

DESCRIPTION

DATA CACHING METHOD



159159

TECHNICAL FIELD

The present invention relates to a data caching method for caching data by employing a read buffer.



G00002

BACKGROUND ART

A conventional data caching method will be described with reference to figures. Figure 6 is a block diagram illustrating an apparatus, to which the conventional data caching method is applied. In figure 6, numeral 61 denotes a storage medium, numeral 62 denotes a read buffer, and numeral 63 denotes a computer.

Figures 7(a) to 7(d) are diagrams for explaining an example of changes in cache data in the conventional data caching method. Data stored in the read buffer 62, i.e., cache data are changed in the order of figures 7(a), 7(b), 7(c) and 7(d), in which numeral 62 denotes the read buffer, numeral 72 denotes read data, numeral 73 denotes pre-read data, numeral 74 denotes read data, and numeral 75 denotes pre-read data.

Next, the operation of the conventional data caching method will be described with reference to figure 6 to 7(d). When a computer 63 such as host computer issues a reading request for data at addresses 50 to 52 in a storage medium 61 such as CD-ROM or DVD,

the data at addresses 50 to 52 of the storage medium 61 are read out by a reading apparatus not shown and, as shown in figure 7(a), the read data 72 which have been read out are buffered at the head of the read buffer 62 and then transferred to the computer 63. Thereafter, data at address 53 and subsequent addresses of the storage medium 61 are pre-read by the amount which can be written in the available space of the read buffer 62 and, as shown in figure 7(b), buffered up to the end position of the read buffer 62 so that the pre-read data 73, i.e., data at addresses 53 to 101 which have been pre-read are made continuous with the read data 72.

Subsequently, when the computer 63 issues a reading request for data at addresses 48 to 50 of the storage medium 61, it is judged whether the cache hit occurs or not depending on whether or not data at address 48 as data at a reading starting position are stored in the read buffer 62. Since the data at address 48 are not included in the read data 72 and the pre-read data 73 which have been stored in the read buffer, the cache hit does not occur. Thus, data at addresses 48 to 50 of the storage medium 61 are read again and, as shown in figure 7(c), the read data 74 which have been read out are buffered at the head of the buffer memory 62 and then transferred to the computer 63. Thereafter, data at addresses subsequent to address 50 of the storage medium 61 are pre-read by the amount which can be written in the available space of the read buffer 62 and, as shown in figure 7(d), buffered up to the end position of the read buffer 62 so that the pre-read data 75, i.e., data at addresses 51

15 APR 2002

3

to 99 are made continuous with the read data 74.

As described above, in the conventional data caching method, even when part of data for which the computer 63 issues the reading requests have been previously stored in the read buffer 62, it is judged that the cache hit does not occur when data at the reading starting position are not stored, and reading is performed again from the storage medium 61 to rewrite all of data in the read buffer 62, thereby resulting in a lower efficiency.

Further, when the computer 63 issues a reading request for data which are not stored in the read buffer 62, even when the address of the data is slightly forward than those of data stored in the read buffer 62, and there are few changes between pre-rewriting data and post-rewriting data, the data in the read buffer 62 are all rewritten. Accordingly, the efficiency is low.

The present invention is made in view of the above-described problems, and has for its object to provide a data caching method which can utilize cache data efficiently and effectively.

DISCLOSURE OF THE INVENTION

According to the present invention, there is provided a data caching method for caching data by employing a read buffer when a computer issues a reading request for data stored in a storage medium, comprising: a step of judging whether data stored in the storage medium, to which a reading request is issued, are in cache hit with data which have been already buffered in the read buffer; a buffering

step of, when it is judged in the cache hit judgement that data to which the reading request is issued or a part thereof are not in cache hit, reading out only data which are not in cache hit, among the data to which the reading request is issued, from the storage medium, to be buffered in the read buffer so as to be continuous with data which are in cache hit; a step of, after the buffering, transferring the data to which the reading request is issued in the read buffer to the computer; and a pre-reading step of, after transferring the data to which the reading request is issued from the read buffer to the computer, pre-reading data which are stored in the storage medium subsequent to the data to which the reading request is issued from the storage medium, to be buffered in the read buffer so as to be continuous with the data to which the reading request is issued in the read buffer. Therefore, it is judged whether or not all of data to which the computer issues the reading request are in cache hit, and only data which are not in cache hit are read out from the storage medium to be buffered in the read buffer, whereby there is no need to rewrite all of data in the read buffer, and thus it is possible to utilize cache data efficiently and effectively.

According to the present invention, in the aforementioned data caching method, in a case where data have been already stored in the read buffer so as to be continuous with the data to which the reading request is issued in the read buffer, the pre-reading step does not carry out the pre-reading. Therefore, there is no need to rewrite data which have already been stored in the read buffer, and

thus it is possible to utilize cache data efficiently and effectively.

According to the present invention, in the aforementioned data caching method, when it is judged in the cache hit judgement that at least rearmost data among the data to which the reading request is issued are in cache hit, the buffering step reads only data which are not in cache hit among the data to which the reading request is issued, from the storage medium, and buffers the data immediately forward data which are in cache hit which have been already buffered in the read buffer so as to be followed by the data in cache hit. Therefore, even when the computer issues a reading request for data of addresses which are forward than that of data to which the computer issues the reading request the last time, there is no need to read again the all data to which the reading request is issued from the storage medium, and the data can be transferred from the read buffer to the computer, whereby it is possible to utilize cache data efficiently and effectively.

According to the present invention, in the aforementioned data caching method, an arbitrary-sized margin which does not record the pre-read data is provided at an end position of the read buffer, and when the size of data which are not in cache hit among the data to which the reading request is issued is equal to or smaller than that of the margin, the data which are not in cache hit are read out from the storage medium to be buffered in the margin so as to be continuous with the data in the read buffer. Therefore, there

is no need to rewrite all of data in the read buffer, whereby it is possible to utilize cache data efficiently and effectively.

According to the present invention, in the aforementioned data caching method, before the pre-reading step, the margin is set again immediately forward the data to which the reading request is issued. Therefore, the margin is always provided at the end section of data in the read buffer, whereby it is possible to utilize cache data efficiently and effectively.

According to the present invention, in the aforementioned data caching method, the size of the margin can be changed into an arbitrary size on the basis of a past buffering state. Therefore, margin can always have an optimal size, whereby it is possible to utilize cache data efficiently and effectively.

BRIEF DESCRIPTION OF DRAWINGS

Figure 1 is a flowchart showing a buffering procedure of a read buffer in a data caching method according to a first embodiment of the present invention, figures 2(a) to 2(c) are diagrams for explaining an example of changes in cache data in the data caching method according to the first embodiment of the present invention, figures 3(a) to 3(d) are diagrams for explaining an example of changes in cache data in a case where all of data for which a reading request is issued are in cache hit in a data caching method according to a second embodiment of the present invention, figures 4(a) to 4(c) are diagrams for explaining an example of changes in cache data

in a case where part of data for which a reading request is issued are in cache hit in a data caching method according to a third embodiment of the present invention, and figures 5(a) to 5(e) are diagrams for explaining an example of changes in cache data in a case where the size of a margin is set again in a data caching method according to a fourth embodiment of the present invention.

In addition, figure 6 is a block diagram illustrating an apparatus to which a conventional data caching method is applied, and figures 7(a) to 7(d) are diagrams for explaining an example of changes in cache data in the conventional data caching method.

BEST MODE TO EXECUTE THE INVENTION

(Embodiment 1)

Hereinafter, a data caching method according to a first embodiment of the present invention will be described with reference to the figures.

Since the structure of an apparatus to which the data caching method according to the first embodiment is applied is the same as that of the apparatus to which the conventional data caching method is applied shown in figure 6, its description will be omitted.

A description is given of a buffering operation of a read buffer in the data caching method according to the first embodiment with reference to figure 1. Figure 1 is a flowchart showing a buffering procedure of the read buffer in the data caching method according to the first embodiment.

At first, the computer 63 issues a reading request for data stored in the storage medium 61. In step S1, it is judged whether or not the head data among the data for which the reading request is issued (hereinafter, referred to as reading start position data) are in cache hit, and when the data are in cache hit, the operation goes through processes without reading any data from the storage medium 61, and when the data are not in cache hit, the operation proceeds to step S2. In step S2, it is judged whether or not the rearmost data among the data for which the reading request is issued (hereinafter, referred to as reading end position data) are in cache hit, and when the data are not in cache hit, the operation proceeds to step S6 and all of the data for which the reading request is issued are read out from the storage medium 61 to be buffered at the head position of the read buffer 62, while when the data are in cache hit, the operation proceeds to step S3. In step S3, the size of data which are not in cache hit among the data for which the reading request is issued is compared with the remaining buffer size which is obtained by excluding cache-hit data among the data for which the reading request is issued from the read buffer 62, and when the size of the data which are not in cache hit is larger than the remaining buffer size, the operation proceeds to step S6, while when the size is equal to or smaller than the remaining buffer size, the operation proceeds to step S4. In step S4, the data which are not in cache hit among the data for which the reading request is issued are read out from the storage medium 61 to be buffered in the read buffer

62 so that the data are made continuous with the cache-hit data. In step S5, a head position of the data stored in the read buffer 62, for which the computer 63 issues the reading request, is set again as a head position of the read buffer, thereby to end the buffering process of the data for which the reading request is issued.

Next, the data caching method according to the first embodiment of the present invention will be described, with seeing changes in data in the read buffer 62, with reference to the figures. Figures 2(a) to 2(c) are diagrams for explaining an example of the changes in cache data in the data caching method according to the first embodiment of the present invention. The cache data are changed in the order of figure 2(a), 2(b) and 2(c), in which numeral 62 denotes a read buffer, numeral 22 denotes read data, numeral 23 denotes pre-read data, and numeral 24 denotes read data.

When the computer 63 issues a reading request for data at addresses 50 to 52 of the storage medium 61, in step S1 it is judged whether or not the data at address 50 are stored in the read buffer 62, and the operation proceeds to step S2 because the data are not stored, and in step S2 it is judged whether or not the data at address 52 are stored in the read buffer 62, and the operation proceeds to step S6 because the data are not stored, and in step S6 the data at addresses 50 to 52 of the storage medium 61 are read out by a reading apparatus not shown, the read data 22 which have been read out are buffered at the head of the read buffer 62 as shown in figure

2(a), then the buffering process for the data is ended, and the data for which the reading request is issued are transferred from the read buffer 62 to the computer 63. Thereafter, data at address 53 and subsequent addresses of the storage medium 61 are pre-read by the amount which can be written in the available space of the read buffer 62, and buffered up to an end position of the read buffer 62 so that the pre-read data 23, i.e., data at addresses 53 to 101 which have been pre-read are made continuous with the read data 22, as shown in figure 2(b).

Subsequently, when the computer 63 issues a reading request for data at addresses 48 to 50 of the storage medium 61, in step S1 it is judged whether or not the data at address 48 as the reading start position data are stored in the read buffer 62, and the operation proceeds to step S2 because the data at address 48 are not stored, and in step S2 it is judged whether or not the data at address 50 as the reading end position data are stored in the read buffer 62, and the operation proceeds to step S3 because the data at address 50 have already been stored in the read buffer 62. In step S3, it is decided that the data at addresses 48 and 49 which are not stored in the read buffer 62 can be buffered into the read buffer 62, and the operation proceeds to step S4. In step S4, the data at addresses 48 and 49 which are not stored in the read buffer 62 are read out from the storage medium 61, and buffered so that the read data 24 at addresses 48 and 49 which have been read are made continuous with the data at address 50 and the following addresses, as shown in figure

2(c). The read buffer 62 here has a ring structure. Then, the data for which the reading request is issued are transferred to the computer 63. Thereafter, in step S5, the part where the data at address 48 as the reading start position data are stored is set again as a head position of the read buffer 62.

As described above, in the data caching method according to the first embodiment of the present invention, in a case where part of data for which a reading request is issued has been already stored in the read buffer, only data which are not stored in the read buffer are read from the storage medium to be buffered so as to be continuous with the data already stored in the read buffer. Therefore, there is no need to rewrite all data in the read buffer, whereby it is possible to utilize cache data efficiently and effectively.

(Embodiment 2)

Hereinafter, a data caching method according to a second embodiment of the present invention will be described with reference to the figures.

Since the structure of an apparatus to which the data caching method according to the second embodiment is applied is the same as that of the apparatus to which the conventional data caching method is applied as shown in figure 6, the description is omitted.

The data caching method according to the second embodiment will be described with seeing changes in data in the read buffer 62, with reference to the figures. Figures 3(a) to 3(d) are diagrams for explaining an example of the changes in cache data in the data caching

method according to the second embodiment. The cache data are changed in the order of figure 3(a), 3(b), 3(c) and 3(d), in which numeral 62 denotes a read buffer, numeral 32 denotes read data, numeral 33 denotes pre-read data, numeral 34 denotes a margin, numeral 35 denotes pre-read data, numeral 36 denotes a margin, numeral 37 denotes cache data, and numeral 38 denotes cache data.

Next, the operation of the data caching method according to the second embodiment will be described with reference to figures 3(a) to 3(d). When the computer 63 issues a reading request for data at addresses 50 to 52 of the storage medium 61, the data at addresses 50 to 52 of the storage medium 61 are read out by a reading apparatus not shown and, as shown in figure 3(a), the read data 32 which have been read are buffered at the head of the read buffer 62 and then transferred to the computer 63. Thereafter, data at address 53 and the following addresses of the storage medium 61 are pre-read from the storage medium 61 by the amount which can be written in the available space of the read buffer 62, apart from the margin 34 which is set at the end position of the read buffer 62 by 5 addresses and do not record pre-read data and, as shown in figure 3(b), the pre-read data 33, i.e., data at addresses 53 to 100 which have been pre-read are buffered to be continuous with the read data 32 up to an address immediately preceding the margin 34.

Subsequently, assume that a reading request for data at addresses 56 to 61 of the storage medium 61 is issued by the computer 63. Since all of the data at addresses 56 to 61 are in the read buffer

62 and cache-hit, the cache data 37 are transferred from the read buffer 62 to the computer 63. Thereafter, as shown in figure 3(c), the head position of the read buffer 62 is set at the head position of the cache data 37, and the end position of the read buffer 62 is set at an address immediately preceding the head position of the read buffer 62. A margin 36 is set at the end position of the read buffer 62 by 5 addresses, which is the same as that of the margin 34. Thereafter, data at address 101 and the following addresses of the storage medium 61 are pre-read by the amount which can be written from the head position of the margin 34 up to the address immediately preceding the margin 36. As shown in figure 3(c), the pre-read data 35, i.e., data at addresses 101 to 106 which have been pre-read are buffered. The read buffer 62 here has a ring structure.

Subsequently, when the computer 63 issues a reading request for data at addresses 51 to 53 which are forward addresses than data at addresses 56 to 61 of the storage medium 61, for which the reading request is issued the last time, since the data at addresses 51 to 53 are in the read buffer 62 as shown in figure 3(d) and cache-hit, the cache data 38 are transferred from the read buffer 62 to the computer 63. Further, since the data at address 51 as the reading start position data are data at the head position of the margin 36, there is no need to pre-read the data.

As described above, in the data caching method according to the second embodiment of the present invention, an arbitrary-sized margin which are not rewritten by the pre-read data is set at the

end position of the read buffer, whereby data to which the reading request was issued the last time remain in the read buffer, and it is possible to utilize cache data efficiently and effectively.

Further, when all of data to which the reading request is issued are cache-hit, the setting is changed such that the head position of cache data is the head position of the read buffer, and an address immediately preceding the head position of the read buffer is the end position of the buffer, and further the position of a margin provided at the end position of the buffer is also changed, whereby the data to which the reading request was issued the last time remain in the buffer, and thus it is possible to utilize cache data efficiently and effectively.

(Embodiment 3)

Hereinafter, a data caching method according to a third embodiment of the present invention will be described with reference to the figures.

Since the structure of an apparatus to which the data caching method according to the third embodiment is applied is the same as that of the apparatus to which the conventional data caching method is applied as shown in figure 6, the description is omitted.

The data caching method according to the third embodiment will be described, with seeing changed in data in the read buffer 62, with reference to the figures. Figures 4(a) to 4(c) are diagrams for explaining an example of the changes in cache data in the data caching method according to the third embodiment. The cache data

are changed in the order of figure 4(a), 4(b) and 4(c), in which numeral 62 denotes a read buffer, numeral 32 denotes read data, numeral 33 denotes pre-read data, numeral 34 denotes a margin, and numeral 44 denotes read data.

Next, the operation of the data caching method according to the third embodiment will be described with reference to figures 4(a) to 4(c). When the computer 63 issues a reading request for data at addresses 50 to 52 of the storage medium 61, the data at addresses 50 to 52 of the storage medium 61 are read out by a reading apparatus not shown and, as shown in figure 4(a), the read data 32 which have been read out are buffered at the head position of the read buffer 62, then the buffering process of the data is ended, and the data to which the reading request is issued are transferred from the read buffer 62 to the computer 63. Thereafter, data at address 53 and the following addresses of the storage medium 61 are pre-read by the amount which can be written in the available space of the read buffer 62, apart from the margin 34 set at the end position of the read buffer 62 by 5 addresses, and the pre-read data 33, i.e., data at addresses 53 to 100 which have been pre-read are buffered to be continuous with the read data 32 up to an address immediately preceding the margin 34, as shown in figure 4(b).

Subsequently, assume that a reading request for data at addresses 48 to 52 of the storage medium 61 is issued by the computer 63. The data at addresses 50 to 52 have been already stored in the read buffer 62. On the other hand, the data at addresses 48 and 49 are not stored

in the read buffer 62. Further, the size of the data which are not stored is smaller than that of the margin 34. In this case, the data at addresses 48 and 49 are read out from the storage medium 61. As shown in figure 4(c), the read data 44 which have been read out are buffered at the end position of the margin 34 so that the data in the read buffer 62 are made continuous. Then, the data at addresses 48 to 52 of the storage medium 61 are transferred to the computer 63. The read buffer 62 here has a ring structure.

As described above, in the data caching method according to the third embodiment, a margin which are not rewritten by the pre-read data is set at the end position of the read buffer, and in a case where part of the data to which the reading request is issued are stored in the read buffer, only data which are not stored in the read buffer are read out from the storage medium to be buffered at the end position of the margin and transferred to the computer, whereby there is no need to rewrite all of data in the read buffer, and thus it is possible to utilize cache data efficiently and effectively.

(Embodiment 4)

Hereinafter, a data caching method according to a fourth embodiment of the present invention will be described with reference to figures.

Since the structure of an apparatus to which the data caching method according to the fourth embodiment is applied is the same as that of the apparatus to which the conventional data caching

method is applied as shown in figure 6, its description is omitted.

The data caching method according to the fourth embodiment will be described, with seeing changes in data in the read buffer 62, with reference to the figures. Figures 5(a) to 5(e) are diagrams for explaining an example of the changes in cache data in the data caching method according to the fourth embodiment. The cache data are changed in the order of figure 5(a), 5(b), 5(c), 5(d) and 5(e), in which numeral 62 denotes a read buffer, numeral 32 denotes read data, numeral 33 denotes pre-read data, numeral 34 denotes a margin, numeral 44 denotes read data, numeral 54 denotes read data, numeral 55 denotes an increase in a margin, numeral 56 denotes a margin, numeral 57 denotes pre-read data, numeral 58 denotes read data, and numeral 59 denotes a margin.

Next, the operation of the data caching method according to the fourth embodiment will be described with reference to figures 5(a) to 5(e). When the computer 63 issues a reading request for data at addresses 50 to 52 of the storage medium 61, the data at addresses 50 to 52 of the storage medium 61 are read out by a reading apparatus not shown and, as shown in figure 5(a), the read data 32 which have been read out are buffered at the head of the read buffer 62, then the buffering process of the data is ended, and the data to which the reading request is issued are transferred from the read buffer 62 to the computer 63. Thereafter, data at address 53 and the following addresses of the storage medium 61 are pre-read by the amount which can be written in the available space of the read buffer

62, apart from the margin 34 set at the end position of the read buffer 62 by 5 addresses, and the pre-read data 33, i.e., data at addresses 53 to 100 which have been pre-read are buffered to be continuous with the read data 32 up to an address immediately preceding the margin 34, as shown in figure 5(b).

Subsequently, assume a reading request for data at addresses 44 to 52 of the storage medium 61 is issued by the computer 63. The data at addresses 50 to 52 have been already stored in the read buffer 62. The data at addresses 44 to 49 are not stored in the read buffer 62. In addition, the size of the data which are not stored is larger than that of the margin 34. In this case, the data at addresses 44 to 52 in the storage medium 61 are read again, and the read data 54 which have been read again are buffered at the head of the read buffer 62 as shown in figure 5(c) and then transferred to the computer 63. Then, the size of the margin is changed into the same size as that of the data which were not stored in the read buffer. The size of the data which were not buffered is 6 addresses, while the size of the margin 34 is 5 addresses. Therefore, the size of the increase 55 in the margin is equal to 1 address. The margin 56 of 6 addresses is newly set again at the end position of the read buffer 62. Thereafter, data at address 53 and the following addresses of the storage medium 61 are pre-read from the storage medium 61 by the amount which can be written in the available space of the read buffer 62, apart from the margin 56, and the pre-read data 57, i.e., data at addresses 53 to 93 which have been pre-read are buffered to be

continuous with the read data 54 up to an address immediately preceding the margin 56, as shown in figure 5(d).

Subsequently, assume that a reading request for data at addresses 38 to 46 of the storage medium 61 is issued by the computer 63. The data at addresses 44 to 46 have been already stored in the read buffer 62. The data at addresses 38 to 43 are not stored in the read buffer 62. In addition, the size of the data which are not stored is the same as that of the margin 56. In this case, the data at addresses 38 to 43 are read out from the storage medium 61. As shown in figure 5(e), the read data 58 which have been read out are buffered from the head of the margin 56. The read buffer 62 here has a ring structure. Then, the data to which the reading request is issued are transferred to the computer 63. Thereafter, the head position of the read buffer 62 is set at the head position of the read data 58, and the end position of the read buffer is set at an address immediately preceding the head position of the read buffer 62. Further, the margin 59 of 6 addresses, which has the same size as that of the margin 56, is set again at the end position of the read buffer 62.

As described above, in the data caching method according to the fourth embodiment, the margin which are not rewritten by the pre-read data is set at the end position of the read buffer, and in a case where part of data to which the reading request is issued are stored in the read buffer and further the size of data which are not stored in the read buffer is larger than that of the margin, the size of

the margin is set again to the size of the data which are not stored, whereby the margin always has an optimal size, and thus it is possible to utilize cache data efficiently and effectively.

APPLICABILITY IN INDUSTRY

As described above, a data caching method according to the present invention is suitable for a caching method for caching data in a CD, a CD-ROM, a CD-R (CD-Recordable), a CD-RW (CD-Re-Writable), a DVD-ROM (DVD-Read only Memory), a DVD-RAM (DVD-Random Access Memory) or the like.

CLAIMS

1. A data caching method for caching data by employing a read buffer when a computer issues a reading request for data stored in a storage medium, comprising:

a step of judging whether data stored in the storage medium, to which a reading request is issued, are in cache hit with data which have been already buffered in the read buffer;

a buffering step of, when it is judged in the cache hit judgement that data to which the reading request is issued or a part thereof are not in cache hit, reading out only data which are not in cache hit, among the data to which the reading request is issued, from the storage medium, to be buffered in the read buffer so as to be continuous with data which are in cache hit;

a step of, after the buffering, transferring the data to which the reading request is issued in the read buffer to the computer; and

a pre-reading step of, after transferring the data to which the reading request is issued from the read buffer to the computer, pre-reading data which are stored in the storage medium subsequent to the data to which the reading request is issued from the storage medium, to be buffered in the read buffer so as to be continuous with the data to which the reading request is issued in the read buffer.

2. A data caching method as defined in claim 1 wherein

in a case where data have been already stored in the read buffer so as to be continuous with the data to which the reading request is issued in the read buffer, the pre-reading step does not carry out the pre-reading.

3. A data caching method as defined in claim 2 wherein

when it is judged in the cache hit judgement that at least rearmost data among the data to which the reading request is issued are in cache hit, the buffering step reads only data which are not in cache hit among the data to which the reading request is issued, from the storage medium, and buffers the data immediately forward data which are in cache hit which have been already buffered in the read buffer so as to be followed by the data in cache hit.

4. A data caching method as defined in claim 3 wherein

an arbitrary-sized margin which does not record the pre-read data is provided at an end position of the read buffer, and

when the size of data which are not in cache hit among the data to which the reading request is issued is equal to or smaller than that of the margin, the data which are not in cache hit are read out from the storage medium to be buffered in the margin so as to be continuous with the data in the read buffer.

5. A data caching method as defined in claim 4 wherein

before the pre-reading step, the margin is set again immediately forward the data to which the reading request is issued.

6. A data caching method as defined in claim 5 wherein

the size of the margin can be changed into an arbitrary size

on the basis of a past buffering state.