1M2-5in

マルチエージェントシミュレーションのマルチコア実行環境上でのエージェントコード最適化フレームワークの提案

A Preliminary Performance Optimization Support Framework for Multi Agent Simulations on Multi-core Execution

佐野 義仁[*1] 福田 直樹[*1]
Sano Yoshihito Fukuta Naoki

[*1]静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University

On deploying a good agent simulation, it is important to be able to handle detailed behaviors of agents in a simulation as well as scaling up the simulation to cover important phenomenon that could be produced. There are strong demand to utilize multi-core computing resources effectively for such simulations. However, it is not easy task for developers to set the sufficient compilation and execution parameters and analyze their performance characteristics for various execution settings. In this paper, we present a preliminary framework to assist coding agents, as well as the parameter tuning process of multi-core programming for simulation developers to utilize many parallel cores in their simulation programs efficiently.

1. はじめに

エージェントシミュレーションにおいては，大規模化が 1 つの課題である．例えば，マルチエージェントシミュレーションの大規模化についてのアプローチとして，文献 [山下 12] の手法がある．文献 [山下 12] の研究では，シミュレーションを行う対象環境である 3 次元空間全体を考慮してシミュレーションを行う代わりに，空間内にリンクとノードの概念を導入してエージェントの行動可能範囲を制限することによって，エージェントの計算処理を劇的に縮小させることを可能としている．この方法は、マルチエージェントシミュレーションの詳細度を効果的に単純化することにより大規模なマルチエージェントシミュレーションを実現している．

大規模化の実現と同時に，シミュレーションの詳細度を確保できることも重要な課題の 1 つである．例えば，空港の避難シミュレーションにエージェントシミュレーションを導入した例 [Tsai 11] では，従来ではあまり考慮されてこなかった人間の恐怖といった感情や家族といった人間関係をエージェントシミュレーションに導入した避難シミュレーションを行うことで，それまでのシミュレーションでは正常な避難ができると考えられていた災害シナリオにおいて，逃げることができない人間 (エージェント) が存在する可能性を示す結果が得られている．災害やイベントなどの非日常的な状況に対して，車両や人々がどのように行動をするのか分析したり，あるいは行動主体の微細な動きが，シミュレーション対象全体の動きに大きく影響を与える問題に対して綿密に議論を行いたい場合には，エージェントが動的な状況の変化に適切に反応して行動できるようにプログラムされていることが必要となる．すなわち，エージェントが動的な環境変化に対応できるようにコーディングされ，それを合理的な時間内でシミュレーションにより動作させることができる必要がある．例えば，道路交通の分野では，シミュレーション実行途中での細粒度なエージェントのプランニング処理の効果について検証が進められている [Hoz 11].

シミュレーションの詳細度を高めつつ大規模化を実現するためには，マルチエージェントシミュレーションの処理の効率化が 1 つの課題である [Nakajima 10]．例えば，道路交通に注目した場合，現実の世界では，対象を 1 つの都市に注目して考えたとしても，その 1 つの都市の交通が他の都市における交通流入・流出量に影響される場合もあり，たとえ 1 つの都市をシミュレートするためにも，対象とする都市の規模によっては何百万台の車両が存在する状況を再現できることが必要となる．実際に，電気自動車専用レーンを都市に導入した場合の影響を検討するためのマルチエージェントシミュレーションでは，およそ 300 万のエージェントによるシミュレーションが行われた [Kanamori 11]．しかしながら，非常に大規模でかつ複雑な環境 (空港，混み合った駅あるいは都市全体など) で複雑な動きをするエージェントの振る舞いを，何百万ものエージェントに対して再現したシミュレーションを実行するには，多くの演算処理能力が必要となる．

すでに，文献 [佐野 13a][佐野 13b][佐野 13c][Sano 13a][Sano 13b][Sano 14] で GPU を利用したマルチエージェントシミュレーションの大規模化を著者らは検討している．マルチエージェントシミュレーションは様々な対象に対して適用されており，例えば道路交通シミュレーション [Balmer 08]，空港の避難シミュレーション [Tsai 11]，群衆シミュレーション [山下 12]，などへの適用が行われている．GPU に限定されず，多様な種類のマルチコア計算資源を利用してハイスケーラブルなシミュレーションを実現をするためには，シミュレーションプログラム内でそれらの計算資源の演算コアを効率的に利用できることが重要となる．本論文では，マルチコア計算資源を利用したプログラムをシミュレーション開発者がコーディングする際の，処理の最適化を支援することを可能とするフレームワークについて述べる．具体的には，我々が提案するフレームワークが，様々なシミュレーション実行環境に合わせて，どのように処理の最適化を支援できるかを，最適化シナリオに基づいて示す．

2. 関連研究

マルチエージェントシステムを構成するためのフレームワークとしては，例えば JADE[Tilab] がある．JADE では，提供される機能を利用することによって，複数のエージェントが自律的にそして協調的に動くマルチエージェントシステムを構築することが可能である．JADE を利用することによって，マル

連絡先: 佐野義仁, 静岡大学大学院情報学研究科, 〒 432-8011, 静岡県浜松市中区城北 3-5-1, gs13017@s.inf.shizuoka.ac.jp

チエージェントに従った並列処理環境を構築することが可能であるが，我々がスケーラビリティ改善のために利用する GPU などのマルチコアには，容易に適用できない．

マルチエージェントシミュレーションの処理に GPU を適用している研究としては，例えば，文献 [Richmond 09] がある．この研究では，エージェント処理などに GPU を利用することによって，CPU を処理に利用する場合に比べて，高速に処理できる場合があることを示している．このように GPU を利用することによって，高速に処理をすることができ，スケーラビリティの改善につながる可能性がある．

大規模並列処理コンピュータによる大規模シミュレーションの効率的な処理の実現をするために，エージェントシミュレーションを行うための基礎研究として，文献 [Yamamoto 07] では，スレッドレベルの並列プログラムを効率的に実行することができる IBM Zonal Agent-based Simulation Environment，通称 ZASE と呼ばれるエージェントサーバを開発した．ZASE は，スレッドレベルによる並列実行によって処理の高速化がされた 2 つ以上のエージェントサーバを組み合わせ，そして，大規模並列処理コンピュータ上で大規模なエージェントシミュレーションの実行を可能とするために，エージェントシミュレーション内の処理を効果的に分解してそれぞれに対してエージェントサーバに適切に割り当てることによって大規模化を図っている．このようなアプローチは，SMP に基づいたスカラプロセッサコンピュータクラスタに適用可能だが，一方で，我々が対象にしている民生用コンピュータ上での GPGPU を利用した処理には，その基本アーキテクチャの違いから，容易には適用できない．

3. 提案フレームワーク

エージェントシミュレーションでマルチコア計算資源を利用するための方法として，本研究では，OpenCL を利用する．様々なハードウェアをサポートしている OpenCL を元にコードを記述することによって，本研究が対象としている CPU や GPU といった多種多様なマルチコア計算資源でエージェントの処理などを並列的に行うことが可能である．また，本研究では，汎用コンピュータに搭載されている複数のマルチコア計算資源 (例えば CPU と GPU) を利用することも考える．複数の計算資源を利用することによって，処理の効率を向上することが可能であると考えられる．OpenCL を用いて計算資源を管理をする場合，通常，複数のコマンドキューを用いてデバイスを管理するが，その際，処理が複雑になり，複数のデバイスを扱うことが容易ではないという課題が挙げられる．本研究では，これらの処理を本フレームワーク側で補助することにより，シミュレーション開発者がより容易に複数デバイスを扱えるようにする．

図 1 に我々が提案するフレームワークの構成を示す．文献 [佐野 13c] などでも述べている通り，フレームワークは，Simulation Module, Real-time Rendering Module, Benchmarking Module, User Interface Module，および Parameter Tuning Module の 5 つのモジュールから成る．Simulation Module は，シミュレーション開発者が作成したカーネルプログラムでのエージェント処理などを含んだ簡易的な交通シミュレーションを実現するためのモジュールである．Real-time Rendering Module は，シミュレーション状況をリアルタイムに表示可能とする．Benchmarking Module は，GPU などのマルチコア計算資源を利用してカーネル関数を実行した場合のパフォーマンスとスケーラビリティをテスト，および評価するためのモジュールである．User Interface Module は，フレームワーク上で，様々な関数の実行をコントロールするためのフロントエンドである．そして，Parameter Tuning Module は，開発者が様々なマルチコア計算資源とシミュレーション設定に対するパラメータセットを調節する際の補助機能を実現するモジュールである．

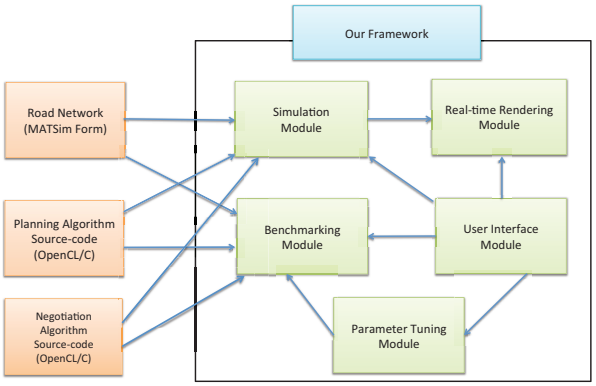

図 1: 提案するフレームワークの構成

本フレームワークを利用する開発者は，C 言語上で OpenCL プログラミングモデルを用いて，プランニングなどのエージェント処理に対するカーネルプログラムの記述を行う．カーネルプログラムを記述した後，開発者は，作成したファイル名，カーネル関数名，引数に関する情報が記述されたファイル名をフレームワークに与えることによって関数をフレームワークに登録する．このような作業によって，開発者はシミュレーション環境上で自身の指定したエージェント処理を OpenCL を用いて実行させることが可能である．

本フレームワークを利用する開発者は，エージェントの処理として，具体的には，エージェントの行動を決定するプランニング処理や他のエージェントとの相互作用を行うための通信処理などをプログラムとして記述することが可能である．開発者は，シミュレーション環境などに対するシミュレーション内のデータに対して，開発者が作成する関数の引数としてアクセスすることができ，それらのデータをもとにエージェント処理の記述を行う．また，開発者は，エージェント処理の中で使用される作業領域などのユーザ定義の引数を定義することが可能である．ユーザ定義の引数は，本フレームワークに変数の型，使用するメモリ量の情報を与えることにより利用可能となる．引数の記述の後，実行プラットフォームによって割り当てられたエージェント ID を取得するための記述と，アルゴリズムなどの実際の処理の主要部分について記述を行う．エージェントが通信を行う仕組みとして，本研究では，エージェント毎にメモリ領域与え，そのメモリ領域に対してデータの読み書きを行うことによって通信を実現する．このメモリ領域に対しても，シミュレーション環境などに対するシミュレーション内のデータと同様に，開発者が作成する関数の引数としてアクセスすることが可能である．

エージェントシミュレーションの開発者が，作成したカーネル関数に対してマルチコア計算機器を効率的に動作させたい場合，調節すべきいくつかのパラメータが存在する [Khronos OpenCL Working Group 12]．本フレームワークでは，開発者がカーネル関数の評価を行う場合，パラレルスレッド数，メモリ配置，コンパイラオプションといったこれらのパ

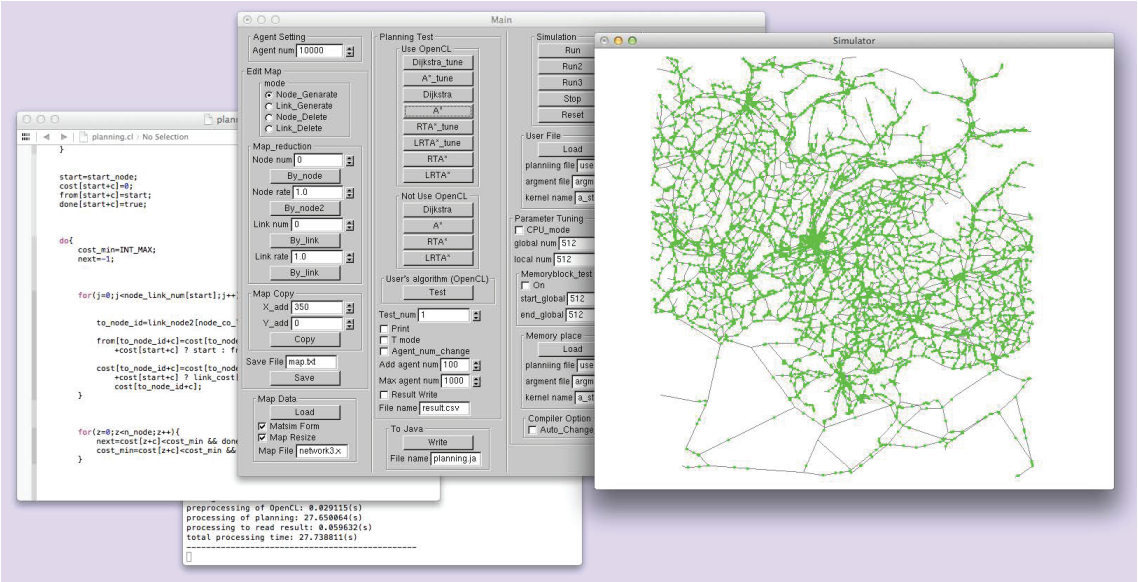

図 2: 実装環境の概観

ラメータを手動で設定をしながらテストを行うことができ，本フレームワークを用いて構成したパラメータでのテスト結果を取得しながら調節が可能である．また，パラメータの組み合わせについてのいくつかの情報をファイルとして開発者が本フレームワークに与えることにより，ファイルで示された組み合わせパターン情報を利用して開発者が利用したいカーネル関数に合わせたパラメータのチューニングを行うことが可能である．具体的には，これらのパラメータ情報に基づいてテストを自動的に行い，そのテスト結果を基により良い性能を発揮することができるパラメータを半自動で選択する．半自動でパラメータを選択する方法として，Budget-Limited Multi-Armed Bandits Model[Tran-Thanh 12] の適用を検討している．

本フレームワークを用いることによって，様々なハードウェア構成やソフトウェア設定の基でエージェントの処理に対してテストを行うことが可能である．しかしながら，考えられるすべての実行環境のために様々なパラメータを手動で評価しセットすることは開発者にとって大きな負担がかかる可能性がある．そこで我々は，本フレームワークをネットワーク上に拡張し，ネットワークにつながれた複数台のコンピュータ上で同時にテストなどの計測を行えるように拡張する．テストシナリオを実行環境に送ることによって，フレームワークの管理コンソールがその情報に基づき実行環境のマネージメントを行う．各々の実行環境は，テストシナリオにしたがって，テストを行いテスト結果を評価していくことによってパラメータチューニングを行っていく．テスト結果は，最後にまとめられ，テストシナリオに沿った最適なパラメータを各々の実行環境ごとに取得することが可能となる．

4. 実装

我々は，3 章で提案したフレームワークに基づいたランタイムプラットフォームの実装を行った．図 2 は，実装したランタイムプラットフォームの概観を表す．ランタイムプラットフォームは，道路交通に対するマルチエージェントシミュレーションを対象にしており，簡易的な交通シミュレーションや，OpenCL に基づいたコーディングが行われたエージェントの処理 (やそのテスト) を行うことが可能である．

我々は，本フレームワークの有効性を示すために，経路探索を行うプランニングアルゴリズムおよびエージェント同士の通信処理を並列実行した際の性能を計測する予備実験を行った．実験環境には，MacBook Pro(OS: OS X 10.8.5, CPU: 2.4 GHz Intel Core 2 Duo, compiler: gcc4.2.1 build 5658, GPU: NVIDIA GeForce 320M, memory: 8GB 1067 MHz DDR3), Mac Pro(OS: OS X 10.8.5, CPU: 3.0Ghz Quad Core Xeon, GPU: GeForce 8800GT, memory: 32GB 800 MHz DDR2) および MacBook Pro(OS: OS X 10.9, CPU: 2.6 GHz Intel Core i7, GPU: GeForce GT 750M, memory: 16GB 1600 MHz DDR3) を利用した．

表 1 に，最適化に関する比較結果を示す．ここでは処理に対するメモリブロックの割り当て (すなわち，OpenCL の中のローカルアイテム数とグローバルアイテム数) の最適化を行った．エージェント数を 10000 として，ノード数 768 の地図データを用いて，A*法に対して OpenCL を利用したマルチコア (CPU) 処理について計測をした．なお，計測上エラーが生じる結果は，除外した．処理時間の最良値，平均値，最悪値の比較を行ったところ，最良値，平均値，最悪値で大きな速度差が見られた．このように，OpenCL のパラメータのチューニングを効率的に行うことによって，同じアルゴリズムを利用する場合でも，高速化が可能となる．

5. おわりに

本論文では，シミュレーション開発者がシミュレーションプログラム内で効率的にマルチコア計算資源を利用するために，プログラミングを行う際のコーディングおよびパラメータチューニングを補助するためのフレームワークについて提案した．我々は，コーディングおよびパラメータチューニングを支援するために，ランタイムプラットフォームを実装した．OpenCL プログラミングモデルに従ってエージェントのプランニング処理や通信処理などを作成すれば，本フレームワークを利用することによってマルチコア計算資源の種類やパラメータの違いによる特性の分析にかかる負担を軽減可能である．

表 1: メモリブロック最適化の例

	Core i7	Quad Core Xeon	Core 2 Duo
A*(最良値)	101.27[msec]	92.25[msec]	411.61[msec]
A*(平均値)	338.26[msec]	1563.32[msec]	4669.16[msec]
A*(最悪値)	1394.25[msec]	9142.18[msec]	14459.04[msec]

今後の課題としては，本研究で作成したフレームワークによって，どの程度スケーラビリティの改善に寄与することが可能であるかを，具体的なシミュレーション問題における改善事例などを通じて評価することがある．また，処理の高速化として，複数のマルチコア演算器を同時に組み合わせて使用するような場面への適用を容易にすることも考えられる．このような場面では，あらゆる実行環境の組み合わせを実環境として準備することは容易ではないため，ある程度の典型的な構成の機材とそれらから事前に収集した実行特性を加味して，効果的な環境を実現する機材の組み合わせを，それらを準備することなく事前に検証できるようなフレームワークの実現も必要になると考えられる．

参考文献

[Balmer 08] Balmer, M., Meister, K., Rieser, M., Nagel, K., and Axhausen, K.: Agent-based simulation of travel demand: Structure and computational performance of MATSim-T, in *2nd TRB Conference on Innovations in Travel Modeling* (2008)

[Hoz 11] Hoz, de la E., Marsa-Maestre, I., Lopez-Carmona, M. A., and Perez, P.: Extending MATSim to allow the simulation of route coordination mechanisms, in *Proc. The 1st International Workshop on Multi-Agent Smart Computing(MASmart 2011)*, pp. 1–15 (2011)

[Kanamori 11] Kanamori, R., Morikawa, T., and Ito, T.: Evaluation of special lanes as incentive policies for promoting electric vehicles, in *Proc. The 1st International Workshop on Multi-Agent Smart Computing(MASmart 2011)*, pp. 45–56 (2011)

[Khronos OpenCL Working Group 12] Khronos OpenCL Working Group, : *The OpenCL Specification Version: 1.2 Revision: 19* (2012)

[Nakajima 10] Nakajima, Y., Yamane, S., and Hattori, H.: Multi-model Based Simulation Platform for Urban Traffic Simulation, in *13th International Conference on Principles and Practice of Multi-Agent Systems(PRIMA 2010)*, pp. 228–241 (2010)

[Richmond 09] Richmond, P., Coakley, S., and Romano, D. M.: A high performance agent based modelling framework on graphics card hardware with CUDA., in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pp. 1125–1126 (2009)

[Sano 13a] Sano, Y. and Fukuta, N.: A GPU-based Framework for Large-scale Multi-Agent Traffic Simulations, in *Proc. 2nd IIAI International Conference on Advanced Applied Informatics (IIAI AAI2013)* (2013)

[Sano 13b] Sano, Y. and Fukuta, N.: A GPU-based Programming Framework for Highlyscalable Multi-Agent Traffic Simulations, in *Joint Agent Workshop and Symposium(JAWS)* (2013)

[Sano 14] Sano, Y., Yoshiaki, K., and Fukuta, N.: A Performance Optimization Support Framework for GPU-based Traffic Simulations with Negotiating Agents, *in Proc. of 7th Int. Workshop on Agent-based Complex Automated Negotiations (ACAN2014)* (2014), (to appear)

[Tilab] Tilab, : Java Agent Development Framework, http://jade.tilab.com

[Tran-Thanh 12] Tran-Thanh, L., Chapman, A. C., Rogers, A., and Jennings, N. R.: Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits, in *AAAI* (2012)

[Tsai 11] Tsai, J., Fridman, N., Bowring, E., Brown, M., Epstein, S., Kaminka, G., Marsella, S., Ogden, A., Rika, I., Sheel, A., Taylor, M. E., Wang, X., Zilka, A., and Tambe, M.: ESCAPES - Evacuation Simulation with Children, Authorities, Parents,Emotions, and Social comparison, in *Proc. International Conference on Autonomous Agents and Multiagent Systems(AAMAS 2011)*, pp. 457–464 (2011)

[Yamamoto 07] Yamamoto, G., Tai, H., and Mizuta, H.: A platform for massive agent-based simulation and its evaluation, in *Proc. International Conference on Autonomous Agents and Multiagent Systems(AAMAS 2007)*, pp. 900–902 (2007)

[佐野 13a] 佐野 義仁, 福田 直樹：マルチエージェントシミュレーションにおける動的経路探索の高速化手法の検討, 第 75 回情報処理学会全国大会 (2013)

[佐野 13b] 佐野 義仁, 福田 直樹：災害などの大規模マルチエージェントシミュレーション実現に向けたスケー ラビリティ改善のためのフレームワークの試作, 第 27 回人工知能学会全国大会講演論文集 (2013)

[佐野 13c] 佐野 義仁, 福田 直樹：大規模マルチエージェント交通シミュレーション実現にむけたパラメータ最適化についての検討, 第 172 回 情報処理学会 知能システム研究会 (2013)

[山下 12] 山下倫央, 岡田崇, 野田五十樹：大規模群集流動の制御に向けたシミュレーション環境の構築, in *Joint Agent Workshop and Symposium(JAWS)* (2012)