Simulation and Control of Computational Cluster Components Based on Parallel Mechanisms of the Caper Language

Sergey Vartanov
Institute for Informatics and Automation Problems of NAS RA
Yerevan, RA
e-mail: s.vartanov@iiap.sci.am

Veronika Minasyan
Institute for Informatics and Automation Problems of NAS RA
Yerevan, RA
e-mail: veronika.minasyan@edu.isec.am

***Abstract*—The article deals with the problems of creating a software environment for preparing projects of job realization on computational clusters by means of software simulation of the work of their main components: nodes, control machines, and storage. The second task of the software environment is to develop an interface for monitoring the state of the real cluster, job queues, etc., accumulating statistical information on job execution, failures, etc., and providing means for analyzing the accumulated information. The article describes some already implemented solutions and approaches to the development of the software system.**

***Keywords*—Computing clusters, cluster simulator, development and analysis environment, cluster monitoring and management.**

I. Introduction

Modern computing clusters are complex distributed systems that provide high-performance computing for scientific, industrial and commercial tasks. As the scale and complexity of such systems grow, the need for effective management, monitoring and control of their components increases.

The actual task becomes the development of tools that allow both managing real cluster components and modeling their behavior for testing and optimization purposes.

The purpose of this work is to investigate the problem and develop a program complex providing simulation of the work of components of a computing cluster using parallel mechanisms of the Caper (dynamic Compilable, Asynchronous, Parallel Events Routines) language [1], to solve the problem of simulating the work of the cluster as a whole and collecting information in the environment of real clusters. The possibilities of system control using MPI (Message Passing Interface) [2] and Slurm (Simple Linux Utility for Resource Management) [3] tools in the wrapper functions of the library are also investigated.

To achieve the goal, the following tasks are being solved:

- simulation of the work of all cluster components in a software environment,
- creating a visual interface with node status display,
- organization of data collection from real equipment,
- implementation of task management mechanisms and analysis of their fulfillment,
- a research of cluster behavior under different load and control scenarios.

II. Program Model and Its Features

A program is created that reflects all key components of the cluster including computational nodes, storage, and control machines. Using parallel mechanisms of Caper language, the simulation of the simultaneous work of all components is realized, which provides a realistic reproduction of complex cluster scenarios.

In addition to simulation, the system collects information from a real cluster with binding to specific nodes and storage units. This allows conducting validation of the model and collating the behavior of the simulated and physical environment.

Special attention is paid to the possibilities of cluster management by means of the developed environment for cluster monitoring and management. For direct interaction with the cluster, a library consisting of wrapper functions realizing Slurm command calls and MPI functions is being developed. The simulation part of the program should not only simulate the operation of nodes, but also provide full support for task scheduling scenarios, computational resource management and analysis of the system behavior under different conditions.

In the course of the work, the characteristics of the cluster are studied when performing typical computing tasks. The system load indicators are analyzed based on statistical data, as well as changes in the cluster behavior over time. Particular attention is paid to assessing the impact of various control schemes on the overall efficiency and stability of the system.

III. Using MPI and SLURM in Cluster Management

To organize interaction between the simulation and task management processes in the developed system, two key tools are used:

- MPI is a standard for organizing interaction between processes in a cluster. It allows data to be transferred

https://doi.org/10.51408/csit2025_18

between cluster nodes through message exchange, which is important for parallel computing.

- SLURM is an open source job scheduler that provides a platform for job management, scheduling, and resource allocation in high-performance computing environments.

Integration of these tools makes it possible to implement a hybrid approach: simultaneously analyze both simulated and real environments within one software package.

IV. Analysis of Cluster Behavior and Load Dynamics

The developed system provides opportunities for complex analysis of cluster behavior. It examines node utilization based on statistical data, the effectiveness of different task management schemes, and the cluster behavior when the load changes over time. For example, it is possible to track how resource distribution changes, which nodes are overloaded, how task queues affect overall performance, and what events occur in the cluster over long periods of time.

This approach allows not only to assess current performance characteristics, but also to predict the consequences of management decisions, optimize planning, and improve system resilience.

The developed environment enables the dynamic connection of various software modules in the Caper language, implementing various schemes of behavior and analysis. Independent modules realize direct access to the real cluster, collection and processing of real information, as well as the execution of control actions in relation to the cluster components.

V. Implementation Scheme Of The Program Shell

Simulation of the cluster operation starts with launching parallel processes of the Caper virtual machine by means of the developed graphical interface. Each node of the cluster, the controlling machine, corresponds to a parallel process, which is engaged in continuous or through a quantum of time analysis of the state of its ward (in the current version, 620 processes corresponding to the structure of the cluster “Aznavour” with 612 nodes). In this case, it is possible to derive an integrated description of the entire cluster and quantitative indicators of the resources involved in it.

The interface provides output of text information with node description by clicking on the node icon, as well as output of actions of each control machine in separate windows (in the so-called System Monitors).

In addition, a component has been developed that provides viewing and management of queues, accumulators of withdrawn or accidentally completed jobs, ˝sump˝, and so on.

Means of describing scenarios of job behavior in the cluster and its reactions to certain events are being developed, as well as means of access to monitoring functions of an existing cluster.

VI. Conclusion

The program complex under development today consists of a shell that enables the provision of all the aforementioned functions. As mentioned above, the stated functions are or will be realized by independent program modules, which will be loaded as they are ready. Currently, access to the functions of these modules is provided either by the actual program code or by stub functions.

References

[1] Вартанов С.Р. Распределение вычислений для многоядерных/многопроцессорных вычислителей на основе системы виртуальных машин языка параллельного программирования Caper. - Параллельные вычислительные технологии – XV международная конференция, г. Волгоград, с. 176–185, ПаВТ'2021.

[2] [Online]. Available: MPI Forum Documents. https://www.mpi-forum.org/docs/mpi-5.0/mpi50-report.pdf, https://www.mpi-forum.org/docs/

[3] [Online]. Available: SLURM Documentation. https://slurm.schedmd.com/