External Sorting

Zhe Li

October 2020

1 Background

Suppose we have a file containing R records, each of size S_r bytes. The file is initially stored consecutively on a disk. Each page has size of S_p bytes. The size of available memory is S_b bytes. For every question below, use the variables that appeared before to answer the questions.

2 Phase 1

2.1

What is the number of records that can be contained in one page? The result is represented by P.

$$P = \lfloor \frac{S_p}{S_r} \rfloor$$

.

2.2

What is the number of pages in memory? The result is represented by B

$$B = \frac{S_b}{S_p}$$

2.3

What is the number of pages required to store the file? The result is represented by F.

$$F = \frac{R}{P}$$

2.4

What is the number of SSLs that are going to be produced in external sorting as discussed in class? The result is represented by N_{ssl}.

$$N_{ssl} = \lceil \frac{F}{B} \rceil$$

2.5

What is the number of random read N_{1rr}, random write N_{1rw}, sequential read N_{1sr}, sequential write N_{1sw} in **Phase 1**?

$$N_{1rr} = N_{ssl}$$

$$N_{1rw} = N_{ssl}$$

$$N_{1sr} = F$$

$$N_{1sw} = F$$

3 Phase 2 without Double Buffering

Suppose we are not going to use double buffering, and B is large enough to perform **Phase 2** in one pass.

3.1

What is the minimum number of buffer pages required to perform **Phase 2** in one pass (do not bother to deal with ceiling or floor)? The result is represented by B_{min}.

$$B_{min} = \frac{1 + \sqrt{1 + 4F}}{2}$$

3.2

How many pages can each SSL get in buffer for **Phase 2**? The result is represented by B_{ssl}.

$$B_{ssl} = \lfloor \frac{B}{N_{ssl} + 1} \rfloor$$

3.3

What is the size of the output buffer in page? The result is represented by B_{out}.

$$B_{out} = B - B_{ssl} * N_{ssl}$$

3.4

What is the number of random read N_{2rr}, random write N_{2rw}, sequential read N_{2sr}, sequential write N_{2sw} in **Phase 2**?

$$N_{2rr} = \frac{F}{B_{ssl}}$$

$$N_{2rw} = \frac{F}{B_{out}}$$

$$N_{2sr} = F$$

$$N_{2sw} = F$$

4 Phase 2 with Double Buffering

Suppose now we are going to use double buffering, and B is large enough to perform **Phase 2** in one pass.

4.1

How many pages can each SSL get in buffer for **Phase 2**? The result is represented by B_{ssl}

$$B_{ssl} = \lfloor \frac{B}{N_{ssl} + 1} \rfloor$$

4.2

Assume B_{ssl} is odd. What is the size of the output buffer in page? The result is represented by B_{out}

$$B_{out} = B - (B_{ssl} - 1) * N_{ssl}$$

4.3

Assume B_{ssl} is odd. What is the number of random read N_{2rr}, random write N_{2rw}, sequential read N_{2sr}, sequential write N_{2sw} in **Phase 2**?

$$N_{2rr} = \frac{F}{(B_{ssl} - 1)/2}$$

$$N_{2rw} = \frac{F}{B_{out}/2}$$

$$N_{2sr} = F$$

$$N_{2sw} = F$$

5 Further Discussion

5.1

The above calculation will lead to the conclusion that using double buffering is even slower than not using double buffering. Explain why in practice the conclusion is incorrect.

Double buffering is used to overlap the CPU processing time and the data transfer time. However, in the calculation above, the CPU time is not considered. Without double buffering, the disk need to wait. But disk may not need to wait with double buffering. Therefore, in practice, double buffering may better the performance.

5.2

If B is not large enough, how do we perform external sorting? Express in big-O the number of passes required.

We can always use k-Way merge sort where $k \leq B - 1$. The number of passes is $O(log(F))$.