Lezione n.16
DHT: LOAD BALANCING

Peer-to-Peer Systems and Applications
Capitolo 9
Laura Ricci

DHT: LOAD BALANCING

- DHT: in generale assumono che
  - Funzione hash distribuisce uniformemente gli indirizzi
  - Ogni nodo ha lo stesso carico
  - Non è necessario un algoritmo di load balancing

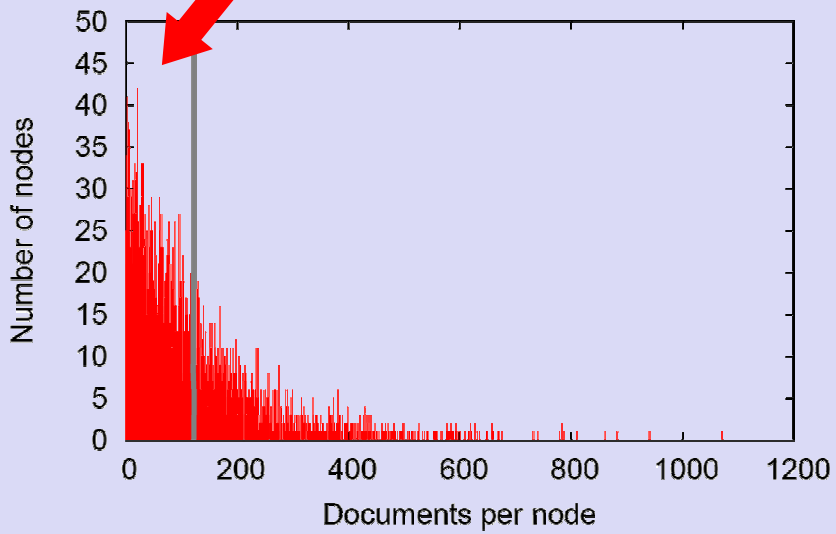
- **Distribuzione uniforme**
  - Nodi all'interno dello spazio degli indirizzi
  - Dati nei nodi

- Ma tutto questo accade realmente?
  - Analisi delle distribuzioni mediante simulazione

CHORD SENZA LOAD BALANCING

- Analisi della distribuzione dei dati
- Esempio
  - Parametri
    - 4,096 nodi
    - Spazio degli indirizzi di m=22 bits
    - Numero massimo di documenti e/o nodi = 2^{22}=4.194.304

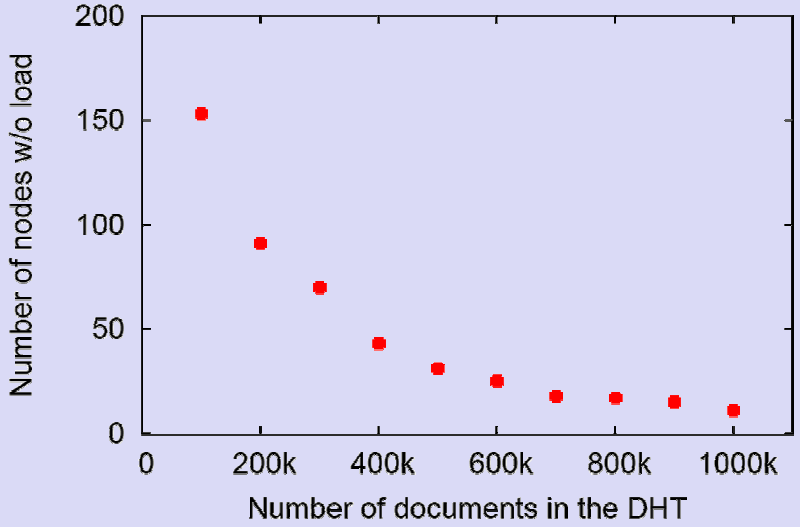
Distribuzione Ottima dei Documenti sui Nodi del Sistema



  - Esperimento
    - Più simulazioni, le chiavi per nodi e documenti generate in modo random, in media 500,000 documenti
    - Ottimo = ~122 documenti per nodo
    - Grafico: mostra quanti nodi (asse y) memorizzano un certo numero di documenti (asse x)

CHORD SENZA LOAD BALANCING

- Numero di nodi che non memorizzano documenti
  - Parametri
    - 4,096 nodi
    - da 100.000 a 1.000.000 di documenti
  - Alcuni nodi senza carico



- Perchè il carico non è bilanciato?
- Sefinizione di algorimi di bilanciamento del carico

BILANCIAMENTO DEL CARICO: PROPOSTE

- Alcune Proposte
  - Power of Two Choices (Byers et. al, 2003)
  - Virtual Servers (Rao et. al, 2003)
  - Thermal-Dissipation-based Approach (Rieche et. al, 2004)
  - .....
- Argomento di ricerca.....

POWER OF TWO CHOICES

- Una sola funzione hash per tutti i nodi
  - h_0
- Più funzioni hash per i dati
  - $h_1, h_2, h_3, \ldots h_d$
  - quando si inserisce un dato x , viene calcolato il risultato dell'applicazione di tutte le funzioni $h_1(x), h_2(x), h_3(x), \ldots h_d(x)$ al dato x
- Due opzioni
  - Il dato viene memorizzato sul nodo che presenta il carico minore
  - quando si ricerca il dato vengono di nuovo applicate le d funzioni
  - $h_1(x), h_2(x), h_3(x), \ldots h_d(x)$ e si effettua il look up su tutti i nodi individuati

oppure

  - Il dato viene memorizzato su un nodo N e tutti gli altri nodi puntano a N
  - Il look up avviene ricalcolando solo una tra le d funzioni $h_1(x), h_2(x), h_3(x), \ldots h_d(x)$

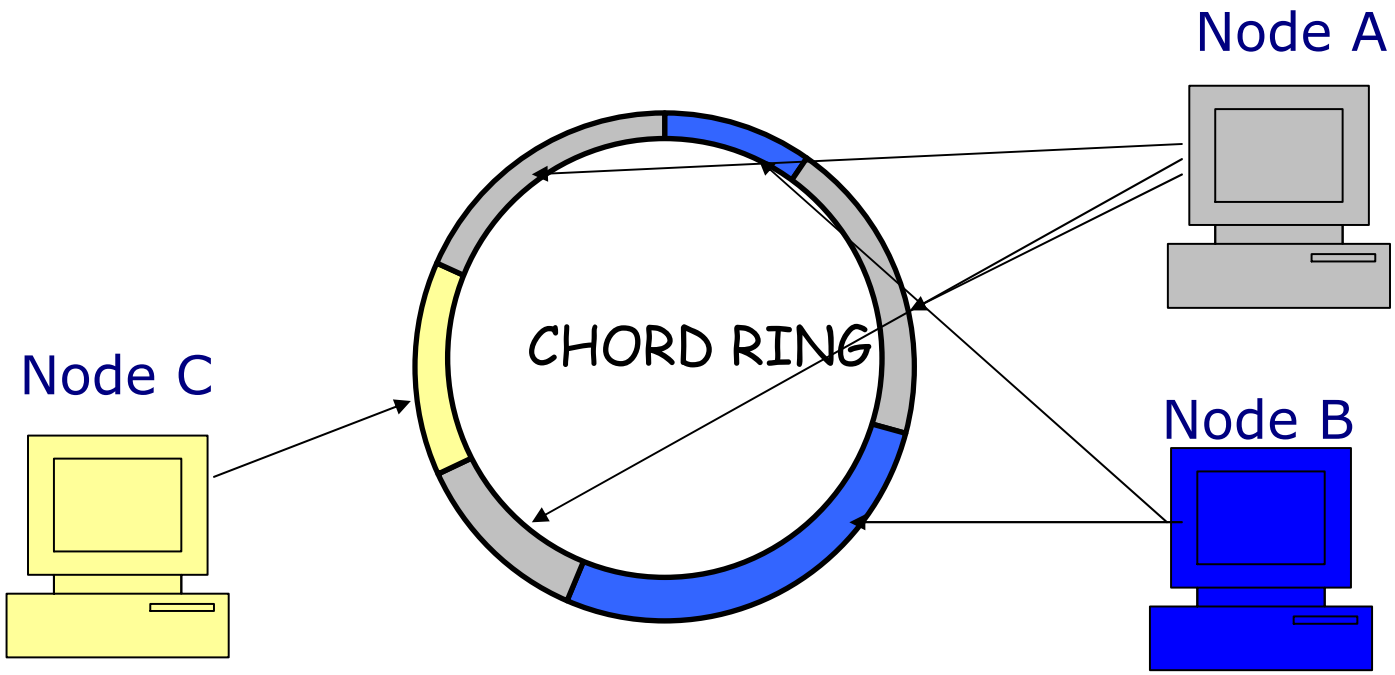
POWER OF TWO CHOISES

- Vantaggio
  - semplicità

- Svantaggio
  - Overhead al momento dell'inserzione dei dati
  - Se si utilizzano i puntatoti
    - gestione dei puntatori
  - Se non si utilizzano i puntatori
    - overhead per ogni ricerca

VIRTUAL SERVERS

- Ogni nodo fisico è responsabile di più di una partizione dello spazio degli indirizzi
- Ogni nodo fisico gestisce un insieme di virtual servers
- La figura si riferisce a Chord



VIRTUAL SERVERS

.Un intero virtual server può essere trasferito da un nodo ad un altro nel caso in cui il carico debbe essere bilanciato

Heavy Nodes = nodi con una quantità di dati maggiore di un valore soglia

Light Nodes = nodi con meno dati del valore soglia

.Scopo dell'algoritmo di bilanciamento del carico = diminuire il numero di heavy nodes spostando i virtual servers verso i nodi light

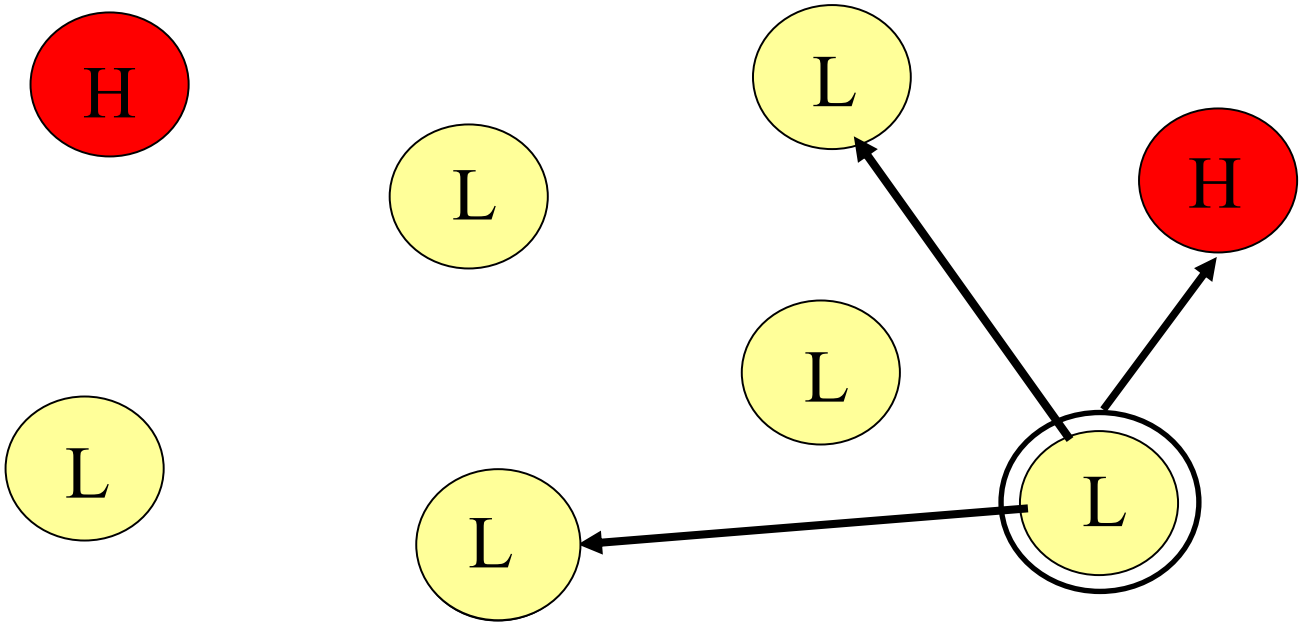
.Dato un heavy node h ed un light node l, la scelta del virtual server da trasferire da h ad l è guidata dalle seguenti regole:

- il trasferimento di v da h ad l non rende l un heavy node
- v è il virtual server 'meno carico' che rende h un light node
- se non esiste v che rende h un light node, si trasferisce il virtual server 'più carico' da h ad l

VIRTUAL SERVERS ONE TO ONE

One-to-One: lights nodes ricercano heavy nodes

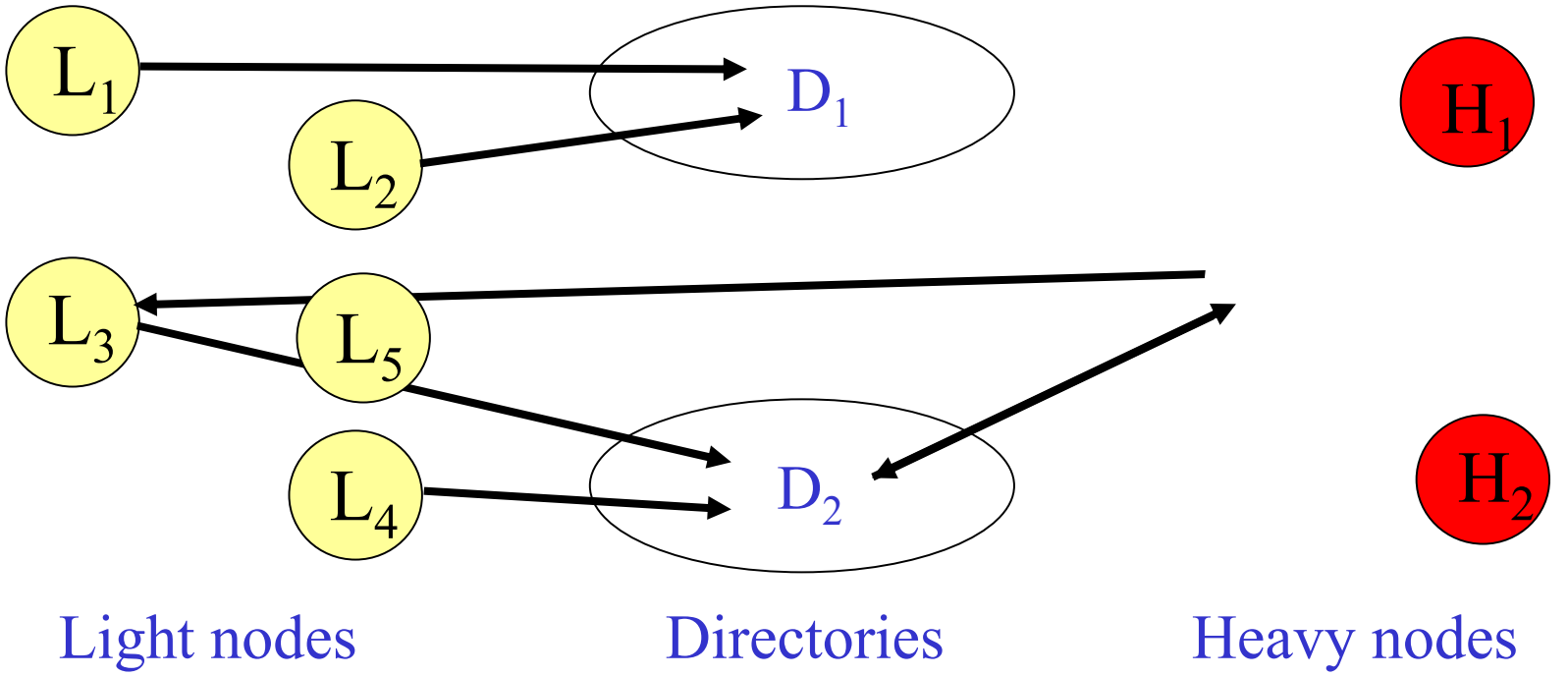
- Ogni light node sceglie periodicamente un ID, in modo random
- Contatta il nodo x responsabile di quell'ID
- Accetta del carico da x se x è un heavy node



[Rao 2003]

VIRTUAL SERVERS: ONE TO MANY

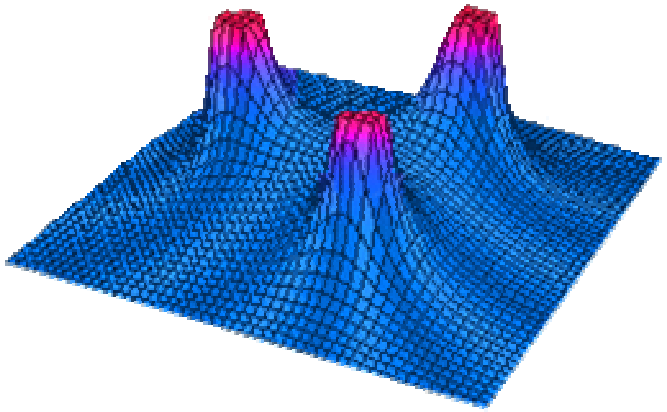
- I light nodes riportano il loro carico in alcune *directories*
- Gli heavy nodes reperiscono l'informazione dalle directorie
  - Ogni heavy node contatta il light node che può accettare il suo carico in eccesso



[Rao 2003]

THERMAL DISSIPATION

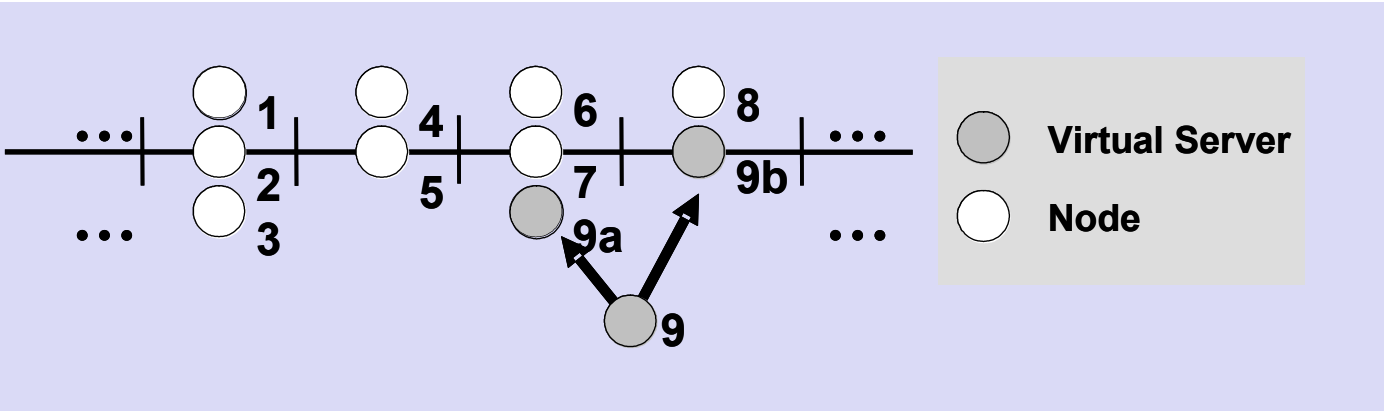
- I dati vengono spostati tra i peers
  - simile al processo di diffusione del calore
  - il carico in eccesso passa da un nodo sovraccarico (heavy node) ad uno più scarico (light node)
  - carico come energia



- Approccio generale applicabile a diverse DHT (Chord, CAN)
- trasferimento di intervalli completi dello spazio degli indirizzi da un nodo ad un altro
- Obiettivi dell'approccio: Load Balancing + Affidabilità, replicazione automatica dei dati

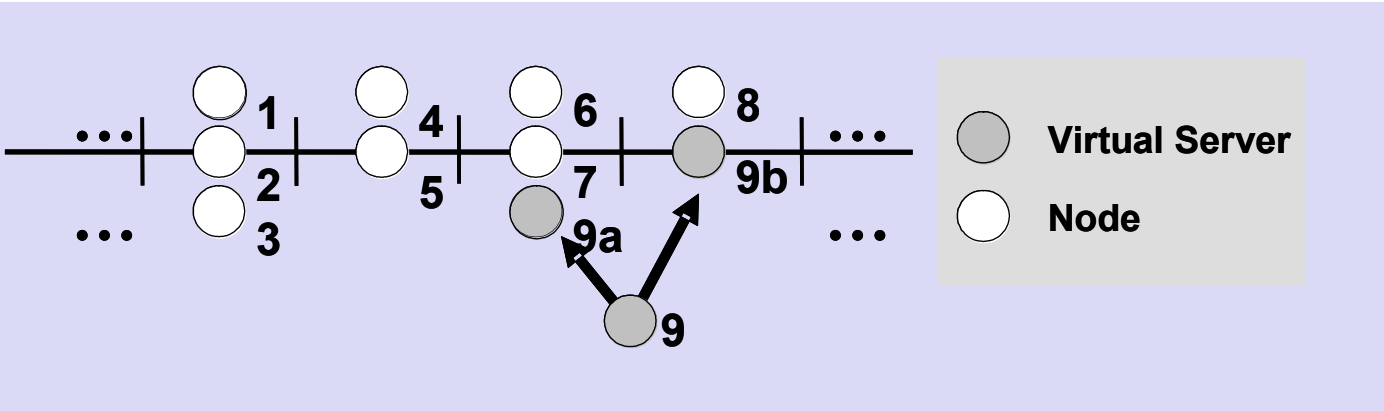
THERMAL DISSIPATION

- Ogni intervallo della DHT è gestito da più nodi
- Ogni nodo mantiene puntatori (es in Chord entrate della fineger table, in CAN puntatori ai vicini) a tutti i nodi appartenenti allo stesso intervallo
- Replicazione: ogni nodo appartenente ad un intervallo gestisce tutti i dati di quell'intervallo
- Quando un nodo si aggiunge ad un intervallo copia dagli altri tutti i dati appartenenti a quell'intervallo
- Nell'esempio: i nodi 1,2,3 gestiscono dati appartenenti allo stesso intervallo
- La definizione di un insieme di nodi gestori per ogni intervallo aumenta la affidabilità del sistema



THERMAL DISSIPATION

- Quando un dato viene inserito in un intervallo viene replicato su tutti i nodi che gestiscono quell'intervallo
- Possibilità di definire servers virtuali
- Un nodo con maggior potenza computazionale può gestire servers virtuali che gestiscono dati in intervalli diversi
- Esempio: il nodo 9 gestisce i servers virtuali 9a e 9b che gestiscono dati in due diversi intervalli



THERMAL DISSIPATION

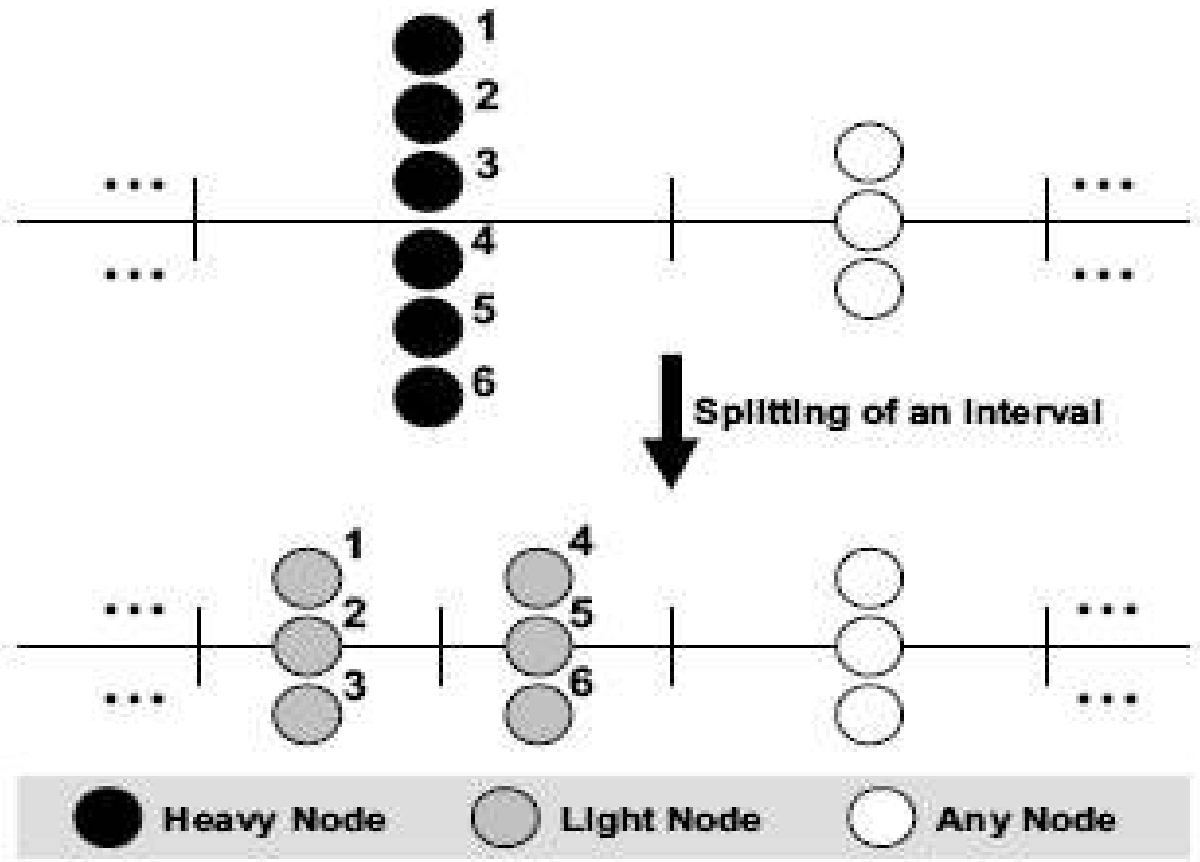
- Poiché esistono più nodi che gestiscono lo stesso intervallo, è possibile distribuire il carico dovuto alle queries su questi nodi
- Esempio, in Chord ogni nodo può scegliere i nodi da inserire nelle finger tables secondo criteri diversi di selezione
  - random
  - in base al carico
  - in base alla prossimità fisica del nodo
  - in base alla latenza

THERMAL DISSIPATION

- Si stabilisce un valore intero f, che indica il numero minimo f di nodi che devono gestire i dati appartenenti ad un intervallo
- Bilanciamento del carico basato sulle seguenti regole.
  - se esistono 2f nodi diversi nello stesso intervallo ed i nodi sono sovraccarichi
    - L'intervallo viene diviso, ogni nodo si prende carico di una parte di dati
  - se esistono più di f ma meno di 2f nodi
    - alcuni nodi possono essere rilasciati ad altri intervalli
  - non esistono più di f nodi nell'intervallo ed alcuni nodi sovraccarichi shift dei bordi degli intervalli tra nodi vicini

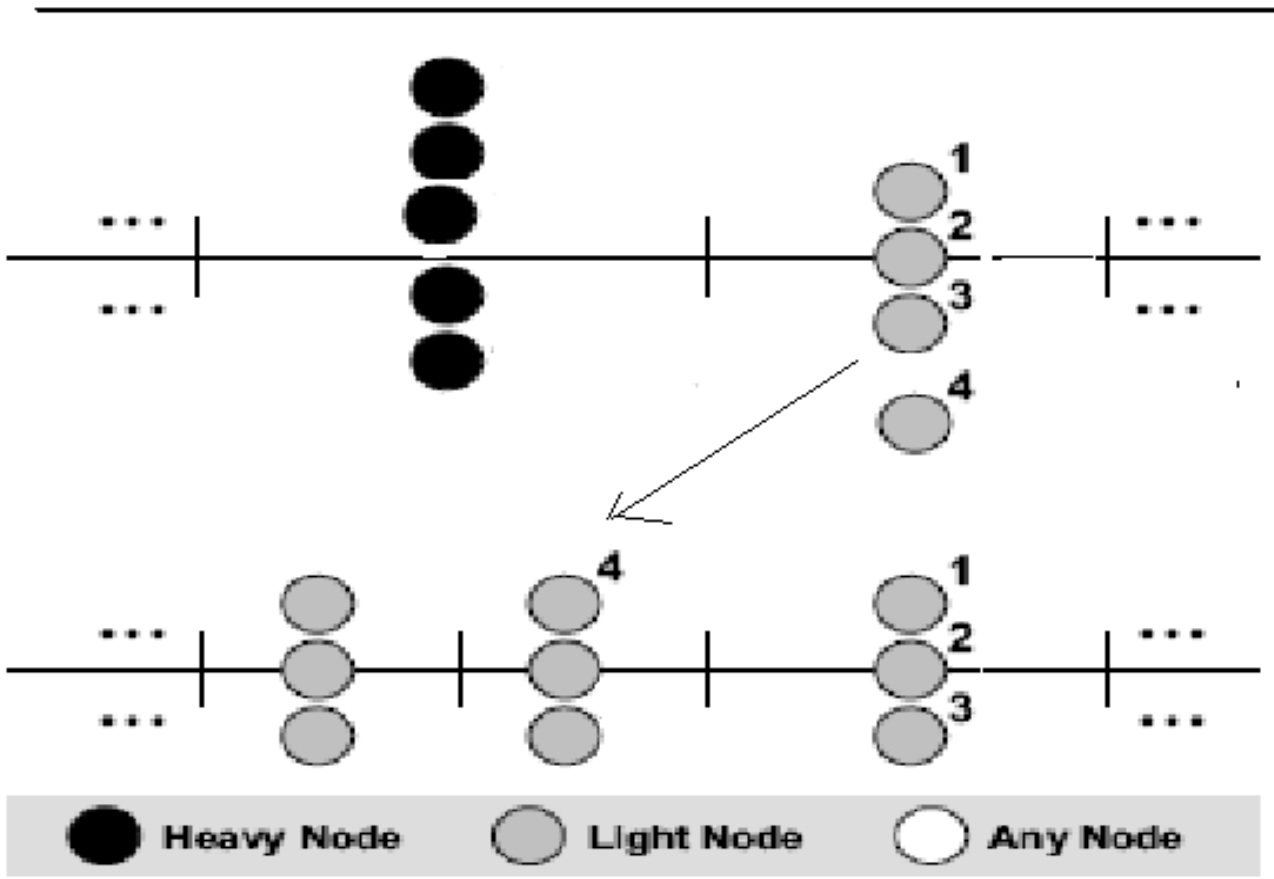
THERMAL DISSIPATION

- *Caso 1:* 2f nodi in un intervallo sovraccarichi



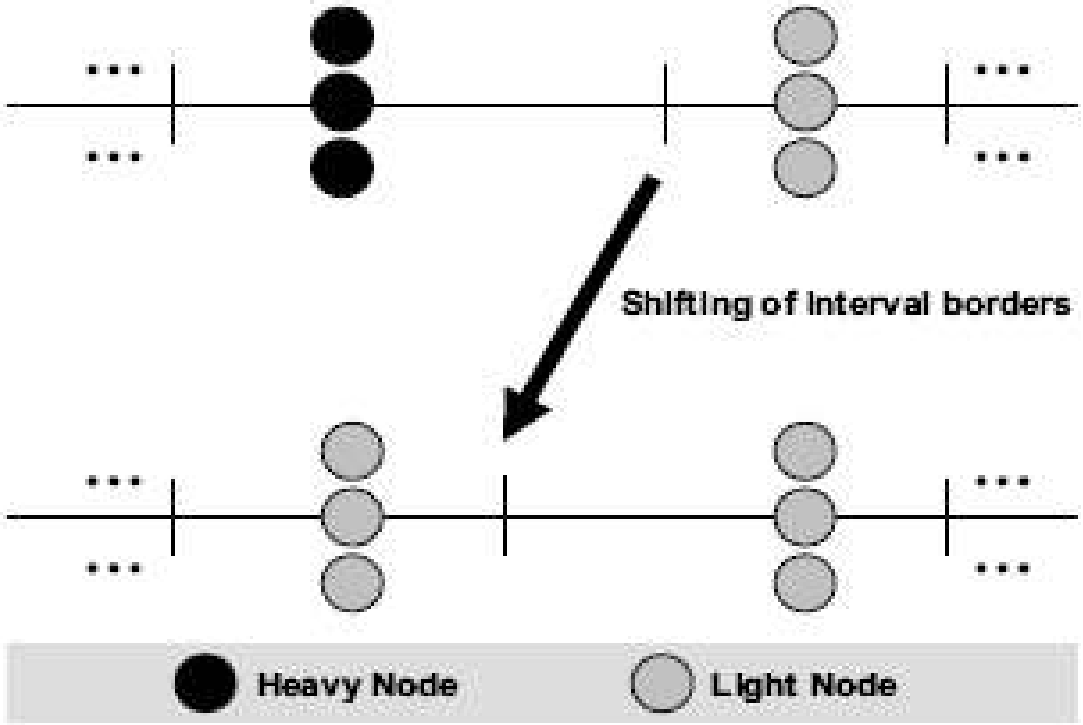
THERMAL DISSIPATION

- Caso 3: più di f (f=3) nodi non sovraccarichi in un intervallo



THERMAL DISSIPATION

- *Caso 3:* Shift dei bordi di un intervallo



DHT: PROPOSTE DI TESI

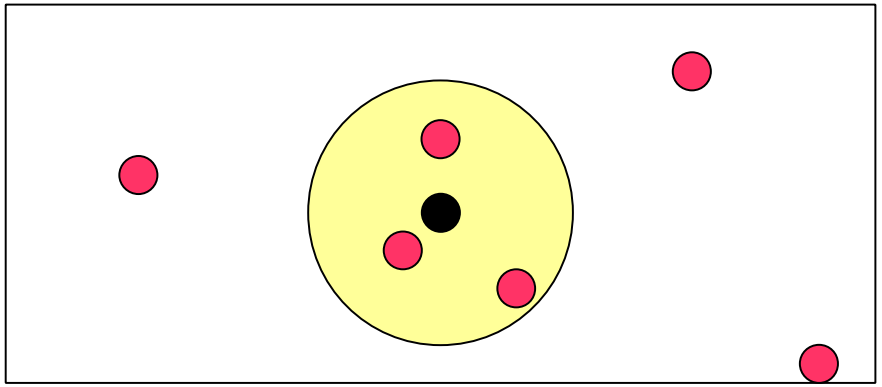
- Tecniche di bilanciamento del carico in DHT
- Esempio: utilizzare ORB (Orthogonal Recursive Bisection) per le DHT
- Implementazione di query complesse, esempio range queries
- Uso di DHT nell'ambito di ambienti virtuali distribuiti

DISTRIBUTED VIRTUAL ENVIRONMENTS

- DVE (Distributed Virtual Environments)
  - Massively Multiplayer Online Games (MMOG) (esempio World of Warcraft, Seconf Life,...)
  - Simulazioni distribuite: simulazioni militari, simulazioni che richiedono un grosso carico computazionale,...
- La diffusione di questo tipo di applicazioni dipende da
  - Scalabiltà: Permettere a milioni di utenti di giocare contemporaneamente
  - Diminuzione dei costi
- Problemi legati allo sviluppo di questo tipo di applicazioni:
  - Scalabilità
  - Consistenza
  - Interattività
  - Sicurezza, cheating,....
  - Persistenza dello stato

DISTRIBUTED VIRTUAL ENVIRONMENTS

- Ogni partecipante è rappresentato da un avatar ed accede all'ambiente virtuale mediante un PC
- La visibilità di un nodo è in generale ridotta
- Introduzione del concetto di Area di Interesse di un nodo



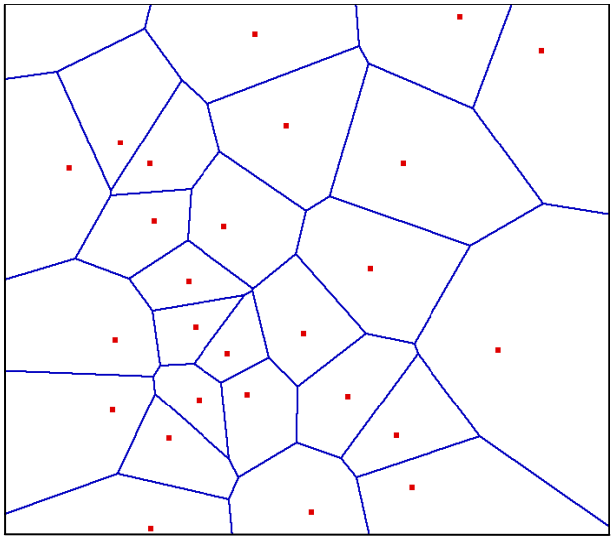
- Quando un partecipante si muove o esegue un'azione (esempio spara, si muove, interagisce), l' azione deve essere comunicata solo ai vicini che fanno parte della sua area di interesse

- Problema: Definizione di un supporto di comunicazione in grado di supportare questa comunicazione basata sul concetto di vicinato

DISTRIBUTED VIRTUAL ENVIRONMENTS

- L'overlay network deve collegare ogni giocatore a tutti i suoi vicini contenuti all'interno della sua area di interesse
- L'area di interesse di un giocatore cambia quando il giocatore si muove all'interno dell'ambiente virtuale
- La overlay network deve essere altamente dinamica
- Approccio proposto: ogni nodo determina i suoi vicini mediante la costruzione di diagrammi di Voronoi
- Diagramma di Voronoi 2D: dati n punti in un paino 2D , un Diagramma di Voronoi divide il piano in n regioni disgiunte in modo che tutti i punti appartenenti ad una regione sono più vicini al punto associato alla regione di qualsiasi altro punto del piano

DISTRIBUTED VIRTUAL ENVIRONMENTS



- I diagrammi di Voronoi possono essere utilizzati per identificare i vicini di un certo nodo

- costruzione dinamica di diagrammi di Voronoi permette la costruzione dinamica dell'overlay

- il diagramma viene modificato mano a mano che l'avatar si muove

DISTRIBUTED VIRTUAL ENVIRONMENTS

- Gestione oggetti passivi: ad esempio armi, pozioni
- Definizione di gestori
  - Replicazione completa
  - Elezione di un peer coordinatore (esempio uno per ogni regione)
    - Cosa succede nel caso in cui quel peer si disconnette volontariamente dalla rete oppure è soggetto ad un crash?
  - Definizione di un sottoinsieme di coordinatori
- Consistenza: mantenere uno stato consistente nel caso di accessi concorrenti
- Consenso Distribuito

DVE: PROPOSTE DI TESI

Proposte di tesi e di tirocinio:

- Utilizzo di DHT per la gestione di oggetti passivi (esempio Bamboo)
- Cheating
- Definizione di modelli di consistenza
  - Consistenza dello stato degli oggetti passivi
  - Consistenza percettiva: posizione degli avatars sui diversi hosts
- Definizione di algoritmi distribuiti basati su tali modelli
- Approcci ibridi client server/P2P