Hu's algorithm application for task scheduling in N-version software for satellite communications control systems

Article · July 2015

DOI: 10.1109/SIBCON.2015.7147270

3 authors:

Alexey Sergeevich Chernigovskiy
Siberian Federal University

Roman Tsarev
Siberian Federal University

Alexey Knyazkov
Siberian Federal University

Hu's Algorithm Application for Task Scheduling in N-Version Software for Satellite Communications Control Systems

Alexey S. Chernigovskiy, Roman Yu. Tsarev
Department of Informatics
Siberian Federal University
Krasnoyarsk, Russia
Email: achernigovskiy@hotmail.com, tsarev.sfu@mail.ru

Alexey N. Knyazkov
Department of Transport
Siberian Federal University
Krasnoyarsk, Russia

***Abstract*—Ensuring the reliable operation of satellite communications control systems is an important objective. The reliability of such control systems is largely determined by the level of reliability of their software. The N-version programming has been one of the topical approaches to increase the reliability of the software. The N-version software is based on software redundancy that requires a larger amount of execution time. Satellite communications systems are real-time applications for which any increase in the software execution time is critical. Hu's scheduling algorithm is suggested for solving this task. The proposed approach allows for generation of an optimal execution schedule for versions of the N-version software of the control system of a satellite communications system. The schedule has the advantage of using a reduced time for task solving. The paper presents a solution to determination of the minimum execution time to run the N-version of the control system of a satellite communications system for a given number of processors. The paper shows a procedure for determination of the number of processors required in the minimum time for execution of a task flow.**

Keywords—communications satellite system; control system; N-version software; scheduling; Hu's algorithm

I. INTRODUCTION

The deployment of a satellite constellation is preceded by the scheduling phase in which the schedule of sequence of flight operations is defined [1]. Pre-flight scheduling results in generation of an initial nominal flight schedule to be regulated by the control cycles. The initial nominal flight schedule is used for control of communications satellites. The predetermined flight schedule is implemented during a flight. The on-board systems of satellites receive control inputs from the flight control center or an on-board control system.

Flight monitoring to be carried out regularly during the implementation of the flight schedule in accordance with the control cycles during standard operation of communications satellites includes the following steps: acquisition of data of the actual state of a communication satellite, processing of the data to determine the accurate values of the parameters of a satellite, analysis of the data by comparing them either with the target values or with the scheduled or predicted values, and evaluation of response of a communications satellite to a control input.

Based on the results of flight monitoring, a decision is made regarding further flight control. This decision specifies if a flight is to be maintained according to the initial schedule (if the actual values of monitored parameters are within the target limits), or if the initial flight schedule is to be adjusted and an adjusted schedule is to be used for further flight control (if these values are out of target range).

In the latter case, the essential components of a solution are selection of a control strategy, selection of an option among the prioritised control actions and adjustment of the flight schedule.

When a satellite communications system is in operation, it receives data from the satellites for processing. As a result, control inputs can be generated for adjustment of the flight schedule. Calculations are performed using complicated algorithms to process large volumes of data.

It is critical to maintain control of communications satellites. Failure to process data or any processing of data that results in invalid data will generate an incorrect control input and, as a consequence, cause a failure to perfrom functional tasks, or even loss of a communications satellite.

The control system of a satellite communications system is a real-time system that is capable of performing the whole range of tasks of data processing and generation of control inputs by a given point in time to maintain a constant and timely interaction with communication satellites.

Thus, the control system of a satellite communications system solves complex tasks of data processing and control of communications satellites. The control system includes sophisticated software that consists of a large number of components closely co-operating for the purpose of finding a solution to a joint objective.

As it is the software of a control system that is used for data processing, the reliability of programmed control determines the reliability of a control system. To ensure a high level of reliability and fault tolerance of software, different

A. S. Chernigovskiy, R. Y. Tsarev and A. N. Knyazkov, "Hu's algorithm application for task scheduling in N-version software for satellite communications control systems," *Control and Communications (SIBCON), 2015 International Siberian Conference on*, Omsk, 2015, pp. 1-4.
DOI: 10.1109/SIBCON.2015.7147270
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7147270&isnumber=7146959
SCOPUS citation: http://www.scopus.com/inward/record.url?eid=2-s2.0-84941135322&partnerID=40&md5=28da35cd2024b4426fe2128e39b59acb

approaches are used based on the application of time, information and software redundancy [2].

One of the most promising methods for the design of highly reliable software control systems is the N-version programming [3, 4, 5]. This approach guarantees that the errors of one version of the software module do not lead to disruption of a control system that is commonly fulfils strict requirements for reliability.

As this approach to ensure a high level of software reliability is based on software redundancy, a period of time for the execution of tasks will increase with a greater number of versions being used. This is of critical importance to real-time systems that include control systems of satellite communication systems [6]. In this regard, minimization of the software execution time of a control system becomes of immediate interest.

One solution to this task is application of a number of processors (computing machines) for parallel execution of multiple versions of the N-version software. The solution to the task of processor-based distribution of versions can be found in the theory of multiprocessor schedules.

Multiprocessor schedules allow for processor-based distribution of computational tasks in order to minimize the run-time of a task flow [7, 8]. By applicaton of multiprocessor scheduling to the N-versioning software, one can achieve the optimal allocation of tasks that represent the execution versions of the N-version software, distributed for available processors, thereby obtaining the minimum run-time of N-version modules.

In this paper, we propose the application of Hu's algorithm to generate a multiprocessor schedule for the execution of the N-version software of the control system of a satellite communications system to determine the minimum time required for data processing and implementation of the required control functions.

II. PRINCIPLES OF N-VERSION PROGRAMMING

N-version programming, as an approach to providing the fault tolerance of software, was suggested by Algirdas Avizienis in 1977 [3]. A. Avizienis defined N-version programming as an independent generation of $N \geq 2$ functionally equivalent programs (versions) in accordance with identical initial specifications for software design.

In this context an independent generation of programs is understood to refer to the laying-out of the design of software in such a way that prevents interaction each of the N versions of software modules with the others with respect to the software implementation process.

These N versions are provided with available means of competitor-based execution, during which the correct results of execution of these versions are verified and selected for certain control points (Fig.1). Usually voting algorithms are used for this purpose [9].

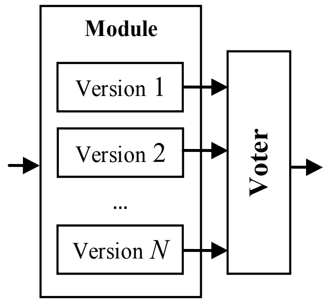

Fig. 1. N-version software module.

By analogy with multi-channel hardware that is designed to operate in extreme conditions and to satisfy high requirements for fault tolerance (ground and on-board control systems for aircraft, including spacecraft, computerized means of control of critical risk facilities, facilities posing potential hazard to human activity) that represents $N \geq 2$ hardware units, N-version software is an extension of the classical software by integration of redundant versions of software modules.

Unlike hardware, the method of a simple duplication for software is ineffective for the purpose of providing fault tolerance to errors. This is due to the fact that duplication of components also results in duplication of programming errors that have remained undetected during the testing and debugging phases [4].

In this regard, during the design and development of versions of N-version software, a number of methods and algorithms for the solving of identical tasks are implemented. The preferred method is to assign separate and independent teams of programmers to the development of versions using different programming languages and environments [10].

This method allows to compensate and mask malfunctions or failures of separate versions of software modules, thus ensuring the fault tolerance and to guarantee the implementation of the objective functions of N-version software.

III. HU'S SCHEDULING ALGORITHM

Hu's algorithm allows to generate a task execution schedule resulting in the minimum execution time for a given number of processors [11, 12]. In the generation of the optimal schedule of task execution, initial input data is a graph showing the sequence and the relationship of these tasks.

Firstly, the nodes of a graph task are labelled. Node N_i is labelled $a_i = X_i + 1$, where X_i – length of the longest path from N_i to a final vertex in the graph. The labelling begins from the final vertex to be labelled $a_1 = 1$. Nodes that are further away by a distance of 1 from the final node, are labelled 2, and so forth. The meaning of this labelling scheme is that the minimum time $T_{\min,}$ needed for processing of a graph is associated with $a(max)$, nodes with the highest indices, by the ratio

$$T_{\min} \geq a(max).$$

Using the graph labelling procedure, one can obtain the optimal schedule for m processors according to the following algorithm.

Algorithm

1. Schedule the first m initial nodes with the highest numberred labelling indices. The term “entry node” is applied to a node with no predecessors. If the number of such nodes is greater than m, select m nodes whose a_i is greater than or equal to a_i of those nodes that are not selected. In case of a tie, choose a node arbitrarily.

2. Remove the m scheduled nodes from the graph.

3. If the graph is empty, algorithm is stopped, otherwise — go to step 1.

IV. Results and Discussion

Let us consider the example of application of Hu’s algorithm for the control system of a satellite communications system implemented with the use of N-version programming (Fig. 2).

The first module of N-version software is responsible for the preparation of initial data for scheduling and control (marked as PREP), followed by a number of modules for data processing. Some of these modules are executed sequentially, some concurrently. The execution of this software ends with the module that generates, on the basis of the input data, ballistic output data (marked as BALLISTIC). It is worth noting that the modules are interdependent and each subsequent module can not be executed until the previous one has been executed.

Let us build a graph of execution of module versions of N-version software of this control system and perform labelling in accordance with Hu’s algorithm. The graph of execution of the versions is presented at Fig. 3; Table 1 shows the values of the vertices in the graph and their corresponding versions of the modules.

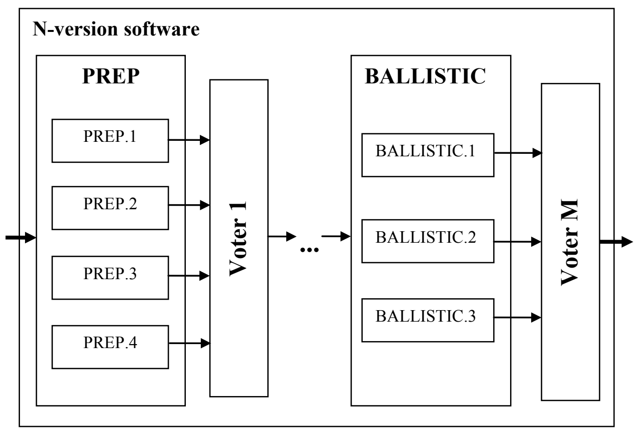

Fig. 2. N-version software of control system of communications satellite system.

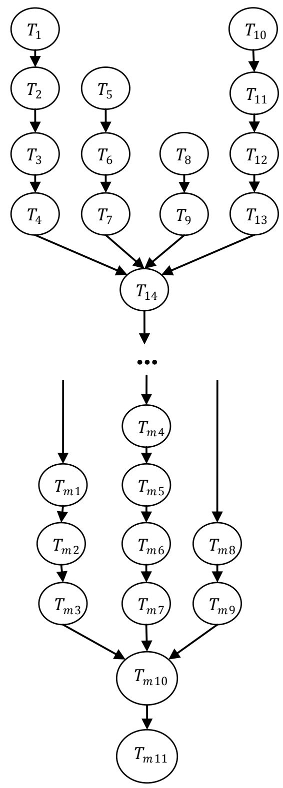

Fig. 3. Graph of execution of the versions.

A graph of execution of the versions has been built assuming that one version may be executed during a number of time units. A time unit is taken to be a specific period of time, that was adopted in the beginning of scheduling. It is used to determine the number of time units that are required to execute each version. One vertex of the graph represents one time unit of operation of one or another version of the module.

TABLE I. Versions of the Modules in the Scheduling Graph

Vertices	Versions of modules
$T_1 - T_4$	PREP.1
$T_5 - T_7$	PREP.2
$T_8 - T_9$	PREP.3
$T_{10} - T_{13}$	PREP.4
T_{14}	Voter 1
$T_{m1} - T_{m3}$	BALLISTIC.1
$T_{m4} - T_{m7}$	BALLISTIC.2
$T_{m8} - T_{m9}$	BALLISTIC.3
T_{m10}	Voter M
T_{m11}	Information transmission

Other data processing units are omitted from the graph. The graph may have units that operate both sequentially and concurrently. Modules may be heterogeneous and differ in execution time. In this connection, the scheduling of task execution is very important for achieving the minimum time to execute all the software modules of the system control of a satellite communication system.

In the graph, m represents the number of tasks that have been obtained by partitioning the versions into parts preceding the last module of the system (module BALLISTIC in the present example). Then the index m <number> corresponds to the task m + <number>.

The schedule of execution of the module versions of N-version software of the control system of a satellite communications based on Hu's algorithm is shown in Fig. 4.

It should be noted that it is possible to solve the inverse problem of determining the number of processors of a computing center of the control system of a satellite communications systems that is required to process the graph in a given time.

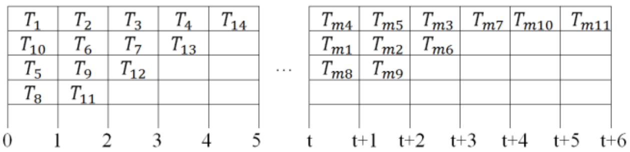

Fig. 4. The schedule of version execution for four processors.

As previously noted, the minimum time to process a graph labelled in accordance with Hu's procedure equals $a(max)$. Let us assume that a graph is required to be processed in a predetermined time t, where $t = a(max) + C$, C – a positive integer. The minimum number of processors m required to process the graph in the time t is calculated as

$$m-1<[1/(y^*+C)]\sum_{j=1}^{y^*} p(a_{\max}+1-j)\le m,$$

where $p(i)$ indicates the number of nodes in the graph labeled a_i, and y^* – value of the constant y, which maximizes this expression.

V. Conclusions

The reliability of control systems of a satellite communications system is of critical importance as the correctness of the results gained by such control systems is key to successful implementation of the target objectives to ensure communication and control. A high level of reliability of these control systems can be achieved by the use of N-version software during the design and development of software systems of this class. The proposed approach is based on the application of software redundancy, which imposes high demands on system performance. For optimal implementation of versions of the modules of N-version software of the control system of a satellite communications system, one is required to build a schedule of version execution. This problem has been solved using Hu' algorithm. It is worth noting that the problem of building the optimal schedule of execution of N-version software of the control system of a satellite communications system has been solved for the first time.

The use of multi-processor schedules for a processor-based distribution of the tasks that are versions of execution of N-version software in the system control of a satellite communications system allows to reduce the solving time for specific tasks and obtain the guaranteed result at the specified moments of time, which is of utmost importance to real-time applications.

Acknowledgment

The authors would like to thank the organizers of the International Siberian Conference on Control and Communications (SIBCON-2015) for the opportunity to take part at the conference. The authors gratefully acknowledge financial support from the Mikhail Prokhorov Foundation.

References

[1] V. Kolici, X. Herrero, F. Xhafa, L. Barolli, "Local Search and Genetic Algorithms for Satellite Scheduling Problems," Broadband and Wireless Computing, Communication and Applications (BWCCA), 2013 Eighth International Conference on Compiegne, pp. 328-335, October 2013.

[2] M. Jain, R. Gupta, "Redundancy issues in software and hardware systems: An overview," International Journal of Reliability, Quality and Safety Engineering, vol. 18 (1), pp. 61-68, 2011.

[3] A. Avizienis, and L. Chen, "On the implementation of N-version programming for software fault-tolerance during program execution," In Proc. IEEE Comput Soc Int Comput Software & Appl Conf, COMPSAC '77, pp. 149-155, November 1977.

[4] I.M. Golubev, R.Ju. Tsarev, and T.I. Semenko, "N-version software systems design," In Proc. of the 11th International Scientific and Practical Conference of Students, Postgraduates and Young Scientists; "Modern Techniques and Technologies", MTT 2005, pp. 147-149, March 2005.

[5] M. Xie, C. Xiong, S.-H. Ng, "A study of N-version programming and its impact on software availability," International Journal of Systems Science, vol. 45(10), pp. 2145-2157, 2014.

[6] X. Zhu, J. Wang, X. Qin, J. Wang, Z. Liu, "Fault-Tolerant Scheduling for Real-Time Tasks on Multiple Earth-Observation Satellites," IEEE Transactions on Parallel and Distributed Systems, vol. PP(99), October 2014.

[7] K. Hashimoto, T. Tsuchiya, T. Kikuno, "New approach to fault-tolerant scheduling using task duplication in multiprocessor systems," Journal of Systems and Software, vol. 53 (2), pp. 159-171, 2000.

[8] D. Poles, "One-level and two-level scheduling for real-time systems," IEEE EUROCON 2013, Zagreb, pp. 569-577, July 2013.

[9] A. Karimi, F. Zarafshan, S.A.R. Al-Haddad, A.R. Ramli, "A novel N-input voting algorithm for X-by-wire fault-tolerant systems," Scientific World Journal, vol. 2014, 2014.

[10] D. Powell, J. Arlat, Y. Deswarte, K.Kanoun, "Tolerance of design faults," Lecture Notes in Computer Science, vol. 6875 LNCS, pp. 428-452, 2011.

[11] T.C. Hu, "Parallel Sequencing and Assembly Line Problems," Operations Research, vol. 9(6), pp. 841-848, 1961.

[12] O. Sinnen, "Reducing the solution space of optimal task scheduling," Computers and Operations Research, vol. 43 (1), pp. 201-214, 2014.

A. S. Chernigovskiy, R. Y. Tsarev and A. N. Knyazkov, "Hu's algorithm application for task scheduling in N-version software for satellite communications control systems," *Control and Communications (SIBCON), 2015 International Siberian Conference on*, Omsk, 2015, pp. 1-4.
DOI: 10.1109/SIBCON.2015.7147270
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7147270&isnumber=7146959
SCOPUS citation: http://www.scopus.com/inward/record.url?eid=2-s2.0-84941135322&partnerID=40&md5=28da35cd2024b4426fe2128e39b59acb