APPLICATIONS

NONSTOP-32, A Fault-Tolerant System Used in Metallurgical Plants

Florin Hartescu, Mihaela Cosma, SavinaTeodorescu

Real-Time Systems Laboratory
Research Institute for Informatics
8-10 Averescu Avenue,
71316 Bucharest
ROMANIA

Abstract:The paper presents a fault-tolerant system designed for the pellet production process control in metallurgical plants. The system is based on a RT-ARCH (Real Time ARCHitecture)[1], an architecture of which software tools are used in development and run-time application phases for the specific equipments used in process control in metallurgical plants.
The software tools make the 16/32-bit equipments work together in an integrated fault-tolerant system, capable of meeting the general and specific requirements of the application. NONSTOP-32 proves fault-tolerance by high redundancy, by data and code replication(via a high speed communication connection) on the different nodes of a distributed system.

Florin Hartescu was born in Bucharest, Romania in 1950. He received his M.Sc. in Computer Science from the Polytechnical Institute of Bucharest. He also graduated the Faculty of Mathematics, the University of Bucharest. He works as senior researcher at the Research Institute for Informatics in Bucharest. He is currently preparing a doctoral thesis in the field of Real-Time Systems.
His research interests include real-time systems, process control systems, databases, parallel architectures, CAD tools for automation, networks.
He published numerous papers and he is a member of DECUS.

Mihaela Livia Cosma was born in Bucharest, Romania, in 1964. She graduated in Computer Science from Control Engineering Department, the Polytechnical Institute of Bucharest, in 1988. The first two years after graduation, she worked at Computer Manufacture Factory in Bucharest. At present she works as a researcher at the Research Institute for Informatics in Bucharest.
Her fields of interest include real-time systems, networks, CASE systems, object-oriented design and programming, modern user programs interface. She is a member of DECUS.

Savina Teodorescu was born in Bucharest, Romania in 1956. She received her M.Sc. in Computer Science from the Polytechnical Institute of Bucharest in 1980. From 1980 to 1986 she worked at the Research Institute for Computing Technique, Bucharest. Since 1986 she has been employed at the Research Institute for Informatics, Bucharest.
Her main fields of interest are in real-time systems, open systems, integrated industrial systems, databases.

1. Introduction

It is by now proved that the development and implementation of real-time computer process inspection and control systems need a very complex applied software be designed and large teams of specialists be involved.

Several years of experience in the field of industrial informatics have contributed to the development and current use of a conceptual frame architecture and of a set of software tools based on very flexible modular architecture and methodology. Consistent and well-defined data management methods as well as different types and configurations of computing equipments (PLC programmable logic controllers, distributed control systems, microprocessor controlled weighing/dosage equipments and/or other industrial equipments, 16/32-bit minicomputers, etc.) have been dedicated.

NONSTOP-32 is meant for ensuring non-interrupted operation of a computer system consisting of two minicomputers (32-bit CORAL 8730 or DEC-VAX compatible minicomputers). One of the two computers is MASTER and the other one is SLAVE.[3]

MASTER and SLAVE labels can be now and then attributed to each of the two minicomputers and this depends on the operation mode of each minicomputer.

The two minicomputers are linked by a high speed communication connection (which is a DEC-DR 11 compatible parallel connection interface DI-40) [6]. This connection makes it possible to update the process database existing on the SLAVE minicomputer. A specialized program can detect any malfunctioning of MASTER minicomputer and switch on the SLAVE minicomputer to MASTER system. The latter will continue to gather and process input/output data flow, with minimal loss of information.

2. The NONSTOP-32 System

Hardware Components

NONSTOP-32 exerts a reàl-time distributed industrial process control in metallurgy, also accepting fault- tolerance (i.e. severe faults and faults due to the system failure)[4].

At the first level of the hierarchy of the industrial process control a series of PLC personal computers, (PDP-11 compatible) CORAL-4021 minicomputers are installed, while at the second level of the hierarchy two (VAX compatible) CORAL 8730 computers are installed. All these equipments are connected to a DECnet network [2], [3].

Fault-tolerance is possible at the first level by PLC redundancy and by a hard switching when faults occur.

Fault-tolerance at the second level of the hierarchy is permitted by two (VAX compatible) CORAL 8730 computers, by a bus switch, by two (DR 11-DEC compatible) DI-40 boards, one board for each computer.

One of the two computers is MASTER, the other one is SLAVE.

When the MASTER computer fails, a bus switch from MASTER to SLAVE takes place.

The MASTER computer controls the whole industrial process. The RTDO-RDB [1] real-time process database is contained. In order to quickly save this database, its most recent updating is periodically transferred to the SLAVE computer via a high speed parallel line and a DI-40 interface [6].

The DI-40 interface allows the data transfer between the two computers.

The DI-40 interface is programmed by a processor. After initiating the processor-made transfer (by setting the word count register, bus address register, commands states register), the processor's intervention comes to an end. DI-40 has five registers.

The two computers communication will be facilitated by the WCR word count register, BAR bus address register, CSR commands states register, DBR data register.

There is a half-duplex connection between the two computers, fully provided by flat cables and using a DI-40 interface. In case of data transfer, this connection let one system interrupt the second one.

The DI-40 interface allows that any of the two computers (either MASTER or SLAVE) asks for the data 'transfer. The data flow is bidirectional. The two systems can alternatively act as transmitter or receiver.

Software Components

From the software point of view, fault-tolerance is possible by data and code duplication on the two (VAX compatible) CORAL 8730 computers [9].

Data duplication is a real-time process database duplication.

Code duplication will be a duplication of the NONSTOP programs enabling a continuous transfer of the process database, of the DI-40 interface driver, of the process database storage procedures. In this respect, MASTER and SLAVE programs will be emulated by a virtual disk in the computer memory and their interswitching will be possible.

The process data are stored (collected) in a RTDO-RDB real-time process database [1]. For fast access to it, the process database is stored on the CORAL 8730 computer memory. A disk emulating technique in the computer memory is used.

The RTDO-RDB real-time process database is passively duplicated on the SLAVE computer.

The MASTER computer memory stores a map for the RTDO-RDB real- time process database. The map's number of bits is equal to the number of records in the process database. Each bit of the RTDO- RDB realtime process database map corresponds to a process database record ("0" - the record hasn't been updated since the last successful record transfer, "1" - the record hasn't been updated since the last successful record transfer).

For a quick duplication of the RTDO-RDB real-time process database only updated records will be transferred [5].

In order to transfer the RTDO-RDB real-time process database at a process convenient time, the NONSTP1 and NONSTP 2 programs run periodically in the NONSTP 1 and NONSTP 2 processes on MASTER computer and SLAVE computer, respectively.

The NONSTP 1 program running on a MASTER computer reads the real-time process database map and according to the map, the SLAVE computer will only be transferred the records which have been updated since its last running time. It will modify the process database by turning to 0 the bit corresponding to the successfully transferred record. In case of transmission fault, the transmission is resumed.

The records on the MASTER computer which have been transferred by the NONSTP 1 program are received by the NONSTP 2 program running on the SLAVE computer. The NONSTP 2 also updates the RTDO-RDB memory resident real-time process database. If the NONSTP 2 detects a fault on the MASTER computer by time-out error, after some time, it starts the NONSTP 1 process, MASTER program and stops the NONSTP 2 process.

The MASTER program running on the SLAVE computer defines the former SLAVE computer as MASTER computer in the DECnet network, if the MASTER computer fails.

After recovering a faulted computer, the NONSTP2 process (for receiving data) and the SLAVE program that defines the computer as a SLAVE computer in a DECnet network will start again. A DI-40 interface driver has been designed for transfer purposes.

The NONSTP 1 and NONSTP 2 programs will run provided that the DI-40 interface driver should be installed on the system [7].

The YAMES routine will be used by the NONSTP 1 and NONSTP 2 programs for transmitting and receiving a record in the process database. The YAMES routine is duely synchronized. Its programming procedure is as follows:

- the transmitter (MASTER) computer sends an interruption to the receiver (SLAVE) computer;
- the receiver (SLAVE) computer prepares the reception of the number of bytes of the block that will be transmitted and gives an interruption to the transmitter (MASTER) computer;
- on receiving the interruption from the receiver computer, the transmitter (MASTER) computer will transmit the number of bytes of the block to be transferred;
- the receiver (SLAVE) computer sets the registers for receiving the data block that will be transferred by transmitter (MASTER) computer and sends an interruption to the transmitter (MASTER) computer;
- the transmitter (MASTER) computer receives this interruption and starts the transmission of the whole block to be transmitted.

Thus the data block transfer and the due synchronization take place.

The parameters of the YAMES routine are the following:

- the address of the data block (the process database record, in our case) that will be transferred and its size;
- the transfer direction (0 for transmitting, 1 for receiving);
- a flag for transfer;
- iosb that will contain the state of transfer: an error or a success code (the error can be time-out error, parity error, driver error, incomplete operation).

The Control System

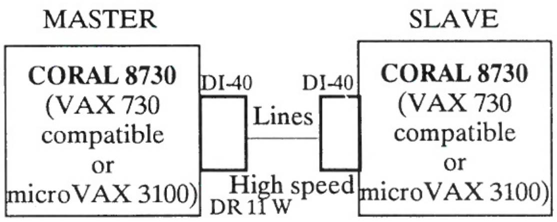
The pellet production is a multivariable process characterized by delay units, dead time units and nonlinear units. It is composed of dosing, grinding, burning and pellet producing subprocesses.

Process control is achieved through a complex multivariable system controlling the loading channel, the humidity channel and the pellet cycle channel using advanced algorithms. Based on the mathematical process model, a controller was designed so that the control loops should be decoupled [10], [11].

The Control System implements digital process control functions through algorithms performing classic control types such as P, I, PI, PID or other special types [12].

The designed control algorithms aim not only at getting a satisfactory output, but also at observing the time evolution of the command, because this affects the life cycle of the execution unit, the evolution of the controlled item between the sampling moments, the addiction to non-linear regimes, generated by saturation, etc.

In case of switching on the reservation system, the algorithms will avoid to produce commands which might be striking; strong oscillations will get the execution unit worn-out faster [8].



(Including RTDO-RDB Database Developed in CORAL 8730 Storage)

3. Applications

The implementation of this system at several metallurgical plants in Romania (Calarasi, Hunedoara, Resita), in India (Mangalore), and Ukraine(Krivoi-Rog, under way) accounts for its advantages: limited efforts required in developing a new application in a short period of time and high efficiency in meeting the fault-tolerance requirements of the applications.

4. Conclusions

NONSTOP-32 is suitable for a large variety of real-time fault-tolerant applications, of which configuration covers computers and various process control interface equipments, such as programmable logic controllers, distributed process control systems, dosage and weighing devices and other special interfacing devices.

Flexibility and modularity associated with the efficient data acquisition and management and with a built-in support for the main functions of the metallurgical applications are the main features of the system.

REFERENCES

1. **RT-ARCH - A New Approach in Real-Time Application Design.Mini-and Microcomputers and Their Applications,** Lugano, 19-21 June 1990
2. **VAX/VMS Networking Manual and VAX/VMS Network Control Program,** DIGITAL
3. **DECnet-DOS Network Management Manual,** DIGITAL
4. POWELL, D., **The Delta-4 Approach to Dependability in Open Distributed Computing Systems** (FTCS-18)
5. **ESPRIT-Information Processing Systems and Software**
6. **The DI-40 Board,** Computer Manufacture Factory Bucharest
7. **Writing a Device Driver for VAX/VMS,** DIGITAL
8. DAVIDOVICIU, A., **MIX and MACRO**
9. LAPRIE, J.C., **Dependability: A Unifying Concept for Reliable, Safe, Secure Computing,** IFIP Congress, Madrid, 1992
10. **Algorithms, Software, Architecture, Information Processing,** IFIP Congress, Madrid, 1992
11. NAKAMURA, H. and TAKESHI, K., **Fault-Tolerant Microcomputer Design and Application for Railway Train Control,** IFIP Congress, Madrid, 1992
12. CALIN, S., **Digital Tuning of Technological Processes**