ESTEDI

European Spatio-Temporal Data Infrastructure
For High-Performance Computing

IST Project IST-1999-11009

Commission of the European Communities

VLDB Workshop
Supercomputing Databases

Roma, Italy 14. September 2001
Pontificia Università Urbaniana
at the VLDB 2001 Conference

Tertiary Storage Support for Multidimensional Data

Dipl.-Inf. Dipl.-Ing. Bernd Reiner
(FORWISS, TU-München)
reiner@forwiss.tu-muenchen.de

Contents

Tertiary Storage Support for Multidimensional Data

Tertiary Storage Support for Multidimensional Data

1 Motivation

Large-scale scientific simulations, experiments, and observational projects of the ESTEDI high performance computing (HPC) partner, generate large multidimensional data sets and then store them permanently in an archival mass storage system (tertiary storage systems). Typically, these data sets are stored on up to hundreds of magnetic tapes, cartridges, or optical disks. The access times and/or transfer times of these kinds of tertiary storage devices (i.e. one level below magnetic disk), even if robotically controlled, are relatively slow. Taking into account the time it takes to load, search, read, rewind, and unload a large number of cartridges, it can take many hours to retrieve a subset of interest from a large data set. In the past the HPC partners had always manually loaded whole files (data sets) from the magnetic tapes. Even if only a subdivision of the file is needed for the query. The problem was to find the relevant tape. The HPC partners must first search the right metadata in a separate database. With the metadata information the tape must be loaded manually from the tertiary storage system (TS-System). Then the needed data must be transferred via FTP to the Terminal of the user. This is a very time and work intensive process.

RasDaMan is a database system specially designed for multidimensional data sets. RasQL queries allow users (HPC partners) to access the specified small part (subset) of stored large amounts data generated by simulations instead of loading a whole large data set which correspond to one simulation (as necessary in the file-based approaches of the past). This feature of RasDaMan, in particular, now enables HPC partners to develop applications, which require high performance (e.g. presenting data via the WWW). Up to now RasDaMan is based on hard disks (short: HDD), e.g. RAID systems, for data storage. Tertiary storage systems (short: TS-Systems) could only be used for backup purposes, i.e. the user has to manually backup the data stored on HDD. To overcome this lack of exploiting tertiary storage media for the storage of the enormous amounts of HPC data sets (hundreds of Tera Bytes) was one of the user requirement of top priority and thus one of the main tasks of WP2. It results in an integration of a tertiary storage system into the multidimensional database system RasDaMan.

In this paper we will describe the implementation of the tertiary storage connection to the data-

base system RasDaMan using Storage Management Systems (HSM-Systems). This paper is organised as follows: in chapter 2 we present the extended RasDaMan architecture with tertiary storage connection. Chapter 3 will have a focus on the developed Super-Tile concept. Super-Tile is the granularity of the data stored on tertiary storage media. Strategies for reducing tertiary storage access time is discussed in chapter 4. In chapter 5 the tertiary storage interface of RasDaMan for the user and the system administrator will be shown. Chapter 6 summarises the achievements and give an outlook of future work.

2 Extended RasDaMan Architecture with TS connection

Within the first part of the ESTEDI project we implemented an easy to use method to save and load data from a TS-System. We developed a first prototype of the connection of TS-System to the multidimensional database system RasDaMan. In Figure 1 you can see the architecture of this system.

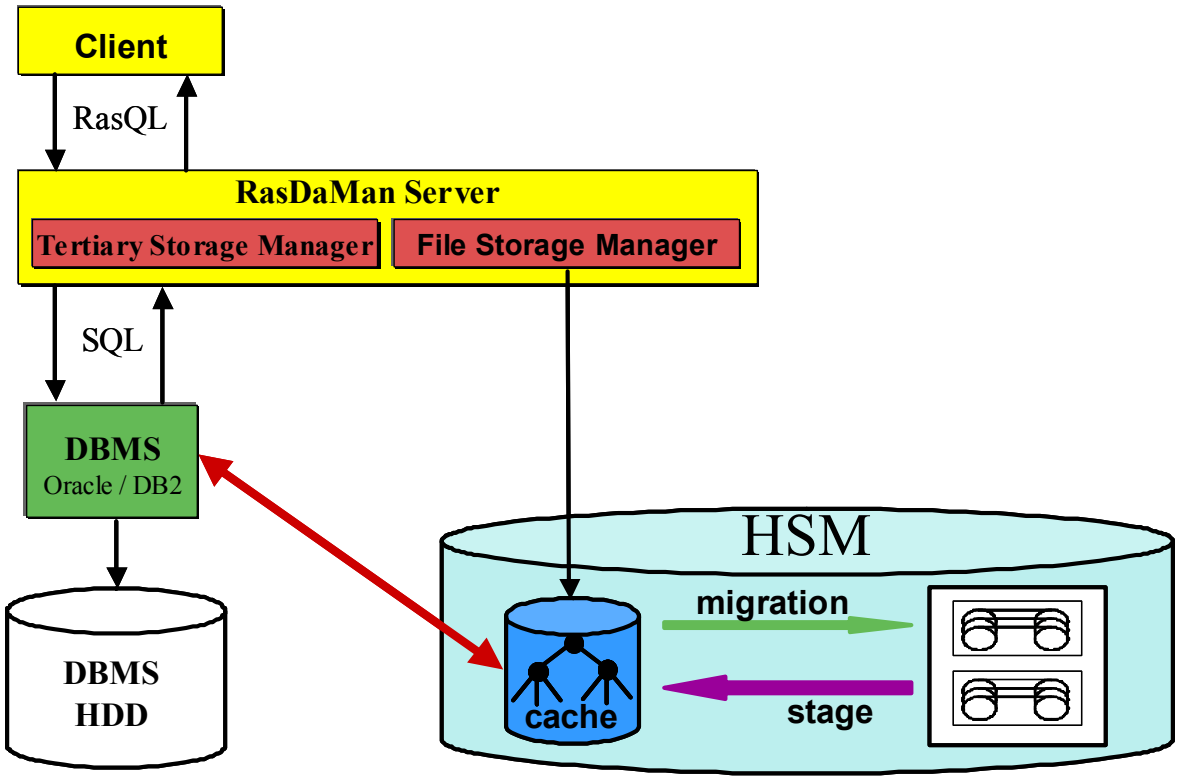


Figure 1 RasDaMan architecture with tertiary storage connection

On the left side of the figure you can see the original RasDaMan architecture with the RasDaMan client, RasDaMan server and conventional database management system (e.g. Oracle, IBM/DB2). The additional components for the TS connection are the Tertiary Storage Manager, File Storage Manager and Hierarchical Storage Management System (HSM-System). The TS-Manager and File Storage Manager are included in the RasDaMan server. The HSM-System is a conventional product like SAM (Storage Archiving System) from LSC Incorporation or Uni-Tree. Such an HSM-System (light blue in the figure) can be seen as a normal file system with unlimited storage capacity. In reality the virtual file system of HSM-Systems are separated to a limited cache (dark blue in the figure) on which the user works (load or store his data) and a tertiary storage system with robotic libraries. The HSM-System automatically migrates or stages the data to or from the tertiary storage media, if necessary. Tertiary storage media is the general term for media which can be used in tertiary storage system to save data. Examples for tertiary storage media are magnetic tapes, magneto optical tapes, CD-ROMs, DVDs, etc.

Generally we had two possibilities to connect tertiary storage systems to the database system RasDaMan. First of all we can use an existing system like an HSM-System. The other possibility is to develop a special and new connection to a tertiary storage system. We decided to use conventional HSM-Systems for the connection of tertiary storage devices to RasDaMan. Such HSM-Systems are especially developed to manage tertiary storage archive storage systems and to handle thousands of tertiary storage media (e.g. magnetic tape). Leading HSM-Systems are sophisticated and support robotic libraries, optical libraries, tape libraries, optical drives, tape drives, magnetic discs and disk arrays of a lot of manufactures. Another important reason for this decision was, that the HPC partners of the ESTEDI project already use such HSM-Systems with big robotic libraries (more than 100 TByte storage capacity). The HPC partners demands to use the available HSM-Systems also with RasDaMan. It is also more straightforward to develop and implement a connection of the RasDaMan system to the already stable HSM-System than to develop a whole new connection to Tertiary Storage Systems. The development of a new connection to TS-Systems is very complex, would take a long time and at the end we only have basic functionalities. It is even less complex to realise the connection to tertiary storage devices by HSM-Systems, because we only need a connection to the virtual file system of the HSM-System. This is a very flexible method for the tertiary storage integration of RasDaMan. E.g., we have even the possibility to export/import data stets of RasDaMan to tertiary storage devices over Internet per FTP.

The new RasDaMan TS functionality is based on the new component TS-Manager (shown in Figure 1). This TS-Manager is implemented and integrated into the RasDaMan kernel. If a new

query (RasQL[1]) is executed the TS-Manager knows whether the needed data sets are stored on hard disk or on a tertiary storage media. This metadata used by the TS-Manger is stored in RasDaMan respectively the database management system (DBMS). The performance is much more higher, if the metadata are stored permanently in DBMS and not be exported to tertiary storage media. If the data sets are on hard disk (DBMS), the query will be executed and handled without specific TS management. This is the normal procedure of the RasDaMan system without TS connection. If the data sets are stored on one or more tertiary storage media, the data sets must be imported into the database system first. The import of data sets stored on tertiary storage media is done by the TS-Manager automatically whenever a query is to be executed and those data sets are requested. The relational DBMS (e.g. Oracle) is used as a HDD cache for the data sets stored on tertiary storage media. After the import process of the data sets is done, RasDaMan can handle the data sets (stored in the DBMS cache area) in the normal way. The TS-Manager of RasDaMan has the information and metadata about all data sets, for example where the data sets are stored, how the data sets are organised on media, etc.

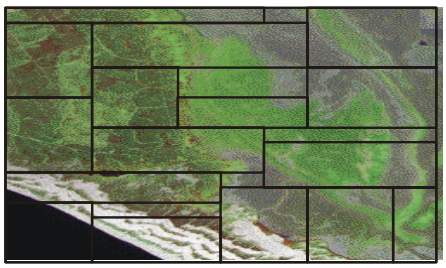
If the system administrator wants to insert new data sets (collections) into RasDaMan he can use the insert utilities of RasDaMan(e.g. the insert loading toolkit auf RasDaMan). In this case RasDaMan writes the collection into the DBMS. The data sets will not be exported to tertiary storage media automatically. Sometimes it is necessary to store the data sets only in DBMS, if the performance to load the data should be very high, e.g. some users have several accesses to the data sets. The system administrator can decide, whether the data sets are to be stored on hard disk (is already be done by the insert tool) or the data sets should be exported on tertiary storage media. These two possibilities are flexible in several cases. For example data sets are very often required from users at the beginning (insert time of data sets) and after several month the data sets are less important for users. In this case the data sets are first inserted into the DBMS can be exported to tertiary storage media after several month. If the data sets should only be stored in DBMS, the system administrator does not need to do anything else. When the collection should be exported to tertiary storage media the system administrator must start the export of the data sets manually. In the prototype version the system administrator must insert the collection name into the tertiary export configuration file (TertiaryExport.conf). After that the user must execute a query named ”export”. Then the named collection will be exported to the HSM-System, i.e. to tertiary storage media. In the next version of RasDaMan the system administrator can use a RasQL command to export the data sets to tertiary storage media instead of using the tertiary export configuration file.

[1] Query language of RasDaMan. Extended SQL language with multidimensional commands.

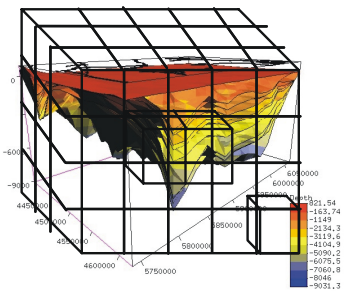
Now we will discuss the performance of the export and import functionality of the new tertiary storage version of RasDaMan. The export of data sets to tertiary storage media is very fast because the data sets must only be written to the virtual file system of the HSM-System. This means writing data sets to the HDD cache of the HSM-System (dark blue in Figure 1). The migration of the data sets from the HSM cache to the tertiary storage media does not concern the RasDaMan system. For the import functionality two cases must be distinguished. In the first case we assume that the data sets needed are already held in the HDD cache of the HSM system. This is not very implausible because the size of the HSM cache is normally hundreds of GByte. In this case the import of the data sets is as fast as the export of the data sets because the data sets must not be staged from the tertiary storage media. We assume for the second case that the data sets requested not be held in the HSM cache. This means the HSM-System must first stage the requested data sets from the tertiary storage media to the HSM cache and then the data sets are transferred to the RasDaMan system.

3 Super Tile concept

In RasDaMan the multidimensional data (multidimensional arrays) are organised as MDDs (Multidimensional Discrete Data). The MDDs are subdivided in regular or arbitrary tiles (see Figure 2). Tiling is a subdivision of multidimensional arrays into sub-arrays with the same dimensionality as the original array. Regular or aligned tiling is identical with chunking and is the most common tiling concept in array systems. Chunks of a multidimensional array all have the same shape and size and are aligned. Tiling is more general than chunking, i.e. sub-arrays do not have to be aligned or of equal size. More information about tiling strategies of RasDaMan can be found in the RasDaMan manuals [AK01a, AK01b] or the dissertation of Paula Furtado [Furt99]. Figure 2 shows arbitrary and regular tiling strategy of RasDaMan.



Arbitrary Tiling (2D)



Regular Tiling (3D)

Figure 2 Arbitrary and regular tiling strategy of RasDaMan

In RasDaMan tiles are the smallest units of data access. In order to manage these units in main memory, a limit on tile size is usually imposed. This limit could be set to the size of one database page, the smallest unit of data access in the database management system (Oracle or IBM/DB2). Tiles are stored as BLOBs (Binary Large Objects) in the database. It is necessary to choose different granularities for HDD and Tape access. On hard disk a data block could be the size of a Tile (typical 32 - 256 KByte) while on a tertiary storage media (e.g. magnetic tape) a data block should be much larger (more than 100 MByte). The larger data blocks on magnetic tape consequently have a better transfer rate (2 - 20 MByte/sec) and minimise the expensive exchange and positioning costs of magnetic tapes (20 – 180 sec). The size of data blocks on tertiary storage media must be multiples of RasDaMan MDD tiles. Such a multiple amount of tiles are called Super-Tiles. The Super-Tiles are the granularity of storing/reading RasDaMan data from tertiary storage media.

The computation of the Super-Tiles (of one MDD/collection) will only be done, if the MDD/collection should be exported to tertiary storage media. To get intra Super-Tile clustering we maintain the clustering of tiles given by the R+ tree (multidimensional index [GG98]) of which is used as index structure in RasDaMan. The conventionally R+ tree of RasDaMan was extended to handle such Super-Tiles which are stored on tertiary storage media. This means that information which tiles are stored on HDD or tertiary storage media must be integrated into the index. It is a good possibility that all tiles combined to a Super-Tile stored on tertiary storage media are tiles of the same subindex of the R+ tree of RasDaMan. The TS-Manager of RasDaMan knows on the basis of the structure of the multidimensional R+ tree, that all tiles below this subindex (subtree of a R+ tree node) are combined to a Super-Tile stored on the same tertiary storage media. The node of such a subindex is called Super-Tile node. We developed a special algorithm for computing Super-Tile nodes inside the R+ tree. For a better performance, the whole index of all data (hold in DBMS and tertiary storage media) is stored on hard disk. For 1 TByte of stored data the index will need approximately 100 MByte (0,01% of 1 TB) hard disk space.

Figure 3 depicts an example of the R+ tree index of one MDD with the corresponding Super-Tile nodes. Only complete nodes of the R+ tree can become Super-Tile nodes (dark green dots in Figure 3). This means that all tiles (red dots) of the included leaf nodes (in R+ tree data is only stored in leaf nodes) of one Super-Tile node are combined to one Super-Tile (light green) with intra Super-Tile clustering. A a consequence Super-Tiles can only be multiples of one tile.

A general restriction of the Super-Tile algorithm is that only one Super-Tile node may be contained in one path of the R+ tree. For the detection of Super-Tile nodes the predefined size of the Super-Tiles is used. The user can define his own suitable and optimised Super-Tile size with the configuration file TertiaryStorage.conf. If no Super-Tile size is defined in the configuration file a default size of 200 MByte will be used.

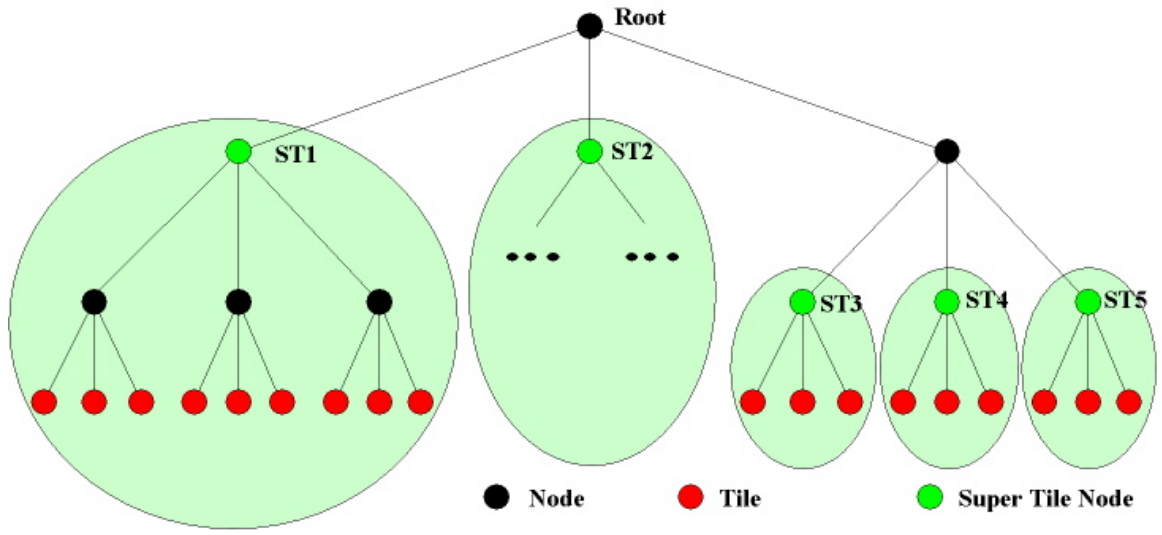


Figure 3 R+ tree index of one MDD with Super-Tiles

Super-Tile nodes can be on everyone of several levels of the R+ tree. In the example of Figure 3 we have 5 Super-Tile nodes (ST1 - ST5). The Super-Tile nodes ST1 and ST2 are on the second level of the tree and the corresponding Super-Tiles include 9 Tiles (red dots). The Super-Tile nodes ST3, ST4 and ST5 are on the third level of the tree and the corresponding Super-Tiles include only 3 Tiles.

4 Strategies for reducing TS access time

In the following we want to discuss four strategies for reducing tertiary storage access time. These four strategies are clustering, caching, scheduling and prefetching.

4.1 Clustering

Clustering is particularly important for tertiary storage systems where positioning time of the device is very high. The clustering of data sets reduces the positioning and exchange time of terti-

ary storage media. Data clustering uses knowledge about the expected query types and their execution probabilities to compute the storage order. Given a set of data items, it is important to identify the subsets of data items that are accessed together. Clustering uses the neighbourhood of the spatial location of tiles of data sets. Clustering of tiles according to spatial location in one disk or storage systems (tape, CD) provides an additional level of preservation of spatial locality, which is important for the typical access to array data. The clustering of data sets on storage system reduces the exchange and the access time, because a user query often requests data from a range query, which contains tiles neighbourhood. RasDaMan uses the R+ tree index for realising the access to stored MDDs. The R+-Tree of RasDaMan predefines the clustering of the stored MDDs. With the developed Super-Tile concept we can differ intra Super-Tile clustering and inter Super-Tile clustering. The implemented algorithm for computing the Super-Tiles (see chapter 3) maintains the predefined clustering of subtrees (of Super-Tile node) of the R+ tree index. Inside one Super-Tile we have clustering, i.e. neighbourhood of the spatial location of the included tiles. The export algorithm realise the inter Super-Tile clustering within one MDD. The several Super-Tiles of one MDD are stored to tertiary storage media in the clustered order (predefined R+ tree clustering). Inter and intra Super-Tile clustering of tiles stored on magnetic tape is shown in Figure 4.

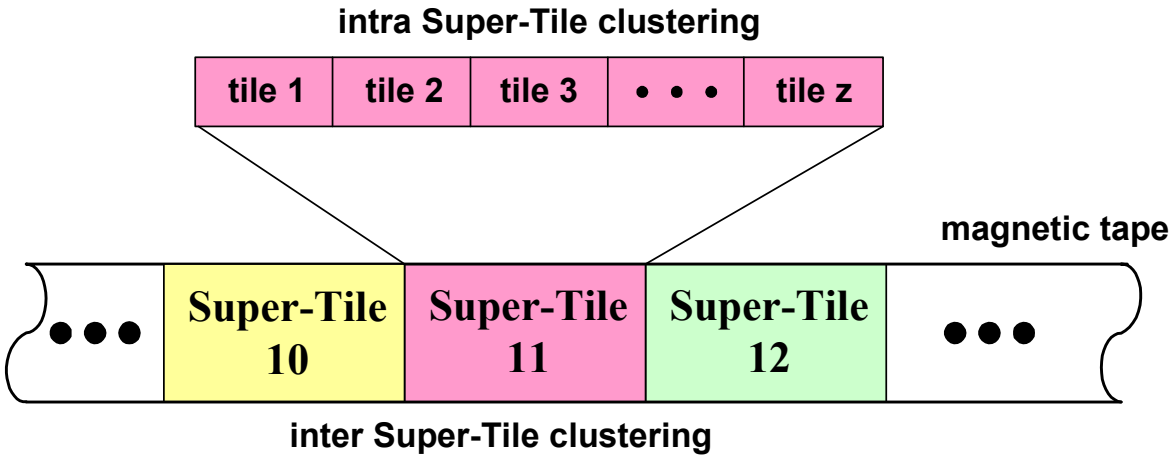


Figure 4 Inter and intra Super-Tile clustering of tiles stored on magnetic tape

4.2 Caching

The general goal of cache systems is to minimize expensive loading operations from slower storage levels (e.g. tertiary storage media). Caching of data sets held on tertiary storage media

reduces time expensive tertiary storage media access. For data (Super-Tile granularity) held on tertiary storage media the DBMS is used as hard disk cache area. In Figure 5 you can see the relational DBMS Oracle with the hard disk storage device.

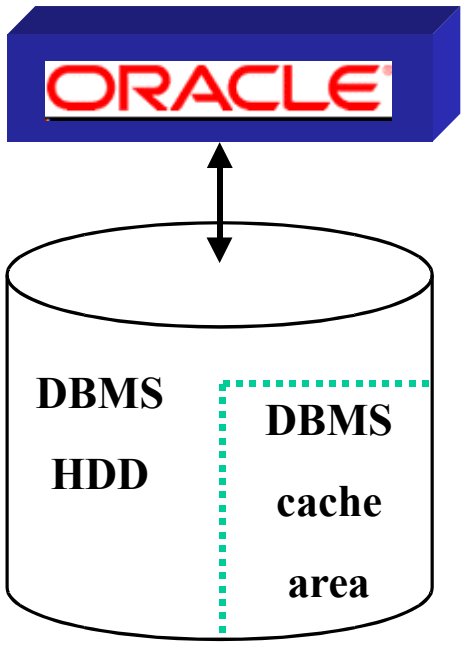


Figure 5 Relational database system with DBMS cache area

In the RasDaMan version with tertiary storage connection the relational DBMS is divided into two areas. The original part of the DBMS system (DBMS HDD) and the DBMS cache area for caching of data held on tertiary storage media. The original part of DBMS system holds the data sets which are only stored on hard disk (i.e. how data sets are held in the current version of RasDaMan). In the tertiary storage version of RasDaMan requested data sets held on tertiary storage media are migrated to the DBMS cache area. This organisation of the relational DBMS hard disk into two parts is a logical organisation. Physically the data sets held on hard disk (DBMS-System in Figure 5) and the data sets migrated from the tertiary storage media (DBMS cache area in Figure 5) both hold in the same table of the relational DBMS. This cache for the TS data is persistent. The advantage is, that the data held on tertiary storage media must not be imported to the DBMS cache area for every request. If the data is requested, it would be retrieved from the tertiary storage media to the DBMS cache area if the data are not already be hold in the cache. The import of the requested data from the tertiary storage media is extremely expensive because the transfer from TS-System to HDD is slow (media loading time, media exchange time, media rewind time, transfer rate, etc.). The second request to this date is very fast because the data already hold on the DBMS. The TS-Cache-Manager only evicts data (Super-Tiles) from the DBMS cache area if necessary (the upper limit of the cache size is reached). When and which data (Super-Tiles) should be

evicted from the DBMS cache area, if new data needed for a request? At the moment the eviction of Super-Tiles must be done manually by the user (start a query named TSeviction). In the near future the eviction will be done automatically by the TS-Cache-Manager. The TS-Cache-Manager can use replacement strategies to evict data from the DBMS cache area. Several replacement strategies e.g. LRU (Least Recently Used), LFU (Least Frequently Used), FIFO (First In First Out), CLOCK and others must be tested with the tertiary storage devices. The general goal of these polices is to substitute only the data (Super-Tiles) with minimal expectancy for reuse. In this direction optimisation algorithms for tertiary storage systems must be found.

4.3 Scheduling

Scheduling for tertiary storage media means the optimisation of the media read order. This optimisation reduces expensive media seek and media exchange operations. The focus is on scheduling policies that process all requests on a loaded medium before exchange it from the loading station of the robotic library. This optimisation is already supported by the HSM-Systems (e.g. UniTree, SAM). If a RasDaMan query needs several Super-Tiles from the tertiary storage system the HSM-System itself optimise (reducing media seek and exchange operations) the request order of the needed Super-Tiles. HSM-Systems try to transfer all requested Super-Tiles, which are stored on the same media in one transfer process without rewinding the tape. RasDaMan itself cannot influence the HSM-System optimisation of the media read order of data. A possibility how RasDaMan can schedule the media read order is to optimise the request order of the Super-Tiles needed from tertiary storage media. That means the import algorithm optimise the order of needed Super-Tiles before importing the Super-Tiles from the tertiary storage system (of one request). This optimisation of the import order of Super-Tiles is not yet implemented in the RasDaMan kernel. According Figure 4 we must import the Super-Tiles in order of they are stored on magnetic tape. The import order of the Super-Tile would be Super-Tile 10, Super-Tile 11 and Super-Tile 12 if only these three Super-Tiles are needed for one request.

4.4 Prefetching

Prefetching means to load those data (Super-Tiles) into the cache, which is supposed be needed in the near future. In RasDaMan prefetching can be done by examination of the query queue of actual and future requests. If the actual query needs Super-Tiles from one MDD and a future query of the query queue also needs several Super-Tiles from the same MDD all needed Super-Tiles will be imported at the same time into the DBMS cache area. This reduces expensive media seek and exchange operations. Further optimisation can be achieved by combining prefetching and scheduling algorithm. In a first step the prefetching algorithm combines all Super-Tiles

of the same MDD requested for the current and future queries of the query queue. Then the scheduling algorithm optimises the order of the needed Super-Tiles computed by the prefetching module. Generally this means that the scheduling algorithm will be extended to optimise the import order for more than one request. This feature is not supported in the current RasDaMan version and probably will be realised in future.

5 TS Interfaces for users and system administrator

The whole complexity of the RasDaMan storage hierarchy (Cache, HDD, HSM-System, TS media) is hidden from the end user. He only needs knowledge about the application interface (e.g. RasQL). The end user should not need any information about where the data sets are stored (HDD, i.e. database system cache area or TS media) in order to formulate the query. If the user sends his request to the RasDaMan system the tertiary storage manager reads the data sets from DBMS or imports the required data sets from the TS media. Only the response time of a query is different of requests on data sets hold on HDD or TS media. In near future the end user get a warning if the required data must be loaded from tertiary storage media. Summarising we can say that end users can handle the TS version of RasDaMan like the original RasDaMan version.

The RasDaMan system administrator should have more information about the system and the data. The tertiary storage RasDaMan system need information about the hierarchical storage system (HSM), the size of Super-Tiles stored on TS media, etc. The administrator must have also information about the data structure (data format) he wants to import into the RasDaMan system. He must be able to use the insert utility to save the HPC data to the RasDaMan system. He must also decide where the data should be stored (HDD or TS media). He can also decide which tertiary storage medium[2] should be used (DLT, Exa-Byte, DDS, MO, etc.) for the data sets of RasDaMan. The system administrator have several variation possibilities in access time, capacity and media live time of the tertiary storage media. More information of the TS interface for the system administrator can be found in [Rein01b].

[2] If supported by the HSM-System

6 Summary

6.1 Achievements

In this chapter the achievements of the tertiary storage integration into RasDaMan will be summarised. Large-scale scientific simulations, experiments, and observational projects of the ESTEDI high performance computing (HPC) partner, generate large multidimensional data sets and then store them permanently in an archival mass storage system (tertiary storage systems). Typically, these data sets are stored on up to hundreds of magnetic tapes, cartridges, or optical disks. Whereas data sets in RasDaMan are stored on hard disk (HDD), e.g. RAID systems.

One bottleneck of RasDaMan was, that RasDaMan was not originally designed to use tertiary storage media for holding the enormous size (hundreds of Tera Bytes) of the HPC data sets. On basis of this bottleneck one result of the ESTEDI project is to realise the connection of a tertiary storage system to the multidimensional database system RasDaMan.

The connection of a tertiary storage system to RasDaMan is realised by exploiting existing Hierarchical Storage Management Systems (e.g. UniTree or SAM), which allows a high degree of flexibility (e.g. import/export of data sets over the internet, usage of specific optimisation of existing Hierarchical Storage Management System, etc.).

To make the connection of the TS-System to RasDaMan operational, the RasDaMan tiling concept has to be adjusted appropriately. In RasDaMan multidimensional data are subdivided into tiles as storage units. These tiles are stored as BLOBS (binary large objects) in the underlying relational database. Typically, tiles (BLOBS) range between 32 and 256 Kbytes. Data sets stored on tertiary storage media (e.g. magnetic tape) should be much more larger (more than 100 MBytes) in order to achieve better transfer rates (2-20 MBytes/sec), and to minimise the expensive exchange and positioning costs of magnetic tapes (20-180 sec). To this end the so-called Super-Tile concept was developed: the size of one Super-Tile is a multiple of a RasDaMan Tile. Now Super-Tiles form the export/import granularity of storing/reading RasDaMan data from tertiary storage media. More details of the Super-Tile algorithm can be find in [Rein01b].

The integration of a TS-System into RasDaMan based on the Super-Tile concept encompasses:

- Implementation of a Interface to Hierachical Storage Management System (e.g. UniTree, AVM).
- Super-Tile Management in RasDaMan (extension of used R+-trees as index structure) including a special feature where the size of computed and stored Super-Tiles is very close to the predefined size of Super-Tiles.
- Specific Super-Tile algorithm, which computes Super-Tiles. Super-Tiles are partition of the RasDaMan data space and are granularity for import/export data sets from/to Hierarchical Storage Management System.
- The Super-Tile algorithm also support intra and inter Super-Tile clustering of multidimensional data.

The concept described of integrating a TS-System into RasDaMan is fully implemented in the RasDaMan kernel. The tertiary storage RasDaMan version is integrated into RasDaMan version 3.5 and runs on SUN Solaris 2.7. The relational database system must be Oracle 8i.

Extensive Tests of the export/import of Super-Tiles, the Super-Tile algorithm, and the corresponding intra Super-Tile clustering (also taking into account different sizes of Super-Tiles) have proven their workability. The tertiary storage version of RasDaMan is stable for all tested queries and delivers the correct set of data to the client application (e.g. Rview).

After an evaluation phase we have improved the performance of the Super-Tile algorithm by about 30 percent. For details of the integration of a TS-System into RasDaMan please refer to [Rein01b].

6.2 Outlook

The new tertiary storage features will be integrated into the new version 5.0 of RasDaMan as soon as possible. This will enable the evaluation of the new tertiary storage capabilities by ESTEDI HPC partners. We will also adapt the actual tertiary storage version of RasDaMan (Unix operating system) to Linux operating system and extend import/export functionality for the database system IBM/DB2.

An FTP client will be integrated into the RasDaMan kernel. This will improve the flexibility of the connection of RasDaMan to Hierarchical Storage Systems since it will allow the export/import of RasDaMan data to tertiary storage devices over Internet by FTP.

The most challenging task for the future will be the implementation of a specific RasDaMan cache manager module for the relational DBMS cache area with different replacement strategies taking into account the Super-Tile concept. Caching of data stored on tertiary storage media will reduce the number of tertiary storage accesses, and thus the expensive access time to tertiary storage. Besides the proof of workability, tests and evaluation of the new cache manager together with the tertiary storage facilities of RasDaMan will allow HPC partners / users to choose the optimal Super-Tile size for their specific applications.

7 References

[AK01a] Active Knowledge: "RasDaMan Documentation Version 5.0", München, 2001

[AK01b] Active Knowledge: "RasDaMan C++ Developers Guide Version 5.0", München, 2001

[Furt99] Furtado P.A.: "Storage Management of Multidimensional Arrays in Database Management Systems", PhD Thesis, Technical University Munich, 1999

[GG98] Gaede Volker, Günther Oliver: “Multidimensional Access Methods”, ACM Computing Surveys Vol. 30, No. 2, 1998

[Rein01b] Reiner Bernd: “End User and Administration Documentation of tertiary storage access with RasDaMan”, ESTEDI paper, FORWISS TU-München, 2001