Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

ネットワーク座標系による ノード間の通信遅延管理手法の検討

中村 貴[†1] 菅谷至寛[†1] 大町真一郎[†1]

ネットワーク上の計算資源を利用して並列計算環境を構築することが注目を集めているが, この環境における処理時間は通信時間を内包する. そのため, 利用する計算資源を適切に管理し通信を最適化する必要があると考えられる. 本研究ではネットワーク上のノード間通信遅延を P2P ネットワークにより分散管理を行うことを目指す. 通信遅延を座標系により管理する手法として Vivaldi が提案されているが, 局所的に誤差が残り通信遅延の推定が精密に行えない場合がある. 本論文では座標系を階層型に拡張することを提案し, その効果を比較実験によって示す.

Consideration for Control RTTs by Decentralized Network Coordinate System

Takashi Nakamura,[†1] Yoshihiro Sugaya[†1] and Shinichirou Omachi[†1]

Recently constructing parallel computation environment by resources on the network is attracted attention. Under this environment, processing time involves communication time. Therefore, it need to control resources for optimaization the communication. Our study aims to manage RTTs by P2P network. Vivaldi which manages RTTs by coordinate system was proposed, but local error remain the coordinatesystem and in some cases we cannnot estimate RTTs. We improve Vivaldi by expanding hierarchical-coordinate system, and show the effectiveness by experiments.

†1 東北大学大学院工学研究科
Graduate School of Engeneering, Tohoku University

1. はじめに

近年並列化による計算ユニットの性能向上が試みられており, パーソナルコンピュータから科学計算用 PC クラスタまで幅広く取り入れられている. 並列処理自体はスーパーコンピュータなどでは早期から行われていたものではあるが, 近年においてはネットワーク上の計算資源を用いて並列分散処理を行う計算環境の構築など, 多様な方法で行われている. インターネットなどの広域ネットワーク上から計算資源を結びつけ, だれでも・どこでも計算資源を利用可能するシステムの構築が望まれている. この手法をグリッドコンピューティング[?)] と呼ぶ. グリッドコンピューティングが近年注目される背景として, コンピュータネットワークの伝達速度や安定性の向上があげられる. また, ネットワークに接続される計算資源の高性能化や各資源の遊休時間の存在も理由のひとつであると考えられる. ボランティアコンピューティング (例:SETI@home[1)]) はネットワーク上の計算資源を利用し並列計算を行う手法であるが, 資源の利用者と提供者が明確に区別され, 誰でも・どこでも利用可能とする設計思想は達成されていない.

そこで本研究では, ネットワーク上に存在する計算資源をシステムへの参加者なら誰でも利用可能とするシステムの構築を目指す. システムは参加者へ適切に利用可能な計算資源を通知できるものであり, さらに利用者の望む性能 (遅延が小さい, 性能が高い, 等) の検索を可能とする. 本システムではいかに資源を保有するノードを効率的に探索するかが課題となる. 本論文では特に通信遅延に関する検索に焦点をあて, 通信遅延を各ノードの座標値として表現し, 分散管理を行うことが効果的であることを述べる. また座標系による通信遅延の表現を行う手法である Vivaldi について述べ, 本システムにおいて利用を行う際に障害が存在することを説明し, 座標系を階層型に拡張することで改良を行う.

2. 座標系による通信遅延管理

2.1 計算資源管理システムに求められる要件

本システムにおいて参加者が並列分散計算を行う際には, 管理システムによって利用可能ノードの検索を行うことが考えられるが, 計算性能や通信遅延による参加ノードの検索が想定される. skip graph[2)3)] などの構造化オーバーレイを利用し, ID に性能の対応づけを行うことや非構造化オーバーレイを用いて Flooding による検索メッセージの伝搬を行えば計算性能による検索は容易に実現が可能である. しかし通信遅延による検索の実現は容易ではない. その理由として, まず全対全ノードの通信遅延というデータの量が膨大であることがあ

Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

げられる．また全対全ノードの通信遅延をシステムが管理する場合, 全ノードとの通信遅延を測定し分散管理システムへ登録を行う必要があり, 通信コストの面から現実的ではないといえる．そこで通信遅延に関して, 分散管理を行うことが容易であり, かつ分散管理を行うことで通信の最適化を行う上で有用となる指標として座標値による通信遅延の表現を検討する．

2.2 座標系による通信遅延推定

座標系を利用し, 座標系におけるユークリッド距離でノード間の通信遅延を表現することを考える．この座標系がノード間の距離を正確に表現することが可能であるとすれば, 管理システムは全ノードの座標情報だけを保持すればよい．各ノードはこの管理システムが保有する他ノードの座標情報にアクセスすることで他の全てのノードとの通信遅延情報を取得することが可能である．新規ノード参加時には該当ノードの座標を決定するだけで管理システムへの更新が行われ, 全ノード間との通信遅延を測定せずに通信遅延を把握することができる．また全参加ノードにおいて統 1 的な座標系を構築することにより, 通信遅延が小さいノードを発見することが容易になると考えられ, 利用可能な計算資源の中から最適なものを選択して利用することが可能になると期待できる．

3. ネットワーク座標系 Vivaldi[4)]

3.1 Vivaldi 概要

Debek らはネットワークに存在するノード間の通信遅延を座標系で表現する手法としてネットワーク座標系 Vivaldi[4)] の提案を行った．Vivaldi は通信を行ったノード間をばねでつなぐ力学モデル (図 1) に基づく手法で, 通信を行うたびに自ノードの座標修正を逐次的に行うものである．ばねモデルによる修正を実現するため Vivaldi における座標修正は以下の式に基づいて行われる．

$$N_i = N_i + \delta \times \ Error(i,j) \times u(N_i - N_j),$$

$$Error(i,j) = rtt_{i,j} - |N_i - N_j|.$$

N_i はノード i の座標, u は単位ベクトルを表す．$rtt_{i,j}$ はノード i とノード j との間で計測される通信遅延 (RTT) であり, $Error(i,j)$ はノード i,j 間で通信が行われた際に, 計測された実通信遅延と座標を修正する前の座標系から得られる推定通信遅延との誤差を表す．δ は 1 回の通信でどの程度修正を行うかを表す定数である．

Vivaldi では, 各ノードが上記のアルゴリズムと式に従った座標修正を自律的に行う．さらに Vivaldi の特徴としてあげられるのが, アプリケーションレベルの通信に便乗して通信遅延を計測することである．この特徴のため座標系を構築・維持するための通信が必要ないという利点を持つ．また通信を行うたびに通信対象との座標修正を逐次的行うため, ネットワークへの参加が保証されるノードとの位置関係を正しく保つ方向へ修正が行われるという特徴を持つ．

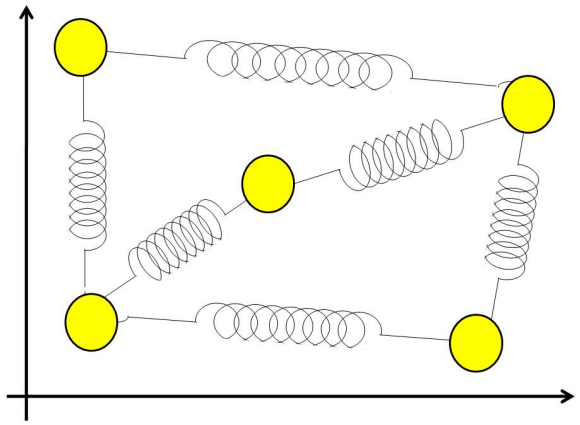


図 1 Vivaldi:ばねモデルを用いた座標系

3.2 Vivaldi の成果

座標系によるマッピングが 2D メッシュなどのネットワークにおいてはほぼ完全に可能であることを示す 1 方, Debek らは PlanetLab[5)] を利用した実ネットワークモデルに対する Vivaldi の利用・評価を行った．Debek らは Vivaldi の性能評価として, 座標系から推測される各ノード間の通信遅延と実通信遅延の相対誤差を評価指標として用い, 座標系として用いる次元や形状による誤差分布を評価した．その結果, 単純に次元を増やすことで誤差を小さくすることが可能であることや球形空間表示による座標空間は 2D 座標系に劣ることなどが示された．

また Debek らは 2 次元座標系に高さの概念を加えた 2D+h 次元の座標系を提案し, その評価を行った．2D+h 次元においては各ノードは 1 次元の高さを正の値として持ち, 2 次元座標系から得られるユークリッド距離に各ノードの高さを加えることで通信遅延の表現を行うものである．2D+h 次元を用いることにより, 単純に次元数を増やすよりも座標系全体の誤差を小さくすることが示された．

Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

3.3 Vivaldi の問題点

Vivaldi の座標修正式における δ はシステム全体の振動を小さくし, 座標を収束させるためにある程度小さい値を指定する必要がある. この理由によってネットワーク上で近傍に存在するノード群を複数座標系でマッピングした場合, 適切な位置どりをすることができないノードが存在する可能性が考えられる. これは Vivaldi がばねモデルであるために, 全体での誤差を最小にするものの局所的な誤差を残すことに起因する. その結果座標修正が行わず, 離れた位置にあるべきノードを近傍ノードと誤認識してしまうような事態が発生する. 各参加ノードが自身と最も通信遅延が小さいノード (もしくはノード群) の探索を可能とすることが本研究における座標系の導入理由であるため, Debek らが評価したシステムにおけるノード間相対誤差は本研究が評価するものと若干の違いがある. つまり, 本研究では座標系から推定される近傍ノードが実ネットワークにおいても近傍ノードであることが重要であり, 相対誤差が小さくなったとしても, 座標系において推定される通信遅延により近さの順に並べられた順序が実ネットワークにおけるものと異なっては座標系による管理の効果が薄くなってしまう. 前述の適切な位置どりができないノードの存在により, 正確な順序推定がなされない可能性は高く本研究において座標系の利用を行う際に障害となると考えられる.

また Vivaldi はネットワーク全体における誤差を最小にする方向に動作するシステムであり, そのためネットワーク上で近傍に存在するノードの集まりをひとつのノード群とすると, システムに参加しているノードの中で存在比率が高いノード群を中心に他のノード群が位置を修正していく. したがって存在比率が高いノード群は高い精度で他ノードとの通信遅延を座標系から推定することが可能である. しかしその他のノード群では, 存在比率が高いノード群との位置を保つ状態で座標を修正しなければならないために, 他のノード群との区別が困難になり推定精度が低くなってしまうという問題が起こりうる. これは前述の問題と同様に本研究における障害となると考えられる.

4. 階層型座標系による通信遅延管理

4.1 階層型座標系

計算資源共有を行うために参加ノード間の通信遅延を座標系で管理する際に起こる問題について述べたが, 座標系を階層型に拡張することで問題の解決を図る. 座標系の次元数を増やすことで誤差を小さくすることが可能であるが, 高次元の座標系を用いることはそれだけ管理するデータ及びデータを作成するための通信遅延計測が増えることになり, 座標系の導入動機を満足しない. 階層型に拡張する狙いは, 1 つの座標系では不正確な推定しかできないノードのみを含む上位の座標系を作成することにより, 推定が不正確なノード間で再び座標の適正化を行うことで推定精度をあげることである (図 2). 単一の座標系では存在比率が高いノード群を中心に座標の修正を行っていたため, それ以外のノード群では誤差を残していた. 階層型座標系では誤差が残るノード群の通信遅延を新たな座標系で表現するために, 新しい座標系ではもとの座標系と比較してノード間の誤差を小さくすることが可能である.

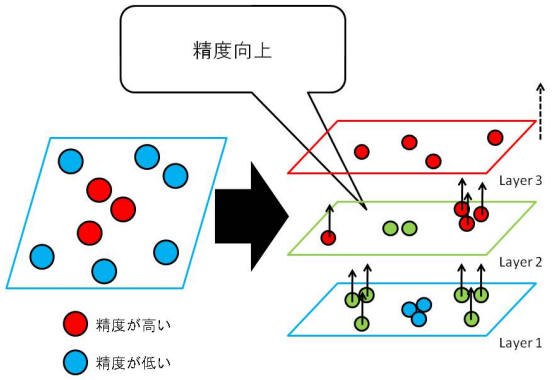


図 2 階層型座標系

4.2 ノードの安定性

階層型座標系では通信遅延の推定が不正確なノードにより新たな座標系を構築するため, 推定が正確に行えているノードと不正確なノードを分離することが不可欠である. 座標系から通信遅延を精密に推定できているかどうかを判断する指標として, ノードがその座標系で安定しているかどうかは役に立つ指標であると考えられる. なぜなら, 通信遅延を精密に推定できているノードは実際に通信を行った際に通信相手との座標系から得られる推定通信遅延と実通信遅延の誤差が小さく, 座標修正がほとんど行われないからである. したがってノードが座標修正を行う際に取得される誤差を元にノードの安定性を求めることで, ノードの分離が行えると考えられる.

4.3 安定度によるノードの階層移動

安定性の指標として, 以下の要領で求められる安定度 ($Stability(i)$) を各ノードが自律的に計算することを考える.

Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

$$Stability(i) = \begin{cases} +1 & Error(i,j) < e \\ -1 & Error(i,j) \geq e \end{cases}$$

e は定数でノードの座標修正値から安定しているかどうかを判断する閾値である.

ネットワークにおいて通信が十分に行われ座標修正によるノードの振動が微小になり, 座標系が 1 定の落ち着きを得た際には存在比率の高いノード群の近傍には比較的通信遅延が小さいノードが集まるため安定度は増加を続けることが予想される. 存在比率が高いノードを中心とした座標系が構築されるために, 逆にその他のノードは近傍に通信遅延の大きなノードの存在する可能性が比較的高く, 安定度は減少を続けることが予想される. したがって 1 定の安定度を取得したノードが存在する座標系において, 安定度が負の値を持ったノードは不安定で座標修正を繰り返していると判断することができる. 提案手法では 1 定の閾値 (C) を設け, ある層の座標系において閾値を越える安定度を持ったノードが出現した際には安定度が負の値を持つノードを全て上層へコピーをする. これにより不安定なノードの安定化を別の層で行うことができる.

階層型座標系におけるノード間の通信遅延推定手法及び座標修正方法は後述するが, 1 度不安定と判断し上層へコピーされたノードも下層において再び安定する場合が想定される. その際には安定した階層より上の階層に存在する自身を消去し, 階層の下降が行われることとなる.

4.4 階層型座標系における通信遅延推定

1 つのノードが多数の層に跨って存在するようになることが本手法の特徴であるが, 通信遅延を推定する際に, 用いる座標系の階層を選択する必要がある. そこでふたつのノード間の通信遅延を推定する際に座標情報を取得する階層を, 共に存在している階層のうち最も上の階層で行うこととする. ノードが存在する階層のうち上層にて比較を行うことで, 安定した状態で通信遅延推定を行うことを可能とし, また無駄に下層において座標修正を行って他のノードの安定を崩すこともない.

4.5 管理ネットワーク

各参加ノードは所属する階層の数だけ座標値を持つ. したがって形成されている座標の数だけネットワークを構築し, そのネットワークを用いて各階層の座標値を管理する. まず各階層において所属するノードの座標を管理し, 効率的に階層に存在するノードの座標値を管理することを考える. これについては多次元空間においての検索効率や管理ノードの負荷分散を考えてネットワークを構築すればよく, ZNet[6][7] や GLOBASE.KOM[8] などを用いることで簡単に実現が可能である. 加えて負荷の分散や検索の効率から, 近傍ノード間で直接リンクを持ち近傍ノードの探索に利用することが考えられる. リンクを形成したノードの利用頻度や, リンク先のノードとの通信遅延の分散を利用するリンクの更新などが考えられるが, 詳細は今後の課題としたい.

表 1 日本の主要都市人口

都市	人口 [百万人]	12 都市内での割合 [%]
Tokyo(区合計)	8.5	30.5
Yokohama	3.6	12.9
Osaka	2.6	9.3
Nagoya	2.2	7.9
Sapporo	1.9	6.8
Kobe	1.5	5.4
Kyoto	1.5	5.4
Fukuoka	1.4	5.0
Kawasaki	1.3	4.7
Saitama	1.2	4.3
Hiroshima	1.2	4.3
Sendai	1.0	3.5

5. 評 価 実 験

シミュレーションにより提案手法と既存手法 Vivaldi の比較実験を行う. またシミュレーションから得られた座標値を元にオーバーレイネットワークを構築し, その性能について評価を行う.

5.1 実験方法と条件

実験は仮想ネットワークにおいて通信遅延を各ノード間で設定し, この通信遅延を基に座標系を構築することで行う. 仮想ネットワーク構築の際には後述の統計などを参考にすることで, 実際のネットワークに近いと考えられる通信遅延の設定を各ノード間で行った. 仮想ネットワークは日本の都市人口分布[9](表 1) とインターネットサービスプロバイダーの利用人数動向[10](表 2), 及び NTT コミュニケーションズグローバル IP ネットワーク[11] を参考に参加ノードの設定を行った. また Internet eXchange[12] の存在を東京と大阪に仮定することで ISP 間の通信を考慮し, ノード間の通信遅延を設定した.

5.2 実 験 手 順

システムに参加するノード数を 2000 とし, 各ノードが他のノードと通信を行い座標の修正を行う動作を 2000 回繰り返した. 通信相手の選定方法は以下である.

Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

表 2 日本の ISP 利用動向

ISP	利用者割合 [%]	6ISP 内の割合 [%]
OCN	16.5	29.4
Yahoo!BB	13.3	24.1
BIGLOBE	7.2	13.1
@nifty	6.7	12.2
ぷらら	6.6	12.0
au one net	5.0	9.2

$$\text{階層型座標系} \begin{cases} \text{同 1 階層 かつ 推定 RTT} < 5[ms] & (95\%) \\ \text{ランダム} & (5\%) \end{cases}$$

$$\text{Vivaldi} \begin{cases} \text{推定 RTT} < 5[ms] & (95\%) \\ \text{ランダム} & (5\%) \end{cases}$$

ランダムな通信相手を選択する理由として, ノードのネットワーク座標系全体における位置関係を正しく保つために近傍ノード以外との通信が必要とされるためである. なお, 95%で選択される条件式から得られる通信相手が存在しない場合もランダムに通信相手を選定する.

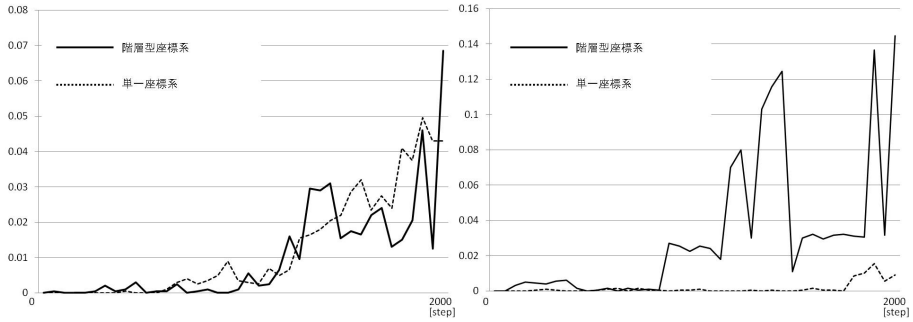
なお座標の修正を行う際に用いられる δ を 0.05, 階層型座標系の階層移動に関するパラメータとして, 安定度の増減に関する指標 e を 1.0, 安定ノードの出現を判断する閾値 C を 200 とした.

5.3 実験結果

まず既存手法と提案手法の比較評価を行うため, 2 つの手法を用いて top-k 検索の精度を比較する. 本研究における top-k 検索とは, システムに参加しているノードが座標系から推測される通信遅延に基づき, 自身に近い順に k 個のノードを検索することである. 並列分散環境を構築するうえで, 1 つの処理内容が実行完了するまでの時間は, 並列化を行った際に利用した計算資源の中でもっとも通信遅延が大きいものに左右されると考えられるため, top-k 検索によって取得された k 個のノードのうち実際の通信遅延が最も大きいノードとの通信遅延を本実験での評価指標とした. この指標として以下で定義される *topk_e* を用いる.

$$topk\_e(k) = \frac{estimateRTT(k) - realRTT(k)}{realRTT(k)}$$

estimateRTT(k) は座標系から推測された最近傍 k 個のノードとの間で実通信遅延と設定された値の最大値, *realRTT(k)* は仮想ネットワークの設定の時点で取得できる最近傍 k 個のノードとの間に設定された通信遅延の最大値である.



(a) *k=5* (b) *k=30*

図 3 *topk_e* の比較結果

k=5,30 において *topk_e* が 0.1 以下であるノードの割合について図 3 に示す. 検索を行うノード数が増加した際に, 階層型に拡張を行った効果が顕著に表れている. 従来手法では精密に推定がほぼできていなかった *k=30* において 7 倍程度の精度向上が見られる.

次に階層型座標系によるマッピングの変化を確認した. インターネットサービスプロバイダー 1 に所属するノードの座標値について, layer0 と layer1 の分布を図 4, 図 5 に示す. layer0 の分布図は単一座標系のものとほぼ同義である. layer0 において座標系に散在していた Kobe や Hiroshima のノードが階層移動したことで layer1 では近傍にマッピングされていることがわかる. 階層型座標系に拡張した効果により, 低階層において混在していた他のノード群が階層があがることで分別することが可能になっていることがわかる.

6. 結　　論

大規模並列分散環境をネットワーク上に存在する計算資源によって構築するために, 参加ノード間の通信を最適化するための手法として座標系による通信遅延の管理手法が効果的であることを述べた. 座標系による通信遅延の管理を行う手法として Vivaldi の解説を行い, 存在比率が高くないノード群において誤差が残ることが, 本研究が目的とするシステムにおいて障害となることを述べた. 本研究では座標系の階層型への拡張による Vivaldi の改良を

Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

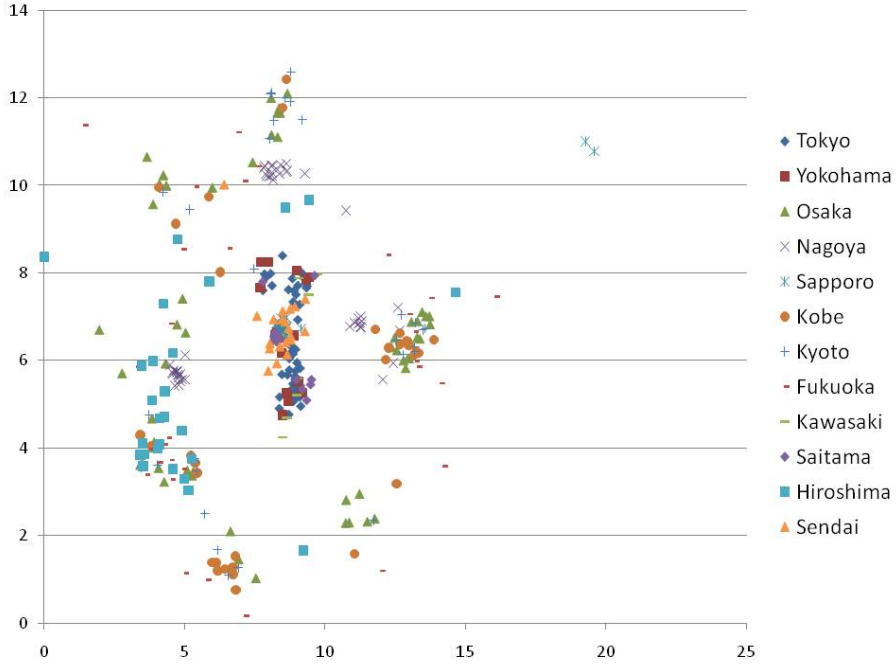


図 4 単一座標系 (階層型座標系 layer0)

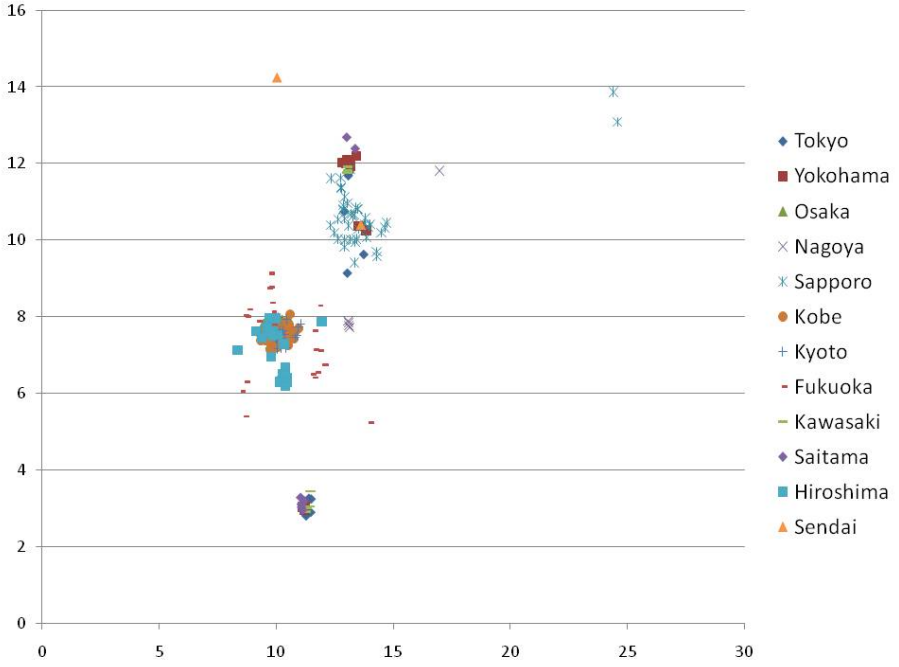


図 5 階層型座標系 layer1

Vol.2011-DPS-146 No.25
Vol.2011-CSEC-52 No.25
2011/3/11

提案した. 提案手法の性能評価として比較実験を行ったところ, 近傍ノード推定を行った際に精度の向上がみられることを示した. また, 提案手法によって形成された座標系を実際にマッピングすることで, 階層型座標系により不安定ノードの分離が可能となることを示した.

不安定ノードの分離を行う際に単純に不安定ノードをひとつ上の階層にコピーすることで分離を行っているが, 安定ノードの喪失や新規階層での安定ノードの集結が座標系の乱れを招いていると考えられるため, 分離を行った際に座標系の乱れが大きくならない改良が必要である. コピーを行った際に座標の修正値を大きくし, 早期に安定するようにする手法や, 下層において安定しているノードをランドマークとして上層にコピーする手法が考えられるが, これらの手法について検討・導入を行うことを今後の課題としたい.

また, 実際にネットワーク上で運用を行うためにはノードの情報を分散管理するためのオーバーレイネットワークの構築が必須であるので, 安定ノード間のネットワークを含め, その構築及び評価実験を今後の課題としたい.

参 考 文 献

1) SETI@home: http://setiathome.berkeley.edu/
2) James Aspnes and Gauri Shah: Skip graphs, *ACM Trans, on Algorithms*, Vol.3, No.4, pp.1-25, 2007.
3) William Pugh: Skip lists: Aprobabilistic alternative to balanced trees, *Communications of the ACM*, vol.33, pp.668-676, 1990.
4) Frans Kaashoek, Frank Debek, Russ Cox and Robert Morris. Vivaldi: A decentralized network coordinate system. *Proceeding of the ACM SIGCOMM '04 Conference*, pp. 149-160, Portland, Oregon, August2004.
5) PLANETLAB: http://www.planet-lab.org/
6) Shu Y., Ooi B., C. Tan, K.-L, and Zhou A.: Supporting multi-dimentional range queries in peer-to-peer systems, *Proceedings of Fifth IEEE International Conference on Peer-to-Peer Computing(P2P 2005)*, pp. 173-180, 2005.
7) Tropf H., and Herozog H.: Multidimentional Range Serch in Dynamically Balanced Trees, *Angewandte Informatik(Applied Informatics)*, Wiesbaden, Germany, Viewg Verlag, pp. 71-77, February 1981.
8) Aleksandra Kovavević, Nicolas Liebau, and Rald Steinmetz, Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search, *proceeding of Seventh IEEE International Conference on Peer-to-Peer Computing*, pp. 87-94, 2007.
9) 総務省統計局 政策統括官 (統計基準担当) 研修所 : http://www.stat.go.jp/data/-kokusei/2010/index.html
10) 「インターネット白書 2010」, インプレス R&D インターネットメディア総合研究所 (著), 財団法人インターネット協会 (監)
11) IP バックボーン : http://www.ocn.ne.jp/business/bocn/backbone
12) WIDE プロジェクト: http://www.wide.ad.jp/index-j.html