

Aula Prática 10

Lucas Albano Olive Cruz | 2022036209

1. Plano de experimentos:

- Manter o tamanho do registro em bytes fixo e variar a ordem m da árvore.
- Manter a ordem m da árvore fixa e variar o tamanho do registro em bytes.
- Variar o tamanho do registro em bytes e a ordem m da árvore simultaneamente.

2. Justificativas:

- Ao aumentar o valor de " m ", a árvore B terá nós maiores e mais dados serão armazenados em cada nó. Isso pode aumentar a localidade de referência espacial, já que os dados adjacentes serão acessados em conjunto. No entanto, se o valor de " m " for muito grande, a árvore B terá menos níveis e isso pode afetar a localidade de referência temporal, já que cada nível pode precisar ser acessado com mais frequência.
- Ao aumentar o tamanho do registro em bytes, cada nó da árvore B armazenará mais dados. Isso pode aumentar a localidade de referência espacial, pois mais dados relacionados serão acessados em conjunto. No entanto, se o tamanho do registro for muito grande, pode haver um aumento no uso da memória, o que pode levar ao uso do swap e impactar negativamente o desempenho.
- A variação simultânea da ordem m e do tamanho do registro em bytes permitirá avaliar como esses parâmetros interagem entre si. Diferentes combinações podem levar a diferentes compromissos entre localidade de referência temporal e espacial. Por exemplo, uma árvore B com um valor de " m " maior e um tamanho de registro maior pode ter uma melhor localidade de referência espacial, mas um pior uso de memória.

3. Experimentos:

Fiz algumas modificações no código para medir o tempo de execução e utilizei o valgrind para medir as perdas de memória e instruções de leitura e escrita do programa.

Ordem m	Carga	Tempo Exec (s)	Leitura	Escrita	Perda Mem (bytes)
2	100	0.001253	2,298,233	1,009,430	496
3	100	0.000998	2,349,682	1,013,915	736
5	100	0.001016	2,376,953	952,344	1,216
10	100	0.001042	2,796,253	1,086,130	2,416
20	100	0.001251	3,722,904	1,374,417	4,816
5	200	0.001084	2,756,993	1,332,373	2,176

5	500	0.001260	3,929,291	2,504,733	5,216
5	1000	0.001558	5,892,672	4,468,214	10,176
5	10000	0.007722	316,478,495	316,351,816	100,176
2	10	0.000943	1,926,157	637,328	144
20	1000	0.002484	10,443,064	8,094,580	40,656
200	500	0.002300	9,275,065	5,747,863	208,016
200	2000	0.004185	24,749,679	21,222,477	806,416
2	2000	0.001999	9,399,608	8,101,919	8,080
200	100	0.001522	1,606,032	551,552	48,016

4. Análise dos resultados:

- Variação somente da ordem m:

À medida que aumentamos a ordem da árvore o tempo de execução se mantém praticamente constante enquanto a quantidade de leituras aumenta lentamente. A escrita tem um comportamento anômalo para $m=5$, para demais valores o parâmetro também cresce lentamente. A perda de memória segue um comportamento linear com o aumento da ordem.

- Variação somente da carga:

Escolhi fixar o m em 5 devido ao comportamento anômalo no parâmetro de escrita. O tempo de execução cresce lentamente de forma não linear enquanto a perda de memória cresce de maneira linear com comportamento anômalo com carga=200. A leitura aparenta crescer de maneira exponencial com um aumento considerável em altos valores de carga. A escrita acompanha a leitura em termos de crescimento mas se mantém abaixo desta.

- Variação conjunta dos parâmetros:

Aqui fui fazendo alguns testes aleatórios como manter ambos parâmetros baixos ou altos, e algumas outras variações. percebi que manter a ordem m da árvore baixa melhora consideravelmente a perda de memória, provavelmente devido a problemas de liberação de memória nos nós. A carga parece também possuir maior impacto no tempo de execução. Não consegui notar um padrão para os demais parâmetros, no entanto o caso $m=200$ carga=2000 causou diversos erros de alocação de memória e causou uma perda de memória gigantesca, maior até que valores superiores de carga.