Universidade do Estado de Santa Catarina

Projeto de Arquivos

ORDENAÇÃO EXTERNA PELO METÓDO - SELEÇÃO POR SUBSTITUIÇÃO E A INTERCALAÇÃO DE N CAMINHOS

Grupo
Alexandre Maros
Mateus Boiani

Prof. Responsável
Wesley Bezerra

28 de maio de 2015

Sumário

Introdução

Muitas vezes nos deparamos com quantidades finitas de dados desorganizadas em nossas estruturas, e precisamos organizá-las. Por sorte, quando todos os dados de nossas estruturas cabem na memória fica mais fácil realizar está tarefa. Conhecemos vários algoritmos clássicos para isso, Quick Sort, Bubble Sort, Merge Sort, entre outros.

Mas quando a quantidade de dados que estamos manipulando e precisamos ordenar não cabe na nossa memória, como fazemos para alcançar nosso objetivo? Adicionamos mais memória no computador para executar esta tarefa parece ser uma resposta bastante trivial, não é? A verdade é que esse problema é mais antigo do que imaginamos, e muitos cientistas da computação criaram soluções que resolvem esse problema.

Vamos aqui explorar apenas uma das soluções existentes, ver quanto tempo o algoritmo leva para organizar essas informações e analisar esses resultados.

Objetivos

O objetivo desse trabalho é apresentar o algoritmo de ordenação externa utilizando o método da seleção por substituição.

Apresentar de forma didática e intuitiva o funcionamento do algoritmo já citado demonstrando como é possível organizar um arquivo de dados que não podem ser carregados simultaneamente na memória para que seja feita a ordenação dos mesmos, também demonstraremos o tempo que este mesmo algoritmo leva em média para ordenar um arquivo de tamanho definido a seguir.

Problemática

Ao introduzir uma linguagem de programação é comum nos ensinarem a organizar conjuntos de dados (um conjunto de números inteiros, float, double, etc;). Esses conjuntos de dados podem ser mantidos em uma estrutura de dados que conhecemos como vetores e/ou matrizes, esses vetores e matrizes são blocos de memórias sequências que são responsáveis por facilitar a manipulação e acesso a informação que desejamos manipular.

Quando precisamos organizar esse conjunto de dados em ordem crescente, decrescente ou seguindo algum padrão específico é comum utilizarmos algoritmos de organização como o bubble sort, por exemplo. Quando trabalhos com um conjunto de dados que cabe na memória e queremos organiza-lo denominamos Classificação Interna.

Os dados que não são suportados na memória e que queremos organizar chamamos de Classificação Externa. E para organizar e manipular esses dados foram criados diversos algoritmos. Veremos a seguir o método de substituição por seleção e a intercalção de n caminhos.

1.1 Classificação Externa

Os dados que queremos organizar estão salvos em um arquivo, a classificação externa que utilizamos para organizar estes dados pode utilizar esse arquivo original ou um arquivo auxiliar, e consiste de dois passos.

Classificação: São geradas partições(arquivos) de n registros ordenados.

Intercalação: É a transformação das partições criadas na classificação em uma única partição ordenada com todos os registros originais.

Tanto a Classificação quanto a Intercalação podem ser feitas por métodos variados métodos, aqui abordaremos a classificação utilizando seleção por substituição, e a intercalação de n caminhos.

1.1.1 Seleção por Substituição

A seleção por substituição consiste basicamente da leitura do máximo de registros do arquivo original para memória, (1) selecionar o menor registro, gravar na partição de saída. Após gravar na partição de saída substituir o registro gravado pelo próximo registro do arquivo original, (2) selecionar o menor registro novamente, se o menor registro for menor que o último registro escrito no arquivo de saída então congelar este registro e ignorar no restante do processamento. Enquanto existirem registros não congelados executar a etapa (2) sucessivamente, caso contrário, fecha a partição de saída atual, descongela os registros congelados na memória, abrir nova partição de saída e executar a partir da primeira etapa novamente.

Partições Obtidas por Seleção com Substituição

	ÁREA DE TRABALHO						PARTIÇÕES OBTIDAS													
Registros	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3ª substituição						20															
2ª substituição	10	18				74															
1ª substituição	46	48	4	26	56	7															
Memória	29	14	76	75	59	6	6	7	14	29	46	48	59	74	75	76					
							A 1ª partição ficou com 10 registros														
2ª substituição	19	16	11			15															
1ª substituição	65	22	21	8	5	49															
Memória	10	18	4	26	56	20	4	10	18	20	21	22	26	49	56	65					
							A 2ª partição ficou com 10 registros														
3ª substituição	43																				
2ª substituição	78	9	12	17	30	54															
1ª substituição	77	57	25	55	50	66															
Memória	19	16	11	8	5	15	5	8	11	15	16	19	25	50	55	57	66	77	78		
							A 3ª partição ficou com 13 registros														
3ª substituição		60																			
2ª substituição	36	73	27	13	3																
1ª substituição	79	38	51	32	58	1															
Memória	43	9	12	17	30	54	9	12	17	30	32	38	43	51	54	58	73	79			
							A 4ª partição ficou com 12 registros														
1ª substituição			80	31	47																
Memória	36	60	27	13	3	1	1	3	13	27	31	36	47	60	80						
							A 5ª partição ficou com 9 registros														

Legenda

Registros — Divisão de regiões na tabela

Figura 1.1: Seleção por Substituição

Fonte: FERRAZ, Inhaúma Neves - Programação com arquivos pg. 69

Conforme a imagem acima, podemos observar o comportamento do método de seleção por substituição. Os registros que contém fundo escurecido são os registros que ficaram congelados e que serão utilizados no arquivo subsequente. Podemos observar que nesta etapa pode ocorrer a criação de inúmeros arquivos, e o processo só irá terminar quando não tivermos mais nenhum registro na memória e nenhum registro congelado.

1.1.2 Balanceamento de N Caminhos

Como vimos anteriormente a intercalação é a transformação das partições criadas na classificação em uma única partição ordenada. Para isso vamos utilizar o método de balanceamento de N caminhos.

Esse método consiste na união dos arquivos gerados na classificação. São N etapas unindo arquivos em duplas, ou seja, cada dois arquivos gerados na classificação um novo arquivo é gerado a partir da união. Para isso é feito a leitura registro a registro e comparando ambos. Como os arquivos estão ordenados já, então é só comparar e ir salvando um por um no novo arquivo. Ao final das N etapas teremos um arquivo final com todos os registros ordenados. Você deve estar se perguntando, mas se eu tiver um número ímpar de arquivos gerados na classificação? Isso realmente pode acontecer, para resolver isso, costumamos pegar o arquivo que sobrou na enésima etapa e unir com o primeiro arquivo gerado da etapa seguinte.

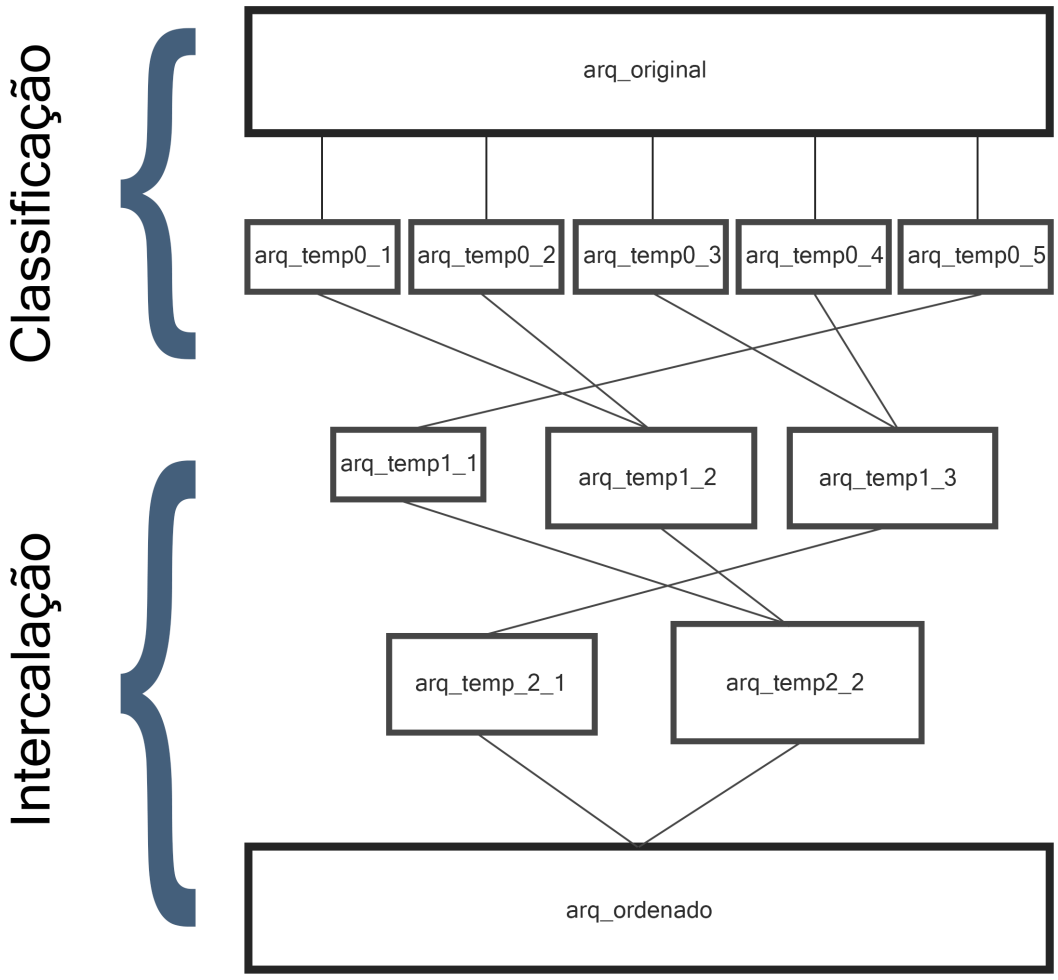


Figura 1.2: Balanceamento de N Caminhos

A ilustração acima retrata exatamente o que acontece ao intercalação, buscamos os N arquivos criados na classifição e realizamos a união em pares. Como a classificação neste caso gerou uma quantidade ímpar de arquivos, pegamos esse arquivo para utilização inicial na iteração seguinte. Isso é feito para evitar que um arquivo de tamanho muito pequeno seja mantido até o final das iterações. Neste caso garantimos que 2 arquivos de tamanhos semelhantes cheguem para ultima iteração.

Experimento

2.2 Informações Técnicas

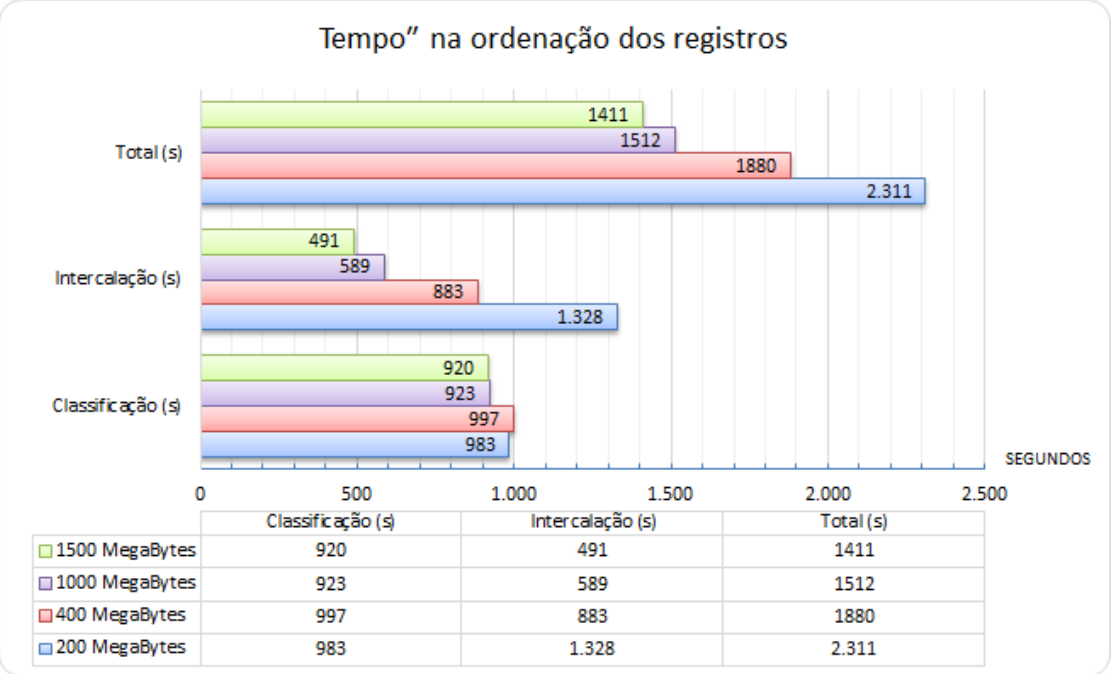
Rodamos os algoritmos anteriormente apresentados, e vamos mostrar os resultados obtidos. Sobre a máquina:

Disco Rígido: Seagate ST1000DM003-1ER162 7200rpm;
Processador: Intel Core i7-4790K CPU @ 4.00Ghz;
Memória: 8,00 GB 1600Mhz;
Sistema Operacional: Windows 8.1 Pro;

Sobre o arquivo a ser ordenado e o algoritmo:
Estrutura utilizada: Estrutura denominada medição, contendo índice, temperatura variando de 0 a 50 graus, dia, mês, ano, sendo todos os campos do tipo long long (8 bytes);
Tamanho de cada medição: 40 bytes;
Quantidade de medições: 322.122,547;
Tamanho total do arquivo: 12,0 GB = 12.884.901.888 bytes;
Linguagem em que o algoritmo foi escrito: Linguagem C;
Forma de ordenação: Crescente, ordenando a temperatura;

2.3 Resultados

Tempo para gerar o arquivo inicial em média: 98 segundos;



	Classificação (s)	Intercalação (s)	Total (s)
1500 MegaBytes	920	491	1411
1000 MegaBytes	923	589	1512
400 MegaBytes	997	883	1880
200 MegaBytes	983	1.328	2.311

Figura 2.3: Tempo para Ordenar 12GB dividindo em blocos de N MegaBytes carregados na memória.

Obs.: Esses valores são aproximações.

2.3.1 Análise dos Resultados

No gráfico acima, tem-se os resultados dos experimentos realizados em cima do algorítmo de ordenação. Temos três grandes sessões, uma mostrando o tempo total do algorítmo, uma mostrando somente o tempo necessário para a etapa da intercalação e por fim, a ultima sessão mostra o tempo de classificação. Os testes foram realizados alterando o tamanho do bloco carregado na memória (1500, 1000, 400 e 200 MegaBytes) em um deteminado instante e o tempo foi medido em segundos.

Tempo de intercalação: Aqui nota-se uma discrepância nos tempos de cada bloco. O maior bloco, 1500 MB, realizou a operação de intercalação em aproximadamente 8 minutos, ja quando carregamos apenas um bloco de 200 MB, seu tempo subiu para 22 minutos. Isso ocorre devido ao fato de que, quando um bloco maior é carregado na memória, este gera menos arquivos na etapa de classificação, logo, o Sistema Operacional realiza menos operações para abrir e fechar arquivos, ocasionando num tempo total muito menor.

Tempo de classificação: Na classificação, o tempo se manteve estável e não se diferenciou muito com o aumento dos blocos. Embora ele necessite criar menos arquivos quando há um bloco maior na memória, ainda assim, ele trabalha com menos blocos que a etapa de intercalação.

Para se ter um desempenho maior no algorítmo, um maior bloco deve ser utilizado em sua execução visando a criação da menor quantidade de arquivos possíveis. É necessário ter um cuidado especial na seleção do bloco, pois precisa-se levar em consideração a quantidade máxima de memória que o sistema operaciona consegue ceder sem que ele mesmo seja prejudicado devido a swap e outros fatores.

Conclusão

Em geral, podemos ver que é possível ordenar arquivos grandes o suficiente tal que a memória não consiga comportar. Para isso dispomos de vários métodos de classificação e intercalação, aqui utilizamos a classificação pelo método de seleção por substituição e o balanceamento de n caminhos.

O tempo total do ordenamento se dá por diversos fatores, sendo eles tempo de escrita e leitura do disco, tempo de processamento das informações na memória, criação de novos arquivos, abertura de arquivo, alocação de informação, são alguns exemplos. Escrever no disco rígido leva mais tempo do que indexar e trabalhar com registros na memória, vimos que com um arquivo muito grande, não é possivel carregar todos os registros na memória, ordenar, e salvar novamente, para que essa tarefa seja possível é necessário muitas operações de escrita e leitura do disco.

Abordamos aqui um dos métodos clássicos para ordenar uma quantidade enorme de dados, atualmente temos uma infinidade de outros métodos para realizar este tipo de tarefa.