

# P2P 型センサネットワークを用いた 分散環境モニタリングシステムの開発

報告者 大学院総合理工学研究科 エレクトロニクス系工学専攻 教授 井口信和  
共同研究者 大学院総合理工学研究科 エレクトロニクス系工学専攻 教授 向井苑生  
大学院総合理工学研究科 エレクトロニクス系工学専攻 准教授 佐野 到  
工学部 情報学科 講師 溝渕昭二

## 1. はじめに

環境計測や環境モニタリング，農業支援，災害対策などを目的として，屋外でのセンサの活用が様々な分野で進められている．環境計測を目的とした環境計測ロボット FiledSever（以下，FS）の開発や，FS の活用を主な目的とした ALFAE などによる活動も活発に行われている．さらに環境モニタリングのためのセンサネットワークに関する活動も報告されている．

環境モニタリングや農業支援においては，他の観測地域のセンサデータを取得し，比較検討することが重要である．このためには，各地で分散管理されているセンサデータがネットワーク上に公開されているか，または自由に交換できる仕組みが必要となる．現在，計測されたデータは，Web サーバ等で公開する方法が一般的である．しかし，多数のセンサおよび計測データの中から，目的のセンサや計測データを検索し，必要なデータのみを取得することは容易ではない．さらに，ネットワークの運営やセキュリティの問題から，全ての計測サイトが Web サーバ等を運営できるとは限らない

そこで，本研究では，各地で分散管理されているセンサデータの容易な取得を目的として，Hybrid P2P ネットワークによる分散環境モニタリングシステムを開発した．本システムは，メタデータを管理する Index Server を設置した HybridP2P 型システムとして開発することを特徴とする．センサのメタデータ情報およびセンサデータを管理する Node の情報と，コンテンツであるセンサデータを分散して管理する．これにより，1) センサデータの検索速度の向上，2) ピアグループへの参加認証による特定のグループ内での安全なデータ交換，3) Node の管理，4) NAT 越え問題の解決，が実現できる．さらに本システムの Node 機能を動作させるだけで，センサデータの公開および交換が可能となるため，計測サイトでの Web サーバ等の運用は不要となる．

## 2. P2P 技術

P2P (Peer-to-Peer) とは、不特定多数の個人間で直接情報のやり取りを行うインターネットの利用形態である。また、それを実現可能にするアプリケーションソフト自体を指すこともある。具体的には、多数のコンピュータを相互に接続して、ファイルや演算能力などの情報資源を共有するシステムである。P2P には大きく 2 つの実現方式が存在する。それらは、Pure P2P 方式と HybridP2P 方式である。

Pure P2P 方式では、そのサービスを利用するコンピュータはすべてが対等な立場となり、相互の通信状況に応じて、サーバとクライアントの役割を分担しあう。Pure P2P 方式では、コンテンツとコンテンツのメタデータはどちらも各ノードにおいて分散管理される。コンテンツとメタデータを一括管理するサーバに当たるコンピュータは存在しないまた、匿名性の高さも Pure P2P 方式の特徴の一つとなっている。このため、著作権保護などの観点からの問題点は多い。

Hybrid P2P 方式は、従来の C/S 方式と Pure P2P 方式の中間に当たる形式である。Hybrid P2P 方式では、PureP2P 方式と異なり、Index Server 呼ばれるサーバを配置する。Index Server はコンテンツのメタデータのみを管理する。メタデータにはコンテンツの所在 (Node) を示す情報などが含まれている。この情報を利用して、各ノードは、交換したいコンテンツを管理するノードとの間でコンテンツの交換が可能となる。Index Server はメタデータのみを管理し、検索の対象もメタデータだけであるため、従来の C/S 方式と比較してサーバへの負荷は少ない。またコンテンツの交換は、ノード同士が直接実行するため、サーバへの負荷およびサーバ付近のネットワークへの負荷は少ない。

## 3. システムの概要

本システムは、Hybrid P2P ネットワークを用いたセンサデータ共有システムである。図 1 にシステムの構成図を、図 2 と図 3 に各構成の詳細を示す。本システムは、IndexServer (以下、IS) と Node、センサから構成される。IS と、Node により Hybrid P2P ネットワークを構築する。いくつかの Node は設置されているセンサに対して、アクセス可能な状態を想定している。IS はセンサのメタデータと Node の情報のみを管理し、Node がセンサとそのセンサが計測したデータを管理する。Node が管理するセンサデータを取得するには、IS に登録されているメタデータから目的のセンサを検索し、Node 間で直接センサデータを交換する。これにより、IS に登録されている多数のセンサの中から、効率的に目的のセンサを発見することができ、分散管理されている各地のセンサデータを Node 間で共有することができる。同時に、IS によりグループ内の認証や管理も可能になる。本システムは、Java 言語を用いて実装した。JDK6 および JRE6 以上の JVM 環境で動作する。P2P のプラットフォームには JXTA を使用した。

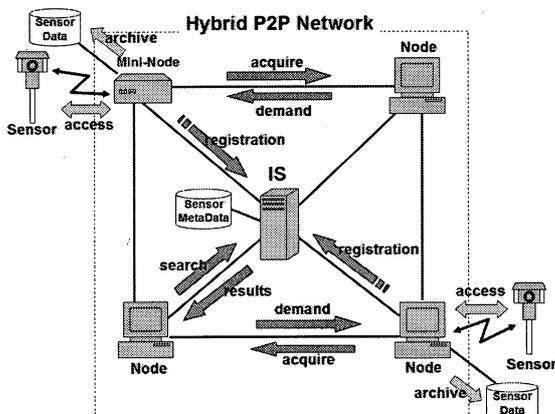


図 1 : システム概要図

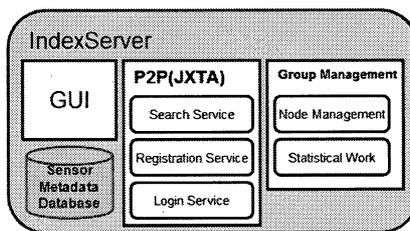


図 2 : Index Server の構成図

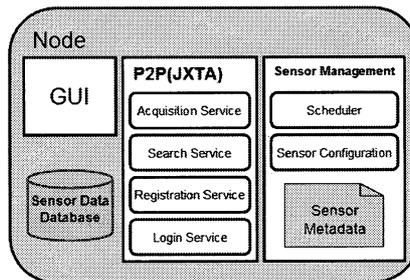


図 3 : Node の構成図

#### 4. システムの詳細

##### 〈4・1〉 Index Server

IS は Hybrid P2P ネットワークにおけるサーバとして、ネットワーク内の情報を管理する。IS の構成を図 2 に示す。IS はシステムに参加している Node を管理する。まず、IS が起動すると、P2P ピアグループを作成する。作成したグループにはパスワードが設定され、Node の参加時に認証を要求する。認証により、特定のメンバー内での共有が可能になる。認証が完了し、グループに参加した Node の情報は、IS のデータベースで管理される。また、IS は Node により登録されるセンサのメタデータを管理する。メタデータは、センサの種類や位置情報などである。表 1 に今回定義したメタデータの一覧を示す。メタデータは、XML 形式で表現される。メタデータも同様に IS のデータベースで管理される。IS のデータベースには、MySQL と SQLite を利用できる。MySQL は高速で信頼性も高く、利用者の多いデータベースである。しかし、サーバ型のデータベースのため、サーバマシンにインストールされている必要がある。SQLite は、アプリケーションに組み込んで利用する軽量データベースである。プログラムに同梱して配布することで、インストール等の作業を必要とせずに利用できる。それぞれのデータベースを、環境に応じて変更できる。

##### 〈4・2〉 Node

Node の構成を図 3 に示す。システムの利用者は、Node の GUI からシステムの機能を利用する。設置されているセンサにアクセス可能な Node は、センサを管理し、センサデータを定期的に収集する。収集したセンサデータは各自のデータベースに保存する。また、Node はデータを収集しているセンサに関する情報をメタデータとして IS にあらかじめ

登録する。各 Node が管理しているセンサデータを取得する場合、IS に登録されているメタデータから、目的のセンサを管理している Node を検索する。その後、Node 同士が直接通信することでセンサデータを取得する。

#### 〈4・3〉 センサ

センサは、様々な場所に設置されていることが想定される。取得できるセンサデータは、温度、湿度、日射量、気圧、風速、雨量などの数値データである。また、本システムでは、web カメラ等もセンサのひとつとして扱い、画像データもセンサデータと同様に取得できる。

#### 〈4・4〉 通信機能

通信機能は、P2P を用いてデータ通信する。Node や IS の通信する処理の内容ごとに、各サービスを定義する。定義した 4 つのサービスについて述べる。

##### 〈4・4・1〉 参加サービス

参加サービスは、Node が P2P グループに参加する際の通信を処理する。参加時の認証も参加サービスで処理する。Node が IS に参加要求メッセージを送信し、認証が成功すれば、IS が Node に参加応答メッセージを返す。Node 終了時は、Node が終了メッセージを送信することで、グループから離脱する。グループ参加後も、Node は定期的に IS に生存確認メッセージを送信する。これにより、Node や IS の予期せぬ終了などを検知できる。

##### 〈4・4・2〉 登録サービス

登録サービスは、Node が IS に管理しているセンサのメタデータを登録する際の通信を処理する。Node は、登録要求メッセージに自身のピア ID とメタデータを記載して送信する。IS は、送信元 Node のピア ID とメタデータを関連付けてデータベースに登録する。また、更新要求メッセージや削除要求メッセージにより、既に登録されているメタデータの登録・削除ができる。

##### 〈4・4・3〉 検索サービス

検索サービスは、Node が IS が管理しているメタデータから、目的のセンサを検索する際の通信を処理する。メタデータの各項目ごとに検索文字列を指定して、検索要求メッセージを送信する。検索方法は、項目ごとに前方一致、後方一致、部分一致、完全一致を指定することができる。検索要求メッセージを受信した IS は、データベースから検索条件に一致するメタデータを検索する。一致するメタデータが存在すれば、そのメタデータと、メタデータを登録した Node の ID を検索応答メッセージとして返す。

#### 〈4・4・4〉取得サービス

取得サービスは、他の Node が管理しているセンサデータを取得する際の通信を処理する。他の Node に対して、取得したいセンサのセンサ ID を取得要求メッセージに記載して送信する。センサ ID はメタデータから取得できる。センサデータを要求された Node は、取得要求メッセージに記載されているセンサ ID が、自身が管理しているセンサ ID と一致すれば、要求元 Node にセンサデータを返す。通信機能には、P2P フレームワークの JXTA(16) を使用した。定義した各サービスを JXTA のプロトコルを用いて実装した。Node が NAT 機器を越えて通信できない場合 JXTA のリレー機能により、IS が Node 間の通信を中継することで、NAT の種類に依存することなくデータを送受信できる。図 4 に JXTA により NAT 機器を越えて通信する様子を示す。Relay-Peer はグローバル IP を持つ、外部からアクセス可能なピアである。Peer A と Peer B は、それぞれ Relay-Peer をあらかじめ知っているものとする。Peer B が Peer A にデータを送信する場合、Peer A は NAT 機器の内側にいるため、外部からアクセスすることができない。この時、JXTA は次のような方法を用いる。Peer A は、Relay-Peer に対して、あらかじめコネクションを作成しておく。Peer B も、Relay-Peer に対してコネクションを作成する。これにより、Peer A と Peer B の間に仮想的なコネクションが作成される。

### 5. センサ管理機能の開発

今回開発した、センサ管理機能について述べる。センサ管理機能は、Node がセンサやセンサが取得・収集したセンサデータを管理する機能である。センサの設定とスケジューラ、さらにセンサデータのデータベースから成る。

#### 〈5・1〉センサの設定

センサの設定では、管理するセンサからデータを取得するためのアクセス方法や、取得するデータの種類などを設定する。また、IS に登録するメタデータもここで設定する。センサ管理機能の GUI の例を図 5 に示す。この GUI からセンサの設定やメタデータを入力できる。センサの設定は、GUI 上のプルダウンメニューから、センサの種類を選択し、設定画面(図 9) から必要な情報を入力する。センサの種類は多様であり、データ取得法だけでも、シリアル接続や USB 接続、ネットワーク接続など様々な方法がある。また、センサによっては、一度に複数のデータを取得できるものもある。そのため、あらかじめ全てのセンサに対応することは困難である。そこで本システムでは、システムで扱う「センサの種類(sensor type)」をプラグイン化することで、対応するセンサの種類を後から追加することを可能とする。現在、標準で用意されているプラグインは、以下のようなものがある。



#### 〈5・2〉スケジューラ

スケジューラは、センサからデータを定期的を取得するためのスケジュールを管理する。データ取得はセンサの設定を用いて取得する。スケジューラの設定画面から、スケジュールを指定する。指定した時刻になると、自動的にデータを取得し、データベースに保存する。自動取得のスケジュールは次の 3 種類から選択できる。

- ・間隔指定
- ・開始時間・間隔指定
- ・時間指定

間隔指定は、自動取得する間隔のみを指定する。スケジュールを設定した瞬間に自動取得が開始し、最初の登録が行われる。その後、指定した間隔ごとに自動取得が実行される。開始時間・間隔指定は、自動取得を開始したい時間と、自動取得する間隔を指定する。指定した開始時間になると、最初の登録が行われ、その後、指定した間隔ごとに登録が行われる。時間指定は、毎日実行したい時間を指定する。毎日指定した時間になると自動取得が実行される。

### 6. 今後の課題

今回開発したセンサ管理機能や、システム全体についての今後の課題について述べる

#### 〈6・1〉センサプラグインの追加

本システムでは、センサ管理機能で扱うセンサの種類は、プラグインによって追加することができる。今回は、標準でネットワーク上から HTTP を用いてデータを取得する 3 種類のセンサを用意した。しかし、現在利用されているセンサの多くは、シリアル接続や USB 接続によりデータを取得するものが主流である。そのため、シリアル接続によりデータ取得が可能なセンサのプラグインの作成が求められる。

#### 〈6・2〉データ表示インタフェースの動的構築

センサデータを取得した際、センサによって計測できるデータの種類や数が異なるため、取得したデータに応じて、動的にデータの表示方法を切り替える必要がある。この問題については、相手 Node からデータを取得する際にインタフェースの情報を取得する法と、メタデータのセンサの種類項目から、受信側が自動的に生成する方法が考えられる。

#### 〈6・3〉Mini-Node の開発

センサはあらゆる場所に設置されていることが想定されるため、設置場所によっては、Node

を屋外に設置する必要がある。この場合一般的な PC の設置は困難である。そこで、小型の組み込み Linux マシンで動作する軽量な Node である Mini-Node の開発が求められる。小型マシンで動作することで、防水対策などが容易になる。HDD などの機械的な駆動部を持たないため、耐久性の面でも有利である。また、センサが動作している組み込み機器の性能次第では、そのセンサ機器単体で Mini-Node を起動することで、P2P ネットワークに参加できるセンサノードとなる。

## 7. まとめ

Hybrid P2P を用いたセンサデータ共有システムにおけるセンサ管理機能を開発した。センサ管理機能は、Node がアクセス可能なセンサを管理する機能である。この機能により、管理するセンサに関するデータの取得方法や、取得するデータの種類を設定できる。さらに、スケジューラにより、指定した時間に定期的にデータを取得できる。

今後の課題として、シリアル接続によるデータ取得が可能なセンサプラグインの追加、取得したデータを表示するインタフェースの動的構築、小型の組み込み Linux マシンで動作する Mini-Node の開発などが挙げられる。