

Perspective on performance: Uma análise da Coluna 6 de *Programming Pearls*

Manassés Feliciano Paterline

Davi Corradi

Daniel Murad

February 2026

Conteúdo

1	Introdução	2
2	Lei da Gravitação Universal	2
3	Teoria do Caos	2
3.1	Tempo de Lyapunov	2
4	N-Body problem	3
5	Otimização do algortimo	3
5.1	Estrutura de Dados	3
5.2	Redução de Precisão	3
5.3	Melhora da Função Principal	3
5.4	Melhor máquina	3
6	Conclusão	4

1 Introdução

O livro *Programming Pearls* Bentley 2000, escrito por Jon Bentley, possui a divisão feita em colunas, devido ao fato de originalmente ser escrito com o intuito de ser publicado em colunas da revista Communications of the ACM. Posteriormente, juntou-se em uma coletânea, que é o livro guia deste artigo.

2 Lei da Gravitação Universal

Em 1687, Isaac Newton Newton 1687 escreve sobre a lei da gravitação universal em seu livro. Essa lei diz que todos os corpos com massa se atraem mutuamente, e essa atração depende da distância entre eles. Logo, em um sistema com múltiplos corpos, cada corpo sofre influência de todos os outros simultaneamente. Essa descoberta permitiu o cálculo da previsão de movimentação de até dois planetas com exatidão, mas a partir de três, dado tempo o suficiente a análise se perdia em caos.

3 Teoria do Caos

Henry Poincaré, foi um matemático e físico, que tentou encontrar uma fórmula geral analítica para responder essa questão da trajetória de muitos corpos. Ele descobriu que pequenas diferenças nas condições iniciais, produzem trajetórias drasticamente diferentes ao longo do tempo. Foi definido então, que esse sistema se tratava de um sistema imprevisível ou, caótico.

3.1 Tempo de Lyapunov

Uma pergunta surgiu: "Por quanto tempo podemos prever o comportamento de um sistema caótico, antes que os erros cresçam demais?". Essa ideia, como sugere o nome, vem do matemático russo Aleksandr Lyapunov.

Nesse cálculo, é possível calcular o horizonte de previsibilidade, antes de um sistema começar a se tornar caótico e imprevisível. Calcula o tempo em que duas trajetórias que começam quase idênticas se afastam uma da outra, definindo um prazo de validade de uma previsão antes que o caos torne qualquer cálculo inútil.

4 N-Body problem

O problema dos N corpos se dá pela complexidade de se calcular a movimentação de corpos celestes interagindo entre si em um longo período de tempo. No caso de até dois corpos celestes, há uma fórmula matemática exata para isso, como descrito por Newton, porém, a partir de três corpos ou mais, o sistema se torna caótico. Um algoritmo computacional foi desenvolvido para prever a movimentação de corpos celestes. O algoritmo originalmente demorava aproximadamente um ano para ser executado.

5 Otimização do algoritmo

O algoritmo tinha uma complexidade computacional de $O(n^2)$, até que Andrew Appel, fez com que um algoritmo que demorava um ano para ser executado, levasse menos de um dia, com uma melhora de 400 vezes.

5.1 Estrutura de Dados

Appel alterou a estrutura em que os dados estavam sendo armazenados, que era um array simples, fazendo com que fosse armazenado em uma estrutura de dados, o que melhorou drasticamente a performance do programa, fazendo com que corpos individuais, agora fossem analisados em conjunto se estivessem longe o suficiente e calculados individualmente se estivessem próximo o bastante.

5.2 Redução de Precisão

Também alterou a precisão das variáveis, fazendo com que os *doubles* virassem *floats*, trocando 64 bits por 32 bits, reduzindo o uso de memória.

5.3 Melhora da Função Principal

Appel percebeu que uma função sozinha, era executada 98% do tempo do programa, então ele reescreveu essa mesma função em *assembly*.

5.4 Melhor máquina

Ao final, um computador melhor foi comprado, custando cerca de um quarto de milhão de dólares, para rodar o algoritmo com a maior eficiência.

6 Conclusão

A análise do problema dos n corpos e das otimizações implementadas por Andrew Appel demonstra que a performance computacional não é fruto de uma única variável mas de uma abordagem multidimensional. Fazer análises reais, sobre valores apontados nos algoritmos podem levar a compreensão do que pode-se otimizar a fim de obter uma melhora significativa com o menor custo. Um algoritmo com tempo de execução de um ano, se transformou em uma tarefa de menos de um dia.

Conclui-se que a otimização eficaz exige uma compreensão profunda tanto do domínio do problema quanto das limitações e capacidade do hardware. O caso de estudo de Programming Pearls reitera que o refinamento algorítmico, muitas vezes, precede e potencializa o ganho obtido pelo investimento tanto do hardware quanto do entendimento do comportamento de sistemas mais complexos.

Referências

Bentley, Jon (2000). *Programming Pearls*. 2^a ed. Addison-Wesley. ISBN: 9780201657883.
Newton, Isaac (1687). *Philosophiæ Naturalis Principia Mathematica*. Royal Society.