key-value store 勉強会
2009年 2月 20日(金)

peer-to-peer の方から来ました

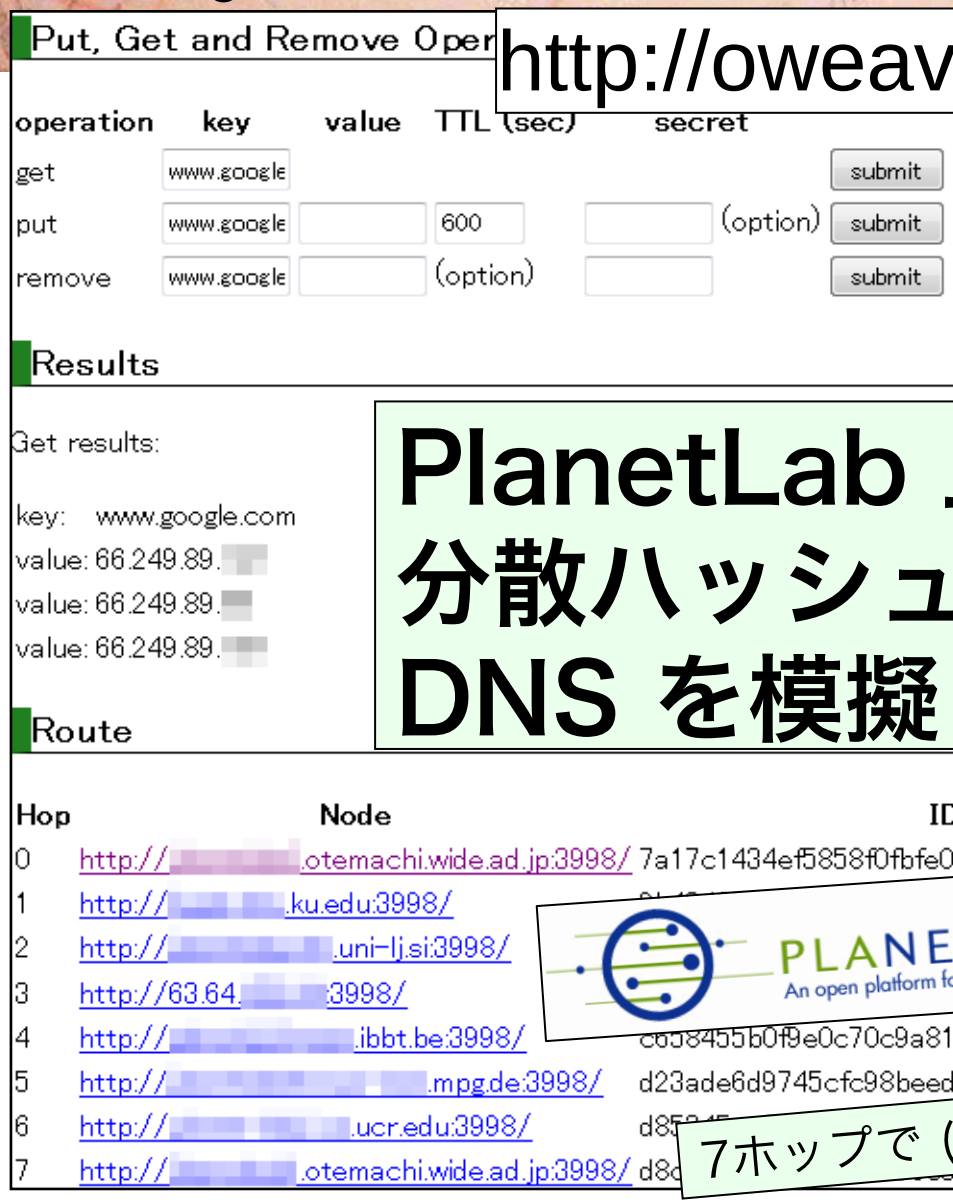
首藤 一幸

Overlay Weaver

オーバレイネットワーク構築ツールキット

Overlay Weaver – デモ

http://oweaver.nyuld.net:3998/



ウェブインタフェース

XML-RPC でもアクセス可

PlanetLab 上で
分散ハッシュ表として運用し
DNS を模擬

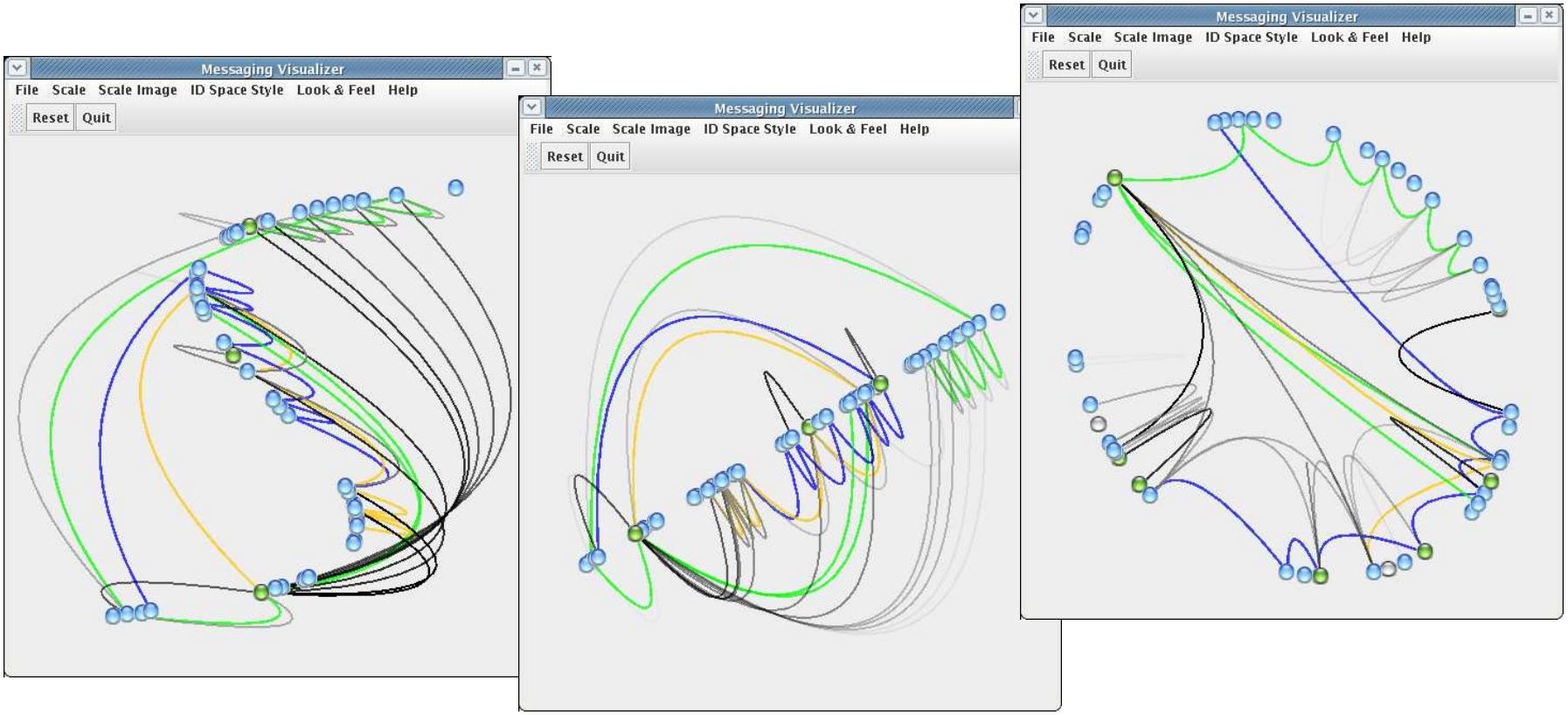
2/19 現在
36ヶ国 526台
最大
35ヶ国 572台

PLANETLAB
An open platform for developing, deployi

7ホップで (データ保持) 担当ノードに到達

Overlay Weaver – デモ

- ノード, メッセージ, マルチキャストの配送木を可視化。
- 実ネットワーク / エミュレータどちらでも動作。
  - PC 1台、メモリ 2 GB で 15万ノードが動作。



consistent hashing な key-value store

- **サーバ側**
  オンメモリ key-value store (の一部)
  - memcached
    - クライアントライブラリ次第
  - Dynamo (Amazon)
  - ROMA (楽天技術研究所)

  - 使い方
    - RDB から得たデータのキャッシュ。
    - 有効期間が短い (volatile) データの保持。
      - e.g. HTTPやSIPセッション情報, 座席予約, 株式・債権取引, 最終ログイン時刻, …

性能重視

no-hop が多い
～ 1,000台？

経路表の大きさ・保守コストと
経路長の
トレードオフ

～ 数百万台？

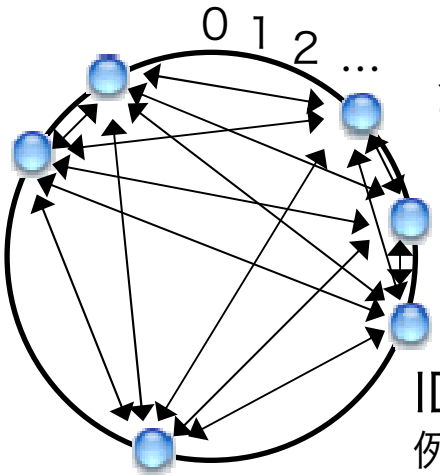
- **peer-to-peer 由来**
  構造化オーバレイ / 分散ハッシュ表
  - Chord, Pastry, Kademlia, …
    - Azure の基盤に両方向版 Chord が使われる?
  - 実装: **Overlay Weaver (首藤)**

スケーラビリティ重視

multi-hop の
routing / forwardingを
いとわない

consistent hashing な key-value store

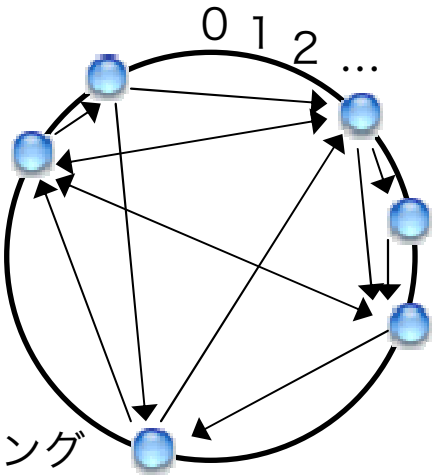
no-hop	multi-hop
•担当ノードに直接到達	•担当ノードまで数ホップ
•サーバ側 •低遅延 •全ノードが全ノードを知る •～ 1,000ノード？	•peer-to-peer 由来 •小さな経路表 O(log n) 以下 •ノードの頻繁な出入りに耐える (べき)。 •～ 数百万ノード？



全対全

ID 空間
例えば 0 ～ 2^160 のリング

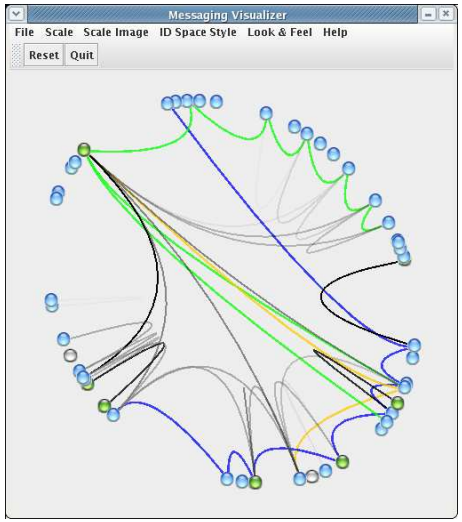
経路表



Overlay Weaver

- 分散ハッシュ表 (DHT) / 構造化オーバレイ実装
  - 記述言語は Java。
  - ３万ステップ。４万８千行。
  - Apache License 2.0。
    - いろいろな目的に使いやすい。

可視化ツール



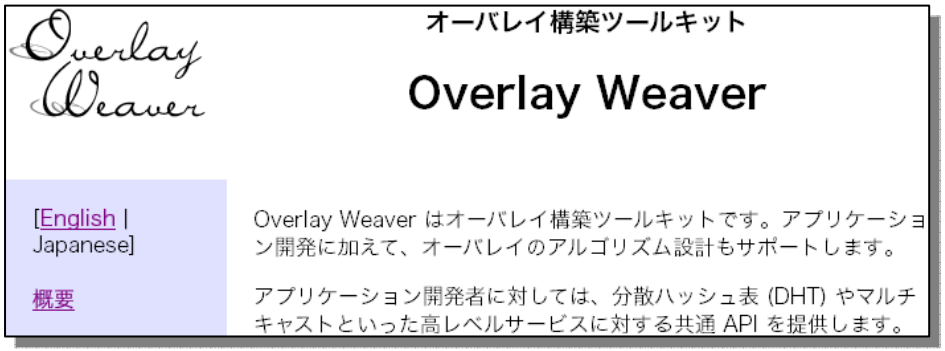
- 特徴
  - 分散ハッシュ表 (DHT) だけじゃない。アプリケーション層マルチキャストも。
  - ルーティングアルゴリズムの差し替え
    Chord, Kademlia, Koorde, Pastry, Tapestry
  - コーディングなしでの運用や実験
    - サンプルツール (DHTシェル等) を使った運用。
    - エミュレータを使った実験: メッセージ数やホップ数の計測。

    PC 1台で 15万ノード動作
  - XML-RPC, memcached プロトコル で DHT を使える。
    - XML-RPC は Bamboo, OpenDHT と同じプロトコル → 同じクライアントを使える。

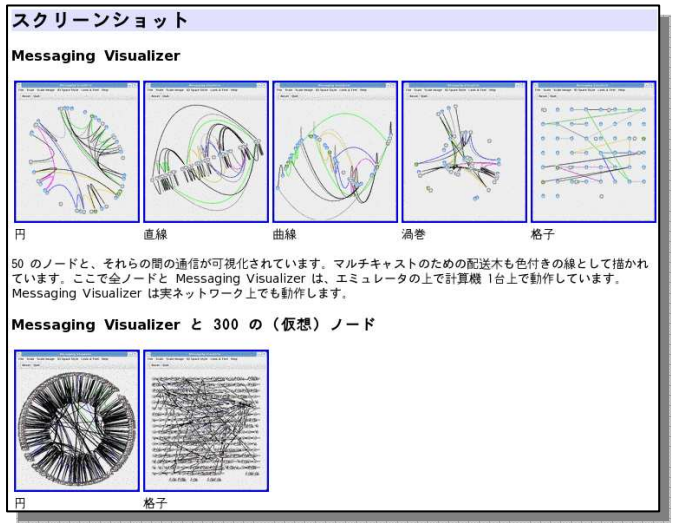
オープンソースソフトウェア，Overlay Weaver 研究プラットフォームとして

- http://overlayweaver.sf.net/ (SourceForge)
  - 2006年 1月 17日、リリース。
  - Apache License 2.0

- 状況 (2009/2/19 20時)
  - ダウンロード 12,500件。
  - メーリングリスト登録数
    - 英語 83名, 日本語 88名
  - Mixi コミュニティのメンバ数 153名



ウェブサイト

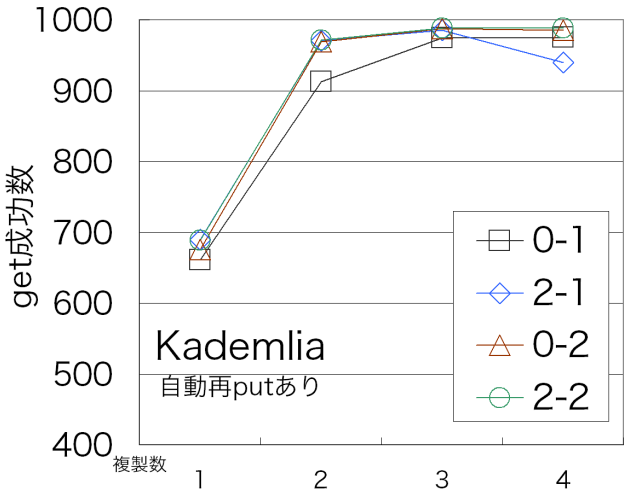


- 研究プラットフォームとして
  - ブラジル, ロシア, 英国, ハンガリー, フランス, 中国, シンガポール, …
  - 第三者が、ルーティング方式を追加実装: Symphony, EpiChord, …
  - 第三者による利用 / 応用多数:
    RDFのDB(産総研), XMLデータ検索(筑波大), 複製数の影響(U Federal do Rio Grande do Sul), 複数サーバの横断検索(お茶大), ウェブアクセス動向集計(東工大), 複製配置手法(早大), コミュニケーション向けユーザ管理サーバ(日立), 多次元範囲検索システム(NEC), …

peer-to-peer なので

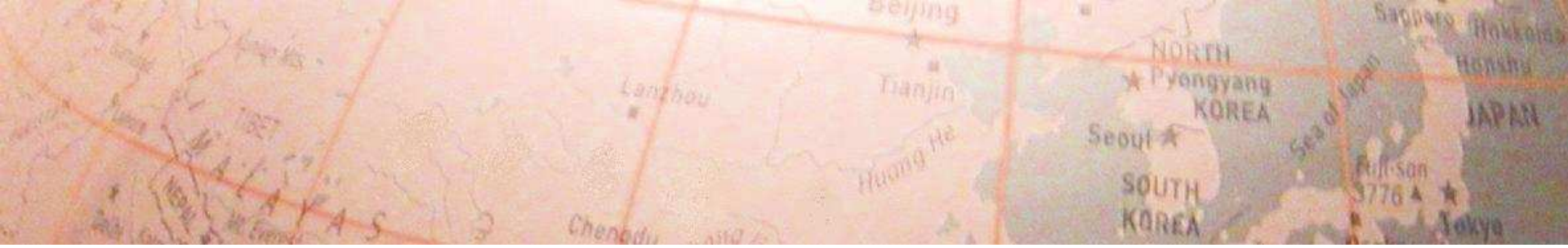
Overlay Weaver

- churn に耐える。耐えなきゃいけない。
  - churn: 絶え間ないノードの出入り
- 次の手法を実装
  - 複製
  - 加入時委譲
  - 複数 get
  - 自動再 put



churn の中でどれだけの get が成功するか

- 首藤一幸: “下位アルゴリズム中立なDHT実装への耐churn手法の実装”, 情報処理学会…論文誌, … 2008年 3月



サーバ側技術と peer-to-peer 由来技術の関係

つまりは

no-hop と
multi-hop の関係

no-hop と multi-hop の関係

1つの枠組みでとらえられる

- Wikipedia の DHT (分散ハッシュ表) についての項
  - degree (次数)　route length (経路長)

O(1)	O(log n)	← constant-degree DHT
O(log n)	O(log n / log log n)	
O(log n)	O(log n)	← 当初の DHT, 構造化オ
O(n^(1/2))	O(1)	

  - no-hop の key-value store は、こう位置づけられる

O(n)	**1**

- 「DHT (分散ハッシュ表)」は、
  従来的には、multi-hop が前提
  → no-hop の key-value store を DHT とは呼ばない。
- しかし no-hop であっても DHT / 構造化オーバレイの枠組み中に位置づけてとらえることが自然。

Overlay Weaver の no-hop 化

- パラメータ調整
  - するだけで、実は no-hop になる。
  - 例: Chord の successor 長や Pastry の leaf set サイズを大きくとる。
  - ノード情報がちゃんと流布するかは、多少不安。
- もっと真面目に
  - ノード情報を gossip か何かで流布させる？

サーバ側での peer-to-peer

- Microsoft の **cloud** (Azure Services Platform) 中での peer-to-peer 由来技術の活用
  - DHT - **分散ハッシュ表**
    - SQL Data Services の表は、Partition ごとに担当ノードが保持する。注: IDが近いノードに複製は作る。
    - Service Bus (pub/sub によるメッセージ配送サービス) では、エンドポイント (sb://…) の担当ノードが、そのエンドポイントに subscribe したノード一覧を管理する。
      - これ、i3 [Stoica2002] そのまんま。
  - DOLR - 宛先IDを担当するノードへの**メッセージ配送**
    - Fabric Controller での Messaging。

- Azure のルーティング方式
  - 両方向版 Chord とでも呼ぶべきもの。
    - 時計回り一方向 (Chord) ではなく、両方向に進める。これがいいのかどうかは ???

まとめ

- 従来的には、サーバ側 key-value store を DHT (分散ハッシュ表) とは呼ばない。
  がしかし、別物というわけでもない。
  同じ枠組みでとらえられるお仲間。

- cloud も peer-to-peer も大規模分散システム。
  活きる技術は割と共通。