HYBRID APPROACH TO DESIGN OF STORAGE ATTACHED NETWORK SIMULATION SYSTEMS

M.E. Karpov, K. Arzymatov, V.S. Belavin, A.A. Sapronov and A.E. Ustyuzhanin
National Research University Higher School of Economics, Moscow - 101000, Russia

A.V. Nevolin
YADRO, Moscow - 123022, Russia

ABSTRACT

Simulators of real-world IT systems are gaining popularity today. However, as it often happens in the early stages of technological readiness, the same term can be understood as different things - from visualisation systems to multi-level multi-agent models. The critical feature of the simulation technology is the degree of trust, or proximity of resemblance of their behaviour to the objects of simulation from the real world. The article presents for the first time an overview of a hybrid approach to modelling Storage attached networks (SAN), in which the parameters of an approximate simulator are dynamically adjusted using machine learning methods, i.e. reinforcement learning. Particular attention is paid to the analysis of the strengths and weaknesses of the existing approaches of simulation and comparison the hybrid approach presented in the article.

Keywords: simulator, system modelling, control optimisation, reinforcement learning, storage attached network


1. INTRODUCTION

The most common form of discrete simulation programs is modelling of sequence of discrete events in which events of interest occur at discrete points in time. Discrete events include for example user logon, errors, system failure, repair or replacement of components, etc. More complex simulations require more development effort, but the principle remains the same.
A chain of events is created, which is ordered by time in the transition from one event to another, and in parallel statistics with changes in system parameters are collected.

The steps needed to build a correct discrete simulation algorithm:

- carefully analyse which components the system consists of, how these components interact with each other under different loads and other external conditions;
- list and classify events of interest;
- determine dependencies and causal relationships between events;
- determine the system statuses, methods of their description and transitions between them;
- to estimate the ranges and distributions of the input parameters of the simulation;
- determine the observed values, set up the collection of statistical data on the functioning of the simulation program.

An additional complication is the need to take into account the distribution of modern systems operation in which the definition of “state” can be blurred between possible different configurations of individual components.

Following these notes consider an example of already existing storage system's simulator, namely, a software package of a CODES project [1, 2]. This project is developed by researcher teams from Computer Science and Math department of Argonne National Laboratory and Rensselaer Polytechnic Institute (US). The CODES simulator is based on the technologies of the Rensselaer's Optimistic Simulation System (ROSS) which allows parallel execution of event-driven system that can significantly decrease runtime of the simulation. The main CODES use cases include large-scale storage systems, scientific distributed applications, parallel and high performance computing systems with high-load input/output operations and computational complexity. The C++ based ns3 framework [3] is also popular among scientific researchers. The ASL simulation toolkit offers broad range of options for defining own behavioural functions with subsequent deployment on parallel core architectures [4]. The general approach to network-like structure simulation is the OMNeT++ framework [5]. Another work presents a simulation compliant with the fiber channel technology often used in contemporary SAN architectures is developed as the SANSim tool [6]. More simulation methods descriptions and studies dedicated to SAN system modelling can be found in [7-11].

All of these works discuss the problem of storage system simulation from the deterministic point of view, having component parameters defined at the beginning of the simulation. The practical solutions for real-life storage systems should reflect a potential for components qualities alteration under user’s load, degradation or even faults.

The alternative method for generating events of hidden states is to use machine learning methods, notably generative adversarial networks (GANs) [12]. The GANs are quite popular nowadays and have shown promising results in various fields of science and techniques [13-16]. But the lack of huge training set can be a formidable challenge along with a poor predictive capability beyond the training domain. This approach cannot be used in case of small number of rare events.

The topic of simulation parameters optimization using reinforcement learning is generally discussed in [17] and similar approach is discussed in application to specific range of Process Network Synthesis simulation problems [18].
An example of a Storage Area Network failure events simulation algorithm [19] is shown at Fig. 1.

![Flowchart of SAN failure events simulation algorithm](image.png)

**Figure 1** An example of a SAN failure events imitation algorithm

Any simulation program requires the setting of some number of input parameters. According to the simulation results, it is often possible to obtain a very small error of the observed values in comparison with the characteristics of the real system due to the possibility of collecting a large amount of data. However, often the developers of the model have to keep a balance between its accuracy and development costs [20]. In this context, the best solution may be to use the following hybrid modelling approach.

2. SAN MODELLING

Below the hybrid approach to the modelling of storage system simulation is described. The deterministic model of the SAN storage is created and includes most important functional elements of the architecture. With the help of controlling agent, trained by reinforcement learning, this model is adapted to significantly improve the quality of simulation process.

A software package for operation simulation (SPOS) of the SAN and a set of algorithms for automated parameter settings (SAAPS) were developed to implement this idea. The SPOS describes the structure of a SAN and the interaction of its components. The tasks to be solved by the SPOS are the following:

- simulation of the storage components (storage controllers, disk chassis, PCI Express, load balancer);
- modelling of SAN in the read/write/store mode;
- implementation of anomaly simulation in the final disc-based media, storage controllers, connecting networks;
- imitation of synchronous/asynchronous interaction between client and storage.

Components shown in Fig. 2. have relationships as part of the simulation of the hardware of the storage at the logical level.
SPOS is a program written in Go, programming language, based on functions imported from the standard language library. The realization of the simulation is implemented on the basis of a discrete-event model, where the system changes represent a chronological sequence of events. The events occur at certain points in time and signify a change in the state of the system.

The tasks of SPOS are the following:

- execution of external control actions scenarios;
- generation of time sequences of SAN states depending on external control actions scenarios.

While running simulation at SPOS, SAN sends messages every discrete step of time as a JSON object to SAAPS. This service uses time sequences of data with the current state of the storage simulation to make decisions about such internal parameters of the simulator as:

- the effective bandwidth of network connections;
- network delay;
- computing resource of the storage controllers;
- read/write speed at disk drives.

Anomaly scenarios and degradation of the performance of SAN are also incorporated in controlling functions of SAAPS:

- degradation of the network;
- network disruption;
- degradation of computing resources;
- breakage of the storage controller.

These scenarios are also induced by machine learning algorithms of SAAPS (neural networks) trained on data from the real SAN of the corresponding configuration.

The usage of the combined scheme of the simulator and the neural network allows to bypass the fundamental shortcomings of the latter, which do not allow to make realistic predictions outside the training range. The possibility of extrapolation beyond the training...
sample is provided by the deterministic nature of the simulator, and the prediction accuracy with limited detail of the latter by the neural network.

The main advantages of this approach are low requirement for detailed simulation and, as a result, a significant reduction in the intellectual cost of its development. Compared to other approaches, parameter adjustment does not require major changes within developing service. Due to the use of methods of reinforcement learning [21], SAAPS allows you to configure the desired parameters online. Within the framework of reinforcement learning the following entities have to be defined:

- controlled object - the object under study;
- environment - the system that interacts with controlled object and which holds the object;
- agent – a system that can control the environment;
- reward – a measure that determines the goodness of the controlled object state change.

The goal of the reinforcement learning algorithm is to bring the controlled object to some specific target state by means of finding optimal control strategy.

In our case the environment and controlled object (SAN) is described by the simulator state and set of external (environmental) parameter, the target state is a degree of realism of the simulation of SAN, reward is a distance between real SAN behaviour and simulated behaviour. Agent’s functions are implemented by means of Q-learning algorithms and neural networks. It should be noted that training of such hybrid model requires non-negligible computational resources for numerous runs of different scenarios of SAN model behaviour.

The results of neural network training are shown in Fig. 3, which displays the average value of the metric on the set of loads used in training with a confidence interval. The smaller the value of the metric, the closer our system is to the reference one [22].

![Figure 3](image_url) The average value of the metric on the set of loads used in training with a confidence interval
3. CONCLUSION

Today there is a wide choice of modelling software tools for SANs. Their common feature is a detailed description of the structure of the simulated system and, thus, it is rather difficult to customise to a specific instance, and it requires a significant investment of time and human resources. In the approach, when all modelling is implemented solely by in-depth training methods, it is necessary to use a lot of data to train such models.

A simulator with automatic parameter tuning developed by the authors of this article is a synthesis of these two approaches, namely, forming the logical structure of the physical processes of the data storage system and using reinforcement learning methods that allow changing the parameters of the storage hardware components on the fly. Such a hybrid provides greater ease of setup and high realism even outside of the training sample ranges.

ACKNOWLEDGMENTS

The research was carried out with the financial support of the Ministry of Science and Higher Education of Russian Federation within the framework of the Federal Target Program “Research and Development in Priority Areas of the Development of the Scientific and Technological Complex of Russia for 2014-2020”. Unique identifier – RFMEFI58117X0023, agreement 14.581.21.0023 on 03.10.2017.

REFERENCES


Hybrid Approach to Design of Storage Attached Network Simulation Systems


