

WFS とローカルデータベースを併用した分散型空間情報データベース構築の試み

嘉山陽一

An attempt of construction for distributed spatial databases, using WFS and local databases

Yoichi KAYAMA

Abstract: WFS(Web feature Service)はインターネットを利用したベクトル空間データを配信するためのプロトコルである。WFS はベクトル地物を XML に変換して通信する。通信とデータのエンコード、デコードの処理にコンピュータの計算リソースが必要なため WFS では高速なデータ利用や大量のデータ利用が困難である。そのような WFS の弱点を補完するため WFS で配信したデータをローカルコンピュータの spatialite database にキャッシュして運用する試みを説明する。

Keywords: WFS (Web feature Service), 分散データベース (distributed database), FOSS4G

1. はじめに

インターネットの発展によりネットワークに接続されたコンピュータ間で通信サービスを利用したコンピュータシステムの利用形態が多様化している。GIS の世界でも様々なネットワークサービスを使ったデータ共有や配信の仕組みが利用されている。

Web Feature Service(以下 WFS)は Open Geospatial Consortium(以下 OGC)で制定されているベクトル地物データの配信と更新のためのプロトコルである。WFS はサーバに格納された個別地物の空間情報・属性情報をクライアントに配信して利用するためには便利なプロトコルである。また WFS では編集のためにサーバ地物をロックする仕組みや、更新のためのトランザクションが制定されている。それらの機能を実装しているサーバとクライアントがあればネットワーク分散型空間情報利用システムを構築できる。

嘉山陽一 〒350-1165 埼玉県川越市南台3丁目1-1

朝日航洋株式会社

Phone: 049-244-4874

E-mail: youichi-kayama@aeroasahi.co.jp

2. WFS 利用時の問題点

2.1 WFS プロトコルの概要

WFS ではクライアントからサーバに対して XML や KVP (Keyword-value pairs) でリクエストを行い、結果を受け取ることを処理の基本とする。WFS2.0 で定義されているオペレーションは 11 個ある (表 1)。

この時メッセージの通信層については HTTP GET, HTTP POST または SOAP over HTTP POST のプロトコルを利用することが可能である (図 1)。

リクエストに対するレスポンスは XML が返される。地物のクエリに対するレスポンスは GML がクライアントに送信される。またサーバ側処理で例外処理が発生した場合はステータスを示す XML が返される。

2.2 WFS 利用時の問題点

WFS は空間データの共有をオープンな仕様で行うことができる便利な仕組みである。サーバ側で格納されている個別地物データをクライアント側でも図形、属性ともに個別地物データとして利用することができる。

しかし WFS は運用の形態によっては問題とな

る事象がある。

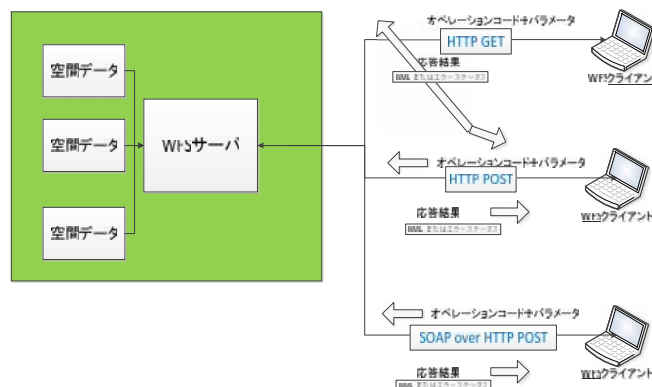
表- 1 WFS オペレーションコマンド

分類	Operation	機能
discovery operation	GetCapabilities	サービスメタデータの取得
discovery operation	DescribeFeatureType	地物タイプの取得
query operation	GetPropertyValues	値の取得
query operation	GetFeature	地物の取得
query & locking operation	GetFeatureWithLock	地物のロック付き取得
locking operation	LockFeature	地物のロック
transaction operation	Transaction	トランザクション発行
stored query operation	CreateStoredQuery	ストアドクエリ作成
stored query operation	DropStoredQuery	ストアドクエリ削除
stored query operation	ListStoredQueries	ストアドクエリリスト
stored query operation	DescribeStoredQueries	ストアドクエリのメタデータ取得

WFS はデータ転送時の処理のオーバーヘッドが大きいので、データの高速通信を行いにくい。WFS のデータ通信は XML や KVP というテキスト形式でデータの送受信を行う。WFS サーバに対してはクライアントからダイナミックにクエリをかけてデータを取得するためデータ取得時に毎回 XML へのエンコーディング作業が行われる。この作業にコンピュータの CPU パワーが大量に必要なためデータ配信のためのコンピュータの CPU コストが高いといえる。またクライアント

側でも受信した XML をデコードする必要があるためデータ受信に時間がかかる。そのためダイナミックに領域別データを通信しながらリアルタイムにサーバのデータを利用するような処理に WFS を利用することは困難である。

図- 1 WFS 通信構成



また WFS クライアントの仕様にもよるが、WFS クライアントでデータデコードの速度をあげるためにデコードしたデータをメモリ上で保持している場合、大量のデータを受信した場合データを保持しきれない場合がある。QuantumGIS をメモリ 4GB の WindowsXP マシンで運用した場合に大量の地物を WFS で受信したところ約 40 万レコードの地物受信が上限であった。

さらにサーバが単一の時にクライアントの数が増大した場合応答が低下するという問題がある。WFS のデータ配信はサーバに多大な負荷をかけるため単一のサーバに同時に多くのクライアントからデータ送信要求があった場合サーバの反応速度は低下する。参照系についてはサーバ多重化で処理分散を行うことができるが、トランザクションを発行してデータ更新を行うオペレーションについては多重化を行うことはむづかしい。

本発表では WFS のこのような問題点をふまえて WFS サーバのデータをクライアントコンピュータ上のローカルデータベースに格納することによって大量の地物を扱うことができるように

する試行について解説する。

図-2 システム構成

3. 分散データベースシステムの設計と構築

3.1 分散データベースの要件

今回サーバで管理するベクトルデータをクライアントコンピュータで共用するシステムのプロトタイプを作成した。このときサーバとクライアントはWANの回線で通信することを想定している。データの更新はサーバで行われ、クライアント側ではサーバでのデータ更新を感知して更新データの取得を行うものとする。この更新データ取得の Protokol として WFS を利用する。クライアント側ではサーバでのデータ更新が行われなければ更新データの取得は行わず、クライアントのローカルデータベースにキャッシュされたデータベースを利用して処理を行うこととする。

また WFS Protokol による更新データダウンロードは時間がかかることが予想される。そのためダウンロード処理が何等かの障害やユーザの操作によって中断された場合にデータダウンロードを再開した時に中断前のダウンロード結果をできるだけ再利用してダウンロードを再開するレジューム機能を実装するものとする。

3.2 システム構成

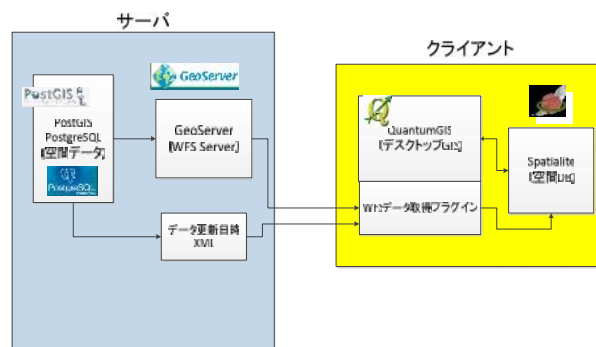
プロトタイプシステムを以下のソフトウェア群を利用して構成した。(図2)

○サーバシステム WFS サーバ GeoServer.
地物データベース PostGIS/PostgreSQL.

○クライアントシステム 地物データベース spatialite. デスクトップ GIS システム QuantumGIS1.7.4 WFS データ更新クライアント QuantumGIS python plugin(自作).

3.3 spatialite の機能

今回は WFS データをダウンロードしてローカルコンピュータで利用するためのローカルデータベースとして spatialite を利用してみた。



spatialite は軽量データベースエンジン SQLite に地理空間情報を扱う機能拡張を行ったオープンソースソフトウェアであり、イタリアの Alessandro Furieri 氏によって作成されたものである..

SQLite は単一ファイルに複数のテーブルを格納して運用する軽量のシングルユーザ向けデータベースシステムである。データベースのファイルはマルチプラットフォームに対応しているため Windows, Macintosh, Linux をはじめ Android や iOS でも利用されている。Spatialite は GEOS ライブラリを利用して SQLite に対して空間情報拡張を行っている。

3.4 WFS 取得機能

本システムでは WFS Protokol でサーバからデータを取得してローカルの Spatialite データベースに格納する機能を QuantumGIS のプラグインプログラムとして実装した。QuantumGIS のプラグインとはメインの実行形式プログラムとは別に Python や C++ で作成したプログラムを特定のディレクトリに格納しておくことで QuantumGIS 本体から起動することが可能なプログラムである。この方式により QuantumGIS 本体を改造することなく QuantumGIS に機能を独自に追加することが可能である。

取得する WFS データについてはレイヤ名、WFS サービス URL、レイヤ、キーカラム名それぞれの最終更新日時を XML ファイルでサーバに格納し、

そのファイルを取得プログラム側がアクセスすることによって更新があった WFS データを特定してダウンロードするものとする。

ダウンロードを行う時に指定されたキーカラムを昇順にソートした形でデータを取得するものとする。データ取得時は 1 回のクエリで取得するレコード数を固定の数（デフォルトは 100 レコード）として、取得したレコードを Spatialite の作業用テーブルに insert,commit が完了してから次のレコードの取得を行うという繰り返しでレコード取得を行う。このことによってダウンロード作業が中断された場合に次に同じダウンロード処理を行う場合 commit が行われたデータの続きから処理を再開することが可能になる（サーバのデータ更新日時が中断された作業よりも新しいものでない場合）。

またダウンロードデータは QuantumGIS の実作業で利用しているテーブルではなく、作業用に作成したテーブルに対して insert,commit を行い、ダウンロード作業が終了した後で実作業用テーブルの名前に変更している。この方式によってダウンロード時に実作業用テーブルがロックされない。さらにダウンロード作業がなんらかの障害やユーザの操作で中断された場合でもデータ運用に影響を与えず、ダウンロード再開時のレジューム機能に利用できる。

4. 評価

4.1 データ転送、ロード時間

Corei7-3520M 2.9GHz メモリ 16GB Windows7 のマシンに前記サーバ、クライアントをインストールした場合の WFS データの単純ロードは以下の時間がかかった。

20 万ポリゴン 50 秒 38

50 万ポリゴン 2 分 03 秒 23

75 万ポリゴン エラー（ロード不能）

また 300 万ポリゴンのデータを Spatialite データベースにロードすると 9 分 51 秒 48 かかった。同じ 300 万レコードポリゴンを QGIS にロードし

た場合かかった時間は以下の通りである。

Spatialite 54 秒 49

PostGIS/PostgreSQL 29 秒 42 （参考）

ShapeFile 46 秒 75 （参考）

WFS から Spatialite のダウンロードとデータベースロードは上記の時間にネットワークの転送時間が加算されたものになる。本システムを利用すると単純な WFS 取得では利用できない 75 万レコード以上の空間データを分散システム上で利用できる。また Spatialite を利用することによりサーバから取得したデータとローカルデータを混在させたデータベース処理が利用できる。

4.2 課題

本プロトタイプシステムは QuantumGIS のプラグインプログラムとして作成した。しかしダウンロード作業は時間がかかるため独立したバックグラウンドプログラムにしたほうが実用には良いと思われる。またクライアントからサーバのデータを更新する仕組みを WFS-T を応用することによって作成できる。サーバからクライアントのデータ更新はレイヤ単位の更新しか現状ではできないので地物単位のデータ更新の仕組みを実装したほうがデータ転送量を減少させることができるとと思われる。

参考文献

OGC : OpenGIS Web Feature Service (WFS) Implementation Specification , <http://www.opengeospatial.org/standards/wfs/>.

Alessandro Furieri :The Gaia-SINS federated projects home-page , <http://www.gaia-gis.it/gaia-sins/>.

Alessandro Furieri (2011) : Viewing Spatialite layers in QGIS- Spatialite Cookbook , <http://www.gaia-gis.it/spatialite-3.0.0-BETA/spatialite-cookbook/html/qgis.html>.

QGIS community:PyQGIS Developer Cookbook , <http://www.qgis.org/pyqgis-cookbook/index.html>.