

FX10 スーパーコンピュータシステム「大規模 HPC チャレンジ」採択課題のお知らせ

1. はじめに

東京大学情報基盤センターでは、FX10 スーパーコンピュータシステムで「大規模 HPC チャレンジ」サービスを実施しています。「大規模 HPC チャレンジ」は、FX10 スーパーコンピュータシステム (Oakleaf-FX) がもつ最大計算ノード数である、4,800 ノード (76,800 コア) を、最大 24 時間・1 研究グループで計算資源の占有利用ができるサービスです。大規模 HPC チャレンジ (2014 年度第 2 回) の採択課題について、以下の通り決定しましたので、お知らせいたします。

2. 採択課題について

今回は、2014 年 2 月 18 日 (火) ~ 2014 年 7 月 14 日 (月) までの期間で課題募集を行い、3 件の応募がありました。課題採択委員会による厳正な課題審査の結果、全ての課題について採択することとしました。

採択課題一覧

課題名	ppOpen-HPC ライブラリ群を利用する地震波動-建築物振動連成マルチスケール・マルチフィジックスシミュレーション
代表者名 (所属)	松本 正晴 (東京大学 情報基盤センター)
地震による建物の揺れをその震源から解析するためには 100km×100km 程度以上の広い領域で起きる地震波動の進展と、数 10m×数 10m 程度の狭い領域で起きる建物の振動を同時に解く必要がある。しかしこのような解析は、対象とするスケール・物理が両方で大きく異なるため、そのモデル化から実装・評価が非常に難しい。そこで本研究課題では、地震波動解析と建築物振動解析それぞれに適した既存の解析コードを、カップラを介して連成させるマルチスケール・マルチフィジックスシミュレーションを目指す。すなわち、我々の研究グループで開発が進められている ppOpen-HPC ライブラリ群を用いて、有限差分法 (FDM) による地震波動の広域的解析と有限要素法 (FEM) による建築物振動の局所的解析をカップラにより連成させ、FDM-FEM 大規模連成解析を実証するとともに、その性能・精度評価、課題点の洗い出し等を行う。	

課題名	コストス配置問題に対する超並列組み合わせ最適化解法
代表者名 (所属)	須田 礼仁 (東京大学 情報理工学系研究科)
本研究では、コストス配置問題 (以下 CAP と略す) の史上最大規模問題 (n=32) を、我々が開発した制約ベースの局所探索アルゴリズムである Adaptive Search で解く。 CAP は、ノイズについでレーダーとソナーの周波数を配置するという実世界の問題をモデル化したものである。CAP の探索空間はサイズ n に対して指数的となり、探索が困難となる。45 年前に提唱されて以来、n=32 に解があるかどうかは未解決問題となっている。 我々は Adaptive Search の効率的な並列実装とともに、その実行所要時間を正確に見積もる手法を開発している。これに基づいて評価したところ、FX10 4,800 ノードでは n=32 の問題に対して約 24 時間以内に解が得られると見積もっている。 これを達成することにより、CAP 問題の新記録が達成されるだけでなく、探索問題の超大規模並列化に関する多数の知見が得られると期待される。	

課題名	並列多重格子法ソルバーの最適化および性能評価
代表者名 (所属)	中島 研吾 (東京大学 情報基盤センター)
連立一次方程式の反復解法、前処理手法としての多重格子法は、問題規模が増加しても収束までの反復回数に変化しないスケラブルな手法であり、大規模問題向けの解法として注目されている。並列計算においてもその効果が確認されている。本研究は 2 つのフェーズに別れている。第 1 フェーズは 2013 年 6 月に Linpack に代わるスーパーコンピュータシステムの新しい性能評価ベンチマークとして提案されている HPCG (https://software.sandia.gov/hpcg/) のチューニング及び性能計測である。HPCG は三次元ポアソン方程式を差分格子のような規則的形狀において有限要素法によって離散化して得られる疎行列を係数とする連立一次方程式を幾何学的多重格子法前処理による共役勾配法 (Conjugate Gradient Method preconditioned by Geometric Multigrid) を使用して解くベンチマークである。第 2 フェーズは申込者がこれまで実施してきた幾何学的多重格子法前処理による並列共役勾配法における通信手法削減に関する研究である。申込者の開発した hCGA 法 (Hierarchical Coarse Grid Aggregation) を更に改良し、計算を実行し、性能を計測する。	