International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:6, 2014

Proposal to Increase the Efficiency, Reliability and Safety of the Centre of Data Collection Management and Their Evaluation Using Cluster Solutions

Martin Juhas, Bohuslava Juhasova, Igor Halenar, Andrej Elias

***Abstract*—**This article deals with the possibility of increasing efficiency, reliability and safety of the system for teledosimetric data collection management and their evaluation as a part of complex study for activity "Research of data collection, their measurement and evaluation with mobile and autonomous units" within project "Research of monitoring and evaluation of non-standard conditions in the area of nuclear power plants". Possible weaknesses in existing system are identified. A study of available cluster solutions with possibility of their deploying to analysed system is presented.

***Keywords*—**Teledosimetric data, efficiency, reliability, safety, cluster solution.

I. Introduction

THIS contribution solves a partial problem falling within an issue of research of monitoring and evaluation of non-standard conditions in the area of nuclear power plants. It is focused on the possibility of increasing efficiency, reliability and safety of the system for teledosimetric data collection management and their evaluation. Based on research of communication processes between units of measurement and control center for data collection the new knowledge will improve communication technologies and server of the measurement and evaluation data system.

II. Specification of the Teledosimetric System

A. Basic Specification

The purpose of TeleDosimetric System (TDS) is a continuously monitor of gamma radiation dose rate, volumetric activity of aerosols, volumetric activity of radioiodine (it has two modes: background screening unit measurement mode - standby mode for normal operation mode and measurement activities in emergency situations) and additional data on state of the technology.

In all measuring stations TDS are continually rated gamma radiation dose rate, volume activity radioisotopes of iodine and volume activity of aerosols in the atmosphere. There are 24 pieces detectors with lead collimator for determining the dose rate from a possible leak of radioactive cloud deployed in areas A and B, in vicinity (200-400m) from the building where reactors are located. Space solution of measurement points layout at 4 m from the ground meets the requirement to record and evaluate each leakage of radioactive substances into the atmosphere by each path beyond ventilation stack.

M. Juhas is with the Institute of AIAM FMST SUT in Trnava, Hajdóczyho 1, 917 01 Trnava, Slovak Republic (phone: +421 918 646 021; e-mail: martin_juhas@stuba.sk).

B. Juhasova, I. Halenar, and A. Elias are with the Institute of AIAM FMST SUT in Trnava, Hajdóczyho 1, 917 01 Trnava, Slovak Republic (e-mail: bohuslava.juhasova@stuba.sk, igor.halenar@stuba.sk, andrej.elias@stuba.sk).

B. The Main Technological Parameters

Total number of measurement stations – 24, distributed in three areas:

- 5 stations – inside areal circuit
- 15 stations – outside areal circuit (3-6km)
- 4 stations – outside areal circuit (6-15km)

Monitored variables:

- gamma radiation dose rate
- volume activity of aerosols
- air flow for aerosols volume activities measurement
- volume activity of radioiodine
- object temperature
- temperature of a pump electromotor
- battery voltage
- the presence of power supply

Number of measurement points

- area A–24
- area B–24

Monitored variables:

- gamma radiation dose rate from the cloud

Scanning cycle:

- TDS – 300 sec.
- main area measurement:
  - Immediate values every 3sec.
  - Average values every 300 sec.
- waste water:
  - Activity< 20 Bq/l every 60min.
  - Activity> 20 Bq/l every 300sec.

Data archiving:

- immediate data – last 20 items
- monthly data – composed of hourly averages

The network is composed by standard PC LAN, which includes an object X850 of Radiation Monitoring Laboratory (RML).

PC stations installed in TDS:

- control computer situated in TDS control room (object X850 RML)
- 2 node computers located in A and B areas
- 5 collecting computers located at individual stations in whole areal

A control computer provides:

- management of sequence of collected data from telemetric network deployed in and around the areal
- conversion of received values from the individual detection systems to physical variables
- archivation of specified values and report generation
- monitoring of minimal and maximal permitted values
- providing the transmission of measured values to the selected workplace
- collecting of required values from individual reactor blocks and sites of complex

C. Data Collection Management

The required data collecting is divided into two types: managed and unmanaged collection. In the first case, it is a data acquisition from sensors at the request of the TDS server, in the second case they are the data that are automatically sent to the server periodically.

Teledosimetric stations in the complex area belong to managed collection category. Data collection management system with structure shown in Fig. 1 uses an endpoints configuration database. This database provides information about network topology, transmission routes, necessary drivers, data validity range, as well as data collection intervals.

Data collection is organized in groups by location. This means that at first a data from all sensors at first area are received, then at second area, etc. In the case that some sensor data are unavailable, a record of this sensor is added into the queue of temporarily unavailable data. The system automatically attempts to get data for sensor in this queue at the end of the collection period (after receiving value from the last available sensor).

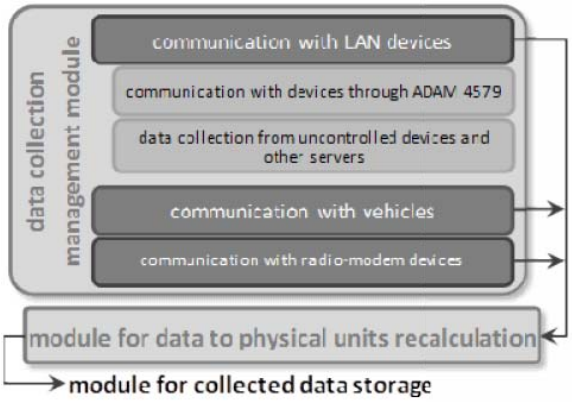

Fig. 1 Scheme of data collection management

D. Data Processing and Storage

The structure of data processing and its storage is shown in Fig. 2.

The main TDS server (master) makes available to secondary TDS server (slave) information about current state of the queue, as well as received data. Based on this information a slave server is creating its own copy of the received data database.

Received data are stored in the local database of received data after processing. The data from this database are periodically exported to the institution main database, including function of data adding after link failure.

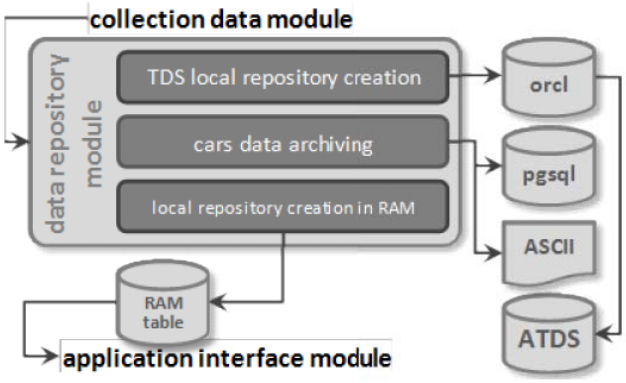

Fig. 2 Scheme of data processing and storage

E. Server and Sensor Structure Configuration

Configuration of sensors structure, as well as some other parts of the server can be performed through the web interface. Through this interface is also possible to view current status of the system and error reports.

The main advantage of this solution is that it eliminates the need of complex software installation on the user (administrator) workstation. It also eliminates the need to log on to the server directly. The protection is provided by an enforced using of encrypted connection (HTTPS), by limiting access for selected IP addresses only and of course by logging in (username, password).

The configuration is automatically mirrored to secondary TDS server after saved, so it remains valid even after a possible crash of one of the servers [6].

III. Identified Risks of Existing Solution

The analysis of the existing system solutions results in the following potential risks:

- The database server TDS using PostgreSQL does not contain the optimal problems solution in case of a database server unexpected failure.
- Data collection management server contains ineffective solutions for functionality failures of individual elements and communication between subsystems.
- The compactness of system of data collection and their distribution to clients bear the risk of system flooding resulting in a possible total failure of required data distribution.
- Centralized solution of the whole system has a significant impact on the transmission path load, that is associated with the risk of transmission channel overloading, followed by functionality reducing, or even the system basic functions failure.

IV. Clusters Classification

A. High-Availability Clusters

High-availability clusters (HA clusters), also known as failover clusters, are groups of computers that support server applications that can be reliably utilized with a minimum of unavailability. The principle is inclusion of redundant computers in groups or clusters that provide continued service when any system component fails (Fig.). If there is no clustering and a server running specific application crashes, the application will be unavailable until the crashed server is fixed. HA clustering eliminates this situation by detecting hardware/software faults, and immediately restarting the application on another system without requiring administrative intervention. As part of this process, clustering software may configure the node before starting the application on it. HA clusters are often used for critical databases. HA cluster implementations try to build redundancy into a cluster to eliminate single points of failure, including multiple network connections and data storage which is redundantly connected via storage area networks (SAN). HA clusters usually use a heartbeat private network connection which is used to monitor the health and status of each node in the cluster.

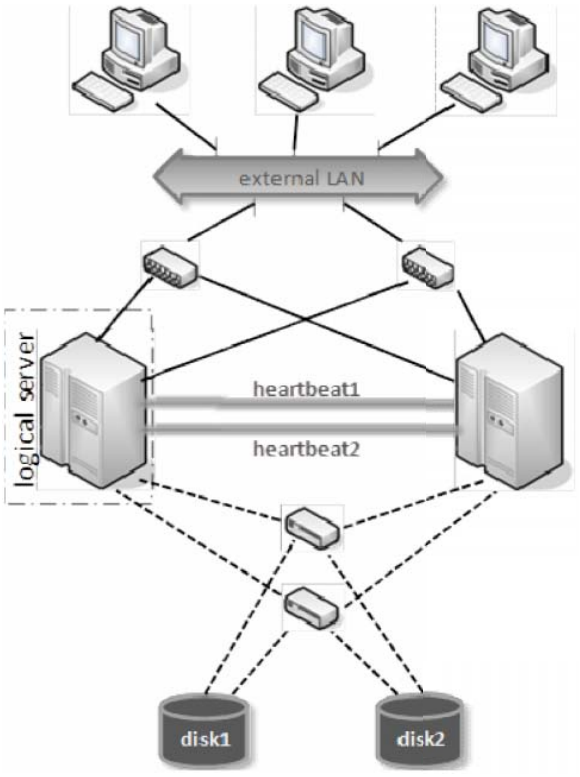

Fig. 3 High-availability cluster

It is usually used more than two nodes involved in the cluster in practice, and there are different methods to configure these nodes (Active/Active, Active/Passive, N +1, etc.).

HA cluster concept is not suitable for any type of applications. Its use must be considered already in the early design phase of a software product.

To be able to deploy HA cluster technology, the application must satisfy following minimum specifications:

- There must be a relatively easy way to start, stop, force-stop, and check the status of the application. This means the application must have a command line interface or scripts to control the application, including support for multiple instances of the application.
- The application must be capable to use shared storage (NAS/SAN)
- The application must store maximum of its state on permanent shared storage as is possible.
- The ability to restart application on another node at the last state before failure using the saved state from the shared storage must exist.
- The application must not corrupt data in case of crash, or restart from the saved state.

The last two of specifications are critical to its reliable function in a cluster, and are the most difficult to satisfy them fully.

The most frequently type of an HA cluster is a two-node cluster, since that is the minimum required to provide redundancy. Many clusters consist of more, sometimes dozens of nodes. Cluster configurations can be categorized into one of the following models:

- Active/Active – Traffic intended for the failed node is concurrently passed onto an existing node or load balanced across the remaining nodes. This is usually only possible when the nodes are deployed on a homogeneous software configuration.
- Active/Passive – Provides a fully redundant instance of each node, which is only brought online when its associated primary node fails. This configuration typically requires the most extra hardware.
- N+1 – Provides a single extra node that is brought online to take over the role of the node that has failed. In the case of heterogeneous software configuration on each primary node, the extra node must be universally capable of assuming any of the roles of the primary nodes it is responsible for. This normally refers to clusters which have multiple services running simultaneously.
- N+M – In cases where a single cluster is managing many services with solitary dedicated failover node, this FO node may not offer sufficient redundancy. In such cases, more than one (M) standby servers are included and available. The number of standby servers is a trade-off between cost and reliability requirements.
- N–to–1 – Allows the failover standby node to become the active one temporarily, until the original node can be brought back online, at which point the services or instances must be failed-back to it in order to restore high availability.
- N–to–N – A combination of active/active and N+M clusters, N to N clusters redistribute the services, instances or connections from the failed node among the remaining active nodes, thus eliminating the need for a

standby node as in active/active case, but generates a new need for extra capacity on all active nodes.

HA clusters usually utilize all available techniques to ensure the reliability of systems, which include:

- Disk mirroring - failure of internal disks does not result in all system crashes
- Redundant network connections - single cable, switch, or network interface failures do not result in network malfunction
- Redundant storage area network or SAN data connections - single cable, switch, or interface failures do not lead to loss of connectivity to the storage
- Redundant electrical power inputs on different circuits, usually both or all protected by uninterruptible power supply units, and redundant power supply units - single power feed, cable, UPS, or power supply failures do not lead to loss of power to the system[1], [2], [3].

B. Load-Balancing Clusters

Load-balancing clusters (LB cluster, Load Balancer) are used to increase system performance by tasks distributing among multiple computers connected to the cluster (Fig. 3). A different service on each computer, that is a part of the cluster, is implemented. The cluster behaves as a virtual server (all physical servers encapsulated), so user does not know that the server, requesting specific service to, is composed of multiple computers in fact. The requests handling ensures the load balancer with which user comes into contact. It shall take the request and then sends it to the appropriate server that carries it out.

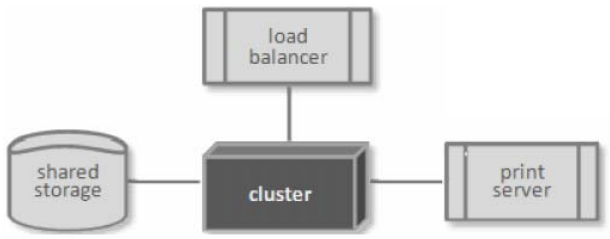

Fig. 3 Load-balancing cluster

In network solutions a load balancing is a methodology of distributing workload across multiple computers or computers in a cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Instead of a single component using, a multiple components with load balancing may increase reliability by redundancy. The load balancing service is usually provided by dedicated software or hardware.

For network services, the load balancer is usually a software application that is listening on the port where external clients connect to access services. The load balancer forwards requests to one of the backend servers, which usually communicates with the load balancer. This allows the load balancer to reply to the client without the client ever knowing about the internal separation of functions. It also prevents clients from contacting backend servers directly, which may have security benefits by hiding the structure of the internal network and preventing attacks on the kernel's network stack or unrelated services running on other ports.

Some load balancers provide functionalities that are able to react when all backend servers are unavailable. This might include forwarding to a backup load balancer, or displaying an error message of the failure.

An important thing when using a load-balanced service is how to handle information that must be stored across the multiple requests in a user's session. If this information is stored locally on one backend server, then following requests going to different backend servers would not be able to find it. This might be cached information that can be recomputed, in which case load-balancing a request to a different backend server just means a performance consequence.

One of possible solution of this problem is to send all requests in a user session consistently to the same backend server. This is known as persistence or stickiness. A significant disadvantage to this technique is its lack of automatic failover. If a backend server crashes, its per-session information becomes inaccessible, and any sessions depending on it are lost.

Another possible solution is to keep the per-session data in a database. This option leads to performance decreasing because it increases the database load. The database is best used to store information less transient than per-session data. To prevent a database from becoming a critical single point of failure, and to improve scalability, the database is often replicated across multiple machines, and load balancing is used to spread the query load across those replicas. Microsoft's ASP.net State Server technology is an example of a session database. All servers in a web farm store their session data on State Server and any server in the farm can retrieve the data.

Load balancing is often used to implement failover functionality. The system components are monitored continually, and when one becomes non-responsive, the load balancer is informed and no longer sends traffic to it. And when a component comes back online, the load balancer begins to route traffic to it again. This is conditioned by an existence of at least one component in excess of the service's capacity. This can be much less expensive and more flexible than failover approaches where a single live component is paired with a single backup component that takes over in the event of a failure.[4], [5].

V. Analysis of Implementation Possibilities of Selected Cluster Technologies into the System and Evaluation of their Impact on System Properties

In view of the identified potential risks of the current configuration of data collection management system and evaluation, the following possibilities of using selected cluster technologies into the system to eliminate these risks, were analysed.

A. TDS Database Server

The TDS database server implemented as PostgreSQL can be solved as a high-availability cluster solution, for example,

as MySQL Cluster Data Node Daemon. With regard to the rate of data increment, an active/active configuration is proposed to use (Fig. 4). Due to the database server is not loaded by extreme performance requirements, it is not necessary to implement load-balancing solution.

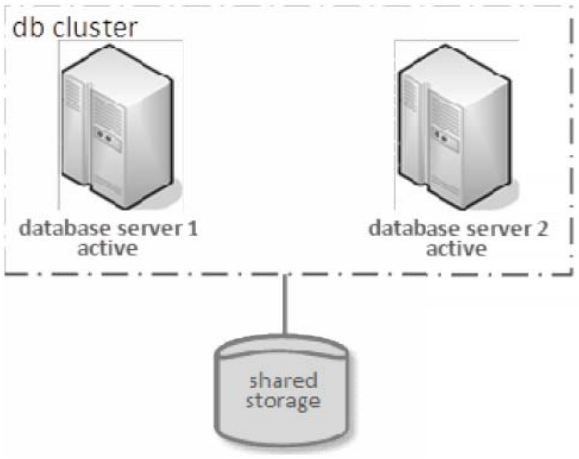

Fig. 4 Proposal of HA solution for TDS database server

B. Data Collection Management Server

The data collection management server can be solved as high-availability cluster solution with at least two servers. The proposed implementation is using OpenSSI with web-based administration of the cluster. There are no significant stress peaks during the data collection management server operation, thus there is no need for the load balancing implementation.

C. Data Distribution

The data distribution to clients can be solved as load-balancing cluster solution to ensure uniform load distribution at potential increasing of the system requirement number. The proposed solution can be implemented, for example, as JBoss load balancer (Fig. 5) or load balancer of Resin cluster.

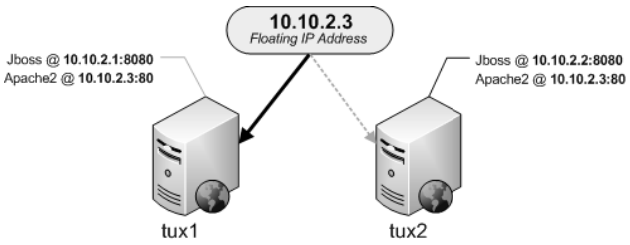

Fig. 5 Jboss Load-balancer for data distribution

D. System Decentralization

Additional proposal for system optimization is the decentralization of individual subsystems in the structure:

- Firstly one database archive and distribution server situated to Radiation Monitoring Laboratory of wider surroundings for the needs of the emergency centre.
- Secondly next database archive and distribution cluster situated in areal. Its function would be requirements from all clients processing that would leads to load of transmission path reducing.

VI. Conclusion

The possibilities of increasing efficiency, reliability and safety of the system for teledosimetric data collection management and their evaluation as a part of complex study within project "Research of monitoring and evaluation of non-standard conditions in the area of nuclear power plants" was presented. A list of possible weaknesses of existing system was identified. A study of available cluster solutions with possibility of their deploying to analysed system for weaknesses elimination was presented.

Acknowledgment

This publication is the result of implementation of the project: "Research of monitoring and evaluation of non-standard conditions in the area of nuclear power plants" (ITMS: 26220220159) supported by the Research & Development Operational Programme funded by the ERDF.

We support research activities in Slovakia.
The project is co-financed from EU resources.

References

[1] P. S. Weygant, *Clusters for High Availability: A Primer of HP Solutions*. Upper Saddle River, NJ: Prentice-Hall, Inc., 2001.
[2] A.Davies, *High Availability MySQL Cookbook.* Birmingham, UK: Packt Publishing Ltd., 2010.
[3] O. Lascu, S. Bodily, M.-K. Esser et al.,*Implementing High Availability Cluster Multi-Processing (HACMP) Cookbook.* International Business Machines Corporation, 2005.
[4] T. Bourke, *Server Load Balancing*. Sebastopol, CA: O'Reilly & Associates, Inc., 2001.
[5] B. Ediger, *Advanced Rails.* Sebastopol, CA: O'Reilly & Associates, Inc., 2007, ch. 4.6.
[6] A. Elias, P. Tanuska, P. Vazan, "The dose rate monitoring system through the mobile units," *Materials Science and Technology*, vol. 5, no. 5, 2005.