Applied Mathematics & Information Sciences
An International Journal

@ 2012 NSP
Natural Sciences Publishing Cor.

A Prediction Recovery Method for Supporting Real-Time Data Services

Yingyuan Xiao[1, 2] , Hua Zhang[1, 2] , Guangquan Xu [3] and Jingsong Wang[1, 2]

[1] Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, 300384 Tianjin, China

[2] Key Laboratory of Computer Vision and System, Ministry of Education, 300384 Tianjin, China

[3] School of Computer Science and Technology, Tianjin University, 300072 Tianjin, China

Corresponding author: Email: yingyuanxiao@gmail.com

Received 23 June, 2011; Revised 2 September, 2011; Accepted 23 September, 2011
Published online: 1 May 2012

Abstract: Real-time systems, which are often employed to monitor and interact with dynamic environments, are widely applied in time-critical applications, such as autopilot systems, medical patient monitoring, robot navigation, military command and control systems, agile manufacturing, etc. These time-critical applications require real-time systems can provide real-time data services ceaselessly. However, real-time systems cannot completely avoid all kinds of failures, so real-time systems must prepare for possible failures and provide fault tolerance capability. The conventional failure recovery methods cannot guarantee real-time data services available when some data items are damaged by failures. In this paper, we present a novel prediction recovery method through the integration of regression model and grey theory. The prediction recovery method guarantees real-time data services available by means of providing predictive values of damaged data to application activities which have to access these data immediately. Performance test shows that the proposed prediction recovery method can significantly improve the real-time performance.

Keywords: Real-time Systems, Real-time Data Services, Failure Recovery, Regression Model, Grey Theory

1 Introduction

Recently, the demand for real-time data services has been increasing [1, 2]. Many time-critical applications, such as autopilot systems, medical patient monitoring, robot navigation, military command and control systems, etc., often have to process various kinds of queries in a timely fashion, so they require the support of real-time systems. Typically, a real-time system (RTS) consists of a controlling system and a controlled system. For example, in an automated factory, the controlled system is the robots, assembling stations and the assembled parts, while the controlling system is the computer and human interfaces that manage and coordinate the activities on the controlled system. Thus, the controlled system can be viewed as the environment with which the computer interacts. The states of the application's environment are often modeled by a set of data within real-time systems. The controlling system interacts with its environment based on the data set available about the environment. However, the data obtained by various sensors may become invalid with the passage of time. Hence, timely monitoring of the environment as well as timely processing of the sensed information is necessary.

In addition to the timing constraints that arise from the need to continuously track the environment, data availability in a RTS is a crucial aspect for the controlling system to make its decision timely. However, real-time systems cannot completely avoid all kinds of failures caused by malicious attacks or human errors. Minor failures can cause some data unavailable and severe failures will result in system crash. When data being accessed by application activities become unavailable, the conventional failure recovery methods based logging [3-11] need to stop data services until these damaged data are recovered absolutely. Obviously, the conventional failure recovery methods cannot already meet the timing constraints that arise from time-critical applications. In this paper, we present a novel prediction recovery method, which can guarantees real-time data services available by means of providing predictive values for damaged data.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 analyzes the characteristics of data within real-time systems and presents the proposed system architecture for supporting prediction recovery methods. Section 4 presents the proposed prediction recovery methods. Section 5 shows an extensive experimental evaluation that demonstrates the superiority of the proposed methods. Finally, Section 6 concludes this paper.

2 Related Work

To aim at real-time database systems, some failure recovery methods have been proposed. Sivasankaran et al. [3] analyzed the characteristics of real-time data and discussed the issues of logging and failure recovery in real-time active databases. Choi et al. [4] proposed a parallel processing architecture adopting double-CPU. In the architecture, one CPU is responsible for usual transaction processing, while another CPU only responsible for recovery processing. To solve the problem of low efficiency of the sequential permanent logging, partitioned logging and ephemeral logging have been proposed respectively [5, 6]. Partitioned logging stores the log records according to transaction class, that is, the log records belonging to different transaction class are stored in different partitions that can alleviate the performance bottleneck caused by severe contention for single logging store partition. Ephemeral logging adopted non-volatile high-speed store as logging storage device and the log records would be deleted once the corresponding transactions were committed. Xiao et al. [7] presented a real-time dynamic crash recovery scheme (RTDCRS). The RTDCRS adopted the real-time logging scheme which integrated the characteristics of partitioned logging and ephemeral logging. Panda et al. designed a data dependency for failure recovery based damage assessment [8]. Liu et al. [9] developed a technique to rearrange affected and non-affected transactions in order to accelerate the recovery process. In [10], a damage confinement technique was proposed. In addition, Gadiraju et al. [11] presented an application log management scheme, which uses a mobile-agent-based framework to facilitate seamless logging of application activities for recovery from transaction or system failure. However, each of these methods either shuts down the database or blocks transactions from accessing the damaged data once recovery procedure is executed.

3 Preliminary

In this section, we firstly analyze the characteristics of data within real-time systems and then present the proposed system architecture for supporting prediction recovery methods.

3.1 Data Characteristics in a RTS

RTSs need to manage and maintain a large amount of data objects for application activities. These data objects, usually obtained from sensors, are called as temporal data objects which become invalid with the passage of time. A temporal data object in a RTS models the state of a real world entity, for example, the position of an aircraft. Associated with the state is the temporal validity interval of a temporal data object that loses validity after its temporal validity interval. To monitor the states of the real world faithfully, a temporal data object must be refreshed periodically by a sensor transaction before it becomes invalid.

Suppose that *DS* denotes the data object set within a RTS and *R* denotes a temporal data object in *DS*. In the following, we define two concepts to formalize temporal data objects.

Definition 1. A *temporal data object R* is defined as follows:

$$R = (o\text{-}id, attr_1, attr_2, \dots, attr_k, SI, VI)$$

where *o-id* denotes the object identifier of *R*, $attr_1$, $attr_2$, …, $attr_{k-1}$ and $attr_k$ denote *k* different attributes of *R* which correspond to *k* different characteristics of a real world entity, *SI* is the sampling instant at which *R* is refreshed by its sensor transaction and *VI* is the temporal validity interval of *R* within which *R* remains valid. We also call ($o\text{-}id$, $attr_1$, $attr_2$, …, $attr_k$, SI, VI) a state of *R* at sampling instant *SI*.

Consider a medical patient monitoring system in which each patient is monitored as a real world entity and its vital signs, such as heart rate, body temperature and blood pressure, are collected as attributes of a temporal data object. That is, each patient is modeled by a temporal data object R = (*patient-id*, *heart rate*, *body temperature*, *blood pressure*, t_i, t_i+25*ms*) where *SI* = t_i and *VI* = t_i+25*ms*.

Definition 2. A *history* of a temporal data object R is defined as the sequence $H = \{R_i\}_{i=1}^{m}$, where R_i denotes the state of R which was refreshed by its periodic sensor transaction at the *i*th sampling instant.

We use ***DS***$_c$ to denote the current state set of all data objects in a RTS and $\boldsymbol{DS_h}$ to denote the overdue state set of all data objects in the RTS. For a temporal data object, there is usually correlativity among its attributes. For example, in a medical patient monitoring system, a patient's heart rate, body temperature and blood pressure at the same sampling instant are closely related.

3.2 The Proposed System Architecture

Figure 1 illustrates the proposed system architecture for supporting prediction recovery methods.

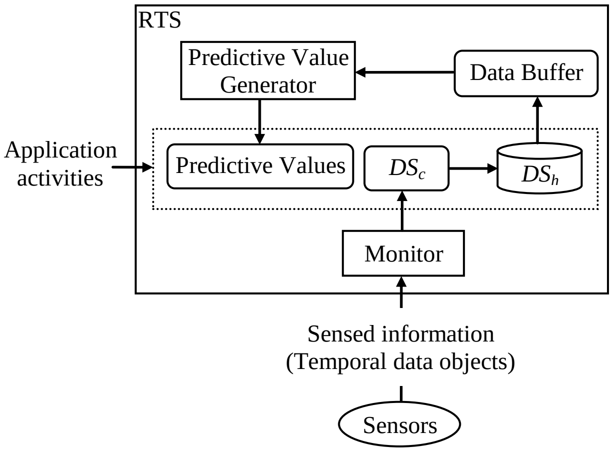


Figure 1: The proposed system architecture

In the proposed system architecture, the monitor is responsible for timely monitoring and processing the sensed information. When the sensed information collected by a sensor is monitored, the monitor generates a sensor transaction which is responsible for refreshing the sensed information into DS_c and the overdue state of the corresponding temporal data object is written into DS_h. DS_c is placed at main memory for the convenience of access while DS_h is stored at disk storage as a history of DS_c. When the current state of a temporal data object (resided in DS_c) is damaged due to some failures, its recent history in DS_h is load into data buffer from disk and the predictive value generator is responsible for generating its predictive value based on the recent history. The concrete methods of generating predictive value will be described in the next section. Once the current state of a temporal data object becomes unavailable, its predictive value will be provided to these application activities, which need to access to the temporal data object immediately, to guarantee data services available all the time.

4 Prediction Recovery Methods

In this section, we firstly propose two basic prediction methods which adopt multiple linear regression and grey theory respectively. Then we present the weighted combination prediction method which combines regression model with grey theory.

4.1 Prediction Method Based on Regression

Let (*o-id*, $attr_1$, $attr_2$, …, $attr_k$, t_i, t_i+*period*) denote the current state of a temporal data object R. For convenience, we use $attr_m(t_i)$ to denote the value of $attr_m$ at sampling instant t_i. Assume that an attribute value $attr_{k\text{-}j}(t_i)$ ($0 \leq j \leq k$-1) of the current state of R is damaged and becomes unavailable. We use $\{attr_{k-j}(t_l)\}_{l=i-n+1}^{i-1}$ to denote the nearest n-1 states of $attr_{k\text{-}j}$ from the current state $attr_{k\text{-}j}(t_i)$. Considering the fact that there is correlativity among attribute values of a temporal data object, we use multiple linear regression to model the relationship among attribute values of a temporal data object. The model expresses the value of $attr_{k\text{-}j}$ as a linear function of k-1 attribute values and an error term. Formally, the model for multiple linear regression, given n observations, is

$$attr_{k\text{-}j}(t_j) = b_0 + b_1 attr_1(t_j) + b_2 attr_2(t_j) + ...+ b_{k\text{-}j\text{-}1} attr_{k\text{-}j\text{-}1}(t_j) + b_{k\text{-}j} attr_{k\text{-}j+1}(t_j) +...+ b_{k\text{-}1} attr_k(t_j) + e_j \text{ for } j = i\text{-}n+1, i\text{-}n+2, \ldots, i \qquad (1)$$

where b_0 is regression constant, b_1, b_2, … , $b_{k\text{-}2}$ and $b_{k\text{-}1}$ are coefficients on the corresponding explanatory variables and e_j is error term. The model (1) is estimated by least squares, which yields parameter estimates such that the sum of squares of errors is minimized. The resulting prediction equation is

$$att\hat{r}_{k-j}(t_i) = \hat{b}_0 + \hat{b}_1 attr_1(t_i) + \hat{b}_2 attr_2(t_i) + \ldots + \hat{b}_{k-j-1} attr_{k-j-1}(t_i) + \hat{b}_{k-j} attr_{k-j+1}(t_i) + \ldots + \hat{b}_{k-1} attr_k(t_i) \qquad (2)$$

where "^" denotes estimated values.

4.2 Prediction Method Based on Grey Theory

The grey theory, one of the methods that are used to study uncertainty, is superior in theoretical analysis of systems with imprecise information and incomplete samples [12]. GM(1, 1) is a newly developed prediction method based on the grey theory. It has been proved that GM(1, 1) is very suitable for modeling those systems with unknown parts through the sampled data.

For a damaged attribute value $attr_d(t_i)$ ($1 \le d \le k$) of the current state of a temporal data object *R*, we can use $\{attr_d(t_l)\}_{l=i-n+1}^{i-1}$={ $attr_d(t_{i-n+1})$, $attr_d(t_{i-n+2})$, … , $attr_d(t_{i-1})$ } as the sampled data to predict $attr_d(t_i)$ by means of GM(1, 1). Specifically, the prediction method based GM(1, 1) is described as follows:

By the Accumulated Generating Operation (AGO), a new data series $\{attr'_d(t_l)\}_{l=i-n+1}^{i-1}$ is produced, where $attr'_d(t_l) = \sum_{j=k-n+1}^{l} attr_d(t_j)$.

By performing the algebraic mean operation on the series $\{attr'_d(t_l)\}_{l=i-n+1}^{i-1}$, the data series $\{attr''_d(t_l)\}_{l=i-n+2}^{i-1}$ is obtained, where $attr''_d(t_l) = \frac{1}{2}(attr'_d(t_{l-1}) + attr'_d(t_l))$ for l = i-n+2, i-n+3, …, i-1.

GM(1, 1) model (i.e., grey differential equation) may be established as follows:

$$\frac{dattr'_d(t)}{dt} + a \times attr'_d(t) = b \qquad (3)$$

where the coefficients, *a* and *b*, can be obtained by least square method, as shown in equation (4).

$$\begin{bmatrix} a \\ b \end{bmatrix} = [B^T B]^{-1} B^T Y \qquad (4)$$

here,

$$B = \begin{bmatrix} -attr''_d(t_{i-n+2}) & 1 \\ -attr''_d(t_{i-n+3}) & 1 \\ \cdots & \cdots\cdot \\ -attr''_d(t_{i-1}) & 1 \end{bmatrix} \qquad (5)$$

and

$$Y = [attr_d(t_{i-n+2}), attr_d(t_{i-n+3}), \ldots, attr_d(t_{i-1})]^T \qquad (6)$$

Once *a* and *b* are obtained, the GM(1, 1) can be used to predict $attr_d(t_i)$ by the equations below:

$$attr'_d(t_i) = (attr_d(t_{i-n+1}) - \frac{b}{a})e^{-a(i-1)} + \frac{b}{a} \qquad (7)$$

and

$$attr_d(t_i) = attr'_d(t_i) - attr'_d(t_{i-1}) \qquad (8)$$

4.3 The Weighted Combination Prediction Method

The above two basic prediction methods construct prediction models from two different aspects respectively. The prediction method based on regression considers the correlativity among attribute values of a temporal data object and adopts multiple linear regression to model the damaged data, while the prediction method based on grey theory use the history of the damaged data as sampled data to establish grey differential equation for prediction. The main problem of two basic prediction methods is that their predictive accuracy relies on data property in some measure. The weighted combination prediction method alleviates the main problem by the integration of regression model and grey theory.

Suppose *t* denotes an application activity and *R* is the temporal data object that is accessed by *t* at running time. Let R_r denote the predictive value of the current state of *R* based on multiple linear regression, R_g denote the predictive value of *R* based on grey theory and R_c denote the predictive value of *R* based on the weighted combination prediction method. The algorithm of the weighted combination prediction method is described as follows:

```
Algorithm Weighted Combination Prediction Method
Input: data request of t for the current state of R
Output: the available data value of R
1. if the current state of R is available in $DS_c$
2.   The current state of R is directly read from $DS_c$ by t;
3. else
4.   Call the prediction method based on regression to
     compute $R_r$;
5.   Call the prediction method based on grey theory to
     compute $R_g$;
6.   The predictive value $R_c = \omega \times R_r + (1 - \omega) \times R_g$ is
     provided to t;
END Weighted Combination Prediction Method
```

In the above algorithm, ω is the weight that depicts to what extent attribute values of a temporal data object are linear dependent. The ω value ($0 \le \omega \le 1$) can be adjusted dynamically according to data property. When the ω value is set to 0 or 1, the weighted combination prediction method degenerates to the prediction method based on grey theory or the prediction method based on regression.

5 Performance Evaluation

In this section, we first study how the different values of ω to influence the prediction accuracy of the weighted combination prediction method. Then we evaluate the real-time performance of the proposed methods through comparing them with the conventional recovery method based on logging. The algorithms were developed in C++ and the simulation experiments were executed on a PC with a 2.4GHz CPU and 1GB main memory. We employ an LRU memory buffer of 1Mb and the page size is set to 4Kb.

Our experimentations use three different datasets, i.e., independent, linear dependent and random datasets. All the reported results are the average of 10 runs of identical settings in order to reduce the randomness effect. The main performance metrics used for the evaluation include relative error η and the ratio of application activity missing its deadline, denoted as *MD*. Given some value v and its predictive value $v^{'}$, the relative error is $\eta = \left|\frac{v - v^{'}}{v}\right|$ which reflects the prediction accuracy. The lower value of η infers the better prediction accuracy. *MD* is defined as the number of deadline-missing application activities over the total number of application activities generated in the system. The lower value of *MD* means that the real-time performance is better. For convenience, we use PMR, PMG, WCP and CR to denote the prediction method based on grey theory, the prediction method based on regression, the weighted combination prediction method and the conventional recovery method based on logging, respectively.

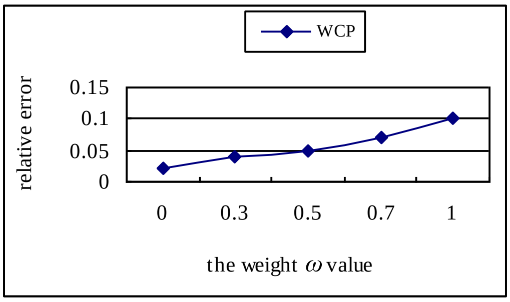


Figure 2: Relative error vs. weight value on independent dataset

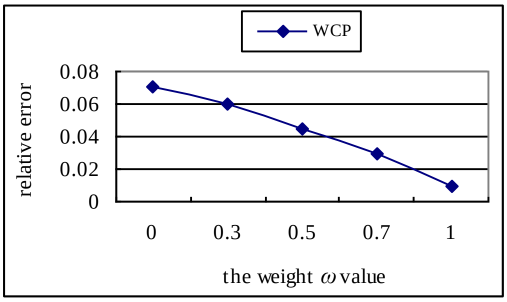


Figure 3: Relative error vs. weight value on linear dependent dataset

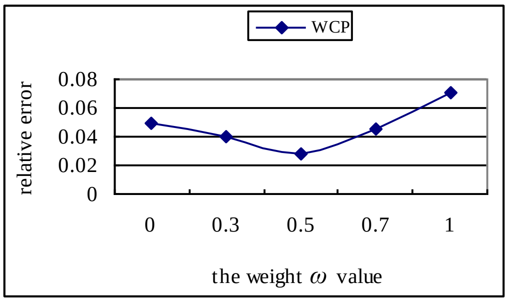


Figure 4: Relative error vs. weight value on random dataset

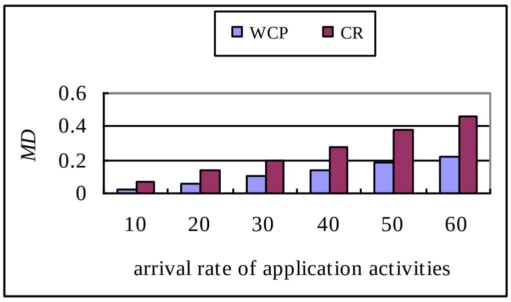


Figure 5: Comparison of *MD* between WCP and CR

Figures 2, 3 and 4 show how the different values of ω to influence the prediction accuracy of the WCP on three different datasets (independent, linear dependent and random datasets). We can see from Figure 2 that the relative error of the WCP on independent dataset enhances with the increment of ω, and the relative error reaches its largest value at ω =1. Note that the WCP degenerates to the PMR at ω =1, i.e., the PMR has the largest relative error among all WCPs with different ω values over independent dataset. This result is not surprising because independent dataset is not suitable for the PMR which assumes the linear dependence existing among attribute value of each data object. Figure 3 demonstrates the result that the relative error of the WCP on linear dependent dataset descends with the increment of ω. This is because linear dependent dataset is in favor of the PMR and the increment of ω results in the PMR puts on a greater impact on the WCP which leads the better predictive accuracy. Figure 4 shows that for random dataset the WCP has the minimum relative error at ω =0.5, that is, the arithmetical average of predictive values of PMR and PMG is the most suitable for predicting random dataset.

Figure 5 illustrates the real-time performance of the WCP. As we expect, the WCP outperforms the CR in real time performance significantly. The main reason is the WCP removes the overhead of writing up log which is required by CR.

6 Conclusion

Time-critical applications usually require real-time systems to provide ceaseless data services in a timely fashion. This means that data must fresh and available all the time. However, real-time systems cannot completely avoid all kinds of failures which can cause data damaged, so real-time systems must prepare for possible failures and provide fault tolerance capability.

In this paper, we first present the proposed system architecture for supporting prediction recovery. On the basis of this architecture, we present two basic prediction methods which adopt multiple linear regression and grey theory respectively. Further, we present the weighted combination prediction method which combines regression model with grey theory. We conduct extensive experiments, and the experimental results demonstrate the feasibility and effectiveness of our methods.

Acknowledgements

This work is supported by the Natural Science Foundation of China under Grant No. 61170174 and the Natural Science Foundation of Tianjin under Grant No. 11JCYBJC26700. The authors thank the reviewers for their comments.

References

[1] Y.Y. Xiao, Y.G. Chen and K. Lü, An efficient real-time concurrency control protocol for guaranteeing temporal consistency. International Journal of Innovative Computing, Information and Control, Vol.6, No.8. (2010), 3519-3528.

[2] K.Ramamritham and S.H.Son, Real-time databases and data services. Real-Time Systems Journal, Vol.28, No.2. (2004), 179-215.

[3] R.M. Sivasankaran, K. Stankovic and J. Stankovic, Data placement, logging and recovery in real-time active database. In Proc. of the 1st Int'l Workshop on Active and Real-time Database Systems. (1995), 1-15.

[4] M.S. Choi, H.S. Yoon and E.M. Song, Two-step backup mechanism for real-time main memory database recovery. In Proc. of the 7th Int'l Conference on Real-time Computing Systems and Applications. (2000), 453-457.

[5] K.Y. Lam and T.W. Kuo (Eds), Real-time database system architecture, Kluwer Academic Publishers, Boston, 2001.

[6] S.K. John and J.D. William, Performance evaluation of ephemeral logging. In Proc. of the 19th Int'l Conference on Management of Data. (1993), 187-196.

[7] Y.Y. Xiao, Y.S. Liu and X.Y. Chen, A time-cognizant dynamic crash recovery scheme suitable for distributed real-time main memory databases. In Proc. of the 3rd Int'l Conference on Management of Data. (2006), 591-600.

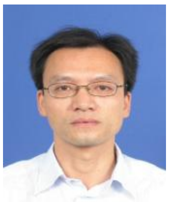
[8] B. Panda, J. Giordano (Eds), Reconstructing the database after electronic attack, Kluwer Academic Publishers, Boston, 1999.

[9] P. Liu, P. Ammann and S. Jajodia, Rewriting histories: recovering from malicious transactions. Distributed and Parallel Databases, Vol.8, No.1. (2000), 7-40.

[10] P. Liu and S. Jajodia, Multi-phase damage confinement in database systems for intrusion tolerance. In Proc. of 14th IEEE Computer Security Foundations Workshop. (2001), 191-205.

[11] S. Gadiraju and V. Kumar, Recovery in the mobile wireless environment using mobile agents. IEEE Transactions on Mobile Computing, Vol.3, No.2. (2004), 180-191.

[12] J.L. Deng (Eds), The course on grey system theory, HUST Press, Wuhan, 1990.



Yingyuan Xiao received the PhD degree in computer science from Huazhong University of Science and Technology, China. He was a visiting scholar in the school of computing at the National University of Singapore from 2009 to 2010. He is currently a professor in the school of computer and communication engineering, Tianjin University of Technology, China. His research interests include advanced databases，real-time system and mobile computing.

Hua Zhang received the PhD degree in computer science from Tianjin University, China. She is currently a professor in the school of computer and communication engineering, Tianjin University of Technology, China. Her research interests include computer graphics, distributed computing and image processing.

Guangquan Xu is a Lecturer at the school of computer science and technology, Tianjin University, China.

He received his PhD degree at Tianjin University in March 2008. He is a member of the CCF and ACM. His research interests include verified software, trusted computing, trust & reputation.

Jingsong Wang received the PhD degree in computer network and information security from Nankai University, China. He is currently a professor in the school of computer and communication engineering, Tianjin University of Technology, China. His research interests include computer network and information security.