

MULTI-ATTRIBUTE, RANGE QUERIES IN P2P SYSTEMS

Problemi con reti DHT

Problema 1: Dinamicità

- ▣ Alcuni sistemi potrebbero essere caratterizzati da oggetti con attributi il cui valore può cambiare con alta dinamicità
- ▣ I sistemi DHT hanno la necessità di dover reindicizzare gli oggetti i cui valori sono cambiati
- ▣ Questo fatto può causare un alto overhead dovuto all'alto numero di messaggi che devono essere scambiati

Problemi con reti DHT

Problema 2: Multi-Attribute

- ▣ Molti sistemi mantengono una DHT separata per ogni tipo di attributo
- ▣ Questo tipo di soluzione può non scalare all'aumentare del numero di attributi, con il relativo aumento dei costi di mantenimento di ogni layer

Problemi con reti DHT

Problema 3: Range Queries

- ▣ I vantaggi delle reti DHT possono diventare meno rilevanti in caso di query con range molto ampi o aperti

Problemi con reti DHT

- ▣ Molti di questi problemi sono legati alla forte strutturazione della rete
- ▣ Si vuole studiare la possibilità di avere un sistema meno strutturato in grado di trattare efficientemente
 - Query multi-attributo con range anche ampi o aperti
 - Update frequenti dovuti a valori molto volatili

Indicizzazione non DHT

- ▣ Nei sistemi non DHT i nodi che fare il forwarding di una query hanno due possibilità:
 1. Fare un flooding non controllato a tutta la rete
 2. Trovare un modo per passare la query a vicini in qualche modo “promettenti”

Routing Indices

- I Routing Indices (RI) sono un meccanismo di indicizzazione distribuito che consente di dare una “direzione” verso una risorsa, piuttosto che la sua effettiva posizione

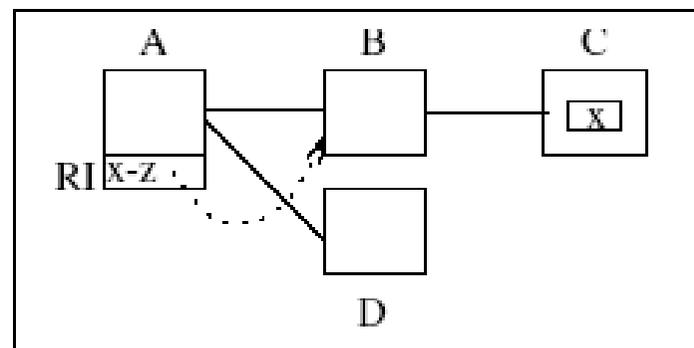


Figure 1. Routing Indices

Routing Indices

- ▣ L'obiettivo dei Ri è quello di consentire a un nodo di selezionare i “migliori” vicini a cui mandare una query
- ▣ Un RI è una struttura dati che, data una query, ritorna una lista di vicini rankati in base alla loro similarità con la query

Routing Indices

- Ogni nodo mantiene un indice locale identificare velocemente se un peer è in grado di rispondere ad una query
- Mantiene inoltre un indice per ogni vicino che indica
 - il numero di documenti presenti lungo il path che parte da quel vicino
 - Il numero di documenti per topic

| Path | # docs | Documents with topics: | | | |
|----------|--------|------------------------|-----|-----|-----|
| | | DB | N | T | L |
| <i>B</i> | 100 | 20 | 0 | 10 | 30 |
| <i>C</i> | 1000 | 0 | 300 | 0 | 50 |
| <i>D</i> | 200 | 100 | 0 | 100 | 150 |

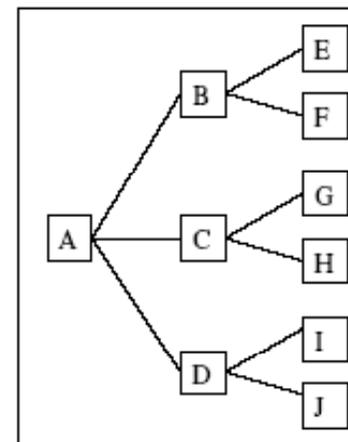


Figure 3. A Sample Compound RI

Routing Indices

- Una limitazione dei RI è che non tengono in considerazione il costo in termini di hop necessario per raggiungere una risorsa

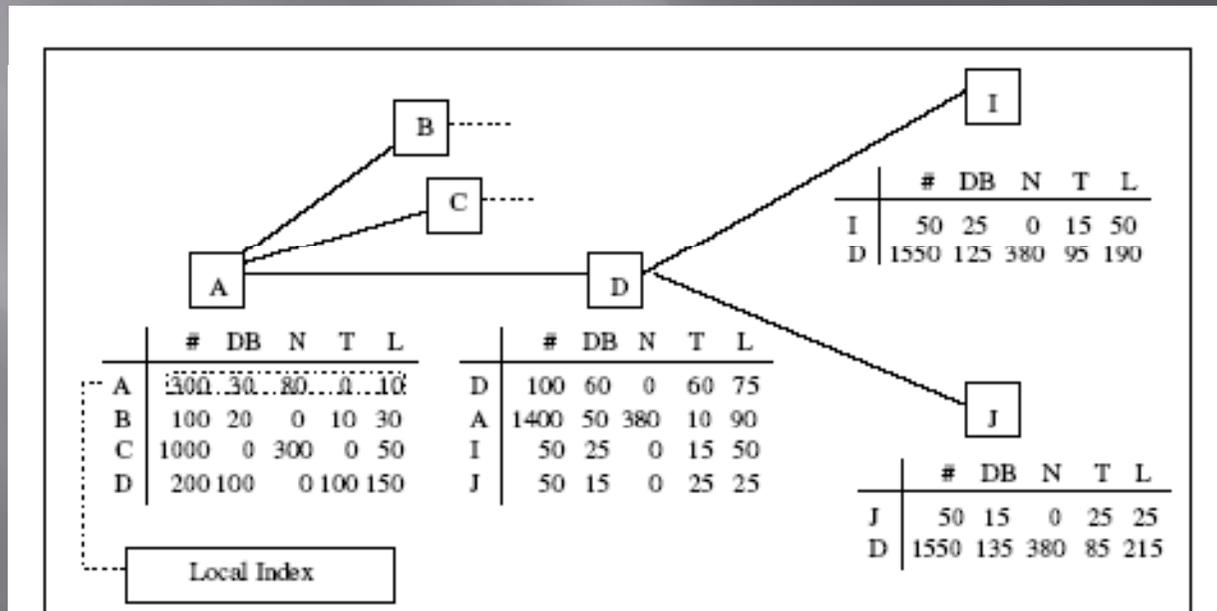
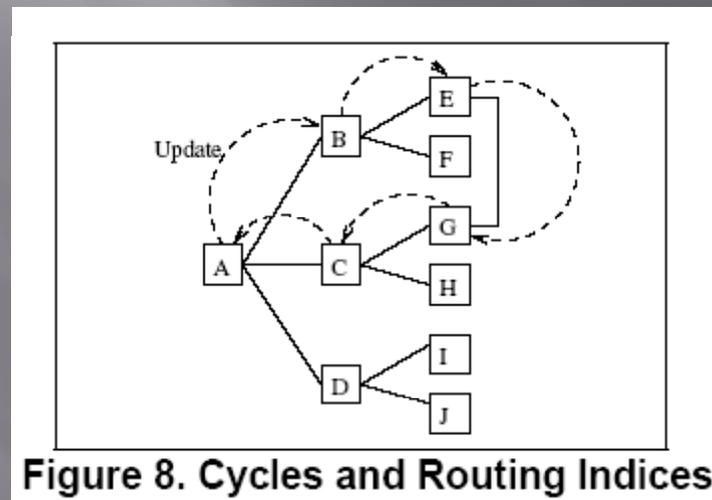


Figure 4. Routing Indices

Routing Indices

- Le reti P2P possono organizzate in topologie che includono dei cicli.



- Per i RI questo aspetto può essere un problema, dato che potrebbe portare ad una replicazione indesiderata delle informazioni

Routing Indices

- ▣ Ci sono tre approcci usati generalmente per trattare i cicli:
 - No-op solution: non ci sono cambiamenti all'algoritmo dei RI. Funziona solo con reti prive di loop o con speciali accorgimenti
 - Cycle avoidance solution: non è consentito ai nodi di creare una connessione di "update" con altri nodi se questa può creare un ciclo
 - Cycle detection and recovery: delle comunicazioni sono effettuate ciclicamente solo per verificare se esistono nodi dopo che la loro formazione. Nel caso vengono presi dei provvedimenti per eliminarli

Routing Indices

Riferimenti

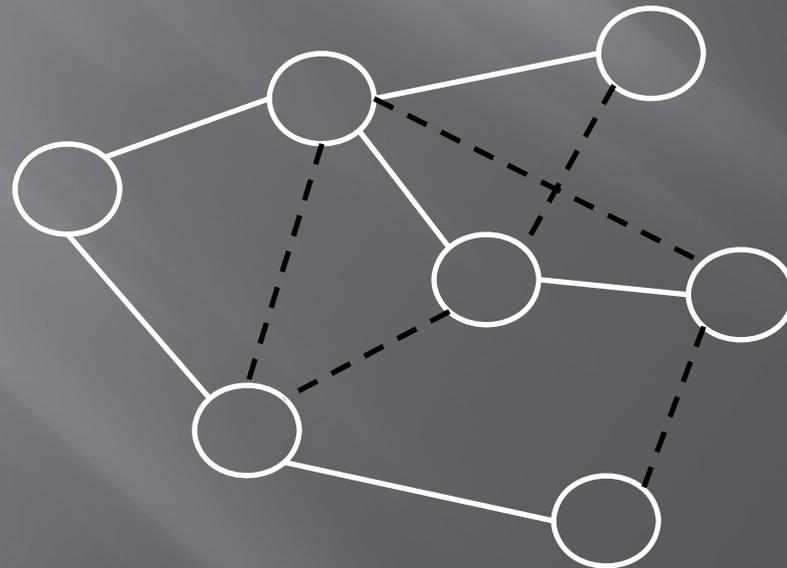
- ▣ Arturo Crespo, Hector Garcia-Molina “*Routing Indices For Peer-to-Peer Systems*”, 22nd International Conference on Distributed Computing Systems (ICDCS'02), July 2-5, 2002 - Vienna, Austria

Tree Vector Indices

- ▣ A differenza delle reti DHT, ogni peer mantiene gli indici delle proprie risorse
- ▣ Ogni attributo viene gestito separatamente
- ▣ Gli indici locali vengono usati per costruire dei RI globali

Tree Vector Indices

- ▣ I nodi della rete sono organizzati in una topologia ad albero
- ▣ Questa topologia rappresenta uno spanning tree delle connessioni fisiche tra i nodi



Tree Vector Indices

- ▣ A differenza delle reti DHT, ogni peer mantiene gli indici delle proprie risorse
- ▣ Ogni attributo viene gestito separatamente
- ▣ Gli indici locali vengono usati per costruire dei RI globali

Local Indices

- ▣ Gli indici sono divisi per attributo
- ▣ Ogni risorsa ha un indice per ognuno degli attributi che la caratterizzano
- ▣ L'indice locale di un peer è costituito dall'insieme degli indici delle sue risorse

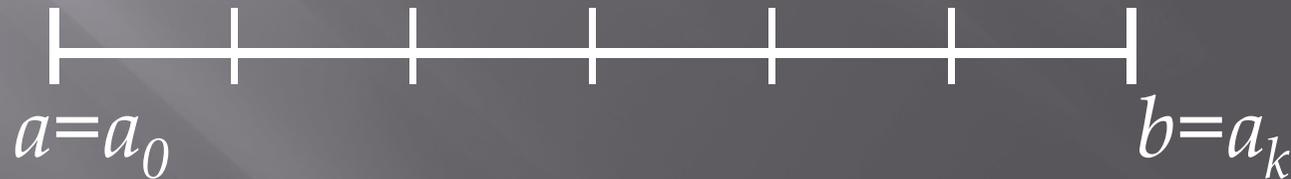
Local Indices

- ▣ Sia R una risorsa locale di un peer P
- ▣ Sia A un attributo della risorsa R e sia v_A il valore di A per R
- ▣ Sia $[a,b]$ il dominio dei valori di A



Local Indices

- ▣ Sia R una risorsa locale di un peer P
- ▣ Sia A un attributo della risorsa R e sia v_A il valore di A per R
- ▣ Sia $[a,b]$ il dominio dei valori di A
- ▣ L'intervallo $[a,b]$ viene suddiviso in k sottointervalli



Local Indices

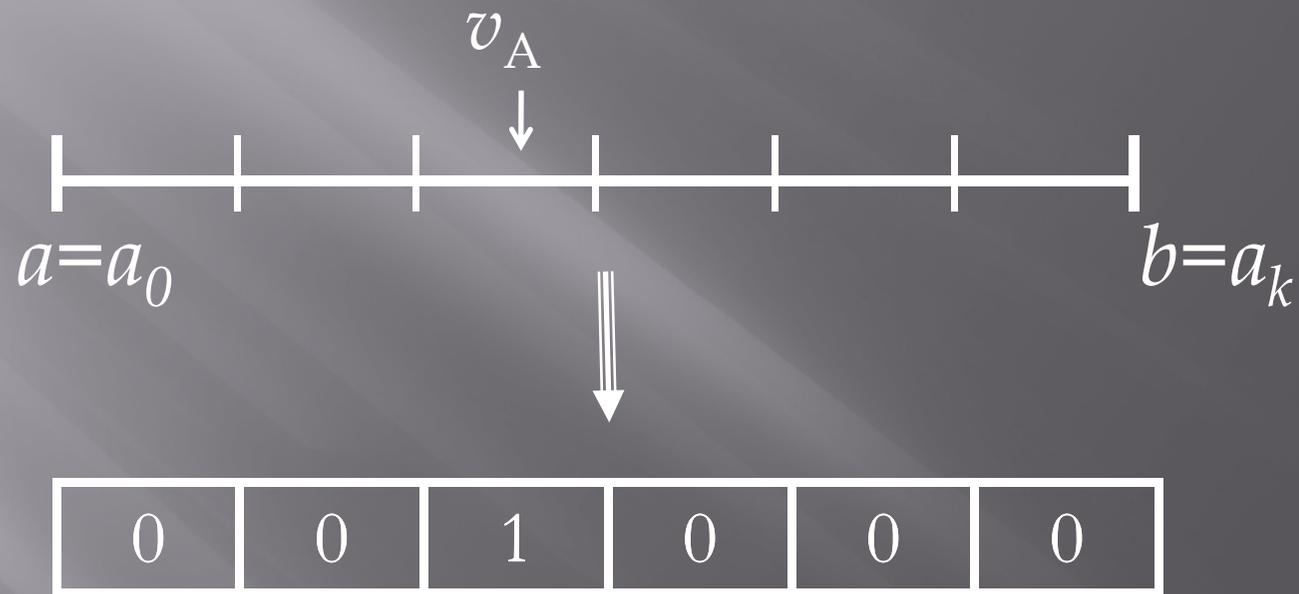
- Il valore v_A viene codificato in un vettore binario di k bit

$$\text{BitIdx}(v_A) = (b_0, b_1, \dots, b_{k-1})$$

tale che

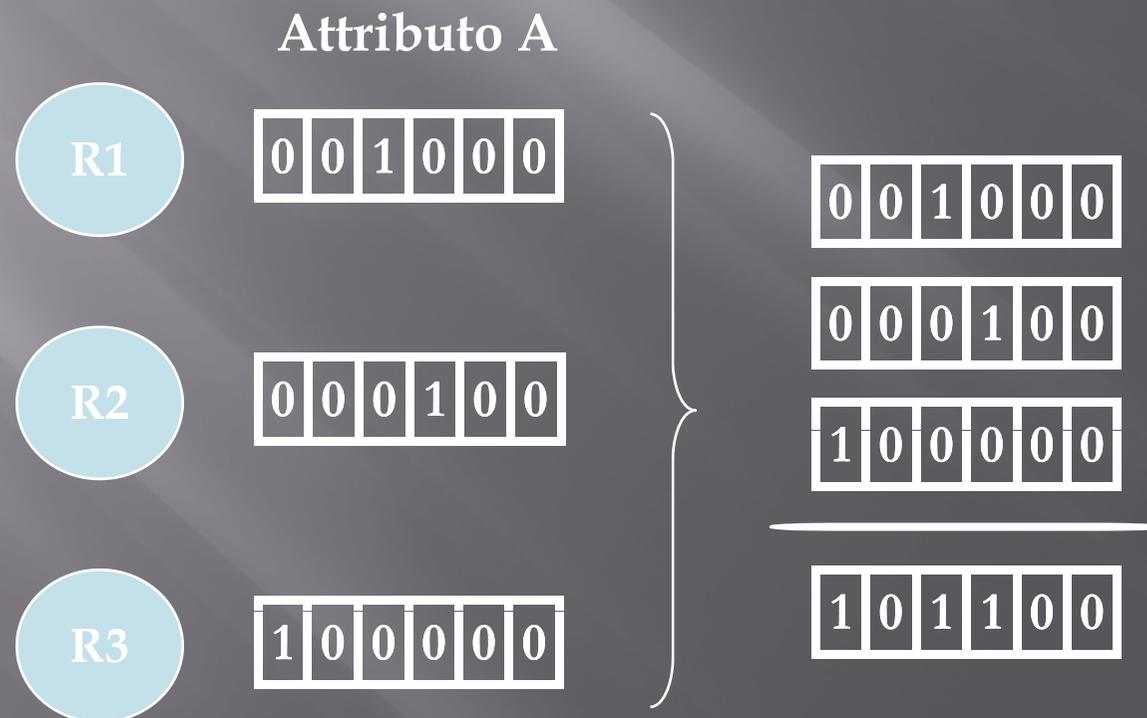
$$b_i \begin{cases} 1 & v_A \in [a_i, a_{i+1}) \\ 0 & \text{altrimenti} \end{cases}$$

Local Indices



Local Indices

- ▣ L'indice locale del peer P per l'attributo A viene computato come l'OR logico dell'indice di tutte le sue risorse



Routing Indices

- ▣ Gli indici locali danno un'indicazione sintetica di dove cadono i valori delle risorse di un peer
- ▣ I RI consentono di dare una visione sintetica dei valori di tutte le risorse di un intero ramo della rete

Routing Indices

- ▣ Gli indici locali danno un'indicazione sintetica di dove cadono i valori delle risorse di un peer
- ▣ I RI consentono di dare una visione sintetica dei valori di tutte le risorse di un intero ramo della rete

Routing Indices

Siano

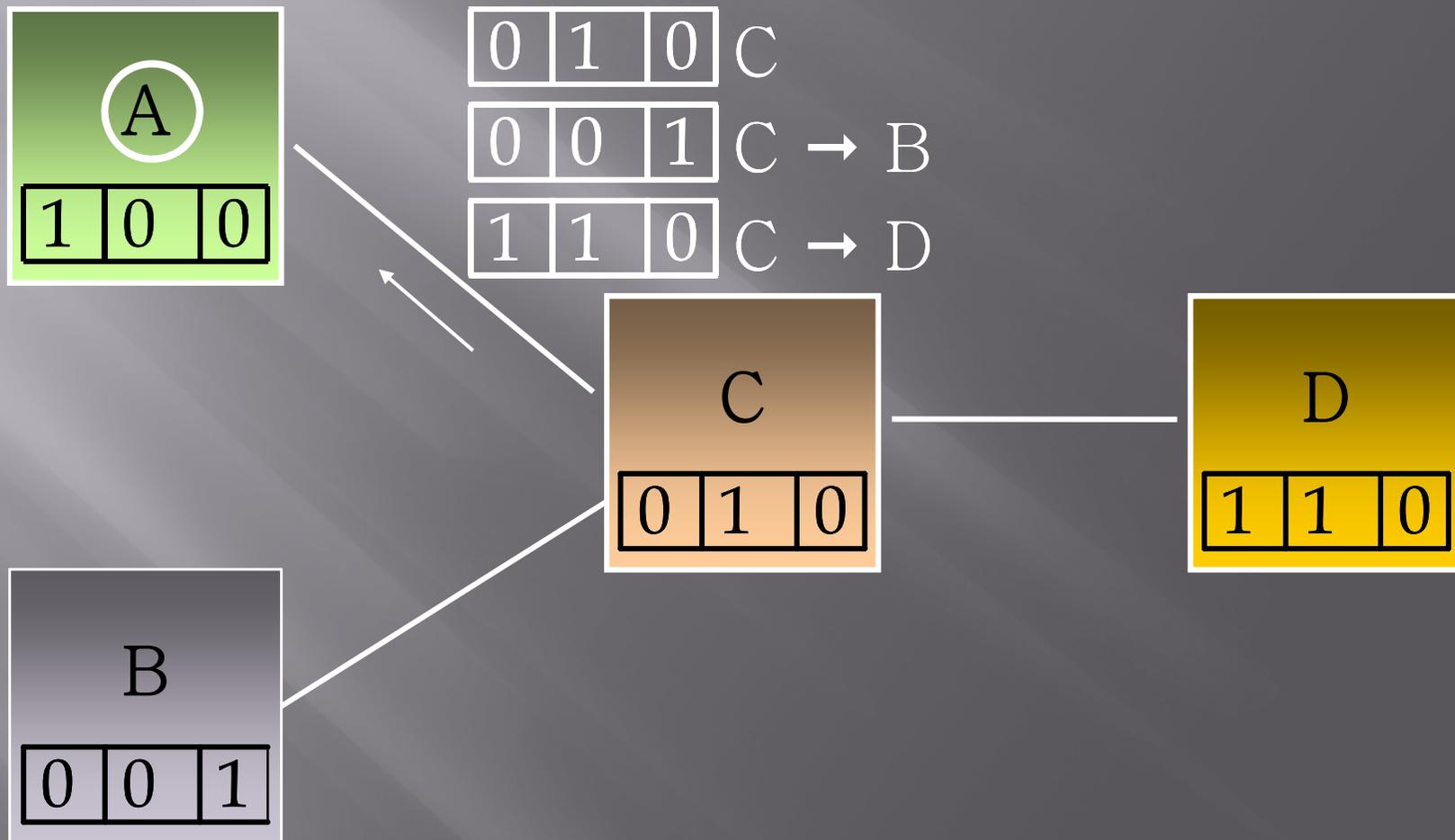
- ▣ $Nb(P)$ l'insieme dei nodi connessi direttamente con P nella overlay network
- ▣ $BitIdx_R(v_A)$ il bitmap index per il valore v_A di una risorsa R .

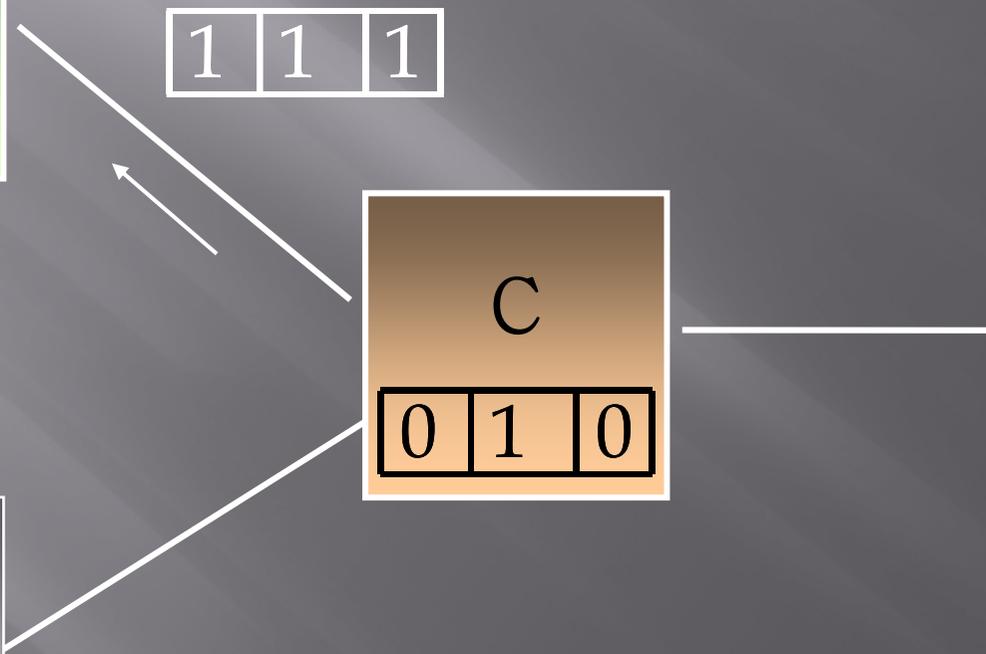
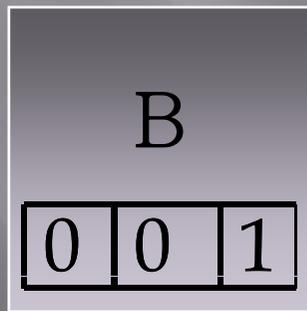
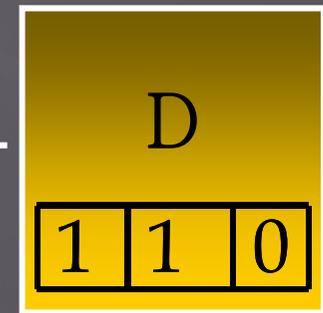
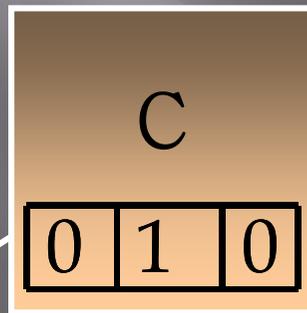
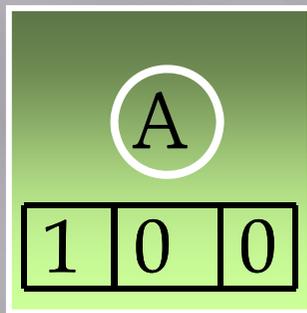
Routing Indices

- ▣ Per ogni $P' \in Nb(P)$, sia $T_{P'}$ il sottoalbero radicato in P'
- ▣ Sia $R(T_{P'})$ l'insieme delle risorse contenute in $T_{P'}$
- ▣ Per ogni attributo A che caratterizza $R(T_{P'})$, P associa al link $P \rightarrow P'$ il seguente indice

$$\text{LinkBitIdx}(P \rightarrow P', A) = \bigvee_{R(T_{P'})} \text{BitIdx}_R(v_A)$$

Routing Indices





Query Processing

- ▣ Quando una peer P riceve una range query multi-attributo Q, provvede a mapparla nello stesso spazio degli indici delle risorse
- ▣ Per ogni sottoquery di Q_A di Q su un attributo A, sia $[v_1, v_2]$ il range specificato da Q_A
- ▣ Si ottiene il seguente vettore

$$\text{QueryIdx}(Q_A) = (q_0, q_1, \dots, q_{k-1})$$

tale che

$$q_i \begin{cases} 1 & [a_i, a_{i+1}) \cap [v_1, v_2] \neq \emptyset \\ 0 & \text{altrimenti} \end{cases}$$

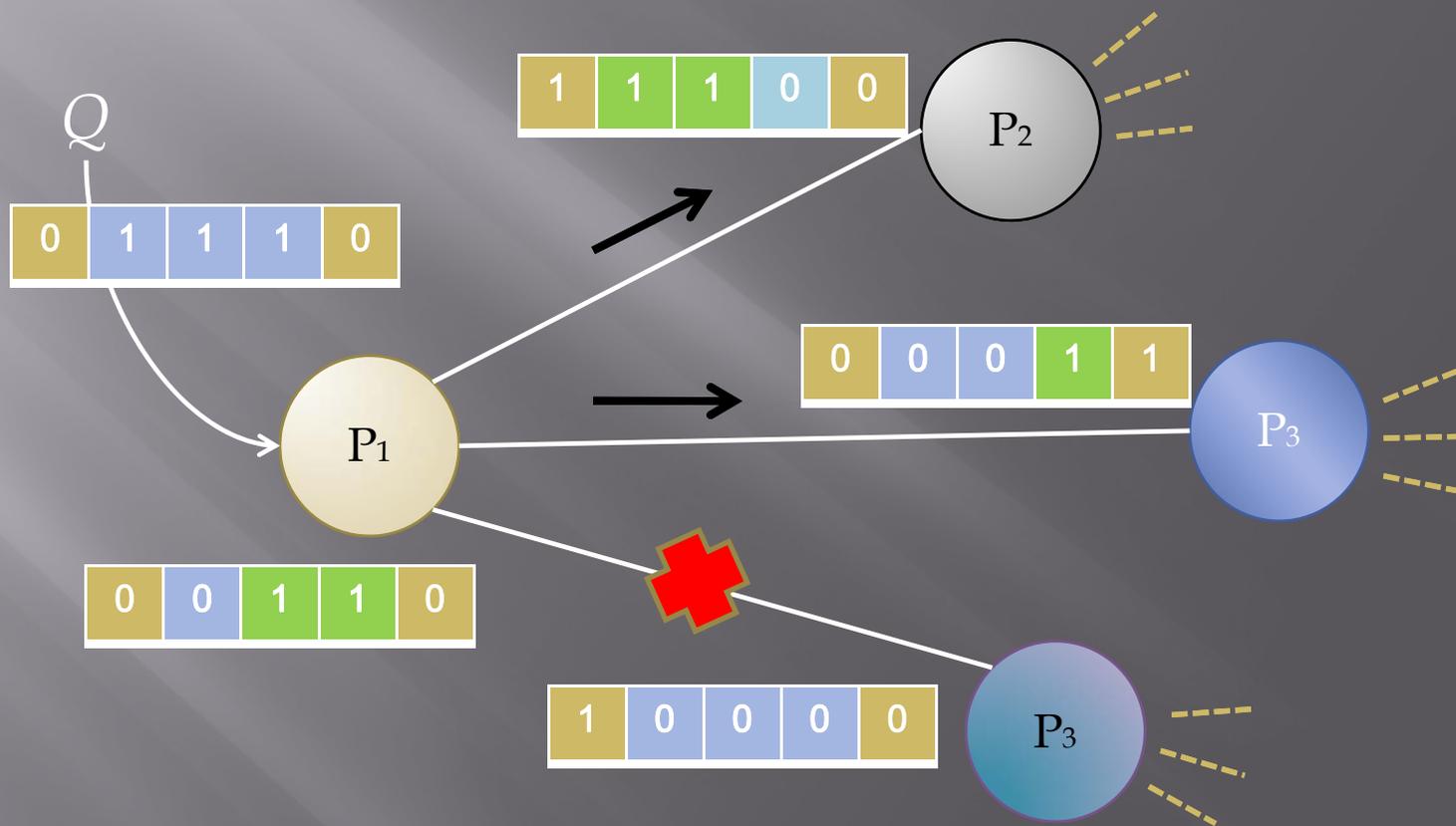
Query Processing

- ▣ Affinché vi sia un match potenziale con Q , occorre che vi sia un match potenziale per ogni Q_A inclusa in Q
- ▣ Dato un bitmap index $I_A = (b_0, b_1, \dots, b_{k-1})$ e l'indice di una sottoquery $Q_A = (q_0, q_1, \dots, q_{k-1})$, esiste un match potenziale se e soltanto se $\exists i$ tale che $b_i = q_i = 1$
- ▣ Dato un indice I e una query Q , I ha un match potenziale con Q se e soltanto se, per ogni A , I_A ha un match con Q_A

Query Processing

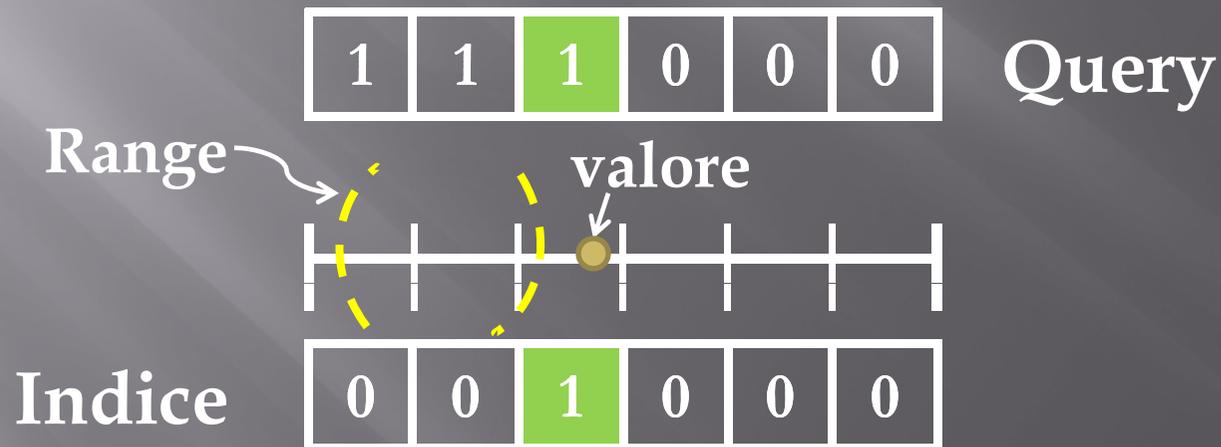
- ▣ Un peer P che riceve una query Q
 1. Verifica, usando i suoi indici locali, se ha dei match potenziali
 2. Effettua il forwarding di Q solo ai suoi vicini i cui RI hanno un match potenziale con Q

Query Processing



Query Processing

- ▣ La verifica dell'esistenza di match è approssimata!!
- ▣ Il meccanismo di indicizzazione discretizza lo spazio degli attributi, portando a possibili *falsi positivi*

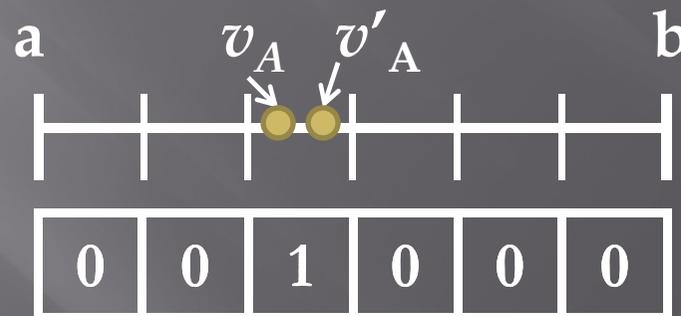


Propagazione degli update

- ▣ Alcune risorse possono essere caratterizzate da attributi in grado di cambiare valore nel tempo
- ▣ In caso di cambiamenti, potrebbe accadere che anche l'indice che descrive un peer P possa cambiare
- ▣ Questo evento può scatenare a cascata un cambiamento dei RI associati ai link che conducono a P

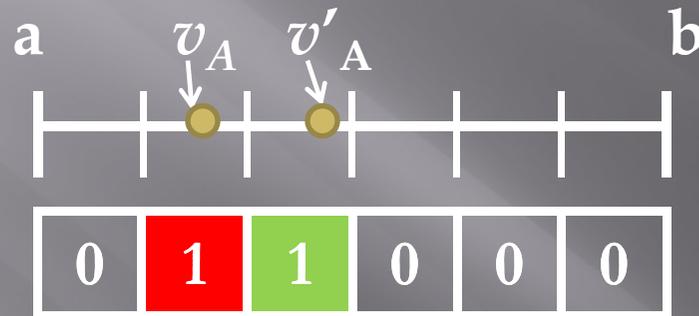
Propagazione degli update

- ▣ Il valore v_A di un attributo A una risorsa R cambia da v_A a v'_A
- ▣ Il proprietario della risorsa P calcola il suo nuovo indice locale per A
- ▣ **Caso 1:** il nuovo indice non differisce dal precedente



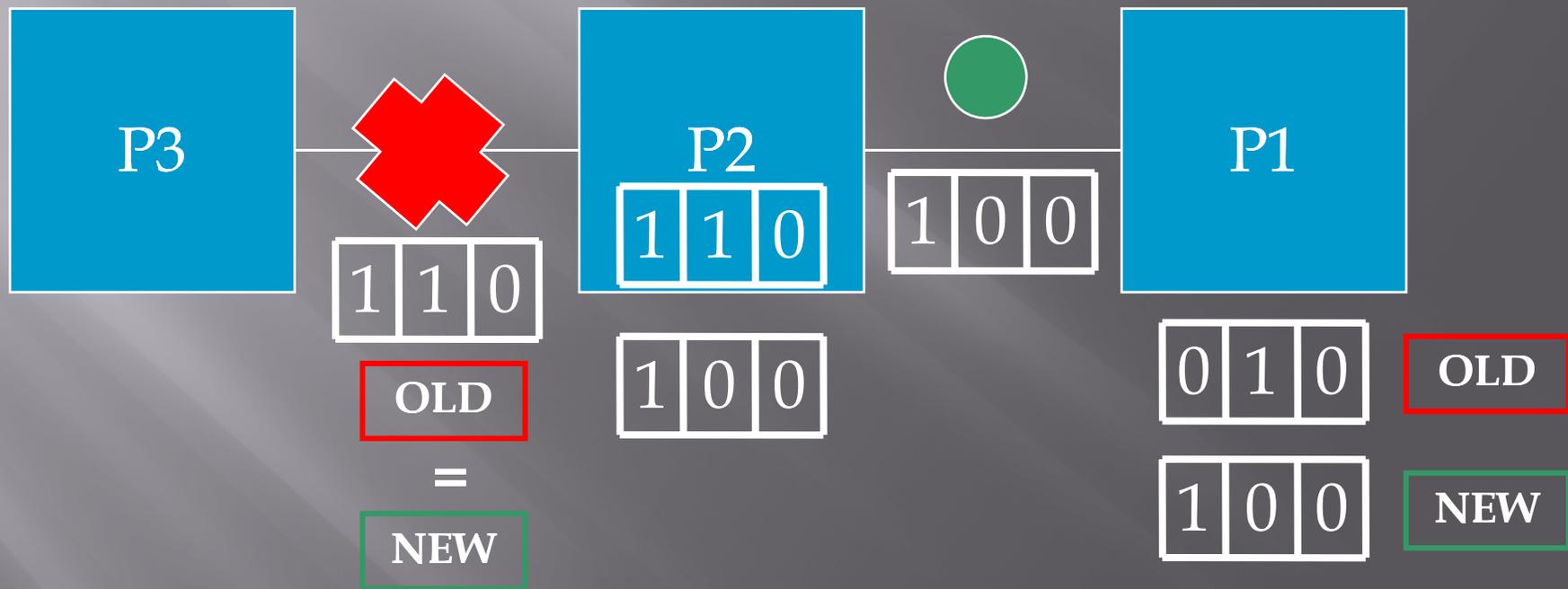
Propagazione degli update

- ▣ Caso 2: L'indice cambia



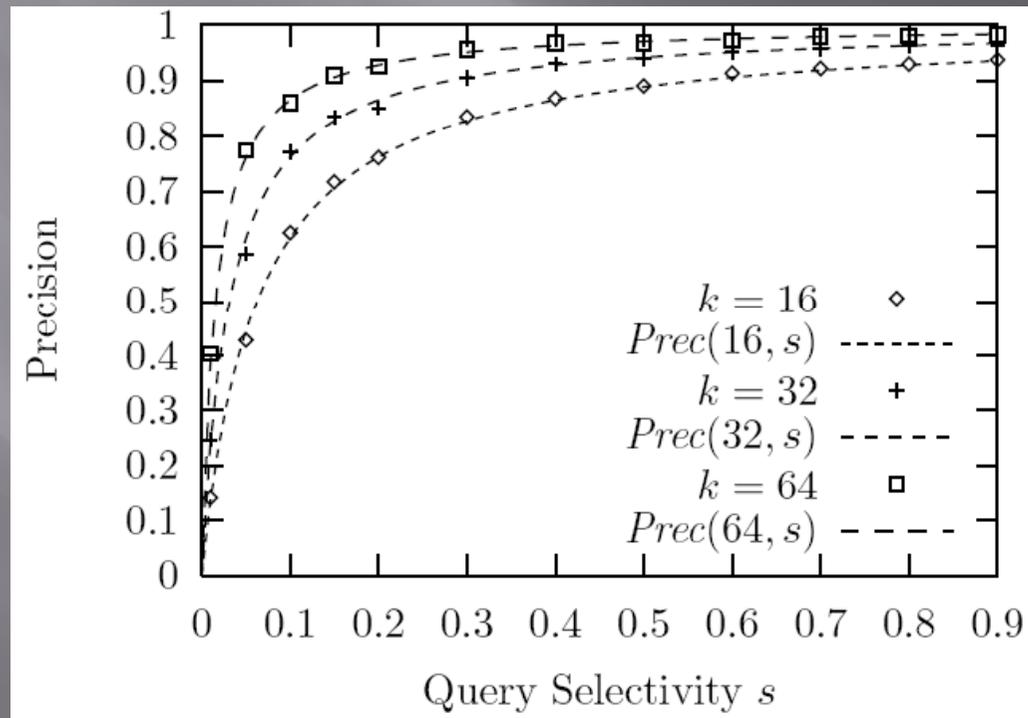
- ▣ P deve aggiornare i suoi indici e invia l'update ai suoi vicini
- ▣ Quando un nodo riceve un update, aggiorna il RI associato al link interessato ed eventualmente invia la comunicazione a sua volta

Propagazione degli update



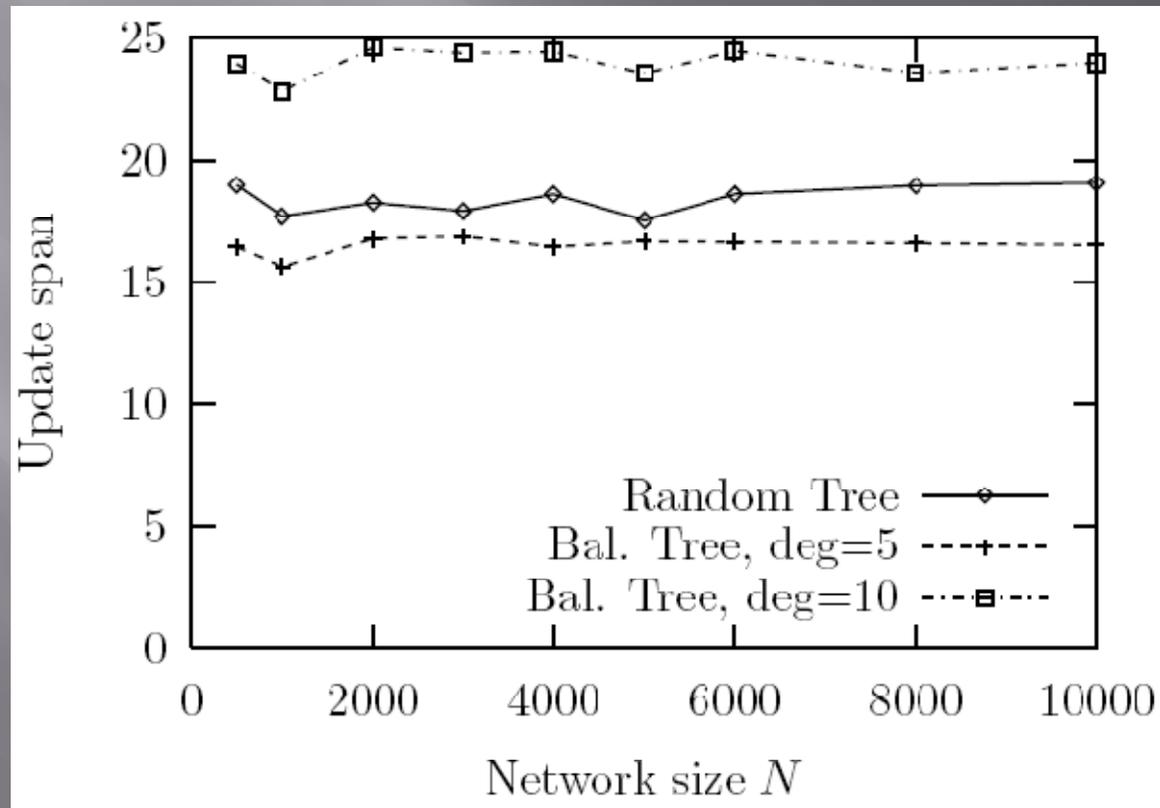
Prestazioni

- Effetto dei falsi positivi e dimensione dell'indice



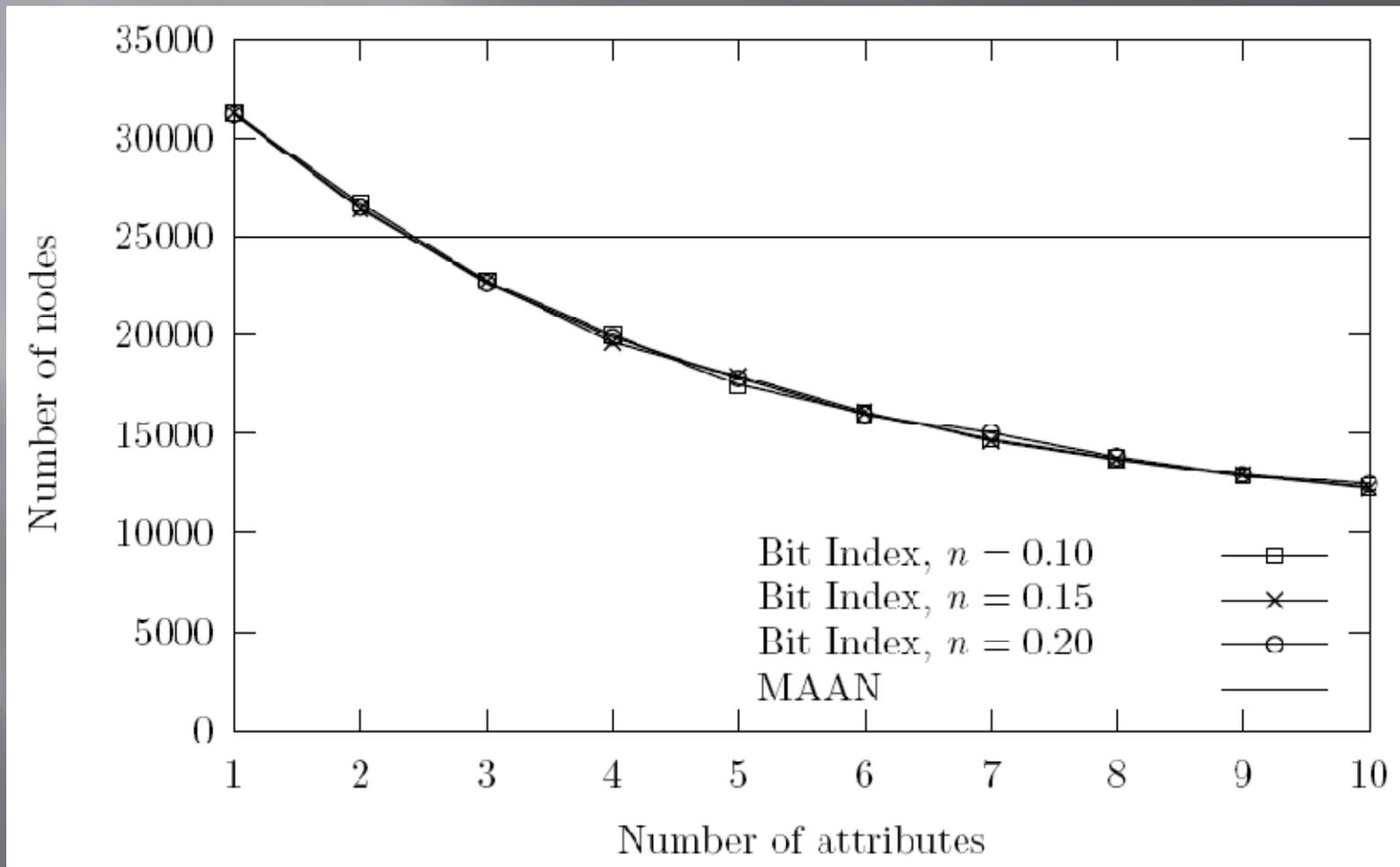
Prestazioni

- ▣ Andamento degli update



Prestazioni

▣ Confronto con MAAN



Tree Vector Indices

Riferimenti

- ▣ M. Marzolla, M. Mordacchini, S. Orlando
“Peer-to-peer systems for discovering
resources in a dynamic grid”, *Parallel
Computing* 33(4-5):339-358, 2007