

# リソース貸借システム“Ourtree”の実現に向けた 事前性能評価

岸端 晃毅\*      平野 太一\*      成田 匡輝\*

## 1 はじめに

計算資源を共有する試みは分散コンピューティング分野では以前から行われており、代表的なものとしてグリッドコンピューティングやボランティアコンピューティングを挙げることができる。グリッドコンピューティングは広域なネットワーク上の計算資源を1つにまとめあげ、膨大なデータ保存や計算処理を行うことを目的としている。ボランティアコンピューティングは計算資源を1つにまとめる点は共通であるが、遊休資源を研究目的のプロジェクトに対して提供することが目的という点で異なる。

一方、これらの分散コンピューティング手法は汎用性が低いため、その恩恵を個人や組織は受けられていないという問題がある。近年、端末の低価格化やIoT化で個人や組織の持つ総合的な計算資源は増加している。しかし、その計算資源の殆どは端末内で利用するデータに対して使われており、最大限にリソースを活用できていない状況にある。我々は個人や個人間、組織が計算資源共有の恩恵を受けられるシステム（以下 Ourtree）の開発を行っている。本稿ではプロトタイプを用いてシステムのタスク処理性能評価を行う。

## 2 Ourtree のシステム概要

Ourtree は異なる端末のストレージ資源を含む計算資源を貸借可能なミドルウェアシステムである。資源の貸借は独自形式で記述されたタ

スクによって行われ、汎用性の高い処理を実行可能としている。ここでの汎用性の高い処理とは、分散することで処理時間や負荷低減を実現できる幅広い種類の処理と定義している。タスクに記述可能な処理の粒度は、頻度と必要となる柔軟性で決定される。定型で利用される処理はまとまった単位、分岐のような柔軟性を提供する処理については単一の単位で提供する。

## 3 既存システムとの比較

### 3.1 ボランティアコンピューティング

BOINC[1]のようなボランティアコンピューティング及びグリッドコンピューティングの既存システムとの違いは2つあり、ユーザが計算資源の貸借の両方を行うことが可能な点と汎用性の高い処理を前提としている点である。多くの既存システムは設計思想として処理を要求する側と資源を提供する側に分かれることを想定しており、資源提供者は資源共有自体の恩恵を受けることはできていない。処理の汎用性という面ではプロジェクト単位で様々な処理が対象となっているが、個別では特定の計算処理に特化したものが多い。

### 3.2 分散処理基盤

分散した端末の計算資源を利用可能な既存システムには Hadoop[2]のような分散処理基盤がある。本研究と分散処理基盤との違いは上述した汎用性と前提としている計算資源の信頼度である。分散処理基盤では計算資源は管理された区域内で提供されることを前提としており、信頼性が高い資源の利用を前提としている。一方で本システムは個人間や組織間での利用も想

\* 岩手県立大学 ソフトウェア情報学部 ソフトウェア情報学科

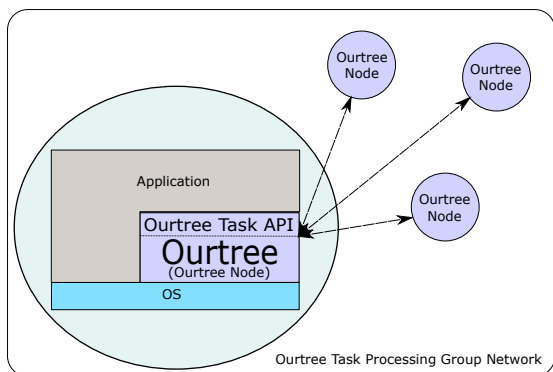


図1 タスク実行グループのノード

定しており、信頼性の低い資源が含まれている可能性も考慮する。

#### 4 システム構成

Ourtree はタスク処理のためのグループを作成するネットワーク管理モジュールとタスクの実行の可否を決定しタスクの実行を行うタスク実行モジュールによって構成される。ネットワーク管理モジュールの役割は P2P 通信によるノード間での端末情報の交換とそれらの情報に基づいたオーバーレイグループネットワークを構成することである [図 1]。タスク実行モジュールは他ノードからのタスク実行リクエストを受け、自らの端末状態に基づいて実行をするか否かを決定する。実行する場合にはタスク実行リクエストの送信元と通信を行い、必要なデータを受け取った後に実行する。この際、タスク依頼側のグループネットワークはネットワーク管理モジュールを介してタスク実行に適した形に構成される。実行しない場合にはノードリストの中からランダムにノードを選びリクエストを受け流しタスクリレーを行う。

##### 4.1 プロトタイプの開発

本稿ではプロトタイプを開発し本システムの開発にあたって前提となる処理分散の性能を確認する。具体的にはある計算タスクを用意しシステム上で実行することで、負荷分散と処理時間短縮が実現可能であると示すことである。計算タスクはグラフィックス分野の 3 次元空間の面とレイの交差判定処理を模したものの作成した。Ourtree とプロトタイプではタスク処理の

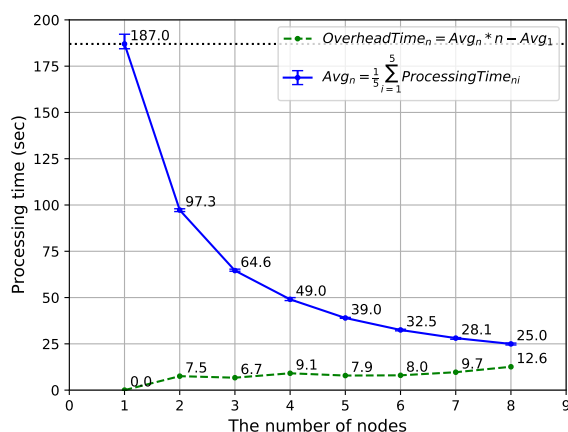


図2 ノード数毎の処理時間とオーバーヘッド時間

境界が異なっており、Ourtree は API に対してタスク要求を行う明確な境界を持つ設計としているが、プロトタイプでは Ourtree の機能がアプリケーションに含まれている。

#### 5 実験結果と考察

タスク処理性能を確かめるためプロトタイプを用いて実験を行った。実験ではノード数を増加させた際の処理時間を取得した [図 2]。ノード数が 2 の場合はノード数が 1 の場合と比較して 4% 程度のオーバーヘッドがあり、ノード数の増加と共に微増した。結果から適切なグループサイズはリアルタイム性が求められる処理の場合には高性能な少ないノード数、処理量が大きくリアルタイム性が求められない処理の場合には多いノード数になると考えられる。

#### 6 おわりに

本稿では計算資源の貸借を実現する Ourtree のプロトタイプを作成し性能を評価した。プロトタイプ実験の結果では単一端末での処理性能と比べてシステム上での処理性能が高いことが確認された。今後はシステムの実用化に向けタスク実行の汎用性を高めるとともにセキュリティの改善とシステム最適化が必要である。

#### 参考文献

- [1] BOINC: <https://boinc.berkeley.edu/>.
- [2] Apache Hadoop: <https://hadoop.apache.org/>.