情報処理学会
グループウェアとネットワークサービス研究会
2014年11月27日, 28日

センサデータサーバにおける空間情報を用いたデータ分散手法の検討

永井 琢也[1]　満田 成紀[2]　福安 直樹[2]　松延 拓生[2]　鯵坂 恒夫[2]

概要：
センサデータを提供するサービスは，特定のデータベースノードやネットワークに障害が発生した場合でも継続利用可能であることが望ましい．この際，各ノードが担当する空間座標を制御することで，障害発生時にも有用性の高いセンサデータ処理を実現できるようになる．本稿では，センサデータサーバの構築に P2P 型分散 KVS データベースを利用し，空間座標によるデータ分散はコンシステントハッシングと Z 曲線を組み合わせることで実現した．また，実運用状況を考慮したセンサデータの検索実験を行った．

1. はじめに

センサを用いて実世界の様々な環境や状況を観測し，その結果を利用するアプリケーションが多く存在する．多様なセンサ種に対応したセンサデータサーバを用意し，データを一元的に蓄積することで，センサアプリケーションの開発が容易になる [1]．センサデータは持続的に生成されつづけるものであるため，センサデータサーバは，システムを停止することなく増えつづけるデータを管理しなければならない．この課題を解決する方法の 1 つに分散化がある．分散環境を構築することで，スケールアウト戦略によるリソースの追加が可能になる．これにより，処理能力の不足や容量が圧迫された状況になった場合に，柔軟に対応することが可能である [2]．

分散データベースを構築する場合，大きく分けて，マスター・スレーブ型と P2P 型の 2 通りのアーキテクチャが考えられる [3]．持続的に生成されるセンサデータを扱うためには，単一障害点の存在は望ましくない．また，センサデータの特徴として単純な処理要求が大量に発生することから，処理要求の受付も可能な限り分散化させることも必要である．これらのことから，P2P 型分散データベースとしてセンサデータサーバを構築することが望ましいと考えられる．

本研究は，センサデータサーバの構築において P2P 型分散データベースの特徴である可用性 (Availability) と分断耐性 (Partition-tolerance)[4] を積極的に活かす手法を検討したものである．これらの性質は，特定のノードやネットワークに障害が起きてもシステムとしては継続動作可能であることを意味している．しかし，そのような障害が起きた際には，故障したノードや接続できないノードが保持していたセンサデータにアクセスできないことから，障害発生時に利用できるセンサデータは制限される．ノード間のデータ複製によって障害発生時に利用できるセンサデータを増やすことは可能であるが，過度の複製処理は平常時のシステム全体のパフォーマンスを落とすことにつながり，避けなければならない．

そこで，本研究では，障害発生時の制限されたセンサデータの利用においても，有用性の高い処理が可能となるデータ分散手法を提案する．センサデータの特徴として，すべてのデータに時間情報と空間情報が付与されていることがあげられる．このうち，空間情報に着目すると，局所的に閉じた地理空間に含まれるセンサデータ群を対象とした処理要求が多いことが想定される．したがって，障害発生時においても，ある程度のサイズの空間範囲に含まれるセンサデータ群については，欠けることなく利用可能であることが望ましいと言える．これを実現するために，分散データベースで利用されているコンシステントハッシングに，空間充填曲線の一種である Z 曲線を組み合わせることで，各ノードが保持するセンサデータを空間情報を用いて制御する．

2. P2P 型分散 KVS

センサデータサーバには，増え続けるセンサデータに対応するため，システムを停止することなくスケールアウト

[1] 和歌山大学大学院システム工学研究科
Wakayama Univ., Graduate School of Systems Engineering
[2] 和歌山大学システム工学部
Wakayama Univ., Fucalty of Systems Engineering

が可能なことが望ましい．P2P 型ネットワークで分散環境を構築し，KVS のデータモデルを持つデータベースが構築に適していると考えた．

2.1 KVS と P2P ネットワーク

KVS とは，データを保存する際のデータモデルの一種である．キーとバリューの組み合わせを一塊のデータとして保存する．バリューはキーに紐づけられているため，データを検索する場合にキーを使用する．KVS はシンプルなデータ構造のため，複数のサーバでデータを管理することに適している．データ容量や処理能力の不足をサーバの追加によって補うスケールアウトが容易にでき，ビッグデータを扱うシステムに利用されている．

P2P ネットワークとは，単一障害点を持たないノード同士の接続によって構成されるネットワークである [5]．一カ所に障害が発生した場合でも，他のノードが処理を補うことが可能で，全体のシステムを継続することができる．単一障害点を持たないため，すべてのノードがリクエストを受け付け、レスポンスを返すことが可能である．単一障害点を持つ分散環境に比べ、高い可用性を低コストで実現できる．

2.2 コンシステントハッシング

P2P 型のデータベースの多くは，DHT(Distributed Hash Table) によってデータを分散保存する．DHT の一種にコンシステントハッシングがある．コンシステントハッシングとは環状のハッシュ空間を用いて，データの保存先を決定するためのアルゴリズムである [6].

図 1 に 4 つのノードで構成した分散環境におけるコンシステントハッシングの例を示す．キーからハッシュ値を生成し，それを元にデータを各ノードに振り分ける．各ノードには担当するハッシュ値の範囲が設定される．図 1 のような構成でコンシステントハッシングを構成する場合，キーから生成したハッシュ値が 0.2 だとするとノード 2 に保存される．コンシステントハッシングのメリットはデータを均等に分散できることに加えて，ノードを追加，削除する場合のシステム全体の負荷が少ないことが挙げられる．新たにノードを追加する場合は，既存のノード間に配置されるようハッシュ値を割り当てるだけでノードの追加を行うことが可能である．

2.3 P2P 型データベースの検討

コンシステントハッシングを採用している P2P 型データベースとして，Apache「Cassandra」[*1]，Basho「Riak」[*2]，LinkedIn「Voldemort」[*3] が候補として挙げられる．これ

[*1] http://cassandra.apache.org/
[*2] http://basho.com/
[*3] http://www.project-voldemort.com/voldemort/

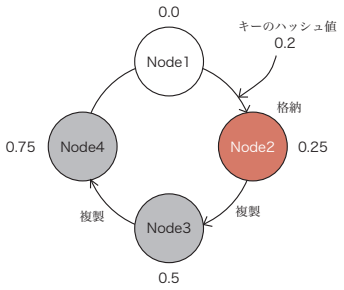

図 1 コンシステントハッシングの例

表 1 データベース比較結果

	通信方式	検索	センサ種の追加
Cassandra	P2P	◎	×
Riak	P2P	○	○
Voludemote	P2P	×	○

らのデータベースを検索機能とセンサ種の追加の観点から比較した．表 1 に比較表を示す．

Cassandra のデータモデルはキースペースやカラムファミリと呼ばれる RDB におけるテーブルのような構造をあらかじめ作成する．データに構造を持たせることで高度な検索が可能である．しかしセンサデータを蓄積することを考慮した場合，新たなセンサ種，データ規格の追加を行うためには，キースペースを追加設定する必要がある．

Riak と Voldemote はよく似た機能を提供している．この 2 つのデータベースはデータを JSON や XML 形式で格納できるため，センサの種類や規格の違いによってデータの格納が制限されることはない．ただし，Cassandra のようなデータ構造を持たないため，高度な検索は苦手である．Riak と Voludemote を比べると，Riak はデータをグループ化するバケットという概念を持っている．このためバケットにセンサ種を指定するなど Voldemote よりも検索機能に幅を持たせることができると考えた．検索機能とセンサ種の追加のどちらにも対応している Riak を本システムの構築に利用した．

3. ネットワーク障害を意識したデータ分散方法の提案

本システムは，大量のセンサデータを蓄積し検索が可能な分散データベースサーバを目指している．通常時は分散保存されたすべてのセンサデータに対し，検索，データ取得を可能にする．加えて本システムは，ネットワークが切断されてしまっても局地的なサービスであれば継続できることを目標にしている．ネットワークから分断されたノードが，単体で稼働しつつ局地的なサービスを継続するためには，ノードが全体のネットワークから分断されてしまった状態でも，一定の範囲のデータを参照できることが求められる．この要件を満たす方法として，各ノードに担当する空間座標範囲を設定し，その範囲のデータを蓄積する

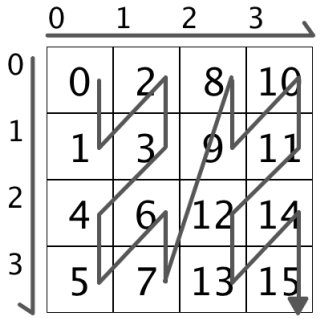

図 2 Z 曲線の例

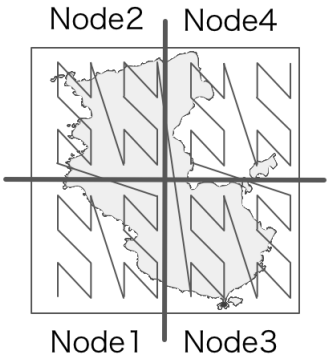

図 3 Z 曲線を用いて和歌山県を分割した図

手法を提案する．単体ノードが担当する座標範囲に限り，ネットワーク障害時でもセンサデータ提供が継続して可能となる．

ノードが局地的なデータを保持することは，各センサがデータベースを指定して保存することでも可能である．しかし，新たなノードを追加した場合，無数のセンサの設定を変更する必要がある．また，GPS はセンサそのものが移動するため，データの保存先を指定してしまうと，空間によるデータ分散ができない．そのため手法を実現するために，コンシステントハッシングと Z 曲線を組み合わせてセンサデータサーバを構築した．

3.1 Z 曲線

Z 曲線とは，空間充填曲線の 1 種で，多次元空間を一筆書きでなぞる 1 次元の曲線である [7]．Z を描くような形で 2 次元の空間を隙間なくなぞることが可能である．空間座標のような緯度，経度の 2 次元空間を Z 曲線を用いることで 1 次元の情報に変換できる．座標点に一意な Z 値を与えることができる．図 2 に Z 曲線の例を示す．矢印がでなぞられた線が Z 曲線である．空間を一筆書きで表すことで，図 2 のような座標空間を 1 次元情報によって分割が可能となる．例えば 0 から 3，4 から 6，8 から 11，12 から 15 といったように Z 値を用いて 2 次元空間を 4 つの空間に分割することが可能である．

表 2 各ノードが担当するハッシュ領域

ノード	ハッシュ値 (h)
1	$0(1.0) <= h < 0.25$
2	$0.25 <= h < 0.5$
3	$0.5 <= h < 0.75$
4	$0.75 <= h < 1.0$

3.2 Z 曲線を用いたコンシステントハッシング

本システムはコンシステントハッシングと Z 曲線を組み合わせることで，分散かつローカルなデータを保持することを可能にする．コンシステントハッシングでは，0 から 1.0 の値をハッシュ関数によって生成し，データの保存ノードを決定している．理論上はリング状にノードを繋げるため，0 と 1.0 は同値である．通常コンシステントハッシングでは，ハッシュ値をランダムで生成する．このままでは一定の範囲のデータを指定のノードに保存することはできないので，Z 曲線を用いてハッシュ値を生成する．本稿では和歌山県を対象とした実装を行った．和歌山県が入る (135.0, 33.0)，(136.6, 34.6) の 2 点を対角とした正方形領域を対象としている．この範囲を Z 曲線に変換し，それをコンシステントハッシングに対応づけた．Z 曲線の値（Z 値）に変換すると座標 (135.0, 33.0) は 0, 座標 (136.6, 34.6) は 0.99999 に対応する．

Z 曲線によって，和歌山県を図 3 のように座標分割が可能となる．今回の実装では座標空間を 4 分割した．Z 曲線は空間のなぞり方に特徴がある．最も単純に範囲を分割するため，正方形で範囲を区切っている．各ノードが担当するハッシュ値を表 2 に示す．

ネットワークが不安定，または障害が起った際でもデータを収集，提供できることが提案手法の特徴である．この特徴を活かすと，防災センサネットワークなどへの活用が考えられる．集中豪雨で発生した土砂崩れなどで，全体のシステムとのネットワークが切断されてしまう可能性がある．災害など発生した場合，まず最初に地域や集落単位で安否の確認や，情報収集が求められる [9]．ハザードマップのような災害時用のアプリケーションが，避難場所などの情報を得ることが可能となる．本システムはそういった場合に，電力とローカルなネットワークがあればノードが保持しているデータを提供することが可能である．また，災害時の対応を行いつつ，その地域の情報を災害前から災害中，そして災害後まで続けて収集できる．これにより災害時の地域の動きを観測でき，その後の防災に活かすことが可能である．

4. システム検証

4.1 システム復旧時

位置情報によりデータを分散保存しているシステムを想定する．各ノードは地理的に分散していて，指定された担

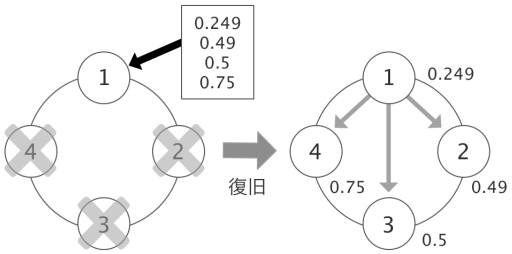

図 4 復旧時のデータの移動

当範囲のデータを保存する．このような状態で，災害などでネットワークが分断されてしまった状況を考える．各ノードは稼働しているが，物理的に回線が分断されたことでネットワーク的に独立してしまう状態が考えられる．

提案手法の特徴はシステムを P2P 接続により構成することで，単一障害点がないことである．単一障害点がないということは一カ所でシステムを管理するノードがなく，すべてのノードがリクエストを受け付け，レスポンスを返すことが可能な状態である．また Z 曲線によってデータの保存範囲を指定することで，自ノードの場所が含まれる範囲のデータを保持している．これにより，あるノードがシステムから独立してしまった状態でもそのノード単体でサービスを提供し続けることが可能で，災害中などもデータを保存し続けることが可能である．

提案手法はコンシステントハッシングをデータ分散手法に選択したことでシステムを構成するノードの追加，除去が柔軟に行えることも特徴である．災害後にネットワークが復旧した場合に，切り離されていたノードのシステムへの参加が容易に行える．この時に各ノードがデータを保存する位置情報の範囲が再設定され，範囲に会うようにデータが移される．ネットワークが復旧した後のデータの整合性を低コストで保つことが可能である．

ネットワーク的に分断されたノードにデータを格納し，復旧後格納されたデータが期待したノードに格納されているか確認した．図 4 は実験内容を示した図である．ノードは表 2 に示すようにハッシュ値と対応している．ノード 1 のみ動かした状態で，Z 値が 0.249, 0.49,0.5, 0.75 に対応する 4 つのデータを格納する．その後すべてのノードを起動させる．担当すべきデータが期待したノードに格納されていることが確認できた．

4.2 Z 曲線を用いた検索

Z 曲線をインデックスに指定し，検索に利用した．範囲を変えて実行時間を計測する．データは約 200 万件のアメダス気象データを利用している．検索によって約 2 万件のデータを取得する検索を行い，その実行時間を計測する．検索は以下の 2 種類を行う．

(1) ノードの担当範囲内に収まるデータの検索
(2) 複数ノードの担当範囲にまたがるデータの検索

表 3 計測した平均実行時間

検索	平均実行時間 (s)
1	0.9282
2	0.6132

実行時間は 5 回計測した平均を示している．結果を表 3 に示す．検索 1 よりも検索 2 のほうが実行時間は短かった．これは，検索 2 のほうが複数のノードを使って処理を行ったからだと考えられる．座標によるデータ分散方式は，データが均等に分散保存されていないため通常のコンシステントハッシングに比べ検索効率が悪い．広範囲な空間に対する範囲検索であるほど，検索効率が良くなると考えられる．

5. 考察

提案手法のメリットを活かすために，分散保存しているデータのレプリケーションについて考察する．また，提案手法はデータを分散保存することに注目しているため，検索や運用に関してどのような影響があるのか考察する．本稿では DHT として一般的なコンシステントハッシングに着目したが，その他の DHT についても考察する．

5.1 データのレプリケーション

P2P 型の分散環境では，RDBMS のようなトランザクションによるデータの整合性を確保しない．データのレプリケーションによって結果整合性を保証している．結果整合性とは，同時にレプリケーションによってデータのバックアップを提供している．センサデータサーバは，ネットワークから分断された状態での稼働を想定している．このような状況において，ノードがネットワークを構成するどのノードのレプリケーションを持つかによって，提供できるサービスが変化する．

Z 曲線を用いて空間を分割する場合，正方形領域で区切ることが望ましい．そのため，正方形領域で空間を分割した場合に，どのようにレプリケーションを持つべきか考察する．局所的なサービスを提供するためには，隣り合った担当範囲のレプリケーションを保持するべきである．隣り合っていないノードのレプリケーションを保持していた場合，局所的な使用シーンなどを想定すると大部分のデータの実用性がなくなってしまう．図 5 は同じ大きさの正方形で空間を区切った分散環境を表したものである．この場合，あるノードの周りに 8 つのノードが存在する．中心のノードが周囲 8 つのノードのレプリケーションを保持していた場合，システムから分断されてしまった状態で，自分を含め 9 つの担当範囲のデータを参照することができる．ただし，レプリケーションを 8 つ行ってしまうと，1 ノードあたりのデータ数が 9 倍になり，システムの容量を圧迫する可能性がある．

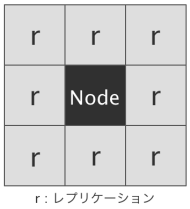

図 5 r = 8

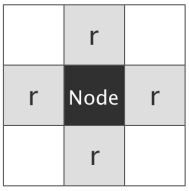

図 6 r = 4 : パターン 1

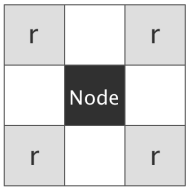

図 7 r = 4 : パターン 2

表 4 各ノードのデータ数

ノード	データ数
1	30 万件
2	100 万件
3	8 万件
4	62 万件

レプリケーションを 4 つ行う場合を考える．レプリケーションを 4 回行う場合は，図 6 と図 7 の 2 種類のパターンが考えられる．この 2 種類のレプリケーションによって参照できる範囲は灰色で塗りつぶした部分になる．どちらの図形が望ましいかは，中心ノードを中心とした円形と重なる面積が多いものとした．使用シーンを考えると，自分を中心とした円形の範囲のデータが参照できることが望ましいと考えたからである．図 6 にくらべ，図 7 は中心ノードの担当範囲と接する部分が少なく，無駄の多い図形となってしまう．図 6 は比較的円に近い形となるため，局所的なデータを参照するためには，図 6 が適している．

5.2 座標によってデータを分散した場合の処理効率

本稿では，コンシステントハッシングと Z 曲線を組み合わせることで空間情報によるデータの分散を実現した．対象範囲を 4 つの同じ大きさの正方形領域で空間を区切った．本来は分散してノードを構成する場合，計算効率が上がるようにデータを均等に保存するのが一般的である．座標による分散を行ってしまうと，ノードごとのデータ数に偏りが生じる可能性が高い．構築したセンサデータサーバの各マシンに保存されているデータ数を表 4 に示す．同じ大きさの正方形領域による分割では，ノード 3 ような無駄な領域が多くできてしまう課題がある．ノード 3 は担当する範囲に海が多く含まれるので，データ数が少ない．分割する数が少ないため，1 つのノードが担当する正方形の面積が大きくなったことが原因として挙げられる．

検索に関しても 4.2 の結果が示すように，検索条件によってはパフォーマンスが変化する．分散環境に対しての検索方法に，MapReduce フレームワークの利用が挙げられる．MapReduce とは，対象とするデータを絞り込む Map 処理とデータを集計，加工して出力データを生成する Reduce 処理の組み合わせによって結果を得る検索方法である．本稿で構築したように座標空間によってデータ分散を行う場合，Map 処理はすべてのノードに送られるが，検索対象になるデータが一部のノードに集中していると，大部分のノードは処理対象とするデータがない状態になる．その結果，分散構築されたノードの一部でしか処理を行うことができない．本来 Map 処理によって分散して行う処理を，一部のノードで処理するため検索効率が下がる．ただし，対象範囲の広い検索を行う場合は複数のノードで Map 処理が行われるため，分散処理のメリットを活かすことができる可能性がある．

保存するデータ量の偏りや検索効率は，細かい範囲をノードに指定することで対応できると考えられる．

5.3 運用中のスケールアウト

センサデータサーバは，システムを停止することなく増え続けるデータに対応する必要があり，スケールアウトが容易に行える必要がある．しかし，スケールアウトは分散環境をデータセンタなどの施設内，または仮想環境に構築することを前提とした戦略である．そのような環境であれば，柔軟にリソースの追加を行うことができ，スケールアウトすることが可能である．しかし本システムは，地理的に分散してノードを構築することを想定している．ノードを追加する場合は，建物の建築，電力やネットワークの供給など物理的なコストが発生する．

また，本稿での提案方式による空間分割では，ノードを追加すると各ノードの担当範囲が変化する．新たにノードを追加する場合は，追加後に各ノードの範囲と新ノードが担当する範囲を割り出した上で，その範囲内にノードを設置する必要がある．5.2 で述べたように，あらかじめ細かく分割した領域をノードに割り当てる仕組みがあれば，ノードを追加する場合の設置場所の問題は対応できる．

5.4 CAN の利用

Z 曲線は空間のなぞり方に特徴があるため，正方形で区切ることが望ましい．運用を考慮すると，担当範囲を正方形に保つのは困難である．本稿では Z 曲線の性質を考えて 4 つのノードで構成したが，運用の中でノードの数を増やすことが想定される．P2P 型の分散データベースはノード

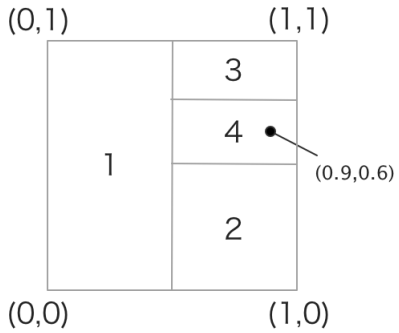

図 8 2 次元トーラスをハッシュ空間とする CAN

が追加された場合，システムが新たなノードを検知し，各ノード数が担当する範囲が均等になるよう調整される．この調整により，正方形による領域分割がくずれてしまう．ノードの担当範囲を柔軟に設定できない課題がある．

DHT のその他の手法として, CAN(Content-Addressable Network) がある [8]．CAN とは，N 次元トーラスのハッシュ空間を利用する DHT である．N 個のハッシュ関数を利用して N 次元トーラスを構成する．図 8 は 2 次元トーラスで構成した CAN の例である．2 次元トーラスであれば，2 種類のハッシュ関数を用いてハッシュ値を 2 つ生成する．図 8 では，(1,0) × (0,1) をハッシュ空間としている．4 つのノードが存在し，それぞれのノードが 2 次元トーラスの範囲を担当する．例えば，キーからハッシュ関数を用いて生成した値が，(0.9, 0.6) だった場合は，ノード 4 に格納される．ハッシュ領域を 2 次元で表現できるため，空間情報との親和性が高い．CAN を採用することで，より柔軟な空間情報によるデータ分散保存を実現できる可能性がある．

6. おわりに

本稿では，センサデータサーバにおける空間情報を利用したデータ分散手法を提案した．提案手法は高可用性を重視しており，ネットワークに障害が発生した場合でも一定の範囲であればサービスを継続して提供することが目的である．ノードに担当座標範囲を設定することで，ネットワーク障害時などでも各ノードが保持しているデータであれば参照することが可能である．本稿では提案手法を Z 曲線とコンシステントハッシングを組み合わせることで実装した．構築したシステムで，期待したノードにデータが格納されていることを確認した．Z 曲線を利用したデータ分散では，空間分割が Z 曲線の特徴を強く受けるため，運用が困難であることが考えられる．今後の課題として，任意の範囲をノードに割り当てるシステムの構築，CAN を用いたシステムの検討が挙げられる．

参考文献

[1] 永井琢也, 満田成紀, 福安直樹, 松延拓生, 鰺坂恒夫：センサデータサーバのための分散データベースの運用・構築に関する考察, 情報処理学会 SIGSE No.23, 2013.
[2] Michael Ferdman, Almutaz Adileh, Onur Kocberber, Stavros Volos, Mohammad Alisafaee, Djordje Jevdjic, Cansu Kaynak, Adrian Daniel Popescu, Anastasia Ailamaki, and Babak Falsafi : A Study of Emerging Scale-out Workloads on Modern Hardware, In 17th International Conference on Architectural Support for Programming Languages and Operating Systems, March 2012.
[3] Hari Balakrishnan, M. Frans Kaashoek, David Karger, Robert Morris, and Ion Stoica : Looking Up Data in P2P Systems, Communications of the ACM, 2003, Vol.46, No.2, pp.43-48.
[4] Eric A. Brewer. Towards robust distributed systems. (Invited Talk) Principles of Distributed Computing, Portland, Oregon, July 2000.
[5] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels : Dynamo: Amazon ’ s Highly Available Key-value Store,
[6] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan : Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, Proceedings of ACM SIGCOMM, 2001.
[7] Y.Shu, B.C.Ooi, K.L.Tan, and A.Zhou: Supporting Mulidimensional Range Queries in Peer-to-Peer Systems, In Processdings of the 5th IEEE International Conference on P2P Computing (P2P ’ 05), 2005, pp.173-180.
[8] Sylvia Ratnasamy, Paul Francis, Mark Handley, and Richard Karp : A Scalable Content-Addressable Network, Proceedings of ACM SIGCOMM, 2001.
[9] 永村美奈, 佐藤翔輔, 柴山明寛, 今村文彦, 岩崎雅宏, 東日本大震災に関する記録・証言などの収集活動の現状と課題, Records Management Society of Japan, No,64, pp.49-66, 2013.