

グリッド環境におけるサーバへのオンデマンドなライブラリ自動配置機構の提案

所属： 高性能コンピューティング学講座
 発表者： 0853020 松本 優人
 指導教員： 本多 弘樹

1 はじめに

グリッド環境上で動作するアプリケーションを開発する上で、動的なサーバ運用状況監視を実装することやサーバへのライブラリ配置は、開発者にとって負担となる。また、利用するサーバを追加する場合、ライブラリ配置作業やアプリケーションの中断など、利用者にとって負担となる作業が発生する。

本研究では、グリッドミドルウェア Globus Toolkit 上で動作する GridRPC を用いたアプリケーションを対象として、

- 各サーバの運用状況に応じて使用サーバを自動的に切り替える機構
- 切り替え先サーバへライブラリを自動的にインストールする機構

の2つを組み合わせることで実現することにより、利用者および開発者の負担を軽減することを提案する。

本機構により、利用者は計算サーバ数を簡単に増やすことが可能となる。また、開発者がサーバへライブラリを個別にインストールする手間がなくなる。

2 背景

2.1 従来の問題点

本文では、グリッドコンピューティングを実現している実行環境をグリッド環境、グリッド環境上で実行されるアプリケーションを開発する人を開発者、グリッド環境においてアプリケーションを実行する人を利用者と呼ぶ。アプリケーションはさまざまなモデルが考えられるが、以下はマスタ・ワーカモデルの一般的な実装である GridRPC を用いたアプリケーションを対象とする (図 1)。

グリッド環境は利用者にとって簡便であるために多数の仕組みが考案されている。しかし、グリッド環境に対応したアプリケーション開発という開発者の観点では、下記のような負担がある。

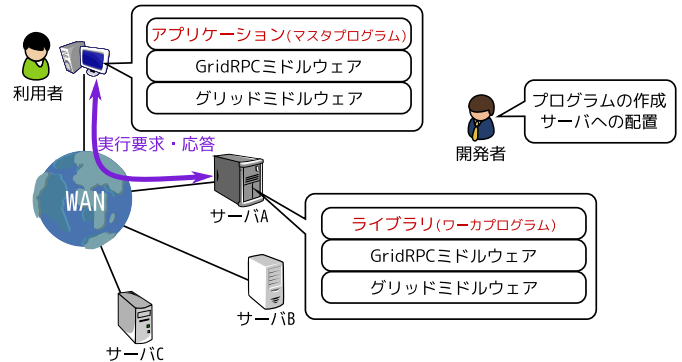


図 1: グリッド環境

- (A) 各サーバへのライブラリ配置 グリッド環境の内部にはさまざまなプラットフォームが混在することが一般的である。開発者はプラットフォームの違いを知り、アプリケーションに必要なライブラリをすべてのサーバへ適切に配置しなければならない。
- (B) サーバ運用状況の監視 グリッド環境は複数の管理者によって運営されているために、運用方針もそれぞれのサーバごとに異なっている。開発者はアプリケーションに対して、各サーバの運用状況に応じて使用するサーバを動的に切り替える機能を組みこまなければならない。

2.2 関連研究

これらの問題に対するアプローチとして、渡邊は2つの機構を提案している [1, 2]。

(A) について、サーバに対して必要なライブラリの一斉配置する機構を提案している [1]。これは、開発者がインストール手順を一定の書式で記述することにより、1つのコマンドを実行すればすべてのサーバにライブラリを配置される仕組みになっている。Globus Toolkit と Bourne シェルスクリプトを用いることで、特定のプラットフォームに依存しないように設計されている。

また (B) について、GridRPC を用いたアプリケーションについて使用するサーバを運用状況に応じて動的に切り替えるための機構を提案している [2]。これは、サーバ切り替えなどをイベントとして通知する仕組みにより、開発者はイベントハンドラを記述するだけで済むように設計されている。

これら 2 つの機構は対象とする問題が異なるために、別のシステムとしてそれぞれ独立に設計されている。

3 目的

渡邊の提案のように機構をそれぞれ別のものとして実装することで、(A) および (B) に関する開発者の負担はそれぞれ軽減する。しかし、アプリケーション利用者の立場から考えると、サーバ追加に関して煩雑な作業が必要であるという問題が依然としてある。利用するサーバを追加しようと利用者が考えた場合、追加しようとしているサーバへライブラリを配置する手間と、アプリケーションのサーバ監視に関する設定を変更する手間が生じる。また、もしアプリケーションの実行中にサーバを追加するならば、アプリケーションを一旦終了してやり直さなければならない。

そこで本研究では、前述の渡邊の提案するサーバ自動切替え機構に対して、さらにオンデマンドなライブラリ配置機構を付加することを提案する。つまり、もしサーバ切り替え要求時に切り替え先のサーバに必要なライブラリがインストールされていない場合、必要ライブラリを切り替え先サーバへ自動的にインストールするという機構である。

GridRPC を用いたアプリケーションは、利用者の作業用計算機に対して処理の分割・統合を行うマスタ、グリッド上の利用するすべてのサーバに対して分割された処理を行うワーカをそれぞれ配置することが必要となる。本文ではワーカはサーバ上のライブラリに、マスタプログラムは利用者の計算機上で実行されるアプリケーションに相当する。一般的な GridRPC アプリケーションは、使用するすべてのサーバにライブラリが正しく配置されていないと処理を行うことができない。本機構は、ライブラリがサーバに正しく配置されているかどうかを、そのサーバへの処理要求時にサーバ自動切替え機構が判断する。そして、必要であればライブラリをサーバ上に構築するという作業を行う (図 2)。

この機構によって、利用者がサーバを追加した

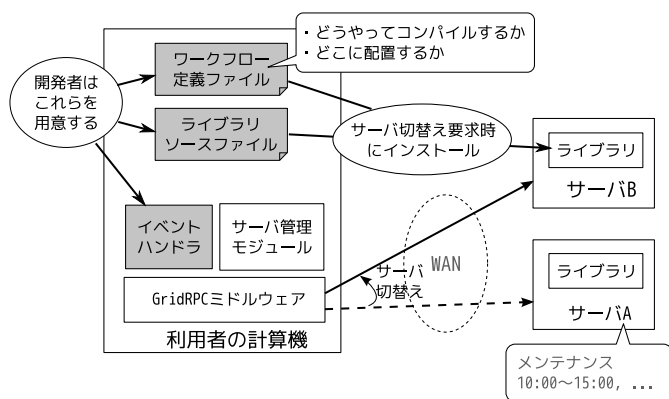


図 2: オンデマンドなライブラリ自動配置

いと考えたとき、アプリケーションのサーバ監視に関する設定に対象サーバを追加するだけで実現できる。さまざまなプラットフォームが存在するグリッド環境において、利用者がインストール作業を行わなくて良いことは有用である。

また、この機構は開発者のライブラリ配置における手間を軽減する。アプリケーションが本機構を実装すると、実行前にサーバへライブラリを配置する必要がなくなる。開発者は必要なライブラリのソースコード、およびインストール先へどのようにインストールするかを定義したファイル(ワークフロー定義ファイル)を用意することで、ライブラリを個別にサーバへ配置する手間がなくなる。そのために、開発者はライブラリ配置作業から解放され、具体的にどのサーバで処理が実行されるかを考慮せずにアプリケーション開発を行うことができる。

4 今後の研究計画

現在、提案する機構の実装作業を行っている。具体的には、[2] によるサーバ切替え機構に対してライブラリ自動配置機能を付加するために追加修正している。

参考文献

- [1] 渡邊啓正, 本多弘樹, 弓場敏嗣, 田中良夫, 佐藤三久. Relis-g : 計算グリッドのための遠隔ライブラリインストール機構. 情報処理学会論文誌. コンピューティングシステム, Vol. 45, No. 11, pp. 196-206, 2004.
- [2] 渡邊啓正, 平澤将一, 本多弘樹. F-omega : サーバ稼動状況に適應する gridrpc アプリケーションの開発・実行フレームワーク. 情報処理学会研究報告. [ハイパフォーマンスコンピューティング], Vol. 2007, No. 17, pp. 181-186, 2007.