UDC 004.853:007.2

Ivan BURLACHENKO, Volodymer SAVINOV, Iryna ZHURAVSKA
Petro Mohyla Black Sea National University

THE ORGANIZING OF COMPETITIVE EVENTS USING MULTI-AGENT TECHNOLOGIES AND THE MODIFIED BORDA METHOD

The hackathons allow collecting at once on one site: the largest industrial companies of the country, technology vendors from the rapidly changing environment in the markets, young developers (including students), engineers with experience in the IT-sphere or specifically required technologies.

The current state of hackathon organizing stages has analyzed to improve the approach to increase the social inclusion of participants. Statistical metrics of vacancies occurrence probability during the period after the hackathon and employee turnover provided by hackathons' sponsors according to business domains were investigated.

The methods of determining the winner in different systems of competitive selection are considered. Particular attention is paid to the peculiarities of the tournament systems used in cybersport championships. The system of selection based on the modified Borda method, consisting of two or a maximum of three rounds and independent of the number of participants, is proposed.

In the paper, the Multi-Agent Sell Funnel Monitoring (MASFM) algorithm has described. MASFM algorithm allows searching sponsorship efficiently because it helps detect about 16–23% of new sponsors according to last 2 years statistics.

In the software architecture of the online hackathons' platform, a real scenario of increasing performance 15 times from 6 to 94 requests/sec was applied, which does not require serious refactoring and complex code changes. Besides, the steps mentioned above can reduce the cost of infrastructure like Heroku. The next functionality of the online hackathon platform will be possible thanks to the microservices architecture.

As the result, efficient software architecture has implemented and allow to decrease the maximum response time down to 3 seconds and the online hackathon platform's performance has increased from 71 to 94 requests per second.

Keywords: eSport event, organizing of hackathon, selection of teams, multi-agent system, algorithm for determining winners, grading procedure

Іван БУРЛАЧЕНКО, Володимир САВІНОВ, Ірина ЖУРАВСЬКА
Чорноморський національний університет імені Петра Могили

ОРГАНІЗАЦІЯ ЗМАГАНЬ З ВИКОРИСТАННЯМ МУЛЬТИАГЕНТНИХ ТЕХНОЛОГІЙ ТА МОДИФІКОВАНОГО МЕТОДУ БОРДА

Хакатони дозволяють зібрати на одному майданчику одночасно: найбільші промислові компанії країни, вендорів технологій зі стрімко мінливого середовища на ринках, молодих розробників (у тому числі студентів), інженерів з досвідом роботи в IT-сфері або у конкретно затребуваних технологіях.

Проаналізовано поточний стан етапів організації хакатону для вдосконалення підходу до підвищення соціальної інтеграції учасників. Досліджено статистичні показники ймовірності появи вакансій протягом періоду після хакатону та плинності кадрів, надані спонсорами хакатонів за сферами діяльності.

Розглянуто методи визначення переможця в різних системах конкурсного відбору. Особливу увагу приділено особливостям турнірних систем, які використовуються на чемпіонатах з кіберспорту. Детально розглянуто особливості застосування олімпійської системи організації IT-спортивних заходів, зважаючи на те, що кіберспорт неухильно наближається до того, щоб стати включеним у програму Олімпіади 2024. Пропонується система відбору переможців на основі модифікованого методу Борда, яка складається з двох або максимум трьох турів і не залежить від кількості учасників.

У статті описано алгоритм моніторингу воронки продажів із кількома агентами (MASFM). Алгоритм MASFM дозволяє ефективно шукати спонсорство, оскільки він допомагає виявити близько 16–23 % нових спонсорів за статистикою за останні 2 роки. У програмній архітектурі платформи онлайн-хакатонів реалізовано реальний сценарій збільшення продуктивності в 15 разів, що не потребує серйозного рефакторингу та складних змін коду. Крім того, застосовані послідовні кроки можуть знизити вартість такої загальновживаної в онлайн-хакатонах інфраструктури, як Heroku.

В результаті реалізована ефективна архітектура програмного забезпечення, що дозволяє зменшити максимальний час відповіді до 3 секунд, а продуктивність платформи онлайн-хакатону збільшити з 71 до 94 запитів на секунду. Подальше покращення функціональності платформи онлайн-хакатону можливо завдяки імплементації в розглянуту архітектуру мікросервісів.

Ключові слова: кіберспортивна подія, організація хакатону, відбір команд, мультиагентна система, алгоритм визначення переможців, порядок оцінювання

Introduction

Organizing an e-sports event is a complex process. Their influence on the various activities becomes possible due to the gamification of any processes, including in education. Gamification is the use of game practices and mechanisms in a non-game context during the learning process to engage users in problem-solving. The elements of the gamified process include joint actions to achieve their own goals, virtuality, countdown during the task for a limited time are elements of the gamified process, etc.

Hackathon is a way to find technological solutions for e-sports. The goal of the hackathon is to bring together students, developers, designers, data researchers, scientists, artists, 3D modelers, composers, managers with a variety of skills to develop joint projects and address specific challenges.

The hackathon allows you to collect four parties at once on one site: the largest industrial companies of the country, technology vendors from changing markets, young developers (including students), engineers with experience in the IT-sphere, or the specific required technologies. The point is that large companies come with their own tasks, and developers (in our case, students) try to show the concept of their solution at such hackathons. In a successful case, the participants in the hackathons receive contracts on the basis of which a company can be founded. Customers spend two or three days of their time answering questions, but they get a very good picture of technologies and many solution prototypes at once. It is for this reason that the hackathon, as an e-sports event, is an important stage in the educational process. The hackathon allows students to immerse themselves in the corporate culture of industrial companies for a certain period.

The Current State of Hackathon Organizing

There are different types of hackathons according to the type of participants (external in which anyone interested in the topic can take part and internal which are organized for a closed community of a particular company or organization), to the holding format (offline and online), etc. [1]. Offline, all participants gather in one place with round-the-clock access and spend the entire hackathon there. The organizers usually provide them with everything need (food, office, convenient places) so that the teams work on the task without being distracted.

The duration of the hackathon starts from 24 hours, the question arises how to sleep. And no way. At hackathons it is really customary to teach 100% and devote all your time to work on the project, so not everyone can sleep and not always.

During online hackathons, all processes from team building to pitching take place online and do not require the physical presence of participants. All interaction takes place either through special platforms for the hackathons or through separate online tools.

The advantages are that there can be many more participants in such online events than full-time participation, even from anywhere. It should be noted that the corona crisis has significantly changed the nature of the event market, displacing offline activity and significantly increasing the share of online activities, including in the world of hackathons [2]. The disadvantages are the difficulty of staying asleep when you sit at home in bed, much harder than when you work as a team on location.

The topics of hackathons are completely different. If at first, they were held only for IT specialists who gathered together for a group programming session, today hackathons take place in almost all professional fields and solve a variety of tasks. Hackathons conducted by Corporate Social Responsibility (CSR Ukraine) & UNFPA Ukraine (held by the United Nations) under the STEM Girls grant initiative are aimed at changing gender stereotypes in the IT industry [3]. The task of hackathons held under the program "ULEAD with Europe" financing by The European Union is to create information projects or specific programs with the help of information technology, which will become a starting platform for solving community problems [4].

The Entire Process of Organizing a Hackathon
Search for Participants and Advertising Campaigns Based on Multi-Agent Technologies

To find the initial set of sponsors the advertisement campaign needs to be started. Messenger channels and social networks are most suitable for audience coverage involved in the hackathon's workflow. For evaluating the effectiveness of the promoted project, you can use matrix algebra operations [5], fuzzy logic approach [6], evidence theory [7], or multi-agent technologies [8, 9]. During the participation, the metrics of every sponsor will be analyzed. Hackathon's statistical metrics defined by equation (1).

$$F^{MASFM} = \begin{cases} \frac{V(P(T))}{WF_C(BD)} > FR_{avg}; \\ \frac{H^M(S_T, V_S)}{P(S_T)} > A_C . \end{cases} \tag{1}$$

In formula (1) F^{MASFM} is sponsor's fitness function of the agent in Multi-Agent Sell Funnel Monitoring (MASFM). Sponsors are filtered according to the probability of vacancies $V(P(T))$ during the period after the hackathon. Employee turnover $WF_C(B)$ can be provided by sponsors according to business domains BD of IT company to define the sponsors' vacancy fill rate that should be greater than the average value FR_{avg}. One of the key features investigated in F^{MASFM} is hackathon matching function $H^M(S_T, V_S)$ for the technology skills S_T and vacancies skills V_S. Participant's skills $P(S_T)$ are important also and define the relations that should be greater than the average value A_C between all potential sponsors. The F^{MASFM} values have logged to the online hackathons platform database to be analyzed by organizers. Engaging developers is key to a successful hackathon because they know how to build applications. Online hackathon's platform should also involve experts from the business domains, people from the communities, students, the wider the audience, the more creative solutions can be. The most expensive hackathon elements, they provide venue rent, food, and prizes. The algorithm for finding sponsors to support the hackathon and participants is presented in Algorithm 1. Also, engaging sponsors is influenced by what organizers can offer in exchange for support, so it is important to use MASFM to organize the hackathon.

ALGORITHM 1: Multi-Agent Sell Funnel Monitoring Algorithm

```
[participantsAgents, sponsorsAgents] = initializeMASFM(eventDate)
setupSponsorsFunnel(sponsorsAgents)
setupParticipantsFunnel(participantsAgents)
currentDate is inside monitoringPeriod
while currentDate is inside monitoringPeriod, do
        sponsorsAgents scan [probabilityOfVacancies, employeeTurnover] in monitoringPeriod
        for technologySkill in vacancySkills determined by sponsorsAgents, do
                for each participantSkill in list determined by participantsAgents do
                    calculate hackathon_matching_function_value
                    calculate sponsors_vacancy_fill_rate
                MASFMMetrics        =        aggregate(hackathon_matching_function_value        ,
sponsors_vacancy_fill_rate)
            end
        analyze(MASFMMetrics) and sell sponsorship deal
    end
```

Roles of Participants and Categories of Hackathons

To define the categories of hackathons in more detail, it is necessary to describe the roles of the participants. It will help increase the level of organization of the event process. Consider the main definitions used during the organization and conduct of the hackathon.

"The participant" is a specialist who has expressed a desire to participate in the Hackathon and received confirmation of participation from the organizers. "The team" consists of specialists who work together to create a project in Hackathon. "The team leader" is one of the team members who perform certain leadership functions: provides communication with mentors and organizers, represents his team at checkpoints and pitching.

"Themes" and "tracks" of the hackathon are specific areas within which projects will be developed. The tracks will bring together teams working on projects on the same topic. One track will correspond to one hackathon theme.

"Challenge" is a specific task presented by the organizers of the hackathon, in which you can also show your creativity. Teams can develop their unique project within one of the hackathon themes or work on a challenge.

"An expert mentor" is a mentor who has unique expertise, provides informational support and advice to hackathon teams in a particular field of knowledge. "A team mentor" is a mentor who moderates the work of teams. The team mentor will work with 3–5 defined teams.

"The judge" is an expert who will evaluate the projects developed by the participants. "The organizer" provides participants with everything necessary for productive work on the hackathon, monitors compliance with all rules and regulations (see Table 1).

The structure may differ slightly from hackathon to hackathon. The main and longest part of the hackathon is the teams' presentation of their ideas to the judges, who make decisions and award the winners.

Table 1

Detailed structure of the hackathon categories

Hackathon Type	Duration	Participants Roles	Sponsors Support	Judging
Long term	1–3 months	Judge, Participant, Mentor, Expert, Reviewer	Medium	Borda
Short-term	3–4 days	Judge, Participant, Mentor, Expert	Medium	Experts considered decisions
One-day	9–24 hours	Judge, Participant, Reviewer	Low	Experts considered decisions

The Software Architecture for Online Hackathons

The server side of the online hackathon platform is written by Spring Boot. Spring Boot is a project at the IO Execution level of the IO Spring Framework. With Spring Boot, the web application configurations are minimized as much as possible. Spring Boot supports embedded containers that allow web applications to run independently and without the need for a web server (see Fig. 1 below).

The web application of the hackathon platform shown in Figure 1 is working through the Java war command to export a war file to run on the Web Server. We used the "CLI Tools" to run spring scripts. Benefits of Spring Boot for the Hackathon Platform:

1) easily used to develop a Spring-based application with Java or Groovy Spring;
2) minimizes development time and raises productivity;
3) avoids writing a lot of boilerplate, Annotations, and XML configuration;
4) easily allows you to interact with Spring Boot applications with Spring Ecosystems like Spring JDBC, Spring ORM, Spring Data, Spring Security, etc.;

5) follows the "Default Configuration Principles" approach to minimize the time and effort invested in developing applications;

6) provides Embedded HTTP servers like Tomcat, Jetty ... to quickly and easily develop and test web applications;

7) provides CLI (Command Line Interface) tools for developing and testing Spring Boot (Java or Groovy) applications from the command prompt very easily and quickly;

8) provides many plugins for quickly developing and testing Spring Boot applications using Build tools like Maven and Gradle;

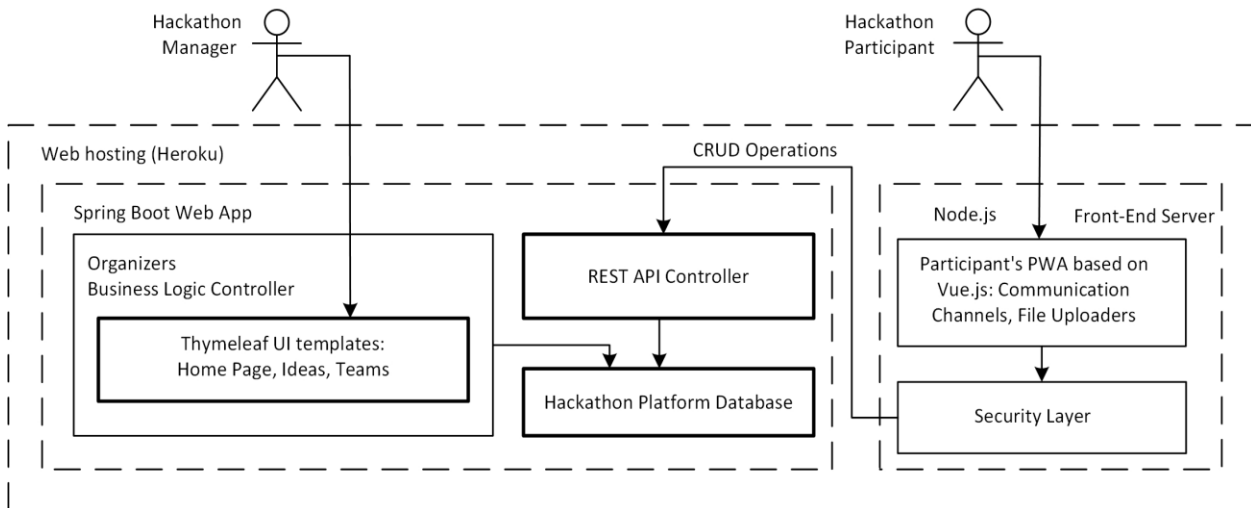9) offers many plugins for easy handling of embedded databases and in-memory Databases.

Fig. 1. Block diagram of the architectural solution for the hackathon online platform software

Heroku runs Spring Boot web applications inside one or more isolated "Dynos", which are virtual Unix containers that provide the necessary environment for your application. The dynos data is completely isolated and has an ephemeral file system (a "short-lived" file system is completely cleaned up and refreshed every time the dyno is restarted). Heroku internally uses a load balancer to distribute web traffic among all "web" dynos. Since the dynos are isolated, Heroku can scale the application horizontally by simply adding more dynos. The file system is ephemeral, so you cannot directly install the services your application needs (i.e., databases, queues, caching, storage, email services, etc.). Instead, Heroku provides services as independent "add-ons" either from Heroku and third parties. At the moment your application launches, dynos access services using the information contained in the configuration variables of your application.

To run your Heroku application, you need to be able to install the appropriate environment and dependencies. Developers interact with Heroku using a custom client application/terminal, which is very similar to a Unix bash script. It allows you to download code from a git repository, control running processes, view logs, and set configuration variables.

To get the hackathon platform application to work with Heroku, we needed to host our web application in the git repository, add the files listed above, connect the database add-on, and configure the settings to work correctly with static files. The Vue.js framework will allow you to create a responsive interface for both web and mobile platforms (Fig. 2).
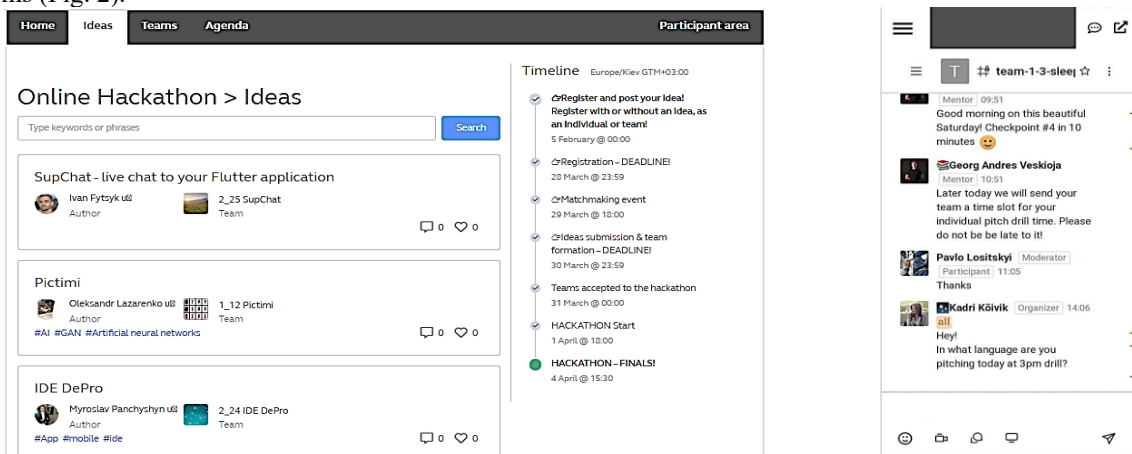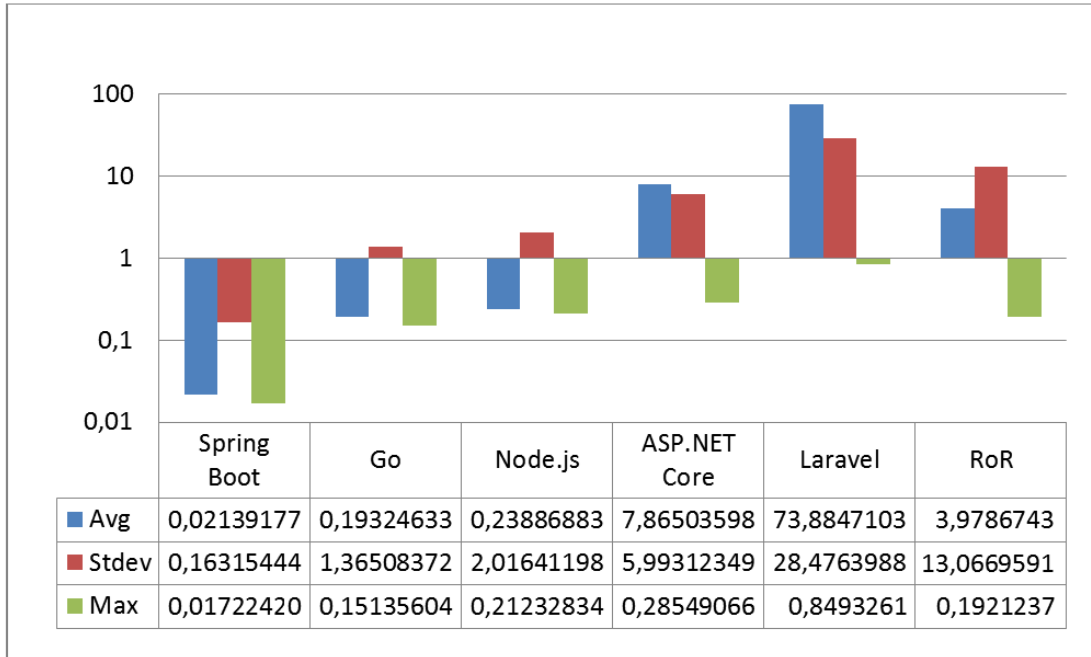

Fig. 2. The user interface of the online platform for hackathons

After completing all the necessary stages of the site of the hackathon platform, we can create a Heroku account, access the Heroku client, and use it to deploy a web platform for conducting hackathons.
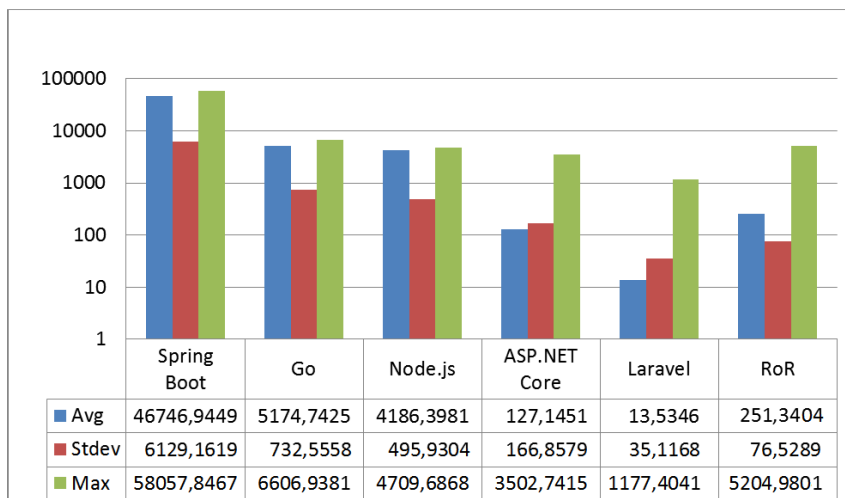
The research used the JMeter application, which is designed to test the performance of web servers and is used as an automated tool for testing with test data, as well as a tool for functional testing of web applications, file servers, web servers, and even databases. During the experiments, important characteristics of JMeter were investigated. The JMeter application can be configured to simulate the N number of users or streams that load a specific web server or web application. JMeter measures web server performance by creating a simulated load on a web application. Moreover, several repeatable cycles can be done to get the average result, as well as see the test results in graphical and statistical form.

Based on the results, we can conclude that it is better to use clean technologies for small web applications. Note that ASP.NET Core used a large number of built-in libraries and MVC pattern. For clarity, a base-10 log scale was chosen to analyze the results was shown in Fig. 3.

	Spring Boot	Go	Node.js	ASP.NET Core	Laravel	RoR
■ Avg	0,02139177	0,19324633	0,23886883	7,86503598	73,8847103	3,9786743
■ Stdev	0,16315444	1,36508372	2,01641198	5,99312349	28,4763988	13,0669591
■ Max	0,01722420	0,15135604	0,21232834	0,28549066	0,8493261	0,1921237

Fig. 3. Web application request processing speed, ms

Web frameworks have specific conditions of use. If you have chosen a small web framework and need to develop a web application that is different from simple applications or the REST API, then you are likely to have problems with enhanced functionality, and vice versa - the redundancy of a full-featured, large web frameworks will cause financial the cost of placing content under high load shown in Fig. 4.

	Spring Boot	Go	Node.js	ASP.NET Core	Laravel	RoR
■ Avg	46746,9449	5174,7425	4186,3981	127,1451	13,5346	251,3404
■ Stdev	6129,1619	732,5558	495,9304	166,8579	35,1168	76,5289
■ Max	58057,8467	6606,9381	4709,6868	3502,7415	1177,4041	5204,9801

Fig. 4. The requests number to the web application per second

As a result, it was concluded that in conditions of hosting hardware resources it is advisable to develop web servers for hackathons based on the Spring Boot web framework.

The Methods of Determination and Awarding of Hackathon Winners
Existing algorithms for determining winners in sports

Over the years certain systems of selecting winners in different activities have developed. Sports are the oldest form of entertainment and competition. Since ancient Egypt and to this day people like to compete and win. Losing in any game, we strive to win the next one. eSports is already included in the Register of recognized sports in many countries. In September 2020 Ukraine also became one of more than 25 countries where cybersport is recognized as an official sport [10].

Therefore, it makes sense to consider the existing systems for determining winners in various sports (including Olympics) and analyze how they are suitable for judging in e-sports. Among the most popular systems of competitive selection of winners are the Olympic system ("playoffs") and the so-called "double-elimination system".

E-sports is steadily approaching to become included in the program of the 2024 Olympics [11], therefore it is advisable to consider in detail the features. Playoffs ensure that a winner is determined in a minimum number of rounds and promote a hard-fought tournament. Among the playoffs' advantages are the minimum number of games compared to other tournament variants and their "uncompromising" nature: there is no possibility and no point in a tie-break.

However, the playoffs are completely unsuitable for tournaments where it is important to ensure a fair distribution of all places, not just 1st – 3rd places.

First, in the playoffs, the distribution of places other than first is extremely influenced by the order in which the pairings are chosen. In a draw, the last places are allocated almost randomly: a weak competitor, who is drawn against an even stronger opponent, may easily rise above a stronger competitor in the first round [12].

Also, in a pure playoff, places other than 1st and 2nd can't be assigned at all. If it is necessary to specify the places occupied by participants, additional games have to be played, which is the greatest disadvantage of this selection system, because the main advantage of playoffs – speed – is lost.

In addition, a major disadvantage of playoffs is the rigid requirements on the number of participants. If that number does not comply with a norm, the only solution is to draw a draw to award technical wins or technical losses to some participants in the first round, which further increases the random factor's influence on the outcome of the tournament. The only alternative is to preempt a playoff tournament with a series of preliminary games for entry into the main tournament.

Thus, the Olympic system is built largely on a series of randomness, and the outcome of the competition is largely decided by lot, which is not fair to the other participants. Unlike familiar to us volleyball, soccer, or chess, in cybersport, everything is not so straightforward, and it becomes much more difficult to choose a winner.

So, in cybersport, a tournament system with elimination after two defeats, or as it is called, "two-consolation" or "double-elimination system" is commonly used [13]. This system is in contrast significantly to the simple Olympic system, in which a single defeat results in elimination. The double-elimination system is used in sports where it is easy to play twice as many matches as in the Olympic system, either due to the short match length or the large number of arenas running in parallel (automobile sports, darts, judo, etc.).

It is currently used in cybersports tournaments, including Dota 2 and CS: GO. In the tournament to two defeats played 2n–1 or 2n–2 games, depending on the outcome of the superfinal. This is at least twice as much as in the Olympic system, and the number of rounds at least one more. You can notice that more games must be played to reach the superfinals in the lower net than in the upper net.

Without detracting from the merits of this system, however, among its disadvantages should be noted the following:

1)   special requirements for the number of participants (ideally a degree of twos). With the use of computer systems for competitions, there are usually no problems with the number of participants;

2)   two sportsmen may face each other twice (and sometimes even three times);

3)   most matches are played between outsiders and mediocre players and are of interest only to a narrow group of fans;

4)   it is difficult to transport participants from one arena to another. In the Olympic system, for example, four stadiums host different branches of the tournament up to the quarterfinals, and then everyone is taken to one stadium where the semifinals and finals are played.

Thus, this system is also not the best way to select a winner and goes along with the Olympic system.

Based on the results of the analysis, it can be argued that traditional selection systems, even the double-elimination system, which is currently used in cybersport, are not universal. Against the backdrop of the sport's growing popularity, they carry with them a certain amount of inconvenience and unfairness. Therefore, it is advisable to develop a new algorithm for determining the winners of hackathons as a component of eSport.

### Existing algorithms for determining winners in sports

Within this study, there is a development of the algorithm for a selection system that consists of two, at most three rounds and is completely independent of the number of participants.

The main goal of such system is to ensure the fairness of the competitive selection concerning all participants. The players will not be able to negotiate or bribe the judge, since the winner is selected by general voting.

The system is based on the Borda method of ranking. The Borda method is a voting system invented in 1770 by Jean-Charles de Borda to make the preferences of the electors more accurate when there are many candidates [14]. According to this method, the results of voting are expressed as the number of points scored by each of the candidates. Often this method does not give intuitively expected results when counting, thus preserving intrigue until the winners are announced [15].

This system can be compared to the selection system in parliamentary or presidential elections [16]. In this case, during the first round, each of the participants is assigned points in descending order from more to less. The system automatically counts the sum of points for each of the participants separately and arranges them in order from more to less respectively. Then a certain percentage of participants from the bottom (those with the lowest number of points) is "discarded". This percentage is not due to the Borda method, so the judge or the game administrator writes it into the program.

As a consequence of this selection, the participants who scored the highest number of points in total, go to the second round. Then the cycle repeats. The algorithm of the method is shown in Fig. 5.
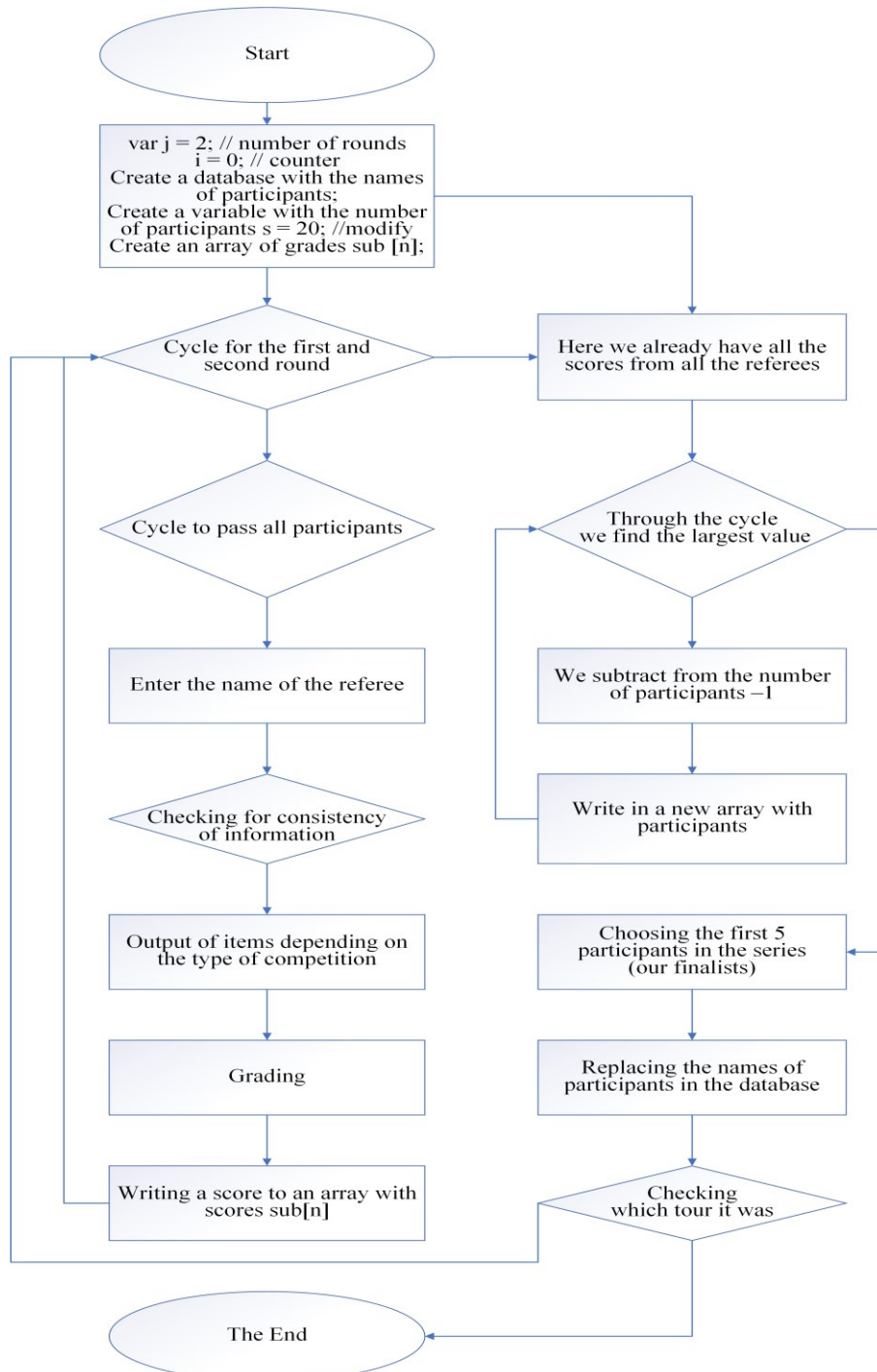
Fig. 5. The algorithm of the modified Borda method

The main advantages of this system for selecting a winner are:
1) the minimum number of rounds, and therefore the minimum amount of time to select a winner;
2) the impossibility of rigged matches, and thus fairness to all competitors;
3) intrigue until the end of the competition, up to the moment of the announcement of winners;
4) the possibility of fair assignment of prizes in any quantity.

The proposed system will allow a fair selection of the winner of any competition by general voting. And since cybersport is mostly a choice of a winner by voting by several judges, this system will help to calculate the number of points for each participant regardless of the others.

Conclusion

The proposed Multi-Agent Sell Funnel Monitoring (MASFM) algorithm allows to search sponsorship efficiently because according to last 2 years statistics about 16–23% of new sponsors was detected by MASFM. Now, when using Spring Boot, the maximum time is less than 3 seconds and the number of requests has increased from 71 to 94 requests per second. As expected, the percentage of errors increased to 29%. These all errors were by the hosting provider fault and test dyno container.

In the software architecture of the online hackathons' platform, a real scenario of increasing performance 15 times from 6 to 94 requests/sec was applied, which does not require serious refactoring and complex code changes. Besides, the steps mentioned above can reduce the cost of infrastructure like Heroku. The next functionality of the online hackathon platform will be possible thanks to the microservices architecture.

The applied system of grading based on modified Borda was assessed by the participants as fair enough. When analyzing the responses of the participants, negative responses due to the subjective conduct of judges decreased to 1–2%. The number of repeated participations in the hackathon is consistently above 70%.

Acknowledgments

References

1. Izvalov A., Nedilko S., Nedilko V. Comparison of game creation and engineering hackathons on the global and local levels. *In Proceedings of the Second International Conference on Game Jams, Hackathons, and Game Creation Events (ICGJ '17).* ACM International Conference Proceeding Series. 2017. Pp. 22–25. https://doi.org/10.1145/3055116.3055119.

2. Coronavirus: European Commission launches #EUvsVirus online hackathon. Publ. 2020, April 16. URL: https://news.liga.net/all/pr/koronavirus-evrokomissiya-zapustila-onlayn-hakaton-euvsvirus (Last accessed: 04.02.2023).

3. @divchataSTEM – Community. URL: https://www.facebook.com/divchataSTEM/ (Last accessed: 04.02.2023)

4. Open set for the 5th hackathon DEHACK 2019. URL: https://chmnu.edu.ua/vidkritij-nabir-na-p-yatij-hakaton-dehack-2019/ (Last accessed: 04.02.2023).

5. Trunov A., Fisun M., Malcheniuk A. The processing of hyperspectral images as matrix algebra operations. In Proceedings of the 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET). IEEE, Lviv-Slavske, Ukraine. 2018. Pp. 368–372. https://doi.org/10.1109/TCSET.2018.8336305.

6. Lavrynenko S., Kondratenko G., Sidenko I., Kondratenko Yu. Fuzzy logic approach for evaluating the effectiveness of investment projects. *In Proceedings of the 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT'2020).* IEEE, Zbarazh, Ukraine. 2020. No. 2. Pp. 297–300, Article number 9321880. https://doi.org/10.1109/CSIT49958.2020.9321880.

7. Shved A., Kovalenko I., Davydenko Ye. Method of detection the consistent subgroups of expert assessments in a group based on measures of dissimilarity in evidence theory. *Advances in Intelligent Systems and Computing. 1080 AISC.* 2020. Pp. 36–53. https://doi.org/10.1007/978-3-030-33695-0_4.

8. Tohoiev O., Burlachenko I., Zhuravska I., Savinov V. The monitoring system based on a multi-agent approach for moving objects positioning in wireless networks. *In Proceedings of the 3rd International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020).* Zaporizhzhia, Ukraine, 2020. No. 2608. Pp. 79–90.

9. Zhuravska I., Obukhova K., Burlachenko I., Savinov V., Boiko A. Heatmaps for catering establishments web-applications available via MAS-improved wireless networks. *In Proceedings of the 5th IEEE International Symposium on Smart and Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS'2020).* IEEE, Dortmund, Germany. 2020. 6 p. Article number 9297107. https://doi.org/10.1109/IDAACS-SWS50031.2020.9297107.

10. Register of recognized sports in Ukraine: approved by the order of the Ministry of Youth and Sports of March 11, 2015, no. 639 (as amended by the order of Sept. 16, 2020, no. 1557). URL: https://zakon.rada.gov.ua/rada/show/v0639728-15#Text (Last accessed: 04.02.2023).

11. Cybersport at the Olympiad'2024? Publ. 21.02.2019. URL: https://futurenow.com.ua/kibersport-mozhe-zyavytysya-na-olimpiad/ (Last accessed: 04.02.2023).

12. Csato L. Two issues of the UEFA Euro 2020 qualifying play-offs. *International Journal of Sport Policy.* 2020. Vol. 12, Is. 1. Pp. 1–14. https://doi.org/10.1080/19406940.2020.1780295.

13. Aziz H., Gaspers S., Mackenzie S., Mattei N., Stursberg P., Walsh T. Fixing balanced knockout and double elimination tournaments. *Artificial Intelligence.* 2018. Vol. 262 (May 2018). https://doi.org/10.1016/j.artint.2018.05.002.

14. Felsenthal D. S., Nurmi H. Voting procedures under a restricted domain: An examination of the (In)vulnerability of 20 voting procedures to five main paradoxes. Springer, 2019. 92 p.

15. Regenwetter M., Grofman B. Approval voting, Borda winners and Condorcet winners: Evidence from seven elections. *Management Science.* 1998. Vol. 44, Is. 4. Pp. 520–533.

16. Electoral Systems. ACE electoral knowledge network. URL: https://aceproject.org/ace-ru/topics/es/ese/ese01/prezidentskie-vybory-dvuhturovaya-sistema (Last accessed: 04.02.2023).

17.   The Faculty of Computer Science held the "Hackathon-2021", dedicated to the 25th anniversary of Petro Mohyla Black Sea National University. URL: https://chmnu.edu.ua/fakultet-komp-yuternih-nauk-proviv-hackathon-2021-prisvyachenij-25-richchyu-chnu-imeni-petra-mogili/ (Last accessed: 04.02.2023).

Ivan Burlachenko **Іван Бурлаченко**	Senior Lecturer of the Department of Computer Engineering, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine, e-mail: ivan.burlachenko2010@gmail.com https://orcid.org/0000-0001-5088-6709	старший викладач кафедри комп'ютерної інженерії, Чорноморський національний університет імені Петра Могили, Миколаїв, Україна
Volodymyr Savinov **Володимир Савінов**	PhD on Engineering, Associate Professor, Associate Professor of the Department of Computer Engineering, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine e-mail: volodymyr.savinov@chmnu.edu.ua https://orcid.org/0000-0002-0862-5879	канд. техн. наук, доцент, доцент кафедри комп'ютерних інженерії, Чорноморський національний університет імені Петра Могили, Миколаїв, Україна
Iryna Zhuravska **Журавська Ірина**	DrS on Engineering, Professor, Head of the Department of Computer Engineering, Petro Mohyla Black Sea National University, Mykolaiv, Ukraine e-mail: iryna.zhuravska@chmnu.edu.ua https://orcid.org/0000-0002-8102-9854	доктор техн. наук, проф., зав. кафедри комп'ютерних інженерії, Чорноморський національний університет імені Петра Могили, Миколаїв, Україна