

Schema for Object Oriented Dynamic Query Optimization and Modified Query using Mathematical Function Oriented Techniques in Databases

Rahman Shariff^{1,2}, Moussa Demba¹

Faculty of Computer Science & Information, Aljouf University, Kingdom of Saudi Arabia¹

School of Information & Communication Technology, Asia E University, India²

Abstract: In this paper we are going to discuss about the Importance of Query in Database, how helpful was it for retrieving data from Database, and it is so complex while in Relational Database. In this study we are going to discuss about Schema used for Dynamic Schema for Object Oriented Dynamic Query Optimization and Modified Query using Mathematical Function Oriented Techniques in Databases, mostly use SQL for retrieving data, some researcher make research on this field for introducing new model and a design of a personal specification query language using Mathematical function objects in a 3 dimensional space, called Dynamic Dimensional Query Language(DDQL). where an end user recovers updates and deletes Dimensional and Multidimensional information regarding the objects of their exact Dimensional locations. Dynamic Dimensional Query Language as an special query languages, which is built using object-oriented database system (OODBS), and works on database 'view' that gets information about dimensional characteristics of the objects and relationships among the objects. We also see a two level data modeling method as a schema for design and implement application based query languages

Keywords: Dynamic Query Optimization, Analyses, Object Data Model, Mathematical Function, Mathematical Object Data Model

I. INTRODUCTION

Dynamic Dimensional Query Escalation DDQE method explain the technique that follow for retrieving data using Grid Layout Techniques and the Second technique using Object Oriented one. Grid database [1] is a fresh research area joined with grid and database techniques, Grid database nodes has the character of dynamic and distribution, With the progress of Globus toolkit[2] and the standard of GGF, GSA-DAI [3] for Grid Database, New researches have paying attention on query handling technology in grid database, Example OGSA-DQP[4] paying attention on query processing technology in grid database, they used static query processing method technique, and the query request was translated and run as the order they appear in grid database. Being dynamic of nodes in grid database, some passing values in processing query and development may not accurate, not available or may be altered during the query processing in grid database[5], maximum schedule of query plan at the beginning of query processing may be costly after the changing of parameter, query optimization processing should be dynamic to go with the original condition of query processing in grid database, that is adjusting dynamic query processing[6].

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

LITERARE REVIEW

In present situation, question on choosing database for interface using Object-Oriented DATABASE MANAGEMENT SYSTEM [7]. Most commonly available ODDBS's combine database deals into a particular object-oriented programming language in a style that the combination of the two is as 'Logical' as probable. Many of these schema have query languages, but they are like retrieval method, that means discover an object by set its attribute value. In extension, the query language is build up to integrated well into the ODDBS object-oriented programming system. Query languages are more confining in data processing as compare to SQL. Example, ORION query can recover objects in only one object series [8]. New researchers argue that like an RELATIONAL DATABASE MANAGEMENT SYSTEM, an ODDBS must give only one interface like SQL query language [7]. the analysis methodology used here should be updatibility. but, it is not easy for the ODDBS to give a universal query facility that gives the requirements of the end user, and is well integrated into the end programming system. The common query facility presented by the ODDBS is separate from a truly user-friendly query language.. The common query/Data handling Language facilities of DATABASE MANAGEMENT SYSTEM use the query facility offered by ODDBS it gives an additional idea. But there major differences between a personal specification query facility and the view method. , this personal specification query language can include correct features of a particular application or those features general to various classes of applications, in this manner the common query language of an ODDBS is not possible. Really, this query language must be connected with an application oriented data model, where data may be controlled by this language. so, there may be many dissimilar query languages personal specification for dissimilar application areas, this query languages, is called as application oriented query languages, which provide a level of application oriented concept.. The object-oriented concepts represent in an ODDBS, Example., class series, difficult object and representation of object behavior, seriously simplify development of these query languages. Here we mentioned 2 level data design way, a schema for plan and execution hopefully that is suitable to all application oriented query languages.

II OBJECT ORIENTED DYNAMIC QUERY OPTIMIZATION IN DATABASE

There are a number of research going on flexible dynamic query dealing and development. Because of not accurate or unavailable of parameters , SWAP[9] here focused on how to get accurate parameters in dynamic location of grid , here get parameters such as servers workload , speed of program , etc by locating Eddy in nodes of grid. SQL Server's Auto admin feature has a capability for generating histograms on sub plans within a query , for use in computing further cost estimates[10]; here see the Reference [11] paying concentration on finest scheduling of query plan, and CPA(Critical Path & Allocation) design was offered to deal out the subquery to grid nodes, the subquery in CPR algorithm was planned as the style of computing function , GRADS[10] and Condor [11] make model of query plan based on CAG (Conducted Acyclic Graph) ,reference[11] proposed min-min algorithm to schedule Subquery in query plan .

Algorithm which scheduled Subquery as the style of computing function in grid has following shortcoming: A Subquery has different character compared with computing function, the resourced used in computing function may involve computing resource of node such as memory, energy of CPU and etc to finish the computing processing, so computing function can be assigned to any node who has adequate computing resource for computing function ; Subquery has different situation , the resource used by Subquery mainly involve data resource of database in node, there is data relation between Subquery and node, so Subquery can not be scheduled as the manner of computing function ,otherwise , it will bring down efficiency, algorithms in reference [9] and [11] overlooked the data relation between Subquery and node. Subqueries come from the same global query , there are rich relation between Subqueries, model of Subquery plan based on CAG in reference[10] and [11] can not depicted rich relation being the deficiency of CAG, it can only depict the simple relation such as predecessor and descendant, subtle relation such as concurrency, embrace and etc can not be depicted in the model of Subquery plan. It is insufficiency for dynamic selection processing being deficiency of query plan model. Based on above analysis, we will propose dynamic selection method.

III. ANALYSIS OF DYNAMIC OPTIMIZATION

As we have find Parameter of cost model in reference [10], but The cost based on cost model reference [10] can not represent the cost spent in current environment , the reason is inaccurate value of parameter because of fluctuation of grid environment, we can get the accurate value of parameter we go for the further development by processing of subquery. In order to get the accurate parameter we go for process of assigning the subquery to the best node, Paper partition processing of subquery into different steps explain in a diagram mentioned below as Diagram 1 which contain different stages. from this different stages we find the proper node to be assigned .

a. Experiment Stage:

In this stage I take a little part of data to complete transmission and join operation, An algorithm given below can be used to test the performance of transmission and join operation, and find the proper node.

b. Combine Stage

In this stage assign the subquery to the proper node what we find in experiment stage and complete the processing of rest of data in the node, this is for dynamic optimization

c. Connecting Stage

In this combine stage we are going to finish the remained operation , that is merging with all the output and result what we get or generated from the Experiment stage and combine stage and guarantee the correctness of whole processing The main purpose of dynamic optimization is to choose the proper node for subquery among the relational nodes, the proper node would make the cost spent by subquery lowest, by using the traditional query technique, except for time cost spent by operation of subquery such as Select, Project and Join ∞ , there is transmission cost for needed data to be transmitted from other nodes to proper node, Time cost spent by subquery SQ_i in proper node P_j include the transmission cost T_{comm} , Join cost T_{join} , Selection cost T_{select} and Project cost $T_{project}$, and we design cost model as step 4.

$$T_{P_j} = T_{comm} + T_{join} + T_{select} + T_{project} \rightarrow 1$$

Time cost spent by operation of subquery has the proportion to energy of node like CPU, I/O and etc, and time spent by join operation is more greater than time cost spent by Select operation and project operation, so the time spent by $T_{select} + T_{project}$ in model (□4), rectify model (□4), the cost model can be simplified as follow

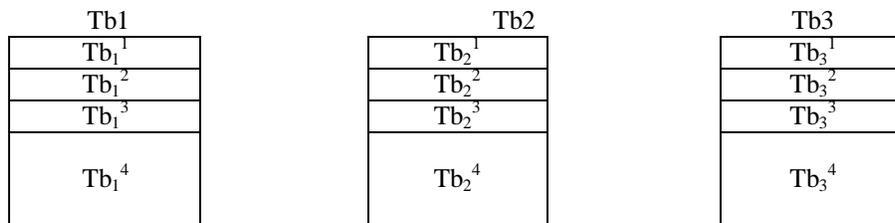
$$T_{P_j} = T_{comm} + T_{join} \rightarrow 2$$

d. Different Stage Algorithms

Different stages dynamic optimization algorithm is explained below by assuming with three nodes as N_1, N_2, N_3 to be subquery SQ_i , here Tb_1, Tb_2, Tb_3 is the table to be processed in different node.

Experiment Stage Algorithm

In this stage we are going to separate the triple of Tb_i in to 2 groups , quantity of triple in one group is more than enough another and name it as Multiple Group MG, and another name it as Single Group SG, SG was separated in to three quantity of group further, denote as Tb_1^1, Tb_1^2, Tb_1^3 and denote MG as Tb_1^4 and separate T_2 and T_3 as the same manner of T_1 , as shown in partition diagram(1) below.



Partition Diagram(1)

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

Different groups processed as follow

- Transmit Tb_2^1 and Tb_3^1 to node N_1 ,
make combine operation: $Tb_2^1 \infty Tb_3^1 \infty Tb_1^1 \rightarrow 3$
- Transmit Tb_1^2 and Tb_3^2 to node N_2 ,
make combine operation: $Tb_1^2 \infty Tb_3^2 \infty Tb_2^2 \rightarrow 4$
- Transmit Tb_1^3 and Tb_2^3 to node N_3 ,
make combine operation: $Tb_1^3 \infty Tb_2^3 \infty Tb_3^3 \rightarrow 5$

e. Combine Stage

In this stage we are going to monitor the different time cost spent by operation in nodes N_1, N_2, N_3 , let as assume Transmission Cost as T_{comm} and time cost as T_{join} and of table $Tb_2^1 \infty Tb_3^1 \infty Tb_1^1$ as mentioned above as (□3), and calculate the time cost T_{N1} through cost model formula given below(□2) and also calculate the time cost Tb_{N2} and Tb_{N3} as the same manner. Choose the lowest cost

$$\min(T_{N1}, T_{N2}, T_{N3})$$

and confirm that which node is proper node N_i ($0 < i < 4$), assume N_2 is a proper node, transmit Tb_1^4 and Tb_3^4 to node N_2 , combine operation: $Tb_1^4 \infty Tb_3^4 \infty Tb_2^4$

f. Connecting Stage

We finish the processing of

$$Tb_1^i \infty Tb_2^j \infty Tb_3^k \quad (i=j=k)$$

in Experiment and combine stage and the remainder combine operation of

$$Tb_1^i \infty Tb_m^n \infty Tb_k^1 \quad (i=j=k)$$

should transmit the data of

$$Tb_1^1, Tb_1^2, Tb_1^3, Tb_3^1, Tb_3^2 \text{ and } Tb_3^3 \text{ to node } N_2$$

to finish combine operation of

$$Tb_1^1 \infty Tb_3^1 \infty Tb_2^2, Tb_1^1 \infty Tb_3^1 \infty Tb_2^3 \dots, Tb_1^3 \infty Tb_3^4 \infty Tb_2^4,$$

that is

$$Tb_1^j \infty Tb_m^n \infty Tb_k^1 \quad (i=j=k),$$

from this we merge the result of Experiment stage, combine stage and connecting stage to complete the final processing

IV. MODIFIED QUERY USING MATHEMATICAL FUNCTION ORIENTED IN DATABASE

Most of the functions define logical relationships of Object Data's, example. on left of and on-Upper-of, distribute some less functions describe Mathematical function relationships. These less, Mathematical function-oriented be supposed to be well-known from the upper-level(more) Object Data-oriented function. default by general approach, these Mathematical function may be built-in in a library functions, the way choose by the OSI model. This library functions may be used to support the improvement of DDQL, together with a outlook of the Object Data Model to the end user. The outlook of the Object Data World, where DDQL is the data definition and handling language, is called Object Data model. a less function data model, called Mathematical function object data model, where the Mathematical function are defined. With this way, the implementation of the DDQL is more free of the underlying ODDBS. Thus, the Object Data can be defined is most suitable for the end user, and free of any constraints introduced by the basic ODDBS. Example, Object Data level object may not find at the Mathematical level. While an ODDBS may be able to give consideration during its class series design, we are able to give more detailed graphical representations of information to show some relationships within the data among objects on the 2 levels. besides, the end user can also

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

choose to go out low-level, unnecessary information about the Object Data . At last, a obviously defined data model at the ODDBS level called Mathematical level will simplify the function of maintaining, modifying or extending the DDQL. In this System planning it contains 3 layers. The end user pose query at the Top Layer using DDQL and application program can run using DDQL as Application program interface , the centre is D-Kernel called as software layer contains query processor and passing object manager. The supervisor of unique identifier generation and management of all Object Data is done by passing object manager .The techniques and plans for query processing, and essential mathematical function in query, like our relationship, computing of object data for the Mathematical function object data model are develop by query processor. The lower layer is ODDBS that contain interface part called function implementer that permit the excellent method to perform each function in techniques and plans for query processing designed by query processor in D-Kernel.

The Object Data model mainly consists of Object Data's, a set of linguistic geographical functions, and some type hierarchies, through which the topological relationships of Object Data objects are defined or derived. DDQL is the language for handling of Object Data objects. The Mathematical function object data model, contains Mathematical function objects which are the actual geographical representation of the Object Data's, and a set of Mathematical function that retrieve, update and compute for Mathematical function objects. most probably, the common query language of ODDBS could be the language for handling of Mathematical function objects, but we wish to use functions as the interface to Mathematical function object data model. These models explained in Object Data Model Section and Mathematical Function Object Data Model respectively.

V. OBJECT DATA MODEL SECTION

The object data model is a room like box where a number of commonly construct Object Data's are located. Each Object Data is (a) leave on the floor, or (b) hold from the ceiling, or (3) stacked over other Object Data. This room like box is easily seen by the user, or its pictures can be obtained by using 3D computer graphics and scene on screen. The user is wants to know the knowledge about the Object Data's, example., like sizes, and the geographical relationships with other Object Data's. Mostly, the user wants to use the objects in motions like rotate or move the objects in the 3D space. For this, a dynamic geographical query language (DDQL) is made ready to the user. , DDQL have similar capacity as a common record (or tuple) oriented query languages such as SQL, which recovers, updates, inserts and deletes.

Even though our aim and idea is to separate the view of the object data model from the underlying the view of Mathematical function , it is compulsory that the user's understanding of geographical locations of the Object Data's follows properly on Mathematical level. This show that the user knows some thing about the actual Mathematical representation of the objects data model. By luckily, we are able to build our models in such a way that the user infrequently needs to be familiar with the Mathematical representation details. One or some of exceptions appear when location between two Object Data's is calculated or measured. The return value may be understand by the location between the Object Data's centres, or centroids . from this case, the user recognize that the Object Data is stated or defined by a point on the Mathematical level. For example, we define one type series, which arrange objects by their shapes and combine with Mathematical properties. At the root of the type series is the object type object model, having a number of subtypes for objects of different Mathematical properties like , e.g. SPH, CUB, and PYR. The explanation for the object type , object model and their subtypes on are given below:

Ex 1

Obj Typ-: OBJECT DATA
Sup Typ: null
Functs:
Name: STRING
Vol: REAL
Total-area: REAL

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

SubType
SPH
CLY
CUB
PYR

Ex 2

Obj Typ CUB
Sup Typ: OBJECT DATA
Functs:
Upper: REC
Sides: REC
Bottom: REC

Ex 3

Obj Typ CLY
Sup Typ: OBJECT DATA
Functs:
Upper: DISC
Side: CYL-SUR
Bottom: DISC

Ex 4

Obj Typ REC
Functs:
Color: STRING
Leng: REAL

The type DISC and CYL-SUR are considered as Mathematical objects represent in Section Mathematical Funct: Object Data Model. We defined two types of functions: one is System built-in functions and one is other depended function(location, linguistic). System built-in functions are autonomous of application environment. The types of other depend function are given below as follows and explained

A. Location Function:

A location function defines the location between the centres of two objects. An example is Location(Object, Left-Wall)->REAL. This function will return the actual location of the centre Object, from the Left-Wall, where the latter is a system-defined object.

B. Linguistic Function:

A linguistic function describes a geographical relationship between two objects whereas their centres are not identical. A linguistic function describe geographical relationship between two objects that share the same point as their centres. Hence the geographical relationship is one of position. distinctly there cannot be such a relationship between two real objects. The main aim of this type of functions is to calculate the difference in positions of an object before and after a rotation around an axis through its centre. There will be further explanation on this in the analysis of our query language. An example of a rotation is: Backward(Object₁,90°)->Object₂ where Object₂, a realistic object, refers to Object₁, after being rotated in a backward direction by 90 degrees. The realistic object concept will be explained later.

In the article, handling of an object is generally considered as an operation applied to the object (Ex 5). In this paper, we are treating object handling differently. here we manage that the user allow to select the object for operation not only by its object ID and also by geographical or non geographical criteria, even though the operation is applied to an specific object. For example, the user allow to specify the Object Data next to that big green sphere for motion. This shows a database viewpoint. Naturally in a common database query language, for Example SQL, treat the database updating functions in the same manner as retrieval functions. so, we introduce these functions at the query language level, and it

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

clears user about the details involved in updating the geographical location of the object. We use realistic object concept to design an inter-face for the user to allow clear Function in to updating function at the mathematical level

There are 2 types of object motion in the space([12]): rendering and rotation. Rendering of an object results in motion of the object without changing the object's position. here we called term as i.e. Move. In order to describe correctly the new location (object position) where the object is to be moved to, we design a realistic object which is totally same as the object except that it is assumed to be situated in the desired new location. so, this new location can be started through a topological relationship between the realistic object and another object in the Object Data World, which may be the object to be moved. For example, if an object is to be moved 3 meters to the left, then the realistic object is started to be on the left of the object (through a linguistic function (explained above)) and at a location 3 feet away (through a location function). The general format for the command MOVE in query language is: MOVE (an known object) TO (an known realistic object). An example 6, will be given to clarify the points made here.

Rotating an object motion is treat in a same way as Move. Since any rotation of an object in space can be consider to be geographically equivalent to one sequence of one Move and one Rotate about a line through its centre, we only consider rotations of a Object Data about the X1, Y1 and Z1-axis through the centre of the Object Data. This means that the centre of the Object Data will not be moved during the rotation. The linguistic functions (Mentioned above) define all possible rotations. See Ex 6 for further illustration.

Some frequently used motions can be compress for the welfare for the user. For example, "lift up an object by 2 meters" is called as "move an object to a realistic object which is above the current object and is at location 2 meters from the current object." Another motions that result in changes in the Object Data World include Property motion and Retreat motion. These motions can be also considered as special cases of the motion Move. For the Property motion, it is equivalent to moving an object from the outside world to a exact location in the Object Data world using the realistic object construct. This construct is not needed for the Retreat motion, since the object simply abandon from the Object Data World. At last, we use the simple SELECT-WHERE Object Data construct for recovery of objects. The WHERE clause is also used to expressed the description of objects and realistic objects involved in a motion.

Ex 5:

Move the cylinder called 'Sample' to be on Upper of a cuboid with a Green upper face.

```
MOVE Object1 TO Object2,  
WHERE  
Name (Object1)='Sample'  
AND Upper_Of (Object3)=Object2  
AND Type (Object3,) =CUB  
AND Color (Upper (Object3,) )='Green'
```

Ex 6:

Rotate the cuboid which is on upper of another cuboid with a Green upper face, around in clockwise direction by 90 degrees.

```
ROTATE Object1 TO Object2,  
WHERE  
Upper_Of (Object3) =Object1  
AND Color ((Upper (Object3) ) )='Green'  
AND Type (Object3) =CUB  
AND Around_Clockwise (Object1 90°) = Object2
```

VI. MATHEMATICAL OBJECT DATA MODEL

The main aim of this mathematical object data model is to support the data design and querying activity at object level by defining Mathematical object and low-level mathematical calculating functions. We plan to determine the Mathematical

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

object data model as convenient to a common data model as much as possible. By this way all the Mathematical objects and computational operation enclosed in the data model can support different Object Data by object data model explained here. It is up to the D-Kernel to plan a personal specification interface (or data model) to fulfill the specific needs of the user.

In this paper we use the border representation model([13] and [14]) ,which consists of four types of object. faces, bends, borders and vertices. A front is a connected portion of the surface of the object enclosed by the border which consists of either bends such as circles or collection of intersecting (borders). Vertices are Interchange points of borders. , the benefit of this creation is easy to find out topological relationships. See, here we define which borders are attached to a face, which vertices are attached to an edge, and so on. Example, notice how a cylinder, is define in Example 3, mentioned above, can be built simultaneously by Mathematical function objects, which are explain as follows:

Example 7

Obj Typ DISC
 Sup Typ: PLANE
 Funct:
 Normal: VECTOR (POINT)
 Border: CIRCL
 Radius: REAL

Example 8

Obj Typ : CYL-SUR
 Sup Typ: CUR-SUR
 Funct: :
 Border: CIRCL

Border is defined for each Mathematical function object part of a cylinder. This functions supply a Function from a DISC and a CYL-SUR objects into the same CIRCL object, which supply the 'attach' for assembly. Many Mathematical functions required for functions operation at the Object Data level. Here see explanation with example to elaborate the character of these functions. Consider the Object Data level function Upper-Of(O_1, O_2), O_1 , and O_2 , being cuboids, the following Mathematical level functions are required

a. Alternate-Way(V_1, V_2)->BOOLEAN : Whether the two vectors V_1 , and V_2 (e.g. normal of faces of O_1 and O_2) are in alternate direction was determine by this function

b. Identical-Plane(P_1, P_2)->BOOLEAN: Whether the two planes P_1 , and P_2 , (e.g. faces of the two cuboids) are on the Identical plane, that is whether they are equivalent to each other and have zero location between them by this function.

c. Interchange(P_1, P_2 ,->REAL: This function dealings the size of the related portion of the two planes. If the two planes do not come across, a null value will be returned. When a Mathematical function object is moved, its geographical location has to be updated and repeated on the geographical database. The geographical location of an object is represented by the geographical location of its centre, which is assigned a value for its spatial coordinate, which consists of 3 real numbers for X1-, Y1- and Z1-axis respectively. Centre as a function maps a Mathematical object into a COORDINATE object. Thus update the geographical location of an object is equivalent to give a different value to its function Centre. Centre is defined at the root of the object series GEO-OBJECT, so that it is applicable to all Mathematical function objects defined in the series. To facilitate updates of geographical locations, we define the following functions:

d. Interpret(Obj,X1,Y1,Z1):Add values to the real variable X1, Y1, and Z1 to the geographical coordinate of the centre of Obj by using this function

e. Alter(Obj1,Obj2,a,d):The new geographical coordinate of the centre of Obj₁ rotated about a axis through the centre of Obj₂ by 90 degrees by using this function

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

Function

There are different types of Function to design data build between the Object Data object level and the Mathematical function object level. Otherwise the construct following at the Object Data object level need to be mapped into build at the Mathematical function object level, and reversed: (i) Object Data ID, (ii) Object Data, (iii) non-geographical function, and (iv) geographical function and DDQL query. For the remaining of this section, we will see each item separately.

A Object Data in the Object Data World is described by a POINT object at the Mathematical function level. The D-Kernel keeps a table that subordinate the object IDS for the Object Data's at the Object Data level with the object IDS of the related POINT objects at the Mathematical function level. We assign a special function, Centering, to represent this table, i.e. Centering(Object Data) □ CEN.CEN is a subtype of the object type POINT. The creative activity of this subtype differentiate object centres from vertices which belong to another subtype VERTEX.

Following, we are going to demonstrate how the whole Object Data or its creation at the Mathematical function level, can be recalled given its Object Data ID. See with an example a Object Data object which is cylinder. From Ex 3, we know that the functions Upper, Side and Bottom map the Object Data ID into the Mathematical function objects ID_s of 2 DISC and 1 CYLINDRICAL objects, accordingly. Therefore the faces of the cylinder are easily set. With the border representations of these faces, the Object Data can be collected together from these faces, since each face is gathered as an separate object with its own geographical coordinate. All the non-geographical, Object Data level functions that contain Object Data's in their I/O function arguments will become Mathematical function level functions, by applying the Centering function to all objects of Object Data types. For example, Name(Cone₁) → 'Sample' when stored in the underlying ODDBS will become

Name(Centering(Cone₁)) → 'Sample'.

Geographical function at the Object Data level must be Interpreted on individual basis by the query processor of the D-Kernel into a program that execute on the Mathematical function level. Likewise, a DDQL query will be processed by the D-Kernel to change of course into a program, except that more difficult techniques will be required if performance is an important requirement. However, optimization is not an point in this paper. We will show by way of the following example how a query can be converted into a program. All the functions in the examples are on the Mathematical function level and have been explained in Section Function Ex 5.

Ex 9:

Interpret the query in Example 6 into a (pseudo-)program that executes on the Mathematical function level.

Determine Object₁, and then convert it to CENTRE type via the Centering function

```
FOUND = FALSE
FOR each object O1 within the Obj Typ CLY,
DO:
IF Name(Centering(O1)) = 'Sample' THEN FOUND = TRUE
ENDDO
ENDFOR
IF FOUND = FALSE, THEN EXIT-PROG
Determine Object3
FOUND = FALSE
FOR each object O3, within the Obj Typ CUB,
DO:
IF Color(Centering(Upper( O3 ))) = 'Green', THEN FOUND = TRUE
ENDDO
ENDFOR
IF FOUND = FALSE, THEN EXIT-PROG
```

The centre C₂, of the realistic object Object₂, is L meters above the upper face of F, whose centre is denoted by C_F. Create a temporary POINT object C₂,

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

$$L = 1/2 * \text{Height} (\text{Side}(O_1))$$

$$\text{Centre} (C_2) = \text{Interpret} (C_r, X1 = +O, Y1 = +O, Z1 = +L)$$

Measure the differences in X1, Y1 and Z1-axes between the old and the new centre of the cylinder

$$X1T1 = X1(\text{Centre}(C_2)) - X1(\text{Centre}(C_1))$$

$$Y1T1 = Y1(\text{Centre}(C_2)) - Y1(\text{Centre}(C_1))$$

$$Z1T1 = Z1(\text{Centre}(C_2)) - Z1(\text{Centre}(C_1))$$

Update C_1 , the centre of O_1 plus the centres of its upper, bottom and side, which should move in the Identical direction and for the location C,

$$\text{Centre} (C_1) = \text{Centre} (C_2)$$

$$\text{Centre} (\text{Upper} (O_1)) = \text{Interpret} (\text{Upper} (O_1), X1 = +XT1, Y1 = +YT1, Z1 = +Z1T1)$$

$$\text{Centre} (\text{Bottom}(O_1)) = \text{Interpret} (\text{Bottom}(O_1), X1 = +XT1, Y1 = +YT1, Z1 = +Z1T1)$$

$$\text{Centre} (\text{Side} (O_1)) = \text{Interpret} (\text{Side} (O_1), X1 = +XT1, Y1 = +YT1, Z1 = +Z1T1)$$

VII. CONCLUSION

In this paper, we have defined two methodologies for dynamic structured query language in database management system, one is sub query method and another one is object oriented structured query language method. We represent the data relation between sub queries and nodes that was assigned to sub query. The paper further proposes three phases algorithm (Experiment stage, Combine stage and Connecting Stage) to dynamic optimization schedule to fit for the fluctuation environment. Another method in object oriented structured query language we have presented the design of an Object Data model query language. This query language includes characteristics that are helpful for treating objects motion in a 3-D geographical database. This analyzes positively to additional (general) database interfaces of Object Oriented Database Management Systems. See example the prototype Object Oriented Database management Systems by ([15]) whose model is related to accept one by this job. Its Object Structured Query Language is related to SQL, the Object Structured Query Language functions are restricted to property functions, that means objects attributes is not powerful. Iris has a primitive function called external function, which is fully supplied by user. With the realistic object method represent a position taken up by an Object Data in the space, we are able to describe the destination position in accurate. For that we use technique to include similar information in interface in less user friendly manner. Query languages are for ordinary user, functions and methods are for programmer. The experimental design of Dynamic structured Query language helps easy to be on a higher level than any basic database interfaces of an Object Oriented Data based management System. We have also presented a methodology for implementing Dynamic Structured Query Language. We consider this method helpful for planning and execution of other query languages direct for other applications field.

REFERENCES

- [1] Wang Shan Zhang Kunlong Grid database system in grid
- [2] I. Narang, V. Raman, C. Crone, L. Haas, T. Mukai, S. Malaika, C. Baru. and D. Wolfson Services for Data Access and Data Processing on Grids. GGF Document GFD.14, Global Grid Forum, 2003.
- [3] A. Anjomshoaa et al The Design and Implementation of Grid Database Services in OGSA-DAI. In Proceedings of UK e-Science All Hands Meeting, Nottingham, 2-4 September 2004
- [4] A. Mukhejee, M. Nedim Alpdemir, Norman W. Paton Service based distributed querying on the grid. ICSSOC 2003, First International Conference, Trento, Italy, December 15- 18, 2003, Proceedings, pages 467-482.
- [5] A. Gounaris, J. Smith, R. Sakellariou. P. Watson, A. Fernandes and N. Paton Distributed Query Processing on the Grid. In Proceedings of Grid Computing 2005, pages 279- 290. Springer, LNCS 2536, 2005.
- [6] Pedro Bizarro and Shivnath Babu Adaptive query processing in the looking glass. In CIDR, pages 238-249, 2005

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 2, February 2014

- [7] Stonebraker, Michael, et al Third Generation Data Base System Manifesto. Database Systems Study Group in 1989 to gather information on Object of the third DBSSG workshop was to foster communication among standards groups, in Atlantic City, NJ , 1990
- [8] Banerjee, J. et al Queries in Object-Oriented Databases. Proceedings 4th International Conference on Data Engineering, pages 31-38. L.A., 1988
- [9] Wee Hyong Tok , Beng Chin Ooi, Kian-Lee Tan, and Yongluan Zhou An adaptable distributed query processing architecture. Data Knowl. Eng., To appear, 2005.
- [10] Nicolas Bruno and Surajit Chaudhuri Exploiting statistics on query expressions for optimization. In SIGMOD 2002, Proceedings ACM SIGMOD International Conference on Management of Data, June 3-6,2002
- [11] K.Mandal, J. Jain, Blythe, S.Deelman, Y.Vahi, and E.Gil Task scheduling strategies for workflow- based applications in grids Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on 9-12 May 2005 On page(s): 759- 767 Vol. 2
- [12] Foley, J.D. and Van Dam, Andries Fundamentals of Interactive Computer Graphics. Addison-Welsey, 1982
- [13] Chiyokura.H Solid Modelling with DESIGNBASE. Addison-Welsey, 1988.
- [14] Samet.H The Design and Analysis of Spatial Data Structures. Addison-Welsey, 1990
- [15] Fishman, D. H. et al Iris: An Object-Oriented Database Management System. ACM Transactions on Office Information Systems 5(1):48-69,1987.