

HTML5 を用いた 3 次元可視化システムにおける C/S 動的負荷分散手法の提案

丸島 晃明[†] 峰野 博史[‡]

静岡大学情報学部[†] 静岡大学大学院情報学研究科[‡]

1. はじめに

HTML5 等の登場で Web アプリケーションが高度化し、Web ブラウザを介した様々な表現が実現可能となった。この高度な Web 技術はコンピュータにおける様々なアプリケーションのベースとなる可能性があり、新たな Web のスタンダードとして注目を集めている[1]。しかし、多量のデータ処理が必要となるサービスにおいて、多種多様な性能を持つクライアント端末が様々な通信環境下にあることから、サービス提供者がユーザに対して常に許容可能なレスポンスタイムで情報を提供できるとは限らない。そのため、高負荷な Web サービスにおいてユーザにストレスを与えないユーザエクスペリエンスをサービス提供者が常に提供できるような技術の開発が望まれている。

本研究では、空間内の温湿度等を空間補間技術と HTML5 を用いて 3 次元に可視化するシステムを想定し、空間補間演算をクライアントと Web サーバ間で動的負荷分散する手法を提案する。空間補間演算をクライアントと Web サーバ間で双方の処理能力や通信性能を考慮することで、バランスの良い C/S 動的負荷分散を実現する。

2. 関連研究

クライアントを用いてサーバ側の負荷を軽減する技術として、クライアントとクラウドストレージと連携させ負荷分散する研究がある[2]。この研究では、静的コンテンツをクラウドストレージ上に、動的コンテンツを Web サーバクラスタ上に配置し、Web サーバクラスタ内の負荷情報をクライアントと共有することで、ページ生成処理負荷だけでなくデータ転送量が削減しサーバロードバランスを実現している。しかし、動的コンテンツの生成処理の負荷分散は Web サーバクラスタのサーバ台数に依存するため、スケールアウトやスケールダウンを容易にできる仕組みとの併用が望まれる。

本研究では、Web クライアント上で動作するクライアントサイドスクリプトと、Web サーバ上で実行するサーバサイドスクリプトを組み合わせることで、多種多様なクライアント上で動作可能な C/S 動的負荷分散を実現する。またクライアントに対して負荷を分散するため、負荷分散効率がサーバクラスタのサーバ台数に依存すること無く負荷分散を実現可能である。

3. C/S 動的負荷分散システム

3.1 システム概要

図 1 に、本研究で提案する C/S 動的負荷分散システムの概要を示す。C/S 動的負荷分散システムはクライアントマシン上で動作するブラウザと Web サーバ、DBMS (Database Management System) から構成され、ブラウザ上で実行されるクライアントサイドスクリプトとサーバ上で実行されるサーバサイドスクリプトそれぞれの処理量の割合 (以下処理負荷分散比と呼ぶ) を動的に制御する。

C/S 動的負荷分散システムではクライアントとサーバ間での動的な負荷分散比の制御を行うため、同じ入力から同じ結果を出力できるクライアントサイドスクリプトとサーバサイドスクリプトを利用する。スクリプト毎の処理割合を双方の負荷状況や通信品質状況に応じてサーバが動的に変化させることで、同

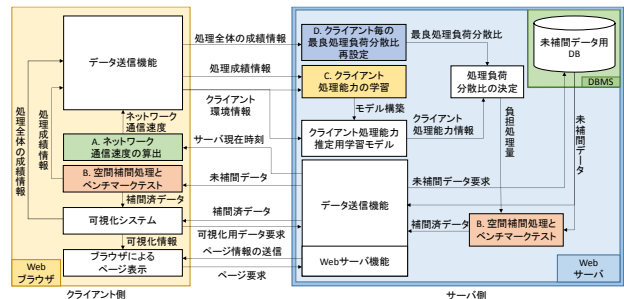


図 1: C/S 動的負荷分散システム概要

じ結果を表示するための処理負荷をクライアントとサーバ間で分散させる。データ生成に要する処理の一部をクライアントに分散させることで、サーバのデータ生成処理負荷だけでなく、クライアントへ送信するデータ転送量も削減できると考える。

3.2 クライアント処理能力の把握

通常のサーバクラスタ内でのサーバロードバランスの場合、各サーバの状況は容易に得ることができるが、C/S 動的負荷分散手法では、多種多様なクライアントとの負荷分散を想定しなければならない。もし、クライアントの処理能力を把握せずに、処理能力の低いクライアントに対して多量の処理を分担させると、クライアントの処理が追いつかず、処理効率を低下させる可能性がある。また、ネットワーク通信速度が遅く、データ演算にかかる時間よりも処理済みデータを転送する時間の方が大きかった場合、データ転送量を減らし、かつデータ演算時間がかかり過ぎないように分散負荷量を調整する必要がある。そのため C/S 動的負荷分散システムでは、クライアントとサーバ間の適切な処理負荷分散比を決定するために、クライアントのネットワーク通信速度と空間補間処理能力を利用する。

クライアントのネットワーク通信速度の算出を行うためには、通信の度にクライアントの現在時刻とサーバの現在時刻をそれぞれ計測し、その差分をとることで上下方向の通信時間を算出する (図 1A)。その後、通信したデータサイズを通信時間で割ることで、クライアントのネットワーク通信速度を算出できる。このネットワーク通信速度の計測は通信がある度に行うため、ネットワーク通信速度の変化に追従できる。

また、クライアントの空間補間処理能力を計測するため、クライアント側での空間補間処理をベンチマークテストとして流用する (図 1B)。空間補間処理を行う前後で時刻を計測して、その差分から空間補間処理に要した時間を算出する。この空間補間処理に要した時間が、クライアントの空間補間処理能力となる。その都度実行される空間補間処理をベンチマークテストとして利用することで、無駄なベンチマークテスト実行の手間を省略し、詳細な空間補間処理能力を把握できる。

ただし、最初にクライアントがサーバに接続する初回時のみ、サーバはクライアントの空間補間処理能力を把握できない。そのため、初回接続時は使用している OS とブラウザ情報とネットワーク通信速度を用いて、通信相手のクライアントの空間補間処理能力の学習と推定を行う (図 1C)。このクライアントの空間補間処理能力の推定には機械学習を用いる。説明変数には、クライアントの使用 OS、ブラウザ情報、ネットワーク通信速度を用い、目的変数にはベンチマークテストで得られた空間補間処理能力を用いる。この学習で得られた学習モデルを用いて、初めて接続するクライアントの空間補間処理能力推定を実施する。

A Proposal to C/S automatic load balancing method for 3D visualization system using HTML5

[†]Koumei Marushima [‡]Hiroshi Mineno

[†]Faculty of Informatics, Shizuoka University

[‡]Graduate School of informatics, Shizuoka University

3.3 処理負荷分散比の決定

3.2 節で示したベンチマークテストと機械学習を用いたクライアント空間補間処理能力の推定でサーバはクライアントの空間補間処理能力を推測することができる。クライアントのネットワーク通信速度や空間補間処理能力、リクエスト送信からデータが揃うまでに要した総応答時間やリソース使用量の情報を用いて、クライアントの環境に応じた処理負荷分散比の決定を、遺伝的アルゴリズム[3]を用いた組合せ最適化で探る(図 1D)。

遺伝的アルゴリズムは、組合せ最適化を解決するヒューリスティック手法の一つである。ある事象に対して、属性群とその属性群に対する解の組合せを個体と定義し、様々な属性群を持つ複数の個体を作成する。複数の個体のうち、最適解に近い個体の特徴を持った新たな個体を生み出していくことで、より最適解に近い個体を探るアルゴリズムである。C/S 動的負荷分散システムでは、処理負荷分散比を属性、レスポンスタイムを解とした個体をクライアントの空間補間処理能力ごとに生み出し、遺伝的アルゴリズムによる最適解の探索を行う。

遺伝的アルゴリズムは、最適解を求める速度は遅いが、最適解に近い準最適解を高速に得られるという特徴を持つ。処理負荷分散比の決定は、最適解を求める必要はなく、その時々準最適解を得て処理負荷分散比を決定できればよいので、遺伝的アルゴリズムを用いることとした。

クライアントとサーバ間の適切な処理負荷分散比は、可視化のための要求に応じて遺伝的アルゴリズムを用いて自動的に決定する。ただし、遺伝的アルゴリズムで用いるパラメータは、経験則によって決まることが多いため、今後の評価実験結果を基に設定する予定である。

4. プロトタイプ実装と評価実験

4.1 3次元可視化システム

本論文で提案する C/S 動的負荷分散システムの実現性を検証するために、研究室内に設置した温湿度センサ(図 2)の計測値を空間補間を用いて 3次元空間に可視化する 3次元可視化システムを開発した(図 3)。3次元可視化システムは温度の高い場所ほど赤に近い色で、低い場所ほど青に近い色で描画し、温度の推移状況を室外から俯瞰した形での描画(図 3(a))や、3次元空間内にカメラを置いたように没入した形で描画(図 3(b))することができる。この 3次元可視化システムに、C/S 動的負荷分散システムの適用を進めている。

可視化に用いる 3次元モデルは WebGL を用いて HTML5 の要素である Canvas に描画している。また、距離で重み付けた平均計算で周囲の値を求める空間補間技術である IDW (Inverse Distance Weighting: 逆距離加重法)を適用し、可視化の解像度を高めている。この空間補間処理を JavaScript と C++ の 2つのプログラミング言語で実装し、サーバ側では C++ のプログラムで空間補間処理を実行した結果をクライアントへ WebSocket[4]を用いた通信で転送する。また、クライアント側では、JavaScript で記述された空間補間処理を実行し、サーバ側から転送されなかったデータの空間補間処理をクライアント側で実施することで、不足データを補完する。以上のようにして、室内の温湿度状況を直感的に 3次元で可視化する。



図 2: 研究室設置の無線センサネットワーク

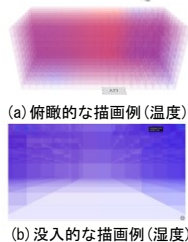


図 3: 3次元可視化システム描画例

表 1: 実験機材一覧

	クライアント		Webサーバ
	デスクトップマシン	iPad Air 2	
OS	Windows 7 Professional SP1	iOS 8.1.1	CentOS 6.4
プロセッサ	COREi5-4570	Apple A8X	Xeon E5645
クロック数	3.2GHz	1.5GHz	2.4GHz
コア数	4	3	6
メインメモリ	8GB	2GB	2GB

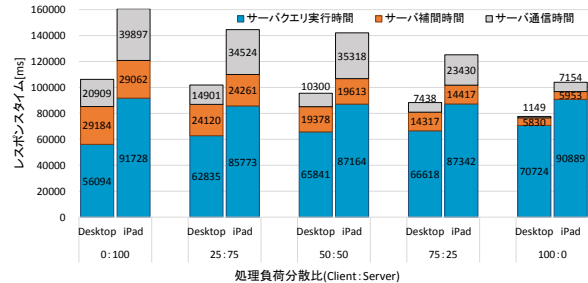


図 4: 基礎評価実験結果

4.2 基礎評価実験

3次元可視化システムを用いて、処理負荷分散比とレスポンスタイムの関係の基礎評価実験を行った。表 1 に使用した実験機材一覧を示す。表 1 のクライアントとサーバを用いて、24 時間分の温湿度のデータ(1,224,720 件)を取得する。本実験では処理負荷分散比決定機能が未実装であるため、クライアントとサーバ間の処理負荷分散比を「100:0」「75:25」「50:50」「25:75」「0:100」と静的に決定した上で各 5 回ずつレスポンスタイムを計測し、平均をとることで結果を比較した。

図 4 に基礎評価実験の結果を示す。DBMS にデータを要求するクエリ実行時間をサーバクエリ実行時間、サーバが空間補間処理を行うために要した時間をサーバ側補間時間、サーバからクライアントへ可視化用データを送信するために要した時間をサーバ通信時間と定義する。図 4 より、クライアントに任せる処理負荷を増やすほどレスポンスタイムが早くなり、クライアントに全ての処理を任せる分散比が最適であることが分かる。これはサーバの処理時間が非常に長いため、サーバ側と並列して行っているクライアント側の補間処理がサーバより先に終了し、クライアント側の補間処理時間が全て無視できる状態となったためである。一方でサーバ内処理にかかる時間の大部分はクエリ実行時間であるため、インメモリデータベースなど高速な処理が可能なデータベースを用いることで C/S 動的負荷分散手法が負荷分散比を動的に決定することによる処理効率の向上が期待できる。

5. おわりに

本論文では、空間内の温湿度等を空間補間技術と HTML5 を用いて 3次元に可視化するシステムを想定し、空間補間演算をクライアントと Webサーバ間で動的負荷分散する手法の提案を行った。今後、現在開発済みの 3次元可視化システムへ、C/S 動的負荷分散手法を適用し、様々な処理能力のクライアントを用いた実験を実施することで、適切な C/S 動的負荷分散システムの運用方法を探る。また、空間補間処理に優先度をつけるなど処理能力の十分でないデバイスでもユーザにストレスを与えないユーザエクスペリエンスを提供できるシステムの研究開発を進めていく。

参考文献

[1] Matthew Hoy, "HTML5: a new standard for the Web,," Medical reference services quarterly, Vol.30, Issue.1, pp.50-55 (2011).
 [2] Sewook Wee, Liu Huan, "Client-side load balancer using cloud," ACM Symposium on Applied Computing (SAC), pp.399-405 (2010).
 [3] Lawrence Davis, "Handbook of genetic algorithms," Van Nostrand Reinhold (1991).
 [4] Ian Fette, Alexey Melnikov, "The websocket protocol," RFC 6455, Internet Engineering Task Force (2011).