Adapted Extremal Optimization For Materialized Views Selection

Samiha BRAHIMI
Misc laboratory
University Mentoury of Constantine (Algeria)

Dr. Mohamed-Khireddine KHOLLADI
Misc laboratory
University Mentoury of Constantine (Algeria)

ABSTRACT

With the development of databases in general and data warehouses in particular, it is now of a great importance to reduce the administration tasks of data warehouses. The materialization of views is one of the most important optimization techniques. The construction of a configuration of views optimizing the data warehouse is an NP-hard problem. On the other hand, the algorithm called extremal optimization is used to solve complex problems. In this paper, we propose a new adapted extremal optimization (AEO) for the materialized views selection problem.

General Terms

Data ware housing, algorithms, and optimization.

Keywords

Materialized views selection, data warehouses, extremal optimization and query's optimization.

1. INTRODUCTION

The variety of data sources that we can find in the same context (subject) calls the concept of data warehousing. Data warehouses collect the data from different sources, organize and store them in order to help the management of their data. It is clear that these DW are giant, which implies auto-administration tasks to optimize them. Among the technique of optimizing DW, we have the materialization of views. Our problem consists of constructing a set of materialized views optimizing the workload assigned to the data warehouse.

During the past few decades, studies on the combination of statistical mechanics or physics with computational complexity in term of analyzing and solving them have been the interest of researchers in both physics and computer sciences. Extremal Optimization (EO) is the makes the link between statistical physics and computer sciences; it is based on the principle of Bak-Sneppen model of evolution. EO has proved its performance in diverse domains, which encourages us to apply it on the problem of materialized views selection, especially because it is an NP-hard problem.
The rest of the paper is organized as follows; the second section represents a state art study, the third is the conception of our approach and the last is the conclusion and the perspective.

2. ART STATE

In this section we represent an art state study about the Bak-sneppen model, the extremal optimization algorithm, the AND-OR view graph and the materialized view selection.

2.1 Bak-Sneppen Model

The Bak-Sneppen model [1] is a model of evolution between species. Its major characteristic is that it considers the whole ecosystem and the co-evolution of many different species rather than focusing on single species.

A "fitness" value between 0 and 1 associated to the species which are located on the sites of a lattice (or graph. At each time or step (iteration), the one species with the worst fitness (poorest degree of adaptation) is selected to be updated randomly, having its fitness replaced by a new random value drawn from a flat distribution on the interval [0, 1]. This corresponds to the natural process of species' development or where a species is replaced by another one. In food chain for instance, the no species lives alone but depends on its successors and predecessors. Bak and Sneppen consider this by arranging the species in a one dimensional line. If one species is mutated, the fitness values of its successors and predecessors in that line are also set to random values [1].

Therefore, all of the species connected to the "weakest" have their fitness affected (replaced by new random numbers as well). After a sufficient number of iterations, the system reaches a highly correlated state known as self-organized criticality (SOC) [2]. Almost all species have reached fitness above a certain threshold. These species possess punctuated equilibrium [3]: only one's weakened neighbor can weaken one's own fitness.

2.2 Extremal Optimization

EO is an evolutionary meta-heuristic oriented local search proposed by Boecher and percus [4], [5], [6], [7] inspired from statistical physiques and the model of co-evolution between species in order to find high quality solutions for hard problems.

As in Bak-Sneppen model, EO merely updates those variables having an extremal (worst) arrangement in the current configuration, replacing them by random values without any improvement of their performance. Large fluctuations allow escaping from local optima.

In order to make EO easier and more understandable, we will compare it with another well-known method such as the genetic algorithms (GA) [8]. First, a GA has a set of parameters to be tuned like the size of population, probabilities of reproduction and number of generations; however, in EO there is only one parameter to be tuned. Second, in EO the fitness value is not calculated for each structure that represents a solution (individual or chromosome) as in a GA but for each component of the structure which is represented by a species; each species is evaluated according to its contribution in obtaining the best solution. Third, EO works with a single solution instead of a population of solutions as in GA. Last, EO replaces the worst components for the next iteration; in contrast, GA promotes a group of elite solutions.

As it is mentioned above, EO has only one species parameter that is often referred to as τ [9]. This parameter is used probabilistically to choose the component value to be mutated at each iteration of the algorithm. The algorithm ranks the components and assigns to them a number from 1 to n using the fitness of each one (where n is the number of components). Therefore, the fitness must be sorted from the worst to the best. The probability is calculated as follows:

$$\boxed{P_i = i^{-\tau} \quad \forall i \; 1 \leq i \leq n \quad \tau \geq 0 \qquad (1)}$$

Where:
n is the total number of components evaluated and ranked, and
P_i is the probability that the i^{th} component is chosen.

Algorithm 1: Standard EO pseudo-code

```
Generate an initial random solution $X=(x_1; x_2;…; x_n)$ and set $X_{best} = X$;
  For a preset number of iterations do
      1.Evaluate and rank fitness $\lambda_i$ for each $x_i$ from worst to best;
      2.Generate the probabilities array P according to Equation 1;
      3.j = Select component based on the probability of its rank $P_i$;
      4.xj = Generate a random appropriate value that is not equal to xj ;
      5.Eva(X) = Evaluate the new solution;
      6.if Eva(X) < Eva($X_{best}$) then $X_{best} = X$;
end for
Return $X_{best}$ and Eva ($X_{best}$);
```

2.3 Materialized Views Selection

Materialized view selection has received extensive attention in the past few decades due to its wide application in many fields, such as speeding up query, update processing, data warehouses and decision support systems. Materialized views are especially attractive in data warehousing environments because of their query intensive nature.

Data warehouse have been introduced and developed to overcome the weakness of traditional databases.

A data warehouse is a very large database system that collects, summarizes, and stores data from multiple remote and heterogeneous information sources [10].

The problem of materialized views selection is the construction of a configuration of views in optimizing the execution cost of a data load. This optimization may be realized under certain constraints such as the storage space allocated for selected views or a superior boundary of the views maintenance cost [11]. We consider the first constraint by the rest of the paper.

Let CV be a set of materialized views that are qualified to be candidates to reduce the execution cost of queries set Q, generally supposed to be representatives of the system load. Let S be the disc space allocated by the administrator of the data warehouse for the creation of views. The problem of MVS is to construct a configuration of views V $\subseteq$CV minimizing the execution cost of Q, under the constraint of space. The problem can be formalized as follows:

Cost (Q, V) = min (cost (Q, E)) ∀ E ⊆ VC;

$$\sum_{v \in V} \text{taille}(v) \leq S$$

Many materialized view selection algorithms have been proposed to deal with this problem, such as greedy heuristic algorithm [12] and GA [13], but these algorithms have some limits. The greedy heuristic algorithm is highly problem dependent whereas in GA, it is hard to acquire good solutions in the beginning (first iterations).

2.4 Graph AND-OR For MVS

A graph G is called AND-OR view graph (figure1) for the views (or queries) $v_1, v_2, \dots, v_k$, if for each v_i there is a sub-graph G_i in G which is an expression AND-OR-direct acyclic graph(AO-DAG) for v_i [14] .

Each node *u* in an AND-OR view graph has the following parameters associated with it:
f_u: frequency of the queries on *u*.
S_u: space occupied by *u*.
g_u: frequency of updates on *u*.

Given set of queries q_1, q_2,..,q_k to be supported at a data warehouse, the AND-OR view graph can be constructed in terms of the following steps: The first step is to construct the expression AO-DAG d_i for each query q_i, then, we combine all the expressions AO-DAG d_1,d $_2$, ,d_k in order to obtain an AND-OR view graph G for the set of queries.

Each view is presented by a node of the graph in relation with a group of views it needs in order to be calculated; there is an AND relation between the views of the same group which is represented by the arc in figure1. The relation OR can be between two or more groups of views indicating that the views can be calculated from any group of them. For instance, in figure1 the view (a) can be calculated from the views (b, c and d) or (d, e and f).

In short, given an AND-OR view graph G and an available space size quantity S provided by the data warehouse, the materialized view selection problem aims to select a set of views V (i.e., a subset of the nodes in G), which minimizes the sum of total query response time and total maintenance cost, under the constraint that total space occupied by V is less than S.

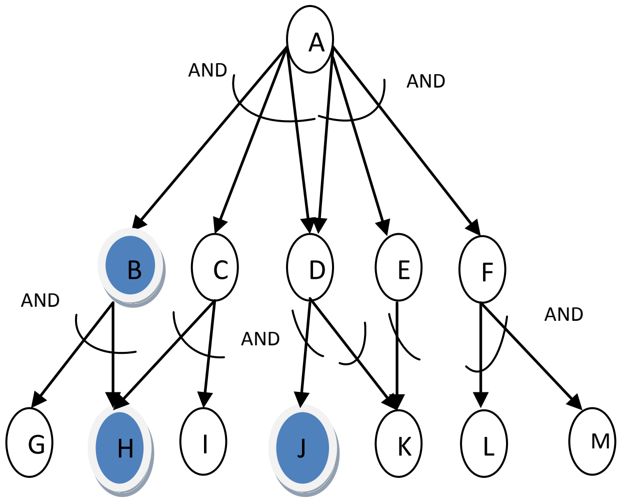

Fig 1: example of expression AO-DAG

3. ADAPTED EO FOR MATERIALIZED VIEW SELECTION

```
Algorithm 2: AEO pseudo-code for materialized views selection.
Generate an initial random solution $V=(v_1; v_2;..; v_n)$ where $\sum_{i=1}^{n} size(v_i) \leq S$
$V_{best} \leftarrow V$;
For a preset number of iterations do
    1. Evaluate and rank fitness $\lambda_i$ for each $v_i$ from worst to best;
    2. Generate the probabilities array P according to Equation 1;
    3. j = Select component based on the probability of its rank $P_i$;
    4. $v_j$ = Generate a random appropriate value that is not equal to $v_j$ ;where $\sum_{i=1}^{n} size(v_i) \leq S$
    5. Eva(V) = Evaluate the new solution;
    6. if Eva(V) < Eva($V_{best}$) then $V_{best}$ = V;
end for
 Return $V_{best}$ and Eva ($V_{best}$);
```

Since the materialized view selection problem has been proved to be NP-hard [15] it is impossible to resolve this problem by using the traditional algorithms. In this section, we propose a meta-heuristic algorithm called adapted extremal optimization (AEO) to deal with this problem using the extremal optimization (EO) method.

As it is mentioned above, the extremal optimization receives only one parameter τ, but since our problem is constrained we have to adapt the algorithm to make it receive our space constraint as a new parameter.

The first step of our adapted algorithm is to generate a random solution composed of a certain number of species (views or queries); this number is limited by the disc space S. The relation between the views is materialized in a graph AND-OR, the view graph is encoded as a binary string, where the constant number is the number of candidate views in the AND-OR view graph, the bit 0 denotes the corresponding node (view/query) is not materialized in the warehouse, the bit 1 denotes the corresponding candidate node (view/query) in the AND-OR view graph is materialized.

The second step is the calculation of fitness: The fitness is the contribution of each species (view or query) in the solution; we simulate our problem to the graph bi-partitioning problem where it is supposed to minimize the relation of the species with the partition's neighborhood [5] but in our case, we need the opposite i.e. we need to maximize the number of relations between the selected species (views or queries) with the unselected ones. We say that v1 is in relation with v2 if v1 is used to calculate v2. If only v1 is used to calculate v2,then the relation is unique for example we can calculate (e) using only (k); the relation is direct if there is no intermediate views between v1 and v2;

Each relation has its characteristics; each characteristic has its own coefficient. According to the relation's characteristics (direct, unique with presence of mates or unique with absence of mates) we multiply the appropriate coefficients to each other:

1. The coefficient is 1 if the relation is direct and unique.
2. If the relation is indirect, the coefficient is $\frac{1}{\text{NIV}+1}$; where NIV is the number of intermediate views.
3. If the relation is not unique, the coefficient is $\frac{1}{\text{NVM}+1}$; where NVM is the number of view's mates.
4. If there is absence of some of the view's mates (unmaterialized); the coefficient is $\frac{1}{\text{NAM}+1}$; where NAM is the number of absent mates.

Example 1: in figure1, suppose that the colored nodes compose the initial solution. Table1 explains how to calculate the number of relation of each species according to their characteristics.

Using the number of relations calculated as in table1, we calculate the fitness according to the equation 2:

$$\lambda_i = \frac{nb_i}{number\ \ of\ unmaterlialized\ \ \ views} \qquad (2)$$

Example2: $\lambda_b = \frac{1/9}{10} = 0.011$

The third step is to rank the species according to their fitness from the worst to the best and replace the worst species (view) with a random one;

The fourth step is to evaluate the new solution; if it is better than the best then it becomes the best;

This process is repeated as desired i.e. according to a preset number of iterations (convergence).

Table1: calculating the number of relations

view	Relation to	direct	unique	Absence of mates	multiplication	increments	number of relations
B	A	yes	no	yes	$\frac{1}{\mathrm{NVM}+1}*\frac{1}{\mathrm{NAM}+1}$	1/(3*3)=1/9	nb_b=1/9
H	C	yes	no	yes	$\frac{1}{\mathrm{NVM}+1}*\frac{1}{\mathrm{NAM}+1}$	1/(2*2) =1/4	1/4+1/8 =3/8 nb_h=3/8
	A	no	no	yes	$\frac{1}{\mathrm{NIM}+1}*\frac{1}{\mathrm{NVM}+1}*\frac{1}{\mathrm{NAM}+1}$	1/(2*2*2)=1/8	
J	D	yes	yes	no	1	1	nb_j=1

5. CONCLUSION AND FUTUR WORKS

The selection of materialized views is one of the most important problems in the design of data warehouses. Its aim is to find the best configuration of queries where the analyzing cost has to be minimal.

In the next step, we will prove the efficiency of our algorithm by creating a tool based on it and test it on a benchmark.

Since multi-agents systems are favorable to interest the complex system, we will propose muti-agent architecture for our tool.

6. REFRENCES

[1] Bak, P., Sneppen, K.: Punctuated equilibrium and criticality in a simple model of evolution. Physical Review Letters 71(24) (1993) 4083_4086

[2] Bak, P., Tang, C., Wiesenfeld, K.: Self-organized criticality: An explanation of the 1/f noise. Physical Review Letters 59(4) (1987) 381_384

[3] S. J. Gould and N. Eldridge, Punctuated Equilibria: The Tempo and Mode of Evolution Reconsidered, Paleobiology 3, 115-151 (1977).

[4] S. Bocttcher and A.G. Percus, "Extremal optimization: an evolutionary local-search algorithm," in Computational Modeling and Problem Solving in the Networked World, edited by H. M. Bhargava and N. Ye, 2003, Kluver, Boston.

[5] S. Bocttcher and A.G. Percus, "Extremal optimization for graph partitioning," Physical Review E, 64(2), pp.1-13, 2001.

[6] S. Bocttcher and A.G. Percus, "Optimization with extremal dynamics,"complexity, 8(2), pp.57-62, 2002.

[7] P. Gomez-Meneses and M. Randall, "A Hybrid Extremal Optimisation Approach for the Bin Packing Problem".

[8] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1989).

[9] Boettcher, S.: Extremal optimization: Heuristics via co-evolutionary avalanches. Computing in Science and Engineering 2 (2000) 75-82.

[10]H.Mistry, P.Roy, S.Sudarshan, and K.Ramamritham, "Materialized view selection and maintenance using multi-query optimization," Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, ACM Press, pp.307-318, May 2001.

[11]J. Darmont. Optimisation et évaluation de performance pour l'aide à la conception et à l'administration des entrepôts de données complexes. Page 6, 2007

[12] C. H. Choi, J. X. Yu and G. Gou, "What difference heuristic make: maintenance cost view selection revisited," Proceedings of the third Intl. Conf. on Advances in Web-Age Information Management, Springer-Verlag.pp.313-350, Jan 2002.

[13] W. Y. Lin and I. C. Kuo, "A Genetic Selection algorithm for OLAP data cubes," Knowledge and Information Systems, vol.6, pp.83-102, Feb 2004

[14]H.Gupta, "Selection of views to materialize in a data warehouse,"Proceedings of the 6th International Conference on Database Theory,Springer-Verlag,pp.98-112,January 1997

[15] H.Gupta and I.S.Mumick, "Selection of views to materialize under a maintenance cost constraint," Proceedings of the 7th International Conference on Data