ベンチマークテストによるライブラリ内アルゴリズム選択

1P−12

須崎 有康　田沼 均　平野 聡

電子技術総合研究所

1　はじめに

この研究はソフトウェアが種々の計算機アーキテクチャや処理するデータのパターンに対し、自律的に自己を最適化するメタシステムの研究をアルゴリズム選択中心に考えたものである。ここでは同一処理を行なうアルゴリズムどうしのうち使われる状況に応じて最適なものを選ぶメカニズムについて述べる。例えば最短路問題、文字列検索などアルゴリズムが多種あるもののうち処理するデータごとに最適なものを選び出し、それを実行させることで効率の向上をねらう。

実現法としてはライブラリを同一処理をまとめたインターフェースとし、ライブラリ内の複数のアルゴリズムを引数の特徴によって選択するメタシステムを作る。

このメタシステムはリフレクションの機能[1]と同様にその実行状況によって最適化を行なうものである。しかしリフレクションと異なりコンパイラ、OSなどのシステムに手を加えるのではなく、アプリケーションで使われるアルゴリズムをこれらに適合させることで性能向上をめざすものである。

2　ライブラリ内アルゴリズム選択法

複数のアルゴリズムを選択するメカニズムではインターフェースとしてライブラリを使用する。これはメタシステムの発火時点を規定するためと同一処理ごとにアルゴリズムをまとめるためである。このインターフェースを通してメタシステムが行なうことは、引数の特徴からライブラリ内のアルゴリズムで最適なものを選択することである。

アルゴリズムを選択する評価基準は多種のテストデータパターンによるベンチマークテストからライブラリ内のアルゴリズムごとに実行時間を求め、そのデータの性質と実行時間の関係を解析処理により作成する。これより得られた評価基準を用いてライブラリ呼び出し時にデータの性質からアルゴリズムを選択するメカニズムを作成する。

Algorithm selection in Library by benchmark test
Kuniyasu SUZAKI　Hitoshi TANUMA
Satoshi HIRANO
Electrotechnical Laboratory

ここではベンチマークテストの結果から解析的処理を行なっているが、必ずしも正確な解析ができるわけではない。このため最適アルゴリズムが選択される保証は、種々の検定により与えられるもの以上ではない。この最適アルゴリズムが選択される精度については実験から検証を行なう。

アルゴリズム選択メカニズムの作成とその実行は次の手順で行なう。

1. ベンチマーク

   各ライブラリごとにアルゴリズム解析用のテストデータを用意する。これらは引数の大きさ、引数の数、引数のパターン(圧縮で現されるシャノン流データ量、周波数解析など)で特徴分けする。このテストデータで実行した時の各アルゴリズム計算時間と引数の特徴のデータを相関関係の解析に渡す。

2. 相関関係の解析

   多項式近似と多変量解析によって引数の特徴とベンチマークから得られたアルゴリズムの計算時間の相関を求める。多項式近似では入力サイズなど一変数量のみに注目し、最小2乗法などで計算時間の関係式として求める。多変量解析ではベンチマークテストで使用したデータの特徴と実行時間の相関を次の手法で行なう。

   - 重回帰分析

     データの特徴(X_n)として各アルゴリズムの計算時間(Y)を$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$に近似させ、この式より最適アルゴリズムを推定する。

   - 判別分析

     N個の特徴量をもつデータをN次元空間にプロットし、そのデータでの最適なアルゴリズムごとに群を作る。未知なプロットがいずれの群に属するかを各群との距離によって判別する。

   - ニューロによるパターン学習

     バックプロパゲーションによる多層ニューラルネットの学習アルゴリズムによってデータの特徴と最適アルゴリズムの相関関係を学習させる。

これらのうち重回帰分析では一次式の近似のため精度の問題があり、判別分析では選択メカニズムに適応した場合判別に時間がかかり、ニューロによるパターン学習では学習するために時間がかかるなど、それぞれ問題点をもつ。いずれの解析方法を用いるかはアルゴリズムの特徴と選択メカニズムの戦略によって決定される。

3. アルゴリズム選択メカニズムの作成

多項式近似と多変量解析によって得られた最適アルゴリズムの判定法をもとに引数の特徴から実行時間を予測し、最小の実行時間で実行できるアルゴリズムを選択するメカニズムを作る。選択メカニズムは次章で示すような特徴を持たなければならない。

4. 通常の実行

通常の実行では、ライブラリ呼び出し時に選択メカニズムで最適なアルゴリズムが選ばれる。この選択によりライブラリは高速に実行できる。

新しいアルゴリズムを作成した場合はそれをライブラリに組み込めみ、この一連の処理を行なえば、ライブラリ呼び出し時には新しいアルゴリズムを加えたうち最適なものが自動的に選択される。また特殊なデータパターンがあれば、そのパターンによる選択メカニズムの再構築を行なうことで精密な最適化ができる。

3 選択メカニズム

相関関係の解析結果より作成された選択メカニズムでは、選択メカニズムに使用する時間とアルゴリズム選択によって得られる計算時間のトレードオフを計らなければならない。このトレードオフとなる入力の大きさのしきい値を求め、しきい値より小さい入力では多変量解析による選択メカニズムを使用しない。また、しきい値において各アルゴリズムの計算時間に対する選択メカニズムの時間が大きい場合には、しきい値を大きくしてこの割合がある一定数以下になるようにする。

しきい値より小さい入力のサイズに対しても、入力のサイズによってアルゴリズムごとの計算時間が大幅に変ることが考えられる。この場合、入力のサイズによる多項式近似の解析結果からアルゴリズムの振り分けを行なう。これは多変量解析によるアルゴリズム選択より入力のサイズのみによる振り分けのほうが実行時間がかからないためである。もししきい値以下の入力のサイズに対して選択するアルゴリズムが一意に決まれば、そのアルゴリズムが選択されるようにする。

この選択メカニズムではテストデータにおけるアルゴリズムの振舞いの解析から未知のデータに対する実行時間を推定している。このため最適なアルゴリズムが選択されるとは限らない。しかし、種々の検定法(F検定、χ^2 検定など)を用いてその精度について判定できる。ここではこれらの検定を一定の精度以上で棄却される相関関係を用いることで対処する。

4 実装と実験

現在、対象アルゴリズムを複数文字列の検索とした試作を行なっている。ここで適用しているアルゴリズムは Quadratic 法、Knuth-Morris-Pratt 法、Boyer-Moore 法、Aho-Corasick 法、拡張 Boyer-Moore 法[2] である。はじめの三種類のアルゴリズムは単一文字列のアルゴリズムであるが、複数回適用して複数文字列の検索を行なっている。これらをサンプルデータによるベンチマークにかけ、各々の計算時間を測定した。サンプルデータとしてユーザコマンド用のオンラインマニュアルを使用した。検索文字列としてはこのマニュアル中で使用されている表題を使用した。

多変量解析では引数の特徴として検索対象となるファイルの大きさ (n)、個々の検索文字列長 (m_x) を与える。これらのデータをもとに入力のサイズによる多項式近似、重回帰分析、判別解析、ニューロによるパターン学習を行ない、それぞれをアルゴリズム選択メカニズムに使用する。

検証としてシステムコール用のオンラインマニュアルから同様の文字列検索し、そのアルゴリズム選択の精度を求める。

5 おわりに

このアルゴリズム選択メカニズムは計算機のシステム構成には依存しないので移植性にも優れている。ライブラリを実現するアルゴリズムのソースコードを保持すればアーキテクチャごとにベンチマークを行ない、それに適したアルゴリズム選択が行なわれる。このためこのライブラリを使うアプリケーションは自動的にその計算機にあったアルゴリズムで実行することができる。

今現在は引数のパターンのみに注目して計算機アーキテクチャへの対応は考慮されていないが、今後は種々のアーキテクチャに対応したアルゴリズムを保持するライブラリを作成することで計算機側にも対応していきたい。

参考文献

[1] Pattie Maes: *CONCEPTS AND EXPERIMENTS IN COMPUTATIONAL REFLECTION*, OOPSLA'87,pp147-155 (1987)

[2] 尹, 高木, 牛島: 5 種類のパターンマッチ手法を C 言語の関数で実現する, 日経バイト,1987-7,pp.175-191 (1987)