

Comparação Otimização O0 e O3 - Knapsack

Andrei Massaini

1 Introdução

O problema da mochila, é um clássico problema de otimização combinatória da computação, onde não se sabe de uma solução polinomial para sua respectiva solução. Tal problema é resolvido em tempo pseudo-polinômial com o uso de técnicas de programação dinâmica, por isso o mesmo foi escolhido para compararmos as otimizações realizadas pelo compilador para resolver o problema. Especificamente, comparamos os grafos de fluxo de controle da função `knapsack_seq` compilada sem otimização (O0) e com otimização de alto nível (O3).

2 Metodologia

Examinamos os grafos de fluxo de controle gerados para a função `knapsack_seq` sob dois níveis diferentes de otimização:

- O0: Sem otimização
- O3: Otimização de alto nível

Os CFGs foram visualizados e analisados para identificar diferenças estruturais, mudanças no fluxo de controle e potenciais otimizações de desempenho. Utilizamos ferramentas de análise estática de código para gerar e interpretar os CFGs.

3 Resultados

4 Discussão

A análise dos dois CFGs revela várias diferenças-chave entre os níveis de otimização O0 e O3:

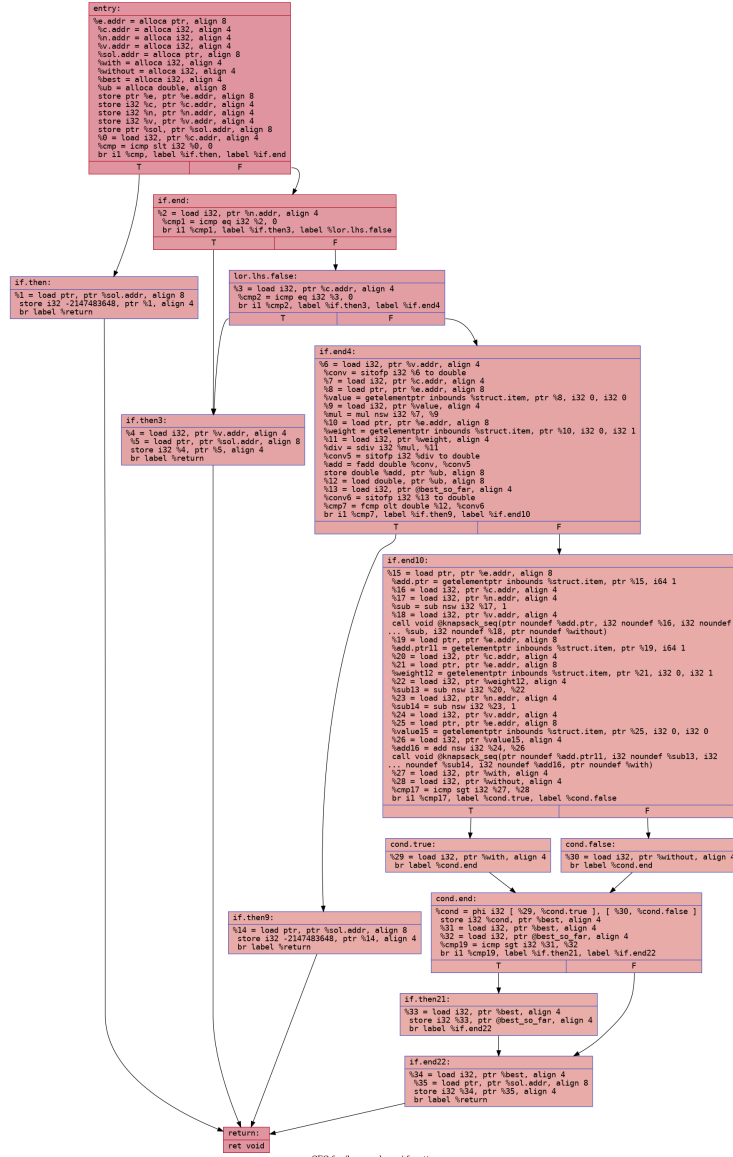
1. **Alocação de Variáveis:** O0 usa instruções `alloca` individuais para cada variável, enquanto O3 reduz o número de alocações, provavelmente favorecendo o uso de registradores. Isso pode levar a um acesso mais rápido às variáveis na versão otimizada.

2. **Complexidade do Fluxo de Controle:** O0 exibe um fluxo de controle mais complexo com numerosos desvios condicionais. O3 simplifica essa estrutura, combinando algumas condições e reduzindo o número de blocos básicos. Essa simplificação pode resultar em melhor desempenho e menor uso de memória.
3. **Chamadas de Função:** O0 mostra chamadas recursivas explícitas para `knapsack_seq`. Essas chamadas parecem ser inline na versão O3, pois não são visíveis no CFG. A inlining pode reduzir a sobrecarga de chamadas de função, potencialmente melhorando o desempenho.
4. **Padrões de Acesso à Memória:** O0 usa mais instruções de carga e armazenamento, enquanto O3 reduz os acessos à memória, potencialmente mantendo mais valores em registradores. Isso pode levar a uma execução mais rápida devido à redução de acessos à memória principal.
5. **Operações Aritméticas:** O3 combina algumas operações aritméticas, reduzindo a contagem geral de instruções em comparação com O0. Isso pode resultar em cálculos mais eficientes.
6. **Nós Phi:** O0 usa mais nós phi para seleção de variáveis em blocos condicionais. O3 reduz esses nós, simplificando o fluxo de controle e potencialmente melhorando o desempenho.
7. **Tratamento de Erros:** O0 tem blocos separados para diferentes condições de erro, enquanto O3 combina alguns tratamentos de erro, reduzindo desvios. Isso pode levar a um código mais compacto e eficiente.
8. **Verificação de Limites:** O3 introduz um cálculo e verificação de limite superior, não presente na versão O0, potencialmente melhorando o desempenho ao terminar precocemente soluções não promissoras. Isso demonstra a capacidade do otimizador de introduzir otimizações específicas do algoritmo.

5 Conclusão

A comparação entre os níveis de otimização O0 e O3 para a implementação do algoritmo da mochila revela diferenças significativas na estrutura do código e nas características potenciais de desempenho. A otimização O3 aplica várias técnicas para melhorar a eficiência, incluindo simplificação do fluxo de controle, redução de acessos à memória e potencial eliminação de computações redundantes.

Essas otimizações provavelmente resultarão em tempos de execução mais rápidos para a versão O3. No entanto, a maior complexidade da estrutura do código otimizado pode apresentar desafios para depuração e manutenção do código. Trabalhos futuros poderiam incluir medições empíricas de desempenho para quantificar o impacto dessas otimizações no tempo de execução e no uso de memória.



CFG for 'knapsack_seq' function

Figura 1: Grafo de Fluxo de Controle da função knapsack_seq com otimização O0

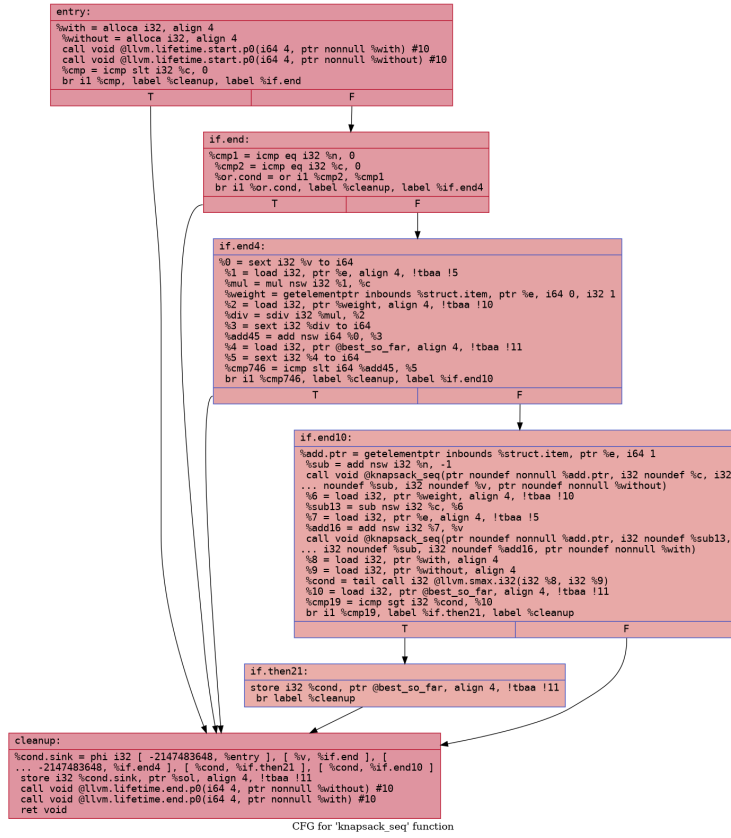


Figura 2: Grafo de Fluxo de Controle da função knapsack_seq com otimização O3