Daniil Nikishyn (Fr. Skorina GSU, Gomel)
Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

CLUSTER STORAGE DATA SYNCHRONIZATION

Information is the most valuable asset nowadays, so there is a need to store, reserve and synchronize it. Clustered distributed file systems amount began to grow in the very fast paste when almost all the data in the whole world became digital, therefore the system is relevant. The product will be useful for any company and any production that needs reliable, fast data storage with guaranteed data safekeeping.

If there is a need to provide the endpoint user with fast and reliable access to the web service or the server cluster it is obvious to choose the representational state transfer architecture development style for the application. System's data replication makes it fault-tolerant using commodity hardware and requiring no specific hardware support. As a result of this design, the system is both self-healing and self-managing, aiming to minimize administration time and other costs.

Developed system increases the availability of data by storing the data on multiple servers making this process transparent by presenting to users what appears to be a single folder in the namespace and keeping the data synchronized on the servers. If one of the servers hosting data is unavailable, clients are referred to another server that hosts the data.

Each application's process, daemon or utility draws its configuration from several sources on startup, include a local configuration, the monitors, the command line, or environment variables. Configuration options may be set globally such that they apply to all daemons, to all replicas or nodes of a particular type, or only to a specific node or replica.

This is a big problem for businesses that need to create storage spaces to store a lot of data. Developed system on the other hand, does not require the files to be cloned locally. Rather, when a file is accessed, the data is taken from the fast-performance cache. This also means that mounting the clustered file system gives the user immediate access to all the hierarchy no matter the amount of data it contains.

The server access log records all requests processed by the server. The location and content of the access log are controlled by the configuration file. The application simply selects and logs all the operations with the contents of the storage. The system records information in the access log and then stores it in the specific database table on the each element of the clustered system.

One of the advantages of the system is the transactional control that is used with the CRUD (create, read, update, delete) operations. That means all the operations are automatically committed in the database only if they are successful, otherwise all the changes are getting rollbacked.

Daniil Nikishyn (Fr. Skorina GSU, Gomel)
Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

ARCHITECTURE OF CLUSTER STORAGE DATA SYNCHRONIZATION SYSTEM

Web application is developed in Java programming language, Servlet API 3.0 is used from the Java EE specification.

Representational State Transfer architecture is one of the best and convenient ways to implement the functionality of the web service, therefore this solution was chosen for my application. In this web service, data is stored in the cache and the database as JSON objects. All the requests made to a resource's URI will elicit a response in JSON format. Each request from a service consumer contains all the necessary information for the service to understand the meaning of the request, and all session state data is then returned to the service consumer at the end of each request.

The developed application has predefined create, read, update and delete HTTP methods. HTTP GET is used for retrieving the data; HTTP POST creates new subordinate resource into the collection of resources; HTTP PUT updates existing resource, if resource does not exist then a new resource is created; HTTP DELETE is used to delete resources, identified by the Request-URI.

Every component of the cluster works independently from the others, which can be deployed separately on multiple environments with different operating systems and various set of the hardware. All the components have synchronization mechanism that allow to keep cluster data consistency and guarantees storage data safekeeping. Last recently used cache implementation, designed as a hashmap, gives quick access to the data by keys that is stored in the node. If requested element is not in the cache then the system consumes it from the database. Access level control logging provides fail-safe cluster workflow.