

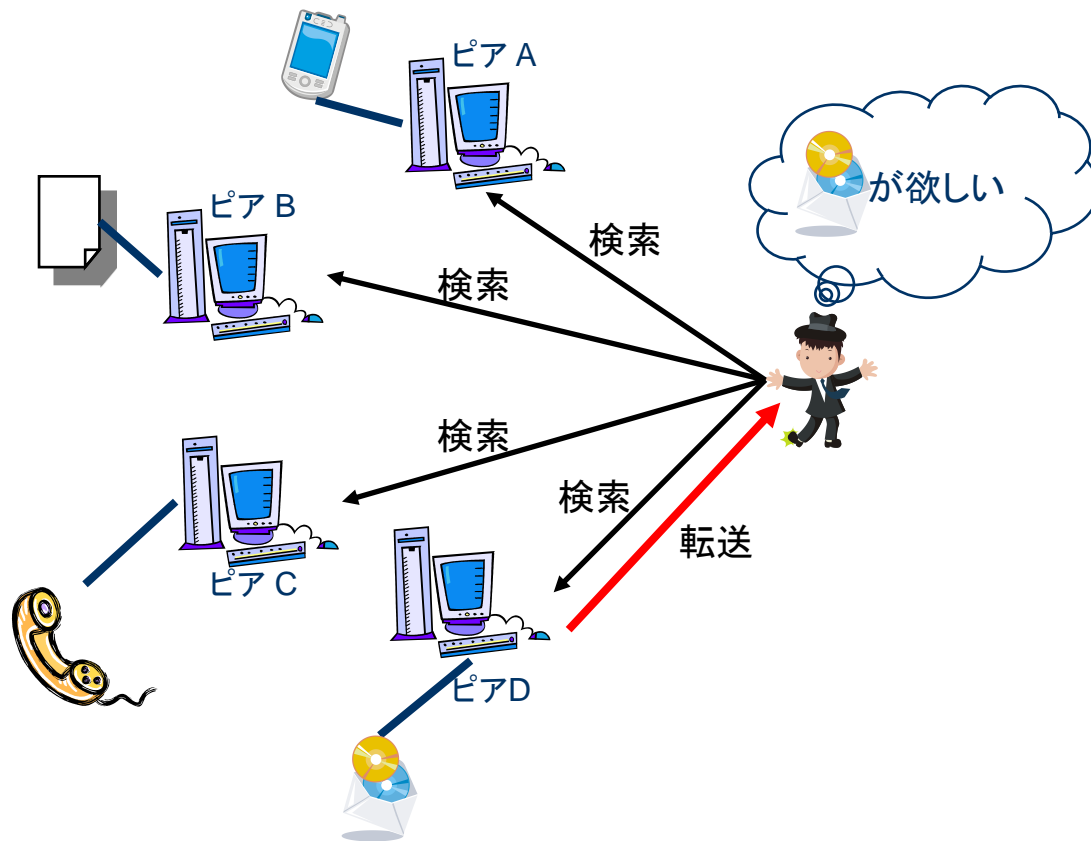
B木構造に基づくBloomフィルタの提案



目次

- 情報検索とは
- 従来研究
- 提案方式
- 比較評価
- まとめ

情報検索とは



情報検索には...

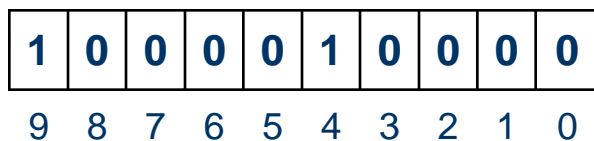
- ピュアP2P
- ハイブリッドP2P
- 分散ハッシュテーブル (DHT)
- Bloomフィルタ

Bloomフィルタについて(1)

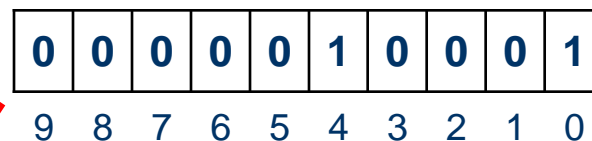
- Bloomフィルタとは...

キーワードに対して複数のハッシュ関数をかけて得られた値をビット列の位置とみなし、その位置のビットを1にすることで得られたビット列である。

キーワード1: "abc"
hash1(abc)=4
hash2(abc)=9

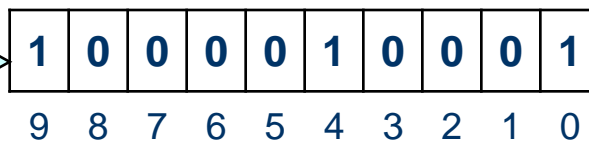


キーワード2: "xyz"
hash1(xyz)=0
hash2(xyz)=4



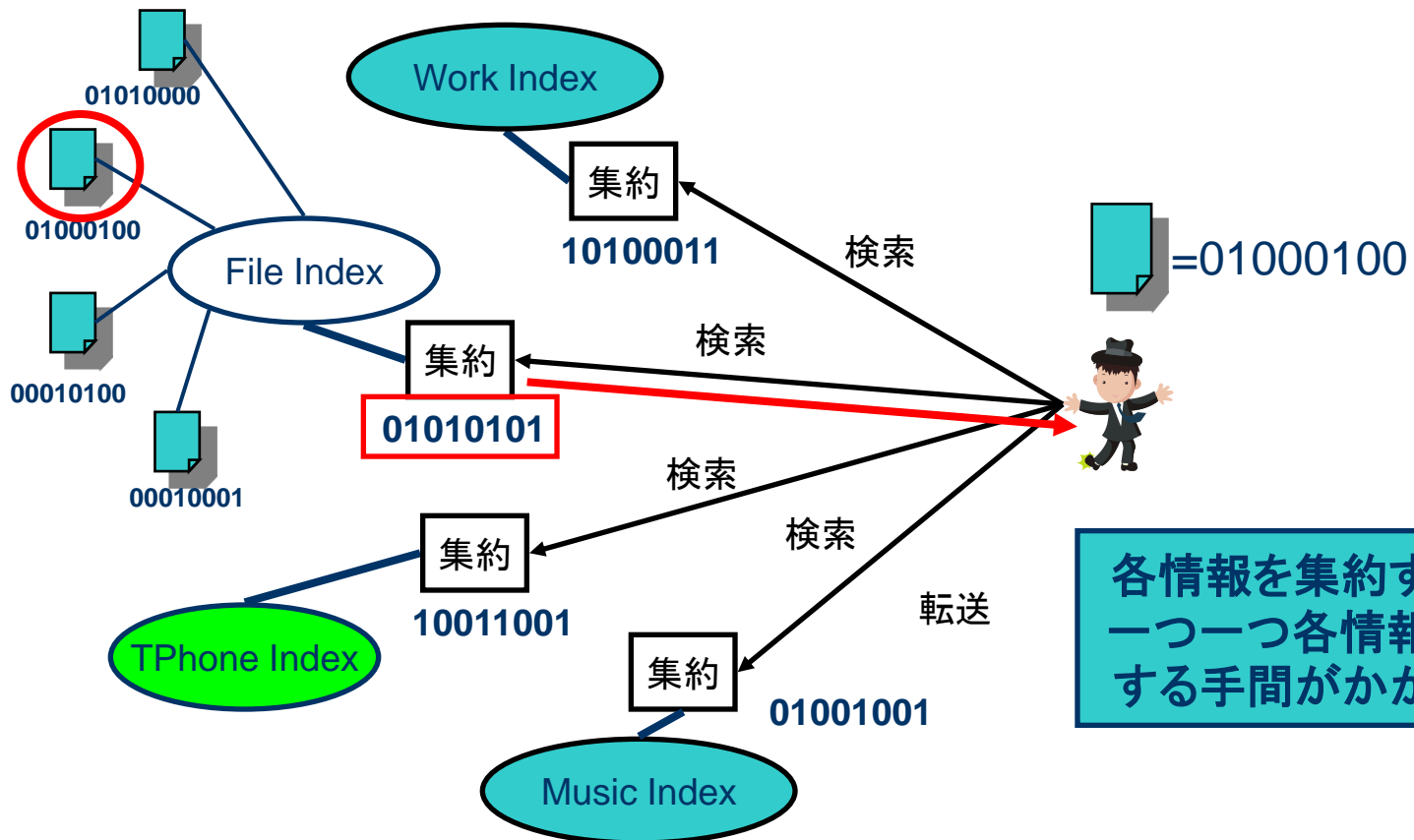
集約

複数のキーワード
をまとめて表現可
能となる



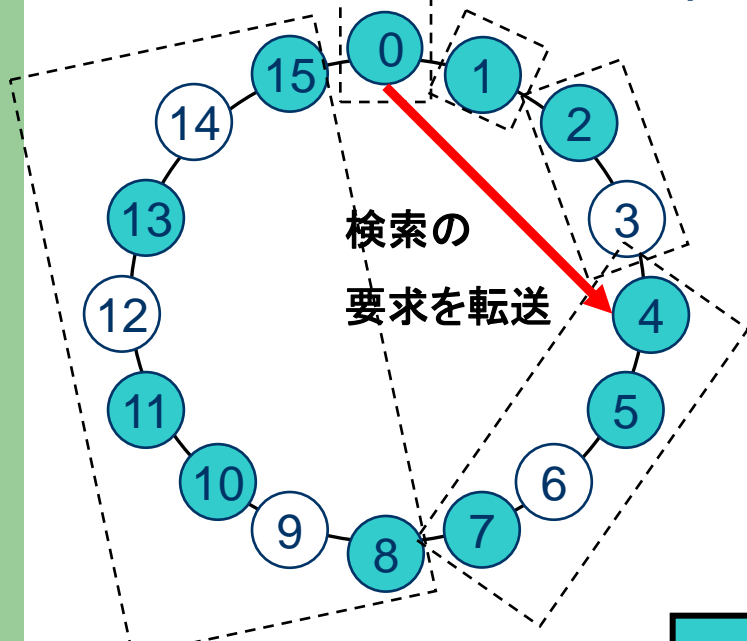
Bloom
フィルタ

Bloomフィルタについて(2)



リング構造でBloomフィルタを使った検索

- ノード0からBloomフィルタ”110000001”の検索方法



ノード0のBloomフィルタ

Finger	Finger bloom filter
0	000001000
1	000010000
2	010001110
4	110110101
8	101001011

ハッシュ空間サイズ $2^4 (=4\text{ビット})$

- 存在するノード
- 存在しないノード

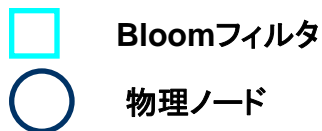
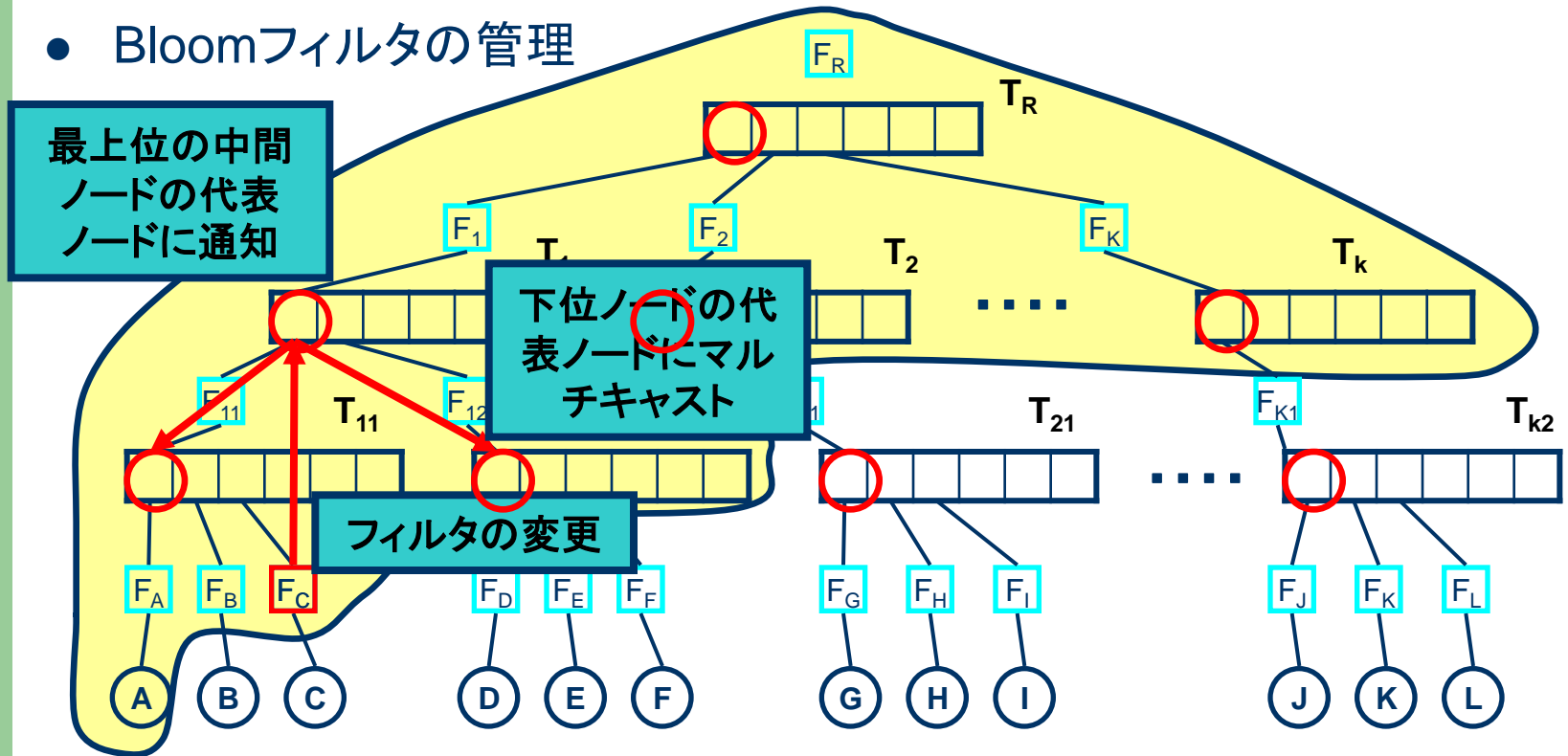
しかし、リングサイズに応じたフィルタ数が必要となり、フィルタに無駄ができてしまう。

提案方式のコンセプト

- 存在するノードのみを考えるのでフィルタ情報量の無駄を削減する。
- B木構造を自立的に各ノードで管理するため情報量が分散される。
- ノード参加・脱退の際に効率の良いフィルタ管理を行う。

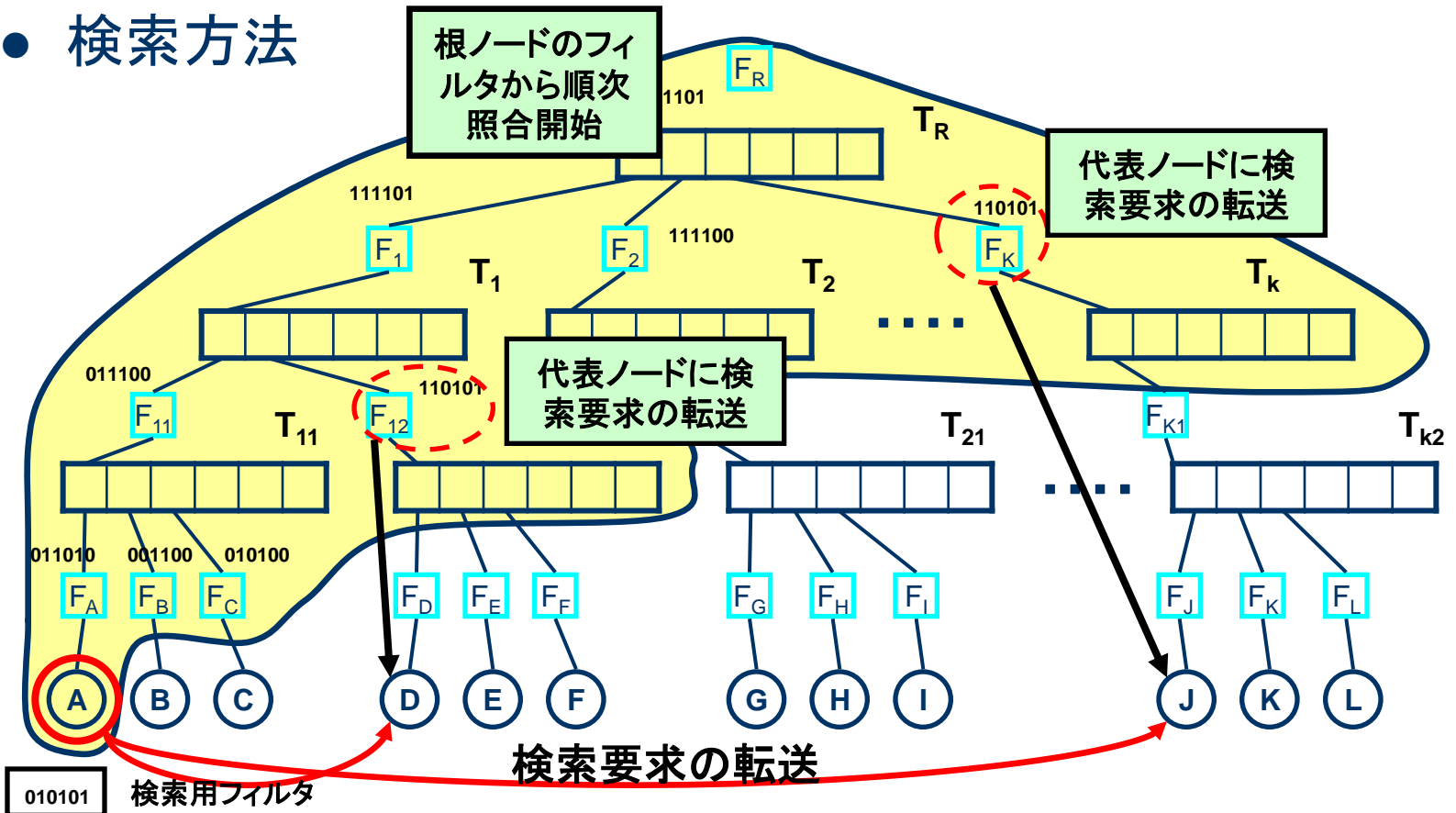
木構造に基づくBloomフィルタの提案(1)

- Bloomフィルタの管理



木構造に基づくBloomフィルタの提案(2)

- 検索方法



評価

	Chordベースの方式[8]	提案するB木に基づく方式
検索要求転送回数	$O(\log_2 N)$	$O(\log_m N)$
フィルタ変更の最大伝搬コスト	$O(N)$	$O(N)$
ノードが管理するフィルタ数	$O(M)$	$O(m \log_m N)$
ノードが管理する他ノードID	$O(M)$	$O(m \log_m N)$
ノードでの検索コスト	$O(M)$	$O(m \log_m N)$

フィルタ情報量が削減される

検索コストが減少

(Chord方式の例 SHA-1)

M=160

各ノードは160程度のフィルタを管理する必要がある。

(B木構造の例 N=10000, m=10)

$m \log_m N = 40$

各ノードは40程度のフィルタを保持してればよい

まとめ

- 本研究ではChordベースの提案方式の無駄な情報量を減らすB木構造の提案をした。
- 検索要求の転送回数の上限が確定するため、検索時間がむやみにかかることがない。
- 固定的なリング構造をもった従来の研究と比較して、本研究では木の高さが動的に変化するB木構造を持つため、各ノードが持つ必要があるフィルタ情報の量が削減できる。



検索処理の速度の向上

今後の課題

- シミュレータの作成
- 代表ノードの持つフィルタ情報の負荷の集中がどれだけ性能に影響があるか評価
- ノード参加脱退の際に効率のよいBloomフィルタの再構成
- 代表ノードを固定しない方式の提案