

ネットワーク数値情報ライブラリ: — Ninf の設計 —

関口智嗣[†], 佐藤三久[†], 長嶋雲兵[‡]

[†] 電子技術総合研究所, [‡] お茶の水女子大学

Email: {sekiguchi, msato}@etl.go.jp, umpei@is.ocha.ac.jp

高速ネットワークの新次元の利用がハイパフォーマンスコンピューティングとして総括される計算科学技術, 数値解析技術に与える新しい可能性としてネットワーク数値情報ライブラリ **Ninf** (Network based **I**nformation library for High Performance Computing) の設計を開始した. 本稿においては **Ninf** の設計方針, ユーザインタフェース, ライブラリ設計・実現に関してその概要を述べる.

— *Ninf* — : Network based Information Library for High Performance Computing

SEKIGUCHI Satoshi[†], SATO Mitsuhsa[†], and NAGASHIMA Umpei[‡]

[†]Electrotechnical Laboratory, [‡]Ochanomizu University

The authors have started the design for **Ninf** (Network based **I**nformation library for High Performance Computing), which is an exciting idea for new era of high performance computing and high speed wide area network. The **Ninf** provides numerical and mathematical constants of physics, chemistry, *etc.*, computing special functions with desired accuracy, and high speed computing for scientific libraries. In this article, it is overviewed that design concept of the **Ninf** and its user interface, and implementation idea of the **Ninf**.

1 はじめに

ネットワーク技術の発展にともない、手元のワークステーション、パーソナルコンピュータからローカルエリアネットワークのみならず、インターネットへの接続も容易になってきた。ネットワークを通して物理的な距離を越えて、計算リソースへのアクセスが可能となっている。rlogin, telnet 等のコマンドで計算リソースにアクセスを行ったり、rcp, ftp 等のコマンドで同様にリソースへアクセスを行ったりすることができる。それ以前の計算センターへ足を運ぶとか、フロッピーディスクを郵便でやりとりすることと較べるとその安定性、高速性、信頼性の点で圧倒的に効率的である。しかし、これらは自分が利用を承認されているリソースにしかアクセスできない。

一方、共有ソフトウェアの配布やデータの公開の目的に用いられている ftp を匿名 (anonymous) で利用することができるようになってきた。この方法は利用制限を加えているとはいえ、一般に公開されたり、anonymous-ftp により、読みだし専用であるがアクセスできるリソースが爆発的に増大している。最近では、Gopher, WWW, Mosaic 等によりマルチメディア情報をオンラインでアクセス可能とするようなサーバーとクライアントソフトウェアが開発され、計算機科学者に限らずネットワークを通じた情報が豊富になり、有用な情報検索ツールになってきた。

社会的にもインターネット接続を提供する事業が認可され、これまでは接続費用、回線の維持管理、ネットワークの利用制限等、敷居の高かったインターネット接続が小事業所、個人経営者、個人趣味のレベルまで広がってきた。また科学技術庁主導の省際ネットワーク構想で大学、研究所等はより高速な回線により接続されるなど、インターネットワークは非常に身近な技術となってきた。

こうした高速ネットワークの利用技術に対する期待が高まる中、筆者らは高速ネットワークの新次元の利用がハイパフォーマンスコンピューティングとして総括される計算科

学技術、数値解析技術に与える新しい可能性としてネットワーク数値情報ライブラリ Ninf (Network based Information library for High Performance Computing) の設計を開始した。Ninf の愛称は『Nymph』に由来するもので、あたかもネットワークの森に棲む妖精のように人知れずながらも願いを叶えてくれるようなシステムを目指している。本稿においては Ninf の設計方針、ユーザインタフェース、ライブラリ設計・実現に関してその概要を述べる。

2 Ninf 設計の目的

2.1 情報資源の仮想的共有化

『ハッカーを追え』というノンフィクションにおいてサイバースペースというコンピュータネットワークがもたらす仮想的な空間が紹介されている。実際、インターネットワークにより従来の物理的な時空間とは異なる世界を通じて様々な情報を得たり与えたりしている。このサイバースペースではネットワーク社会学的な考察が行なわれていたが、科学技術に対してはどのような可能性が秘められているであろうか。

こうしたサイバースペースまたはサイバーネットワーク空間 (Cyber Network Space) について考察してみると、その本質的な特徴はサーバーネットワーク空間における情報資源の仮想的共有化が可能であることと考えられる。すなわち、サイバーネットワーク空間においては情報流通のためのコストが理想的には零になることに基づいている。このことから、情報資源について

- どこに存在しても、誰が管理しているも
- 必要な時点で即時性を持って欲求 / 要求を満足してくれればよく、
- したがって、手元にその資源を保持する必要はない

という仮想的な資源の共有化が可能であるという性質が浮かび上がってくる。これはちょうど『家庭で電灯をつける』ようなものを類推すると理解を助けるであろう。必要なこと

は電気が灯ることであって、

- その電気がどの発電所で生成されたか、
- どのような機構 (火力、水力、原子力等) で発生したか、
- どういった送電経路を通じて家庭まで届いているのか
- ましてや、自家発電設備を持つこともない

言い替えると、送電ネットワークの即時性が諸々の問題をすべて隠蔽し、本質的な「電灯がつく」という機能だけに注目することを可能としたと考えられる。

電気の世界ではほぼ理想的な資源の共有化に成功しているのに対して、情報資源に関してはまだまだ理想からは離れている。例えば、ある優れたソフトウェアをサイバーネットワーク空間において見つけたとしよう。先にも述べたよう、このソフトウェアを実際に利用するには

- まず、ソフトウェアを置いてあるサーバーを探し、
- ftp 等で手元に持ってきて、
- 実行可能な形式に整える

という過程を経る。また、ソフトウェアのバージョンアップ等に即時に対応できるわけではない。このようにソフトウェアは静的な形式で保管されており、そのソフトウェアが提供する機能だけを利用するようにはできていない。

2.2 Ninf 設計の概要

科学技術計算の分野においては、より複雑な事象の解明のため高精度かつ高信頼性のある計算をより高速に実行すること、すなわちハイパフォーマンスコンピューティングが求められてきている。これまでに述べてきたネットワーク技術によりハイパフォーマンスコンピューティングを支える資源は共有されている。例えば、超高速計算機システム、ファイルサーバー、プリンタなどのハードウェア資源、高性能・高精度数値ライブラリ、数値的解法、可視化ツール、最適化支援ツ

ルなどのソフトウェア資源の共有は進んでいる。

しかし、共有化された資源とはいえ、超高速計算機システムではマルチベクトル、クラスタ、超並列機などといったアーキテクチャの差異や共有メモリモデル、データ並列、分散メモリモデルといったプログラムモデルの差異などユーザが留意しなくてはならない点が多々ある。サイバーネットワーク空間の理想を追求すれば情報資源の共有はハードウェア資源、ソフトウェア資源そのものを共有することではない。知識資源の共有を目指し、優れたアルゴリズム・プログラムを

- いつでもどこからでも、
- 必要な時に利用して結果を得られる、
- すなわち、先人達の知識 / 知恵を動態保存する

ことが求められている。先人達の知識 / 知恵という意味では上記のような動態保存に限らず、ハイパフォーマンスコンピューティングが対象とする計算物理学、計算化学などで必要となる物理定数、化学定数、数学定数などの数値定数の記述、数表からの検索、といった定数表、国勢図会や白書などの統計数値データなどをいつでもどこからでもネットワークを通じてアクセスすることにより、科学技術計算、社会科学計算の基礎、基盤技術を提供することが可能となる。

Ninf ではこうした数値定数、数表検索をはじめ、特殊関数の必要な精度に応じた関数値、線形計算ライブラリ処理などの数値情報を取り扱うサービスを提供するサーバーをインターネット上に複数設定し、遠隔地のユーザは Ninf のライブラリ関数を呼び出すだけで、Ninf サービスを受けられるインターフェースを提供する (図 1)。

ネットワークに接続された PC や EWS も最近では高性能であり、大規模メモリ、CPU 酷使型のプログラム以外はローカルに処理した方が高速である。しかし、実際に大規模メモリや CPU を必要とする計算を含んでいた場合、その部分だけを高速な Ninf 計算サーバーにネットワーク経由で依頼し結果を得る。同様に、滅多に利用しない高精度の関

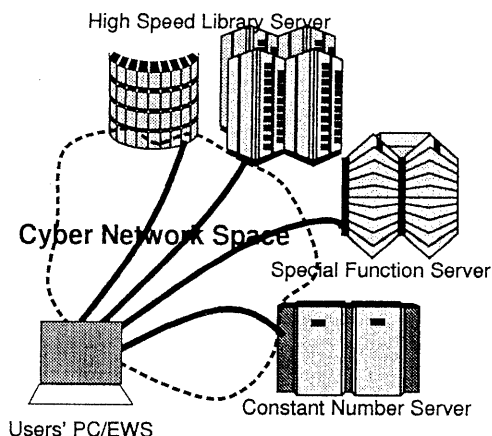


図 1: Ninf: Conceptual Model

数値計算，定数表の参照などがある場合にもその関数や表を維持しているサーバーに要求を出して結果を得ることが可能となる．関数が動態保存してあるため，その場合にもソフトを ftp して実行形式にした後で実際に組み込むといった手間をかける必要がない．

こうしたサイバーネットワーク空間に置ける数値情報サービスを実現することにより，知識 / 知恵を動態保存し，享受一辺倒だったネットワーク世界への情報発信が可能となる．また，システムの相違，言語モデルの差異，ライブラリコールの些細な差異を隠蔽できるため，本質的な計算部分に専念できる．

2.2.1 数値定数，数表検索

物理定数，化学定数，数学定数は計算機に利用できる形式で超高精度に維持，提供される必要がある．しかし，従来のプログラム開発においては，数値定数はプログラマーが数値を個別にプログラムの中で記述して利用する．このため，複写ミス，計算機内部表現における精度などの点で信頼性，移植性に欠ける短所があった．また，観測に基づく最新の数値や超高精度の数値（例えば， π の値を 100 万桁）が必要な場合には個々のプログラムを変更して維持する必要があるが，このことは事実上困難である．そこで，Ninf の数値定数ライブラリを呼び出すことにより，必要な数値を必要な桁数だけ，または計算機に合

せた精度で獲得することができる．

2.2.2 特殊関数ライブラリ

特殊関数の計算は高速計算と精度の信頼性確保のため，様々なアルゴリズム上の工夫が適用されている．最新のアルゴリズム技術が実際の計算で用いられるようになるためには，時間的遅れが生じることと，通常のワークステーション，パーソナルコンピュータ等では演算時間，メモリ容量などの制約により最新のアルゴリズムを適用できない場合にも大規模な計算機資源を有するサーバーで計算を実行する Ninf は有効である．

また，Ninf のようなサーバーを利用することで，滅多に利用することのない関数などもユーザ側でライブラリのインストールなど特に設定することは何もなく，容易に関数呼び出しを行うことができる．必要な桁数を求めるための多倍長演算により短時間に必要な精度を得ることができるためには高速計算機をネットワークで利用することが不可欠である．

2.2.3 数値計算ライブラリ

ネットワークを通じて数値計算ライブラリを入手するにはこれまで，米国の Oak Ridge 研究所と Tennessee 大などで開発されている netlib が有名である．netlib に対してはユーザは mail を送付してプログラムを要求したり，anonymous ftp により，interactive に必要なファイルをコピーする方法によって，ライブラリのソースコードを手にする．ユーザは手にしたソースプログラムを自分の環境でコンパイルし，適当にリンクして利用する．しかし，この方法の欠点はファイルを得てコンパイルを行ない，ライブラリ化する手間が掛かることと，version up への対応にも遅れが生じることである．この計算した結果を二次加工して科学的な結論を導く場合には利用したライブラリのバージョン，コンパイルした環境，実行した環境に依存した数値のふれなどを慎重に考慮する必要がある．しかし，このようにローカルな処理を行った場合には計算環境を一致させることが

困難である。Ninf を用いて同一サーバーで実行することにより、ライブラリに関しては同一環境の結果を与えることが可能となる。

2.3 Ninf ライブラリインターフェース

Ninf は、数値データベース並びに数値計算ライブラリのリモートライブラリである。そのインターフェース（サービス）には、次のようなものがある。

1. Ninf リモートライブラリ

プログラムから、リモートコールをすることによって、数値計算ライブラリあるいは数値データベースを呼び出す。Ninf の中心となるサービスを実現するライブラリである。

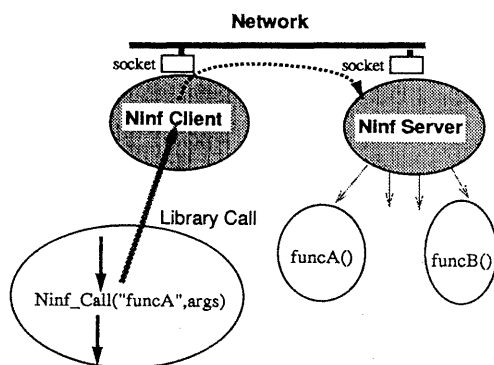


図 2: Ninf: Remote Library Interface

図 2 に Ninf リモートライブラリインターフェースの概要を示す。ユーザはユーザプログラムのコンテキストから Ninf のライブラリ関数 (*NinfCall()*) を呼ぶ。NinfCall はネットワーク接続用のプログラムにリンクされており、その *Ninf_Client* が *Ninf_Server* へ接続を行ない、必要な引数をサーバーに渡す。サーバーでは要求の数表、関数、ライブラリなどを実際に呼びだし、サーバーの計算機において実行した結果を再びクライアント側に戻す。これらの処理をユーザは単に Ninf のライブラリ関数を呼ぶだけで処理される。

2. Ninf ライブラリ ブラウザ・ツール

現在、使用可能なライブラリをブラウズするツール。ライブラリの関数を使う時に、その使いかたを一覧する。簡単なインタプリタ言語を使って、interactive にライブラリを呼び出す。

3. ライブラリドキュメント ならびに ライブラリプログラム配布サービス

Ninf ライブラリのドキュメント並びにプログラムソースコード自体をサービスする。これらについては、既存の ftp ベースのツール、WWW, mosaic などを流用したインターフェースを作成する。

4. プログラム登録サービス

ユーザが自由にライブラリへの登録ができるようなシステムの開発も検討している。ユーザの contribution に関して、快適な環境を支援することは、この種のライブラリが成長していくために必要な鍵となる技術である。contribution を支援する方法として以下の 2 通りが考えられる。登録したいユーザがサーバに対して、プログラム（ライブラリ）を送付し、登録してもらうサービスを提供する。しかし、この方法では整合性のあるように登録するシステムを作成できるか、実際のオブジェクトを作る作業は自動的にやることができるかなど問題点が多い。

一方、Ninf では Contribution したいユーザ自身のマシンで、サーバーを立ち上げてもらうことを推奨したい。各処のサーバーで登録した内容を、他のサーバーにも伝えて、他のサーバにあるようなライブラリでも、自動的に delegate していくようなプロトコルを作成する。こうすることによりクライアントユーザは、自分が呼び出す Ninf オブジェクトがどのサーバーにあるか意識をしなくてもよい。

2.4 性能評価ツールとしての Ninf 応用

Ninf サーバーを様々な計算機システムに設定することの副次的な効果として、計算機システム相互の性能評価が容易にできるようになる。すなわち、Ninf リモートライブラリとして性能評価プログラムを Ninf で記述することによりユーザは関数・ライブラリの呼びだし形式を共通にすることができる。システムライブラリの引数並びやデータ型の整合などはすべて Ninf サーバーが吸収してくれる。このため、サーバーを様々な計算機システムに切替えることが容易である。

例えば、実際のアプリケーション中に用いられている環境の中で、特定のライブラリ部分を Ninf で記述することにより Ninf サーバーが設定してある計算機システムの実行時間や精度が比較できる (図 3)。このように、システムの性能評価においても Ninf は非常に有効なツールとなることが期待されている。

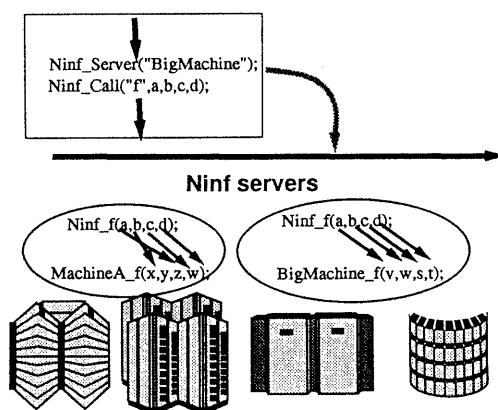


図 3: Ninf: Multiple Server for Performance Evaluation

3 おわりに

超高速ネットワークを前提としたハイパフォーマンスコンピューティングのためのネットワーク数値情報サービス Ninf についてその設計概要を述べた。Ninf はサイバーネットワーク空間におけるハイパフォーマンスコン

ピューティングのための望ましいサービスを提供し、高精度、高信頼性、高速性などの数値解析技術を結集したプログラムを即時利用可能な動態保存として情報発信基地として貢献とサービスを行なう。Ninf を利用することにより、計算の本質でない部分を隠蔽し、ハイパフォーマンスコンピューティングにおいて問題解決に専念できる環境を提供する。

Ninf は現在の低速ネットワークではサーバにアクセスするための時間費用の方が高価で頻繁にアクセスを必要とする場合には実用的ではないが、将来の高帯域、広域通信の実現を前提としたサービス提供を考えている。

現在は データ転送、同期、remote access、数値表現変換、通信路のデータ圧縮等を実現するために、PVM を用いて Ninf の簡易実装を行なっている。これから、効率向上のため、TCP/IP を基本として、socket による直接通信を用いた実装を行なう予定である。

また、検討すべき問題点として、課金やユーザ登録の方法、Security の維持確保、プログラムの異常終了対処、無限ループへの対処などがあり、実際の運用に先だって解決する必要がある。ライブラリの引数の並び等に関しては標準化の動向を踏まえて検討する。

本研究は科技庁原子力特別研究『複雑現象の解明における超高速計算機利用技術の研究』および、電子技術総合研究所とお茶の水女子大学の共同研究契約に基づくものである。日頃より議論をしていただく電子技術総合研究所 太田公廣 情報アーキテクチャ部長、計算機方式研究室各位、ならびにお茶の水女子大学理学部細矢治夫 教授に感謝致します。