UDC 004.414.28

Andrii Boldak, Maksym Vlasov

METHOD OF IMPLEMENTATION OF SOFTWARE FOR SOLVING THE PROBLEM OF OPTIMIZATION OF CONFIGURATIONS OF THE DISTRIBUTED INFORMATION SYSTEM

The article examines how to implement software for solution of the problem of optimizing the configurations of the distributed information system. This method aims to obtain the optimal configuration in the form of a code from the input parameters.

Key words: distributed information system, DIS, optimization.

Relevance of research topic. Modern distributed information systems either already use cloud service providers, or soon begin to use them [1] [2]. Vivid examples of cloud providers are Amazon Web Services, Microsoft Azure, and the Google Cloud Platform [3]. Each of them has a lot of resources and services, with which one task can be performed even with a single provider using different services. In addition, every week new services appear or existing ones are improved, which makes it impossible to comprehend all interconnections and, as a result, obtain the optimal configuration of DIS using only limited human computing power [4].

Target setting. The problem is the lack of implementation or description of the implementation of the optimized distributed information systems.

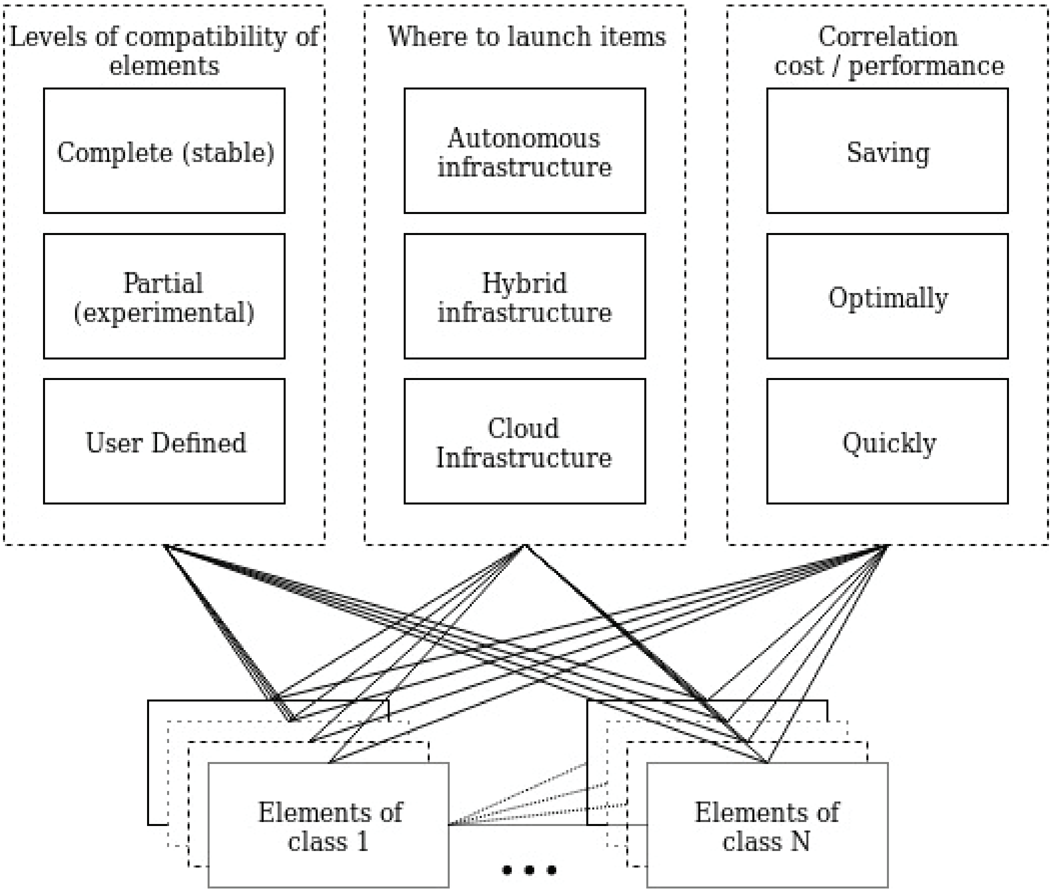
Actual scientific researches and issues analysis. Today, there is a necessary theoretical basis for optimizing DIS [5].

Uninvestigated parts of general matters defining. In this paper, we explore the method of implementing software tools to solve the RIS optimization problem.

The research objective. The task is to develop a concept for optimizing DIS configurations, which involves obtaining the description of applications and infrastructure in the form of a code [6], by which it is easy to deploy the optimal version of the distributed information system.

The statement of basic materials. The proposed concept of the implementation of software consists in finding in the informal description of the system provided by the user, the formal parts of the description of the elements, finding the coverage, which is best at the given time corresponds to the weight of the requirements [5] as shown in the picture 1. As a result of the system's operation, a descriptive system is described in HCL [7] and / or YAML [8], which are the language of the description in systems such as Terraform [9] and Docker-compose [10].

The structure of software looks like three linked modules, which are divided into functional: preprocessor, optimizer, postprocessor shownin the picture 2.



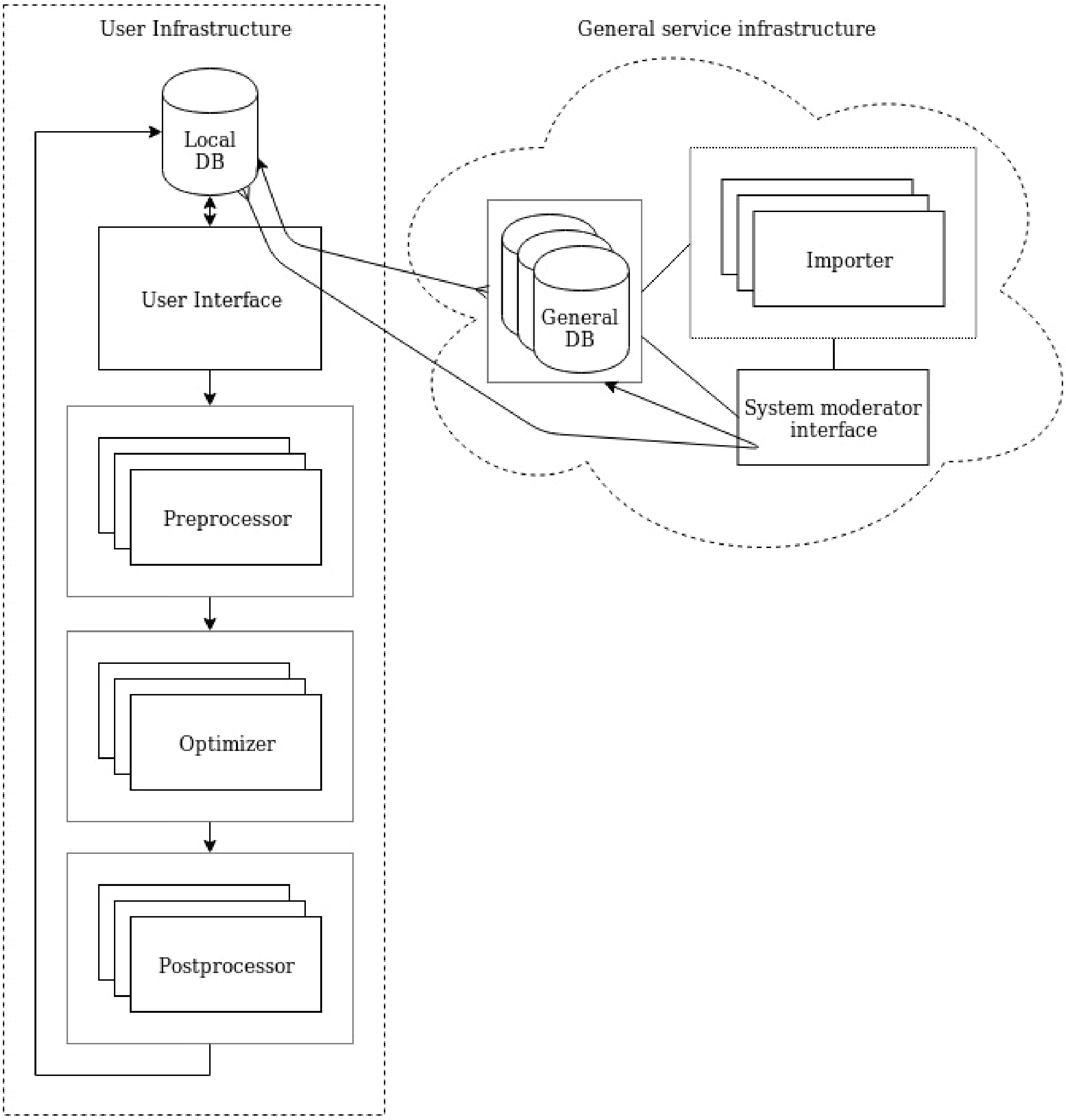
Picture 1. Block of tags

The preprocessor is responsible for data preparation. To find the formal description in the informal, first of all, it is proposed to use usecases that already have the necessary links with the formal description of the elements. To do this we use the concept of tags, and to fill formal descriptions with appropriate tags using a preprocessor with a minimum human participation - Multi-label classification [11].

The optimizer is responsible for finding the coverage and solving the optimization problem. For this purpose, already known algorithms and methods, such as ABC-analysis [12] and Quine-McCluskey [13] BOOM [14], Espresso and others, can be used. They will act as part of the formal description of the elements, looking for all possible matching combinations of elements.

The postprocessor selects the optimal parameters of each of the configurations based on the weight of each of the filters that were specified by the user and compares the results obtained. On the output, if possible, get several similar architectural projects, the user will be given the opportunity to compare the characteristics of each option and choose the one that more closely matches his goals.

At the time of choosing the desired configuration description, the user is given a code that describes its infrastructure dependencies, which allows the user to automatically deploy a system of any complexity.



Picture 2. System architecture

Conclusions. The proposed method of implementation of optimization DIS currently has no analogues and can dramatically increase the speed of development of prototype systems with the use of all the latest features and optimizations, which in turn allows the best practices to people who do not have the relevant knowledge, as well as adjust the decisions of professionals, pointing to shortcomings in their system.

References

1. Nik Simpson (Jun 2012). Comparing Data Center Costs With Public IaaS Cloud Services. URL: https://www.gartner.com/en/documents/2048419 (Access time: 28.04.2019)

2. Sanil Solanki, Michael Smith, Tomas Nielsen (Apr 2015). The Financial Case for Moving to the Cloud. URL: https://www.gartner.com/en/documents/3030826 (Access time: 28.04.2019)

3. Larry Dignan (Feb 2019). Top cloud providers 2019: AWS, Microsoft Azure, Google Cloud; IBM makes hybrid move; Salesforce dominates SaaS. URL: https://www.zdnet.com/article/top-cloud-providers-2019-aws-microsoft-azure-google-cloud-ibm-makes-hybrid-move-salesforce-dominates-saas/ (Access time: 29.04.2019)

4. Yuval Noah Harari (Sep 2014). Sapiens: A Brief History of Humankind, Chapter 10: The Scent of Money - https://www.ynharari.com/book/sapiens/ (Access time: 22.04.2019)

5. Власов М. Д., Болдак А. О. Оптимізація конфігурацій розподілених інформаційних систем URL: http://sait.kpi.ua/media/filer_public/6e/80/6e804b3f-ae13-4899-b336-4daddbd45584/sait2018ebook.pdf (Access time: 21.04.2019)

6. Kief Morris (Feb 2019). Introduction to Infrastructure Patterns URL: https://infrastructure-as-code.com/patterns (Access time: 27.04.2019)

7. Mitchell Hashimoto (Jul 2014). What is Hashicorp corporative language. URL: https://github.com/hashicorp/hcl/blob/master/README.md (Access time: 28.04.2019)

8. Ingy döt Net (Dec 2016). What is YAML. URL: https://yaml.org/ (Access time: 28.04.2019)

9. Mitchell Hashimoto (Jul 2014). What is Terraform. URL: https://github.com/hashicorp/terraform/blob/master/README.md (Access time: 28.04.2019)

10. Misty Linville (Sep 2016) Overview of Docker Compose. URL: https://docs.docker.com/compose/overview/ (Access time: 28.04.2019)

11. Kartik Nooney (Jun 2018). Deep dive into multi-label classification..! (With detailed Case Study). URL: https://towardsdatascience.com/journey-to-the-center-of-multi-label-classification-384c40229bff (Access time: 27.04.2019)

12. Ranganath Muttanna Singari (Feb 2014). Application of Selective Inventory Control Techniques for Cutting Tool Inventory Modeling and Inventory Reduction-A Case Study URL: https://www.researchgate.net/publication/273258286_Application_of_Selective_Inventory_Control_Techniques_for_Cutting_Tool_Inventory_Modeling_and_Inventory_Reduction-A_Case_Study (Access time: 27.04.2019)

13. Jiangbo Huang. Programing implementation of the Quine-McCluskey method for minimization of Boolean expression. URL: https://arxiv.org/ftp/arxiv/papers/1410/1410.1059.pdf (Access time: 27.04.2019)

14. PetrFiˇser, JanHlaviˇcka (Nov 2002). BOOM — A HEURISTIC BOOLEAN MINIMIZER. URL: http://www.cai.sk/ojs/index.php/cai/article/viewFile/450/356 (Access time: 25.04.2019)

Autors

Boldak Andrii – Candidate of Technical Sciences, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: boldak.andrey@gmail.com

Болдак Андрій Олександрович – кандидат технічних наук, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».

Vlasov Maksym – Student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: m.vlasov@post.com

Власов Максим Дмитрович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського».