Semi-Distributed Vacuuming Model on Temporal Database (SDVMT)

Mohammad Shabanali FAMI1, Elham Shabanali FAMI2, Dr Mohammad Ali MONTAZERI3, Dr Mohammad Taghi ISAAI4
1Islamic Azad University , emfami @ hrgsoft.com
2Isfahan university of technology, e.shabanali @ ec.iut.ac.ir
3Isfahan university of technology, Montazeri @ cc.iut.ac.ir
4Sharif university of technology, Isaai @ Sharif.edu

Temporal database is one of the most common types of databases. Portfolio management, accounting, storage, treatment management systems, aerology systems and scheduling are applications which their data have time references. Temporal nature of data and increasing size of temporal databases due to non-removal data requires presenting a solution to overcome this limitation. In this research, firstly the current model of vacuuming systems are simulated and analyzed. Then the proposed model introduced for vacuuming systems using distribution concepts. This model is simulated in the same conditions with current model. Using experimental results, advantages and disadvantages of both models were investigated. The proposed model is more capable than the current model in answering temporal queries. Its response time to temporal queries is less than the current model. But the proposed model's cost is more than the current model. Considering the possibility of idle resources usage in organizations, these costs can be ignored along with optimize usage of facilities.

***Keywords**: vacuuming, Multiclass queuing model, Schema versioning, temporal database.*

1 Introduction

Temporal database is one of the most common types of databases that its data have time references [1]. Among many different applications of these databases, Portfolio management systems, accounting, banking, aerology systems and scheduling can be mentioned [1]. In temporal databases in contrast with other databases, data will never remove from database. It means that temporal database uses append-only policy instead of update-in-place policy of other databases [1].

Due to increasing size of these databases, introducing a solution for overcoming growing volume of database is required. To deal with this problem, database designers had presented many models such as vacuuming data, schema versioning and aging data management.

Jensen introduced temporal data vacuuming [9]. Skyt studied data management methods for physically removed data [10] and suggested a framework for vacuuming temporal data [2]. Roddick aim was preventing some relations from removing in vacuuming process, so he searched about schema versioning [11] [6]. He also did researches about data mining on temporal database systems [3]. Jensen presented a framework for vacuuming temporal data. In this framework he vacuumed data base on organization's rules [4]. Grandi studied schema versioning on object oriented databases [5]. Skyt presented a method for removing data based on their features [12]. In all of these methods, some parts of data have been physically removed from database and it partitioned data in active and

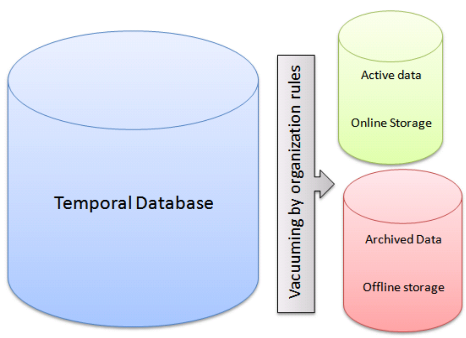


Fig. **1**: the current method for dealing with increasing volume of temporal database. To control temporal database volume, data will be archived base on organization rules.

inactive parts. Inactive data maintained in lateral storage devices, while active data will be remained in online system. In these systems inactive data will removed from online system physically.

To simulate and analyze the current method, a multiclass queuing model is introduced for current vacuuming method. After simulating current model, the proposed method is suggested and simulated by introducing its model. The assessment of proposed model is done by fixing both models' parameters and varying the amount of temporal queries with respect to ordinary queries.

In this paper, the amount of responses to queries and response time parameters was evaluated and considerable amount of growth in answering temporal queries was determined. These observations also show that the response time of temporal queries occurred in shorter time interval than current model. Due to existence of idle resources in organizations, the proposed method makes optimum usage of facilities possible.

In the rest of this paper, firstly the presented model for current vacuuming data methods explained in section 2. Then in section 3 the proposed model for vacuuming data introduced. These models are simulated in the same circumstances and the results obtained are shown in section 4. At last final conclusion of whole paper was states in section 5.

2. Current Vacuuming Methods

In this paper, at first current methods was investigated and it was found that most of them use the same logic. Some of these methods applied data vacuuming to confront infinite increase in database volume. Some others used schema versioning for this purpose. Also aging data management was presented by some others. All of these methods used part-of-data-deletion logic due to conditions.

In this logic, some parts of system's data will be removed base on organization rules. These removed data is maintained in lateral storage devices inactively. Fig. 1 shows current vacuuming systems procedure.

Considering the presented vacuuming method, the model in Fig. 2 visualizes a model for current systems. This is a two-class queuing model which two types of queries enter into queues. These types of queries are temporal queries and ordinary queries. Temporal queries rely on fetching data from inactive vacuums, while ordinary queries will be answered by online system directly.

In model of Fig. 2, D is online server queue. This server directly answers ordinary queries. Activation of inactive vacuums is undertaken by queue M. Since this routine is non-automatic, it often needs more time than automatic systems to retrieve data. To be able in answering temporal queries, activation of required vacuum should be done. When a vacuum retrieved, it may require another vacuum by itself and it will be continued for some number of vacuums. It means that when a vacuum activated and received its required service from online server, it may enter to queue M for retrieving vacuums that it needs.

If we consider average response time of online queries equal to one second, retrieval time of inactive data and activation time of them will be several times of it. If system administrator will finds a vacuum and activate it in 20 minutes, its speed is about 1000 times less than system's retrieval. It should be noted that there is not an ideal condition whole the time, sometimes the time between making a request for finding correspond vacuums and activation of them will be more than this amount of time. So it is rational to assume queue M slower than queue D for 1000 times.

In Fig. 2, there are two classes of ordinary queries and temporal queries. When the system starts up, the amount of data is less than the power of server, so there is no vacuuming and no temporal query. As time passed, the volume of data will be increased and vacuums will be created.

Fig. 3 that is presented by Ponniah [7]

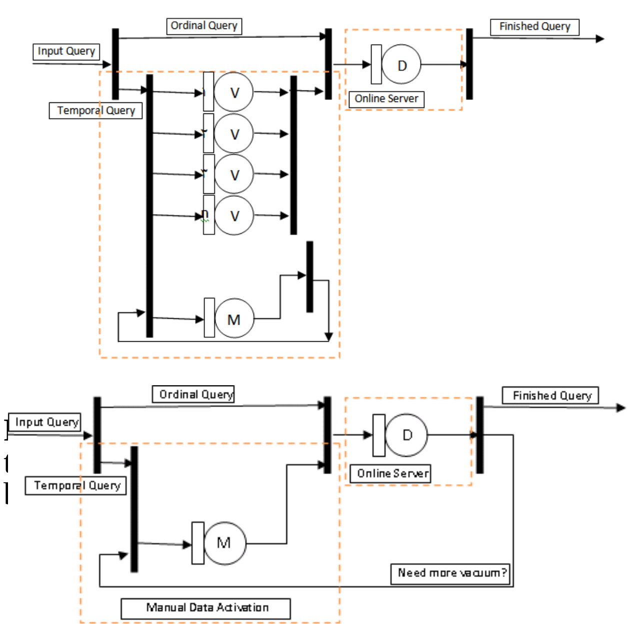


Fig. 2: Modelling of current vacuuming method. In this model, requested vacuums will be activated in queue M

shows that demand for information and number of data systems grow linearly with respect to the time [1]. Suppose that the volume of database at startup time is V_0 and growth rate of data can be calculated according to V_r/T_r. So the volume of database at time t can be obtained by Eq. 1:

$$V = \frac{V_r}{T_r} t + v_0 \qquad \text{Eq. 1}$$

3. Proposed Method

The most important problems of the

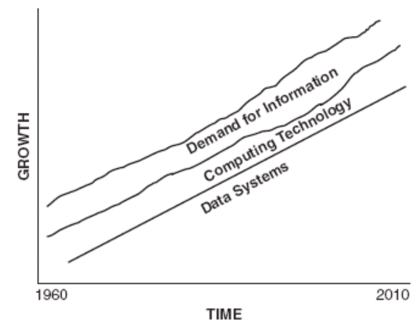


Fig. 3: the growth rate of demand for information with respect to time [7]. As time passed, demand for information and number of data systems will increase

current model were incapability for answering most of temporal queries and its high response time for other temporal queries. The proposed model that is showed in Fig. 4 has been designed to solve these problems with the objective of optimum utilization of resources. In this model, the concepts of distributed systems are used to improve on pre-designed models. In this system, the temporal vacuums data rather than being kept in inactive storage resources will be kept in on-line servers. Since most organizations usually provide the appropriate hardware infrastructure that does not use optimally, presenting this model provides a method for using maximum power of resources to troubleshoot problems about inactive data.

The main difference between proposed method and current method is in optimum usage of organization resources to deliver better services to applicants. More resources possession of the proposed method and scalability of it that obtained from its distributed nature make higher accountability for this method. If required resources of proposed method were not provided, organization has to use current method. In this situation, however some part of data will be kept inactive, there are more resources to return vacuums and maintain them online for organization.

As an instance, consider a small hospital that it has 20 workstations with normal capabilities along with its online server. This hospital can use its workstations as servers for vacuums. These workstations always have some amount of computational capacity and free storages that can be used for storing and retrieving vacuums data. It is obvious that there are limitations on these resources and after a while the organization will need inactive storage. By the way by optimum usage of resources that was costly for organization, the severity of the problem and the number of inactive vacuums will reduce.

In the proposed model, rather than lateral storage devices, data will be stored actively in some servers called vacuum servers. Vacuum servers are always slower and weaker than online server. When a temporal query arrived, it will be sent to related vacuum server. Then online server will gather and combine all results and answer applicant. In this method, vacuum servers will search for user answer simultaneously.

4. Experimental Results

With the aim of making a comparison between the proposed and current model, both of them was simulated using Simulink part of Matlab R2009a. To have a fair comparison, the adjustable parameters of both models assumed to be similar.

In these models, V_{cur} is a parameter that shows the volume of online database. Similarly V_{vac} is another parameter of the model that indicates average volume of each vacuum. T is the percentage of temporal queries than ordinary queries, V_0 is the primary volume of database and V_r/T_r is the growth rate of the database in time. So there are parameters $V_0, V_r, T_r, V_{cur}, V_{vac}$ and T for simulating models. In both models, primary volume of database is 4 megabytes, online database volume is 20 gigabytes, growth rate of data is 6.6 megabytes in time and average volume of each vacuum is 4 gigabytes.

Simulation was ran for 10, 50, 90 and 99 percent ratios of temporal queries than ordinary queries for 7000 time slices. Using central limit theorem [8] and considering the abounded amount of queries, the service time of online server for both models assumed to have a normal distribution which mean equals to 0.005 and standard deviation equals to 0.0001.

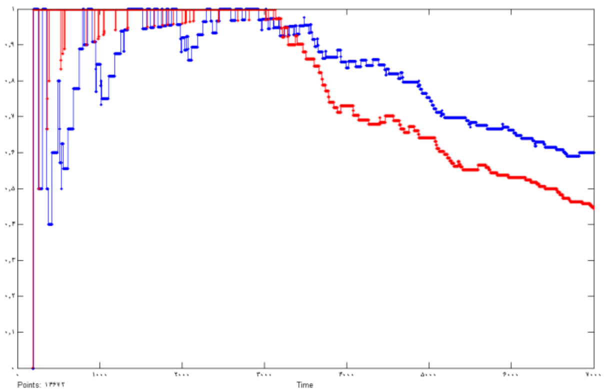
User departure time of current model was considered 4000 time slice. For the proposed model, this parameter was 2000 time slice. Simulation results show that if user patience will be assumed similar in both models, almost all users of current model will be gave up. Consequently user patience of current model is considered more.

4.1 Response Rate per Queries

Obtained results of simulation of both models are shown in TABLE I. In this table, variable T is the percentage of temporal queries than ordinal queries. %T is the percentage of response to temporal queries, while %O is the percentage of response to ordinary queries. TO indicates the number of departure queries.

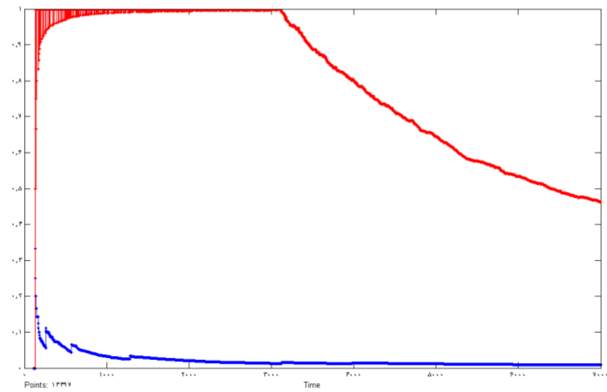
As it can be seen from Table 1, the current system at the best condition can answer 60 percents of temporal queries. When temporal query rate increase, this amount will tend to zero percent. Also when the rate of temporal query increased, the number of departure queries will increase. As it shows in Table 1, the proposed model has higher response power and it is because of parallel usage of vacuum servers and having automatic behavior in contrast with the current model. As temporal query rate increased, response power reduction of model is inevitable.

It can be observed from obtained results of proposed model that systems will answer more queries than current model before occurrence of user departure. About



a

a) When the proposed model has 30 vacuum servers and the ratio of temporal queries is 1%.



b

b) When the proposed model has 30 vacuum servers and the ratio of temporal queries is 10%.

Fig. 5: response rate to temporal queries with respect to time. In these diagrams, red lines are corresponding to current model as well as blue lines are corresponding to proposed model.

ordinary queries both models will answer to whole queries.

4.2 Response Time

In both models when a query is produced, it will be labeled with a time tag. When this query got served entirely, its presence time

Table 1: a comparison between response rates temporal queries in percent. Proposed moc will answer more temporal queries than t current model

Proposed model				Current model			
T	%T	%O	TO	T	%T	%O	TC
1	44.33	100	21	1	60	100	0
10	45.97	100	180	10	0.84	100	29
50	43.95	100	939	50	0.14	100	14
90	43.65	100	1711	90	0.35	100	23

in the system was calculated and the results are shown in Table 2. Because of parallel processing and automatic behavior of the proposed model, its waiting time reduced considerably. Less waiting time in answering temporal queries, less user give up from getting queries responses.

Response times of ordinary queries in both models are relatively similar, while temporal query's response time in the proposed model is so less than current model.

Table 2: query's response time. The proposed model reduced the response time of temporal queries.

Proposed model				Current model		
T	Tres	Ores	TO	T	Tres	Ores
1	198.2	0.005	0	1	275	0.005
10	40.01	0.005	180	10	901	0.005
50	17.85	0.005	939	50	311	0.005
90	18.43	0.005	1711	90	605	0.005

4.3 The Number of Completed Temporal Queries

Temporal queries with different rates of 1, 10, 50 and 90 percent of all queries were fed to both models and obtained results are shown in Fig. 6. While the current model makes better results than proposed model for low number of temporal queries, the proposed model will produce better results when temporal query rate increased.

4.4 Spending Time to Service Temporal Queries

In these models, in addition to average response time to temporal queries the number of completed temporal queries in a significant factor. To have more precise comparison, average response time to temporal queries and number of completed temporal queries was measured for both models. The factor of service time of temporal queries was defined by multiplying number of completed temporal queries with average response time of temporal queries.

$$\text{Spending time} = \text{Response time} * \text{Number of completed requests}$$

This factor compares both models by considering average response time along with number of responded queries. As it is shown in Fig. 7, proposed model delivers service to temporal queries for more time distances than current model.

5. Conclusion

In this paper the current model for vacuuming temporal data was introduced and simulated. Then to resolve some of its drawbacks a new model was suggested and simulated. At last by determining similar values for parameters of models, response time and response rate of them were compared. Simulation results analyzed and the same behavior on ordinary queries observed. About temporal queries, current model can answer about 24 percent of queries in the best condition. In addition, it produces responses in a long time that applicants will give up their requests. Proposed model answers more temporal queries in a time less than the current model. So the proposed model can be used as a key solution for vacuuming temporal data. One of the most important drawbacks of the proposed model is its implementation cost. Considering idle resources of organization, it can be expressed that this model will make an optimum usage of wasted costs of organization.

Current model has only two servers, active and inactive. If efficiency was quick response to user, when an ordinary query enters to system, since its service will be provided by online server, its response time will be short. Consequently this system has a good efficiency for ordinary queries. In situations that temporal queries are raised, system needs references to inactive database. Besides the response time of inactive servers is high, the efficiency in this situation is not well.

Proposed method needs a short service time to answer ordinary queries by online server and it becomes its efficiency to be high.

About temporal queries two situations may be happened. In some situation organization's resources will provide vacuum servers for system. As a result vacuum servers are responsible to answer temporal queries. As it is known, service time of vacuum servers approximately is equal to online server. Answering temporal queries in this situation needs a short service time and its efficiency is pretty good.

But in situations that organization's resources are unable to provide enough vacuum servers, using current method is inevitable. Consequently some parts of data should be searched in inactive servers. In this situation, system needs higher service time. Since vacuum servers are responsible for some part of queries, this time is less than current model. So the efficiency of this situation is not good but it is higher than the current method yet. In these systems some part of queries will be answered by using vacuum servers and some part of them will be answered by using inactive servers.

To conclude the discussion, proposed model and current model have approximately similar behavior about ordinary queries. Current model has low efficiency about temporal queries. Till

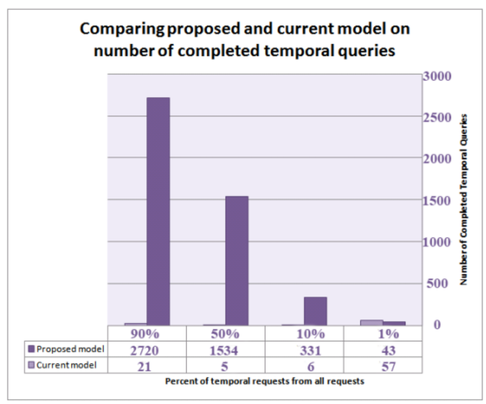


Fig. 6: A comparison between the number of completed temporal queries in proposed model and current model. Histogram bins with light colour are corresponded to current model.

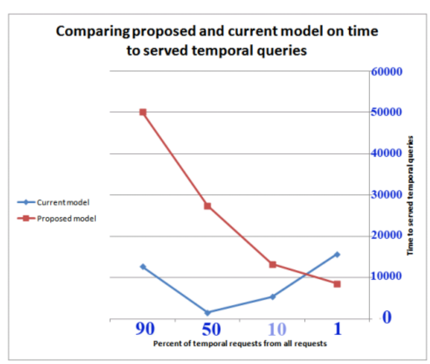


Fig. 7. A comparison between service times of temporal queries for both models. Proposed model spends more times to answer temporal queries than current model.

proposed method has enough vacuum servers its efficiency is good but once they will not sufficient, its efficiency will reduce but it is higher than the current model yet.

6. Future Work

In presenting this method, the performance of all vacuum servers assumed to be equal. Also the importance of all the system's data is the same. In future researches, using intelligent algorithm, a model can be suggested that considers data importance and servers performance. This model will keep important data in the servers with higher performance. These improvements make enhancement in system efficiency because important data need less service time.

One of the solutions to deal with lack of sufficient resources for proposed method is maintaining more widely used vacuums in vacuum servers. These methods previously were used by operating systems when their allocating algorithms keep higher-requested pages actively and lower-requested ones inactively.

References

Periodicals:

[1] C. S. Jensen, "Temporal Data Management" *IEEE Trans. Knowledge and Data engineering*, Vol. 11, No. 1, pp. 36-44, 1999.

[2] J. Skyt and C. S. Jensen and L. Mark, "A foundation for vacuuming temporal databases" *ELSEVIER, Data & Knowledge engineering*, Vol. 44, No. 1, pp. 1-29, 2003.

[3] J. F. Roddick and M. Spiliopoulou, "A Survey of Temporal Knowledge Discovery Paradigms and Methods" *IEEE Trans. Knowledge and Data engineering*, Vol. 14, No. 4, pp. 750-767, 2002

[4] C.S. Jensen and L. Mark, "*A Framework for Vacuuming Temporal Databases*," Technical Report CS-TR-2516, Univ. of Maryland, College Park, 1990

[5] F. Grandi and F. Mandreoli, "*A Formal Model for Temporal Schema Versioning in Object-Oriented Databases*," *ELSEVIER, Data & Knowledge engineering*, Vol. 46, No. 2, pp. 123-167, 2003.

[6] J. F. Roddick, "Schema Vacuuming in Temporal Databases" *IEEE Trans. Knowledge and Data engineering*, Vol. 21, No. 5, pp. 744-747, 2009.

Books:

[7] P. Ponniah, *Database Design and Development: An Essential Guide for IT Professionals*. John Wiley & Sons, pp. 3-10, 2003

[8] H. Fischer, *A History of the Central Limit Theorem: From Classical to Modern Probability Theory*. Springer, pp. 1-10, 2011

Papers from Conference Proceedings (Published):

[9] C. S. Jensen, "Vacuuming," *In Proc. The TSQL2 Temporal Query Language*, Kluwer, pp. 447-460, 1995

[10] J. Skyt and C. S. Jensen, "Managing Aging Data Using Persistent Views," *In Proc. Int. Conf. on Cooperative Information Systems*, Israel, pp. 132-137, 2000

[11] J. F. Roddick, "Schema Versioning," *In Proc. The TSQL2 Temporal Query Language*, Kluwer, pp. 425-446, 1995

[12] J. Skyt, C.S. Jensen, and T.B. Pedersen, "*Specification-Based Data Reduction in Dimensional Data Warehouses*," *In Proc. Int. Conf. on Data Eng*, USA, p. 278, 2002.