

# Questão 1

a) Um *extent* é basicamente um par de números, sendo o primeiro o endereço do primeiro bloco de um grupo contíguo de blocos; e o segundo o endereço do último bloco. Desta maneira, há uma redução gigantesca na quantidade de metadados, pois não precisamos de um ponteiro para cada bloco deste grupo. Além disso, este *extent* é armazenado no próprio inode (na verdade até 4 *extents* são armazenados de maneira direta no ext4, os demais são armazenados em uma árvore que é apontada no inode). Pelo fato de trabalharmos com grupos de blocos contíguos, há uma redução considerável na fragmentação.

b) As principais justificativas para a introdução de *extents* no ext4 foram melhorar/aumentar o desempenho, a confiabilidade e a capacidade do sistema (agora com arquivos de tamanho máximo de 16 TB e volumes de 1 EB OBS: Ambos para blocos com 4 KB). Isso foi obtido pela redução de metadados (da necessidade de leitura e escrita, pois o espaço em si poupado por esta redução é pouco considerável), além da redução da fragmentação, por cada *extent* se tratar de um grupo contíguo de dados.

c) O problema gerado é que no uso de *extents*, não sabemos exatamente quantos blocos contíguos os dados ocuparão, fazendo com que o objetivo dos *extents* seja derrotado com a alocação de um bloco por vez. Para solucionar isso, o alocador do ext4 faz alocação de múltiplos blocos de uma vez, diminuindo muito o *overhead*. Além disso, o ext4 também faz o uso de uma técnica chamada de *Delayed allocation*, que essencialmente guarda em cache, os blocos antes de realmente alocá-los, assim permitindo uma melhor otimização na alocação.

d) Em um caso onde todos os blocos são contíguos, o *overhead* do esquema baseado em *extents* será menor, pois temos essencialmente uma array, com um ponteiro para o bloco inicial, enquanto no esquema de alocação indexada, apesar dos blocos serem contíguos, mesmo assim temos um ponteiro para cada bloco.

Já no caso de não haverem blocos adjacentes, o *overhead* do esquema de alocação indexada será menor, simplesmente porque este sistema possui até 12 ponteiros diretos que significa que todos os ponteiros estarão no inode, enquanto no esquema de *extents*, apenas os quatro primeiros *extents* estarão diretamente no inode, e o restante estará em uma árvore. Contudo, se o arquivo estiver contido em 4 blocos ou menos, o *overhead* será teoricamente idêntico, porém na prática, o sistema de *extents* terá um desempenho melhor devido a diversas outras otimizações que este possui.

## Questão 2

- a) Foram encontrados 70079 arquivos.
- b) O maior arquivo é /usr/share/teams/teams que tem tamanho de 114977216 bytes.
- c) Existem 24 arquivos de tamanho 0, que correspondem a 0.034% do total.
- d) A media de tamanho é 19586 bytes, com 91.109% dos arquivos menores que ela.
- e) A mediana é 2038 bytes.
- f) O menor tamanho de bloco eh 2048 bytes.
  - 50.153% dos arquivos ocupariam apenas um bloco.
  - O maior arquivo ocuparia 56142 blocos de 2048 bytes.
- g) Ponteiros diretos: 92.735%
  - Indirecao simples: 7.102%
  - Indirecao dupla: 0.163%
  - Indirecao tripla: 0.000%
- h) O espaço desperdiçado por fragmentação interna é 176990212 bytes (12.894%).

## Questão 3

Nome	Tamanho do bloco (KB)	Tamanho da área de dados (GB)
Paulo Roberto Albuquerque	16	1

1 GB = 1048576 KB

Então temos 65536 blocos de 16 KB, sendo que cada bloco necessita de 1 bit no mapa de bits. A cada 8 bit, temos um byte, logo **temos um mapa de 8192 bytes ou 8 KB.**

Fontes usadas para a resolução da questão 1:

[https://ext4.wiki.kernel.org/index.php/Ext4\\_Design](https://ext4.wiki.kernel.org/index.php/Ext4_Design)

[https://ext4.wiki.kernel.org/index.php/Ext4\\_Howto#Extents](https://ext4.wiki.kernel.org/index.php/Ext4_Howto#Extents)

<https://unix.stackexchange.com/questions/223310/what-do-extents-feature-do-in-ext4-filesystem-in-linux>

<https://opensource.com/article/17/5/introduction-ext4-filesystem>

<https://en.wikipedia.org/wiki/Ext4>

[https://en.wikipedia.org/wiki/Extent\\_\(file\\_systems\)](https://en.wikipedia.org/wiki/Extent_(file_systems))

<https://www.youtube.com/watch?v=B6kg2zeJ9do>