

平成 25 年度 TSUBAME 産業利用トライアルユース 成果報告書

利用課題名 超大規模行動データを用いた広告出稿最適シミュレーション高速化実験

英文 Faster optimal simulation experiment using ultra-large-scale behavioral data.

利用課題責任者
Takeshi Sasaki

所属
Information Service International-Dentsu, Ltd.
URL <http://www.isid.co.jp>

邦文抄録

我々の課題は、膨大なデータが日々蓄積される今日の消費者生活において、広告業務の媒体計画作成に携わる人向けの業務システムを提供しており、顧客が求める期日迄に媒体計画を策定する機能を実現することが困難なケースがあるということである。その内容は、かなり高額な予算の場合や、複数媒体を5媒体や6媒体使う場合においては提案予定期間以上に集計処理時間が掛かることにある。消費者の行動をアルゴリズムで暗黙知データとして扱いそこで、GPGPUを利用した処理の高速化をテストプログラムを作成し、検証することが今回の目的である。

英文抄録

We have a problem that the business system for ad agency cannot make a media plan before their time limit. When the conditions are high level values, that is too much budget, or using 5-6 media. We'll make a test program to solve the problem via GPGPU.

Keywords: Media-Mix Optimized Media Planning

背景と目的

広告キャンペーンにおいて、様々な条件 例えば、「地区」、「媒体種類」、「媒体予算」、「媒体出稿期間」、「最適化指標」などがあり、その条件から媒体への接触ローデータを元に、ビジネスロジックをコンピューターリソースを用いて最適化された媒体計画を算出するアプリケーションを我々は提供している。

媒体計画において、媒体の種類を複数指定されたり、または予算を多く投入されたり、最適化指標目標が高く指定された場合には、その計画案の構成要素の選出と最適化指標を複数媒体でどのように獲得するのかについて、かなり多くのデータを用いて演算を繰り返しながら求めている。その集計処理における処理時間が時に数日に及ぶことがあり、長時間掛かることによるビジネス・チャンスの消失が起きている点を改善したい。本プロジェクトでは、最適化された媒体計画の算出を現行システムにおいて掛かっている時間約50時間を数時間に短縮するため、東京工業大学の所有する、

TSUBAMEのリソースを使って実現した。データを使って媒体計画の構成要素を単体で生成する部分はCPU処理を高速化することで実現し、その各単体要素の組合せた結果の統合された広告効果を高速に処理するためにGPGPUを用いてCUDAプログラミングで実現し、現行システム対比 約9倍の成果を得た。

概要

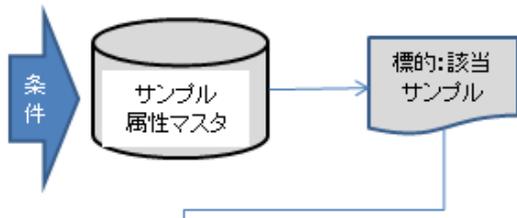
媒体計画立案のための条件を下記のようにした。

- 広告ターゲット＝「4歳以上男性または女性」
- 広告出稿エリア＝「関東エリア」
- 対象媒体＝「TV」、「ラジオ」
- 各媒体は スポット広告も番組に見立てて集計
- 出稿予算＝「1億円」(各媒体 10%刻みで構成)
- 最適化指標＝広告接触回数「3回」
- 最適化目標＝最適化指標到達率「35%」
- 予算許容指数＝1.2(倍)

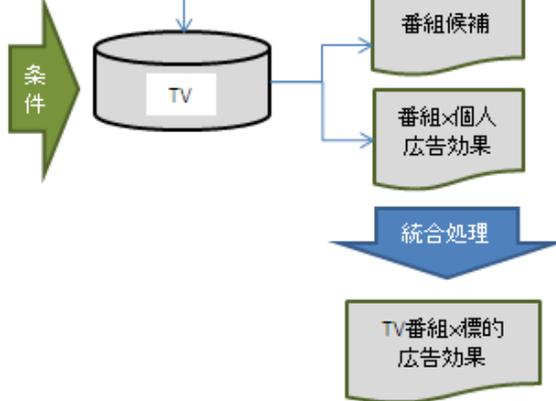
これらの条件を元に、最適化された媒体計画を算出する。

現行の処理概要は下記に処理概要フローを示す。

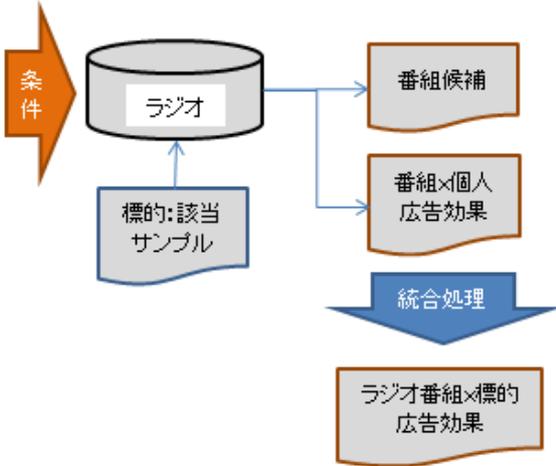
【広告ターゲット(標的)処理】



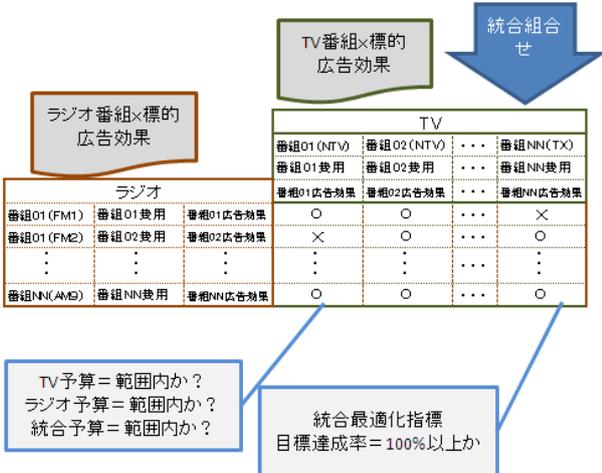
【TV媒体処理】



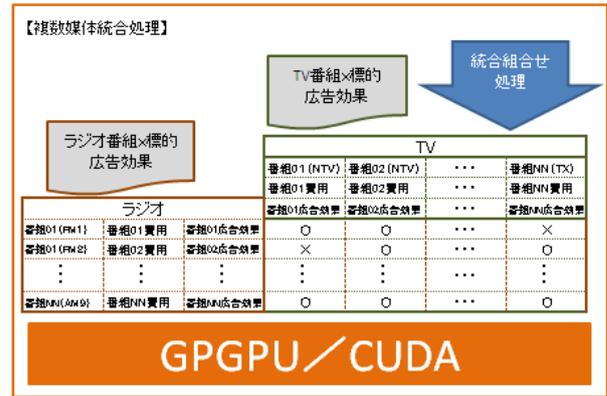
【ラジオ媒体処理】



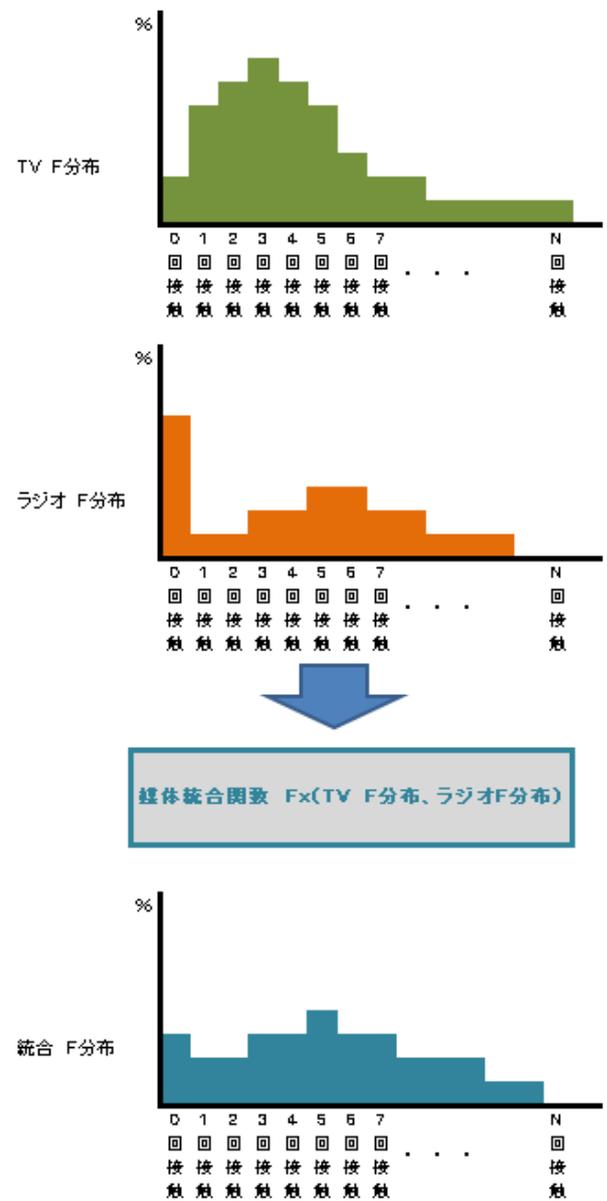
【複数媒体統合処理】



この通常処理の中で、GPGPU による高速化を行う範囲を、複合媒体統合処理に限定した。



GPGPU で処理する主要処理の概念図。



結果および考察

1) 高速化における、DISK ストレージの影響に関する調査。(同一条件での試行回数 5 回の平均値)

■弊社側開発環境での数値検証

No.	サーバの様式	経過時間	
1	StandAloneLinux(LcalDisk)	354 分	(5時間54分)
2	Linux+Strage(NFS)	357 分	(5時間57分)
		差 3 分	

⇒ 全体 354 分における 3 分について。全体処理時間に占める割合は 0.85%程度のため、実際の計測における速度改善には大きく寄与しないことを確認。よって、TSUBAME 上でのネットワーク・ストレージ利用有無を考慮せず計測することとする。

2) MPI による CPU 側処理の並列化の影響調査

(同一条件での試行回数 5 回の平均値)

■弊社側開発環境での数値検証

No.	サーバの様式	経過時間	
1	NormalLinux(NonMPI)	354 分	(5時間54分)
2	DualCPU Linux(MPI)	312 分	(5時間12分)
		差 42 分	

⇒ 全体 354 分における 42 分について。全体処理時間に占める割合は 11.86%であり高速化の有効な手段である点は検証された。

⇒ ただし、後続処理である媒体統合処理に向けて MPI の機能を有効にしたまま、GPGPU で高速化を実現する作業が複雑で時間を要することが判明したため、TSUBAME 環境への実装からは外した。

3) GPGPU による GPU 処理と CPU 処理のハイブリッド型処理との比較(同一条件での試行回数 5 回の平均値)

■CPU のみ、GPGPU ハイブリッド型共に、TSUBAME での数値検証

No.	サーバの様式	経過時間	
1	CPU_Linux(NonGPU)	344 分	(5時間54分)
2	CPU_Linux+CUDA(GPU)	298 分	(4時間58分)
		差 46 分	

⇒ 全体 344 分における 46 分について。全体処理時間に占める割合は 13.37%であり高速化の有効な手段である点は検証された。

⇒ GPGPU 側の更なる高速化余地はありと思われる。

るが、現在のところ CPU 側の残処理の高速化の方が改善余地を多く残っていると思われる。

⇒ CPU 処理と GPU 処理は作りを単純化するため、ファイルをインターフェースとして 実行モジュール1 (CPU のみ) から、実行モジュール2 (GPGPU 有り) を呼び出すバッチ処理で 2 個連続して実行される方式で実現した。

4) GPGPU による GPU 処理と CPU 処理のハイブリッド型処理 と 現行 HP-UX 版との比較(同一条件での試行回数 5 回の平均値)

■HP-UX は HP 社の SuperDome サーバ

DISK は NFS で利用での数値検証

■GPGPU ハイブリッド型 TSUBAME 環境

DISK は ストレージ利用での数値検証

No.	サーバの様式	経過時間	
1	CPU HP-UX(NonGPU)	2996 分	(49時間56分)
2	CPU_Linux+CUDA(GPU)	298 分	(4時間58分)
		差 2698 分	

⇒ 現行システムが稼働する HP-UX の環境は、CPU やメモリも旧世代であり、そもそも処理能力がかなり劣る。

⇒ 全体 2996 分における 2698 分について。全体処理時間に占める割合は 90.05%であり、非常に高速な処理が実現できる点が検証された。

まとめ、今後の課題

■現行システムに対する高速化の目的は、GPGPU を用いた環境で 9 倍高速になる点が確認できた。

■GPU による高速化の余力以上に、CPU に依存している処理部分が残っている。方法としては MPI を利用して CPU 並列でできる部分と GPU 処理を追加して実現する機能を丁寧に修正する必要がある。この部分に手を入れることが、最も高速化が可能な手段と思われる。

■今回は GPU 処理による高速化の効果を検証するため、媒体数を 2 個に限定したが、6 媒体対応して確認を行ないたい。