

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

Dynamic load balancing in DHT-based networks

[Bienkowski and Korzeniowski]

Matteo Tanca
Dejan Kolundžija

Dipartimento di Informatica
Università di Pisa

10 Maggio 2007

- 1 Distributed Hash Tables
- 2 CHORD
- 3 Problema dello sbilanciamento
- 4 Algoritmo di bilanciamento
- 5 Conclusioni

Distributed Hash Tables

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

Compromesso fra:

- Approccio centralizzato (indicizzazione)
- Approccio distribuito non strutturato (flooding)

Motivazioni:

- Scalabilità
- Assenza di punti di centralizzazione
- Efficienza della ricerca
- Impegno di memoria ridotto per nodo

Nodi e informazioni sono mappati su un unico spazio di indirizzamento

DHT: caratteristiche generali

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

- Ad ogni nodo ed ogni dato è assegnato un identificatore univoco
- Ogni nodo è responsabile di una porzione contigua dello spazio di indirizzamento (numericamente prossima al suo ID)
- Il nodo memorizza (direttamente o indirettamente) i dati che ne fanno parte
- L'ID è calcolato mediante funzione hash consistente (ad esempio SHA-1) per garantire con alta probabilità l'unicità
- In generale la dimensione del sottospazio assegnato ad ogni peer varia nel tempo, per via dell'inserimento di nuovi nodi e per l'abbandono di altri

DHT: diverse topologie

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- Topologia ad anello
[CHORD, Pastry]
- Topologia
multidimensionale [CAN]
- Topologia ad albero
[Kademlia]

DHT: diverse topologie

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

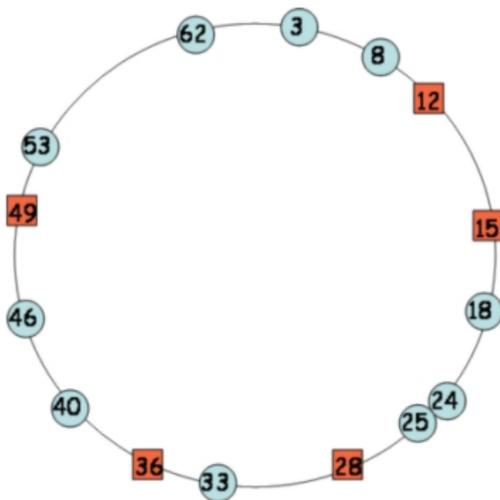
CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- Topologia ad anello [CHORD, Pastry]
- Topologia multidimensionale [CAN]
- Topologia ad albero [Kademlia]



DHT: diverse topologie

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

- Topologia ad anello
[CHORD, Pastry]
- Topologia
multidimensionale [CAN]
- Topologia ad albero
[Kademlia]

DHT: diverse topologie

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

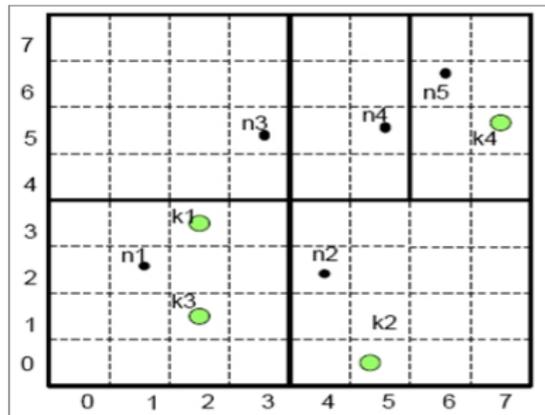
CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- Topologia ad anello [CHORD, Pastry]
- Topologia multidimensionale [CAN]
- Topologia ad albero [Kademlia]



DHT: diverse topologie

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

- Topologia ad anello
[CHORD, Pastry]
- Topologia
multidimensionale [CAN]
- Topologia ad albero
[Kademlia]

DHT: diverse topologie

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

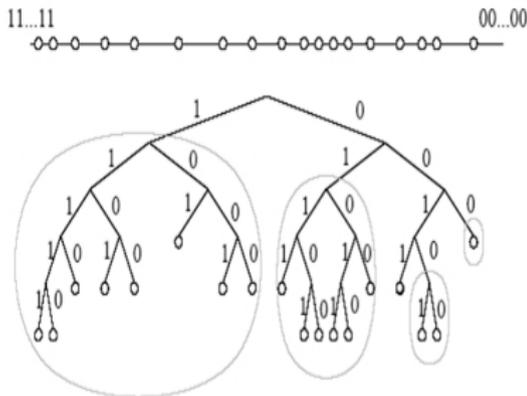
CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- Topologia ad anello [CHORD, Pastry]
- Topologia multidimensionale [CAN]
- Topologia ad albero [Kademlia]



DHT: caratteristiche generali dell'algoritmo di routing

Dynamic load balancing in DHT-based networks

Matteo Tanca, Dejan Kolundžija

Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni

ricerca } routing verso il nodo responsabile del dato
memorizzazione }

- Memorizzazione e ricerca differiscono per il tipo dei messaggi inviati/propagati
- Ogni nodo ha una visibilità soltanto parziale del sistema, determinata dalla sua tabella di routing
- $O(\log N)$ associazioni per tabella
- **Content-based routing**: rotta determinata in base all'ID del dato, secondo una nozione di prossimità logica (vicinanza numerica o adiacenza spaziale)

DHT: pregi e difetti

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Pregi: dato un sistema basato su DHT, composto di N nodi

- Dimensione tabella di routing: $O(\log N)$
- Costo della ricerca: $O(\log N)$ messaggi
- Nessun falso negativo in fase di ricerca

Difetti

- Impossibilità di query complesse
- Necessaria una suddivisione omogenea dell'informazione per una ricerca efficiente
- In assenza di replicazione dell'informazione a livello applicativo il fallimento del nodo responsabile ne determina la scomparsa

CHORD: caratteristiche generali [1/2]

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- **Identificatori:** identificatori di m -bit (tipicamente $m = 160$) calcolati tramite hash consistente (SHA-1)
- **Topologia:** identificatori ordinati su un anello $[0, 1, \dots, 2^m - 1]$ modulo 2^m
- **Distribuzione dei dati:** l'informazione (chiave) k è assegnata al primo nodo n il cui ID sia maggiore o uguale a k , detto successore di k (primo nodo incontrato procedendo in senso orario sull'anello, a partire da k)

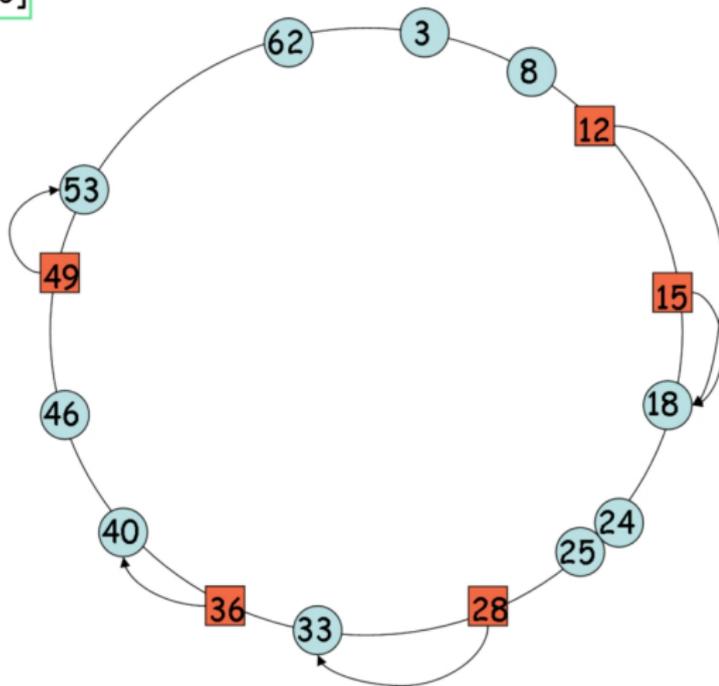
CHORD: esempio

$m = 6$

$[0, 1, \dots, 63]$

Chiave

Nodo



Dynamic load balancing in DHT-based networks

Matteo Tanca, Dejan Kolundžija

Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni

CHORD: caratteristiche generali [2/2]

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Tabella di routing (finger table):

- Per ogni nodo n una tabella di m associazioni
- $finger_n[i] = successore(ID_n + 2^{i-1})$

Routing dei messaggi: data una query per una chiave K

- $Query(K)$ è inviata al più distante finger noto che preceda k sull'anello, in senso orario
- $Query(K)$ è propagata fino al nodo m tale che m precede K e $successore(m) = successore(K)$
- m restituisce il suo successore

CHORD: finger table

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan Kolundžija

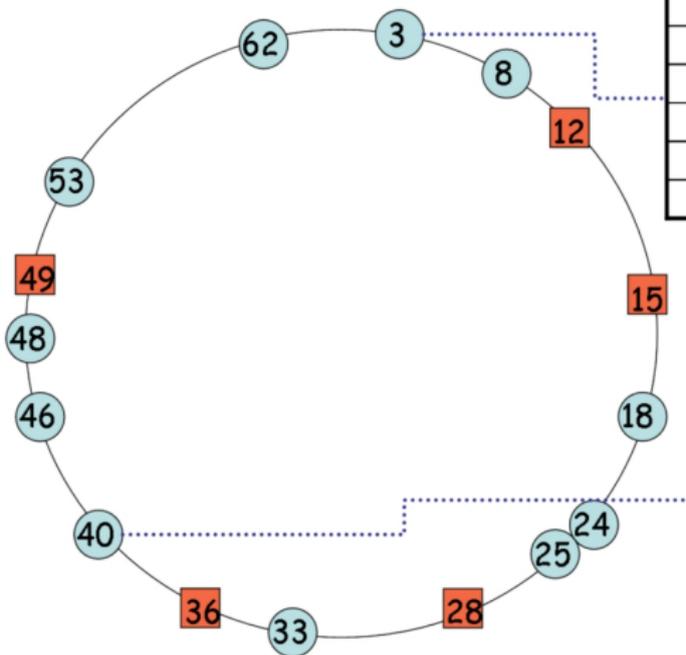
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Finger	Chiave	Successore
1	$3+2^0 = 4$	8
2	$3+2 = 5$	8
3	$3+2^2 = 7$	8
4	$3+2^3 = 11$	18
5	$3+2^4 = 19$	24
6	$3+2^5 = 35$	40

Finger	Chiave	Successore
1	$40+2^0 = 41$	46
2	$40+2 = 42$	46
3	$40+2^2 = 44$	46
4	$40+2^3 = 48$	48
5	$40+2^4 = 56$	62
6	$40+2^5 = 8$	8

CHORD: algoritmo di routing

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

`n.closest-preceding-node(k)`

```
for i = m downto 1
  if finger[i] is in (n, k)
    return finger[i]
```

`n.find-successor(k)`

```
if k is in (n, successor]
  return successor
else
  n' = closest-preceding-node(k)
  return n'.find-successor(k)
```

CHORD: routing di un messaggio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan Kolundžija

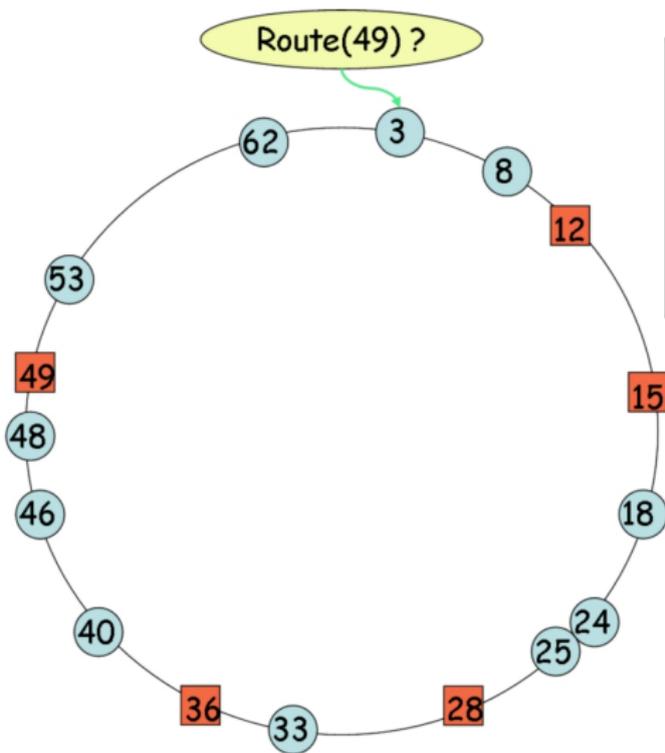
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Finger	Chiave	Successore
1	$3+2^0 = 4$	8
2	$3+2 = 5$	8
3	$3+2^2 = 7$	8
4	$3+2^3 = 11$	18
5	$3+2^4 = 19$	24
6	$3+2^5 = 35$	40

CHORD: routing di un messaggio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

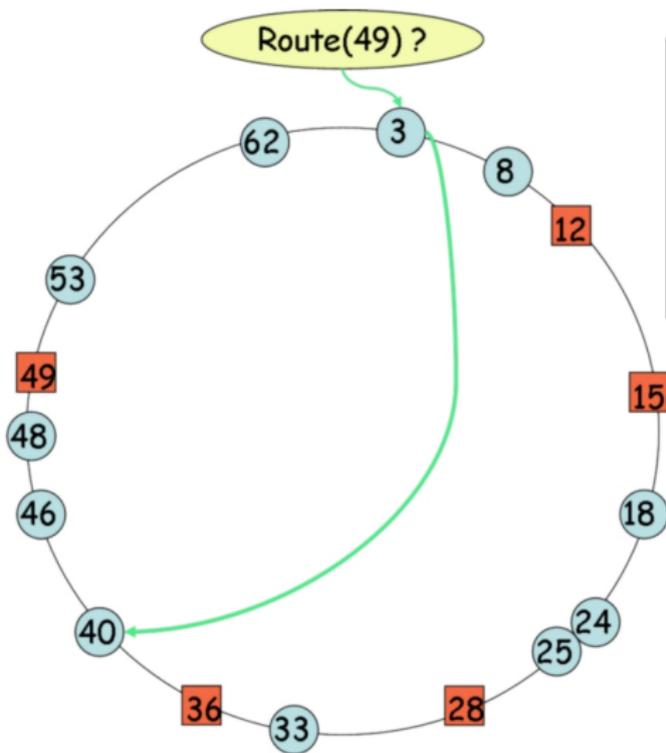
Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni



Finger	Chiave	Successore
1	$3+2^0 = 4$	8
2	$3+2 = 5$	8
3	$3+2^2 = 7$	8
4	$3+2^3 = 11$	18
5	$3+2^4 = 19$	24
6	$3+2^5 = 35$	40

CHORD: routing di un messaggio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan Kolundžija

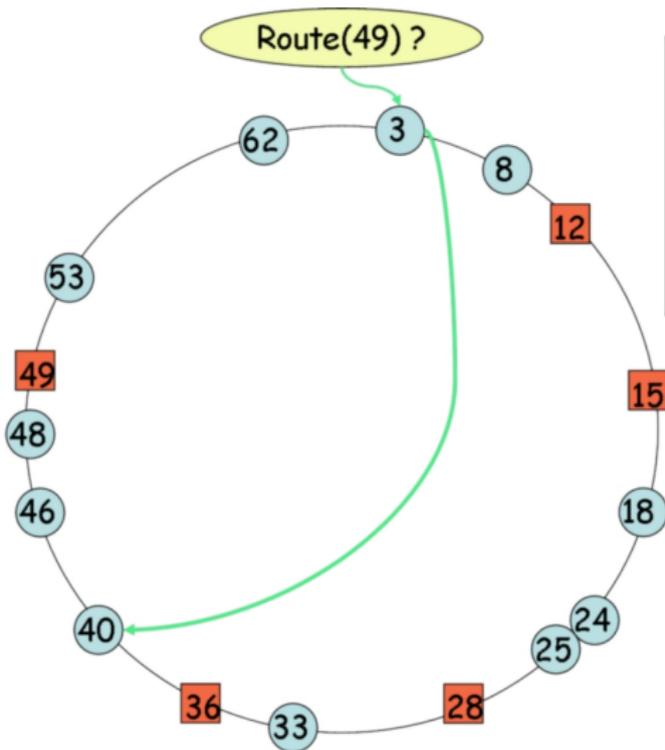
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Finger	Chiave	Successore
1	$3+2^0 = 4$	8
2	$3+2 = 5$	8
3	$3+2^2 = 7$	8
4	$3+2^3 = 11$	18
5	$3+2^4 = 19$	24
6	$3+2^5 = 35$	40

Finger	Chiave	Successore
1	$40+2^0 = 41$	46
2	$40+2 = 42$	46
3	$40+2^2 = 44$	46
4	$40+2^3 = 48$	48
5	$40+2^4 = 56$	62
6	$40+2^5 = 8$	8

CHORD: routing di un messaggio

Dynamic load balancing in DHT-based networks

Matteo Tanca, Dejan Kolundžija

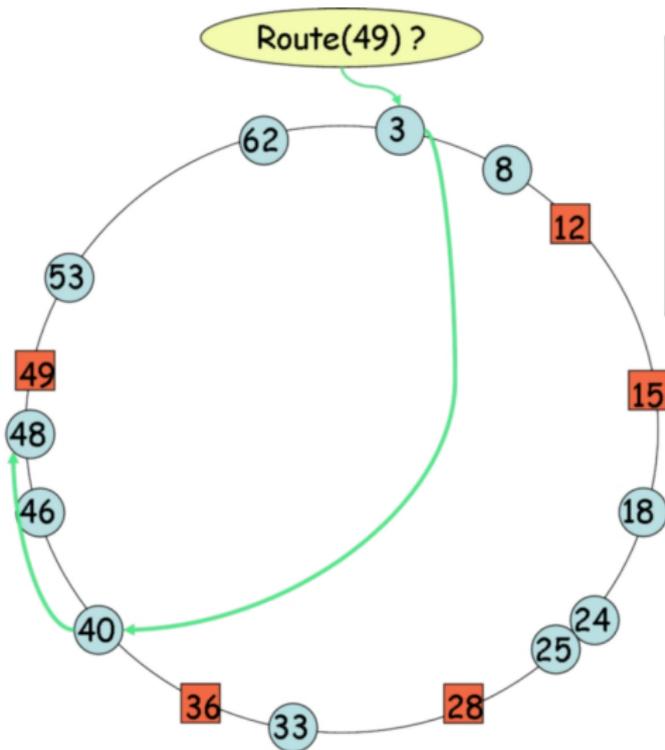
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Finger	Chiave	Successore
1	$3+2^0 = 4$	8
2	$3+2 = 5$	8
3	$3+2^2 = 7$	8
4	$3+2^3 = 11$	18
5	$3+2^4 = 19$	24
6	$3+2^5 = 35$	40

Finger	Chiave	Successore
1	$40+2^0 = 41$	46
2	$40+2 = 42$	46
3	$40+2^2 = 44$	46
4	$40+2^3 = 48$	48
5	$40+2^4 = 56$	62
6	$40+2^5 = 8$	8

CHORD: routing di un messaggio

Dynamic load balancing in DHT-based networks

Matteo Tanca, Dejan Kolundžija

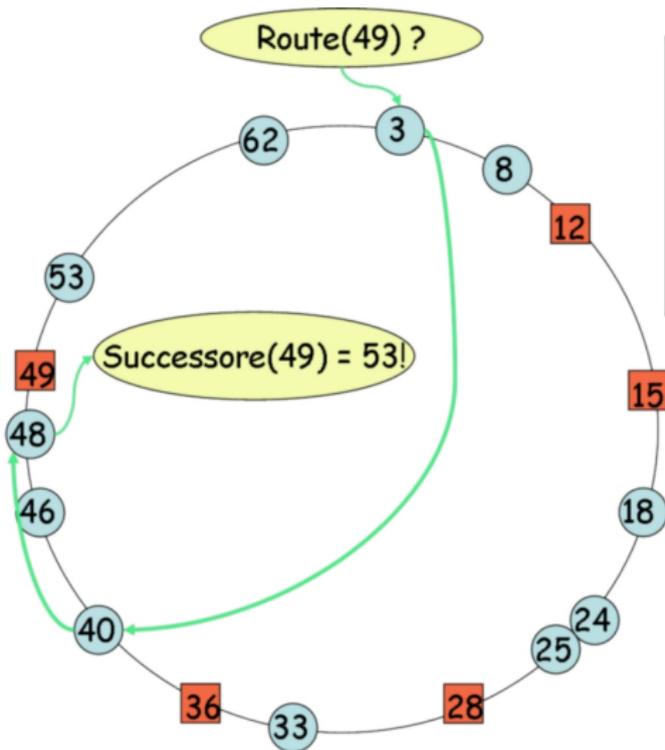
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Finger	Chiave	Successore
1	$3+2^0 = 4$	8
2	$3+2 = 5$	8
3	$3+2^2 = 7$	8
4	$3+2^3 = 11$	18
5	$3+2^4 = 19$	24
6	$3+2^5 = 35$	40

Finger	Chiave	Successore
1	$40+2^0 = 41$	46
2	$40+2 = 42$	46
3	$40+2^2 = 44$	46
4	$40+2^3 = 48$	48
5	$40+2^4 = 56$	62
6	$40+2^5 = 8$	8

CHORD: auto organizzazione del sistema

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- Il network Chord reagisce dinamicamente ai seguenti eventi:
 - Arrivo di un nuovo nodo
 - Partenza di un nodo esistente
 - Fallimento di un nodo
- **L'efficienza del routing è legata alla stabilità dell'anello**
- **Configurazione stabile:** tutte le finger table sono consistenti (aggiornate correttamente riguardo a nuovi nodi e nodi non più presenti)
- **Stabilizzazione dell'anello:** algoritmo eseguito periodicamente da ogni nodo n
 - `stabilize()` : aggiorna il riferimento a `successore(n)`
 - `fixFingers()`: inserisce eventuali riferimenti a nuovi nodi, mediante ricerca del successore di un finger scelto a caso dalla finger table

CHORD: procedure

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

n.create()

```
predecessor = null  
successor = n
```

n.join(n')

```
predecessor = null  
successor = n'.find-successor(n)
```

`n.stabilize()`

```
x = successor.predecessor
if (x is in (n, successor))
    successor = x
    successor.notify(n)
```

`n.fixFingers()`

```
next = random(1, ..., m)
finger[next]=find-successor(n+2^(next-1))
```

`n.notify(n')`

```
if predecessor = null or
    n' is in (predecessor, n)
    predecessor = n'
```

CHORD: inserimento di un nodo

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- 1 **Calcolo dell'ID**: il nuovo nodo n genera il proprio ID (hash sul proprio indirizzo) e contatta un nodo noto attivo sulla rete (bootstrap node)
- 2 **Calcolo della finger table**: ricavata mediante invio di query successive
- 3 **Notifica della presenza**: n ha una finger table corretta, ma non è ancora noto agli altri peer. L'esecuzione della procedura `stabilize()` causa la notifica della propria presenza al suo successore sull'anello, che si preoccuperà di inviare le chiavi non più di sua competenza al nuovo arrivato
- 4 **Aggiornamento delle finger table**: l'esecuzione periodica di `FixFingers()` rende le tabelle consistenti rispetto ad inserimenti e fallimenti di nodi

CHORD: fallimento di un nodo [1/2]

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

Timeout: ogni messaggio ha una scadenza. In caso di mancata risposta in tempo utile

- Il messaggio è inviato al finger che precedeva il destinatario in tabella (si evita di oltrepassare la destinazione)
- Nell'associazione il finger destinatario è sostituito dal finger seguente

Fallimenti multipli: nel caso di fallimenti multipli dei finger di un nodo n , se la sostituzione si propaga fino alla prima posizione della finger table

- $\text{finger}[1]$ potrebbe non essere più il vero successore di n
- la correttezza del routing non è più garantita

CHORD: fallimento di un nodo [2/2]

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Lista successori:

- dimensione r (costante)
- consultata in caso di fallimento di `finger[1]`
- aggiornata durante la stabilizzazione (tramite lista del successore)
- r grande \Rightarrow maggior robustezza del sistema
- r piccolo \Rightarrow minor overhead di auto-organizzazione

Abbandono volontario: all'abbandono del sistema il nodo

- avvisa il proprio predecessore
- avvisa il proprio successore, inviando ad esso le proprie chiavi

CHORD: riassunto

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Complessità: dato un anello $[0, 1, \dots, 2^m - 1]$, $N = 2^m$

- Routing: $O(\log N)$ messaggi
- Spazio richiesto: $\log N + r$ associazioni per nodo, dove r è la dimensione (costante) della lista successori
- Auto-organizzazione: $O(\log^2 N)$ messaggi

Problematiche

- Nessuna ridondanza dell'informazione (il fallimento di un nodo comporta la perdita delle relative chiavi)
- *Nessuna politica di bilanciamento del carico*

Sbilanciamento del carico

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- **Granularità (smoothness)**: rapporto fra la lunghezza dell'intervallo più ampio e dell'intervallo più piccolo
- **Obiettivi**:
 - Valori di smoothness piccoli ed indipendenti dal numero di nodi
 - Costo sostenibile dell'algoritmo di bilanciamento

Sbilanciamento del carico: esempio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan
Kolundžija

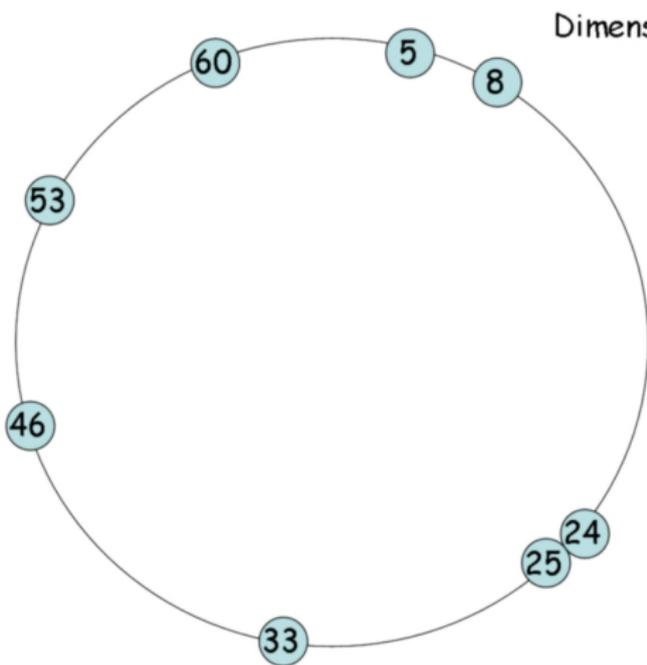
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Dimensione dell'anello = 2^6 identificatori

Numero di nodi = 8

Dimensione ideale dell'intervallo

$$64 / 8 = 8$$

Sbilanciamento del carico: esempio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

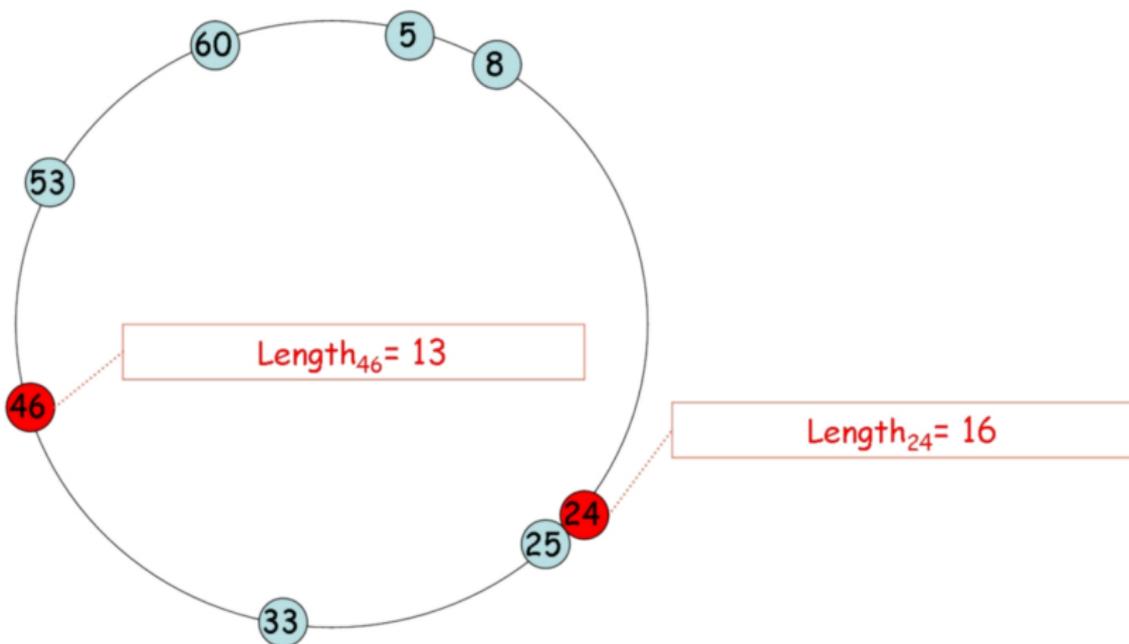
Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni



Sbilanciamento del carico: esempio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan Kolundžija

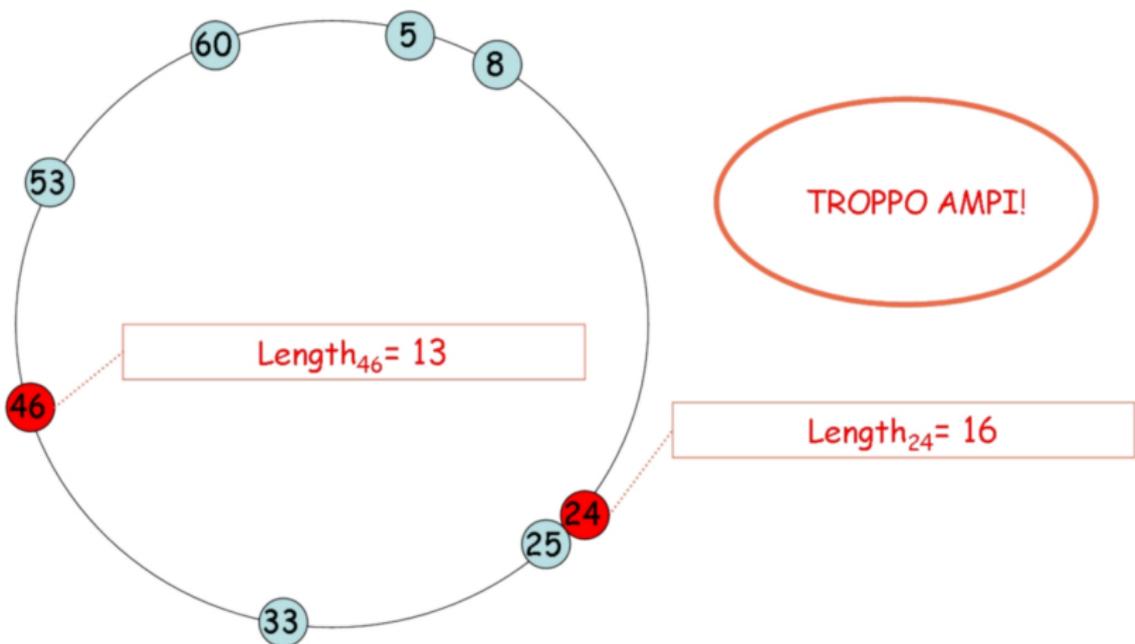
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Sbilanciamento del carico: esempio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan
Kolundžija

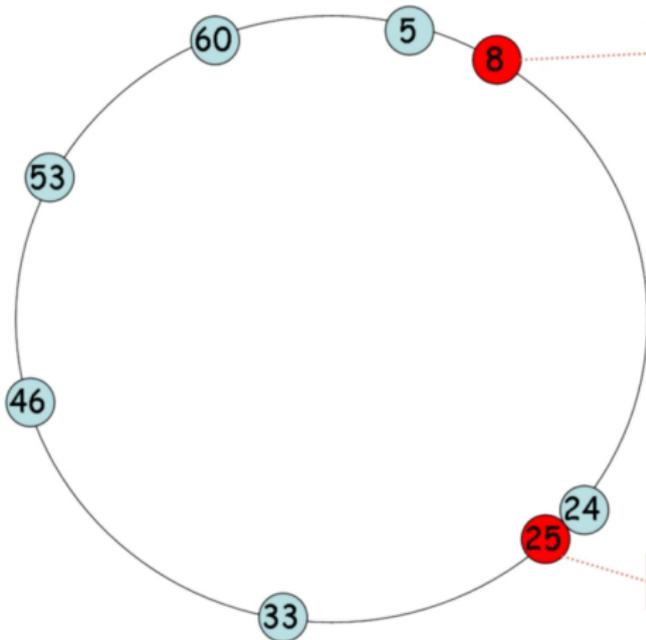
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



$Length_8 = 3$

$Length_{25} = 1$

Sbilanciamento del carico: esempio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan
Kolundžija

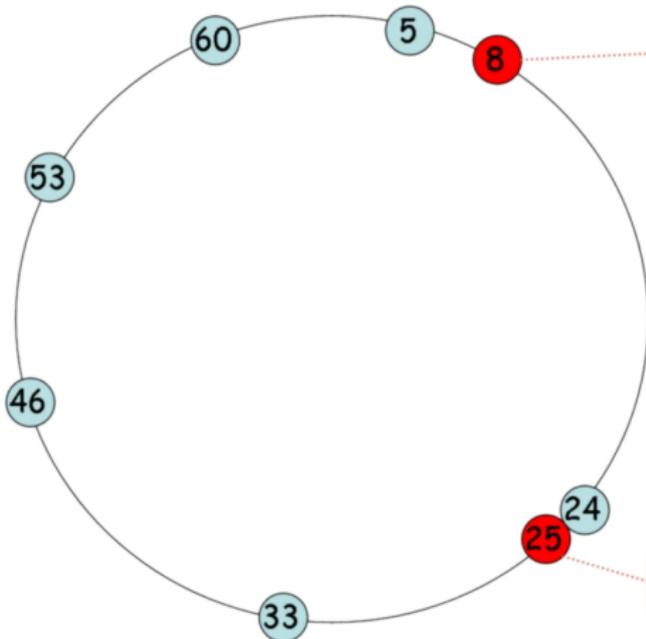
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



$Length_8 = 3$

TROPPO PICCOLI!

$Length_{25} = 1$

Sbilanciamento del carico: esempio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

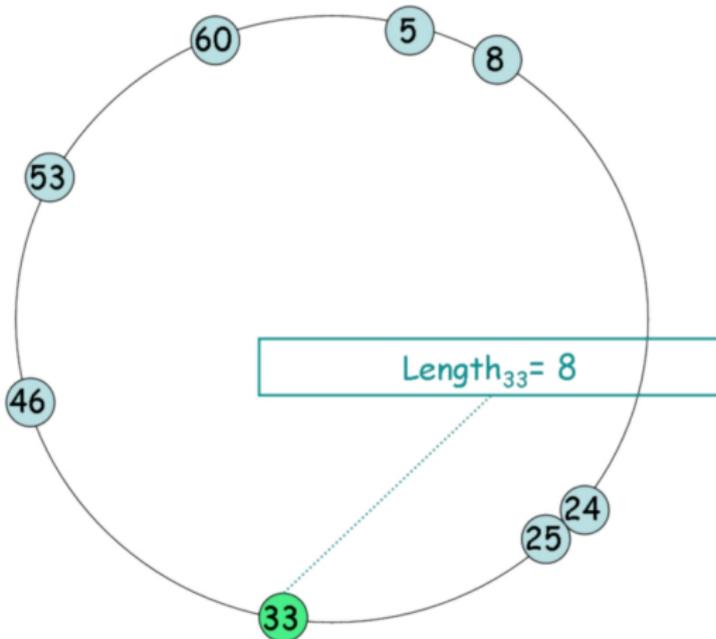
Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni



OK!

Sbilanciamento del carico: esempio

Dynamic load balancing in DHT-based networks

Matteo Tanca,
Dejan
Kolundžija

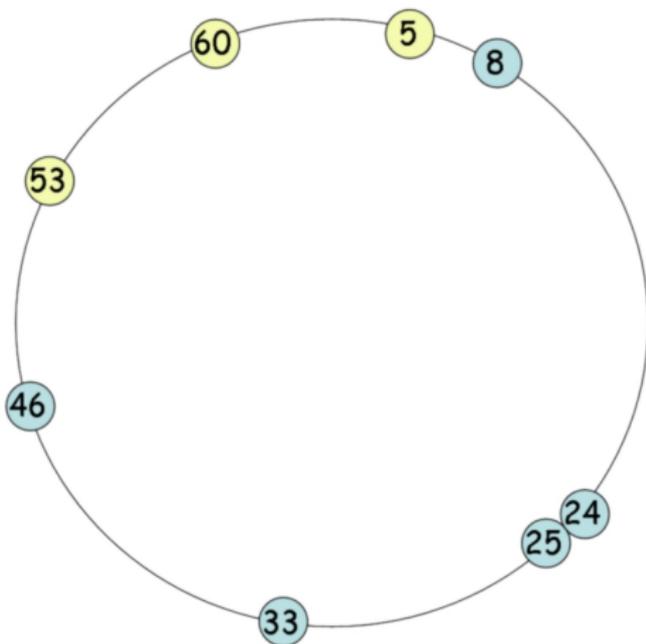
Distributed Hash Tables

CHORD

Problema dello sbilanciamento

Algoritmo di bilanciamento

Conclusioni



Intervalli tollerabili...

Caratteristiche del sbilanciamento

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- la rete P2P è dinamica - nodi entrano e escono
- gli intervalli assegnati ai peer cambiano
- obiettivo: mantenere il bilanciamento
- minimizzare il numero dei messaggi per bilanciare la struttura

Soluzioni per bilanciamento

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

■ Virtual servers

- un nodo reale gestisce $O(\log n)$ nodi virtuali
- si trasferisce un nodo virtuale da un nodo reale pesante ad un altro nodo leggero
- varie strategie: *One-to-one*, *One-to-many*, *Many-to-many*

■ Power of two choices

- si usano più funzioni hash: h_1, h_2, \dots, h_d
- inserimento - si calcolano tutte le funzioni e si sceglie il nodo con il carico più piccolo
- ricerca - calcolare tutte le funzioni o usare puntatori di redirectione

Soluzioni per bilanciamento

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

■ Heat dispersion algorithm

- l'idea è simile al processo di dispersione del calore in un ambiente
- i materiali più caldi emettono calore verso oggetti vicini
- molto efficiente

■ Simple address-space and item balancing

- ogni nodo ha un insieme fisso di $O(\log n)$ posizioni sull'anello

■ Virtually binary tree

- i nodi formano un albero binario
- la granularità è $1 + \epsilon$ mentre il costo di comunicazione è $O(\frac{1}{\epsilon})$

Algoritmo di Bienkowski e Korzeniowski

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- miglioramento della versione statica
- comunicazione limitata da $\log n$ messaggi per nodo in un half-life
- per mantenere il sistema in bilanciamento necessari $\log n$ volte più messaggi di quanti ne servano per mantenere la connettività

■ Misura di bilanciamento

$$\text{granularita (smoothness)} = \frac{\text{max lunghezza del intervallo}}{\text{min lunghezza del intervallo}}$$

- carico di un peer
 - grado di un nodo
 - congestione e dilatazione
- bisogna mantenere *smoothness* basso e costante
- quando non si gestisce lo sbilanciamento la granularità è $\Omega(n \log n)$, con probabilità costante

Assunzioni

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

- ogni nodo conosce un limite superiore Q di $\log n$
- si assume $Q = \Theta(\log n)$
- ogni nodo mantiene connessioni con $q = \alpha \cdot Q$ posizioni sull'anello
- con D segnaliamo **la dilatazione** della rete, cioè la massima lunghezza del cammino fra ogni coppia di nodi

Definizioni

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

Definition

Il peso di un intervallo I_k , $w(I_k)$, è il numero di identificatori che appartengono a I_k .

Definition

Il peso di un nodo u è il peso dell'intervallo del quale u è responsabile.

- $E[w(I_k)] = q \cdot len(I_k) \cdot n$ ci da una stima: $n_k = \frac{w(I_k)}{qlen(I_k)}$

Definizioni

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Definition

Intervallo l_k è :

- **leggero** se $w(l_k) \leq 16 \cdot q$
- **pesante** se $w(l_k) \geq 32 \cdot q$
- **medio** se $16 \cdot q \leq w(l_k) \leq 32 \cdot q$
- **molto leggero** se $w(l_k) \leq w_{min} \cdot q$ - nodi di questo tipo escono dalla rete

Ruolo dei nodi

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Definition

Un nodo leggero può essere:

- **passivo**: non fa niente finche non riceve uno stimolo - ricezione o perdita di un link
 - **attivo**: cerca nodi pesanti per condividere il loro carico
-
- un nodo pesante accetta le proposte per condividere l'intervallo che gestisce
 - un nodo medio non fa niente
 - i parametri A e F sono il massimo tempo per l'attività e la distanza d'inoltro

Algoritmo per i nodi *passive light*

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

passive light (permanently in staying state)

```
if (any change in markers)
    change activity level to A
    become active
```

Algoritmo per i nodi *active light* [1/3]

active light

```
if (locked)
    wait until unlocked
    by successor

if (predecessor is light)
    do
        perform one of the following
        selected chosen randomly:
            - let the successor lock me
              and wait until unlocked
            - try to lock the predecessor
until (predecessor locked)

p = random (0 .. 1)
```

Algoritmo per i nodi *passive light* [2/3]

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

```
P = node responsible for p
if (P is heavy and migrating to p
    does not produce a very
    light interval)
    migrate to p
else
    contact consecutively the node
    P and its  $F * D$  successors
    on the ring

    if (a contacted node R is heavy)
        migrate to the middle of
        the interval of R
unlock the predecessor
```

Algoritmo per i nodi *passive light* [3/3]

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

```
if (any change in markers)
    change activity level to A
decrease activity level
if (activity level = 0)
    become passive
```

Algoritmo: esempio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

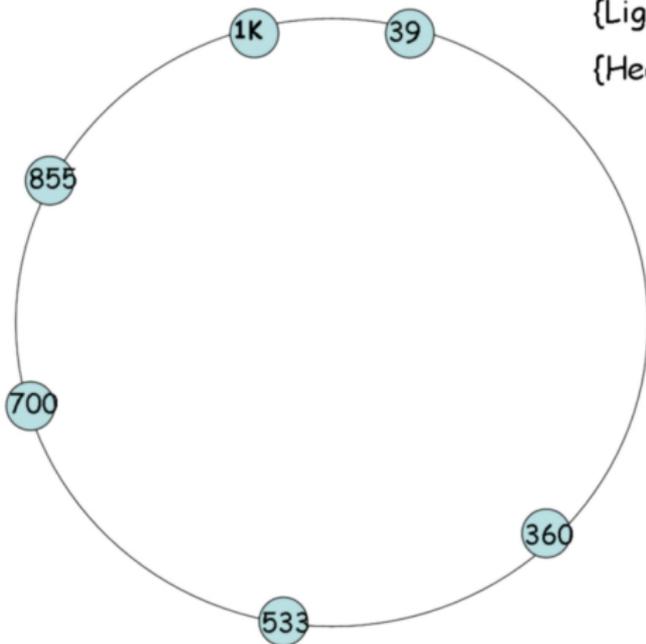
Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni



$N = 1024, \alpha = 1$

{Light Intervals} $\leq 16 * 10 = 160$

{Heavy Intervals} $\geq 32 * 10 = 320$

Algoritmo: esempio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

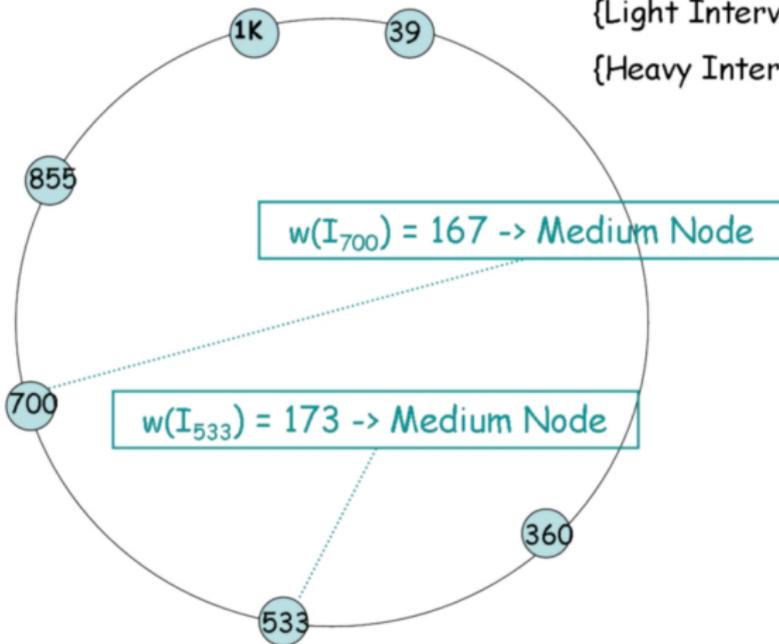
Algoritmo di
bilanciamento

Conclusioni

$N = 1024, \alpha = 1$

{Light Intervals} $\leq 16 * 10 = 160$

{Heavy Intervals} $\geq 32 * 10 = 320$



Algoritmo: esempio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

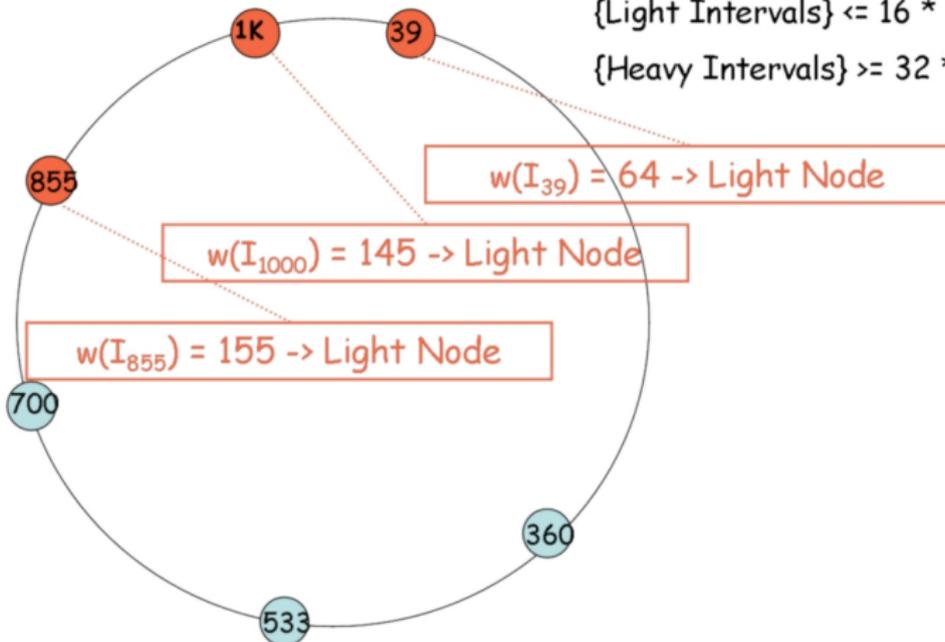
Algoritmo di
bilanciamento

Conclusioni

$N = 1024, \alpha = 1$

{Light Intervals} $\leq 16 * 10 = 160$

{Heavy Intervals} $\geq 32 * 10 = 320$



Algoritmo: esempio

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

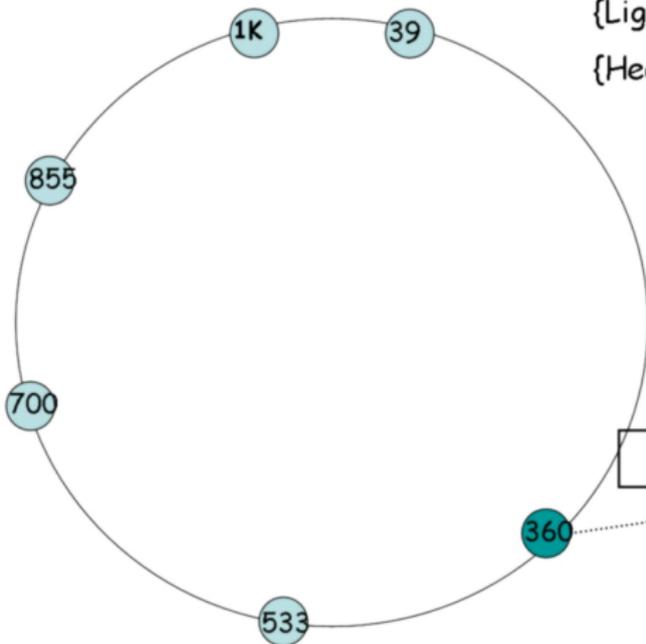
Algoritmo di
bilanciamento

Conclusioni

$N = 1024, \alpha = 1$

{Light Intervals} $\leq 16 * 10 = 160$

{Heavy Intervals} $\geq 32 * 10 = 320$



$w(I_{360}) = 321 \rightarrow$ Heavy Node

Definition

Se abbiamo n nodi nel momento t , allora:

- **doubling time** è il tempo passato da t all'arrivo di n nuovi nodi
- **halving time** è il tempo necessario perché escano dal sistema la metà dei nodi presenti
- **half-life** τ_t è il più piccolo fra doubling time e halving time
- **half-life** τ dell'intero sistema è il minimo fra tutti gli half-life degli istanti t

Teorema per le reti P2P

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Theorem

Ogni rete P2P con n nodi che rimanga connessa con alta probabilità, per ogni sequenza di entrate e uscite con half-time τ deve notificare ogni nodo con un numero medio di messaggi $\Omega(\log n)$ per un tempo τ .

Lemma importante

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

Lemma

Esistono $w_{min} < \frac{1}{2}$, l_{min} e l_{max} , tale che per ogni intervallo l_k e tempo t vale (con alta probabilità): se $w(l_k) \geq w_{min} \cdot q$, allora $l_{min} \cdot \frac{w(l_k)}{q \cdot n} \leq len(l_k) \leq l_{max} \cdot \frac{w(l_k)}{q \cdot n}$, per un α abbastanza grande.

Con questa lemma possiamo concludere che la granularita è costante con alta probabilità

Proof.

- I un intervallo con $len(I) \leq \frac{l_{min}}{n}$
- I' intervallo, tale che $I \subseteq I'$ e $len(I') = \frac{l_{min}}{n}$
- $X = w(I')$, allora $E[X] = l_{min} \cdot q = \Theta(q) = \Omega(\log n)$
- Usando limite di Chernoff si ottiene che $X = \Theta(q)$ con alta probabilità
- $w(I) \leq w(I') \Rightarrow w(I) = O(q)$
- se $w(I) \geq w_{min} \cdot q = \Theta(q) \cdot q$ allora con alta probabilità vale $len(I) \geq \frac{l_{min}}{n} = \Omega(\frac{w(I)}{q \cdot n})$
- per vedere l'altra disuguaglianza si assume $len(I) \geq \frac{l_{min}}{n}$
- $E[X] = len(I) \cdot q \cdot n = \Omega(\log n)$
- quindi con alta probabilità $w(I) = \Omega(len(I) \cdot q \cdot n)$

Conclusioni

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanciamento

Algoritmo di
bilanciamento

Conclusioni

- Le tabelle DHT hanno molte caratteristiche positive
- Esistono diverse topologie per DHT
- Vari problemi riscontrati durante l'evoluzione di un sistema DHT based
- Il problema di bilanciamento è cruciale ed esistono varie soluzioni per affrontarlo

Domande

Dynamic load
balancing in
DHT-based
networks

Matteo Tanca,
Dejan
Kolundžija

Distributed
Hash Tables

CHORD

Problema
dello sbilanci-
amento

Algoritmo di
bilanciamento

Conclusioni

???

[bie05] Marcin Bienkowski, Miroslaw Korzeniowski
**Bounding Communication Cost in Dynamic Load
Balancing of Distributed Hash Tables**
OPODIS 2005: 381-395, 2005

[p2p] RRalf Steinmetz, Klaus Wehrle
Peer-to-Peer Systems and Applications
Lecture Notes in Computer Science - Volume 3485,
Settembre 2005