

超広域分散数値情報データベース NinfDB の構築について*

3 A a - 2

稲木 貴光

松岡 聡

小川 宏高

東京工業大学 / 富士通 (株)

東京工業大学

東京大学

inagi@is.titech.ac.jp[†]

matsu@is.titech.ac.jp

ogawa@ipl.t.u-tokyo.ac.jp

1 はじめに

科学技術計算では、様々な物理定数や行列データを用いて計算が行われており、計算結果及び計算アルゴリズムの信頼性・正当性の検証という観点からは、それらの値が高精度かつ高速に得られることが求められる。また、これらの数値データは膨大な数に上り、測定技術の向上や定義の変更等によって逐次値が更新されるという性質を持つ。従って、数値情報の管理をユーザの手に委ねることは非常な労力を強いることに他ならない。

我々は広域ネットワーク上で利用できる数値・行列情報データベースシステム NinfDB を実現した。本システムはネットワーク上に分散して設置された数値情報データベースに対して、クライアント数値アプリケーションの中から query を発行し、答えを受け取って計算を行う枠組を提供する。

本稿ではこの NinfDB システムの実装に関して詳細を述べる。

2 NinfDB について

2.1 NinfDB システムの概要

NinfDB は、広域ネットワーク上に分散して配置された数値情報データベースを簡便に利用できる枠組みを提供する。本システムはバックエンドのデータベース管理システム (DBMS)、フロントエンドサーバ、クライアントのいわゆる 3-tier モデルで構成されており、これらの要素は commodity 技術を用いてプラットフォーム可搬に実現されている (図 1)。

また、NinfDB システムは動的データベースとしての機能も有しており、DB のレコードに外部の計算機での計算を必要とするデータを、計算のスキームのみを記述したデータ (Ninf_call 手続き) として格納できる。この「Ninf_call 手続き」データは広域計算システム Ninf[1] の計算サーバに NinfDB サーバから計算要求することで、クライアントには計算結果のみを返すので負担にはならない。

またサーバシステムは、検索結果が行列の場合にクライアントの要求するさまざまな行列形式に応じ、行列データを行列形式トランスレータにて適宜変換することができる。

2.2 ポータビリティ

本システムは分散環境を前提としているため、ポータビリティを維持する必要がある。従って、システム

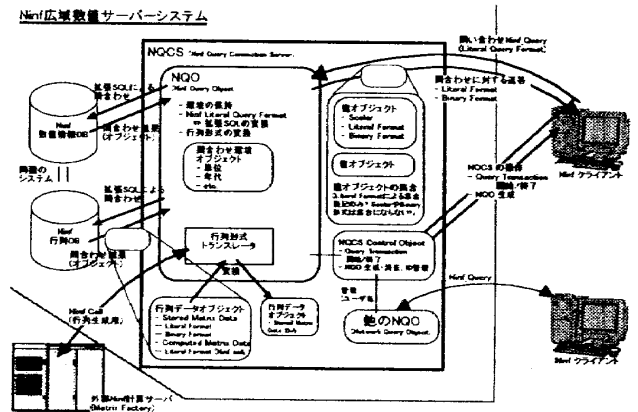


図 1 NinfDB システムの概要

の各層について以下の点を考慮して実装を進めた。

バックエンド DB DBMS の種類を限定しないように、Open Data Base Connectivity (ODBC) をサポートしたものを採用する。

サーバシステム Java 言語および HORB を用いてシステムを実装することでポータビリティを確保。クライアントプログラム クライアントプログラムを記述する言語を制限せずに、文字列リテラルによる問い合わせの発行、結果の取得を可能にする。

3 NinfDB の実装

数値情報データベース及びサービスプロセス、問い合わせのインタフェースの実装について述べる。

3.1 数値情報データベース

科学技術計算で用いられる数値情報としては、値そのもののデータのみでは不十分である [2]。本システムのバックエンド DB では、値自身の他に、値の名前、年代、誤差、有効桁数、単位、単位系列の情報を持たせる。具体的には、DB のレコードの各フィールドを表 1 のように設定した。

値自身のフィールドには数値データの値自身、行列データまたは Ninf_call のための計算スキームを指定できる。以下は、正弦関数 sin の値を Ninf_call を用いて得る例を示したものである。

```
"NinfCall (sin IN double a OUT double x(1))"
```

*Implementation of Global Numerical Information Database Server System "NinfDB"

[†]Takamitsu Inagi, Tokyo Institute of Technology / Fujitsu Limited.

表1 数値情報データベースのレコード

フィールド名	型	説明
VALUE	文字列型	値自身
VALUENAME	文字列型	値の名前
NFROM	整数型	年代の始まり
NTO	整数型	年代の終り
ERROR	実数型	値の誤差
PRECISION	整数型	値の有効桁数
値の単位	ID	値の単位
値の単位系	ID	値の単位系

* 値の単位と単位系に関しては別途テーブルを用意。

3.2 サービスプロセス

フロントエンドサーバなどのサービスプロセスは、Java 言語を用いてオブジェクト指向フレームワークによる実装を行った。特にデータベースへのアクセスに関しては Java Data Base Connectivity (JDBC) を利用した。JDBC は ANSI SQL2 Entry Level をサポートし、標準的な SQL を実行できるよう設計されている [3]。これにより、JDBC-ODBC ブリッジを介し DBMS に対するアクセスを容易に構築することができる。

フロントエンドサーバは、クライアントのトランスアクションごとにプロセスを生成し、検索環境の設定、問い合わせ、結果の応答までを管理する。検索環境には、対象となる DB 名、値の名前など検索の主だった環境を設定する。またフロントエンドサーバには、行列データの形式をクライアントが要求する行列形式に変換できるように行列形式トランスレータを実装した。行列形式トランスレータは、バイナリ形式、密行列表現リテラルおよび疎行列表現リテラルを相互に変換可能である。例としてクライアントが検索結果として疎行列形式の行列を要求した場合、以下のよう形式に変換する。

```
if (resultType is Sparse) {
    // 疎行列リテラルに変換
    String result = toLiteralSparseFormat(dense);
    return result;
}
```

3.3 クライアントインタフェース

クライアントには、一連のトランスアクションを構成するための下記に示すインタフェースが用意されている。

```
NinfQuery_Proxy トランスアクションを開始する。
setenv 検索環境を設定する。
query 検索を行う。
close トランスアクションを終了する。
```

問い合わせは、クライアントプログラムを記述する言語を限定しないために、文字列による形式を採用した。この形式は設定されたキーワードとそれに対する値の組として表現される。例として、「1990年から1995年の光速 c の値を数値情報データベースから」得る場合の一連の手続きを以下に示す。

```
NinfQuery_Proxy nq = new NinfQuery_Proxy(url);
nq.setenv("Database Constant DB Name c");
Fundamental f = nq.query("From 1990 To 1995");
nq.close();
```

ここで Fundamental オブジェクトは、DB に定義されているレコードと等価な情報を含むオブジェクトであり、これを文字列として取り出すメソッドも提供されている。

また、検索結果に関しても文字列による表現を採用した。特に行列に関しては、密行列、疎行列の表現を定義した。以下は 2 行 2 列の単位行列をそれぞれの表現で実装したものである。

密行列表現: "[dense [2 2] 1 0 0 1]"

疎行列表現: "[sparse [2 2] 0 0 1 1 1 1]"

4 まとめと今後の課題

科学技術計算に用いられる数値情報をプログラム中から簡便に利用できるシステムとして NinfDB を構築した。本システムのように広域分散環境での利用を前提としたシステムでは、動作するプラットフォームを限定しないように設計と実装を進める必要がある。今回の実装では、ODBC をサポートする DBMS での DB 構築、Java 言語での実装などにより十分なポータビリティが確保できた。実行性能に関しては、バックエンド DB に対する query に処理時間の大半を占めるため、ネットワーク遅延の問題を除けば、フロントエンドサーバ及びクライアントインタフェースでの処理性能は問題にならない。

また、広域計算システム Ninf では、このような DB システムとの協調によって、常に最新の数値データ・計算アルゴリズムを用いた、ネットワーク透過な数値アプリケーションを構築することが可能である。同種の目的を持つテネシー大の NetSolve [4] では、NinfDB のような DB システムの構築には至っていない。

今後の課題としては、サーバ側でデータをキャッシュすることによる全体的な性能向上等を検討する。

なお、本研究は情報処理振興事業協会 (IPA) の創造的ソフトウェア育成事業「並列・分散処理基盤ソフトウェアの開発」の一環として行われた。

参考文献

- [1] S. Sekiguchi, M. Sato, H. Nakada, S. Matsuoka, U. Nagashima: "Ninf: Network based Information Library for Globally High Performance Computing", *Parallel Object Oriented Methods and Applications*, Santa Fe, 1996.
- [2] 飯岡美恵, 松岡聡, 小川宏高, 関口智嗣, 佐藤三久, 長嶋雲兵: ネットワーク数値情報システム Ninf における広域数値情報データベースシステム, 第 13 回オブジェクト指向計算ワークショップ (WOOC'97), 1997.
- [3] 日本サン・マイクロシステムズ株式会社編: "Java ライブラリ -2 Java プログラミング JDBC", サイエンス社, 1997.
- [4] H. Casanova and J. Dongara: NetSolve: A Network Server for Solving Computational Science Problems, *Proceedings of Super Computing '96*, 1996.