Adaptive Fault Tolerance in Real-Time Information Systems

Shubinsky I.B., Rozenberg I.N., Papic L.

Russia, Moscow, JSC NIIAS, Serbia, Prijevor, ICDQM

Abstract

Real-time information systems (IS) control mission-critical processes. Violation of functioning in these systems may lead to dangerous errors in control and to intolerable risks. The general disadvantage of traditional ways of IS reliability assurance is an autonomous implementation of fault tolerance mechanisms, as well as breaks of calculation which is unacceptable for real-time systems. All known ways to assure IS reliability are based on the application of large volumes of artificial structure and information redundancy. The technology of adaptive fault tolerance proposed in this article consists in the active use of natural time and structure redundancy, as well as in the active (and automatic) reassignment of available computer power not only for operational processing of information, but also for implementation of observability of the system under the conditions of limited control means. The technology of adaptive fault tolerance in information systems when solving real tasks in limited time conditions provides for a timely automatic detection and handling of failures and glitches by means of operational localization of faulty computation modules and by subsequent automatic rearrangement of the system with removal of faulty modules from the process of functioning.

Keywords: Information system, computation modules, faults, failures, errors, reliability, fault tolerance, beats of active protection, reassignment of modules, adaptation to failures, automatic control, comparison of results, failure detection, rearrangement, process recovery.

I. Introduction

Real-time information systems (IS) control mission-critical processes. Violation of functioning in these systems may lead to serious errors in control and to intolerable risks [1]. Main causes of dangerous errors in control processes are known – these are glitches and software errors, data errors, failures of a system's equipment [2]. That is why to assure a reliable task solution under the conditions of failures, two essentially different approaches are applied – recovery of the solution after a failure of the system (or its component) and prevention from the system failure (fault tolerance). In real-time systems termination of the control process for the time necessary to recover the system functioning is in most cases unacceptable – the main way of the reliable solution of control tasks is to assure fault tolerance.

Traditional ways to assure fault tolerance are as follows: *reservation of resources* (for instance, computation modules (CM); *protecting against overconsumption of resources*; *clusterization*; *rejection of failures and fault shielding,* i.e. prevention from the distribution of fault consequences while the system continues the execution of its functions; *applications isolation; creation of microkernel architecture of the operating system (OS); isolation of the OS kernel from applications* and isolation of applications from each other, etc. [3,4,5, etc.].

The general disadvantage of the above mentioned ways is an autonomous implementation of fault tolerance mechanisms, as well as breaks of calculation, which is unacceptable for real-time systems. Attention is drawn to the fact that absolutely all considered ways to assure IS reliability are based on the application of large volumes of artificial structure and information redundancy, i.e. are practically based on the extensive way of fault tolerance assurance. That is why this is obviously very much a current challenge – to construct such a system to assure IS fault tolerance that under little structure and time reserve in real time it shall guarantee the assurance of IS adaptation to faults and failures of technical devices, as well as to exclude the cases of termination of the control process during the period of time that is longer than the acceptable period. The main way to solve this task is to develop the adaptive fault tolerant systems.

II. Task description

Adaptive fault tolerance is possible in information systems by means of introducing a subsystem of assurance of fault tolerance (SFT) into their structure. This software or software and hardware subsystem is formed at the stage of IS design using the provisioned redundant computing means with the help of communication media available in IS. It serves to provide timely protection or prevention from the failures of basic IS hardware and software.

A high level of SFT organization can be achieved using an adaptation mechanism. Let us consider a variant of the creation of SFT adaptive structures. This system contains (Figure 1):

– **information transducer** (IT), performing two groups of tasks: *the first group* is to connect the measurable states X of the system, unmeasurable states E and an adaptive action U. Measurable states are the data about current states of basic hardware and software and a resource. Unmeasurable states are the flows of failures, faults, software errors. *The second group* of tasks performed by IT is to form a vector T of time of adaptation to failures, faults, software errors, as well as of the control commands Y on the ongoing change of the resource, on the rearrangement of the information system, on the adjustment of current states of the system;

- **hardware and software resource *R*** of the system. It includes both, natural and artificial resources;

- **operator of adaptive control AC,** intended to form adaptive action in compliance with a certain algorithm F. A task of adaptation is to find such adaptive action U, so that vector T of SFT system in the field of measurable states X and in the field of unmeasurable states E stay in line with the objectives Z to be achieved

$$Z:\begin{cases} q_j(X,T)\to extr,\ j=1,...,k_1; \\ h_i(X,T)\le \tau_A; i=1,...,k_2; \\ g_i(X,T)\ge \beta\ ,i=1,...,k_2 \end{cases}$$

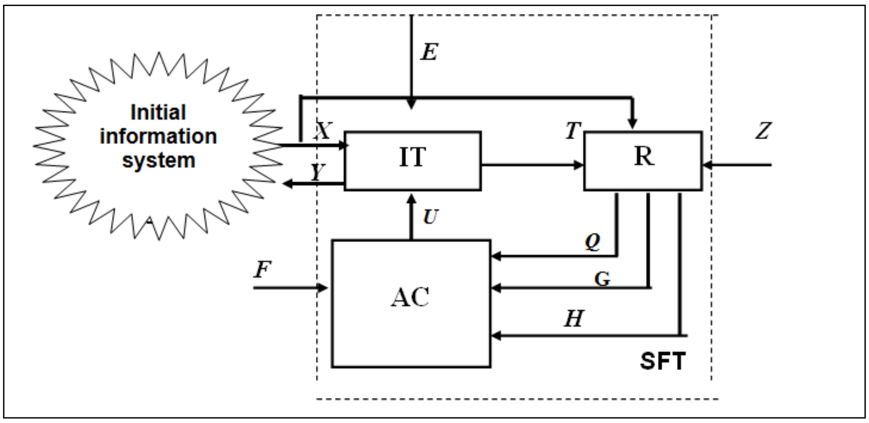

Figure 1: *System of assurance of fault tolerance in an information system*

where the parameters of object function q_j for each *j*-th control are related to the reaching of extreme values, for example, minimum losses of IS efficiency due to SFT actions, minimum additional storage consumptions, minimum delays when solving the tasks of observability and controllability, minimum structure reserve, maximum level of performance indicators of IS in the field *X*, etc. These parameters that could be minimized (maximized) are the aim of adaptation. Not only the information about the aim of adaptation should be transferred to the center of adaptive control, but also the following information: data about resource R, within which adaptation is possible ($U \subset R$), as well as data about adaptation algorithm F that assures the synthesis of adaptive action according to available information:

$$U = \Phi(X, T, Z, R).$$

Algorithm F solves the task of optimization. In this regard the task of reaching the goals Z is reduced to a famous task of multicriteria optimization

$$q_j(X, T) \to extr_{U \subset S}, j = 1, \dots, k_1,$$

where a set S is defined by condition $U \subset R$ and restrictions h_i, g_i.

Restriction h_i is in fact a requirement for a random time of adaptation $\nu_i = t_i - x_i$, that for the period from the moment x_i when a failure occurs till the moment t_i when the protection against this failure is over, not more than allowed time τ_A of the interruption in operation should be spent. Restriction g_i consists in the fulfillment of the first requirement of restriction h_i provided that the given level of assurance – the given probability of successful adaptation of SFT to the system failure – is kept.

To solve this task of multicriteria optimization, it is necessary to reveal the dependence of the object function q_j on the control *U* by a direct calculation of the values of restriction h_i, g_i and of the object function. Fig. 1 shows the adaptation scheme where Q, H, G are the vectors with components q_j and h_i, g_i, that are required to implement adaptive control.

III. Ideas of adaptive fault tolerance of information systems

The basic concept of adaptive fault tolerance is an active use of natural time and structure redundancy, as well as reassignment of available computer power not only for operational processing of information, but also for implementation of observability of the system under the conditions of limited control means. We shall further substitute the notion "adaptive fault tolerance of real-time information systems" with a short notion *"active protection (AP)"*

Active protection (AP) is intended to reach the required levels of fault tolerance of real-time information systems (IS) under little time reserve, limited efficiency of means used to detect failures of computation modules, as well as provided that the scope of redundant equipment does not exceed the scope of basic equipment. It also deals with the assurance of the given probability level of successful IS adaptation to faults and failures of the constituent elements and programs without much increase in the number of provisioned means of control and diagnostics.

Active protection is based on the following ideas [6,7]:

1. The duration of all cycles of the information processing divides into certain time intervals that shall be further called **A3 beats** or just beats. The beats are introduced to sample the continuous time of the information processing. Moments of sampling serve to register the fault-free CMs that are available at these moments in IS, as well as to tie the operations of IS observability and controllability to the A3 beats (vector X of measurable states is being formed). *Each A3 beat ends with*

forming a hold point, that stores the results of CM operation during the previous beat. Operations of observability and controllability, in particular the mechanisms of hold point, restart and etc. are in a strict compliance with the indicated moments of sampling. Thus, the hold point formed for a computation process of any CM, is updated in the A3 beats in time moments t_i, t_{i+1}, t_{i+2} etc.; a restart is made for a depth not exceeding the time τ_A; comparison of the results of the parallel operation of two CMs is done either in one, or in two beats and etc., but not more than in m of A3 beats. These processes in IS perform the functions of the **information transducer** in SFT – **IT** (see Fig.1).

A3 beats can have both *constant duration* (for instance, in IS with a pipeline processing of information), and *random duration*, which is specific to concurrent IS with different architecture. Average duration of a beat corresponds to the time spent to perform several hundreds of computation modules.

2. The whole set of the constituent computation modules of an information system is divided into two compound sets: *computing environment* – a set consisting of $l \leq m$ similar CMs; *protective environment* – a set consisting of $k \leq m$ similar CMs (**resource of** SFT - **R**). If $l + k = m$, (where m is a maximum number of the main CMs), then the IS is considered to have no artificial redundancy (there are no additional CMs). But with such correlation there is still the possibility to use $k < m$ of naturally redundant CMs, in favor of IS fault tolerance. If $l + k > m$, then the system has $l + k - m$ of additional CMs, and $m - l$ of naturally redundant CMs. At each moment of sampling to solve the tasks there may occur the necessity to use $l_i \leq m$ of the main CMs, as well as the availability of $m - l_i + k_i$ of redundant CMs in IS. If at this moment of time only l_i CMs are operable, it means that the protective environment has reached its limits, but there are enough available main CMs to solve the task of the information processing. Therefore, the registration of fault-free CMs of the main and protective environments at each moment of sampling is required to define the possibility to proceed with solving the processing tasks during an A3 beat with a sufficient amount of operable CMs. It is also necessary for a sustainable assurance of IS fault tolerance during the next beat with the help of redundant operable CMs if there are such modules at the moment t_i.

3. *Dynamic rearrangement of IS is carried out in the A3 beats* for the organization of a beat-by-beat parallel operation of the required number of the main CMs and available fault-free redundant modules. This will ensure the implementation of external control of CM operating capability. Thus, if at the moment t_i a restraint $m - l_i + k_i \geq l_i$ is fulfilled, then it is possible to form l_i of CM pairs and, therefore, to implement the control of faults and failures of all main CMs. If at this moment there is only one redundant fault-free CM, then, as expected, it switches to the next main CM, and during the next A3 beat one pair of CM operates in parallel, and the rest $l_i + 1$ CMs are not controlled during one A3 beat. Then in the next beat, with the help of the this particular fault-free redundant CM another pair of CM is formed and etc. As the result, for the number of beats equal or less than l_i it is possible to detect an event of failure in any CM from the cope of $l_i + 1$ that remained in the system arrangement. These processes in IS solve the tasks of **adaptive control** – **AC**.

4. *Virtual redundancy of all l of the main CMs with the availability of at least one fault-free redundant CM* is achieved owing to the fact that during a very short period of time that does not esceed l of beats, each of the main CMs operates in parallel with a similar CM. Therefore, in these short time spaces those pairs of modules operate in which all main CMs participate. With k of fault-free redundant CMs there is k which is a multiple virtual redundancy of all main CMs, as each main CM commutates with any redundant module.

5. All stages of IS observability (detection of an event of failure, localization, classification and location) **are performed on real tasks with no application of detecting devices during the processing of information**. Therefore, for the estimation of IS fault tolerance it does not matter that

under a parallel operation of the pair of CMs no faulty element of the module was detected, that was not used when solving this task. It should be noted that the higher is the intensity of applications coming to IS for handling (i.e. the higher the system is loaded), the more often IS is observed in the A3 beats. And conversely, the lower the system is loaded, i.e. the more pauses are there between the tasks, the less often IS is observed in the A3 beats. In long pauses between the tasks it is reasonable to use traditional means of control and diagnostics.

6. For the classification of failures, and for their location on the CM level, the system should have **not less than *m* = 2 main and one redundant CM**. With the simple active protection the additional CM opeates in parallel with the first main CM, and during the next beat (or in a beat) — with the second main CM. If the results of operation of the pair of CMs do not coincide, a double count is made during the previous beat, and it helps to eliminate error or identify a failure of one from the pair of CMs in case of repeated noncoincidence of the results. The failed CM is identified during a current beat upon the results of operation of the additional CM with the second main one. If the results do coincide, a decision is made in relation to the failure of the first main CM, If the results do not coincide, a decision is made in relation to the failure of the additional CM.

7. **Capabilities of active protection** considerably depend on the choice of **average duration of the beat τ of active protection.** Value τ should be selected so that during the time allowed for failure (fault, error) detection, defined by the duration of the processing cycle $\tau_{Д}^{\bullet} = \tau_{Д} - t_{У}$ the failed CM with the given level of assurance is localized and reswitched to an additional fault-free CM. Time τ_{L} is required for the recovery of a computation process from the last hold point with a respective implementation of active protection. When determining the value τ it is necessary to consider the times of solving the tasks and the pauses between them, laws of idistribution of these time periods, and the duration of a beat of active protection, number *m* of the main CMs, ways of implementation of active protection, number of redundant CMs.

VI. Example of automatic detection and elimination of failures of the system modules

In certain A3 beats CMs are redistributed between the computing and protective environments. Certain modules of the protective environment for an A3 beat hold the functions of the main modules, and vice versa. Under the reassignment all modules participate in pair operation, and as the result the A3 cycle is getting shorter. Let us illustrate this situation on the following example. Let *1 = m = 4, k=l.* The main CMs are enumerated from 1 to 4, the initial redundant CM is 5. Let us assume that in the first beat there was no reassignment of modules, and module 5 performed the control of the main module 1 (pair 5—1). In the second beat the modules have already been reassigned. And moudule 5 performd the functions of the main module 2, ehich now performd the control of the main module 3 (pair 2—3 in Table1). As the reuslt of this operation, for two A3 beats it is possible to control the operation of four CMs out of five (1, 2, 3 and 5). Under the CM reassignment for five A3 beats all modules are controlled twice.

The efficiency of CM reassignment grows with increase of *m* of initial main modules. Thus, with *m* = *6* and *k* = 2 it is possible to control all 8 modules during two A3 beats. System with 8 reassignable CMs is organized as follows. The cycle of single check of modules contains.

A = *2* beats. In the first beat module 7 of the protective environment performs the functions of the main module 2, which performs the control of the main module 1 (pair 2—1). Besides, in this particular beat module 8 of the protective environment performs the control of the main module 5 (pair 8—5). In the second beat module 8 performs the functions of the main module 4, forming the pairs 4—3 and 7—6. Other variants of the organization of the system with 8 reassignable CMs are also possible, but this variant has an important benefit — only modules 7→2 and 8→4 are reassigned, it significantly simplifies necessary means of commutation and control in the system.

Table 1

No. of beat	Numbers of main CMs	No. of controlling CM	Pairs of CMs under control	Reassigned CMs
1	1 2 3 4	5	5—1	-
2	1 5 3 4	2	2—3	5 2
3	1 2 3 5	4	4—5	5 4
4	5 2 3 4	1	1—2	5 1
5	1 2 5 4	3	3—4	5 3

In general, with the availability of fault-free *m* of the main CMs and *k* of the redundant modules it is possible to define the number of beats *A* in the cycle A3, meaning they underwent a single check, based on the considered logic of CM pair assignment

$A = int[(m + k)/2k]$

Here operation *int* is the operation of rounding of the result up to the nearest whole number. For example if $m = 5, k = 2$, then $A= int(1,75) = 2$. The same values of *A* shall hold with $m = 6, k = 2$. That is why under the organization of A3, having a known value *k* of redundant CMs, it is reasonable to protect such amount *m* of the main modules, so that the following relations are true.

$int\ (m/k) = m/k$ and $int[(m + k)/2k] = (m + k)/2k.$

Based on the CM reassignment it is possible to organize their *priority control***.** If under the reassignment of the modules it was necessary to equalize the frequency of controls of the main and redundant CMs, then under the priority control we solve the task to increase the frequency of controls of the indicated modules.

Let us illustrate the possibilities of the construction of systems with one module indicated as a priority one. It is assumed that all modules except the indicated one are controlled with one and the same frequency, and the indicated CM is controlled with a higher frequency.

Let us assume that the system contains $m = 4$ main (numbers from 1 to 4) and one (number 5) redundant CM. Module 2 is indicated as a priority one (Table 2). It is required to assure the frequency of controls of module 2 to be twice higher in comparison to the other four CMs. The solution of this task is described in Table 2. The cycle A3 is realized for three beats, and module 2 is controlled twice during the cycle, and modules 1, 3, 4, 5 are controlled just once.

Table 2

No. of beat	No. of cycle	Numbers of main CMs	No. of controlling CM	Pairs of CMs under control	Reassigned CMs
1	1	5 2 3 4	1	1—2	5—1
2		1 5 3 4	2	2—3	5—2
3		12 3 5	4	4—5	5—4
1	2	5 2 3 4	1	1—2	5—1
2		15 3 4	2	2—3	5—2
3		12 3 5	4	4—5	5—4

The reduction of time intervals between controls of some CMs is possible by increasing the time between controls of non-priority CMs. It is necessary to keep a rational compromise when solving such tasks of active protection.

V. Conclusion

Limited capabilities of redundancy, means of concurrent detection of failures, faults, errors during the implementation of information processes, limited capabilities of the set “hardware – software” – these all calls for the necessity to develop non-typical technologies to assure reliability of information systems. One of them is the proposed technology of *adaptive fault tolerance.* This technology consists in the active use of natural time and structure redundancy, as well as in the active (and automatic) reassignment of available computer power not only for operational processing of information, but also for implementation of observability of the system under the conditions of control means. The technology of adaptive fault tolerance in information systems, when solving real tasks in limited time conditions provides for a timely automatic detection and handling of failures and glitches by means of operational localization of faulty computation modules and by subsequent automatic rearrangement of the system with a removal of faulty modules from the process of functioning.

References

[1] Shubinsky, I. B. Functional reliability of information systems. Methods of synthesis [Text] / I. B. Shubinsky. – M. : Journal dependability, 2012. – 296 p.

[2] Shubinsky, I. B. Structural reliability of information systems. Methods of analysis[Text] / I. B. Shubinsky. – M. : Journal dependability, 2012. – 216 p.

[3] Avizienis, A. Dependability of computer systems [Text] / A.Avizienis, J-C Laprie, B. Randell / Fundamental concepts, terminology and examples. Technical report, LAAS – CNRS, October, 2000.

[4] Zyl, S. Standard devices QNX Neutrino to provide fail-safety of mission-critical computer systems [Electronic resource] / S. Zyl. – CTA. – 2009(3). – 118 p. – Access mode: http://www.cta.ru/cms/f/389405.pdf.

[5] Fogelin, D. Implementation of high availability in embedded systems [Electronic resource] / D. Fogelin, K. Kving. – Access mode: http://www.asutp.ru/?p=600410.

[6] Shubinsky, I. B. Active protection against the failures of microprocessing computer systems [Text] / I. B. Shubinsky. – M. : Znanie, 1987. – 60 p.

[7] Shubinsky, I. B. Reliable fail-safe information systems. Methods of synthesis [Text] / I. B. Shubinsky. – M. : Journal dependability, 2016. – 541 p.

About the author

Shubinsky Igor Borisovich – Dr.Sci., Professor, member of several academies of sciences, Expert of Research Board under RF Security Council, editor-in-chief of scientific and research journal "Dependability", deputy editor-in-chief of journal "Reliability: Theory and Applications" (USA), director general of JSC "Information safety on transport – IBTrans" (Russia, Moscow).

Contacts:
119333, Russia, Moscow, Vavilova Str. 48-339;
Tel. +7(499) 137 70 42; tel/fax +7(495)786 68 57; mobile +7(985) 774 34 29
e-mail: igor-shubinsky@yandex.ru