Algoritmo external merge sort e análise de sua performance

Thainá Simões Pires – Cefet-RJ

1. Introdução

Este relatório tem como objetivo explicar o que é o external merge sort, apresentar os algoritmos desenvolvidos a partir das aulas da disciplina de organização e estrutura de arquivos e fazer uma avaliação do desempenho do algoritmo.

2. External merge sort

Algoritmos de ordenação são aqueles que ordenam uma dada sequência de elementos. Cada tipo de algoritmo utiliza uma técnica diferente para ordenar, podendo ser mais ou menos eficiente.

O algoritmo merge sort é um desses algoritmos de ordenação, e utiliza a técnica dividir para conquistar. Na fase de divisão o problema é dividido em vários sub problemas e na fase de conquista ocorre a junção dos subproblemas através da recursividade, para que no final o problema esteja devidamente ordenado.

O External merge sort é um algoritmo aconselhável para quando a quantidade de dados a ser ordenada é muito grande a ponto de não caber totalmente na memória ou demorar uma quantidade de tempo significativa. O mesmo é um algoritmo com estratégia hibrida pois é composto por duas fases:

1) **Fase de ordenação.**
2) **Fase de junção (merge).**

Apesar de não ser o melhor algoritmo para ordenar, ainda assim sua utilização minimiza o número de acessos e melhora a performance da ordenação.

2.1. Fase de ordenação

Fase de ordenação em memória. Nesta fase o arquivo é dividido em N blocos de X linhas, onde N é a quantidade de blocos e X é a quantidade de linhas escolhidas pelo usuário

1. Trazer um bloco para memória (Ou seja, X linhas).
2. Fazer a ordenação deste bloco.
3. Escrever o resultado da ordenação em outro arquivo.
4. Repetir o procedimento até que todos os N blocos estejam ordenados individualmente.

2.2. Fase de junção (merge)

Fase de junção (merge) dos blocos criados na fase anterior.

1. Juntar de dois em dois blocos e ordená-los.
2. Repetir o procedimento recursivamente até que resulte em um único bloco do tamanho do arquivo e que esteja totalmente ordenado.

3. Algoritmos

Para a realização deste relatório o external merge sort foi escrito de duas formas diferentes:

1) **Primeira forma:** Utilizando somente um arquivo auxiliar para ordenar.
2) **Segunda forma:** Dividindo o arquivo principal em uma quantidade específica de sub arquivos para que facilite a ordenação.

Para os dois algoritmos, o arquivo escolhido para teste foi o arquivo de CEPs disponibilizado em sala de aula. O arquivo contém uma lista de endereços de todo o brasil, totalizando 699.307 registros. A ordenação foi feita através da coluna do CEP (coluna 5).

3.1. External merge sort utilizando um arquivo.

Na fase de ordenação desta primeira forma do algoritmo foi utilizada a estratégia de leitura da quantidade de linhas escolhida pelo usuário e escrita das mesmas sempre no final de um arquivo auxiliar. Ao final, teremos um arquivo que será composto dos blocos individualmente ordenados. Podemos observar através da imagem abaixo uma análise desta fase, que foi obtida utilizando somente 100 linhas do arquivo de ceps, com blocos de 10 linhas cada.

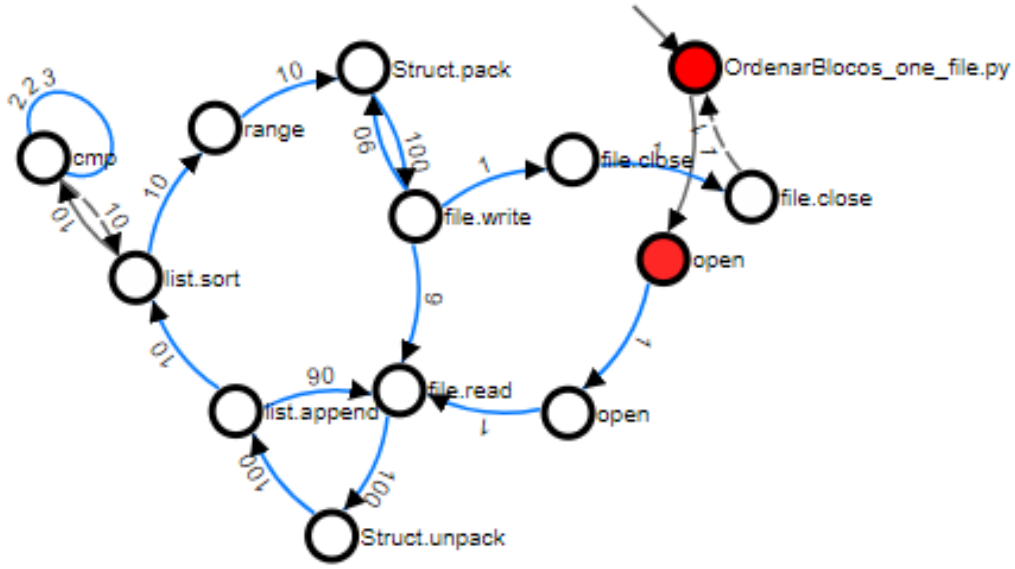


Figura 1 – Captura de proveniência obtida pelo noWorkflow.

Já na fase de junção (merge), a estratégia utilizada foi de fazer as comparações de dois em dois blocos e escreve-las em um novo arquivo. Assim que o primeiro ciclo de comparações é finalizado, o arquivo antigo é deletado e o novo é renomeado. Estas operações ocorrem até que o arquivo esteja totalmente ordenado. Podemos observar esta fase através da seguinte figura:

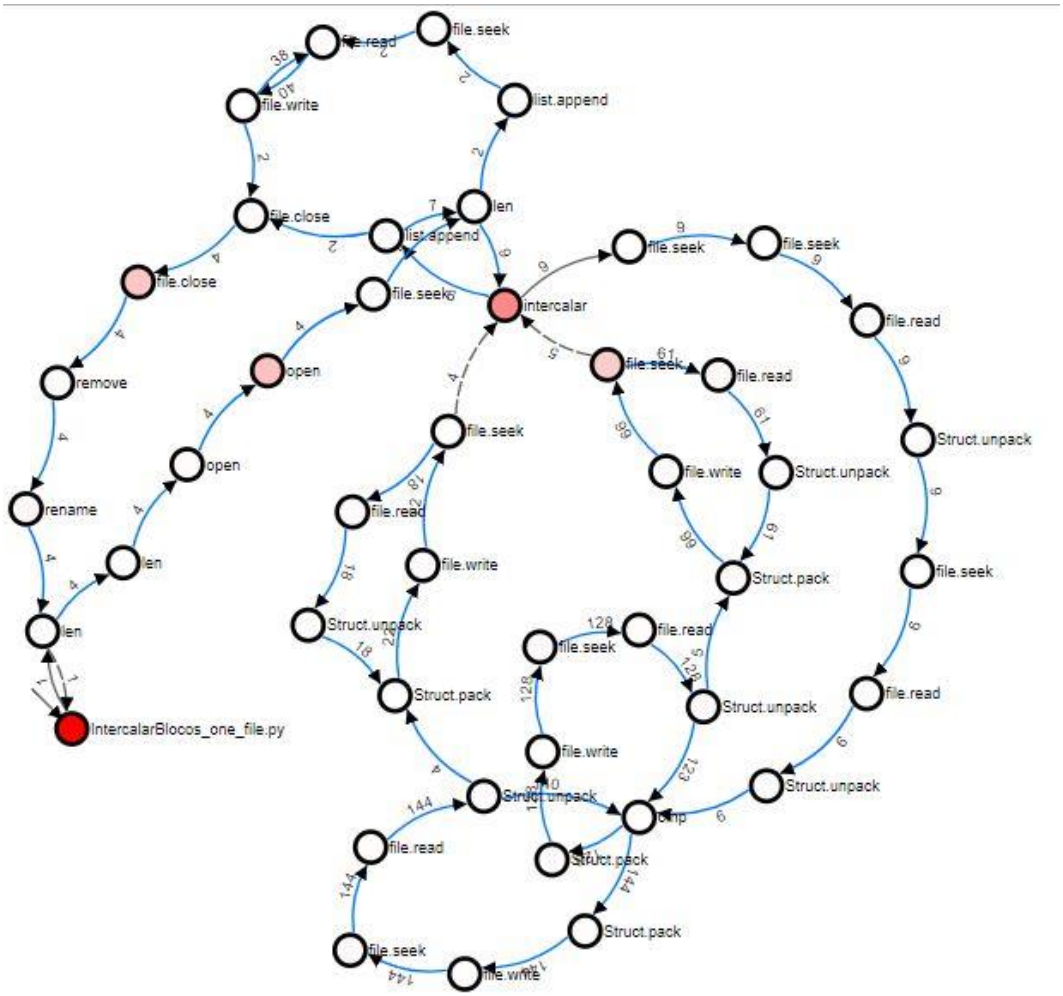


Figura 2 – Captura de proveniência obtida pelo noWorkflow.

3.2. External merge sort utilizando sub arquivos.

A diferença da fase de ordenação deste algoritmo para a do anterior é que ao invés de escrever os blocos ordenados em somente um arquivo auxiliar, cada bloco é escrito em um sub arquivo separado. Portanto, no final da execução desta fase, obteremos N arquivos que representam a quantidade de blocos. Segue abaixo a imagem desta fase:

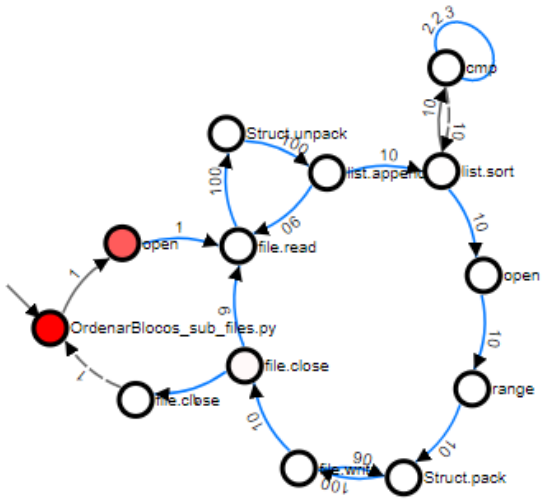


Figura 3 – Captura de proveniência obtida pelo noWorkflow.

Na fase de junção (merge), é seguido o mesmo princípio de um arquivo para cada bloco ordenado. O ponto principal da estratégia de subarquivos nesta fase é que a quantidade de sub arquivos vai diminuindo cada vez mais, pois os blocos vão sendo reunidos de dois em dois. O processo continua até que seja obtido somente um arquivo que esteja totalmente ordenado. A imagem abaixo ilustra esta fase:

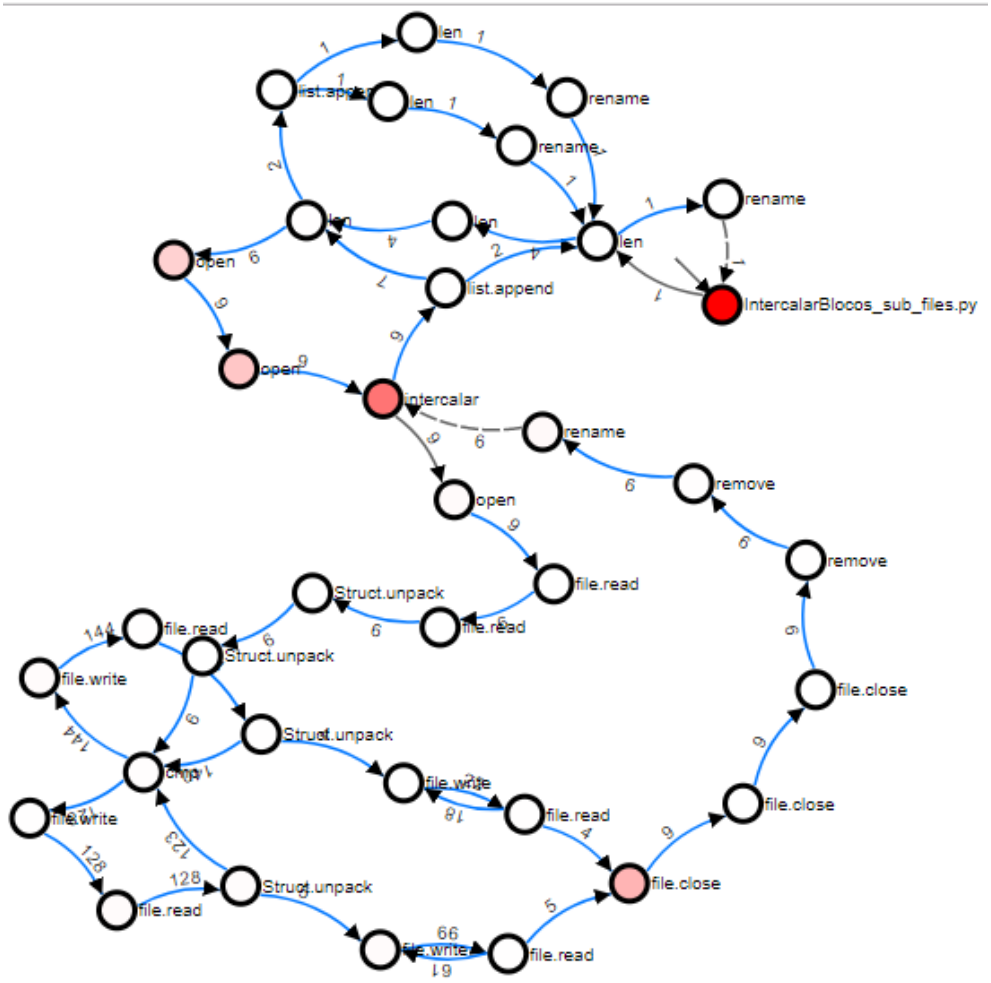


Figura 4 – Captura de proveniência obtida pelo noWorkflow.

4. Resultado da comparação dos dois algoritmos

Para fazer a comparação das duas formas do algoritmo também foi utilizado o arquivo dos CEPs porém agora com todos os 699307 registros. Os algoritmos foram executados em um PC com um processador intel core i5-3470 CPU @3.20GHz x4 com 7,7GiB de RAM e sistema Ubuntu 16.04 LTS.

É importante ressaltar que os seguintes gráficos não seguem um padrão linear.

Ao fazer ordenações do arquivo com diferentes tamanhos para os blocos, foram obtidos os seguintes resultados:

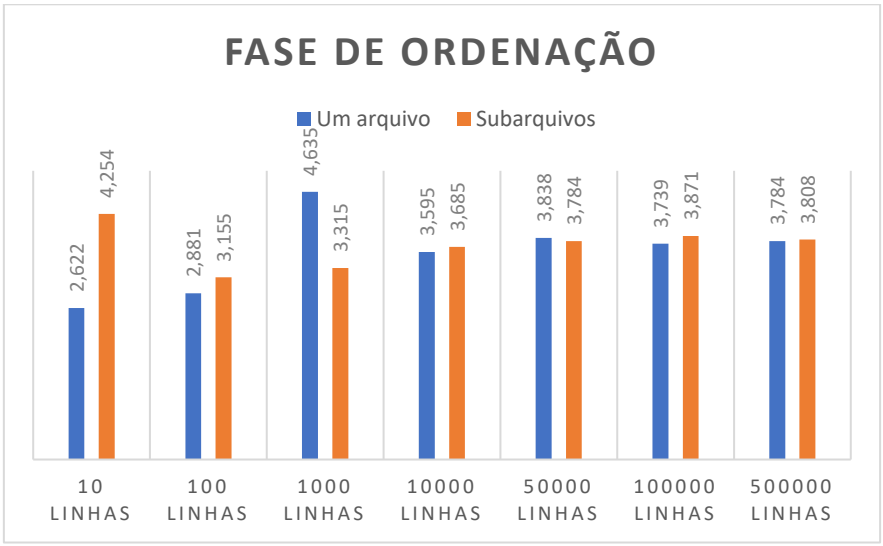


Figura 5 – Gráfico da fase de ordenação dos algoritmos do external merge sort

Na fase de ordenação, podemos observar que no geral, os dois algoritmos obtiveram um resultado semelhante para quase todos os tamanhos de blocos.

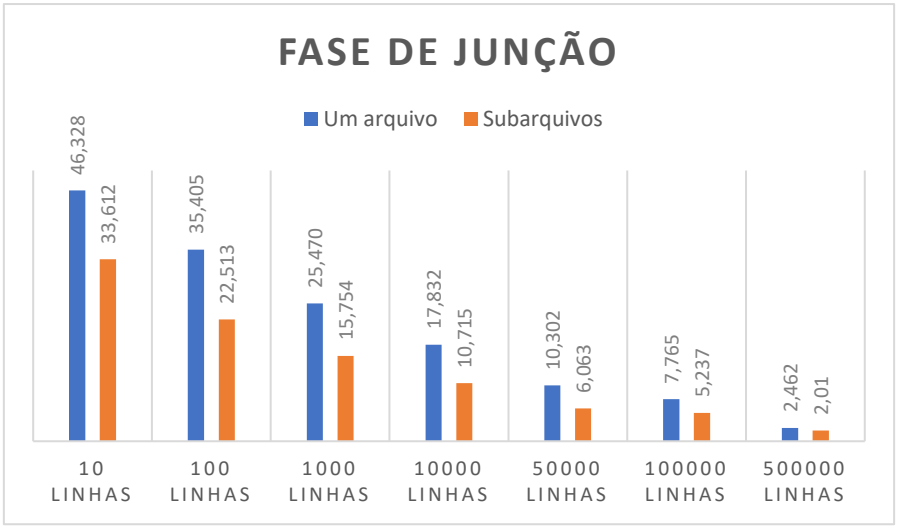
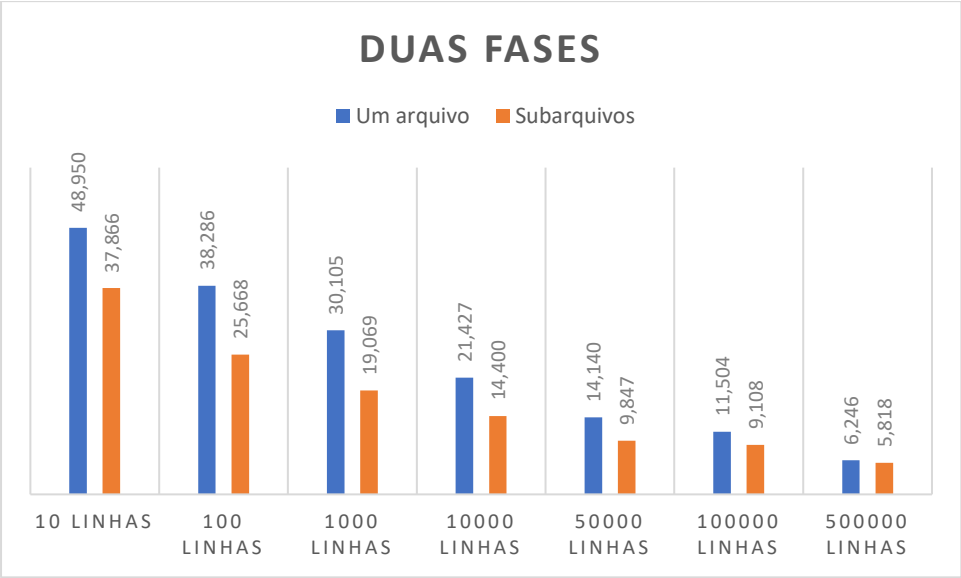


Figura 5 – Gráfico da fase de junção dos algoritmos do external merge sort

Na fase de junção o algoritmo que divide em sub arquivos obteve uma performance melhor em todos os tamanhos de blocos. Também é possível observar que para as duas formas quanto maior a quantidade de linhas de um bloco, menor o tempo para junção.



Juntando o tempo das duas fases podemos observar que o external merge sort que utiliza a estratégia de divisão em sub arquivos se saiu melhor em todos os tamanhos de blocos, até mesmo no bloco de 10 linhas no qual tinha obtido desempenho pior que o algoritmo de um arquivo na fase de ordenação.

5. Referências

External sorting – Disponível em: http://www.geeksforgeeks.org/external-sorting/ Acesso em: 05 dez. 2017.

External merge sort – Disponível em: http://mlwiki.org/index.php/External_Merge_Sort Acesso em: 05 dez. 2017.

External sorting – Disponível em: http://www.csbio.unc.edu/mcmillan/Media/Comp521F10Lecture17.pdf Acesso em: 05 dez. 2017.

noWorkflow – Disponível em: https://github.com/gems-uff/noworkflow Acesso em: 07 dez. 2017.