

並列分散型シミュレーションエンジンにおける マルチスレッド化性能評価

渡部 修介[†] 尾崎 敦夫[†]

三菱電機 (株) 情報技術総合研究所[†]

1. はじめに

同種・異種システム間の連携によって実現される、複雑で大規模な構成のシステムを SoS (System of Systems) [1] と呼び、例えば防災や防衛の分野において近年要求が高まりつつあるような、監視・観測システム、通信システム、情報処理システム等が連携した大規模システムのことを指す。SoS では、単独であっても複雑な振る舞いを持つシステム群が有機的に連携し、全体として多様なミッションを遂行することになるため、その企画・開発段階や、試験・評価段階において M&S (Modeling and Simulation) 技術の活用が必須である。そこで我々は、このような複雑かつ大規模なシステムをモデルとしたシミュレーションを高速かつ高精度に実行でき、また開発コストの低減のためシミュレーションプログラムの再利用性向上に寄与するような汎用の開発・実行環境であるシミュレーションエンジン (SE) の検討と試作を行っている [2]。

2. SE の概要

図 1 に、SE の基本概念を示す。SE は、シミュレーションに登場するオブジェクト (模擬対象となる物体) や発生するイベントを処理するシミュレーションプログラムを論理ユニット (LU) として管理し、複数の LU 間の通信制御と論理時刻進行制御を行うミドルウェアである。1 つの SE によって管理される LU 群と SE は 1 つのプロセスとして実行され、そのプロセス中において個々の LU がスレッドとして並列処理される。なお、多数のオブジェクトが登場する大規模シミュレーションを分散システムとして構築・実行する場合や、市販のシミュレータやツール等との連携のために、SE は、分散シミュレーションシステムの標準規約である HLA [3] に準拠したインタフェースを備えている。

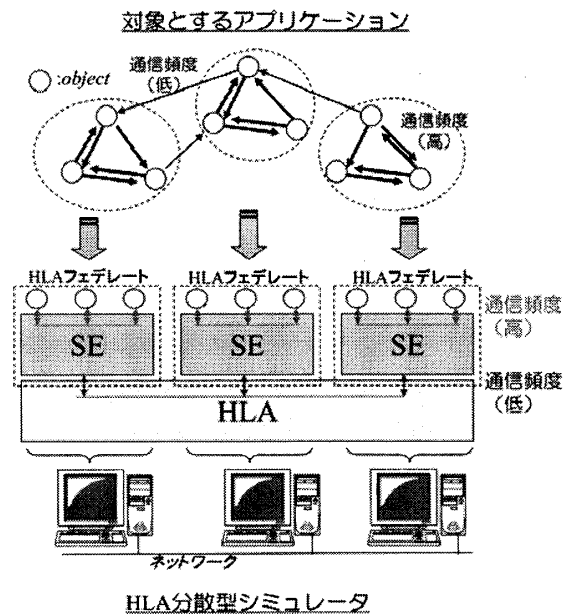


図 1 SE の基本概念

3. LU のマルチスレッド処理

SE によって管理される複数の LU は、他の LU からメッセージを受信しない限りにおいて並行して処理を行うことが可能であり、他の LU からのメッセージを受信する、あるいは受信する可能性のある論理時刻においてバリア同期を行う。

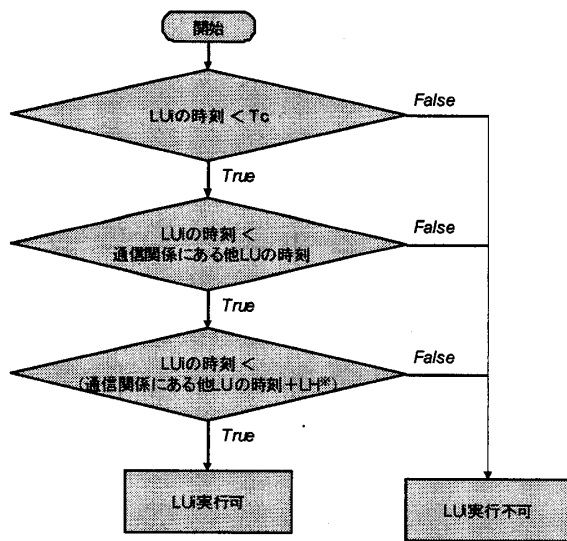
SE は、メッセージ通信を行う LU の組合せを Publish/Subscribe モデルによって管理しており、また、論理時刻進行を、LU の模擬周期 (ΔT) と受信したメッセージ有効時刻 (Time Stamp) の組合せにより管理している。SE は、これらの通信と論理時刻に関する管理情報を用いて LU の並列処理スケジューリングを行う [4]。図 2 に、ある時刻 (T_c) において同時に並列処理する LU を決定する基本アルゴリズムを示す。

4. 性能計測結果と評価

SE による LU の並列処理効率を評価するために、SE が管理する 1 つのプロセス中に 8 個の LU を実行し、同時に実行するスレッド数を 1 から 4 に

Performance Evaluation of Multi-thread Mechanism in Parallel and Distributed Simulation Engine

[†]Information Technology R&D Center, Mitsubishi Electric Corp.



*Lookahead: そのLUがメッセージを送信しないことを保証する時間間隔

図 2 並列処理 LUI 決定アルゴリズム

回の性能評価結果を基に、SE 内部の逐次処理部分の高速化と、個々の LU の処理負荷を考慮した並列処理の手法についての検討と評価を実施する予定である。

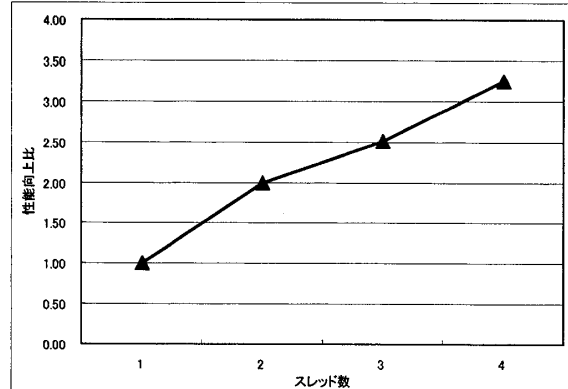


図 3 ケース 1 計測結果

変化させた場合の性能向上比を評価した。また、基礎的な性能の確認と、実利用時に近い条件での確認を行うために、次の 2 ケースについて計測した。なお、計測環境には一般的な PC (OS: WindowsXP, CPU: Intel Core2 Quad) を用いた。

ケース 1: 同じ処理負荷 (50msec) の LU を並列処理させた場合 (図 3)

ケース 2: ある既存の SoS シミュレーションプログラムの一部を LU として移植し、並列処理させた場合 (図 4)

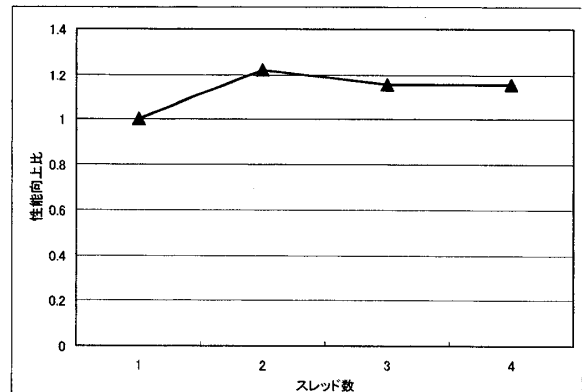


図 4 ケース 2 計測結果

ケース 1 の計測結果から、並列処理におけるアムダールの法則により、SE 内部における並列化できない処理の割合が、計測条件下では 8%程度であると見積ることができる。一方、ケース 2 の計測結果ではスレッド数が 2 つで性能向上が頭打ちとなっているが、このケースでは SoS のサブシステムを模擬する LU 間で処理負荷のばらつきが非常に大きく、負荷の大きな LU のスレッドが全体処理に対する足枷となっており、SE を実際に SoS シミュレーションに適用した場合における課題を確認することができた。

5. おわりに

本稿では、SoS を想定した大規模シミュレーションを対象に、高速かつ高精度なシミュレーションの実行と、ソフトウェアの再利用性向上に寄与するような開発・実行環境として試作した SE について説明し、そのマルチスレッド化処理についての性能評価結果を示した。今後は、今

参考文献

- [1] 鎌田, “SoS: 超システム工学への現実的アプローチ—現場/チーム重視, 変更自由の巨大システムを実現するモデル駆動工学,” OR The Object Report, Vol.1 No.2, pp.1-4, Aug. 2007.
- [2] IEEE Std 1516-2000 IEEE Standard for Modeling & Simulation (M&S) High Level Architecture (HLA) -Framework and Rules.
- [3] 尾崎, 松下, 白石, 渡部, “自律型及び他律型オブジェクト群を対象としたシミュレーション開発実行環境”, 信学技報 SANE2009-5, pp.25-30, Apr. 2009.
- [4] 尾崎, 渡部, 松下, “並列分散型シミュレーションエンジンにおけるマルチスレッド化方式の検討”, 第 10 回 計測自動制御学会 システムインテグレーション部門講演会, Dec. 2009.