

・グローバルコンピューティングへの夢

– メタコンピュータ・Java・リフレクション技術による将来の計算基盤へのいざない –

松岡 聡 (東京工業大学) †

高速なコンピュータネットワークにより、地球上の任意の地域に置かれた計算機に対して遠隔的に計算させることが可能となり、それに基づいて超高性能広域計算を行なう仮想的な「グローバルコンピュータ」が米国を中心に我々の *Ninf* プロジェクトを含め研究が活発化してきている。今回は、グローバルコンピューティングの紹介を行なうとともに、オブジェクト指向・Java・リフレクションがどのようにその構築技術となっているかを論じる。特に、Javaとデザインパターンを活用した *Ninflet* システムと、リフレクションを用いて JIT コンパイラレベルで異機種環境に適合を行なう *OpenJIT* を中心とする。

Towards Global Computing

– Future Computing Platform with Metacomputer, Java, and Reflection –

Satoshi Matsuoka (Tokyo Institute of Technology) †

Fast networks are now making it feasible to remotely delegate computation to any computer in the world, and as a result, research on “Global Computing” is becoming active, including our *Ninf* project. In the talk, we introduce various aspects of global computing, especially the architectural support of object-orientation, Java, and reflection. In particular, we concentrate on our *Ninflet* system, which employs Java and Design Patterns, and also *OpenJIT*, which allow adaptation to heterogeneous computing environment at JIT-compiler level.

1. はじめに

インターネットに代表される世界規模で結ばれた高速なコンピュータネットワークにより、地球上の任意の地域に置かれた計算機に対して遠隔的な計算が可能となった。現状では Java Applet のブラウザ内実行や、CGI による Client-Server 型の処理など、個々のユーザの独立した非常に単純な処理が主であるが、将来はさらなるネットワークの高速化により、多数のスーパーコンピュータを接続して大規模な超広域高性能計算を行なったり、ネットワークに接続された莫大な数の計算機のうち、夜間に遊休な計算機を地球の裏側の昼間の地域に貸し出して有効利用することが可能となる。

このように、キャンパスや組織内のような台数的には数百台から数千台の中規模ネットワークから、インターネットのような地球規模の広域ネットワーク上に、仮想計算環境を構築し、大規模な並列処理を行なえるようなシステムをグローバルコンピュータ(Global Computer)

あるいはメタコンピュータ(Metacomputer)と呼び、最近特に米国の HPC-NGI(Next Generation Internet)関係の研究による研究が盛んになっており、ワークショップやシンポジウムが開催され、実際 Globus[1], Legion[2], Netsolve[3]などのシステムが研究開発されている。

我々は 1995 年より、グローバルコンピューティングのための *Ninf* システム[4]を提案してきた。*Ninf* システムは、サーバー・クライアントモデルに基づいており、*Ninf* サーバ上に予め用意されている計算ルーチンを、クライアントであるユーザプログラムから RPC (Remote Procedure Call) で呼び出すことを実現する。また、計算をサポートする広域データベース機能や、各種言語処理系に対する API や GUI なども兼ね備える。

2. グローバルコンピュータの技術的課題

グローバルコンピュータは、複数の異なる管理化にある計算機群を一つの計算機として仮想化せねばならず、セキュリティや対故障性など、管理上の様々な技術的問題が生じる。これらの技術は従来の分散システム技術としても存在したが、それらに加えてネットワークのレーテンシに抗して高性能を保たなくてはならならず、有効な資源配分、スケジューリング、大幅なレーテンシ

†東京工業大学 情報理工学研究所 数理・計算科学専攻
〒152 東京都目黒区大岡山 2-12-1
matsu@is.titech.ac.jp, http://www.is.titech.ac.jp
Department of Mathematical and Computing Sciences, Tokyo
Institute of Technology

の隠蔽などが分散コンピューティング技術と緻密に連携しシステムが構築されなくてはならない。逆に言えば、現状の Unix や分散 OS, WWW 技術のみでは機能的に不足しており、Globusをはじめ各システムではそれらに関し柔軟でかつロバスタなフレームワークを提供している。我々の Ninf システムでも、超広域高性能計算の資源配分の枠組みを提案し、かつ各種の性能評価・シミュレーションを行なっている[5]。

3. グローバルコンピューティングとオブジェクト指向・Java・リフレクション

グローバルコンピューティングにおいても、オブジェクト指向の諸技術がその構築において有効であると認識されており、様々な試みがなされている。Globus[1]ではセキュリティ、負荷分散、通信などがツールキット化されており、ユーザは必要なツールキットのモジュールを用いてカスタム化することにより、グローバルコンピューティングのアプリケーションを構築する。Legion[2]はより野心的で、システム全体がオブジェクト指向で統一されており、かつメタクラスで組織やアプリケーションによりそれぞれ異なるシステムの各機能の管理ポリシーを導入するという、リフレクションの技術を用いている。また、CORBA をデータ並列通信を行なえるように拡張したり、グローバルコンピューティング用のコンポーネント技術など、米国を中心に盛り上がりを見せている。

さらに最近では、Java を用いた Global Computing 構築技術が、複数する提案されている。Java は I/O、ネットワーク、スレッドなど比較的低位の言語機能から GUI、WWW やコンポーネントフレームワークなど高い機能まで標準化されており、かつ異機種でポータブルな実行が可能で、グローバルコンピューティングの基盤として有用であるとみなされる。特に、セキュリティの面では、リモートファイルやネットワークのアクセスの制限、異なる ThreadGroup のスレッドに対する操作などの可否を制御する `java.lang.SecurityManager` クラス、不正なメモリ番地へのアクセスを起こさないことを保証するオブジェクトポインタや添字チェックされる配列といった言語仕様、不正なコードが含まれていないかチェックする `bytecode verifier` といった、グローバルコンピューティングの基盤となるセキュリティ技術を多々内包している。

現在までに提案されている Java を基盤としたグローバルコンピュータのシステムは、このように Java のセキュリティに関する言語的特徴を活かしたものが多い。これらは Javelin[6]に代表される Applet のセキュリティに基づいたもの、および我々の Ninplet システム[7]のように、独自の計算サーバーをベースとしてセキュリティをカスタム化したもの、に大別される。一見 Applet の方が

一般のブラウザで実行可能など、有利なように思えるが、実はそれぞれ一長一短があり、両者を融合する技術も考えられる。

Java をグローバルコンピューティングに用いた場合、問題なのは性能とそのポータビリティである。通常の Java の実行の高速化には JIT コンパイラが用いられるが、我々の研究ではそれらは性能的改善は進んでいるものの、Fortran, C と比較してまだまだ速度的に十分であるとはいえない[8]。これは単純な速度の問題のみならず、Fortran, C では比較的普遍的な高レベルの最適化が、Java の言語セマンティクスおよび実装の特性によって、それぞれの処理系で効果が大きく異なるという、異機種間の性能の可搬性の問題も含まれている。また、Java 自身、その可搬性は、JVM が提供する機能に限定されており、分散共有メモリ、ユーザレベル通信、スレッドマイグレーションなど、JVM が未サポートな機能は基本的に可搬にかつ効率よく実装するのは非常に困難である。この問題を解決するために、我々は富士通と共同で OpenJIT の研究開発を進めている。OpenJIT はリフレクション技術を用い、世界で初の Java で書かれた可搬性の高い JIT コンパイラである。OpenJIT はクラスフレームワークであるため、実行時にアプリケーションや実行環境に特化したクラスファイルを動的にダウンロードすることを可能としている。これにより、様々なグローバルコンピューティング環境上で、異なる並列プログラミングのパラダイムを効率よくサポートする。

参考文献

- [1] Ian Foster and Carl Kesselman, "Globus: A Metacomputing Infrastructure Toolkit", Proc. International Journal of Supercomputer Applications, 1997.
- [2] A. S. Grimshaw et. al. "Legion: The Next Logical Step Toward a Nationwide Virtual Computer", Tech. Rep. CS-94-21, University of Virginia, 1994.
- [3] H. Casanova and J. Dongarra. "NetSolve: A Network Server for Solving Computational Science Problems", Proc. IEEE Supercomputing '96, San Diego, Nov. 1996.
- [4] S. Sekiguchi, M. Sato, H. Nakada, S. Matsuoka, and U. Nagashima. "- Ninf -: Network based Information Library for Globally High Performance Computing", Parallel Object-Oriented Methods and Applications (POOMA'96), 1996, <http://phase.etl.go.jp/ninf/>
- [5] Atsuko Takefusa, Satoshi Matsuoka, et. al., "Multi-client LAN/WAN Analysis of Ninf: a High-Performance Global Computing System", Proc. IEEE/ACM Supercomputing '97, San Jose, CA, Nov. 1997.
- [6] B. Christiansen, P. Cappello, M. Ionescu, M. Neary, K. Schauer, and D. Wu. "Javelin: Internet-Based Parallel Computing Using Java", Proc. ACM Workshop on Java for Science and Engineering Computation, 1997.
- [7] H. Takagi, S. Matsuoka et. al. "Ninplet: a Migratable Parallel Objects Framework using Java", Proc. ACM

Workshop on Java for Science and Engineering
Computation, Palo Alto, 1998.

- [8] OpenJIT: a Reflective Java JIT Compiler in Java,
<http://openjit.is.titech.ac.jp/>.