分散ハッシュの基本とその応用

2004年9月4日

西谷　智広(Tomo's Hotline)

自己紹介

当日発表。

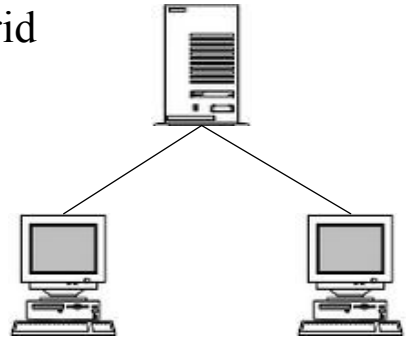
お楽しみに！！

目次

第１章：分散ハッシュとは？

1-1 今までのP2Pの方式の欠点

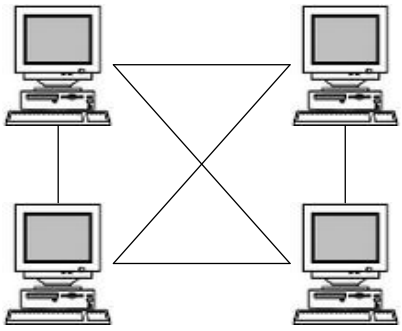
Hybrid P2P



・サーバダウンによる影響大

・管理者の初期・運用コスト大

Pure P2P



・同期を取るのが大変！

・一部の範囲のみ検索・通信可

分散ハッシュはこれらの諸症状に良く効きます！！

1-2 ハッシュってなんだろう？

P2P Today は横田さん。
P2P Todayは横田さん？

→ ハッシュ関数 →

1aksajkddjasdkga
0298saka;gkej1lgs

ハッシュ値

少しの文字の変化でハッシュ値は全然違う

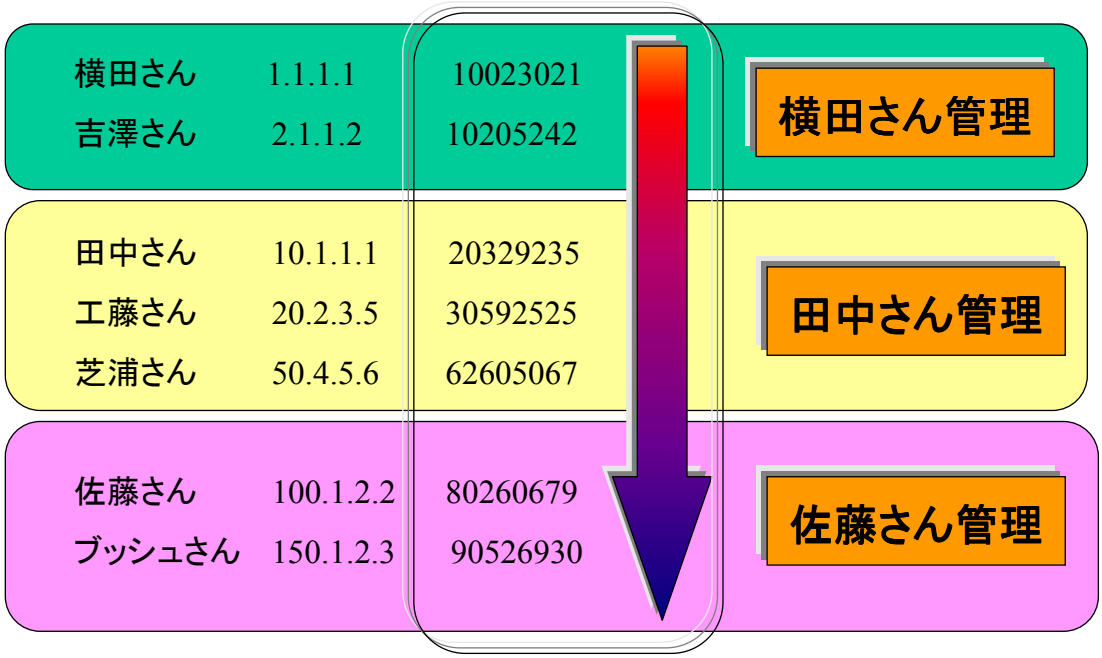
■同じ文字以外はハッシュ値は（基本的には）違う！

⇒ハッシュ値を全て把握できればデータは管理できそうだ♪

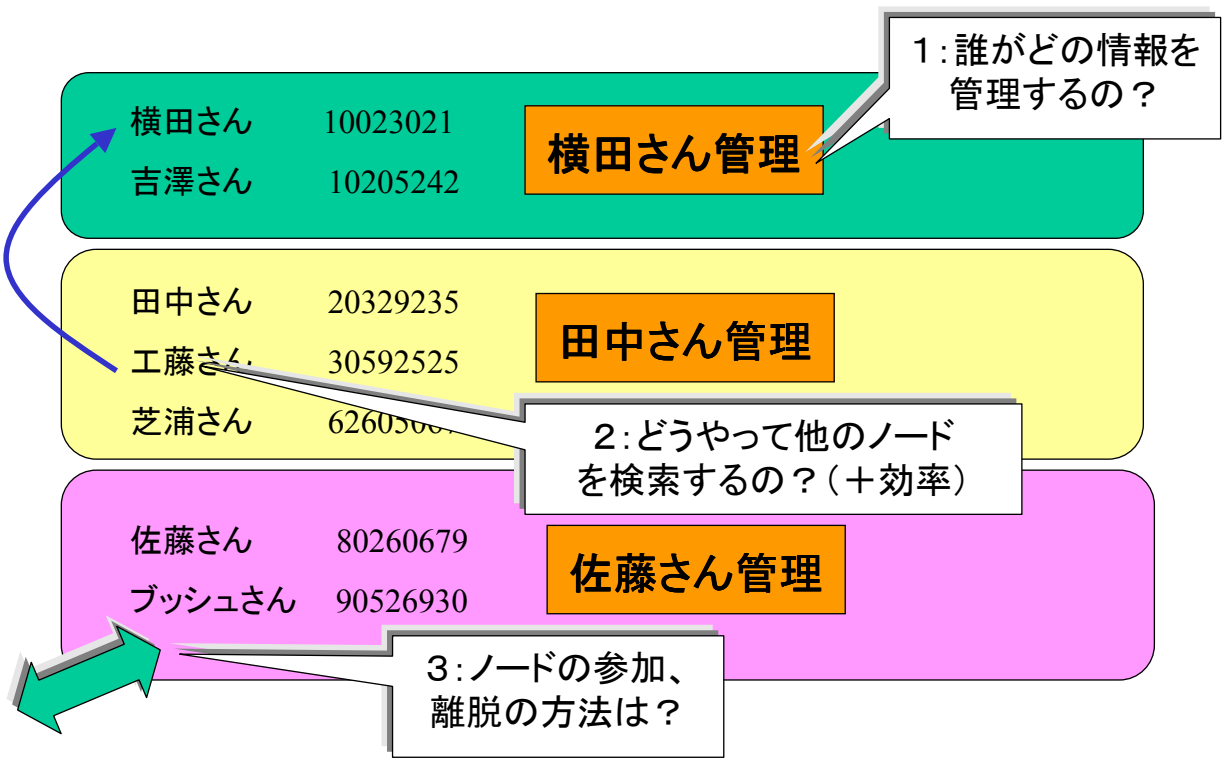
（データとハッシュ値が1対1に対応）

※充分にハッシュ空間が大きい事（+良いハッシュ関数）が必要

1-3 ハッシュ法と分散ハッシュの考え方



1-4 分散ハッシュでチェックすること



第2章：分散ハッシュの原理

2-1 分散ハッシュの種類

- Can　　Hyper-Cube
- Chord (Achord)　　skiplist
- Pastry,Tapestry　　Plaxton algorthm
- P-Grid,kademlia　　Tree

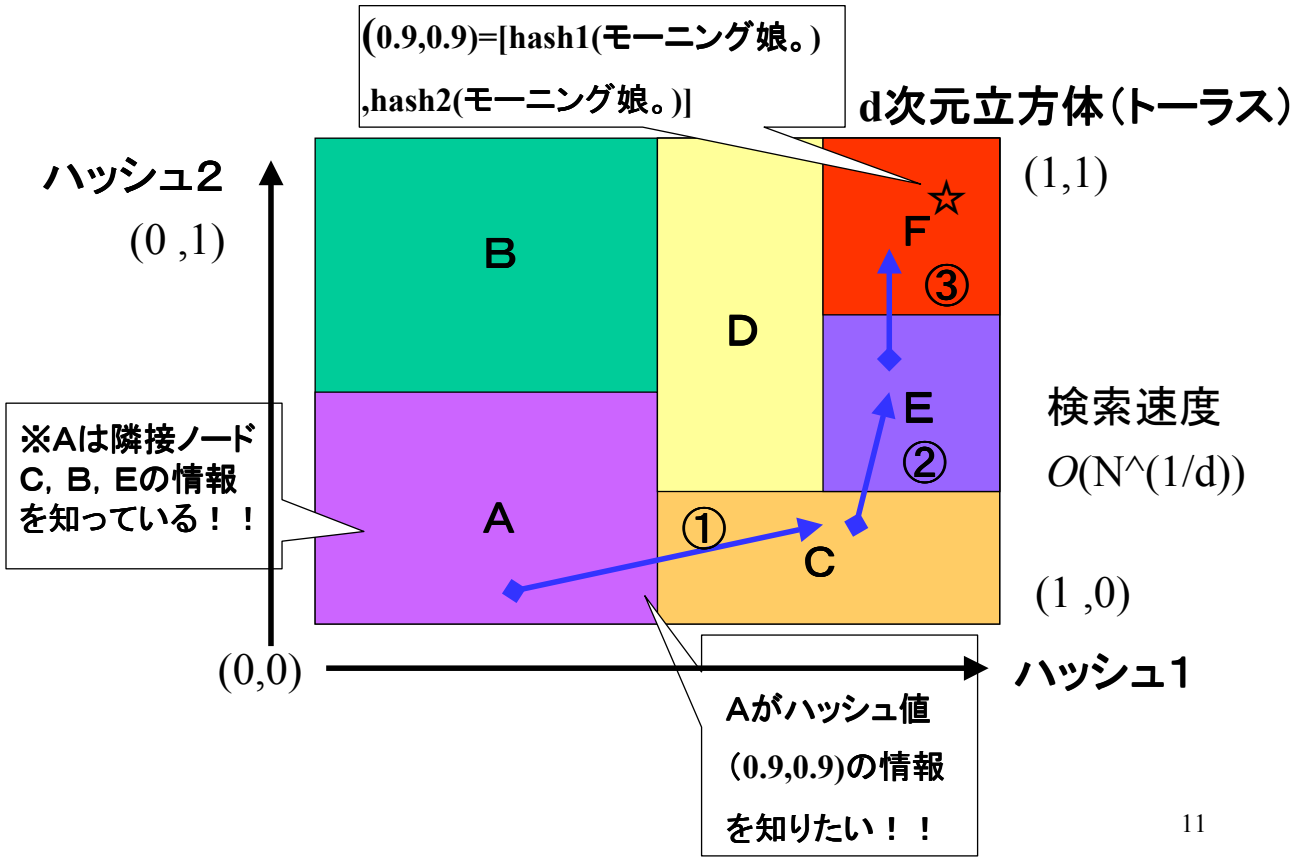
….

ハッシュ関数：SHA-1,MD5 ….

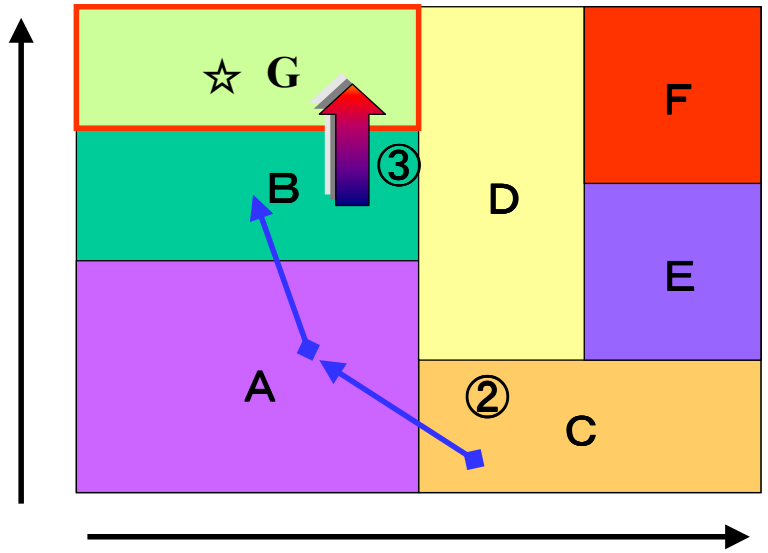
ハッシュ値：ファイル、ユーザのニックネーム

Node_ID:ノードのハッシュ値（IPアドレス、名前）

2-2-1 CAN その1：領域、検索

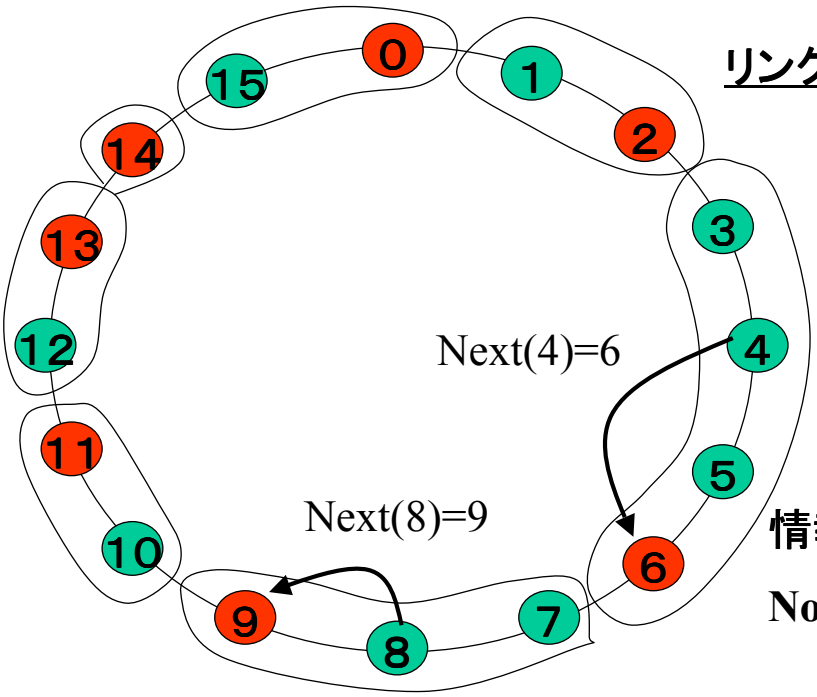


2-2-2 CAN その2：新規ノード追加



1:ノードGは乱数Pを発生
ノードGはノードCを知っている
とする。
2:Pの情報エリアを管轄する
ノードBと通信
3:Bの分散ハッシュの情報
エリアの一部を割譲してもらう

2-3-1 Chord その1:情報を管轄する領域



リング状

0～15までのハッシュ値を取るとする

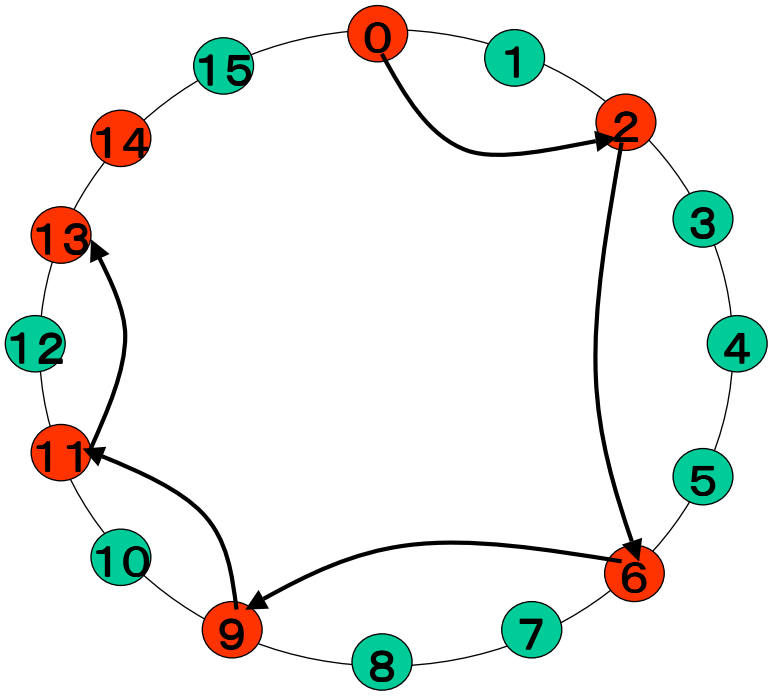
実際に存在するノード

そのハッシュ値にノードが存在しない

情報を管轄するノード:

Node_ID=Next(hash(name))

2-3-2 Chord その2:検索方法1



Node_ID=0から

Node_ID=13への

検索:

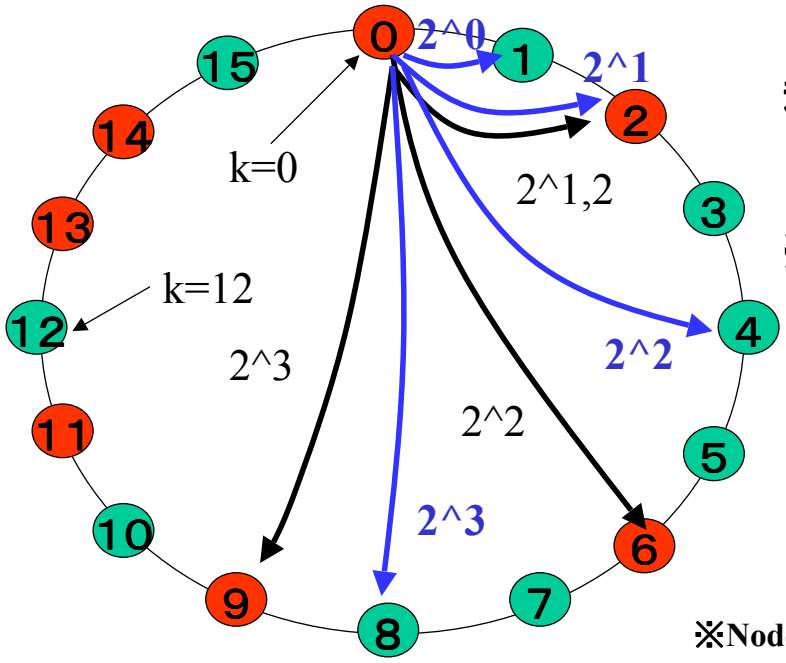
単純な方法:

検索スピード

遅い！！

O(N)

2-3-3 Chord その2:検索方法2



飛び飛びのリスト

(Skipリスト)を作って置く

※ショートカット！！

2^0 ⇒ Node_ID=2

2^1 ⇒ Node_ID=2

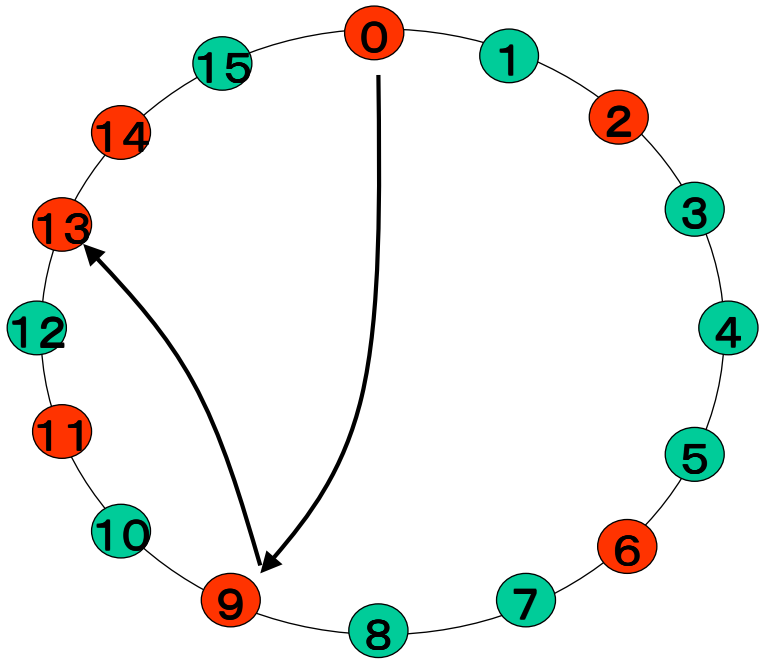
2^2 ⇒ Node_ID=6

2^3 ⇒ Node_ID=8

Node_ID(N)=Next(2^N)

※Node_ID(N)=Next(2^N+k)[一般化]

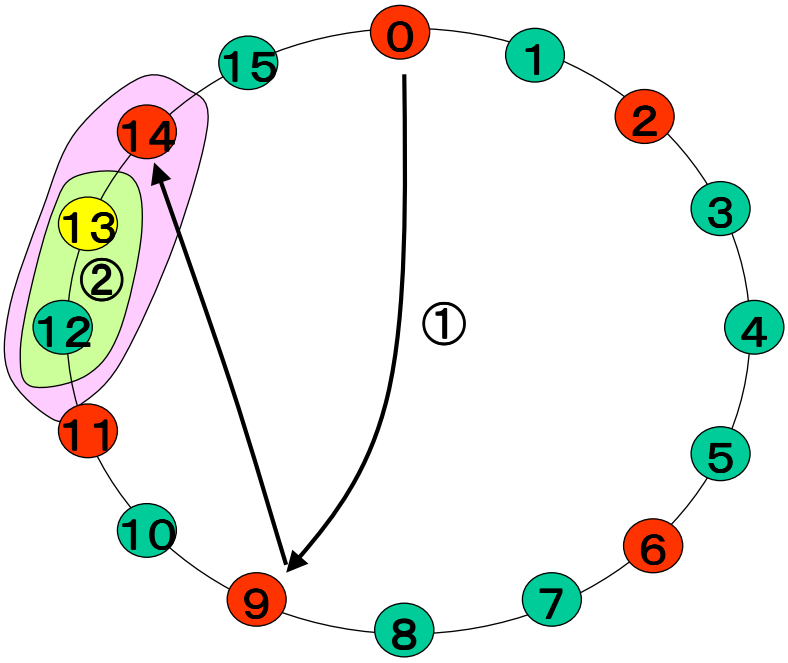
2-3-4 Chord その2:検索方法3



※単純方法では
5回の検索だった

一般に
$O(\log_2 N)$

2-3-4 Chord その3:ノードの追加



※Node_ID=13に追加した場合

ノード0を知っている場合:

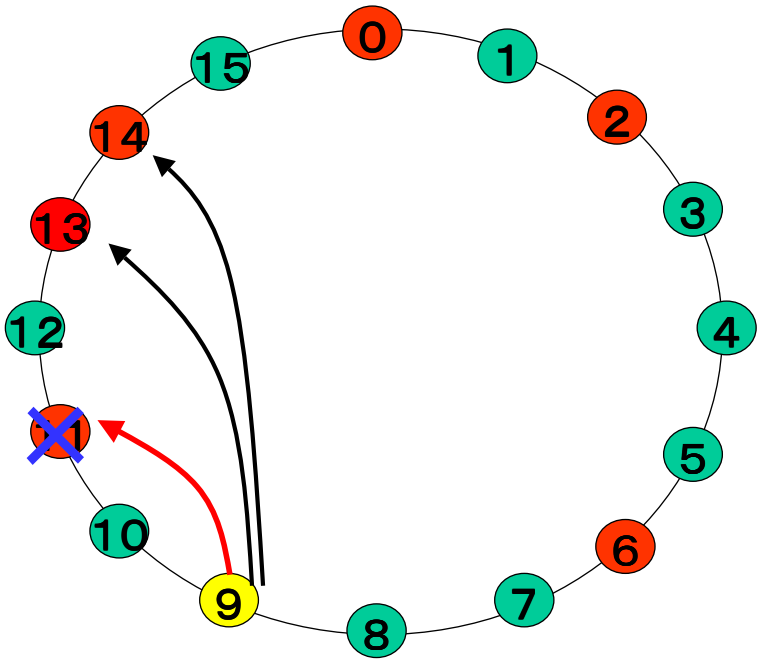
1:Node_ID=13を管轄しているノード(Node_ID=Next(13)=14へ通信。)

2:Node_ID=14からNode_ID=13へ情報の管理エリアの譲渡

3:skipリストの更新

(周囲のノードに対しても周知)

2-3-5 Chord その4:ノード離脱に対する対策



Next(自ノード)についての情報を複数確保して置く

※自分の情報をNext(自ノード)に複製しておく事も考えられる。

■skipリストの更新は絶えずチェックする

第3章：分散ハッシュの応用

3-1 分散ハッシュアプリの基本方針

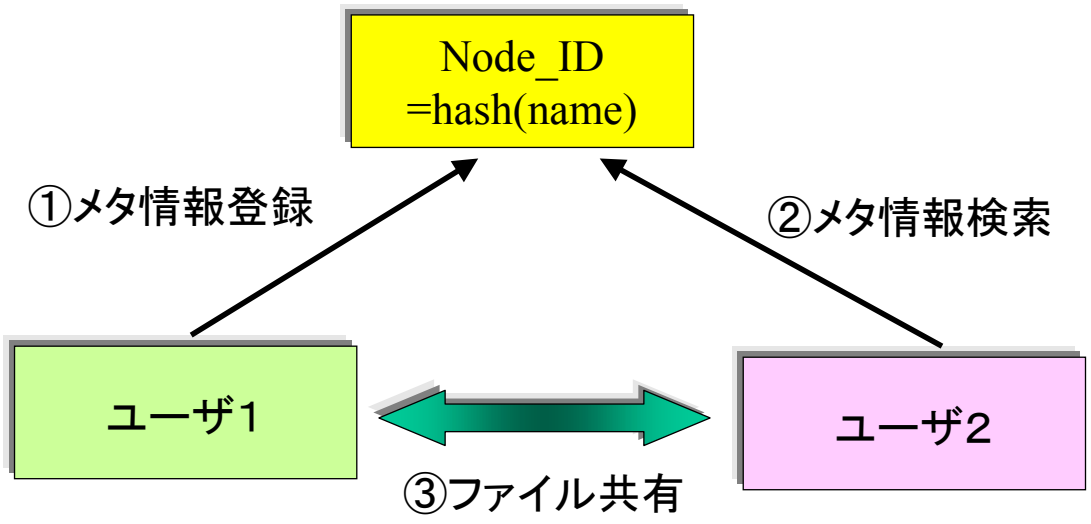
1．登録する情報キー（例：ファイル名）のhash値を作成
　例えば、hash(name)とする
2．Node_ID=hash(name)となるノードに情報を登録
3．Nameについての情報を知りたければ、
　Node_ID=hash(name)となるノードを検索
4．Node_ID=hash(name)となるノードにアクセスし、
　情報を得る

※手順はHybrid-P2Pとほぼ同じ！！

3-2 ファイル共有

1. ノード1がファイル名をnameとして、Node_ID=hash(name)となるノードにIPアドレス、ファイルのメタデータを登録
2. ファイル名nameを欲しいノード2はhash(name)を計算し、Node_ID=hash(name)となるノードからファイル名nameを有するノード1のIPアドレスを知る。
3. ノード2はノード1とP2Pでファイル共有を行う。

3-2 ファイル共有



3-3　P2Pメール、P2Pメッセンジャー

1.参加ノードには名前nameがあるとする。ノード1(name1)、
ノード2(name2)がメール（またはIM）をしようとする。

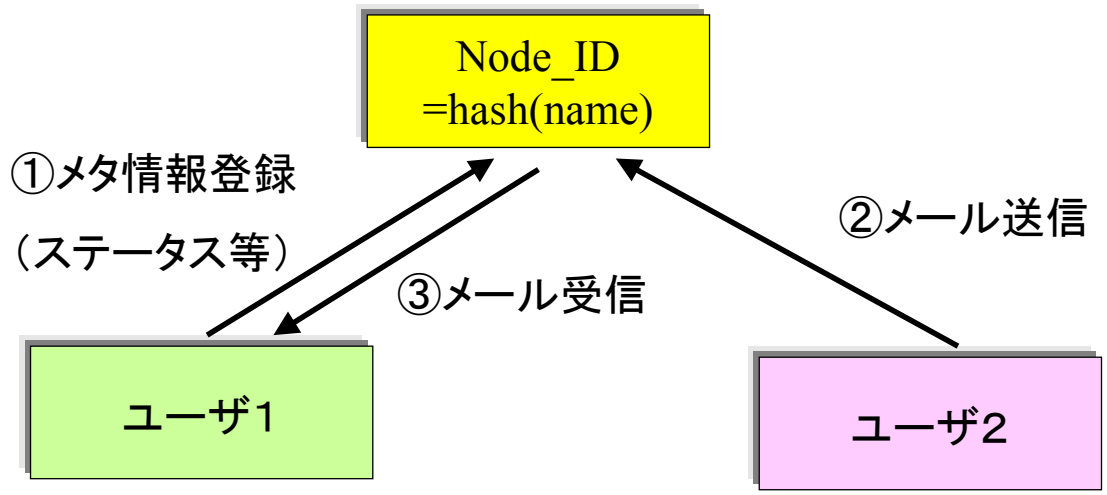
2.ノード1はNode_ID=hash(name1)に自分の属性（IPアドレス、現在のステータス）を登録。ノード1はNode_ID=hash(name1)にステータスを送るため定期的に通信。ノード2も同様。

3. ノード2はノード1にメール（またはIM）を送信するため、Node_ID=hash(name1)に送信。

4.Node_ID=hash(name1)はノード1が生きていれば、それを転送。

それ以外の場合、ノード1がNWに再接続するまで保存。

3-3　P2Pメール、P2Pメッセンジャー



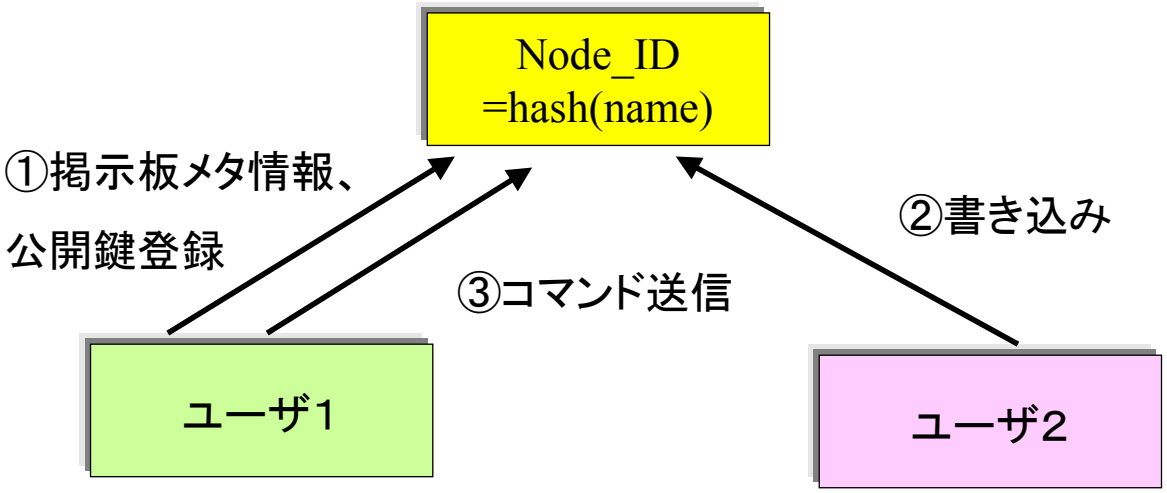
3-4 P2P掲示板、P2PBlog

1.掲示板名をnameとする。掲示板を立てる人は公開鍵、秘密鍵ペアを作成。

2.公開鍵、掲示板コンテンツはNode_ID=hash(name)に蓄積。掲示板コンテンツは書き込み者によって電子署名を行う。Node_ID=hash(name)は電子署名を見て、書き込み等が正当にされているか、チェック。

3. 掲示板を立てた人のコマンドによって、Node_ID=hash(name)は掲示板等の削除が可能。そのとき、コマンドは電子署名が　されていることが必要。

3-4 P2P掲示板、P2PBlog



3-5 P2P匿名プロキシ

1.ユーザが閲覧したいURLをurlとする。

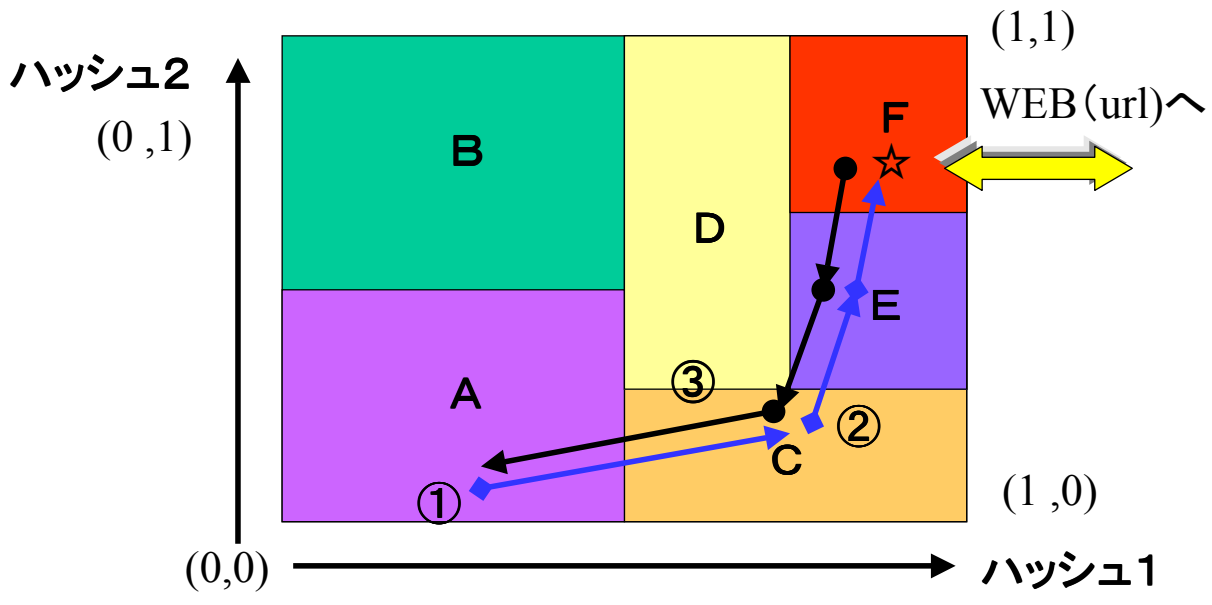
Node_ID=hash(url)にURLを送信。

2.ユーザ⇒Node_ID=hash(url)までに何個ものノードを介する事に注意。適切な処置をすれば、Node_ID=hash(url)はどのユーザからWEB閲覧を要求しているかはわからない。

3.Node_ID=hash(url)はurlのWEB情報をユーザに返す。

このときに、ユーザがNode_ID=hash(url)を検索したノードを介して情報を返すこと。

3-5 P2P匿名プロキシ



第3章：分散ハッシュの総まとめ
（skypeを目指して）

4-1 VoIPを行うための準備

■全てのユーザは分散ハッシュに参加する

-ユーザのニックネームをnameとすると、そのユーザの

メタ情報はNode_ID=hash(name)となるノードに格納

■公衆電話網への接続は考慮しない

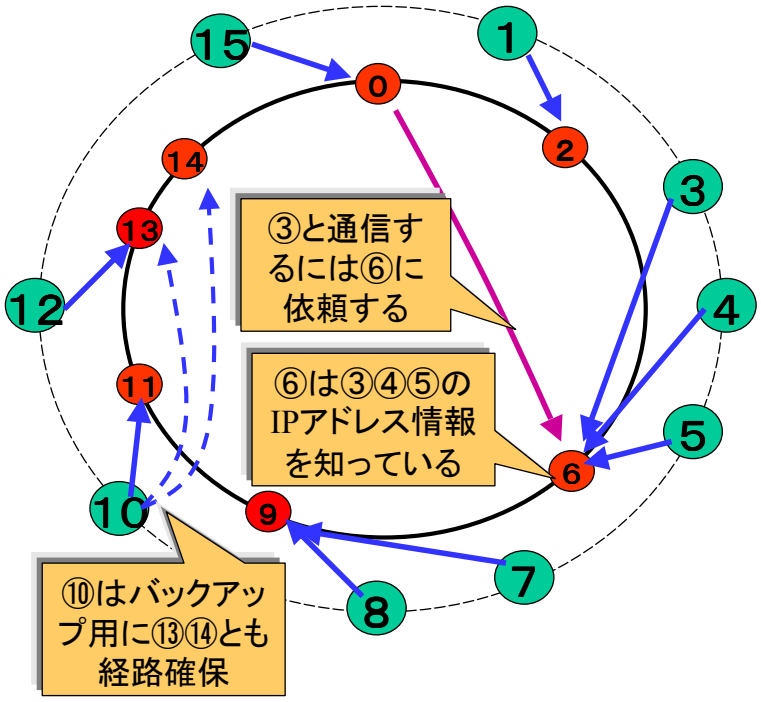
■NAT越えをする必要がある

案1：グローバルIPを持つノードを介する　⇒システムがシンプル

案2：UDP Hole punching ⇒中間ノードの負担が少ない

※今回は案1で説明する

4-2-1 グローバルIPを介したNAT越え（提案）：その1



グローバルIPのノード

プライベートIPのノード

2重DHTモデル

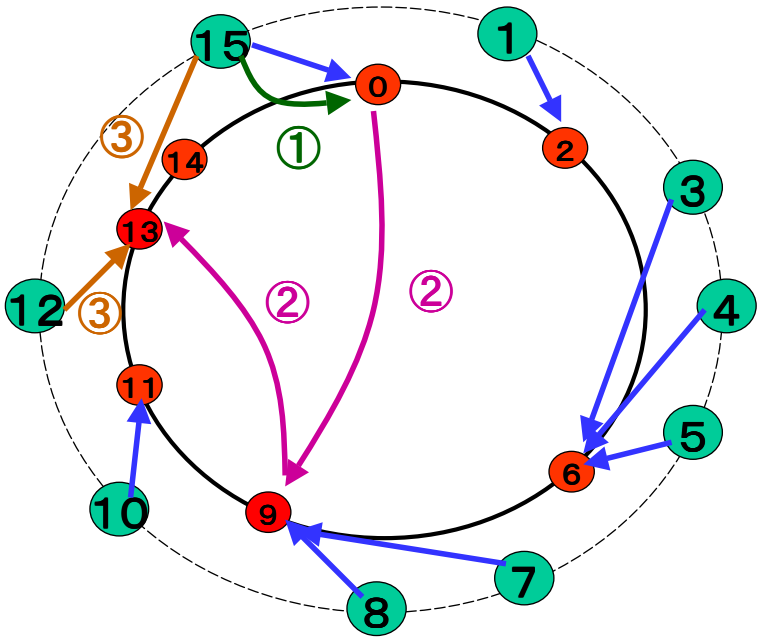
経路情報保持をDHTに応用

1:G-IPノード間のみで経路情報（ノードのIPアドレス）のDHTを形成。

2:P-IPノードは右回りで1番近いG-IPのノードと経路情報交換。

たたし、P-IPノードはバックアップ用に複数のG-IPノードを知っておく

4-2-2 グローバルIPを介したNAT越え（提案）：その2



グローバルIPのノード

プライベートIPのノード

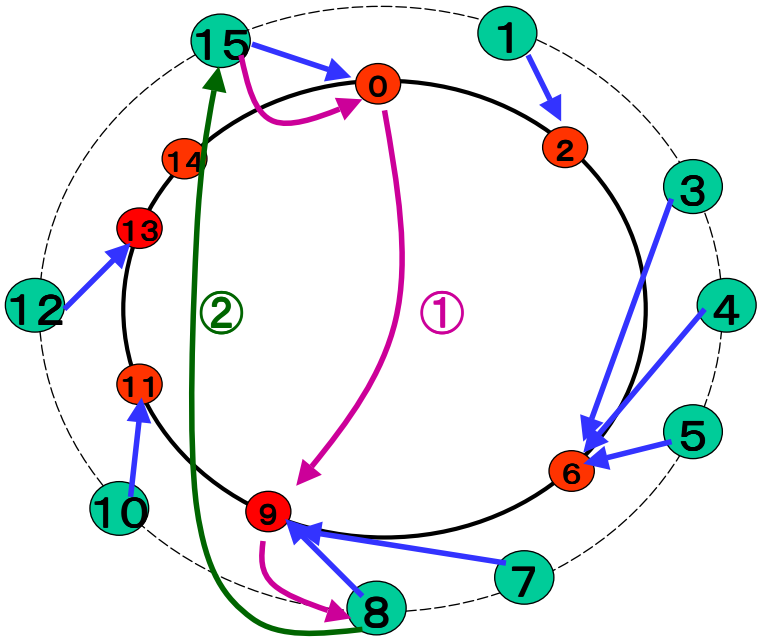
Node_ID=15がNode_ID=12と通信する場合

①G-IP　Node_ID=0にNode_ID=12の検索依頼

②Node_ID=13はNode_ID=12の情報をNode_ID=0を介してNode_ID=15に返す

③Node_ID=15はNode_ID=13を介してNode_ID=12と通信

4-3-1　VoIPの手順：その１



Node_ID=15が

Nameの人と話したい！

①Nodeのサーチ

Node_ID=hash(name)=8

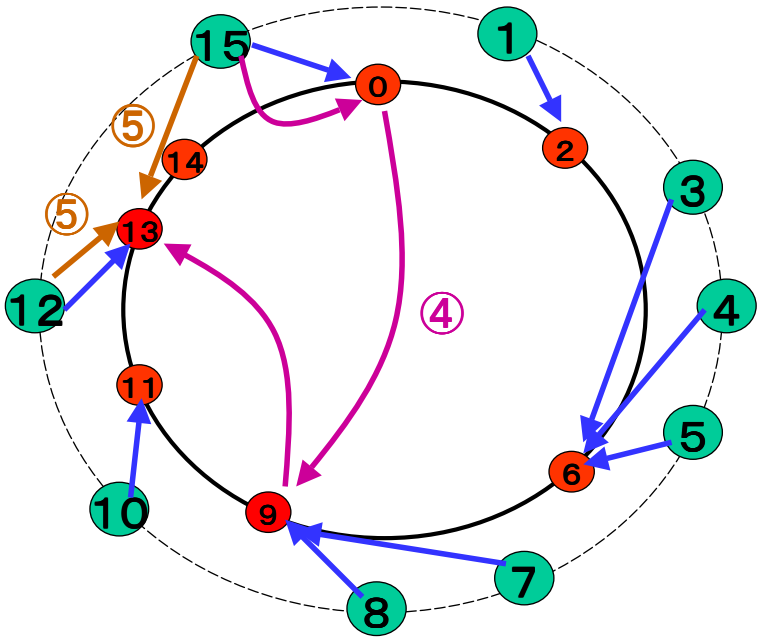
②Node_ID=8からNode=name

のメタ情報（Node_ID）を入手

③Node=nameのNode_IDは

12であることが判明

4-3-2　VoIPの手順：その２



Node_ID=15が

Nameの人と話したい！

④Node_ID=12との通信要求

⑤Node_ID=13を介して

Node_12と通信。

SIPについてもキャッチアップすべき事項

4-4 まとめ

■分散ハッシュは

-スケーラビリティに優れている

-高速な検索が可能

-様々なアプリへの応用の可能性を秘めている

-実装は研究レベルを脱してない

日本発の分散ハッシュのアプリを作っていこう！

ご清聴ありがとうございました！！

HP： http://homepage3.nifty.com/toremoro/index.html

Blog： http://toremoro.tea-nifty.com/

※質問は tnishita@yahoo.co.jpまでお願い致します。

Mixi,Greeに参加しています（Tomo)