

## 1. はじめに

本研究では、演習用 PC を用いた大規模計算システムの構築を計画している[1]. 本研究では、ネットワーク上の異なるセグメントに存在する演習用 PC を管理するために HTTP 通信によるカプセル化を導入し、カプセル化によるオーバーヘッドを明らかにする.

## 2. 提案するグリッドシステム

本校は、その性質から演習室に多くの PC を持っている. これらの PC の多くは過去の調査より遊休状態にあることがわかっている. そこで、遊休状態の PC を利用した大規模計算環境を構築する目的で、タスクスケジューラ Condor を用いたデスクトップグリッドの構築を計画している[1]. このグリッドシステムは、タスクを投入する SM とそのタスクを実行する EM, および EM へのタスク割当てを制御する CM から構成される(図 1).

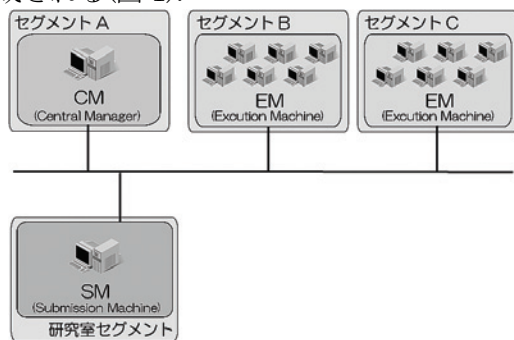


図 1 グリッドシステムの概要

EM が配置される各演習室はネットワーク上の異なるセグメントに存在しており、セグメント間の通信をできない. そのため、グリッドシステムを運用する場合、セグメントを超えたタスクの管理ができない状況にある. すなわち、各セグメントに CM を用意し、各セグメント内の EM にしかタスクを割当てることができない.

そこで、本研究では SOAP (Simple Object Access Protocol) を Condor に適用し、グリッドシステム上の通信を HTTP 通信でカプセル化する. その結果、セグメントを超えたタスクの管理が可能となる. しかし、HTTP 通信でカプセル化することによる通信上のオーバーヘッドが心配となるため、オーバーヘッドを数値実験により明らかにする.

## 3. 実験

実験環境を図 2 に示す. 実験では、100MB, 300MB, 500MB, 800MB, 1GB のデータのクイックソートをタスクとして用いる. 各タスクの実行時間を 10 回測定する. 図 3 では、(1)SOAP あり (SM: 研究室セグメント, EM: セグメント B), (2)SOAP あり (SM: 研究室セグメント, EM: セグメント C), (3) SOAP なし (SM, EM, CM: セグメント A) の 3 通りの実行速度の平均値と直線回帰の結果を示す. 直線回帰の切片よりオーバーヘッドが 20 秒程度であることがわかる.

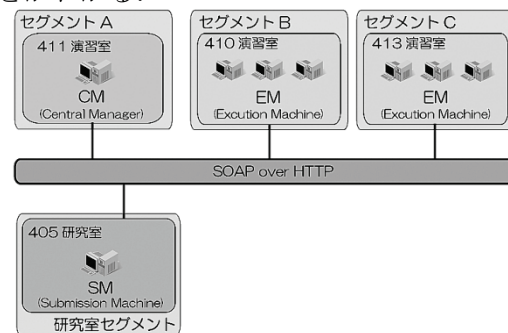


図 2 SOAP による Condor 構築環境

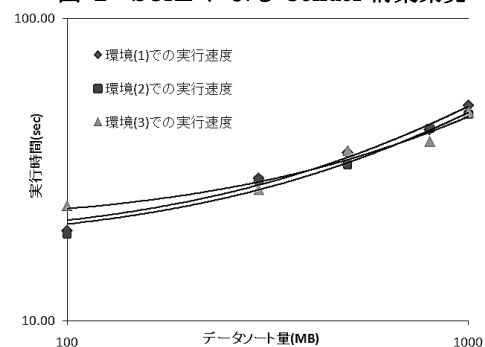


図 3 クイックソート実行時間の平均値と直線回帰

## 4. おわりに

本研究では、異なるネットワークセグメントに存在する演習室 PC を大規模計算システムの資源として利用するために、SOAP を Condor に導入した. 数値実験の結果、SOAP による通信のカプセル化によるオーバーヘッドは、SOAP を使用しない場合のものとはほぼ等しくなることが明らかになった.

## 参考文献

- [1] 中村 紘土, “サレジオ工業高等工業専門学校における計算資源の調査に基づくグリッドシステムの検討”, サレジオ高専卒業論文, 2006