

Backup utilizando Streams

Dionmax Figueiredo Nobre¹, João Francisco da Silva de Paula¹

¹Ciência da computação – Universidade do Vale de Itajaí (Univali)

Caixa Postal 88302-202 – Itajaí – SC – Brasil

(dionmaxfn , joaofxp) @edu.univali.br

Abstract: *This report presents several lower-level analyses, byte for byte, in order to perform the backup action, i.e. a copy of a file to another location, showing results as runtime, file size and graphics comparatives, this report requires a basic knowledge about computer science.*

Resumo: *Este relatório apresenta várias análises de nível mais baixo, byte a byte de arquivos, com o intuito de realizar a ação de backup, ou seja, a cópia de um arquivo para outra localização, mostrando resultados como tempo de execução, tamanho do arquivo e gráficos comparativos, este relatório exige um conhecimento básico sobre ciência da computação.*

1. Introdução

Com este relatório apresentaremos do processo de backup de arquivos que começam de 5 mb até 50mb. Realizando testes de transferências, para se obter dados da velocidade de transferência através do método de transferência bit-a-bit, comparando resultados com números e gráficos.

2. Desenvolvimento

Foram realizados testes com arquivos dos seguintes tamanhos: 5 mb, 10 mb, 15 mb, 20 mb, 25 mb, 30 mb, 35 mb, 40 mb, 45 mb, 50 mb, gerados de forma aleatória através do site <http://pinetools.com/random-file-generator>.

O computador utilizado para coletar os dados foi um notebook da linha Dell, com as seguintes especificações de hardware: Processador Intel i5-5200U 2.20GHz, memória RAM 8Gb DDR4, memória de armazenamento HD 1 Tb 5400 RPM, placa de vídeo: GeForce 820m 2Gb de memória.

O código foi escrito na linguagem de programação Java, e o programa utilizado para fazer a execução do código e para coleta de dados foi o NetBeans, na versão 8.2. Durante toda a execução

do código, a IDE esteve funcionando juntamente com o aplicativo Microsoft Office Excel, usados para coletar os dados e fazer o tratamento dos dados e os devidos gráficos que serão mostrados mais abaixo.

Para todos os tipos de tamanhos de arquivos foram realizados cinco coletas do tempo de execução de hardware na unidade de milissegundos, em seguida os dados foram colocados em uma tabela para comparar.

2.1 Buffer Stream

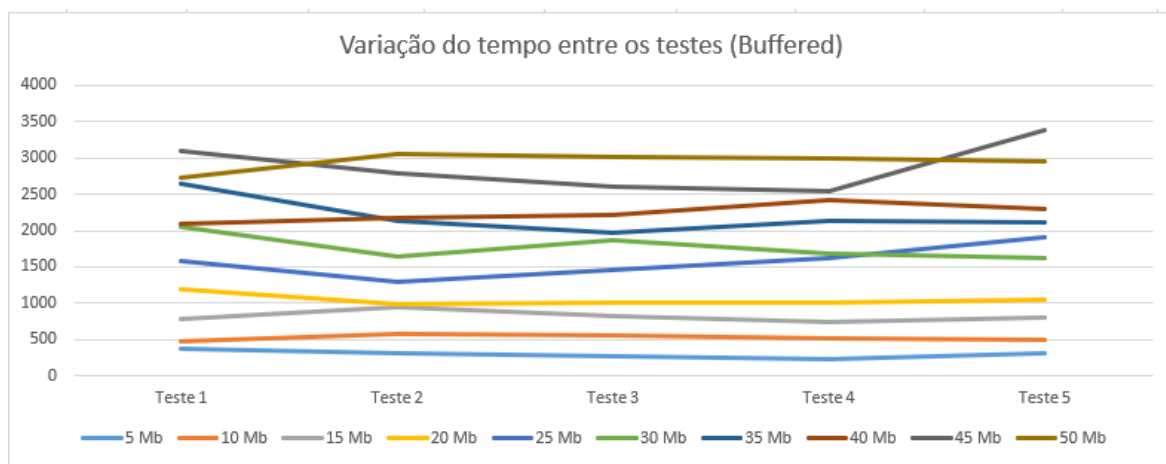
A tabela abaixo mostra os dados coletados usando o método com buffer, isso fez com que o desempenho na escrita e leitura de arquivos fosse superior em relação ao método sem buffer, ou seja, apenas o FileStream.

	5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Teste 1	367	471	771	1199	1573	2056	2645	2087	3098	2719
Teste 2	318	581	943	979	1295	1647	2129	2183	2780	3065
Teste 3	267	552	814	996	1467	1877	1973	2210	2609	3014
Teste 4	236	513	747	1010	1617	1676	2139	2411	2549	2998
Teste 5	300	492	796	1045	1899	1612	2109	2288	3381	2954
Média	297,6	521,8	814,2	1045,8	1570,2	1773,6	2199	2235,8	2883,4	2950

Para melhor compreensão da tabela acima, foi feita outra tabela, porém dessa vez com os valores em milissegundos convertidos para o formato “mm:ss.ms”, ou seja, minutos mais a esquerda, segundos no centro, e milissegundos arredondado mais a direita, melhorando a compreensão do tempo de forma habitual e não de máquina.

	5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Teste 1	00:00.367	00:00.471	00:00.771	00:01.199	00:01.573	00:02.56	00:02.645	00:02.87	00:03.98	00:02.719
Teste 2	00:00.318	00:00.581	00:00.943	00:00.979	00:01.295	00:01.647	00:02.129	00:02.183	00:02.780	00:03.65
Teste 3	00:00.267	00:00.552	00:00.814	00:00.996	00:01.467	00:01.877	00:01.973	00:02.210	00:02.609	00:03.14
Teste 4	00:00.236	00:00.513	00:00.747	00:01.10	00:01.617	00:01.676	00:02.139	00:02.411	00:02.549	00:02.998
Teste 5	00:00.300	00:00.492	00:00.796	00:01.45	00:01.899	00:01.612	00:02.109	00:02.288	00:03.381	00:02.954
Média	00:00.297,6	00:00.521,8	00:00.814,2	00:01.45,8	00:01.570,2	00:01.773,6	00:02.199	00:02.235,8	00:02.883,4	00:02.950

Abaixo mostra o gráfico de variação entre os testes, quanto maior o arquivo que era transferido, notou-se maior variação do mesmo.



2.2 File Stream

A tabela a seguir mostra os valores dos dados coletados em milissegundos através do método FileStream, sem usar buffer, isso faz com que demore mais do que os números mostrados anteriormente.

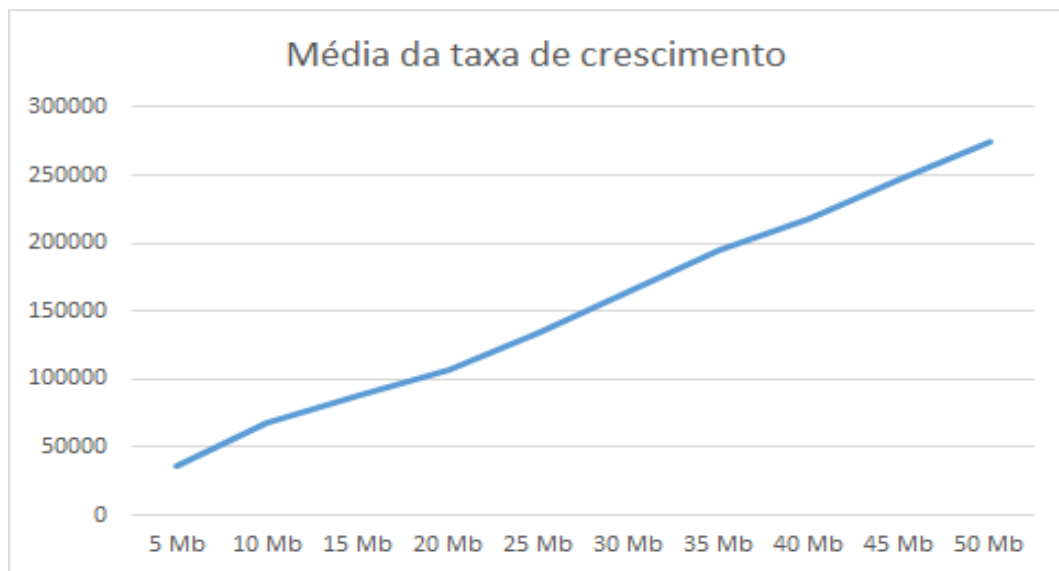
	5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Teste 1	45678	83726	96549	111729	139898	165049	192785	219013	247387	277099
Teste 2	40878	64216	91048	107524	133758	164063	190522	220950	247003	275444
Teste 3	32025	62579	79231	106619	132852	166085	206396	219495	249425	277671
Teste 4	31467	65630	81869	106337	133274	164740	192737	217731	247787	272905
Teste 5	31428	62504	94222	106416	132524	165612	195786	217765	247447	272890
	5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Média	36295,2	67731	88583,8	107725	134461,2	165109,8	195645,2	218990,8	247809,8	275201,8

Novamente, os valores das tabela anterior foram transformados para valores de tempo mais

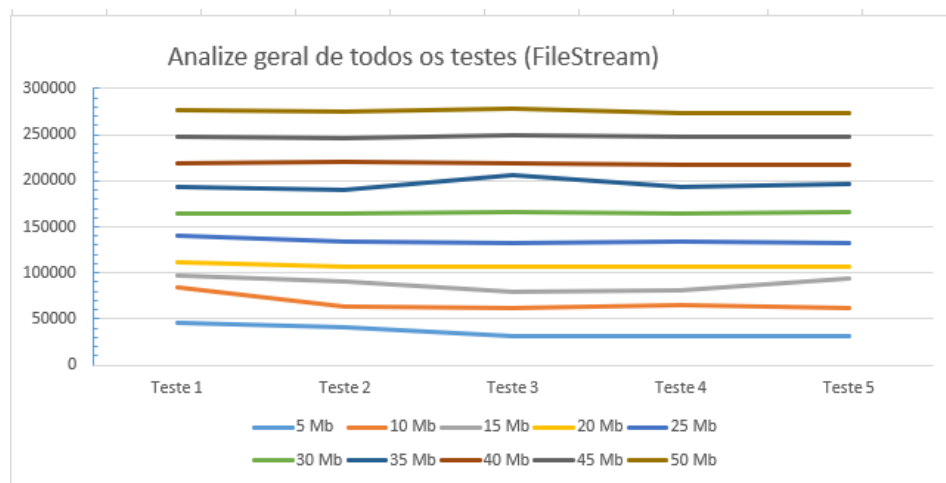
compreensíveis para humanos, formatado para “mm:ss”, no caso minutos a esquerda e segundos a direito, como não houve nenhum valor menor que um segundo (1s), não foi mostrado valores menores.

	5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Teste 1	00:45	01:23	01:36	01:51	02:19	02:45	03:12	03:39	04:07	04:37
Teste 2	00:40	01:04	01:31	01:47	02:13	02:44	03:10	03:40	04:07	04:35
Teste 3	00:32	01:02	01:19	01:46	02:12	02:46	03:26	03:39	04:09	04:37
Teste 4	00:31	01:05	01:21	01:46	02:13	02:44	03:12	03:37	04:07	04:32
Teste 5	00:31	01:02	01:34	01:46	02:12	02:45	03:15	03:37	04:07	04:32
Média	00:36	01:07	01:28	01:47	02:14	02:45	03:15	03:38	04:07	04:35

O gráfico abaixo mostra o aumento do tempo de execução proporcional ao tamanho do arquivo, embora se pareça muito com o gráfico de crescimento do tipo Buffer, a diferença é notada se observar o eixo de valores a esquerda do gráfico, que pode ser considerado dez vezes (9x) maiores que os do tipo buffer, isso porque é realmente mais lento.



A seguir os gráficos mostrando a variação dos resultados nos testes com FileStream, apesar de mais demorado, a variação foi menor em relação ao do tipo Buffer.



2.3 Comparativo

De acordos com os dados dos testes, provou-se que o modo de transferência de arquivos usando Buffer é superior ao modo de transferência mais simples que é o FileStream, como pode se observar nas tabelas e gráficos a seguir.

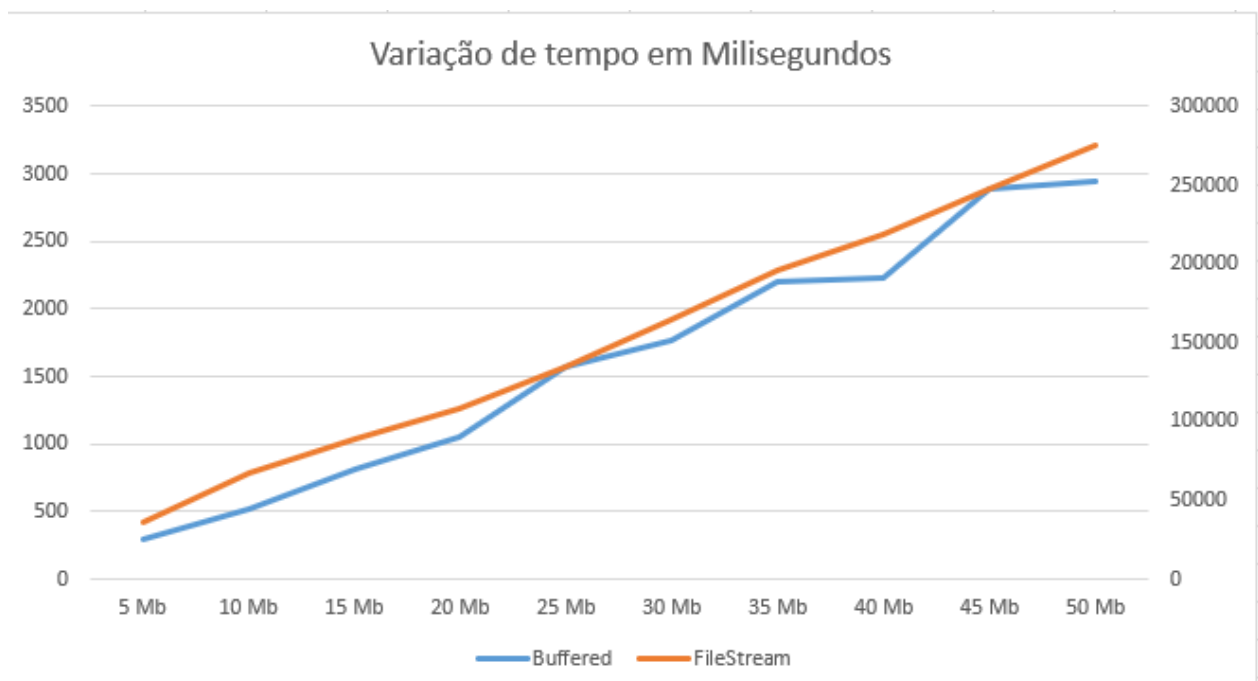
A tabela abaixo mostra a média comparativa do tempo de execução em cada um dos testes nos seus respectivos tamanhos, a parte de cima mostra em tempo real melhor compreensível a humanos e abaixo, o tempo em milésimos de segundos que é normalmente usado para comparações em hardware ,tendo em vista, que em todos eles o modo usando Buffer se saiu melhor.

Média em Milisegundos		5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Buffered		297,6	521,8	814,2	1045,8	1570,2	1773,6	2199	2235,8	2883,4	2950
FileStream		36295,2	67731	88583,8	107725	134461,2	165109,8	195645,2	218990,8	247809,8	275201,8
Média em tempo real		5 Mb	10 Mb	15 Mb	20 Mb	25 Mb	30 Mb	35 Mb	40 Mb	45 Mb	50 Mb
Buffered		00:00.297	00:00.521	00:00.814	00:01.45	00:01.570	00:01.773	00:02.199	00:02.235	00:02.883	00:02.950
FileStream		00:36.295	01:07.731	01:28.583	01:47.725	02:14.461	02:45.109	03:15.645	03:38.990	04:07.809	04:35.201

O gráfico abaixo mostra a comparação de do tempo médio dos dois modos de transferência, como o modo usando buffer custou menos tempo de transferência, não fica claramente visível, isso se deve porque não chegou aos três mil milissegundos.



Para resolver a grande diferenciação dos gráficos, e tornar melhor a compreensão, foi criados eixos separados, com o da esquerda sendo para os testes com buffer, e os da direita com um tamanho de dez vezes maior para os testes do tipo FileStream, assim fica clara a tamanha diferença de tempo entre os dois.



3. Conclusões

Concluimos então que através desse trabalho de pesquisa científica sobre Streams, usando dois modos diferentes de tipos de transferência: FileStream e o BufferedStream, chegou-se à conclusão através de testes e comparações com os dados obtidos, que o modo de transferência usando Buffer é superior ao modo File, podendo ser até dez vezes mais rápido.

4. Referências Bibliográficas

Gerador de arquivos usados no trabalho: “<http://pinetools.com/random-file-generator>”.

Configurações do computador usado nos testes para coletas de dados:

“<https://www.dell.com/support/home/br/pt/brbsdt1/product-support/product/inspiron-14-3443-laptop/manuals>”.

Ide usada nas pesquisas dos dados: “<https://netbeans.org/about/index.html>”