

En un sistema con bloques de 4KiB el administrador desea dar formato a un disco de 1 TiB. Para ello hace un estudio de los ficheros a almacenar llegando a las siguiente distribución de ficheros:

Tamaño del fichero	porcentaje de ficheros
512 B	10 %
32 KiB	30 %
1 MiB	50 %
50 MiB	10 %

Se pide:

a) Determinar el tamaño máximo de partición que se puede tener cuando las direcciones de bloques físicos son de 32 bits.

b) Determinar el mínimo número de nodos\_i que deberíamos incluir al dar formato al disco.

Consideraremos ahora tres posibilidades:

- 1.- Un sistema de ficheros clasico de tipo UNIX. Con las siguientes características:
  - Direcciones de bloques físicos de 32 bits.
  - Nodo\_i de 128B.
- 2.- Un sistema de ficheros modificado, con las siguientes características:
  - Direcciones de bloques físicos de 32 bits.
  - Nodo\_i de 1KiB, que además de la información clásica incluye los primeros bytes del fichero.
- 3.- Un sistema con extents, con las siguientes características:
  - Nodo\_i de 256 B que incluye 13 extents o referencias a bloques de índices.
  - Extents de 12B.

c) En el supuesto de no existir fragmentación en el disco y de que éste esté lleno, calcular para cada alternativa el espacio dedicado a la siguiente metainformación:

- Mapas de bits.
- Nodos\_i.
- Bloques de indirección o extents.

d) Repetir el cálculo anterior para el caso de que cada fichero mayor de 32 KiB esté formado por 4 fragmentos.

## Solución

a) Con 32 bits se pueden referenciar  $2^{32}$  bloques de 4 KiB, lo que es igual a  $2^{44}$  B = 16 TiB.

b) El número de ficheros que cabrán, y, por tanto, el número de nodos\_i necesarios, dependerá del espacio ocupado por la metainformación, lo que depende de la solución empleada para la misma. Por otro lado, un fichero ocupa un número entero de bloques de datos.

Si despreciamos el espacio ocupado por la metainformación, el número de ficheros esperado X se puede calcular por:  $(4*0,1 + 32*0,3 + 1.024*0,5 + 50*1.024*0,1)$  KiB \* X = 1 TiB. Luego, el número de ficheros esperado es de 190.312 ficheros, que se reparten de la siguiente forma:

512 B	19.031
32 KiB	57.094
1 MiB	95.156
50 MiB	19.031

Si, para tener en cuenta la metainformación, consideramos la primera solución, tendríamos que un fichero de 0,5 KiB ocupa un nodo\_i de 128 B más un bloque de datos de 4KiB, el de 32 KiB ocupa 128 B más 8 bloques, etc. Por lo que sería:

$$X * ((4\text{KiB} + 128\text{B}) * 0,1 + (32\text{KiB} + 128\text{B}) * 0,3 + (1\text{MiB} + 128\text{B}) * 0,5 + (50\text{MiB} + 128\text{B}) * 0,1) = 1 \text{ TiB}$$

## 142 Problemas de sistemas operativos

Luego  $X = 190.308$  ficheros

Se observa que el tener en cuenta la metainformación y la ocupación real de los ficheros pequeños no afecta casi nada al resultado.

En un caso real, lo lógico es darse un margen de seguridad, por ejemplo, del 20 %, puesto que es muy desagradable encontrarse con que no tenemos suficientes nodos<sub>i</sub>. Además, el número final debe ser tal que los nodos<sub>i</sub> ocupen un número entero de bloques.

Para el resto del problema consideraremos, por tanto, que formateamos para  $190.308 * 1,2 / 1.024 = 223$  Kinodos.

e) Los espacios ocupados son los siguientes:

Mapa de bits de bloques: Número de bloques del disco: 1 TiB / 4 KiB. Número de bloques de mapa de bits:  $(1 \text{ TiB} / 4 \text{ KiB}) / (4 \text{ KiB} * 8 \text{ bits/B}) = 2^{13}$  bloques = 32 MiB = 8 Kibloques.

Mapa de bits de nodos<sub>i</sub>:  $223 \text{ Kinodos} / (4 \text{ KiB} * 8 \text{ bits/B}) = 6,97$  bloques, por lo que se necesitan 7 bloques.

Nodos<sub>i</sub>. Para el caso de nodo<sub>i</sub> de 128 B, son:  $223 \text{ Kinodos} * 128 \text{ B} / 4 \text{ KiB} = 7.136$  Bloques.

Nodos<sub>i</sub>. Para el caso de nodo<sub>i</sub> de 1 KiB, son:  $223 \text{ Kinodos} * 1 \text{ KiB} / 4 \text{ KiB} = 57.088$  Bloques.

Nodos<sub>i</sub>. Para el caso de nodo<sub>i</sub> de 256 B, son:  $223 \text{ Kinodos} * 256 \text{ B} / 4 \text{ KiB} = 14.272$  Bloques.

Obsérvese que en la solución 2 los ficheros de 512 B caben en el nodo<sub>i</sub>, por lo que nos ahorramos 19.031 bloques de datos de estos ficheros.

El nodo<sub>i</sub> tendrá 10 directos, lo que supone que los ficheros de hasta  $10 * 4 \text{ KiB} = 40 \text{ KiB}$  no necesitan bloques de indirección, lo que se aplica a los ficheros de 512 B y de 32 KiB.

Un bloque de indirección contiene 1 Ki direcciones de bloque, por lo que permite ficheros de hasta  $1 \text{ Ki} * 4 \text{ KiB} = 4 \text{ MiB}$ , más los 40 KiB de los directos. Eso significa que se necesita para los casos 1 y 2, un bloque por fichero de 1 MiB, es decir,  $95.156$  bloques de indirección.

Para los ficheros de 50 MiB nos hacen falta  $50 \text{ MiB} / 4 \text{ KiB} = 12,5$  Kibloques, por tanto, nos hace falta el bloque de indirectos, el bloque de doble indirecto más 12 bloques de índices de doble indirecto, es decir 14 bloques. Por tanto, necesitamos  $19.031 * 14 = 266.434$  bloques.

Para el caso de los extents no hace falta ningún bloque adicional, puesto que, al no estar fragmentado el disco, con un extent podremos definir todo el fichero. Veamos que con los datos que nos dan se observa que 50 MiB caben en un solo extent. Nos dicen que el extent es de 12B. Que debe contener, la dirección de bloque lógico, la dirección de bloque físico 4B, el número de bloques y si está en uso. Suponiendo que se dedican 20 bits para el tamaño, lo que da un margen más que suficiente para la dirección de bloque lógico más el bit de uso, un extent permitiría hasta un tamaño de  $2^{20} * 4 \text{ KiB} = 4 \text{ GiB}$ .

d) Los resultados son los mismos que en el caso anterior. Para las soluciones 1 y 2, no influye en absoluto que el fichero esté fragmentado o no. Para el caso de los extents, como caben hasta 13 extents por nodo<sub>i</sub> y solamente tenemos 4 particiones, caben perfectamente en el nodo<sub>i</sub>.