

オーケストレータ機構による防災シミュレーション連携技術のリアルタイム化 Real-time Federation of Disaster Prevention Simulation using Orchestrator Framework

○廣井 慧・小比賀亮仁・篠田陽一

○Kei HIROI・Akihito KOHIGA・Yoichi SHINODA

We develop an orchestration scheme that coordinates several independent simulators into one simulation to observe a mutual related phenomenon (ex. evacuation behavior in flood simulation). In this paper, we propose a real-time orchestration scheme that can cooperate with actual systems and simulators. In real-time orchestration, the processing time of the real systems and simulators is not constant, and unexpected delays can be a major problem in orchestration. We define a standard processing time function for a virtual time system, which is one of the functions of an orchestration scheme, observe the processing time of each actual system or simulator, and develop a mechanism to control timing autonomously according to the processing time. We report the results of real-time orchestration on a simulator with a simple numerical analysis model and an actual system.

1. はじめに

我々は、防災に関するシステム、アプリケーションを性能検証する防災 IT テストベッド上に、災害に関するシステム、シミュレーションを連携動作させる、オーケストレータ機構を開発している。オーケストレータ機構は、機能の分割、分散、協調を重視し、システム、シミュレーション等の防災要素技術を協調連携させる機構である。従来の防災に関するシステムは、1 つのシステムの中にすべての機能を盛り込むモノリシック構成が主であり、いちからシステムを作り込む必要があり、開発導入に際し、多大なコストを必要とし、防災のシステムを考えるうえで、大きな障壁となる。

それに対し、オーケストレータ機構は、防災 IT テストベッドの連携基盤を中心として、防災シミュレーションのコンポーネントを協調連携させる。それぞれのコンポーネントの独立性を維持しながら高度なサービスの提供を重視したアーキテクチャであり、この構成を採用することで、開発、導入、改修コストを削減するだけでなく、各防災要素技術の機能、部品でベストなシステム、シミュレータを採用し、柔軟に入れ替えることが可能となる。

2. 連携技術のリアルタイム化

オーケストレータ機構の実現には、異なるシミュレータ、システム同士の物理的なデータ交換に加え、データ交換の相互性の担保や大規模データ

の逐次的な処理、タイミング制御が必要となる。

我々はこれまでに、オーケストレータ機構に関し、静的なデータ交換を可能にするために仮想時刻サーバによる複数シミュレーションのタイミング制御を実装してきた。ここでは、あらかじめ各シミュレータ、システムの実行時間を設定し、その設定値に合わせてタイミング制御を実行する。本稿ではこのオーケストレータ機構のリアルタイム化を試みる。ここでいうリアルタイム化とは、実際の実働システム、シミュレータを連携させるためのデータ交換、タイミング制御となる。オーケストレータ機能を使って、シミュレーションを実行し、予測結果を受け取るユーザが、例えば「シミュレーション結果を 10 秒後に受け取りたい」とした場合、オーケストレータ機構は要求に従って、シミュレーションの実行粒度を変更し、シミュレーションの実行時間を調整可能となる。

オーケストレータ機構のリアルタイム化に向けて、実行速度に関する解決すべき課題が 2 つある。ひとつめは、実働システム、シミュレータの処理時間の変動に関する課題である。各実働システム、シミュレータの処理時間は一定ではなく、突発的に速度が変化する。これは、実働システム、シミュレータそのものの動作に由来する影響と、並列実装のための分散化による影響の 2 種類が考えられる。特に分散化による影響は、モノリシックに構成されていた従来の環境では発生しにくかった。ふたつめは、数値解析を行うシミュレータの処理

速度に関する課題である。これらのシミュレータは一般的な実働システムより実行時間を要する傾向にあり、また、シミュレータを実行する環境(CPU速度やメモリアクセスのレイテンシなど)や解析内容によって処理速度が変化する。従って、「シミュレーション結果を10秒後に受け取りたい」という要求を満たすには、これらの変動を踏まえたうえで、制約時間内にシミュレーションを終了させて結果を得る必要がある。

これら2つの課題を解決するため、はじめに実働システム、シミュレータを用いて、実行時間について観察を行った。その結果、実行時間の変化や突発的な遅延の発生が確認され、実働システム、シミュレータは、時間パラメータの変化に対して必ずしもリニアな実行時間を返すとは限らず、特異点があることが判明した。このことから、実行時間についてリニアな変化を見込んでオーケストラ機構を設計すると、連携が破綻する可能性があり、実働システム、シミュレータの性質を踏まえた、処理粒度を変化させ、実行時間をリアルタイムに観察する必要があるということがわかった。

次に、仮想時刻サーバにあらかじめ設定していた実行時間を実働システムやシミュレータの実際の実行速度を鑑みて作成できるよう標準処理速度関数を組み込み、実行粒度の制御を実現した。標

準処理速度関数とは、一定の計算機環境とシミュレーションパラメータ(粒度)であれば、どれくらいの時間で処理結果を得られるかということ算出する、単位数を求めるための関数である。さらに、ユーザからの処理時間要求に合わせてシミュレーションのパラメータ(粒度)を選択する仕組みを導入した。これによって、シミュレーションの粒度は粗いが、ユーザからの時間制約に合わせて処理結果を算出することが可能となった。これまでに、簡易な数値解析モデルでその動作を確認している。以上の、実行時間を実際の実行速度をみて決定できるようにしたこと、速度の遅い時刻粒度を変更できるようにしたことにより、実働システム・シミュレータの連携が可能となり、実時間に合わせたオーケストラ機構の運用が可能となった。

参考文献

- [1] 廣井慧, 小比賀亮仁, 片岡拓海, 深谷将, 篠田陽一, ARIA: Society5.0時代の減災オープンプラットフォーム- AIを防災にどう生かすか-, 人工知能, 38巻, 1号, p.10-19, 2023.
- [2] 廣井慧, 小比賀亮仁, 篠田陽一, オーケストラ機構による防災シミュレーション連携技術の開発, 令和3年度 京都大学防災研究所 研究発表講演会, 2023.