Resumo do artigo Compilador exploração de otimização de espaço de Spyridon Triantafyllis, Manish Vachharajani e David I. August

Fábio Moreira Duarte

Sumário

Resumo

Para encontrar a performace demandada pelas arquiteturas modernas, compiladores encoporam numerosas transformações agressivas de codigo. A maioria das transformações não beneficiam universalmente, compiladores controlam suas aplicações atravez de heuristicas preditivas, julgando um efeito otimizativo na qualidade do codigo. Arquiteturas complexas e iterações de otimização, limitam a precisão do julgamento, levando degradação da performace devido a decisões pobres de otimização.

A perda da performace pode ser evitada atraves de iterações de compilação, explorando muitas opções de otimização e selecionando a melhor. Sistemas de compilações iterativas existentes sofrem de excessivo tempo de compilação e limitado dominio da aplicação. Superando essas limitações, otimização exploração espacial torna a primeira tecnica de compilação iterativa adequada para produção de compiladores de proposito geral. OSE limita o espaço para futuras otimizações. Em tempo de compilação, OSE podas as configurações de otimização restantes na busca por espaco, explorando avaliação de configurações anteriores. Em vez de medir o tempo atual atravez de execuções, OSE compara os resultados de otimização atraves de estimações de performace estatistica.

1 Introdução

Um otimização agressiva é essencial para obtenção de boa performace de processadores modernos. Recursos não uniformes, palelização explicita, multi nivel de hierarquia de memoria, especaluzaçao de suporte, e outras tecnicas de performace avancadas de processadores modernos, somente podem ser exploradas se o compilador efetivamente os focar. A depencia da qualidade do compilador é mais evidente em paralelismo.

Para tais processadores, o processo de otimização tem que balancear um conjunto de fatores de performace, peso de dependencia, pressão dos registradores, utilização de recursos. Antecipar efeitos dinamicos.

Um compilador precisa de numerosas transformações complexas. Como eliminação de codigo morto ou dobradura constante.

Um compilador de otimização deve incorporar um conjunto de otimizações e determinar quando aplica-las. Pode ser obtido atraves de heuristicas preditivas. Um tarefa heuristica é complexa e deve antecipar o efeito das transformações do codigo em todos os passos subsequentes.

Para gerencias tais complicações os compiladores não especificam completamente a heuristica e otimização durante o processo de compilação. Deixam varios parametros de otimização abertos. Tais valores são determinados durante a fase de sintonia, que busca maximizar a performace atraves de um conjunto de aplicações.

Parametrização e sintonia provaram ser bastante efetivas em melhorar a performace de compiladores modernos. Porem é uma resposta imperfeita para as necessidades de otimização. Sintonia pode apenas maximizar a performace atraves de algumas aplicações.

Para endereçar tais limitações da organização tradicional do compilador, compilação iterativa foi proposta. Aplica diferentes configurações de otimização

para cada trecho de codigo. Compara cada versão otimizada e decide qual é melhor. Permite o compilador adaptar-se as necessidades de cada fragmento de codigo. Pesquisas indicam que compilação iterativa pode prover beneficios significativos a performace.

O problema de compilação iterativa é sua natureza de força bruta. Identifica o caminho de otimização correto consideranto todos, ou um grande numero de possiveis caminhos. Aumentando o custo do tempo de compilação. Por isso, compilação iterativa foi limitada a pequenas partes do processo de otimização, pequenas aplicações, ou dominios onde grande tempo de compilação é aceito.

O artigo apresenta otimização por exploração espacial, um metodo de compilação iterativo. Realiza a performace potencial da compilação iterativa enquanto endereça as limitações da aplicabilidade dos metodos existentes. Tornando OSE um metodo iterativo para propositos gerais.

Heuristica preditiva não antecipa todo o impacto nas rotinas de otimização na qualidade do codigo final, ainda codifica informações do comportamento da otimização. Utilizando estas informações o compilador pode realizar escolhas inteligentes do espaço a ser explorado, reduzindo o numero de diferentes configurações de otimização.

Uma parte das configurações de espaço causa ganho de performace modesto.

Em um dado segmento de codigo, a performace de diferentes configurações é correlatada. Permite o compilador utilizar a avaliação para futura exploração espacial em tempo de compilação.

OSE utiliza estimadores de performace.