Topologia Leaf-Spine

La topologia *Leaf-Spine* è una rete non bloccante, progettata per garantire elevata scalabilità e bassa latenza nei flussi di traffico *east-west*. La sua struttura è simmetrica e altamente ridondata, consentendo un bilanciamento efficiente del carico tra gli switch. La rete 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.

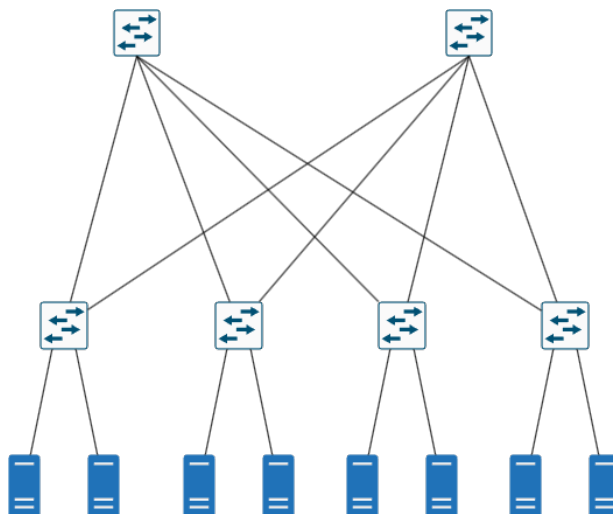


Figura 1: Schema logico della rete Leaf-Spine

Topologia Gerarchica Tradizionale

La seconda rete adottata segue un'architettura gerarchica classica a tre livelli, tipica dei datacenter legacy. Questa topologia è di tipo bloccante risultando più suscettibile a colli di bottiglia in scenari di traffico intenso. La rete 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.

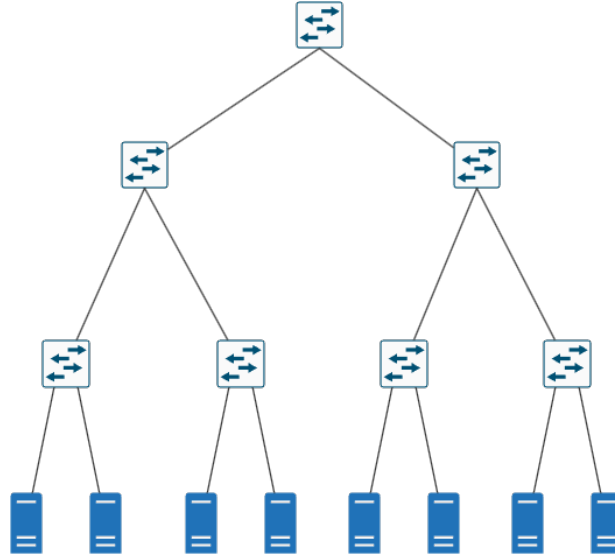


Figura 2: Schema logico della rete gerarchica tradizionale

Entrambe le topologie sono state implementate utilizzando switch virtuali basati su **FRRouting (FRR)**. Per simulare scenari realistici di congestione e limitazioni infrastrutturali, è stato applicato un vincolo di banda tramite il comando `tc` (traffic control), impostando la capacità dei collegamenti a **1000 Mbit/s**. Questo approccio consente di valutare le differenze prestazionali tra le due architetture in condizioni comparabili.

Test

In questa sezione vengono presentati i test condotti per valutare le prestazioni delle due topologie di rete implementate, con particolare attenzione al throughput e alla stabilità delle connessioni.

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

Host Pair	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

Host Pair	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 sia la capacità di trasmissione sia la stabilità complessiva della rete.

La topologia Leaf-Spine, al contrario, è progettata per garantire ridondanza e bilanciamento del traffico: ogni switch Leaf è collegato a tutti gli Spine, offrendo percorsi multipli tra gli host e minimizzando il rischio di congestione. Questo approccio consente una distribuzione più efficiente del carico e si traduce in prestazioni nettamente superiori.

Misurazione della latenza

L'obiettivo di questo test è quello di misurare 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 avg [ms]
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 avg [ms]
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, con una media di circa **51 ms**. Questo comportamento 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 riducendo l'efficienza.

La topologia Leaf-Spine, al contrario, ha registrato una latenza media di circa **5 ms**, grazie alla sua natura altamente ridondata. Ogni nodo è in grado di raggiungere gli altri attraverso molteplici percorsi equivalenti, garantendo una distribuzione più uniforme del traffico e riducendo al minimo i tempi di trasmissione.

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**.

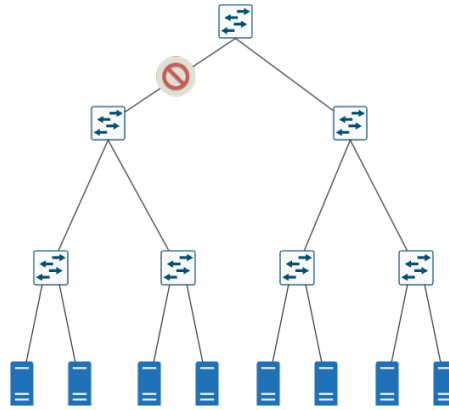


Figura 3: Interruzione del collegamento tra **agg1** e **core**

Come previsto, a seguito di tale interruzione l'host **host4** non risulta più raggiungibile.

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 4: Percorso pre-interruzione nella rete Leaf-Spine

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

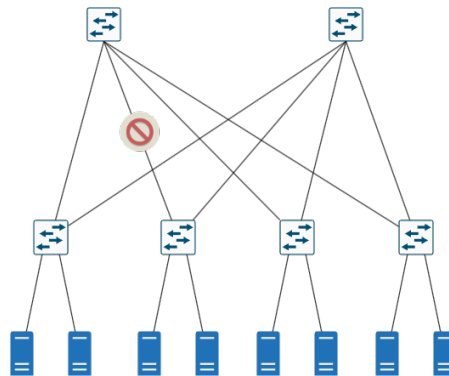


Figura 5: Interruzione del collegamento tra **spine1** e **leaf2**

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 6: Percorso post-interruzione nella rete Leaf-Spine

I risultati della simulazione evidenziano chiaramente la maggiore resilienza della rete Leaf-Spine rispetto alla rete gerarchica.

Nel caso della topologia bloccante, l'interruzione di un singolo collegamento critico comporta la perdita totale di connettività tra i nodi coinvolti, a causa dell'assenza di percorsi alternativi. Al contrario, nella rete Leaf-Spine, l'esistenza di più spine switch permette di instradare dinamicamente il traffico su percorsi alternativi in caso di guasto, garantendo così la continuità del servizio.

Questa capacità di adattamento evidenzia uno dei principali vantaggi dell'architettura Leaf-Spine: l'elevata tolleranza ai guasti. In scenari reali, tale proprietà si traduce in una maggiore affidabilità dell'infrastruttura e in tempi di inattività drasticamente ridotti.

Conclusioni

Nel corso di questa analisi, sono state confrontate due differenti architetture di rete tipicamente impiegate in ambiente datacenter: la tradizionale rete gerarchica a tre livelli e la più moderna topologia Leaf-Spine. Attraverso una serie di test automatizzati, sono emerse le principali differenze prestazionali e strutturali tra le due soluzioni.

I test effettuati hanno evidenziato:

- **Throughput:** la rete Leaf-Spine ha dimostrato prestazioni significativamente superiori, grazie alla disponibilità di molteplici percorsi paralleli e all'eliminazione dei colli di bottiglia caratteristici della rete gerarchica bloccante.
- **Latenza:** nei test di latenza, la topologia Leaf-Spine ha fatto registrare valori medi nettamente inferiori, risultando più efficiente nella gestione del traffico grazie alla sua struttura bilanciata e ridondante.
- **Affidabilità e tolleranza ai guasti:** L'affidabilità della rete Leaf-Spine è emersa chiaramente durante la simulazione di guasti. A differenza della rete gerarchica, che mostra una maggiore vulnerabilità a causa della presenza di pochi punti di transito critici per il traffico dati, la topologia Leaf-Spine è in grado di gestire dinamicamente i guasti, instradando il traffico su percorsi alternativi. Questa capacità di adattamento assicura la continuità del servizio, un aspetto essenziale nei contesti critici come i datacenter moderni.

In sintesi, l'architettura Leaf-Spine si conferma come la soluzione più adatta per i datacenter moderni, dove scalabilità, bassa latenza e alta disponibilità sono essenziali per supportare le crescenti esigenze di traffico e carichi di lavoro distribuiti. Pur comportando costi e complessità iniziali maggiori, i benefici in termini di prestazioni e affidabilità giustificano ampiamente la sua adozione.