

Algoritmos e Estruturas de Dados I

Prática Investigativa

Aluno: João Pedro Camargo dos Santos

Curso: Ciência da Computação

1º Período

Matrícula: 728780

Turno: Manhã

ARTIGO: Análise Comparativa do Código Gerado por Compiladores Java e C++

O artigo em questão tem como objetivo geral a comparação técnica e estatística entre a performance computacional das linguagens Java e C/C++. Mais especificamente, o artigo divide a sua análise em duas grandes áreas, cada uma possuindo seu próprio compilado de testes e resultado final independente da outra.

A primeira área de estudo do artigo consiste na análise da performance (tempo de uso da CPU) de cada linguagem mediante a compilação e execução de distintos algoritmos, cada um almejando estressar, ou seja, fazer uso intensivo de alguma função e/ou região específica do computador. Os algoritmos utilizados e os respectivos estresses de cada um se encontram em seguida:

- Fibonacci □ Estresse à Pilha de Execução (memória contida no *stack*);
- Busca Sequencial e distintos algoritmos de ordenação □ Estresse à Memória Primária (memória funcional onde todo e qualquer dado que é utilizado pelo código se encontra);
- Cópia de Arquivo □ Estressa à Memória Secundária (memória principal de armazenamento a longo prazo, ou seja, HD, disco rígido, etc.);
- Multiplicação de Matrizes □ Estresse à Unidade Lógica e Aritmética (ULA - região responsável por todas as operações matemáticas realizadas no computador, ou seja, praticamente tudo).

A área de estudo em questão consistiu da 1ª principal parte do artigo como um todo, ela tinha como principal objetivo a comparação estatística (o estudo como um todo fez uso constante de técnicas estatístico-matemáticas para validar seus resultados) dos desempenhos de compilação/execução das duas linguagens, ou seja, a 1ª metade do artigo procurou comparar de maneira técnica qual linguagem apresenta um maior desempenho computacional.

A segunda área de estudo consiste na análise comparativa das atualizações cronológicas dos compiladores de cada linguagem (GCC para C/C++, JDK para Java) em paralelo à comparação do uso de técnicas “manuais” de otimização de cada uma dessas versões de compiladores. Sendo assim, cada linguagem foi comparada mediante 2 parâmetros:

- A versão de seu compilador □ uma mais antiga (5 anos mais velha na época do artigo) e uma atual;
- A otimização do compilador □ de forma não-otimizada (padrão inicial do compilador) ou de forma otimizada pelos autores do artigo.

Ou seja, a área de estudo referente à 2ª principal parte do artigo tem como objetivo a definição da taxa de melhora do desempenho dos compiladores de cada linguagem, comparando as versões de 5 anos atrás e as versões atuais, e comparando o uso, ou ausência dele, de configurações de otimização em tais compiladores.

O artigo foi surpreendentemente de fácil leitura e verdadeiramente me interessou nos pensamentos dos autores sobre os resultados apresentados para os leitores. Apesar de utilizarem termos técnicos, os autores nitidamente se esforçaram para esclarecer vários deles (como os conceitos de estatística e certos conceitos dos algoritmos usados nos testes) e também explicaram e deram pausas para “digestão de conhecimentos” no decorrer do artigo. Simplificando, gostei muito dele!