

О.П. ЧЕРНИХ, канд. фіз.-мат. наук, доц., НТУ "ХПІ",
А.В. КОРКОШКО, магістр, НТУ "ХПІ"

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДВИЩЕННЯ ПРОДУКТИВНОСТІ ВЗАЄМОДІЇ МІЖ КЛАСТЕРАМИ СЕРВЕРА

Досліджені основні методи та алгоритми розподілення навантаження. На основі результатів досліджень розроблені алгоритм аналізу коефіцієнта активності інтернет сесій та алгоритм визначення критичного навантаження. За допомогою даних алгоритмів розроблено програмне забезпечення, яке дає можливість уникнути відмови системи із-за перенавантаження. Іл.: 6. Бібліогр.: 8 назв.

Ключові слова: розподілення навантаження, коефіцієнт активності, інтернет сесія, критичне навантаження.

Постанова проблеми. У століття розвитку індустрії ІТ технологій вже важко уявити собі будь-яке підприємство, яке орієнтовано на комерційну або некомерційну діяльність, без використання автоматизації процесів. Все автоматизується за допомогою серверів, навантаження на які з часом тільки зростають. "Падіння" сервера може відбутися в самий невідповідний момент та загрожувати дуже серйозними моральними і матеріальними наслідками. Проблеми недостатньої продуктивності сервера в зв'язку зі зростанням навантажень можна вирішувати шляхом нарощування потужності сервера або оптимізацією алгоритмів, що використовуються, та програмних кодів.

Аналіз літератури. Так як обчислювальні потужності комп'ютерів вже доходять до межі фізичних можливостей, то доводиться вдаватися до кластеризації. Ефективність кластеризації безпосередньо залежить від того, як розподіляється навантаження між елементами кластера. В багатьох дослідженнях пропонуються різні підходи до проблеми балансування навантаження [1, 2].

Для використання балансування потрібно виділяти наступні цілі [3]:

- справедливість: на обробку кожного запиту виділяються системні ресурси;
- ефективність: всі сервери, які обробляють запити, повинні бути зайняті на 100%;
- скорочення часу виконання запиту: забезпечення мінімального часу між початком обробки запиту (або його постановкою в чергу на обробку) і його завершенням;
- скорочення часу відгуку: потрібно мінімізувати час відповіді на

запит користувача.

Дуже бажано, щоб алгоритм балансування мав наступні властивості [3]:

– передбачуваність: чітке розуміння в яких ситуаціях і при яких навантаженнях алгоритм буде ефективним для вирішення поставлених завдань;

– рівномірне завантаження ресурсів системи;

– масштабування: алгоритм повинен зберігати працездатність при збільшенні навантаження.

Алгоритми балансування навантаження можна поділити на дві категорії: статичні і динамічні [4, 5]. Процедура балансування може здійснюватися відповідно рівням моделі OSI: мережевого, транспортного, прикладного.

Процедура балансування може здійснюватися за допомогою алгоритмів балансування [3, 6]:

1) алгоритм Round Robin або алгоритм кругового обслуговування;

2) алгоритм Weighted Round Robin;

3) алгоритм Least Connections;

4) алгоритм Destination Hash Scheduling;

5) алгоритм Sticky Sessions.

Для комерційних додатків [7] електронні магазини найчастіше використовують алгоритм Sticky Sessions. Цей алгоритм розподілу вхідних запитів гарантує, що запити одного і того ж клієнта будуть передаватися на один і той же сервер.

Більшість відомих розподілювачів навантаження забезпечують розподілення навантаження без функціональності, яка б допомогла розвантажити вузол, вони лише можуть виключити її на деякий час з кільця розподілення навантаження, тобто нові інтернет сесії не будуть закріплюватися за цим вузлом.

Одна з існуючих проблем розподілення навантаження – зменшення навантаження на сервер у випадку критичного перенавантаження та нестачі ресурсів для роботи з задовільною продуктивністю. Описаний підхід у даній статті дозволяє вирішити дану проблему.

Мета статті – розробка програмного забезпечення для комплексного балансування навантаження з можливістю перенесення існуючого навантаження між вузлами кластера і здатністю до легкої інтеграції в існуючі програми.

Для досягнення зазначеної мети були поставлені основні задачі:

– розробка алгоритму визначення коефіцієнту активності інтернет сесії;

– розробка алгоритму визначення критичного навантаження;

– програмна реалізація розподілення навантаження з перевизначенням найактивніших сесій на ненавантажені вузли.

Розробка алгоритму визначення коефіцієнту активності інтернет сесії. На рис. 1 наведений алгоритм обробки запиту методом Sticky Session. З цього алгоритму видно, що кожна сесія стійко закріплюється за вузлом і до кінця свого життєвого циклу оброблюється ним.

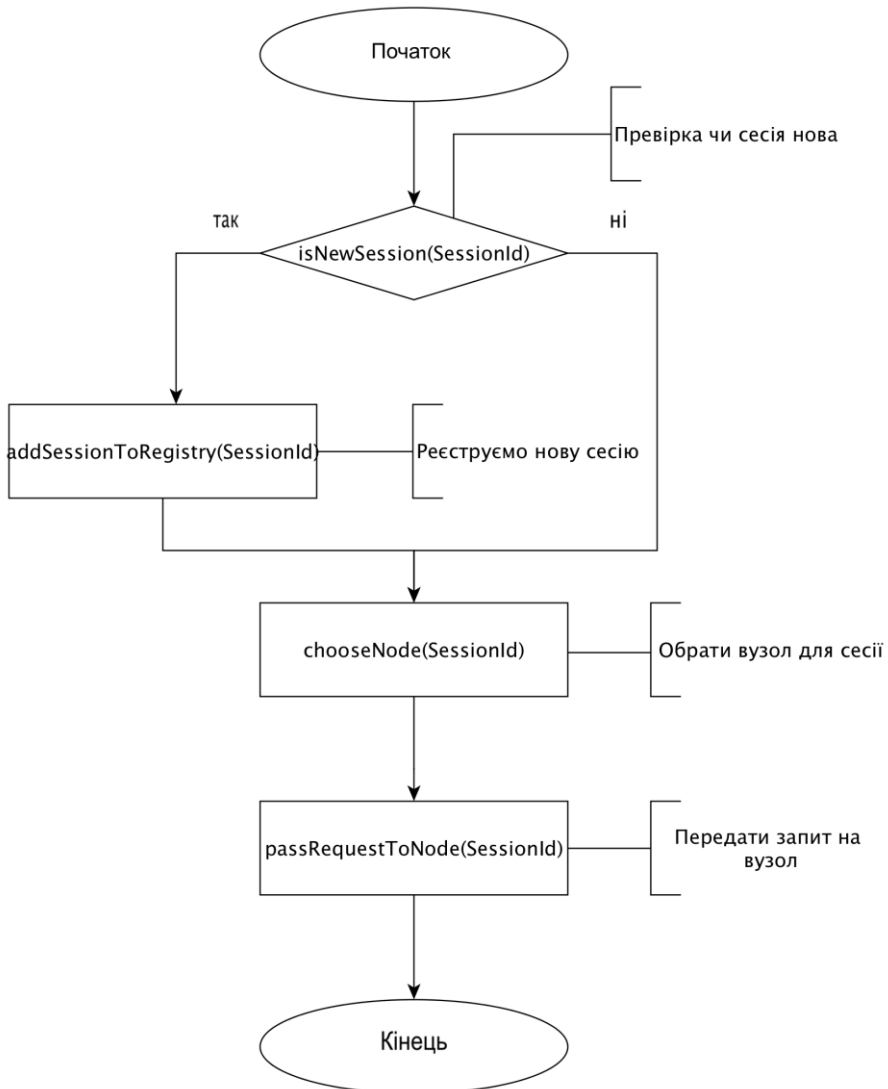


Рис. 1. Алгоритм обробки запиту методом Sticky Session

Метод Sticky Session має недоліки:

- можливе перенавантаження вузла у разі нерівномірного розподілення навантаження, яке не можливо передбачити;
- у випадку динамічного кластеру неможливо перерозподілити навантаження та вимкнути зайві потужності, що призводить до матеріальних втрат.

На рис. 2 наведений алгоритм обробки запиту методом Dynamic Sticky Session, який був розроблений.

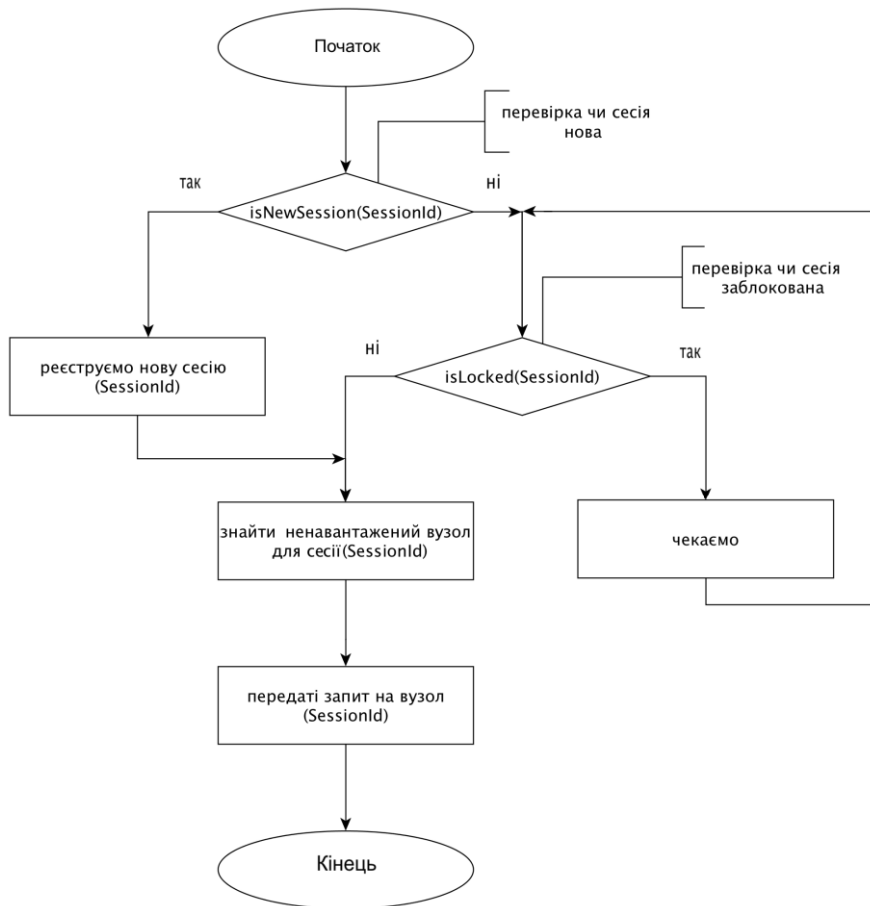


Рис. 2. Алгоритм обробки запиту методом Dynamic Sticky Session

Цей алгоритм не відрізняється від оригінального алгоритму методу Sticky Session. Але незначна модифікація насправді дуже важлива. Вона дозволяє переміщувати сесії між вузлами. Коли сесія повинна бути переміщена, виконується її блокування, за що відповідає операція

IsLockedSession. Надана можливість переміщення сесій дозволяє повністю вирішити недоліки методу Sticky Session [8].

Розробка алгоритму визначення критичного навантаження. На рис. 3 пропонується алгоритм, за яким виконується розвантаження вузла, якщо він у критичному стані. Комп'ютерна система "Session Thrower Load Balancer", яка була розроблена, реалізує метод Dynamic Sticky Session та виконує операцію, яка дозволяє за конфігурований проміжок часу тримати кластер в оптимальному стані.

Розробка комп'ютерної системи для підвищення продуктивності. Комп'ютерна система "Session Thrower Load Balancer" – є програмним забезпеченням для всіх операційних систем, для яких реалізована специфікація JVM (Java Virtual Machine). Функціональні характеристики розробленої системи:

- балансування навантаження методом sticky session;
- легкість інтеграції;
- здатність конфігурувати правила для кожного вузла;
- розвантаження вузла під час виконання;
- будування онлайн графіка навантаження для кожного вузла.

Система визначена як набір спеціальних програмних інструментів, які допомагають в рівномірному розподіленні навантаження між вузлами кластера. Для розрахунку навантаження вузла визначені наступні метрики:

- кількість зайнятої оперативної пам'яті;
- рівень навантаження центрального процесору.

За допомогою розробленого інструмента виконується обчислення часу, яке вузол витратив на обробку запитів кожної сесії. Значення може змінюватися від 0 до 10000.

Розрахунок виконувався за наступною формулою

$$P = t / T * 100, \quad (1)$$

де P – значення активності сесії; t – час витрачений на обробку запитів кожної сесії; T – загальний час виміру.

На кожний запит користувача генерується звіт з кількістю часу, який було витрачено на запит. За кожний зазначений проміжуток часу генерується загальний звіт і відправляється на роподілювач навантаження для подальшої обробки і роподілення навантаження.

Розроблений інструмент складається з розподілювача навантаження Session Thrower Load Balancer (STLB) та клієнтського модулю STLB Client, який буде інтегруватися з існуючими застосуваннями.

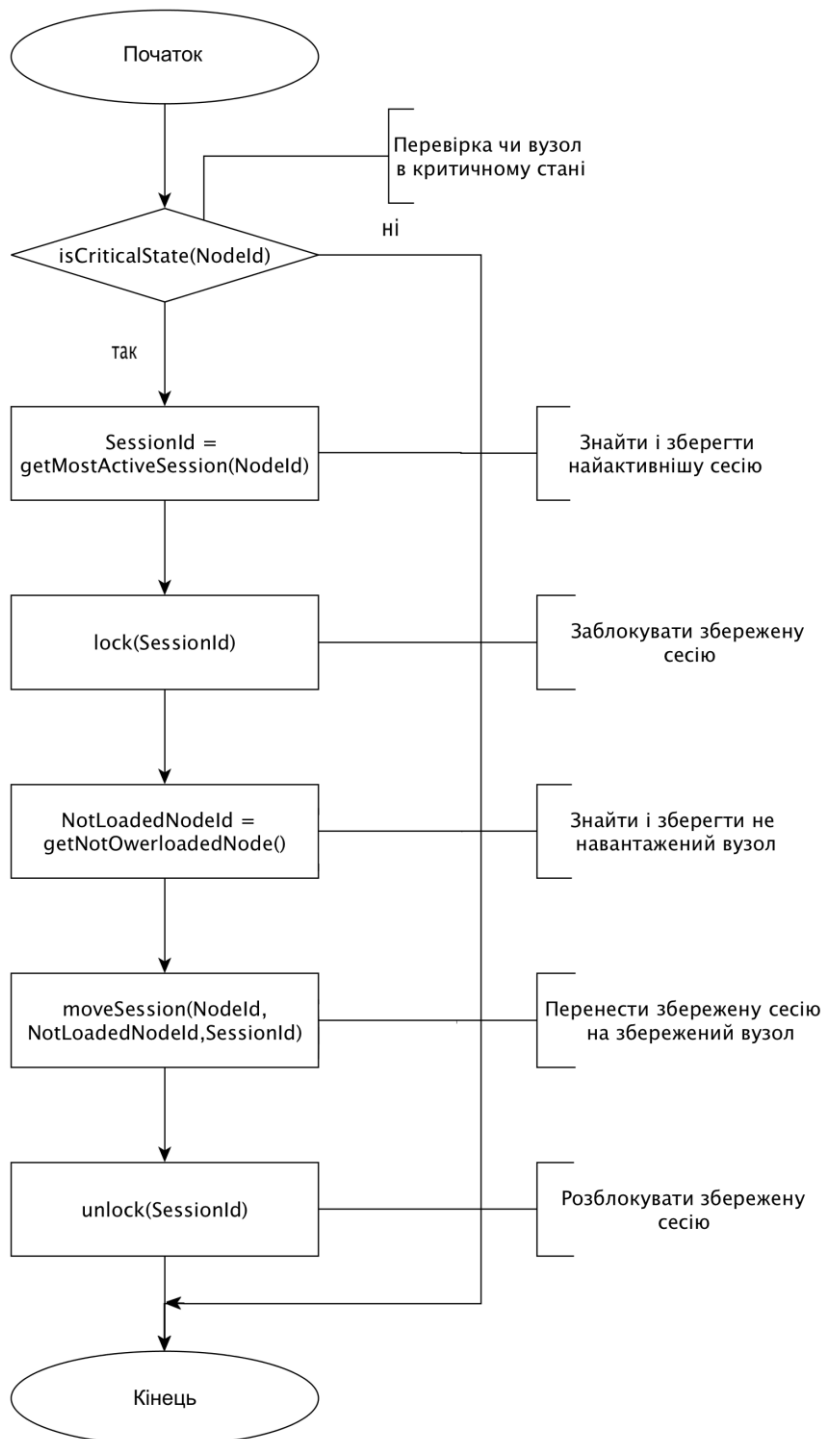


Рис. 3. Алгоритм розвантаження вузла

Для зручності та простоти інтеграції було обрано один з найпопулярніших контейнерів застосувань на мові програмування Java, Spring. Також була розроблена архітектура, з якою можна було інтегруватися за найменший проміжуток часу та з мінімум зусиль.

Для автоматизації зборки проекту використано Apache Maven – фреймворк для автоматизації збирання проектів, специфіковані на XML-мові POM (Project Object Model). Maven, на відміну від іншого збирача проектів Apache Ant, забезпечує декларативну, а не імперативну збірку проекту. Усі завдання з обробки файлів Maven виконує через плагіни. При використанні Maven у проекті використовується чітка структура розташування каталогів.

В пропанованій системі реалізована можливість перегляду службової інформації, яка постійно приходить на розподільвач навантаження. Її можна переглянути на онлайн графіках: навантаження центрального процесору, навантаження ОЗУ, навантаження процесора програмою застосування (рис. 4 – 6).

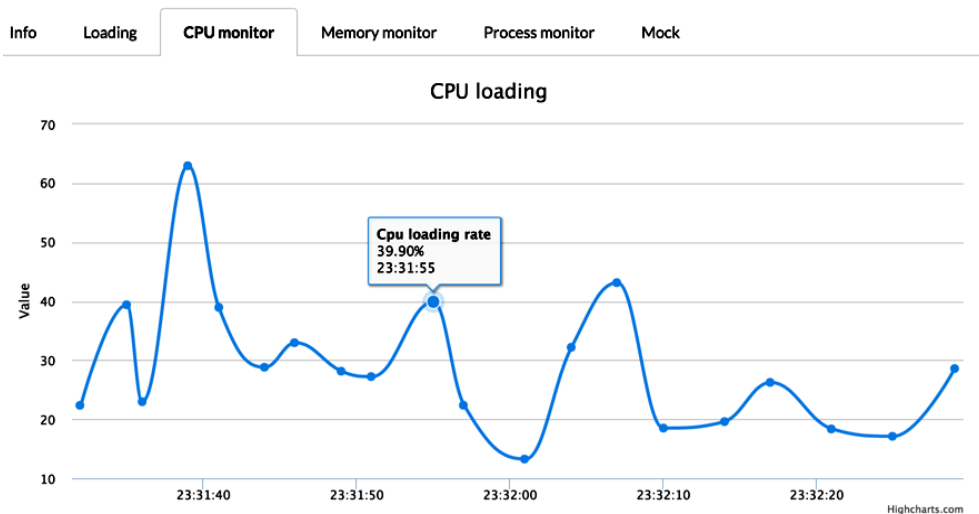


Рис. 4. Графік навантаження центрального процесору



Рис. 5. Графік навантаження ОЗУ

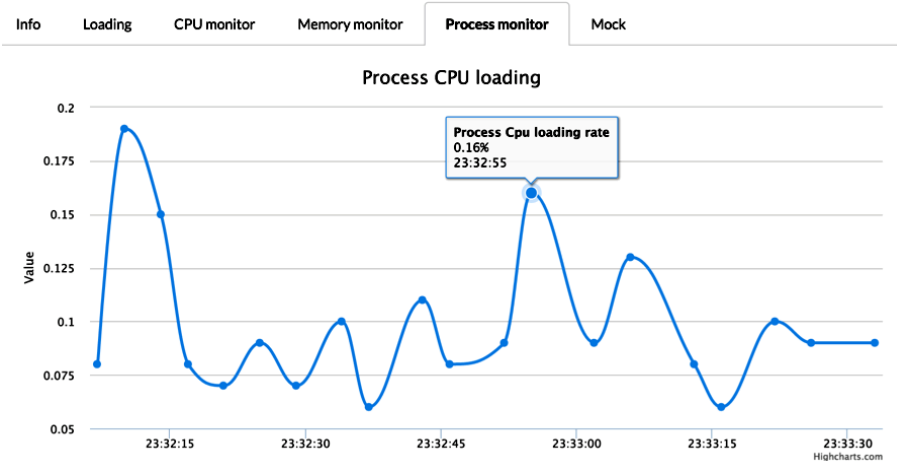


Рис. 6. Графік загрузки пам'яті програмою застосування

Висновки. Таким чином, розроблено програмне забезпечення "Session Throwing Load Balancer" реалізує новий спосіб розподілення навантаження методом перезакріплення сесій. Це значно покращує продуктивність та відмовостійкість всього кластера. За допомогою цього програмного забезпечення можливо проводити моніторинг та аналіз навантаження на кожному вузлі та редагувати правила розподілення навантаження в разі надзвичайних ситуацій.

Список літератури: 1. Жуков І.А. Методи балансування навантаження для web-серверів / І.А. Жуков // Проблеми інформатизації та управління. – 2007. – № 3 (21). – С. 46-54. 2. Chulhye P. A fuzzy-based distributed load balancing algorithm for large

distributed systems / P. Chulhye, J.G. Kuhl // Proceedings of the Second International Symposium on Autonomous Decentralized Systems. – 1995. – P. 266-273. **3.** Часовских А. Обзор алгоритмов кластеризации данных. – Режим доступа www. URL: <http://habrahabr.ru/post/101338/> – 11.08.2010. **4.** Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования / Г.Р. Эндрюс. – М.: Вильямс, 2003. – 512 с. **5.** Топорков В.В. Модели распределенных вычислений / В.В. Топорков. – М.: ФИЗМАТЛИТ, 2004. – 320 с. **6.** Шаповал І.С. Адаптивний алгоритм балансування навантаження в комп'ютерних мережах / І.С. Шаповал // Матеріали міжнародної науково-практичної конференції аспірантів і студентів "Інженерія програного забезпечення 2010". – 2010. – № 3. – С. 26-35. **7.** Коркошко А.В. Исследование методов построения приложений электронной коммерции с высокой скоростью и надёжностью / А.В. Коркошко, О.П. Черних // Матеріали ІХ Університетської науково-практичної студентської конференції магістрів НТУ "ХПІ". – 2015. – С. 74. **8.** Коркошко А.В. Розробка програмного модулю для підвищення продуктивності взаємодії між кластерами сервера / А.В. Коркошко, О.П. Черних // Матеріали XXIV Міжнародної науково-практичної конференції "Інформаційні технології: наука, техніка, технологія, освіта, здоров'я". – 2016. – Ч. IV. – С.148.

References:

1. Zhukov, I.A. (2007), "The methods of load balancing for web-servers", *Problems of information and management*, 3 (21), pp. 46-54.
2. Chulhye, P. and Kuhl, J.G. (1995), "A fuzzy-based distributed load balancing algorithm for large distributed systems", *Proceedings of the Second International Symposium on Autonomous Decentralized Systems*, April 1995, pp. 266-273.
3. Chasovskyh, A. (2010), "Review of data clustering algorithms", <http://habrahabr.ru/post/101338/>. – (accessed 11.08.2016).
4. Andrews, G.R. (2003), *Fundamentals of multi-threaded, parallel and distributed programming*, M.: Williams, 512 p.
5. Toporkov, V.V. (2004), *Models of distributed computing*, M.: FIZMATLIT, 320 p.
6. Shapoval, I.S. (2010), The adaptive algorithm of load balancing in computer networks. *Proceedings of the international Scientific Conference of Graduate and Students "Engineering software 2010"*, 2010, No. 3, p. 26-35.
7. Korkoshko, A.V. and Chernykh, E.P. (2015), "Investigation of construction methods of e-commerce applications with high speed and reliability", *IX University Scientific-Practical Student Conference Master NTU "HPI"*, April 2015, p. 74.
8. Korkoshko, A.V. and Chernykh, E.P. (2016), "Development of the module for increased performance interaction between server clusters", *XXIV International Scientific-Practical Conference "Information Technology, Science, Engineering, Technology, Education, Health"*, May 2016, Part. IV, p.148.

Статтю представив д-р техн. наук, проф. НТУ "ХПІ" Леонов С.Ю.

Надійшла (received) 09.08.2016

Chernykh Olena, Cand. Phys.- Math. Sci., Docent
National Technical University "Kharkov Polytechnic Institute"
Str. Frunze, 21, Kharkov, Ukraine, 61002
tel./phone: +38 (098) 427-01-52, e-mail: lenachernikh@gmail.com
ORCID ID: 0000-0003-3883-8645

Korkoshko Andrii, PhD
National Technical University "Kharkiv Polytechnic Institute"
Str. Pushkinskaya, 79/5, Kharkov, Ukraine, 61024
tel./phone: +38 (099) 279-01-40, e-mail: demosoft@bk.ru
ORCID ID: 0000-0002-4567-5584

УДК 004.07

Розробка програмного забезпечення для підвищення продуктивності взаємодії між кластерами сервера / Черних О.П., Коркошко А.В. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2016. – № 44 (1216). – С. 171 – 181.

Досліджені основні методи та алгоритми розподілення навантаження. На основі результатів досліджень розроблені алгоритм аналізу коефіцієнта активності інтернет сесій та алгоритм визначення критичного навантаження. За допомогою даних алгоритмів розроблено програмне забезпечення, яке дає можливість уникнути відмови системи із-за перенавантаження. Ил.: 6. Библиогр.: 8 назв.

Ключові слова: розподілення навантаження, коефіцієнт активності, інтернет сесія, критичне навантаження.

УДК 004.07

Разработка программного обеспечения для повышения производительности взаимодействия между кластерами сервера / Черных А.П., Коркошко А.В. // Вестник НТУ "ХПИ". Серія: Інформатика та моделювання. – Харьков: НТУ "ХПИ". – 2016. – № 44 (1216). – С. 171 – 181.

Исследованы основные методы и алгоритмы распределения нагрузки. На основе результатов исследований разработаны алгоритм анализа коэффициента активности интернет сессий и алгоритм определения критической нагрузки. С помощью этих алгоритмов разработано программное обеспечение, которое дает возможность избежать отказа системы из-за перегрузки. Ил.: 6. Библиогр.: 8 назв.

Ключевые слова: распределение нагрузки, коэффициент активности, интернет сессия, критическая нагрузка.

UDC 004.07

Development software to improve the performance of interaction between server clusters / Chernykh O.P., Korkoshko A.V. // Herald of the National Technical University "KhPI". Subject issue: Information Science and Modelling. – Kharkov: NTU "KhPI". – 2016. – № 44 (1216). – P. 171 – 181.

The basic techniques and load balancing algorithms were investigated. Based on the research results the algorithm of analyzes the activity coefficient of the Internet sessions and the algorithm for determining the critical load were developed. With using these algorithms the software was designed which avoids system failure due to overloading. Figs.: 6. Refs.: 8 titles.

Keywords: load balancing, activity coefficient, Internet session, the critical load.