

マルチエージェントシミュレーション省電力化のための マルチコアを使用したクラウド制御スクリプト言語の試作

[†] 寛史野房[†] 静岡大学情報学部情報社会学科

福田直樹[†]

[†] 静岡大学大学院情報学研究科

1 はじめに

マルチエージェントシミュレーションの課題の1つに、シミュレーションの詳細化とそれに伴う大規模化がある。シミュレーションの規模に応じた、CPU と GPGPU の計算時間の差を示したのものには、避難シミュレーションにおける事例 [1] があり、GPGPU を用いたシミュレーションではシミュレーションの実行の高速化が実現されている。シミュレーションの詳細化についての検討には [2] があり、感情や人間関係を扱うことでより現実 に則したシミュレーションを行うことで新たな発見がされているが、詳細なシミュレーションを行うことは計算の規模が大きくなることにつながる。

シミュレーションを詳細に行うことで計算が大規模化する場合、その計算時間を短くする方法の1つとして GPGPU といったマルチコア計算資源を用いることがある。CPU・ストレージ・ネットワークの利用を前提とした一般的なクラウドサービスではハードウェアの仮想化を行い、その上でサービスが提供されているのに対し、現状での GPU の仮想化には NVIDIA の GRID[3] など専用のハードウェアによる方法が一般的であり、ハイパーバイザによって利用に制限があるなどの課題 [4] から、GPU の使用を前提として場合には、計算資源の割り当てをベアメタルサーバと呼ばれる実機単位で行う場合や、手元の実験機材との併用を行う場合がある。

本研究では、このような条件において、計算資源の有効活用・省電力化を念頭に置き特に、GPGPUなどのマルチコア計算資源を利用したマルチエージェントシミュレーションを対象としたクラウド環境制御のためのスクリプト言語の試作について述べる。

シミュレーションのタスクを実行するための複数の計算ノードがあることを仮定する。実行するシミュレーションのパラメータやエージェントの行動などによる違いを検証するために、シミュレーションを複数回行うことが考えられる。制御スクリプト言語は、これらの大量のシミュレーションの実行を複数の実行環境に割り当て、同時並列に実行させるための処理を記述する。同時実行のための記述として、シミュレーションを行う際のパラメータの最適化のためのテストパターンの実行などがある。テストパターンの実行時には各実行環境の実行速度や消費電力を推定するための情報が収集される。これらを考慮して計算ノードに対しシミュレーションを割り当てることで、消費電力の削減など計算処理の効率化を図ることができるようにしたい。各シミュレーションの終了後は、シミュレーションの結果をログファイルとして利用者がダウンロードする。ダウンロードされるファイルも、制御スクリプトに記述される。図1に本シナリオにおけるスクリプト言語のフロントエンド部分を含めたシステムの構成を示す。

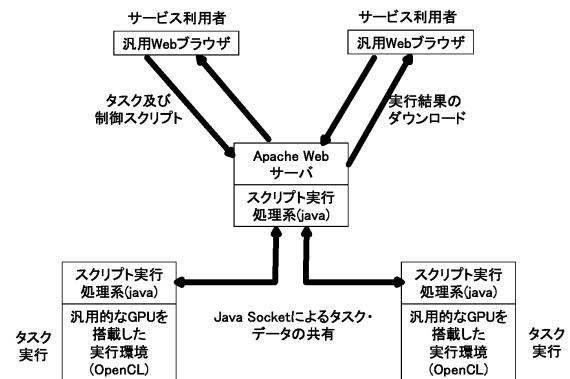


図 1: システムの構成

2 省電力化の手法と提案スクリプト言語

提案するスクリプト言語を利用するシナリオとして、次の例を示す。GPGPUを用いたマルチエージェントシミュレーション環境があり、1台の制御マシンと、シ

3 制御スクリプト言語

制御スクリプトの記述例として、同時に2つの実験のためのマルチエージェント交通シミュレーションを実行するための割り当て制御を示す。

```
Jikken1.setting(agent_num1,agent_itil,  
agent_negocode1,agent_ugoki1);  
Jikken1.DeadLine = currenttime + 3h;  
Set agent_num1 = 10000;
```

A Prototype Scripting Language for Energy-Efficient Multi-Agent Simulations with Multi-Core Processors on a Cloud

†Fumihiro Fusano, Department of Information Arts, Shizuoka University

‡Fukuta Naoki, Department of Computer Science, Faculty of Infomatics, Shizuoka University

```

Set agent_itil = %agent_num%.ini;

Jikken1.testrun(100,fuwabaru,
Jikken1%agent_num%.txt,
(agent_num(1000,10000,1000),
agent_negocode(randombase.cpp,
yamanobori.cpp)));

Jikken1.run(10000,fuwabaru,
Jikken1%agent_num1%.txt,
(agent_num1(1000,10000,1000),
agent_negocode1(randombase.cpp,
yamanobori.cpp)));

Jikken2.setting(agent_num2,agent_iti2,
agent_negocode2,agent_ugoki2);
Set agent_num2 = 10000;
Set agent_iti2 = %agent_num%.ini;

Jikken2.run(1000,fuwabaru,
Jikken2%agent_num2%.txt,
(agent_num2(100,10000,100),
agent_negocode2(randombase.cpp,
yamanobori.cpp)));

autoassign(Jikken1,Jikken2);

```

この例では、シミュレーションの実行に用いる機材の定義の記述については、紙面の都合で割愛している。ここで実行しようとするシミュレーションは、「fuwabaru」というシミュレーションに基づく2つの実験であり、Jikken1とJikken2の実行を制御する場合を扱う。Jikken1とJikken2ではそれぞれ別の実行時パラメータ列の指定を用意しており、Jikken1.settingとJikken2.setting文によってそこで用いる変数としてそれらを定義している。

Jikken1の実行には終了までの制限時間を設けており、現在の時刻から3時間後としている。その次の文では、この実行に用いるエージェント数の初期値として、10000を設定している。

各エージェントの初期位置情報はシミュレーションに用いるエージェントの数に応じて変化するため、エージェントの位置情報は外部ファイルとして定義され、ファイル名は変数「agent_num」の値に応じて変更される。

Jikken1の実行が定義される前に、Jikken1.testrunとしてJikken1の変数を使用して動作テストが行われる。ここではシミュレータが使用するパラメータの最適化を行い、シミュレーションの実行環境への割り当てのための情報を得る。それと同時に実行環境の負荷、および消費電力の測定が行われる。

Jikken1の動作テストが行われた後に、Jikken1.runおよびJikken2.runとして、実際のシミュレーションの動作が定義されている。あらかじめ用意された変数を元に、実行ファイルのオプションを設定している。Jikken1ではシミュレーションが10000ステップ行われ、その実行結果がテキストファイルとして保存される。シミュレーションはエージェントの数が1000から10000まで1000刻みに変化し行われ、実行結果として保存される。

テキストファイルの名前には、そのシミュレーションで使用されたエージェントの数が挿入される。

シミュレーションの実行が定義された段階では、シミュレーションそのものは実行しない。autoassign文にて適切な割り当てがなされ、その時点で実行環境においてシミュレーションが開始される。Jikken1の終了は実行開始から3時間後と設定しているため、それに応じた割り当てが極力可能となるように機材の割り当てが行われる、ということの意味する。

4 まとめ

本研究ではマルチエージェントシミュレーションを省電力に行うことを念頭に置き、GPGPUが利用可能なクラウド環境を制御するためのスクリプト言語の試作について述べた。スクリプト言語の機能として、シミュレーションの負荷を考慮し、シミュレーションを実行環境へ割り当てる手法を検討した。シミュレーションの負荷を測定する手法として、本研究では対象となるシミュレーションを短時間実行するものを挙げた。しかし短時間な実行では、シミュレーション全体の負荷を正確に測定することは困難な場合もある。より高い効率性で実行環境への割り当てを低いオーバーヘッドで行うためには、正確な負荷の計測とそれらによって得られる効率性の向上を勘案しながら、動的な計測方法の最適化を行う機構の実現が課題である。

参考文献

- [1] 安福健祐. GPUを利用した大規模避難シミュレーションシステムの開発. 学術講演梗概集. 一般社団法人日本建築学会, 2011.
- [2] J. Tsai, N. Friedman, E. Bowring, M. Brown, S. Epstein, G. Kaminka, S. Marsella, A. Ogden, I. Rika, A. Sheel, M. Taylor, X. Wang, A. Zilka, and M. Tambe. ESCAPES - Evacuation Simulation with Children, Authorities, Parents, Emotions, and Social comparison. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pp. 457–464, 2011.
- [3] Shared Virtual GPU (vGPU) Technology — NVIDIA. <http://www.nvidia.ca/object/virtual-gpus.html>.
- [4] State of GPU Virtualization for CUDA Applications 2014 — Acceleware Ltd. <http://acceleware.com/blog/state-gpu-virtualization-cuda-applications-2014>.