クライアントサイド描画を用いたWebGISの高速表示手法

山﨑　優† 　井上　潮‡

†‡東京電機大学工学研究科　〒101-8457 東京都千代田区神田錦町 2-2

E-mail: †08gmc24@ms.dendai.ac.jp, ‡inoue@c.dendai.ac.jp

あらまし　Web 技術の進歩により GIS（Geographic Information System）は，WebGIS へと発展した．現在では急速に普及し，より高度な機能を求められている．これらを実現するためには，より自由度の高い地図表現，およびシステム全体の高速化が必要である．本研究室では，クライアントサイド描画モデルを適用した CMap を開発した．CMap によって，自由度の高い地図表現は可能になったが，既存 WebGIS で実装されている差分リクエスト，および先読みリクエストなどの機能は実装されておらず，これらの面で既存 WebGIS と比べてシステム全体の総合的な速度が劣る可能性がある．また，CMap はラスタベースの既存 WebGIS のように画像のタイリングが不可能なため，単純に差分リクエストおよび先読みリクエストは適応できない．本研究では，この二つの機能を実装することで CMap の高速化を図り，高速かつ自由度の高い低通信量型 WebGIS の開発を行う．

キーワード　WebGIS，クライアントサイド描画

Accelerating display of WebGIS with client-side map drawing

Yu YAMAZAKI† 　Ushio INOUE‡

E-mail: †08gmc24@ms.dendai.ac.jp, ‡inoue@c.dendai.ac.jp

Keyword　WebGIS，Client-side Map Drawing

1. はじめに

WebGIS は，サーバークライアントモデルの Web アプリケーションである．地図を表示するクライアントと，地理情報を管理するサーバ，および両者の通信で構成される．

現在普及している一般的な WebGIS では，サーバークライアント間の通信の際，地図画像データを転送している．すなわち，膨大な地図画像データをサーバに保持し，クライアントからのリクエストに応じて，適切な地図画像データをレスポンスし，クライアントサイドで表示している．この方式では，あらかじめ用意された画像を用いた表示しかできず，地図データの一部を利用したり，柔軟なカスタマイズなどはできない．それは，地図表示における自由度が低いと言える．さらに，データサイズが大きいため，通信に時間を要し，サーバに大容量の記憶装置が必要になるという問題もある．そこで，本研究室では通信データ形式をテキスト形式とし，描画をクライアントサイドで行うクライアントサイド描画モデルの CMap を開発した[1].

CMap の開発によって，自由度の向上と，データ通信量の削減は達成された．しかし，CMap の体感速度をあげるためには，従来の WebGIS で行われている差分データ取得や先読みリクエストを行う必要があるが，クライアントサイド描画方式には，単純に適用はできない．その理由は CMap では，リクエスト範囲を矩形としており，街区や建物などを一つのオブジェクトとして扱っている．そして，そのリクエスト範囲のオブジェクトの集合体を一つのテキストデータとして通信している．このため，差分データ取得を行おうとしても，重複して取得してしまうデータ（オブジェクト）が発生する．この差分データ取得ができなければ，通信量が大きくなってしまう先読みリクエストの実装が行えない．本研究の目的は，上記に挙げた問題を解決する事で，CMap の高速化を目指す．

2. 既存 WebGIS

最も代表的な WebGIS である Google マップを筆頭に様々な WebGIS が普及している．本研究室でも，TMAP や CMap が開発された．ここでは，画像ベースである Google マップ[2]と TMAP[3]について特徴と，その問題点述べる．

2.1. Google マップ

Google 社によって無償で提供されている最も広く利用されている WebGIS の一つである．Google マップ[2]では，画像をタイリングすることで差分データ取得およびユーザの操作に合わせた先読みリクエストを行うことで，体感的な速度向上を図っている．

2.2. TMAP

TMAP[3]は，本研究室で開発した WebGIS である．Google マップ同様に，画像のタイリングによる差分データ取得を行っている．Google マップはサーバに画像データを保持しており，クライアントからのリクエストによってあらかじめデータベースに格納された画像データを返す．しかし，TMAP ではクライアントからのリクエストに対し，サーバで動的に地図画像を生成しているため，あらかじめ膨大な画像データを保持する必要がない．

3. 先行研究

既存 WebGIS の問題点は，本研究の先行研究であるクライアントサイド描画を用いた CMap によって解決された．

3.1. CMap

CMap は，Mozilla FireFox ，Google Chrome などの主要 Web ブラウザで動作するクライアントサイド描画モデルを用いた WebGIS である．CMap は既存 WebGIS の二つの問題点を解決し，高度な WebGIS を実現する目的で開発された．CMap によって，既存 WebGIS では不可能だった自由度の高い地図表現と，画像通信により高コストになる通信量をテキストデータで通信することで大幅な通信量の削減が実現された．図 1 に CMap の表示画面を示す．

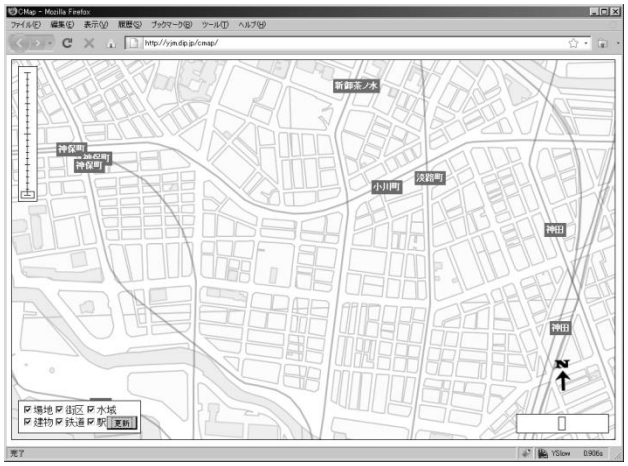

図 1　CMap の表示画面(本学神田キャンパス周辺)

3.2. クライアントサイド描画

CMap の特徴は，通信データ形式を画像では無く非常に簡素なテキストデータを用いて，クライアントサイドで描画を行っている点である．既存 WebGIS とクライアントサイド描画 WebGIS の構造を図 2 に示す．

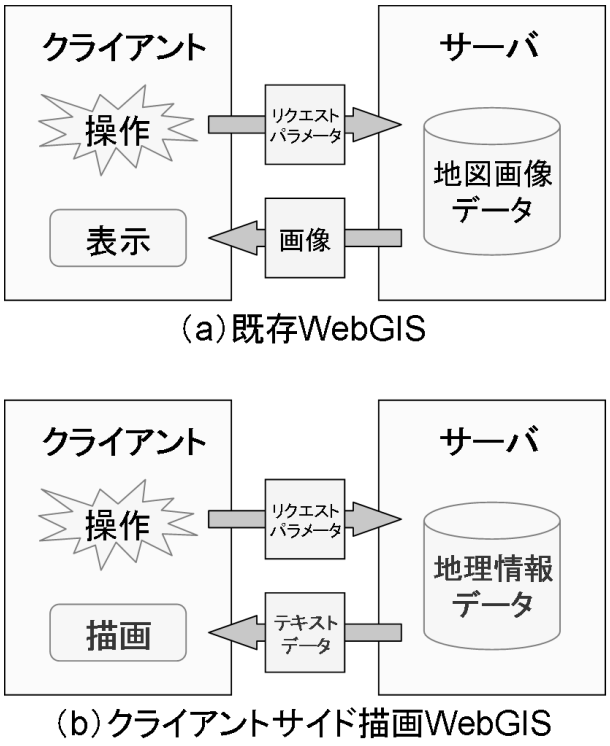

図 2　WebGIS とクライアントサイド描画の構造

3.3. CMap の問題点

テキストデータを用いることによって既存 WebGIS での二つの問題点であるデータ通信量の削減と，自由度の向上は CMap によって実現できた．しかし，CMap にも考慮すべき問題点がある．一つ目の問題点として，差分データ取得が行えない事である．CMap を操作しリクエストを行う時，既に保持している地図データまでもサーバサイドからレスポンスデータとして返ってくる．これにより非常に非効率な通信になる．一方，既存 WebGIS では画像のタイリングによる差分データ取得を可能にしている．二つ目の問題点として，先読みリクエストを行えない事である．既存 WebGIS では，体感速度を上げるため先読みリクエストを行っている．しかし，CMap では既存 WebGIS のように差分データ取得を行うことで同時にスムーズな先読みリクエストも可能にしている．差分データ取得ができない場合，重複したデータを何度もリクエストする事になるので，先読みリクエストを実装したとしても非常に効率が悪い．また，CMap では建物や街区などの地理情報の一つ一つがオブジェクトであり，画像データのようにタイリングする事はできない．

3.4. CMap のデータ通信量

CMap と画像ベースの WebGIS のデータ通信量の比較例を図 3 に示す.

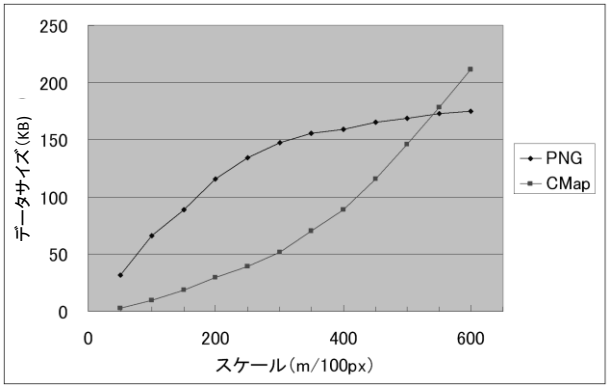

図 3　スケール対データサイズ（神田、640x480）

比較対象は、PNG 形式で保存された画像データであり、東京電機大学周辺の地図である．何メートルを 100px で表すかを示したスケールの数値が 550～600 の間で、データサイズは逆転しているものの、低スケール～中スケールにおいては画像データに比べ減少している．さらに、実用的な範囲はスケール 100～300 程度であり、その間の通信量は大幅に削減されている．

4. 問題解決法

4.1. 差分データ取得

先読みリクエストの効果を最大限に発揮するためには，差分データ取得が不可欠である．しかし，CMap では既存 WebGIS で適用されている画像データのタイリングによる差分データ取得はできない．図 4 に CMap でタイル状に矩形範囲でリクエストを行った場合のイメージ図を示す.

図 4　タイル状に分割した CMap の描画領域

図 4 は，皇居を中心とする地図である．また①～④の番号をタイル別に割り当てた．CMap では，建物や街区など一つ一つがオブジェクトとして扱われている．つまり①～④のどのタイル状の矩形範囲をリクエストしても，全てに皇居のオブジェクトが入ってしまう．仮に①～④の地図データをリクエストした場合，皇居のオブジェクトを 4 回重複して取得することになり非常に非効率である．そこで，本研究では Session[4]を用いた差分データ取得を実装する．本研究で使用している国土地理院数値地図 2500 の地理データにはオブジェクトに対応した ID が付与されている．それを利用し，サーバサイドに SessionID と一度クライアントサイドへ送信した地図データのオブジェクト ID を保存する．2 回目以降の通信の際にはサーバサイドで保持している地図データのオブジェクト ID を持つ地図データ以外の未送信の地図データのみを送信する．Session を用いた初回通信時の差分データ取得の構造を図 5 に示す.

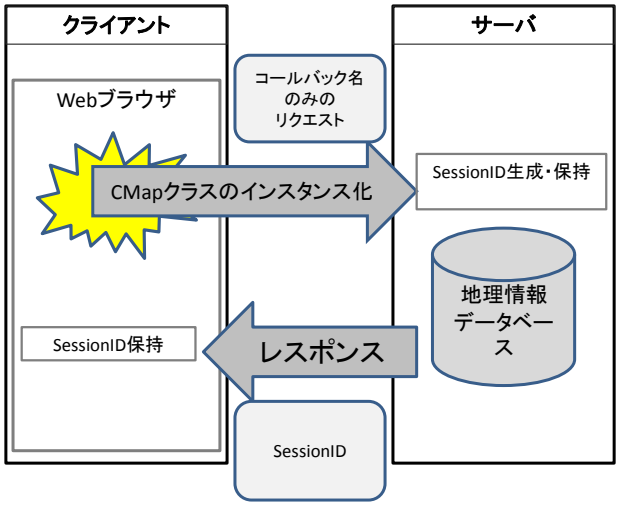

図 5　Session を用いた初回通信時の通信構造

図 5 では，初回 CMap 読み込み時に初期設定されたリクエストパラメータのみをサーバサイドへ送信している．サーバサイドではこの時，SessionID が送信されてきたデータに付与されているかを確認する．SessionID が付与されていなかった場合，SessionID を生成しテキストファイルに保存する．次に，リクエストパラメータに基づいた地図データを検索し，レスポンスデータとして検索した地図データとともに生成した SessionID を付与し，クライアントサイドへ送信する．クライアントサイドでは，取得した地図データと SessionID をキャッシュし描画を行う．図 6 に Session を用いた二回目以降の通信時の差分データ取得の構造を示す.

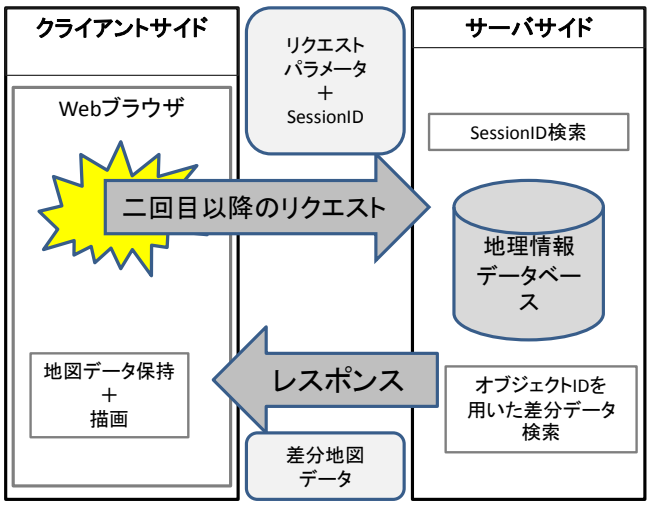

図 6　Session を用いた二回目以降の通信時の差分データ取得構造

4.2. 先読みリクエスト

前節で述べたように，Session を用いた差分データ取得の実装によって効果的に先読みリクエストを行えるようになった．図 7 に，初回 CMap 読み込み時において大きめに地図データをリクエスト・描画した先読みリクエストのイメージ図を示す．

図 7　初回 CMap 読み込み時の先読みリクエスト

図 7 では，地図データを保持しているが Web ブラウザ上に表示していない部分を中央より暗くしている．また実際に Web ブラウザに表示している部分は中央の明るい部分である．図 7 のように内部的に地図データをキャッシュしておくことで，ユーザがドラッグ操作を行っても地図の未表示の部分が減る．以下に，ユーザのドラッグ操作に合わせた先読みリクエストのイメージを示す．

図 8　ユーザのドラッグ操作に合わせた先読みリクエスト

図 8 は，図 7 の状態から矢印方向に地図が表示されるようドラッグしたものである．ドラッグ中に一定区間を超えるドラッグをした時点で差分データと表示された部分をリクエストし，サーバより地図データが返され次第描画を行う．

5. 評価

5.1. 評価目的

二つの提案手法によって，本研究の目的であるクライアントサイド描画を用いた WebGIS の高速化がどの程度達成できたかを確認する．そのために，CMap と差分データ取得のみを行ったもの，先読みリクエストのみを行ったもの，両者を行った新 CMap の 4 つのシステムの速度を実測評価した．

5.2. 評価内容

速度を評価するにあたって，総合速度と通信速度に分別する．総合速度とは，マウスダウン→ドラッグ→マウスアップ→リクエスト→レスポンス→描画間の速度である．通信速度とは，リクエスト→レスポンス間の速度である．総合速度・通信速度を測定する評価用プログラムを作成し，それぞれの速度の数値を Web ブラウザ上に出力する．評価用プログラムは，マウスダウンから描画完了までの一連の処理を一定の条件のもと 4 回行い，その平均値を出力するものである．この試行を 2 回行い，その平均値を測定する．適切な評価を行うために，ドラッグの移動差分，およびドラッグにかかる時間を固定する．手動の動作でいえば，マウスダウン→ドラッグ→マウスアップにかかる時間と移動差分を固定することになる．なお，マウスダウン→ドラッグ→マウスアップまでの総時間をインターバルとする．手動でのマウスダウン，200px 間の移動，マウスアップにかかる時間をそれぞれ測定した．また，

200px に固定した理由として先読みリクエストでは，ｘ軸，ｙ軸方向に 200px ドラッグした時点でリクエストを送信するためである．測定項目は，速めにドラッグする場合と，一般的な速度でドラッグする場合である．表 1 にそれぞれの速度を示す．

表 1　インターバルを構成する項目と所要時間

	速め	一般的
マウスダウン	50(ms)	
マウスアップ	50(ms)	
200ｐｘの移動	100(ms)	200(ms)

また，一回の通信データ量も測定する．測定方法は，FireFox のアドオンである Firebug を利用し，通信を行った際に出力されるデータ通信量を使用する．データ通信量の増減によって，通信速度も大きく増減する可能性があるため，その因果関係を示すためである．評価対象は，従来の CMap，先読みリクエストのみを適用した CMap，差分データ取得のみを適用した CMap，先読みリクエストと差分データ取得の両方を適用した新 CMap である．

5.3. 評価方法

ある地点を中心とし，そこから描画サイズ，スケール，ドラッグによる移動差分に対応したマウスダウン→ドラッグ→マウスアップ間の時間のインターバルを変化させた時の総合速度，通信速度，データ通信量を測定した．

なお，CMap におけるスケールは画面上の 100px に対応する現実のメートル数を示す値であり，同描画領域サイズにおいてはスケールが大きいほど広範囲の地図を表す．表 2 に測定範囲を示す．

表 2　スケール，描画サイズ，ドラッグ方向の測定範囲

スケール(m/100px)	100,300,500
描画サイズ(px)	700x500,1000x700,1400x1000
ドラッグ方向	右右右右,右右左左,右上左下
移動差分(px)	200,400,600,800

適応するインターバルは先読みリクエストの適用の有無によって変化する．先読み切るエストを適用の有無に関わらず，200px までのインターバルは変わらない．しかし，仮に 400px 間のドラッグ移動があったとする．従来の CMap では，マウスアップ時にリクエストを発行するため,インターバルは特に変わらない．先読みリクエストを適用した場合では，200px のドラッグ移動をした時点で，従来の CMap の 400px のドラッグ分のリクエスト範囲を発行するため，400px のドラッグ移動でもインターバルは 200px 分の時間となる．先読みリクエストを適用した CMap と適用していない CMap のインターバルの数値を表 3，表 4 に示す．

表 3　先読みリクエストを適用していない場合

	速め			
移動差分(ms)	200	400	600	800
インターバル(ms)	200	300	400	500

	一般的			
移動差分(ms)	200	400	600	800
インターバル(ms)	300	500	700	900

表 4　先読みリクエストを適用した場合

	速め			
移動差分(ms)	200	400	600	800
インターバル(ms)	200	200	300	400

	一般的			
移動差分(ms)	200	400	600	800
インターバル(ms)	300	300	500	700

また､クライアントとサーバの動作環境を以下に示す。

サーバPC	
OS	Ubuntu Linux 9.10
Webサーバ	Apache2.2.12
DBMS	PostgreSQL8.3.8/PostGIS1.3.5
使用言語	Perl5/JavaScript
地図データ	国土地理院　数値地図2500
CPU	AMD Athlon™ Dual Core Processor 4850e @2.5GHz
RAM	4GB

クライアントPC	
OS	Windows XP Professional
動作ブラウザ	FireFox
CPU	Intel® Core™2 CPU 4300 @1.80GHz
RAM	2GB

図 9　評価の動作環境

5.4. 評価結果

測定によって得た総合速度の評価結果の一部を図 10 に示す．また、図 11 にスケールを変化させた時のデータ通信量を示す．なお，図名の括弧の中は，それぞれ描画サイズ，ドラッグ方向，スケールを表す．

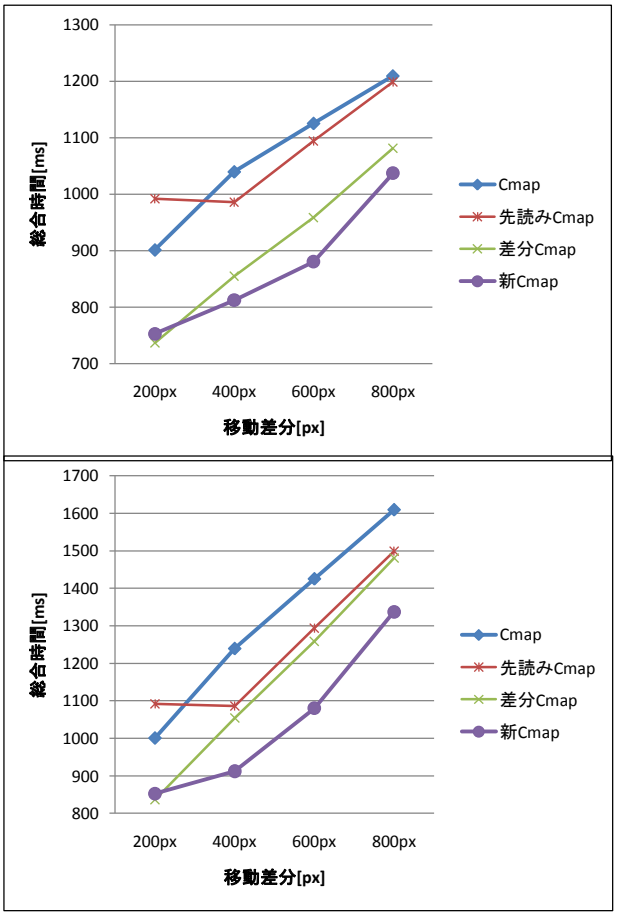

図 10　移動差分対速度(1400 x 1000，右右右右，100)

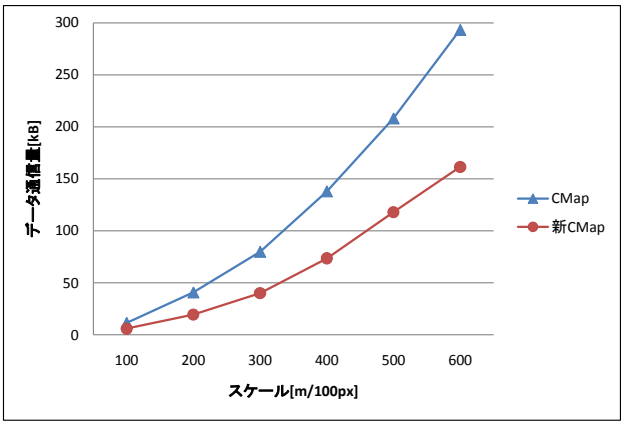

図 11　スケールを変化させた時のデータ通信量

図 10 の時，総合速度においては，差分データ取得と先読みリクエストの併用により，最大 27%の高速化を確認できた．また，通信速度と通信データ量についても同様の効果を確認できた．

図 11 において、スケールの数値が大きくなるとデータ通信量は増加するが，CMap よりも常に低いデータ量になっていることを確認できた．

6. おわりに

本研究では，クライアントサイド描画手法を用いた WebGIS の高速化を行った．具体的には，先行研究である CMap に対し，まず Session を用いた差分データ取得を実装することで通信データ量を減らし，次に効率の良い先読みリクエストを実装した．そして，提案手法の有効性を明らかにするために，評価を行った．その結果，標準セットと描画サイズの変更の条件ではともに 20%前後の高速化が達成された．以上の事から，本研究の有効性を示すことができたが，新たな課題も発見した．今回の評価では，高スケールになると提案手法の効果が著しく落ちる事が判明した．この原因としては主に二つ考えられる．一つ目は，スケールの増減に対して，大量の地図データをクライアントサイドでキャッシュし，描画を行っているためである．高スケールの場合，街区などが凝縮し潰れた状態になる．これを考慮し，高スケールにおいては，簡略化された地図データを表示する必要がある．二つ目の原因として，サーバサイドでの地図データのオブジェクトの検索コストが非常に高い可能性がある．データ量に伴い，検索しなければならないオブジェクト ID も増える．例えば，スケールを 2 倍するとデータ量は約 4 倍した数値になる．この問題に対しても，上記で挙げた高スケール時には簡略化した地図データを通信する事で解消できる可能性がある．

参　考　文　献

[1] 矢島健太郎，山﨑優，井上潮，クライアントサイド描画手法を用いた WebGIS の提案と実装，DEIM，2009，B1-2，March.2009

[2] Google マップ http://maps.google.co.jp/

[3] 田中龍一，井上潮，非同期通信による高インタラクティブ WebGIS フレームワークの研究，GIS 学会第 15 回研究発表大会，No.39，Oct.2006