

Para saber mais: Programas cache-friendly

Algumas linguagens que oferecem um controle maior na alocação de memória, como C e C++, podem ter o tempo de execução de seus programas influenciados pela memória cache. Ou seja, só mudando a ordem de execução de alguns trechos de código, ele pode ficar bem rápido ou lento. Programas que conseguem otimizar o máximo isso são chamados de *cache-friendly* ou bons para o cache.

Para ver isso funcionando na prática, tente executar o [programa escrito em C++ aqui](#) (<http://cpp.sh/5ery6>) ou acesse o [código no GitHub](#) (<https://gist.github.com/AndrewIjano/5a46dfd50693dd2d198266b0f864b738>) e execute [aqui](#) (<http://cpp.sh/>).

Você vai ver que o tempo de execução sem cache é muito maior:

options	compilation	execution
		Tempo com cache: 82 ms
		Tempo sem cache: 1426 ms

Isso acontece porque essa segunda execução não é feita de forma linear. Assim, quando o computador for copiar um bloco de dados para o cache, o próximo elemento da nossa lista nunca vai estar lá.

Para visualizar o que está acontecendo, a animação abaixo ilustra a execução de cada um, caso tivéssemos uma lista de 8 elementos.

Execução 1 (boa para o cache)**Execução 2 (ruim para o cache)**

De qualquer forma, não são todas as implementações de linguagens que precisam desse cuidado com o cache. O Node.js, uma implementação de JavaScript, faz várias otimizações que quase mascaram esse tipo de coisa. Mesmo assim, é algo importante para se ter em mente ao longo de uma carreira de programação.