
発表概要

コンパイル速度の向上を目的とした非反復型レジスタ割付け手法

小川 健一[†] 片岡 正樹[†] 古関 聰^{††}
小松 秀昭^{††} 深澤 良彰[†]

従来のグラフカラーリングを用いたレジスタ割付け手法は非常に有効であるが、スピルごとに反復しながら干渉グラフを再構築するため、レジスタの少ないアーキテクチャでは計算量が大きくなる。そのため、コンパイル時間が実行時間に含まれるダイナミックコンパイル環境では、レジスタ割付けにグラフカラーリングを適用することは難しかった。スピルごとに干渉グラフを再構築しないナイーブな方式としては、スピル用のレジスタを同命令で一度に使用されるオペランドの最大数だけリザーブするというものがあるが、x86のようなレジスタの少ないアーキテクチャには不向きである。本発表では、スピルごとに干渉グラフの再構築を行わず、かつリザーブされたレジスタを用いない新しい手法を提案する。この手法では、スピルされた変数をリザーブされたレジスタに割り当てるかわりに、スピルされていない変数にすでに割り当てられているレジスタを部分的に割り当てる。この操作は干渉グラフ上でノードの併合として表され、ノードの併合は最もコストが低くなるように行われる。このような方式を用いることで、スピル性能と実行速度の両方が高いレジスタ割付けが実現できる。

A Non-iterative Register Allocation Technique for Improving Compilation Speed

KENICHI OGAWA,[†] MASAKI KATAOKA,[†] AKIRA KOSEKI,^{††}
HIDEAKI KOMATSU^{††} and YOSHIAKI FUKAZAWA[†]

Although the conventional graph-coloring-based register allocation is very effective, due to restructure the interference graph by repeating at each spill, the computational complexity grows in architecture with few registers. Therefore, it is difficult to apply graph coloring to register allocation in dynamic compilation that the compilation time is included in the execution time. There is a naive technique that the interference graph is not restructured at each spill. This technique is to reserve the register for spill for only maximum number of operands used at same time in an instruction. But, it is not adequate to the architecture with few registers like x86 architecture. We introduce a new register allocation technique that doesn't restructure the interference graph at each spill and doesn't use reserved registers. In this technique, instead of allocating the spilled variables to reserved registers, registers that have already been allocated in the non-spilled variables are partially allocated. This operation is shown as coalescing the node on the interference graph, and the node is coalesced so that costs become the lowest. Both the high spill performance and the high speed execution can be achieved by using such this method.

(平成 18 年 1 月 17 日発表)

[†] 早稲田大学大学院理工学研究科
Graduate School of Science Engineering, Waseda University

^{††} 日本 IBM 株式会社東京基礎研究所
Tokyo Research Laboratory, IBM Japan Ltd.