

リサイズなどは移動先毎の環境によって大きく変わる可能性が十分にあるが、プログラム実行中に変化することはない。このように静的に決定できる環境は、プログラムを実行する前にその環境に沿ったプログラムを生成することによって、効率の良いプログラムを実行することができる。そこで、動的コード生成を用いることにより、プログラムを実行する前に実行環境に適した効率の良いプログラムを生成する。

適応性だけを考えると、動的コード生成を使わなくともプロシージャコールによって実現することは可能である。また、エンドユーザに近い、比較的複雑な処理を行わないアプリケーションなどではメリットはほとんどないであろう。しかし、これから移動コードはOSのローダブルカーネルモジュールやネットワークコード等への適用が考えられる。こうした状況では適応性とパフォーマンスが非常に重要となり、動的コード生成による効率の向上や、部分計算による最適化は重要な要素になると考えられる。

4.1 処理系の設計

本研究で提案する動的コード生成を用いた適応的移動コードの利点は以下のとおり。

- 環境に適応したプログラムが生成できる。
- 適切なコードテンプレートをインライン展開することにより、余計な条件分岐などのオーバーヘッドを削除できる。
- 環境に応じて、適切に for ループの展開を行うことによる実行効率の増加。

例えば、for ループを適切に展開することは、ループの条件分岐命令を削除することができるので実行効率が上がることが予想される。しかし、SPARCで採用されているパイプラインとキャッシュの技術は、ループや条件分岐を少ない処理速度の低下で扱うことができるので、ループと条件分岐の除去による効果はほとんど期待できず、場合によっては展開しない場合のほうが実行効率が高い場合がある。こうした場合に、実行環境によって動的コード生成は適切なコードを生成することができる。

5 今後の課題

今後の課題として、コードテンプレートの拡張を行うことがあげられる。コードテンプレートは移動

コードを作るときに一緒に作られている。このコードテンプレートに、コードテンプレートの仕様を情報として付加することにより、ただコンパイルされたコードテンプレート以上の意味を持たせる。この情報を利用してコードテンプレートと生成されるコード間においてさらなる最適化ができると考えられる。また、この付加した情報を用いて、自分自身が生成したテンプレートのみならず、移動先に存在するテンプレートを使うことができるようになると考えられる。移動先に存在するテンプレートが利用できると次の利点が考えられる。

- ファイルシステムや、ネットワーク資源にアクセスするためのセキュリティポリシーが、移動先で決定できる。
- 移動コードがテンプレートを持って移動する必要が無いので、移動コードが小さくなる。

しかし、移動先に必要とする全てのテンプレートが存在するとは限らない。そこで、移動先から比較的近い場所から目的のコードテンプレートを取得するという枠組みを考えて見る。この動作はちょうど Telescript [3] や Agent Space [4] といった Agent System が情報を集める様子と同じであるから、Agent System の枠組みをうまく活用することで実現できるのではないかと考えられる。

参考文献

- [1] Engler, D. R.: VCODE: A Retargetable, Extensible, Very Fast Dynamic Code Generation System, Proc. ACM PLDI '96 pp. 160-170, 1996.
- [2] Dawson R. Engler. et al, 'C: A Language for High-Level, Efficient, and Machine-Independent Dynamic Code Generation, Proc. POPL '96, pp 1-14, 1996.
- [3] James E. White, Telescript Technology-Mobile Agents, 1991, General Magic White Paper, 1996.
- [4] 佐藤一郎.: AgentSpace 高階モバイルエージェントシステム, 電子情報通信学会技術研究報告, Vol97, No627, pp 41-48, 電子情報通信学会, 1998.