2J-03　静的なプログラム解析による受信予測の高速化

八幡 義家　板野 修次　足立 涼子　岩本 善行
大津 金光　吉永 努　馬場 敬信
宇都宮大学工学部

1　はじめに

我々の研究室では、並列計算機のメッセージ受信処理を高速化させる手法として、受信メッセージ予測法を提案している。これは、受信処理におけるアイドル時間を利用して到着するメッセージを予測し、それに基づいて受信後の処理を投機的に実行するものである。[1]

メッセージ予測法による高速化のためには、予測的中率の向上と、予測処理のために生じる負荷の軽減が必要である。このことから、本稿では、静的にメッセージの規則性を解析しておき、実行時の予測の際にその解析結果を利用する手法を新たに提案する。これは、プログラムのコンパイル時に、メッセージ受信処理を行うライブラリ関数の名前と引数を順番に取得し、それをマルコフ連鎖に基づく系列とみなしてマルコフテーブルを作成する手法である。そして、実行時の予測の際には、動的なマルコフテーブルの作成を省くことによって高速化が可能となる。あるいは、動的にもマルコフテーブルを作成し、既に静的に作られたものと組み合わせることにより予測的中率を向上させることができる。

この手法は受信処理を行う関数の名前と引数を解析するものであるが、本稿ではその初期評価として、関数の名前のみを対象として静的解析を行い、実際の実行ログと比較することによって本手法の有効性を検証する。評価対象プログラムには、Message Passing Interface(MPI) を用いて記述された各種の NAS Parallel Benchmark(NPB) プログラムを使用する。

2　静的解析の概要

2.1　コンパイル時の処理

図1に、コンパイル時の処理概要を示す。

実際にコンパイルをする前に、ソースプログラムを上から順に解析する。その課程で、アイドルが生じる可能性のある受信処理を含むライブラリ関数(MPIライブラリの場合、MPI_Recv,MPI_Alltoall等)を発見した場合、その関数の名前と引数を取得しておく。解析し終えた後、取得された関数群をマルコフ系列に基づく連鎖とみなし、マルコフテーブルを作成する。

図のマルコフテーブルは、縦軸(数字)を基準とし、横軸(小文字アルファベット)のメッセージが来た回数を示している。関数Aの後には関数Bが1度来るため、1bのマスに1が記録され、関数Bの後に関数Cが1度来るため、2cのマスに1が記録されている。

このとき、プログラム中にループやユーザが定義した手続き、条件分岐といった命令が存在すると、上から順に取得したのでは実際と異なる系列が出来てしまう可能性がある。この解決法については2.3で述べる。

2.2　実行時の予測

プログラムの実行時には、先に作成したテーブルをもとに、メッセージの予測を行う。実行時にA、Bの順にメッセージが到着してその後アイドルが生じたとき、図1のような解析結果である場合、マルコフテーブルはBの後にCというメッセージが最も多く出現していることを示しているので、Cが到着するであろうという予測ができる。

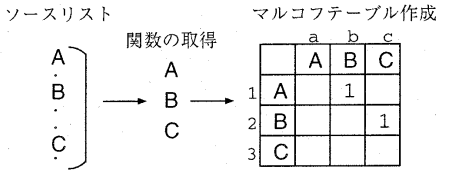

図 1: 静的解析によるマルコフテーブルの作成

2.3　より正確な解析方法

プログラムを解析する際、上から順に記録したのでは実際と異なる系列として解析される場合がある。その解決方針を以下に説明する。

2.3.1　ループ

for文などのループ内に、アイドルが生じる可能性のある受信処理を含むライブラリ関数(以下、解析対象関数と呼ぶ)があった場合を考える。図2において、実際の実行時は解析対象関数AとBが何度か実行されるものとする。しかし、これをループの考慮なしで解析を行うと、関数Bの後に関数Aが実行されることを見落として、関数Bの後は必ず関数Cが実行されるといった解析結果になる。よって、実際は関数Bの後に多く実行されるのはAであるにもかかわらず、何も考慮しない静的解析では、関数Bの後には関数Cが必ず来るとみなされる。これを避けるために、解析対象関数がループ内に存在している場合は、それらの関数をループの回数分だけメッセージ発生系列に追加する。

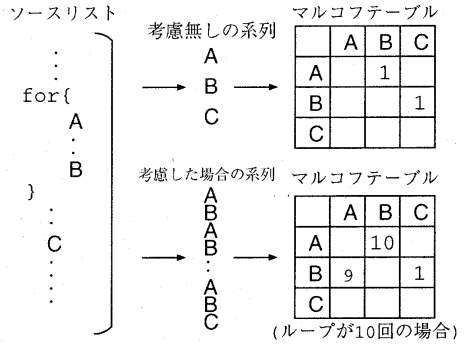

図 2: ループを考慮した場合

2.3.2　ユーザ定義の手続き

ユーザが定義した手続きの中に解析対象関数が存在した場合を考える。図3において、関数Cは、関数Bより先に実行される。しかし、これをユーザが定義した手続きの考慮をせずに解析すると、関数Bが先に実行されるとみなされてしまう。したがって、実際は関数Aの後に関数Cが実行されるはずなのに、何も考慮しない静的解析では関数Bが実行されるとみなされ、実際の系列と一致しなくなる。これを避けるために、あらかじめユーザが定義した手続きとその内部にある解析対象関数を把握

Improvement of Message Receiving Prediction by Static Program Analysis
Yoshiie YAHATA, Shuji ITANO, Ryoko ADACHI, Yoshiyuki IWAMOTO, Kanemitsu OOTSU, Tsutomu YOSHINAGA and Takanobu BABA
Faculty of Engineering, Utsunomiya University

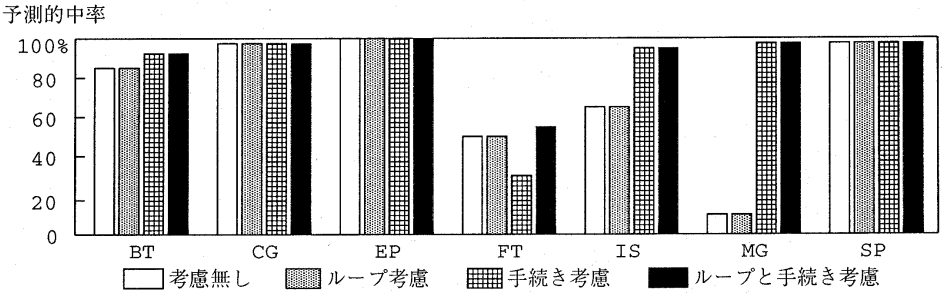

図 4: 静的解析による関数名の予測的中率

しておき、その手続きを呼ぶ文を発見したときに、手続きの内部にある解析対象関数と置き換える。このようにすると正しい系列を取得できる。

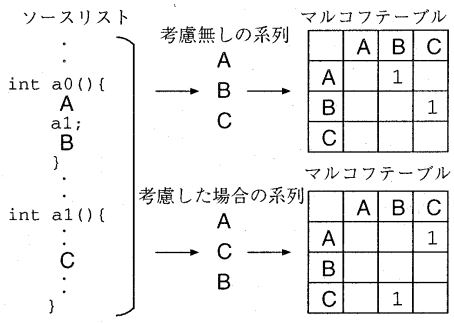

図 3: ユーザ定義の手続きを考慮した場合

2.3.3 条件分岐

ifなどの条件分岐文内に解析対象関数があると、その関数は実行される場合と実行されない場合がある。この場合の解決方法として、条件分岐内の関数が実行された場合と実行されなかった場合の両方の系列をマルコフテーブルに加える、という方法がある。

3 評価

本稿では、以下の場合においての予測的中率について評価する。

a) 何も考慮しない場合

b) ループのみを考慮した場合

c) ユーザ定義の手続きのみを考慮した場合

d) ループと手続きの両方を考慮した場合

評価プログラムには、NAS Parallel Benchmarkプログラムより、BT,CG,EP,FT,IS,MG,SPの7つを用いる。これらのプログラムに対して、解析対象関数の名前について静的解析を行う。その結果を、実行ログに基づいて作成したマルコフテーブルと比較し、一致している部分を予測成功、していない部分を失敗として予測成功率を求める。各アプリケーションの予測成功率を図4に示す。

総評

一部のアプリケーションではa)の場合でも90%を超える的中率を示した。また、a)で低い的中率だったアプリケーションも、c)によってほとんど90%以上となった。さらに、d)では全てのアプリケーションで最も良い的中率を示した。この結果から、d)に基づいて静的に関数を解析すると、実際の処理の流れとよく似た系列を取得することができることがわかった。

アプリケーション毎の考察

CG,SPのプログラムにはある特定の解析対象関数が多く存在し、またその関数は実行回数も多い。そのため、a)でも関数名の予測は容易に可能であった。

EPには、解析対象関数を含むループや手続きが存在しないため、a)でも実行時と同じメッセージ系列を取得することができた。

BT,IS,MGにはユーザ定義の手続きが存在する。このため、a)よりもc)やd)の方が実際に近い系列を取得でき、予測的中率が上昇することがわかった。

FTには、ループ内にユーザ定義の手続きがあり、その手続きから解析対象関数が呼ばれている形となっている。すなわち、図2にて、関数A,Bがユーザ定義の手続きであり、その内部において解析対象関数を呼び出している。このような構造を持つ場合、c)だけでは不十分であり、d)によってこの問題が解決された。

また、FTでは、条件分岐文にもメッセージ受信処理が存在しており、これが現在の静的解析では十分な結果を得られなかった理由である。

4 おわりに

本稿では、メッセージ受信処理を高速化するための手法である受信メッセージ予測法に関して、静的なプログラム解析による高速化の手法を提案し、その具体的方法について述べた。さらに、その手法を利用して関数名の予測について評価を行ったところ、おおむね予測成功率が90%以上であるとの結果を得、本手法の有効性が確認できた。

今後の課題として、プログラムに条件分岐が存在する場合を考慮した静的解析の評価と、関数名だけでなく引数まで解析した場合の予測的中率の評価、が挙げられる。

謝辞

本研究は、一部文部省科学研究費(基盤(B)課題番号10558039, 奨励(A)課題番号11780190), 並列・分散処理研究推進機構の援助による。

参考文献

[1] Y.Iwamoto, K.Ootsu, T.Yoshinaga, and T.Baba: Message Prediction and Speculative Execution of the Reception Process, Proceedings of the Eleventh IASTED International Conference, Parallel and Distributed Computing and Systems(PDCS '99), pp.329-334(1999).

[2] Y.Iwamoto, K.Ooguri, T.Yoshinaga, and T.Baba: A Comparison of Communication Performance in the NEC Cenju-3 and FUJITSU AP1000: Proceedings of the FIRST CENJU WORKSHOP, pp.60-64(1997)