

# 分散型ネットワーク座標系 Vivaldi の仕組み

上田達也

大阪市立大学大学院創造都市研究科  
都市情報環境研究分野

P2P/DHT 勉強会 in 関西 2006 Dec. 3

# 自己紹介（生い立ち編）

- 1964年12月3日，大阪市都島区に生まれる
  - こう見えてトシ食ってます
- 小学校生活を大阪府門真市で過ごす
  - が，河内の兄ちゃんにはならず
- 中学以降は都島区に戻る
  - が，都会的に洗練される事は無かった
- 大学では学業より音楽にはまる
  - 7回の表で，あえなく退場
- 他に出来る事も無かったので，プログラマになって現在に至る

# 自己紹介（仕事編）

- （有）うえだうえおうえあ
  - 2001年12月25日設立
  - 社員1名，パートタイム経理部長1名
- ネットワーク関連
  - 設計
  - コンサルティング
  - 雑誌記事（仕事？）
- ソフトウェア開発
  - C, Java, LL (Lightweight Language)
  - 設計
  - 製造
  - 火消し（←本当は嫌らしい）

# 自己紹介（音楽編）

- カウンターテノール歌手
  - 大阪コレギウム・ムジクム合唱団
  - 大阪H. シュッツ室内合唱団
  - 大阪H. シュッツ声楽アンサンブル
  - ヴォーカルアンサンブル・アウローラムジカーレ
- CD 出てます
- 12/17, 23 日にクリスマス・コンサートやります
  - 詳しくは個人的に…\_o\_
  - <http://www.collegium.or.jp/>

# 自己紹介（コミュニティ編）

- 関西 \*BSD ユーザ会
  - <http://www.kbug.gr.jp/>
- OpenBSD Web Page 日本語翻訳プロジェクト
  - <http://ja.open.4bsd.org/>
  - <http://www.openbsd.org/ja/>
- 関西オープンフォーラム (KOF)
  - <http://k-of.jp/>
- オープンソース系のイベントに顔出してスタッフやるのが趣味（？）らしいです

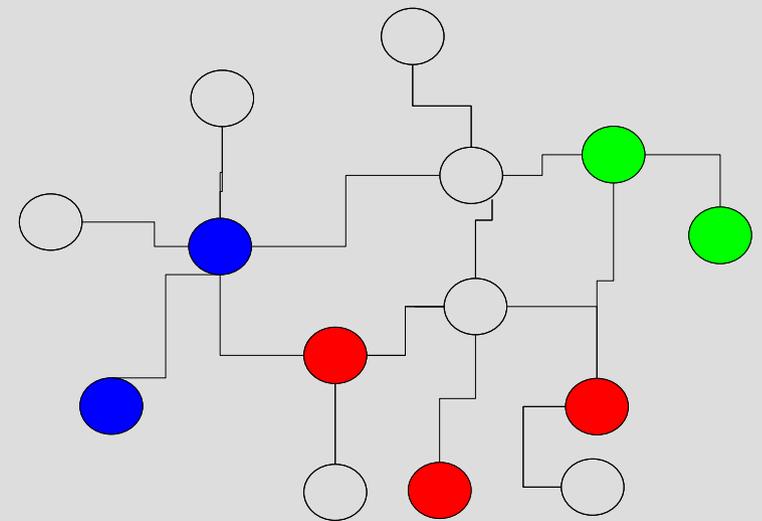
最近はあまり何も出来てません \_o\_

# 自己紹介（研究編）

- 大阪市立大学大学院創造都市研究科  
博士（後期）課程…社会人大学院生
  - 学生募集中です！（たぶん）
  - 詳しくは今日来ている教員に…（汗）
  - <http://www.gscs.osaka-cu.ac.jp/>
- 直近の論文  
上田達也，安倍広多，石橋勇人，松浦敏雄  
「P2P手法によるインターネットノードの  
階層的クラスタリング」  
情報処理学会論文誌，Vol. 47, No. 4, pp. 1063–1076, Apr. 2006

# 研究の背景

- インターネット上のノードのクラスタリング
  - ネットワーク的に近いノードをグループ化
  - インターネットでの分散アプリケーションに特に有用
- クラスタリングの難しさ
  - インターネット上では、事前にネットワークトポロジを知るのは困難



この図の様に  
簡単にはいかない！

# 既存のクラスタリング方式

- 集中型
  - サーバが全ノードの情報を集めてクラスタリング
  - スケールしない
  - 対故障性に不安がある
- 分散型（P2P方式）
  - 各ノードが対等に動作しクラスタリング
  - 例：TOPLUS
    - BGP ルータの情報等の外部情報に依存
    - 実際に使うことが難しい

# 提案手法の特徴

## 1) Pure P2P

- ▶ 耐故障性（高信頼性）
- ▶ 負荷分散

## 2) スケーラブル

- ▶ ノード数の増加に対応

## 3) 外部情報不要

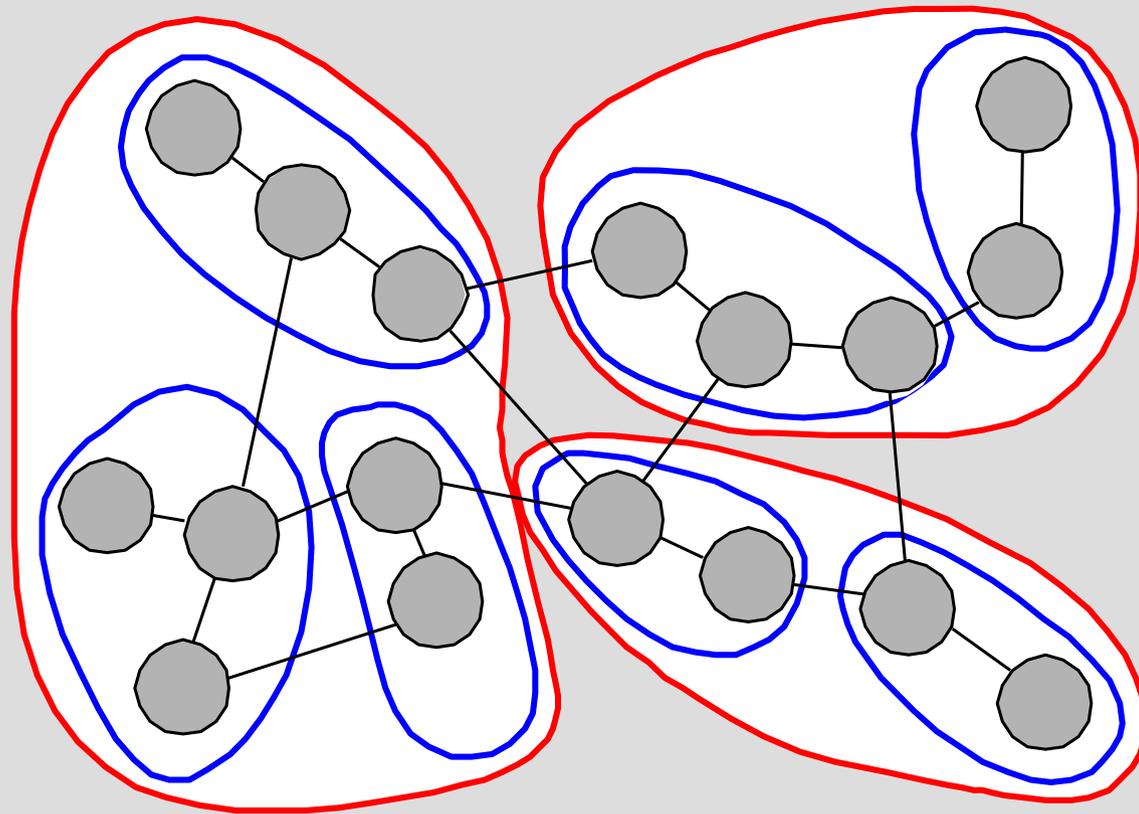
- ▶ ネットワークトポロジー情報は不要
- ▶ ノード間の距離（ホップ数）だけが測定できれば良い

## 4) 動的クラスタリング

- ▶ 事前に参加するノードは決まっている必要はない
- ▶ ノードは自由に参加離脱できる

# 提案手法の特徴 (Cont' d)

## 5) 階層的クラスタリング



# Agenda

( やっと本題・汗 )

- ネットワーク (仮想) 座標系概説
- Vivaldi 概説
- クラスタリングへの応用 (Just Idea!)

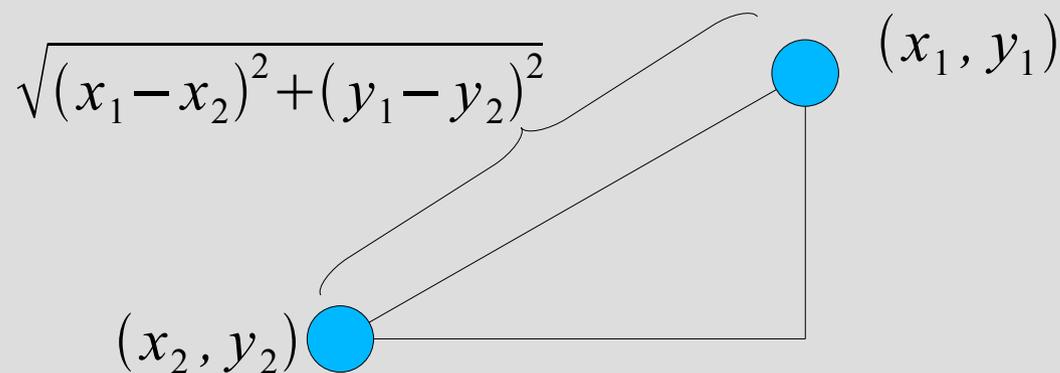


# ネットワーク座標系 (Cont' d)

- ネットワーク距離とは？
  - RTT, Hop 数など
- n次元ユークリッド空間では…
  - ユークリッド距離の定義

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

- 二次元での例



# ネットワーク座標系 (Cont' d)

インターネット上のノードを  
ユークリッド座標で表現すると…



ユークリッド距離が計算で求められる



実測しなくて良いので嬉しい！

ネットワーク距離とユークリッド距離に  
相関関係があればね…

# Vivaldi 前夜

## 集中型ネットワーク座標系

- GNP

T. S. E. Ng and H. Zhang.

Predicting Internet network distance with coordinates-based approaches

In Proceedings of IEEE Infocom, pages 170–179, 2002.

- 距離測定の規準となる「ランドマーク」が必要
- 各ノードはランドマークへの距離を実測し、座標を決定する
- ランドマークの選択が、推測距離の正確さを左右する

ランドマークが単一障害点になりうる



「分散型」とは言えない

# Vivaldi 前夜 (Cont' d)

## 集中型ネットワーク座標系

- NPS

T. E. Ng and H. Zhang.

A network positioning system for the Internet.

In Proc. USENIX Conference, June 2004.

- GNP の後継バージョン
- ランドマークノードの負荷軽減のために階層構造
- 嘘つきノードの影響軽減
- NAT 対応

- Lighthouse

M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti.

Lighthouses for scalable distributed location.

In IPTPS, 2003.

- GNP の拡張 (スケーラビリティ)
- ランドマークでないノードに問い合わせ可能
- その結果からランドマークとの相対座標を算出

# Vivaldi Contemporary

## 分散型ネットワーク座標系

- PIC

M. Costa, M. Castro, A. Rowstron, and P. Key.

PIC: Practical Internet coordinates for distance estimation.

In International Conference on Distributed Systems, Tokyo, Japan, March 2004.

- 座標軸毎に次元数 + 1 のランドマークとの距離を実測
- その際にランドマークの座標を取得
- Simplex(cf. GNP) 等を用いて推定した距離と実測値を比較し, 誤差が最小となるように自らの座標を決定
- 明示的にランドマークを指定する必要は無い

# Vivaldi 概説

- Papers

- R. Cox, F. Dabek, F. Kaashoek, J. Li, and R. Morris.  
Practical, distributed network coordinates.  
In *HotNets-II*, Nov.2003.
- FransKaashoek, Frank Dabek, Russ Cox and Robert Morris.  
Vivaldi: A decentralized network coordinate system.  
In *Proceedings of the ACM SIGCOMM '04 Conference*, pp. 149–160, Portland, Oregon, August 2004.

# Vivaldi 概説 (Cont' d)

- 分散型ネットワーク座標系
  - 各ノードが完全に自立的に動作
- バネの原理
  - ユークリッド距離と実測値の誤差を徐々に修正
  - この点はPICと類似
- Piggy-back
  - アプリケーションレベルのトラフィックに便乗
    - 測距のために特別なトラフィックを用いない

# Vivaldi vs. PIC

- 相互参照して面白
  - Vivaldi (2003 年)
  - PIC (2004 年)
  - Vivaldi (2004 年)
- PIC から見た Vivaldi の問題点
  - 誤差の調整幅小さいので一時に大量のノードが参加すると中々収束しない
  - 嘘つきノードへの対応が無い
- Vivaldi から見た PIC 問題点
  - 専用のトラフィックが必要 (Vivaldi は不要)
  - Simplex アルゴリズムをその都度動かすのでネットワークのダイナミックな変化についていけない
  - 誤差の調整幅大きいので座標が発振する

# 推定誤差 (Prediction Error)

- ノード間のユークリッド距離から推定したネットワーク距離と実測値の誤差

$$E = \sum_i \sum_j (L_{ij} - \|x_i - x_j\|)^2$$

$E$  = 推定誤差,  $L_{ij}$  = ノード  $ij$  間の実測距離,  $x_i$  = ノード  $i$  の座標

# バネの原理

- ネットワークを，物理的なバネを繋ぎ合わせて作った「網」にたとえる
- $E$ （推定誤差）が最小値となるように，バネを緩めるごとく，座標値を調整する
  - 座標軸毎に誤差を調整していく
- GNP で使われている Simplex の様な最適化アルゴリズムより計算量が少ない

# 単純な Vivaldi アルゴリズム

```
Simple_Vivaldi(rtt,  $x_j$ )  
// 誤差の計算  
 $e = rtt - \|x_i - x_j\|$   
// 誤差による力の方向を求める (uは単位ベクトル)  
 $dir = u(x_i - x_j)$   
// 力のベクトルを求める  
 $f = dir \times e$   
// 力の方向に少しだけ移動  
 $x_i = x_i + \delta \times f$ 
```

# 実装段階での調整

- Timestep (  $\delta$  )
  - 大きくとると早く収束する
  - 座標が発振してしまうかも
  - 誤差の大きさによって adaptive に取ると良い
- 座標モデルはどれが良いのか？
  - 2 or 3 dimensions Euclidian
  - 球型モデル
  - 円環モデル

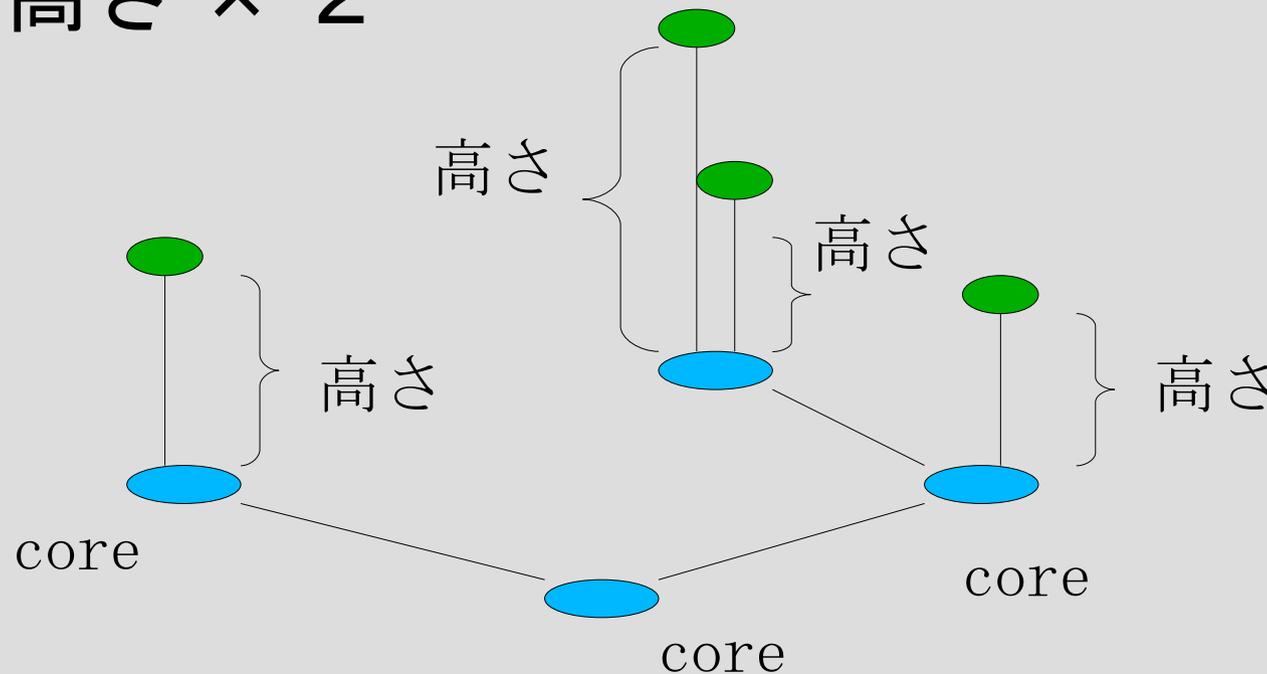
2次元+高さベクトルが良いらしい

# 高さベクトル付き二次元座標

- 高速回線だけで構成されたネットワークなら latency と物理的な距離は相関するが…
  - DSL, Cable Modem, Telephone Modem
- などの場合, core ノードにパケットがキューイングされる時間を考慮したモデルが必要

# 高さベクトル付き二次元座標 (cont' d)

- core ノードを平面座標に配置，リンクノードを core から上空に配置
- 同じ高さにあるノード間の距離は，平面距離 + 高さ  $\times 2$



# 高さベクトル付き二次元座標 (cont' d)

- ベクトル演算の拡張

$$[x, x_h] - [y, y_h] = [(x - y), x_h + y_h]$$

$$\| [x, x_h] \| = \|x\| + x_h$$

$$\alpha \times [x, x_h] = [\alpha x, \alpha x_h]$$

# 実装について

- BambooDHT に Java 実装
  - 実装してみたものの、BambooDHT では用途が無いので使っていないらしい
- p2psim に C++ 実装？
  - すみません、未確認です (C++ 嫌いなので・汗)

# デモ（二次元平面）

- $5 \times 5$  のグリッドなトポロジ
- 距離単位：ホップ数
- BambooDHT の Java 実装を使用
- 総当たりで実行
  - 本当はランダムでやった方が良いのだけど…
- レンダリングには graphviz を使用
- 30 回目くらいの所でほぼ収束

# クラスタリングへの応用 (Just Idea)

- 実測部分にアプリケーション・トラフィックを利用できるので、効率が良くなる
- Joinの際には、ユークリッド空間でのクラスタの重心を算出する
- 高さベクトル付き二次元座標での重心って？
  - 重み付き重心と考えて良いのかなあ？
    - これだと、2点間の重心は2点間の平面上のどこかになる
  - 高さも含めて平均地点を取る？
    - 極端に高い位置にあるノードの場合、支柱の途中に
    - 複数ノードの重心を取る場合、演算が煩雑

# 現在進行中

- DHT はトポロジーの考慮がされていない
- 隣の id が、実は地球の裏側に？
- id を決める時にトポロジーが考慮されると良いなあ
  - クラスタリングを適用してみましよう
  - Overlay Weaver と我々のシミュレータを合体したら、すぐ実験できるやん
    - と、言い始めて早 2 週間？ orz
    - 若い人たちに期待し始めている今日このごろです（弱）

# まとめ

- ネットワーク座標系
  - 集中型（ランドマークが必要）
    - GNP
    - NPS
    - Lighthouse
  - 分散型（ランドマーク，外部情報不要）
    - Vivaldi
    - PIC
- Vivaldi の概要
  - バネの原理
  - Piggy-back
  - **単純かつ実用的 → クラスタリングに適用**

ご静聴ありがとうございました