Application of File Splitter in the Rapid Saving and Fetching Blob Data

Qu Baojun, Zhang YuPing
School of Mechanical Engineering
Shandong University of Technology
Zibo, China
qbj22@sina.com

Hu Lvbing
Jiujiang 707 Institute of Precision Mechatronics
SCI&TECH Co. Ltd
China Shipbuilding Industry Company Limited
JiuJiang, China

***Abstract*—The Blob saving ways in SQL Server and disk saving theory of files in operation systems is analyzed in this paper. A method of splitting files in the Blob saving is given, which has been proved to be available to improve the efficiency of saving multimedia data in PDM system by experiments.**

Keywords-SQL Server, BLOB, B-Tree, Split

I. INTRODUCTION

Now in PDM system，when people carry on the work of CAD and CAM，they need to deal with the multimedia data of grapf,image,voice,video and so on which make the hit of multimedia materials in PDM electronical data office more and more frequently. So the speed of storage and hit to these data has become a key question. Many databases in existence provide support to access of the larger field,for example, ORACLE，SQL Server and so on.In database system,these larger fields are called BLOB（binary system large data）.The access method to the data is different with the change of the programming language and the database.Therefore,the access to BLOB data type especially the research about how to access it fast has a very important meaning. The priority research in this paper is to get performance optimization by cutting up BLOB data in SQL Server.

II. THE BLOB STORAGE MODE IN SQL SERVER

A. The Summary Of BLOB Data Type

BLOB is indefinite binary system or categorical data which is tremendous. Usually it is file (.txt、.doc)and picture (.jpeg、.gif、.bmp),and it can be stored in databases. In SQL Server，BLOB can be text、ntext or image date type.

Text Non-Unicode data whose length is uncertainty is stored in server code page and the maximum length is 2^31-1(2 147 482 647) characters.

Ntext Unicode data whose length is uncertainty has the maximum length which is 2^30-1(1 073 741 823)characters. The storage in bytes of the field number is twice than the number of characters being entered. In SQL-92,ntext is a contraction of national text.

Image Binary data whose length is uncertainty. ranges from 0 to 2^31-1(2 147 482 647)bytes.

SQL Server takes the same way to handle these datatype. The way that SQL Server store image data is the same with it store text or ntext date type in the same conditions. Therefore，text、ntext and image date are usually called as text data.

Each row of text、ntext and image can contain not more than 2GB BLOB data in the table.The BLOB data is stored in the group of 8KB data page,which is separated form other data page stored in the same table.The data page is arrayed in the structure of B-tree.

B. BLOB's Storage Mode

BLOB's storage is divided into out-of-row BLOB and inrow BLOB in SQL Server 2000.

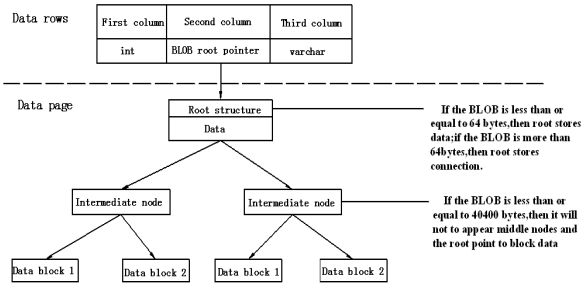

Figure 1. The storage data structure of BLOB data in SQL Server

(1) Out-Of-Row BLOB

When text in row isn't activated , BLOB is stored out of the row and only BLOB root pointer is stored in text、ntext or image row. BLOB root pointer points to the root of B-tree structure which actual stores BLOB data. BLOB data is stored in root structure when its dimension is less than or equal to 64 bytes. The storage of root structure mapped to the connection of the data's path while the data is stored in B-tree's leaf node when BLOB data's dimension is more than 64 bytes.

(2) Inrow BLOB

When text in row is activated， if BLOB data's dimension is less than or equal to limitation within row and the space is big enough to contain BLOB data then SQL Server will store the data in the data row.In row，Text in row is limited to apply to the fist BLOB row. Therefore，if the limitation within row is 256 bytes, and then the BLOB column in each row can store the data not more than 256

Published by Atlantis Press, Paris, France.

bytes as long as the total number of the byte in the line is not more than 8060 bytes.

Though BLOB string is too long compared to data line，and SQL Server can access data more quickly when it is stored in single page.The path that SQL Server hit BLOB data is cut down on account that root structure is stored in data line.

III. SPLIT AND ACCESS BLOB DATA

A. *Principle Analysis*

B-tree which is usually applied to database and file system is a kind of tree data structure. B-tree can make the material stay organized .Besides it has the even movement to input and delete the time of logarithm process. The element in B-tree is inserted by using bottom-up method which is different from the most binary tree which is inserted by using top-down method

Because BLOB data is stored in disk in form of B-tree structure in SQL Serve so when we read or write a BLOB data each time, we should start with root node and find the needed block data following subtree .Finally a integrated file will be formed and stored into database. Supposing that the B-tree's number of nodes is n and the height is H ,after visiting the node whose depth is h(1<h<H),if it wants to visit the node whose depth is h+1,it is inevitable to visit the node whose depth ranges from 1 to and is bound to cause visiting the same block data repeatedly. So in order to visit BLOB data in disk, the disk access time will be O(n*h).

If a BLOB data is divided into n equally sized files which are stored in sequential organization in adjacent record. Suppose the time for hitting each file is 0(1),after hitting the i-th record and then hitting the i+1-th record directly, it doesn't need to go through the i-th record. So the time for hitting the total disk is 0(n). Compare global storage with dividual storage. When n is comparatively large and h is , then O(n) is for less than O(n*h). therefore, theoretically, it's obvious to improve storage efficiency by dividing BLOB data and storing into database separately.

In a computer, all files is stored in storage media in binary form and is handled in binary form ,too. So we can divide the big data file into divers data equal in length and then write a new file separately. Combination is connecting all divided files orderly.

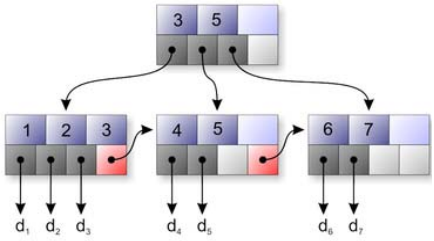

Figure 2. A simple B-tree

From figure 2, we can see a simple B-tree structure.

B. *Storage Experiment*

PDM application can use a variety of database access technology to access the SQL Server database. For quick and easy access to BLOB data, the experiment using ADO technology, because ADO is easy to use, fast, memory consumption, compared to ODBC, DAO, OLE DB and ADO database access technology.

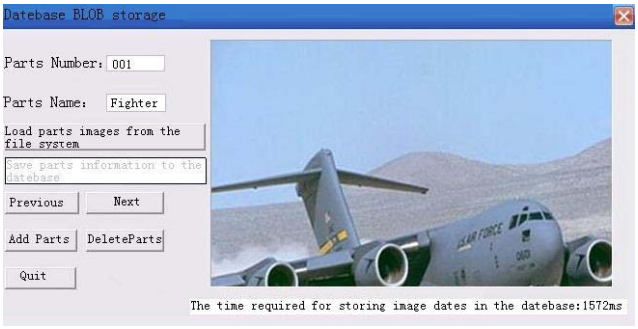

Figure 3. BLOB Datebase Storage Interface

Program interface shown in figure 3, the experiment environment is: high-speed local area network，the server is Intel PentiumⅣ，CPU1.7GHz,EMS memory 256 MB， the operating system is Windows 2000 Professional SP6；the client is Intel Pentium Ⅲ processor, CPU is 583MHz, RAM is 192 MB, the operating system is Windows XP Pro, 18 aircraft image files are stored in the client machine and it use the design process to a server database to store pictures.

When you store a single image, the storage time is not stable, so the random sampling of 18 aircraft ranging in size picture data, respectively, are stored in SQL Server database. In order to eliminate uncertainty in the operation, each image is stored five times. Experimental data are shown in Table I：

TABLE I. THE STORAGE TIME OF UNDIVIDED BLOB DATA

Number	1	2	3	4	5	6	7	8	9
Byte(M)	2.25	3.08	4.08	5.06	6.94	8.58	9.44	10.3	11.3
Storage time (ms)	1422	1032	2250	2188	2828	5657	5079	8172	6594
	1594	1687	2656	3109	3500	4547	4687	5625	11578
	484	1438	1437	3109	3157	3422	3344	5469	9328
	1109	1531	2422	2328	3156	3093	4922	4640	5140
	719	1203	1891	2203	3485	4953	4563	4875	11734
Average time (ms)	1065.6	1378	2131.2	2587.4	3225.2	4334.4	4519	5756.2	8875
Number	10	11	12	13	14	15	16	17	18
Byte (M)	13.4	14.4	16.8	18.0	19.3	20.2	21.9	23.3	34.3
Storage time (ms)	6937	8250	12594	15172	14828	13219	21907	39016	70781
	12719	14172	17860	21344	26093	24078	27250	48828	56265
	11594	14782	18687	24359	30547	15531	26438	46921	74266
	8546	7188	11875	13437	23172	28563	19813	29969	76234
	13969	7454	18531	14016	15188	23109	18922	20938	52141
Average time (ms)	10753	10369	15909	17665.6	21966	20900	22866	37134	65937

Experimental results show that, for each image file, when we store them at different times in the database, sometimes the differences of time-consuming are very large. But when we calculate the average time, we can find that both of the

general trends are consistent: the storage time increase with the picture file size in bytes' growth.

Compared with the analysis of large databases bytes of data storage time, a numerical analysis software Matlab to draw a large size in bytes of data - storage time curve shown in Figure 4。

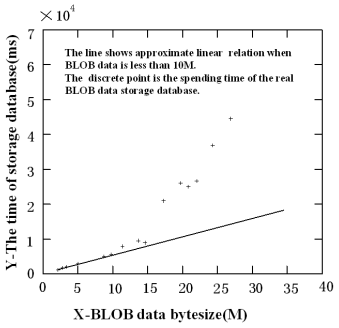

Figure4. The storage time of undivided BLOB data

We can see from Figure 4, when the BLOB data's size is less than 10 megabytes, the storage time increased linearly, and when the BLOB date's size is larger than 10 megabytes bytes, approximate growth rate increased exponentially. For example, it is easy to see when the BLOB date is 7 megabytes, the time stored in the database is about 3300 ms, and if we store 5 same size of the BLOB date, the total time is just 17 seconds. When the BLOB data size increased to 35 megabytes, the storage time is 67 seconds. It can be seen from the above figures, split the data before storing them into the database can save many time.

Therefore, according to the experimental results curves, for the size of more than 15M bytes of images, graphics and other large data files, before they are stored into databases, you should first split them into a number of files in 5 ~ 10M ones, and then put these small files into the database's adjacent records one by one in order to increase storage efficiency. When users need to use these data, then they can merge these small files, restore them to the original documents.

For example, for the size of 34.3M bytes of image files, before you store them into the database,you can divide it into 7 files.The six files is 5M,and the last one is 4.3M,then store them in the database's adjacent records.The storage time is as follows. the measured storage time follows table II below.

TABLE II. THE STORAGE TIME OF DIVIDED BLOB DATA

Number	1	2	3	4	5	6	7	8	9
Byte (M)	2.25	3.08	4.08	5.06	6.94	8.58	9.44	10.3	11.3
Storage time (ms)	1422	1032	2250	2188	2828	5657	5079	8172	6594
	1594	1687	2656	3109	3500	4547	4687	5625	11578
	484	1438	1437	3109	3157	3422	3344	5469	9328
	1109	1531	2422	2328	3156	3093	4922	4640	5140
	719	1203	1891	2203	3485	4953	4563	4875	11734
Average time (ms)	1065.6	1378	2131.2	2587.4	3225.2	4334.4	4519	5756.2	8875
Number	10	11	12	13	14	15	16	17	18
Byte (M)	13.4	14.4	16.8	18.0	19.3	20.2	21.9	23.3	34.3
Storage time (ms)	6936	8057	7964	8272	8628	9247	9731	17367	24961
	13772	12682	13837	14653	15347	17935	18258	20471	29578
	14854	14974	15936	15973	16739	18259	19683	21963	30603
	11963	12827	13148	12684	13473	11364	12349	15256	27843
	8327	9671	10759	9854	10537	12982	15267	12862	19374
Average time (ms)	11170.4	11642.2	12328.8	12287.2	12944.8	13957.4	15057.6	17584	26472

The date from table 2 can be seen, after split the image file, the average time which storing the date into the database is about 26 s, compared with the storage time of undivided, is significantly reduced.

To analysis the storage time after split the large bytes of date; we use numerical analysis software Matlab to draw upon a large partition size in bytes of bytes of data - storage time curve shown in Figure 5.

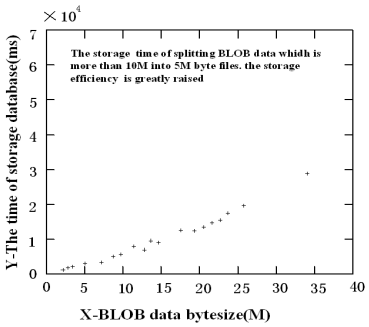

Figure5. The storage time of divided BLOB data

Experiments date from Figure 5 show that after you split the BLOB data more than 10M, and then store them into the datebase's adjacent records , the storage time will be greatly reduced, and significantly improve the efficiency of data access .

IV. SUMMARIES

We consider the requirements of the storage of database and handling large BLOB data slowness and access to shared resources of PDM, we divide the image files into several small pieces in the PDM system and store separately. We can see from the results, the use of file segmentation approach can significantly improve the BLOB data storage time and greatly improve the efficiency of data access. This method has great practical significance

REFERENCES

[1] Microsoft, the. Microsoft SQL Server 2000 Resource Kit [M]. Yun Zhou studio translation. Beijing: Mechanical Industry Press, 2002.

[2] Zhang li.Visual C++ Advanced Programming [M]. Beijing: Posts and Telecom Press, 2002.

[3] Yuan yi and other authors.Visual C++ Practice and improvement - database development and engineering application of article [M]. Beijing: China Railway Press, 2006.

[4] Seeking technology.Visual C++6.0 Database development technology and engineering practice [M]. Beijing: People's Posts and Telecommunications Press, 2004.

[5] Zhun chunhua,Zhang yubiao,Lu xinchun.BLOB Data type access method and its application. Computer Applications and Software [J], 2002,19 (10) :52-54