

L-016

# 複数の Web クライアントの余剰リソース活用システムの実装と評価 Implementation and Evaluation of System for Utilization of Computational Resources via Web Browsers

中野 雄介†  
Yuusuke Nakano

近藤 悟†  
Satoshi Kondoh

森谷 高明†  
Takaaki Moriya

赤埴 淳一†  
Junichi Akahani

## 1. はじめに

ネットワーク上の複数 PC の計算リソースを用いるグリッドコンピューティングが注目されている[1]。特に、Web ブラウザを介する計算リソースを利用は、より多くの PC の計算リソースを利用できる有効な手法である。しかし、既存研究は PC の負荷を考慮せずにリソースを利用するため、ユーザの Web 閲覧を妨げる可能性がある。本稿では、Web 閲覧者の PC の余剰リソースを利用するためのシステムの実装と評価結果について報告する。

## 2. 提案システム

図 1 で先の課題を解決するシステムを提案する。クライアントは、ワーカプログラムを含む HTML ドキュメントを予め取得している。クライアントのブラウザにこの HTML ドキュメントが表示された時点で、ワーカプログラムがマネージャプログラムと接続し、計算内容を取得可能となる。マネージャプログラムは計算内容が書かれたファイル(計算対象ファイル)内の計算対象データをクライアントの以前の負荷(単位量のデータの計算にかかる時間と、一度に計算して問題ないデータの量等)を基に各クライアントに割り当てる計算対象データの量を算出し、各ワーカプログラムに対して計算内容(分割された計算対象データと計算 Script 等)とそのクライアントの以前の負荷を送信する。ワーカプログラムは取得した計算内容に従って計算を行い、計算結果と自身の負荷(先に受け取った負荷を更新したもの)とをマネージャプログラムに返す。各ワーカプログラムからの計算結果を受け取ったマネージャプログラムはそれらを結合 Script に従って結合し、最終的な計算結果とする。なお、提案システムの詳説は文献[2]に譲る。

## 2. 提案システムの実装

提案システムを評価するために、プロトタイプシステムを実装した。ワーカプログラムを Flash で実装し、ワーカプログラムが計算する計算対象データを XML の専用言語で記述する。また、計算スクリプトと結合スクリプトは ECMA スクリプトで記述され、計算対象データと各種パラメータと共に計算対象ファイルに含まれる。ワーカプログラムを Flash で実装することで、多くの Web ブラウザで、Javascript 等より高速にワーカプログラムを動作できる。また、ECMA スクリプトで各スクリプトを記述できるので、Web 系開発者が容易にスクリプトを記述できる。

一方、マネージャプログラムを Red5 上で動作するアプリケーションとして実装した。Red5 はブラウザ上の Flash プログラムと通信するアプリケーションのためのミドルウ

†日本電信電話株式会社

NTT ネットワークサービスシステム研究所

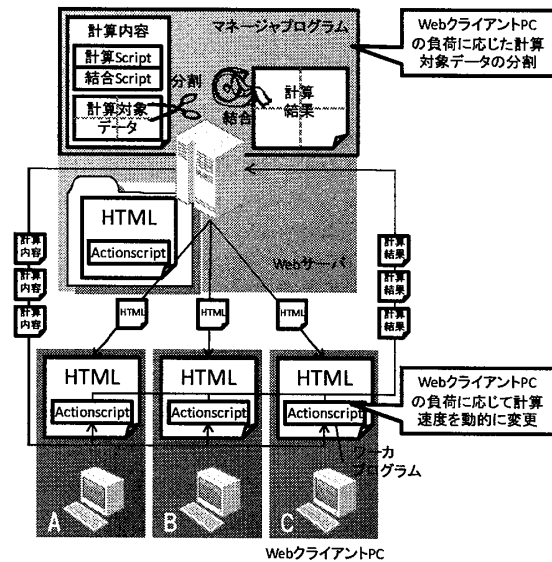


図1 提案システムの概要

ェアであり、これにより、ワーカプログラムに対して計算内容を配信する。

## 3. 提案手法の評価

プロトタイプシステムを用いブラウザユーザの閲覧への影響と、広告推薦のための広告のベクトルとユーザのベクトルとの距離の計算における有効性の評価を行った。

### 3.1 評価環境

表1に評価環境の構成機器を示す。なお、今回は3クライアント上で10個のゲストOSを動作させることで、仮想的に30のクライアントを用意した。各ゲストOSはubuntuとし、それぞれでFirefoxが動作し、ワーカプログラムが含まれるHTMLドキュメントを表示する。このHTMLドキュメントには定期的に正規表現の処理にかかる時間を画面に表示するベンチマークプログラムがJavascriptで組み込まれている。これにより、計算によるJavascriptに対する影響を測定でき、提案手法によるブラウザユーザの閲覧への影響を評価することができる。プロトタイプシステムの評価のために下記の計算を行った。

- 計算内容：512次元ベクトル同士の距離の算出を512回(ベクトルの要素は1~512までの整数)

表1 評価環境

	Webサーバ	クライアント
機種	HP ML110G5	BladeCenter HS21
OS	RedHat Enterprise Linux ES4	CentOS5.2
CPU	Xeon E3110	Xeon E5450
メモリ	4GB	8GB

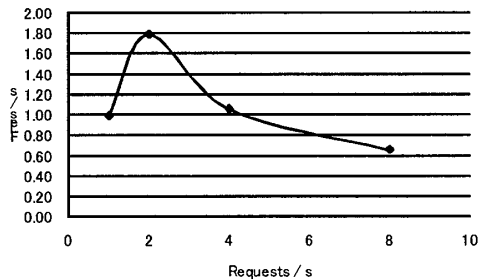


図2 スループット

- リクエスト頻度：1, 2, 4, 8 リクエスト/秒
- リクエスト時間：1 分間

提案手法によるブラウザへの負荷の評価においては、Javascript のベンチマークプログラムが表示する最高値 (最悪値) を用いた。また、リクエスト (マネージャプログラムへの計算対象ファイルの入力) 頻度を 1, 2, 4, 8 リクエスト/秒というように上げたときに単位時間あたりに出力される計算結果ファイル数を提案手法のスループットとし、マネージャプログラムが計算対象ファイルを受け取ってから計算結果を返すまでの時間をレイテンシとした。

### 3.2 評価結果

表 2 は各リクエスト頻度における各負荷のブラウザがいくつ存在するかを表す。なお、ワーカプログラムが動作していない状態での計算時間は 200ms 程度であった。表 2 を参照すると、大部分のブラウザの計算時間は 200ms であり、ブラウザの負荷は抑えられており、ブラウザユーザの閲覧への影響は少ないことがわかる。

図 2 に提案手法のスループットを示す。リクエスト頻度が 2 リクエスト/秒の時にスループットが最大 1.78 ファイル/秒となった。なお、いくつかの計算に失敗したため、2 ファイル/秒とはならなかった。また、4 リクエスト/秒以上の負荷をかけると、スループットは 1 ファイル/秒以下まで下がった。加えて、4 ファイル/秒の計算が正しく完了する割合は 26%、8 ファイル/秒の計算が正しく完了する割合は 8% であり、負荷をかけた場合、かなりのファイルの計算が正しく完了しない(一部のワーカプログラムでの計算時間の上限越え等による)ことがわかった。

レイテンシの計測において、1~2 リクエスト/秒で 1 ファイルの計算にかかる時間は 3~4 秒程度であることがわかった。また、2~3 のワーカプログラムへの計算の分配で処理時間は最小となり、これ以上分散するワーカプログラム数が増大すると計算時間も増大することがわかった。また、4 リクエスト/秒の負荷で 1 ファイルの計算にかかる時間は数十秒程度、8 リクエスト/秒で数分程度であった。このとき、2 ブラウザへの計算の分配で処理時間は最小となることがわかった。

表 2 リクエスト頻度とブラウザ負荷との関係

	リクエスト頻度[リクエスト/秒]			
ブラウザ計算時間の最高値[ms]	1	2	4	8
801~	0	0	0	0
601~800	0	0	2	1
401~600	4	0	14	3
201~400	14	17	14	25
~200	2	13	0	1

### 3.3 考察

以上の評価の結果、30 ブラウザで約 2 ファイル/秒のスループットであることから、15 秒間に 1 回程度のページ遷移をするユーザに対して、広告推薦のためのベクトル間の距離の計算ができる事がわかる。また、2 リクエスト/秒の負荷では計算時間は 3~4 秒程度である。これは、平均滞在時間(ブラウザが 1 つのページを表示し続ける時間)を数分程度とすると、十分な性能であるといえる。つまり、提案システムにより、ブラウザユーザの閲覧に影響を与えることなく、ユーザが閲覧中のページにユーザの嗜好に合う広告を配信することが可能であるといえる。

4 リクエスト/秒以上の負荷をかけた場合、スループットが 1 ファイル/秒以下まで下がってしまった。この理由として以下の 2 つが挙げられる。

- (1) ワーカプログラム上での計算のためのスレッド数が増大すると、ワーカプログラムの計算速度が大幅に下がる。これにより、全体のスループットが下がってしまう。
- (2) ワーカプログラムからの応答時間がしきい値以上長くなると、マネージャプログラムはそのワーカプログラムが異常であると判断し、以降はそのワーカプログラムに計算対象データを分配しないようになる。これを繰り返すと、計算対象データを分配できるワーカプログラムが減少し、スループットが落ちる。

また、4 リクエスト/秒、8 リクエスト/秒の場合、正しく計算できる割合が大幅に落ちた。この課題に関しても、上記 2 つが理由であると考えられる。

レイテンシの計測において、計算対象データを分配するワーカプログラム数が増大することで、レイテンシが上がった。これに関しても 2 つの理由が考えられる。

- (1) マネージャプログラムでの計算対象データ分割、計算結果の結合にかかる時間が増大するため。
- (2) 計算対象データを分配するワーカプログラム数の増大が、一部のワーカプログラムの計算の失敗による、他のワーカプログラムへの計算対象データの再分配が原因であるため。

### 4. おわりに

本稿では Web ブラウザを介して PC の余剰リソースを利用するための手法を提案した。提案手法は Web ブラウザユーザの閲覧に影響を抑え、PC の余剰リソースを利用することが可能である。プロトタイプシステムの評価の結果、本手法は Web ブラウザの負荷を、計算していない時と同等に維持できることを確認した。また、広告の推薦のための各ユーザの嗜好と広告情報とのマッチングを十分な速度で計算できることを確認した。

今後は、スループットとレイテンシの改善のための改良を行う。具体的には、ワーカプログラム上での複数の計算対象データの効率的な計算手法についての研究を行う。また、ユーザが閲覧しているコンテンツによって、ユーザの滞在時間を予測し、その滞在時間を考慮した計算対象データの分配手法についても研究を行う。

#### 参考文献

- [1] S. Choi, et.al, "Characterizing and classifying desktop grid", CCGRID '07, 2007.
- [2] 中野ら, "複数 Web クライアントの余剰リソース活用手法", 2010 信学会総合大会, 2010.