Optimizing the Energy Consumption of On-site Private Cloud Computing Platforms

Anatol Lozinskyi[1]*, Anatoly Gladun[1]

[1] *Institute of information technology and systems of the National Academy of Sciences of Ukraine,
40 Akademik Glushkov Avenue, Kyiv, 03187, Ukraine*

Abstract
Cloud computing offers advantages such as self-service and dynamic redistribution of resource structures, necessitating the development of effective algorithms for optimal use of hardware infrastructure. Recent studies highlight the problems of resource redistribution in on-site private cloud computing platforms software layers, affecting total energy consumption. Existing approaches to mathematical modeling use combinatorial optimization, game theory, and artificial intelligence but lack an integrated approach to energy consumption optimization. The proposed approach involves creating an original algorithm based on a multi-level architecture developed by the author. This architecture divides objects into seven functional layers, from hardware physical infrastructure to application SaaS layer. The algorithm aims to minimize electricity consumption by isolating critical parameters in the process of redistributing resource structures. The optimization problem is segmented into subproblems, allowing for efficient load placement and energy savings. The model is focused on optimizing the power consumption of on-site private cloud computing platforms using a well-known approach – reducing the number of hardware nodes involved. The idea is to switch off not only individual computing nodes but also cooling nodes when idle. The mathematical formulation of the resource allocation in on-site private cloud computing platforms problem is presented as a Multi-dimensional Bin packing. The algorithm employs dynamic programming to maximize load placement density and reduce the number of hardware nodes. The model provides support technologies for both virtual machines and software containers. The paper presents the original author's algorithm, *Optimization of Programmable Infrastructure Resources* (OPIR), focused on reducing electricity consumption through combinatorial optimization.

Keywords
cloud computing, on-site private cloud computing platforms, energy optimization, multi-layer architecture of system, resource allocation

Introduction

A large-scale modern power supply challenge is the rapid increase in the workload of *data centers* (DC) with big data storage and analytics tasks, which leads to the growth of hardware infrastructures of cloud computing platforms. At the present stage of development, the number of cloud computing platforms hardware nodes already reach tens of thousands [1][2]. Several advantages in cloud computing, such as self-service, dynamic redistribution of resource structures in software layers require the development of effective algorithms and methods for optimal use of hardware infrastructure.

1. Related Surveys

A detailed study [3] provides an overview of numerous developments in the field of energy efficiency along the last decade in different fields of computing technologies from chips micro-architecture to data centers scale. The issue of energy efficiency in the future stage of the

Workshop "Intelligent information technologies" UkrProg-IIT`2025 co-located with 15th International Scientific and Practical Programming Conference UkrPROG'2025, May 13-14, 2025, Kyiv, Ukraine
* Corresponding author.
lozinskyy@gmail.com (A. Lozinskyi); glanat@yahoo.com (A. Gladun)
0000-0002-3101-5809 (A. Lozinskyi); 0000-0002-4133-8169 (A. Gladun)

development of computer technology is brought to the fore. The *Energy wall* becomes the barrier around which computing systems will develop in the entire spectrum of implementations. Modern research focuses on finding solutions to the energy barrier problem through various ways of implementing close interaction between energy-efficient hardware and energy-aware software technologies.

Numerous studies and scientific publications highlight the problems of redistributing the resource structures in the program layers of cloud computing platforms from the viewpoint of various aspects in cloud computing organization. Indeed, the components of the sub-system affect the overall power consumption of cloud computing platforms. Highlighting the complexity of implementing the processes of redistribution of resources in individual subsystems, these works provide an understanding of the existing possibilities for optimizing the power consumption of cloud computing platforms in general. The paper [4] highlights the problem of sharing processor resources. The competition of several consumers for the speed of computing is caused by the placement of workloads of several consumers on one computing node. An algorithm for planning the hot migration of running virtual machines between nodes of the hardware cluster is proposed in order to avoid conflicts. The paper [5] concludes that the technology of hot migration of virtual machines can be used for another purpose – to concentrate virtual machines on a smaller number of computing nodes in the cluster. The work [6] highlights the problem of resource utilization from the point of view of providing reliable access for consumers to the storage services of the Amazon cloud platform. The paper presents the capabilities of the distributed Dynamo system in providing highly reliable data storage and availability due to built-in algorithms for distributed duplication of network access and storage subsystems. The analysis of the work suggests that reducing the amount of physical storage equipment can substantially reduce the cloud platform's power consumption. The paper [7] presents the issue of organizing data transfer in DC with computing clusters. Based on the data presented in the work, it can be concluded that the application of the principles of managing the number of network devices involved (switches, routers) to the equipment of the DC network infrastructure can substantially reduce the cloud platform's power consumption. The paper [8] presents a study of hot migration of virtual machines between blade servers as part of the IBM BladeCenter chassis to control the load and temperature conditions. Therefore, we can conclude that the consolidation of virtual machines also affects the capacity of the cooling systems necessary to remove the heat allocated by the computing nodes. The paper [9] presents a subsystem for automating the scaling of the software infrastructure of cloud computing platforms – the Policy Keeper orchestration service. Hot migration of virtual machines and software containers is done based on machine learning scripts, policies, and algorithms. Therefore, the complex and high complexity of scaling processes of cloud computing software infrastructures can be successfully performed by orchestration tools. The work [10] relates the load level of physic computing nodes to the power consumption of heat dissipation equipment. Migration of virtual machines in operation is considered as a mechanism for managing the load of blade servers. It can be concluded that control over the operation of heat removal equipment gives a significant optimization of energy consumption. The work [11] presents a classification of cloud deployment scenarios by form of ownership and location. A cloud deployment scenario that belongs to one enterprise and serves only users of this enterprise, as well as located on the enterprise's own territory is classified as on-site private cloud. Further in the text, the term *on-site private cloud computing platform* (OPCCP or Platform) will be used to refer to cloud computing platforms deployed in this scenario. More complex scenarios for organizing cloud computing can be considered as the combination and interaction OPCCP of several enterprises with each other, including those scenarios where shared access to the platform's cloud services is provided. Therefore, the OPCCP study on energy consumption optimization remains relevant for other cloud computing scenarios.

The current state of the mathematical methods application issue in the field of cloud computing are covered in the work [12]. Existing approaches to mathematical modeling of individual cloud computing processes use the methods of combinatorial optimization, game theory, artificial intelligence, etc. However, the algorithms presented in the papers are focused on individual subsystems and do not provide for modeling the power consumption target function by simultaneously isolating essential factors of all subsystems of OPCCP. Also, the presented algorithms are mainly focused on the consolidation of virtual machines or software containers, and do not delve into the details of the resources and processes involved in the implementation of the OPCCP architecture.

2. The proposed approach

According to the author, only an understanding of the holistic picture of the organization of the work of OPCCP from the physical infrastructure of the data center to the highest level of SaaS cloud services allows for a significant improvement in the algorithms for minimizing power supply. The creation of an original algorithm for optimizing power consumption is possible by highlighting critical parameters as a result of the analysis of internal system processes, which is based on the developed by the author *multi-level architecture of cloud computing platforms* (Architecture). Understanding components of the Platform and the principles by which they interact allows the researcher to identify significant parameters for optimizing energy consumption. A detailed explanation of the organization, purpose, properties and constituent elements of each layer of the Architecture is presented in the paper [13].

2.1. The redistribution process of structure of resources

The basis for presenting a holistic picture of the structure and internal processes of the OPCCP is the author's Architecture. The method of splitting into layers used in the Architecture divides functionally similar components into seven distinct layers. The lowest layer of the Architecture is the physical infrastructure hardware – that is, a set of different *physical nodes* (PN): computing nodes, data storage systems, network equipment, uninterruptible power supply, cooling units. Therefore, all these elements of the Architecture consume electricity, heat up and require cooling, which also adds its share to the total energy consumption of the Platform. The rest of the Architecture consists of software layers (operating systems, software platforms and products). Deployment and launch of elements in software layers leads to the involvement of hardware layer resources (processor cores, RAM, data storage space, network bandwidth). In fact, the process of redistributing OPCCP resource structures is determined by the life cycles of creating, operating and deleting the constituent elements of software layers. The limited number of resources available to the deployed operating system of each individual computing sever is determined by its constituent physical units (processor cores, RAM, data storage drives). Figure 1 shows a revised representation of the Architecture considering the new Kubernetes technology [14], which allows the implementation of flexible and elastic container-based software infrastructures.
It is to represent resources as an abstract array (the main property of cloud computing) that OPCCP is built based on a *computing cluster* (Cluster). In the Cluster, PNs are added and removed transparently (without stopping customer service), thanks to the flexible capabilities of the Architecture to redistribute the resource structures of software layers between PNs. Thus, a separate *cloud service* (CS) is implemented as a distributed information system capable of increasing and decreasing resources volume needed in response to service requests to perform the ordered actions.

Analyzing the possibility of increasing the resources of the Architecture, we assume that there are no restrictions on the number of PNs in the OPCCP. Accordingly, the value of the power consumption of the OPCCP is equal to the total amount of power consumption of all the involved PNs. Obviously, the total resource volume is limited by the total amount of PN resources in the OPCCP. We will also consider the standard practice in the IT industry to produce specialized PNs focused on a separate type of function or resources (computing speed, data storage, network interaction, uninterruptible power supply, heat dissipation). For abbreviated designation in the figures and in the text, we classify the types of PN according to their resource orientation:

computing (CPN), data storage (SPN), network (NPN), uninterruptible power supply (PPN), heat dissipation (DPN).

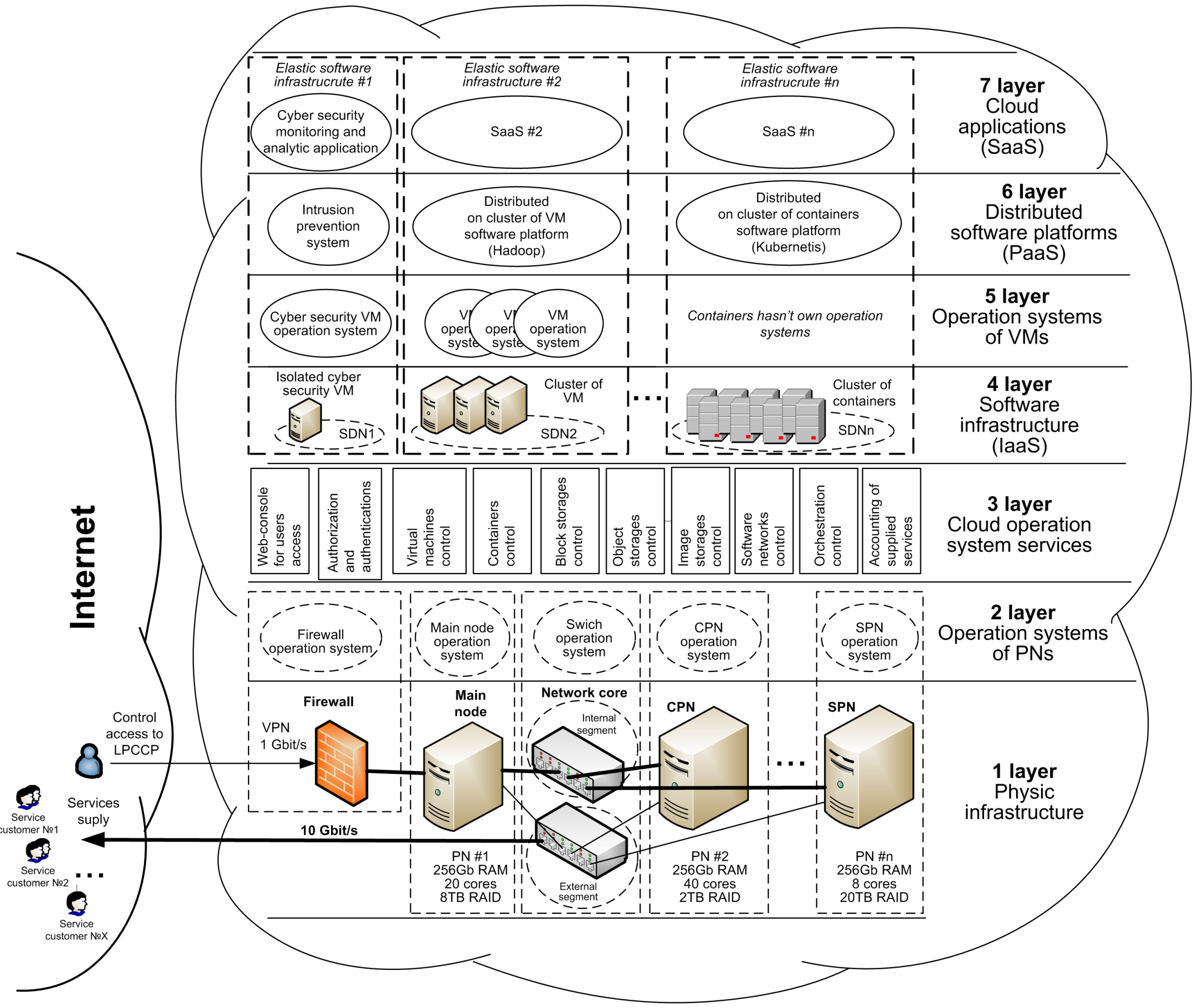

Figure 1: Multi-layer architecture of on-site private cloud computing platforms.

Also, the spatial location of equipment in the data center servers room affects the power consumption costs of the physical infrastructure. Let's simulate the situation with standardized designs of PN enclosures with their placement in typical mounting cabinets in a stack one above the other, allowing for a flexible combination of options for their spatial placement. The cabinets in the machine room are placed in parallel rows to create "cold" and "hot" convection cooling corridors. A solution is possible when the data center engine room is equipped with DPNs of different heat dissipation capacities: a machine room, a cold corridor of several cabinets, a separate cabinet. Thus, the heat dissipation zone is the totality of PN, the cooling of which is provided by one DPN. The number of cabinets, PNs and their types varies according to the needs of the supplier regarding the operation of the OPCCP.

Given the local nature of the network architecture of OPCCP, it is advisable to assume that the network latency is constant and very small, such that its value can be neglected. Therefore, the above-mentioned structure of the network architecture and the principle of equipment location in the data center makes it possible not to analyze the impact of network activity on electricity consumption. From the author's point of view, the network hardware architecture of OPCCP should be built based on flagship models of third-level switches, which allow you to vary the number and bandwidth of ports, programmatically control the configuration of virtual network segments.

The general structure and dynamics of redistribution of OPCCP resources in the implementation of individual CS depends on the ratio of the constituent software elements. Within the framework of the model, four types of elements of the program layers of the Architecture involved in the process of redistribution of resource structures are considered: 1) the *operating system* of the PN (OS); 2) *operating system service* (OSS); 3) *virtual machine* (VM); 4) *software container* (SK). The implementation of cloud services is carried out by the composition of the OSS, as well as the composition of the compositions of the OSS. Further in the text, where appropriate, the elements of OS, OSS, VM and SK will be designated by the general term of *Load.*

The author's idea for solving the problem of optimizing the energy consumption of OPCCP is to segment the total population of PN by two types of zones in the machine room. The first type is a subset of PN whose power supply is provided by one PPN forming *power supply zones* (PSZ). The second type is a subset of PN whose heat dissipation is provided by one DPN forming *cooling zone* (CZ). The idea is, in the absence of loads, to power off not only individual CPN, but also PPN and DPN, since the power consumption capacity of DPN is commensurate with the power consumption of PN inside the CZ. One CZ can cover several PSZs, depending on the implementation of the data center. Minimization of electricity consumption is achieved by minimizing the number of involved both individual PNs and entire PSZ and CZ. Preference is given to the placement of loads in the active zones. Only in the absence of resources in the zones involved, an additional zone is activated, since the DPN has the highest level of power consumption.

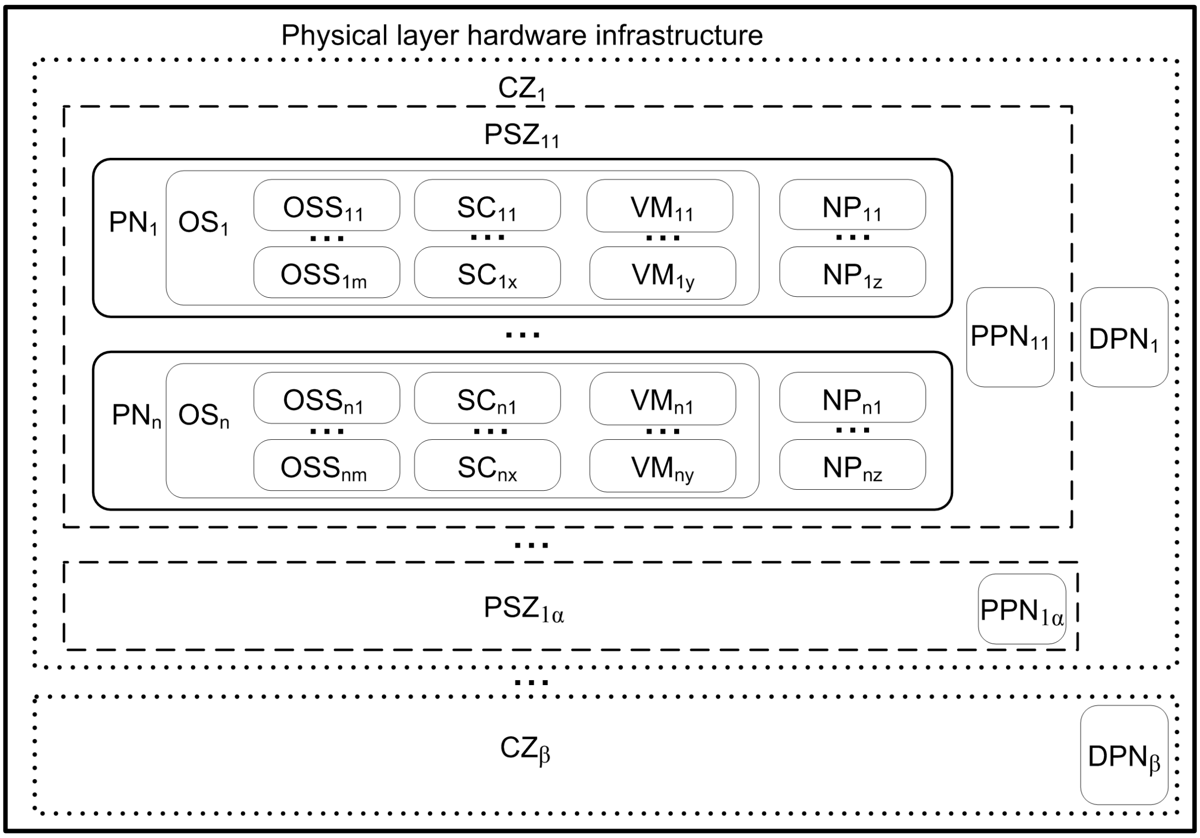

Figure 2: Structure of resources redistribution in OPCCP

Splitting the common set of CPN into subsets of the CZ, which in turn are also divided into subsets of the PSZ, allows the division of the optimization problem into subproblems with a limited size of data. Let's represent the individual elements of the resource structure as packages packed one into one. Thus, we get the idea of an algorithm for packing the multilevel nesting of elements of the program levels into the corresponding elements of the lower levels of the Architecture. PSZ are packed into CZ. At a level higher CPNs are packed in to PSZ. Even a level higher the *loads* are packed in to CPNs. We consider that structural elements are heterogeneous in terms of redistribution technology (VM or SC) and the composition of resource types (computing instances, data storage volume). A general representation of the idea of redistribution of resource structures is shown by Fig. 2.

Also, in order to achieve energy savings, the algorithm for redistributing OPCCP resource structures should perform the following subtasks: 1) selection of the target CZ; 2) selection of the target PSZ; 3) selection of target CPN; 4) selection of loads according to the resource orientation of the target CPN or SPN; 5) packing loads in the selected CPN or SPN; 6) shutdown of unused CPNs, if necessary shutdown entirely all infrastructure PNs in unused PSZs and CZs; 7) elimination of loads placement dispersion across machine room of DC.

We suppose that the type of microprocessors in PN is the same. The computing power of the elements of the Architecture is measured by the number of microprocessor cores involved, the unit of measurement is piece by piece (pcs). We suppose that the type of RAM in CPN is the same. The unit of measurement of the amount of RAM is a byte (Byte). The unit of measurement of the amount of data storage is a byte (Byte). Finally, the result of the target function of the optimization model is the power consumption capacity of OPCCP. The unit of measurement of power consumption is Watt (W). It should be noted that typical operating systems of minimal configuration are deployed in CPN and VM as needed for OSS deployment environment.

2.2. Resource allocation mechanism

The problem of optimizing the redistribution of OPCCP resource structures is caused by the need to ensure effective support of the main property of CS – the elasticity of cloud computing. The elasticity of cloud computing is a property that provides for the ability to change the amount of OPCCP resources involved in individual CSs over time. In accordance with the dynamics of changes in individual consumer needs, the total volume and structure of redistribution of OPCCP resources is constantly changing. An increase or decrease in the number of service requests to an individual CS in the SaaS application layer leads to a downward sequence of requests between the layers of the OPCCP architecture, causing the automatic allocate or release of CS compute resources in the IaaS layer in order to ensure that the volume of resources involved in the infrastructure is matched to ensure the current level and intensity of customer service.

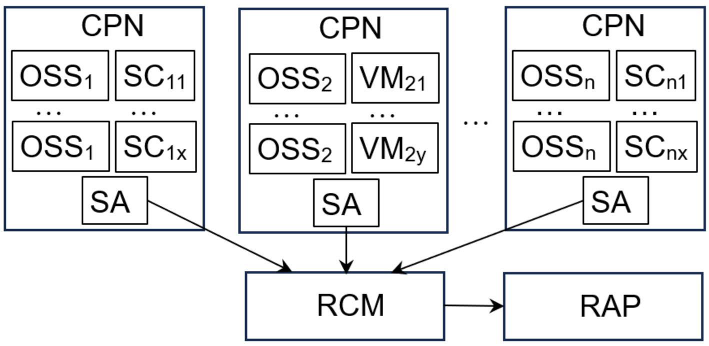

Figure 3: Interaction scheme of PA, RCM and RAP

The life cycle of OPCCP infrastructure facilities takes place in the IaaS (Infrastructure as a Service) layer. More precisely, requests for load placement are performed by the OSS subsystem *Resource allocation planning* (RAP). In choosing the target PN for loads placement, the RAP uses an algorithm of sequential enumeration, which leads to uncontrolled dispersion of loads between the CPN in the machine room.

The mechanism for the optimal selection of CPN can be implemented in the form of a separate *Resource Control and Management* (RCM) service. Fig. 3 shows the receipt by the RCM of information about the current state of redistribution of OPCCP resources. To provide the RCM with operational information about the current state of resource allocation at each CPN, a *software* *agent* (SA) is placed. Each SA notifies the RCM of each successful resource deployment or release event at the local CPN.

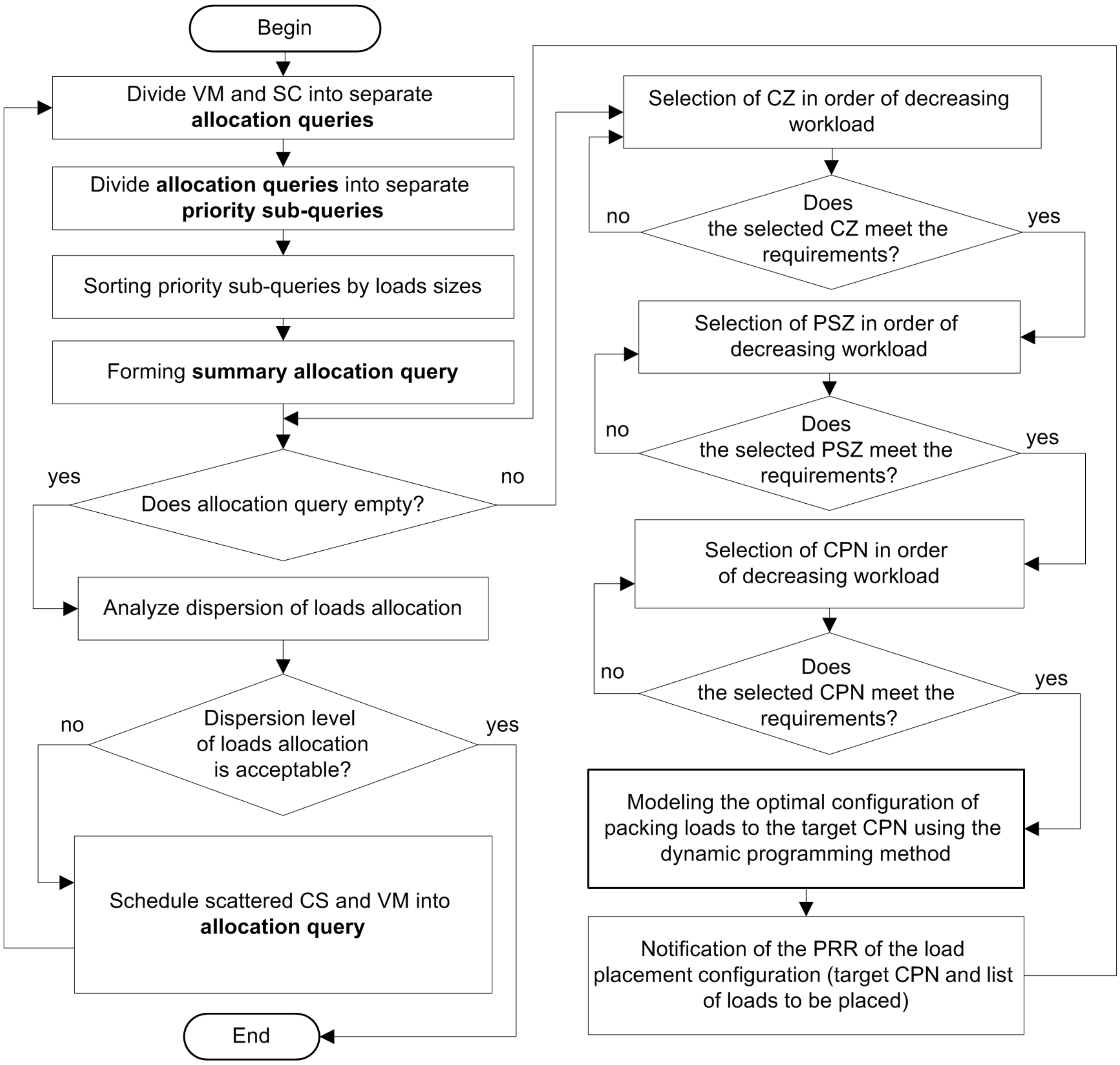

Figure 4: Block diagram of the OPIR algorithm

The calculation of the optimal solution for choosing the target CPN or SPN for *loads* placement should be performed by the RCM based on information about the current state of each PN in the hardware cluster of the Platform. It needs to know the power supply state of each CPN, and SPN, to which PSZ and CZ belong, what resources are currently placed on them and what number of resources for placing new *loads* are still available. The data of each message from the SA is entered by the RCM into the general *Nodes states table* (NST). Thus, the dynamics of changes in the NST reflect the current state of the resource allocation structure of OPCCP in real time. The data collected in the NST is processed by the algorithm of optimal allocation of resources integrated in the RCM. To ensure the optimal allocation of resources, the author proposes a sequence of actions that together make up the algorithm for *Optimizing Programmable Infrastructure Resources* (OPIR) represented by a flowchart in Fig. 4.

2.3. Algorithm for resource allocation optimization

The OPIR algorithm is activated by the occurrence of events of redistribution of OPCCP resource structures. These are two events: 1) placement *loads* to the allocation queue; 2) the removal (shutdown) of the *load* had happened.

We pack loads into the target CPN using the appropriate mathematical method to achieve maximum load placement density. Load placement processing is provided in the order of the queue. The queue is formed on a first-come, first-served basis. Loads can be assigned priority values. Priority loads are processed first.

Thus, the OPIR algorithm minimizes the number of not only involved CPNs by tightly packing the load queue to them but also ensures the consolidation (concentration) of *loads* in the machine room space, by reducing the number of active hardware nodes and turning off PSZ and CZ.

3. Mathematical methods overview

Architecture presents the OPCCP as an integral system composed of sets of elements with their inherent properties and parameters. Elements of the Architecture (should be understood as separate services). Connections between individual services implement other services of a higher level of complexity – the composition of services.

3.1. The packaging problem

The solution to the problem of redistribution of OPCCP resources is presented by the author as a problem of Multi-dimensional Bin packing. At present, the NP-completeness of a large number of problems, including the problem of packing a bin, has been determined. The bin packing problem is NP-complex [15], i.e., it does not have a polynomial solution. The increase in the number of input data leads to an exponential increase in the execution time of the algorithm. However, the packaging problem is very common in practical activities. Heuristic algorithms are usually used to solve problems of this class. There are two approaches to circumventing exponential dependence. Firstly, it is possible to limit the number of input data to reduce the time of calculations. Secondly, they try to find a polynomial algorithm for finding a non-optimal solution, but close to optimal.

Approximate methods for the bin packing problem include Greedy Algorithm, Genetic Algorithms, Ant colony Algorithm, Simulated Annealing, etc. These algorithms, as a rule, are characterized by polynomial complexity, and the fee for solving is an approximate result.

The exact methods of solving the bin packing problem include three main methods of discrete programming:

- *dual simplex* method is a numerical method for solving linear programs, which is an improved version of the simplex method. The introduction of an additional constraint allows you to obtain an optimal integer solution. An example of the implementation of this idea is the Gomori algorithm. The main disadvantage of the method is poor convergence to the whole solution.
- *branches and boundaries* method come down to building a tree of possible options, determining the estimate of the solution boundary for each vertex of the tree, cutting off unpromising vertices.
- *dynamic programming* method is based on the principle of Bellman's optimality. This method was developed in the method of sequential analysis of options.

The analysis of the resource structures redistribution processes shows that among the possible options for redistribution, it is necessary to find the configuration that provides the optimal value of the target function. Therefore, the problem is enumerative and can be reduced to combinatorial optimization problems.

3.2. Selection of mathematical method

The listed methods for solving the problem about the bin packing in the worst case (with any initial data, including the most unfavorable) have exponential complexity. This fact determines the class of complexity of the problem about the bin packing. The first two exact methods are discarded in view of the above shortcomings, and we will apply the method of dynamic programming in solving the problem [16].

4. Scaling the OPIR algorithm

The OPIR algorithm implementation is in the form of a VM image file in the format of common standard for cloud platforms *Open Virtualization Format* (OVF). This solution allows us to incorporate cloud capabilities for scaling the resources involved by the OPIR algorithm. Depending on the scale of hardware clusters with different orders of magnitude of the number of PNs, part of the resources at the main node of the OPCCP can be allocated for the placement of VM with OPIR, as well as the resources of a separate CPN. With the volumes of OPCCP clusters, where the number of PNs reaches tens of thousands of units, the algorithm can be implemented based on a VM software cluster with the allocation of resources in the specified PNs of cluster.

Summary and Further Work

The article presents for the first time the original author's algorithm *Optimization of programmable infrastructure resources* (OPIR) with the target function of reducing electricity consumption. The target function of the minimum electricity consumption of private on-site cloud computing platform operates based on resource redistribution options analysis in order to determine the optimal placement configuration both of virtual machines, software containers and operation system level services inside computing cluster of data center. The mathematical formulation of the resource structures redistribution problem in on-site private cloud computing platforms is presented by the author as a problem of Multi-dimensional Bin packing.

For future work, the author plans to investigate the application mathematic method of dynamic programming for solving the problem of packing virtual machines and software containers to target computing nodes.

Declaration on Generative AI

During the preparation of this work, the author used *Microsoft Office 365* built-in service *Copilot* to: Grammar and spelling check. After using this tool, the author reviewed and edited the content as needed and took full responsibility for the publication’s content.

References

[1] I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski, P. Suter, Serverless computing: current trends and open problems, in: S. Chaudhary, G. Somani, R. Buyya (Eds.), Research advances in cloud computing, volume 474, Springer, Singapor, 2017, pp. 1–20. doi:10.1007/978-981-10-5026-8_1.

[2] M. Tirmazi, A. Barker, N. Deng, M. E. Haque, Z. G. Qin, S. Hand, M. Harchol-Balter, J. Wilkes, Borg: the next generation, in: Proceedings of the Fifteenth European Conference on Computer Systems, EuroSys ’20, Association for Computing Machinery, New York, NY, 2020, pp. 1–14. doi:10.1145/3342195.3387517.

[3] R. Muralidhar, R. Borovica-Gajic, R. Buyya, Energy efficient computing systems: Architectures, abstractions and modeling to techniques and standards, CoRR abs/2007.09976 (2022). doi:10.48550/arXiv.2007.09976. arXiv:2007.09976.

[4] J. Ahn, C. Kim, J. Han, Y.-R. Choi, J. Huh, Dynamic virtual machine scheduling in clouds for architectural shared resources, in: Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing, HotCloud’12, USENIX Association, Berkeley, CA, 2012, pp. 75–80. doi:10.5555/2342763.2342782.

[5] A. Beloglazov, R. Buyya, Energy efficient resource management in virtualized cloud data centers, in: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster,

Cloud and Grid Computing, CCGRID '10, IEEE Computer Society, Washington, DC, 2010, pp. 826–831. doi:10.1109/CCGRID.2010.46.
[6] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, W. Vogels, Dynamo: amazon's highly available key-value store, in: ACM SIGOPS Operating Systems Review, Association for Computing Machinery, New York, NY, 2007, pp. 205–220. doi:10.1145/1323293.1294281.
[7] D. Abts, B. Felderman, A guided tour of data-center networking, Communications of the ACM 55 (2012) 44–51. doi:10.1145/351827.384253.
[8] J. Xu, J. Fortes, A multi-objective approach to virtual machine management in datacenters, in: Proceedings of the 8th ACM international conference on Autonomic computing, ICAC'11, Association for Computing Machinery, New York, NY, 2011, pp. 225–234. doi:10.1145/1998582.1998636.9
[9] J. Kovács, Supporting programmable autoscaling rules for containers and virtual machines on clouds, Journal of Grid Computing (2019) 813–829. doi:10.1007/s10723-019-09488-w.
[10] Z. Wang, N. Tolia, C. Bash, Opportunities and challenges to unify workload, power, and cooling management in data centers, in: Proceedings of the Fifth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks, FeBiD'10, Association for Computing Machinery, New York, NY, 2010, p. 1–6. doi:10.1145/1791204.1791205.
[11] M. Badger, T. Grance, R. Patt-Corner, J. Voas, Cloud computing synopsis and recommendations, 2012. doi:10.6028/NIST.SP.800-146.
[12] B. H. Malik, M. Amir, B. Mazhar, S. Ali, R. Jalil, J. Khalid, Comparison of task scheduling algorithms in cloud environment, International Journal of Advanced Computer Science and Applications 9 (2018) 33–42. doi:10.14569/IJACSA.2018.090550.
[13] A. P. Lozinskyi, Synthesis of cloud computing platform technologies, Control Systems and Computers (2019) 35–45. doi:10.15407/csc.2019.06.035.
[14] Kubernetes, Open source system for automating deployment, scaling, and management of containerized applications, 2025. URL: https://kubernetes.io/.
[15] J. Leeuwen (Ed.), Algorithms and complexity, volume A of Handbook of Theoretical Computer Science, 1st ed., Elsevier Science Publisher B.V., Amsterdam, The Netherlands, 1990.
[16] C. Papadimitriou, K. Steiglitz, Combinatorial optimization, Algorithms and complexity, Prentice-Hall, Englewood Cliffs, London, 1982.