

REPORT TESTING

Di seguito, viene analizzata in dettaglio una istanza di run del sistema, confrontando i risultati attesi con quelli ottenuti.

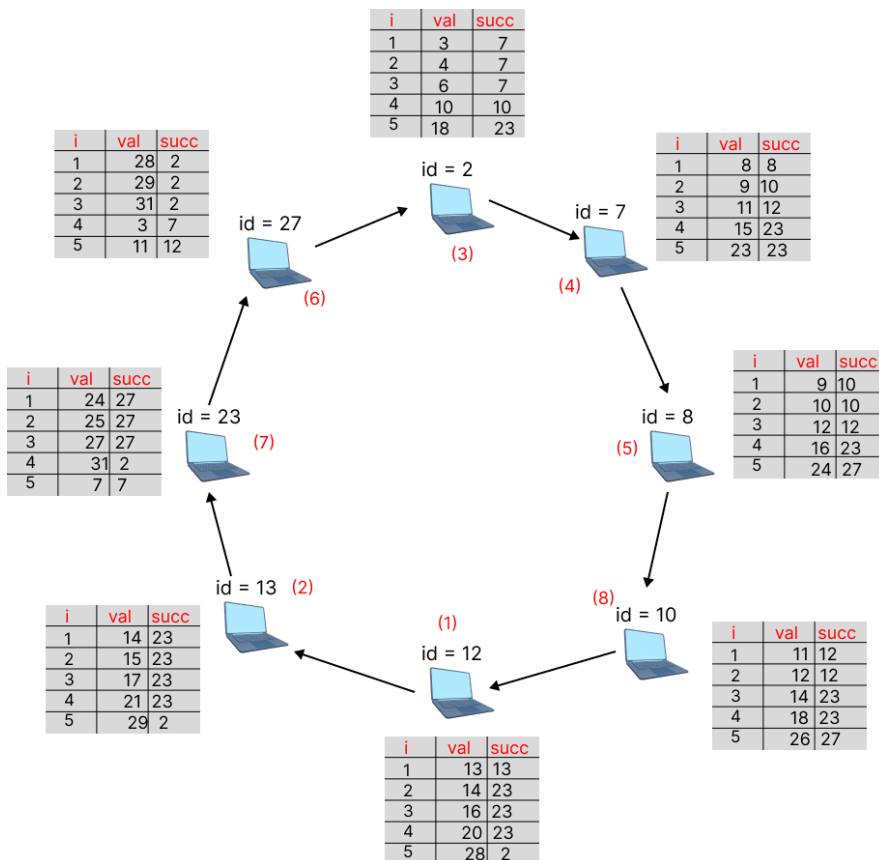
Creazione di una rete.

Il sistema ha il compito di generare 9 *nod*i, sfruttando un $\#GUID\ bits = 5$, permettendo dunque la creazione di massimo 32 *nod*i, a meno di collisioni.

Il sistema istanzia correttamente 8 nodi su 9.

La mancata creazione del nono nodo è data dall'id assegnatogli partendo dal suo Indirizzo *IP*:172.19.0.9:8009. E' stato infatti mappato su un *id* = 13, già assegnato ad un altro nodo presente. Gli otto nodi creati hanno i seguenti *id* ordinati: 2, 7, 8, 10, 12, 13, 23, 27.

Il sistema che ci **aspettiamo** di ottenere è il seguente:

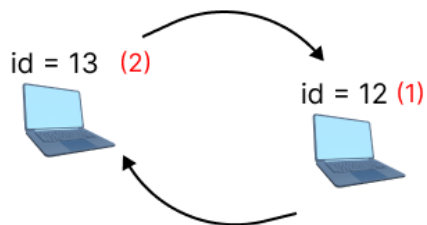


Nella figura notiamo che ogni nodo è identificato da:

- un *identificativo univoco*.
- un *numero che indica l'ordine di entrata nel sistema*. Il primo nodo è il 12, poi il 13, seguito dal 2 e così via. Questo secondo elemento è introdotto unicamente nella figura, non nel sistema, per facilitare l'interpretazione della costruzione del sistema.

Evoluzione dell'anello all'entrata di un nuovo nodo:

- Il primo nodo è 12, che imposta se stesso come predecessore e successore.
- Entra il nodo 13:



- **Output atteso:**

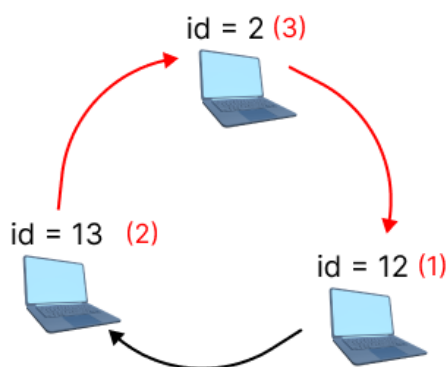
13 imposta 12 come nodo predecessore e successore.

- **Output ottenuto:**

Node 13, il mio nuovo predecessore e' [12]:172.19.0.3:8012

Node 13, il mio nuovo successore e' [12]:172.19.0.3:8012

- Entra il nodo 2:



- **Output atteso:**

2 si colloca dopo di 13 e prima di 12.

Il nuovo successore di 13 è 2.

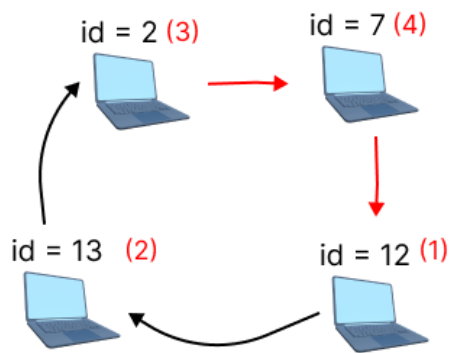
Il nuovo predecessore di 12 è 2.

- **Output ottenuto:**

Node 13, il mio nuovo successore e' [2]:172.19.0.6:8010

Node 12, il mio nuovo predecessore e' [2]:172.19.0.6:8010

- Entra il nodo 7:



- **Output atteso:**

7 si colloca dopo di 2 e prima di 12.

Il nuovo successore di 2 è 7.

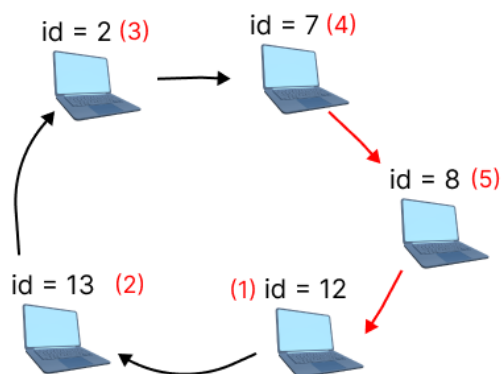
Il nuovo predecessore di 12 è 7.

- **Output ottenuto:**

Node 2, il mio nuovo successore e' [7]:172.19.0.5:8008

Node 12, il mio nuovo predecessore e' [7]:172.19.0.5:8008

- Entra il nodo 8:



- **Output atteso:**

8 si colloca dopo di 7 e prima di 12.

Il nuovo successore di 7 è 8.

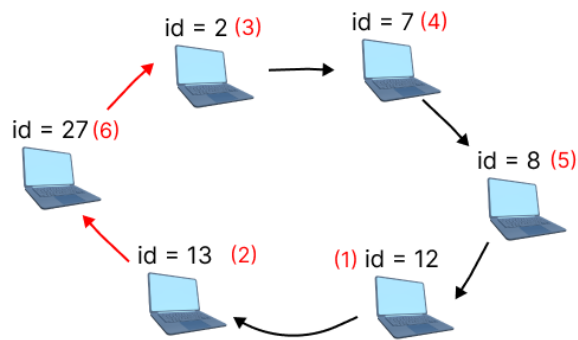
Il nuovo predecessore di 12 è 8.

- **Output ottenuto:**

Node 7, il mio nuovo successore e' [8]:172.19.0.11:8005

Node 12, il mio nuovo predecessore e' [8]:172.19.0.11:8005

- Entra il nodo 27:



- **Output atteso:**

27 si colloca dopo di 13 e prima di 2.

Il nuovo successore di 13 è 27.

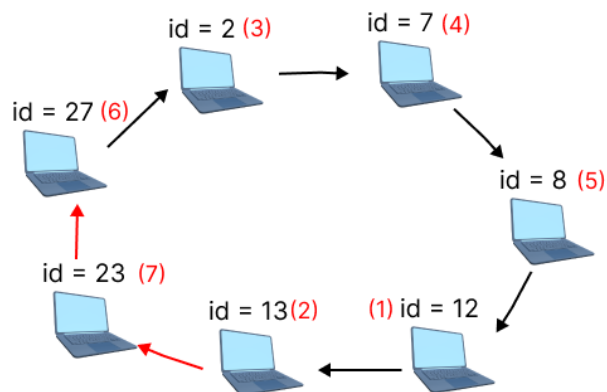
Il nuovo predecessore di 2 è 27.

- **Output ottenuto:**

Node 13, il mio nuovo successore e' [27]:172.19.0.7:8007

Node 2, il mio nuovo predecessore e' [27]:172.19.0.7:8007

- Entra il nodo 23:



- **Output atteso:**

23 si colloca dopo 13 e prima di 27.

Il nuovo successore di 13 è 23.

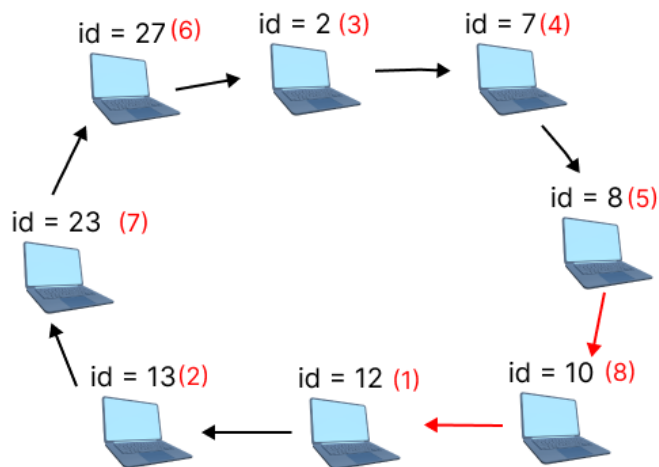
Il nuovo predecessore di 27 è 23.

- **Output ottenuto:**

Node 13, il mio nuovo successore e' [23]:172.19.0.10:8011

Node 27, il mio nuovo predecessore e' [23]:172.19.0.10:8011

- Entra il nodo 10:



- **Output atteso:**

10 si colloca tra 8 e 12.

Il nuovo successore di 8 è 10.

Il nuovo predecessore di 12 è 10.

- **Output ottenuto:**

Node 8, il mio nuovo successore e' [10]:172.19.0.8:8006

Node 12, il mio nuovo predecessore e' [10]:172.19.0.8:8006

Calcolo delle Finger Table.

E' possibile confrontare le FT previste (incluse nella prima immagine) con quelle prodotte dai nodi singolarmente.

```
FT[13]: <1,23> <2,23> <3,23> <4,23> <5,2>
FT[12]: <1,13> <2,23> <3,23> <4,23> <5,2>
FT[2]: <1,7> <2,7> <3,7> <4,10> <5,23>
FT[7]: <1,8> <2,10> <3,12> <4,23> <5,23>
FT[8]: <1,10> <2,10> <3,12> <4,23> <5,27>
FT[27]: <1,2> <2,2> <3,2> <4,7> <5,12>
FT[23]: <1,27> <2,27> <3,27> <4,2> <5,7>
FT[10]: <1,12> <2,12> <3,23> <4,23> <5,27>
```

Il calcolo è coerente con la previsione. La rimozione di un nodo porterà il nodo predecessore ad avere un nuovo nodo successore e quindi ad un nuovo ricalcolo della Finger Table (la quale viene ricalcolata ad intervalli di tempo prefissati). Poichè per la generazione della Finger Table si ha una comunicazione tra nodi, l'aggiornamento del successore si propagherà anche nelle altre Finger Table.

Aggiunta delle risorse.

Di seguito, viene proposta una tabella che include:

- Id della risorsa da aggiungere.
- Nodo che *dovrebbe* gestire la risorsa (*nodo atteso*).
- Nodo che *effettivamente* gestisce la risorsa. Deve coincidere col *nodo atteso*.

ID	Nodo atteso	Nodo ottenuto
6	7	7
9	10	10
11	12	12
12	12	12
13	13	13
16	23	23
19	23	23
22	23	23
24	27	27
27	27	27

Di seguito, l'output prodotto dall'aggiunta delle risorse.

Nodo: 7, map[6:stai]

Nodo: 10, map[9:a]

Nodo: 12, map[11:bene 12:di]

Nodo: 13, map[13:milano]

Nodo: 23, map[16:ni 19:come 22:tutto]

Nodo: 27, map[24:ciao 27:spero]

Ricerca delle risorse.

ID	Nodo/Output atteso	Output ottenuto
6	7	7
9	10	10
11	12	12
12	12	12
13	13	13
16	23	23
19	23	23
22	23	23
24	27	27
27	27	27
32	Oggetto non presente	L'oggetto cercato non è presente.
-1	Oggetto non presente	L'oggetto cercato non è presente.

Di seguito vengono riportati gli output prodotti dal sistema.

```
Digita l'id dell'oggetto da cercare: 6
L'oggetto con id '6' e valore 'stai' è posseduto dal nodo '7'.
```

```
Digita l'id dell'oggetto da cercare: 9
L'oggetto con id '9' e valore 'a' è posseduto dal nodo '10'.
```

```
Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 11
L'oggetto con id '11' e valore 'bene' è posseduto dal nodo '12'.
```

```
Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 12
L'oggetto con id '12' e valore 'di' è posseduto dal nodo '12'.
```

```
Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 13
L'oggetto con id '13' e valore 'milano' è posseduto dal nodo '13'.
```

```
Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 16
L'oggetto con id '16' e valore 'ni' è posseduto dal nodo '23'.

Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 19
L'oggetto con id '19' e valore 'come' è posseduto dal nodo '23'.

Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 22
L'oggetto con id '22' e valore 'tutto' è posseduto dal nodo '23'.

Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 24
L'oggetto con id '24' e valore 'ciao' è posseduto dal nodo '27'.

Inserisci scelta: 2
Digita l'id dell'oggetto da cercare: 27
L'oggetto con id '27' e valore 'spero' è posseduto dal nodo '27'.
```

Rimozione delle risorse.

Di seguito, viene esaminato il comportamento del sistema a fronte di una rimozione di una risorsa, sia essa presente nel sistema o meno. Questa operazione non altera le informazioni della Finger Table nè eventuali collegamenti tra nodi.

ID	Output atteso	Output ottenuto
6	Nodo 7, nessuna risorsa rimasta.	Nodo: 7, Objects: map[]
11	Nodo 12, id risorse rimaste: 12	Nodo: 12, Objects: map[12:di]
22	Nodo 23, id risorse rimaste: 16,19	Nodo: 23, Objects: map[16:ni 19:come]
31	Oggetto non presente.	L'oggetto cercato non è presente.
63	Oggetto non presente.	L'oggetto cercato non è presente.
-1	Oggetto non presente.	L'oggetto cercato non è presente.

Procediamo con alcuni test inerenti alla cancellazione controllata di un nodo. In questo caso, un nodo che viene rimosso comunica la sua uscita ai suoi nodi adiacenti permettendo una gestione corretta delle risorse, che verranno assegnate al nodo successore del nodo uscente.

Cancellazione di un nodo controllata.

ID nodo	Output atteso	Output ottenuto
8	- successore di 7 → 10, - predecessore di 10 → 7, -risorse trasferite: nessuna risorsa trasferita.	Foto 1:
23	- successore di 13 → 27, - predecessore di 27 → 13, - risorse trasferite: 27 eredita le risorse 16 e 19 (22 è stata rimossa.)	Foto 2:
32	Nodo non rimovibile poichè non presente.	Il nodo avente id '32' non è presente e dunque non è eliminabile.
6	Nodo non rimovibile poichè non presente.	Il nodo avente id '6' non è presente e dunque non è eliminabile.

foto 1:

```
Node 7, il mio nuovo successore e' [10]:172.19.0.8:8006  
Node 10, il mio nuovo predecessore e' [7]:172.19.0.5:8008
```

foto 2:

```
Node 13, il mio nuovo successore e' [27]:172.19.0.7:8007  
Node 27, il mio nuovo predecessore e' [13]:172.19.0.4:8013  
Node 27, ho un nuovo elemento: <16, ni>  
Node 27, ho un nuovo elemento: <19, come>
```

Concludiamo il test con la rimozione forzata di un nodo, dovuto ad eventi non prevedibili (come un crash).

In questo caso *non viene gestita la ri-allocazione delle risorse* in quanto non presente un meccanismo di replicazione.

Il sistema è tuttavia in grado di riposizionare correttamente i nodi adiacenti al nodo caduto. Di seguito viene proposto un test condotto per l'analisi dell'evento, con relativo risultato atteso, e l'output effettivamente ottenuto.

Cancellazione di un nodo dovuto ad un crash.

ID nodo	Output atteso	Output ottenuto
2	<ul style="list-style-type: none">- successore di 27 → 7 ;- predecessore di 7 → 27;- <i>risorse trasferite</i>: nessun trasferimento di risorse poichè non gestita la replicazione.	Foto 3:

foto 3:

```
Non riesco a contattare [2:172.19.0.6:8010], procedo con la sua rimozione.  
map[7:172.19.0.5:8008 10:172.19.0.8:8006 12:172.19.0.3:8012 13:172.19.0.4:8013 27:172.19.0.7:8007]  
Node 27, il mio nuovo successore e' [7]:172.19.0.5:8008  
Node 7, il mio nuovo predecessore e' [27]:172.19.0.7:8007
```