DOI 10.31891/2307-5732-2019-273-3-208-214
УДК 004.056.52:004.492.2

О.А. КНЯЗЄВ
Одеська національна академія зв'язку ім. О.С. Попова

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АДАПТИВНОЇ КОМПЛЕКСНОЇ СИСТЕМИ ФІЛЬТРАЦІЇ КОНТЕНТУ

З розвитком інформаційно-комунікаційних технологій та імплементації мережі Інтернет майже в усіх сферах життєдіяльності суспільства виникає проблема розповсюдження в мережі Інтернет та проникнення до користувачів небажаного контенту, який може нанести користувачам як моральну, так і матеріальну шкоду. Одним зі шляхів вирішення цієї проблеми є фільтрація контенту, що поступає до користувача. Це призводить до формулювання науково-прикладного завдання щодо формування системи фільтрації контенту в мережі Інтернет, а також прикладного програмного забезпечення реалізації цього завдання, що обумовлює актуальність даного дослідження. Метою статті є формування та реалізація прикладного програмного забезпечення процесу адаптивної комплексної фільтрації контенту в мережі Інтернет. На основі запропонованої адаптивної комплексної системи фільтрації контенту (АКСФК), що відрізняється від існуючих тим, що завдяки постійному оновленню списків та профілю користувачів має більш викосу якість фільтрації контенту, розроблено програмний продукт «Проксі Блокер», для якого в якості платформи розробки було обрано програмну оболонку Debian. Наведено послідовність проходження процедури фільтрації контенту. Виконано тестування програмного продукту «Проксі Блокер», яке показало, що чим рідше здійснюється сортування, тим швидше відбувається обробка запиту і навпаки. Сформовано математичну модель залежності часу обробки запиту від інтервалу сортування, що надасть змогу визначити подальшу динаміку змін часу обробки запиту при зростанні (скороченні, зміни кроку) інтервалу сортування. Результати моделювання свідчать, що з вірогідністю $R^2 = 0,9576$ модель залежності часу обробки запиту від інтервалу сортування має поліноміальну залежність четвертого ступеню. Моделювання здійснено методом апроксимації за допомогою стандартного програмного забезпечення Microcoft Office Excel. Виконано порівняння роботи запропонованої АКСФК та програмного продукту «Проксі Блокер» з існуючими на прикладі однієї з найбільш розповсюджених програм Rejik+Squid. Визначено, що середнє значення часу сортування під час застосування програми Rejik+Squid збільшується при збільшенні кількості запитів. Втім, аналогічні показники для програми «Проксі Блокер» у більшості проведених експериментів менші. Зокрема, лише у випадку одиничного запиту час обробки продуктом Rejik+Squid на 66% кращий, ніж при використанні програмного продукту «Проксі Блокер». Також на 17,1% швидше відбувається обробка запитів у разі, коли їх кількість становить 100000. В інших випадках розроблений програмний продукт «Проксі Блокер» показує кращій середній час обробки запитів, який в середньому на 2,5% менший за аналогічні показники продукту Rejik+Squid. Таким чином, розроблений програмний продукт «Проксі Блокер», на відміну від існуючих, показує кращі за часом результати фільтрації контенту, що свідчить на користь його застосування у АКСФК. Подальші дослідження будуть спрямовані на визначення швидкості проведення процедури фільтрації розробленою програмою у випадках рандомних запитів.

Ключові слова: *фільтрація контенту, програмне забезпечення, адаптивна комплексна система, сортування, контент.*

OLEKSANDR KNIAZIEV
Odessa National Academy of Telecommunications n.a. O.S. Popov

THE SOFTWARE OF ADAPTIVE COMPLEX SYSTEM OF THE FILTRATION OF THE CONTENT

With the development of information and communication technologies and the implementation of the Internet in almost all spheres of life of the society there is a problem of distribution on the Internet and penetration into users of unwanted content that can inflict on users both moral and material damage. One way to resolve this problem is to filter content that enters the user. This leads to the formulation of a scientific and applied task for the formation of a content filtering system on the Internet, as well as application software implementation of this task, which determines the relevance of this study. The purpose of the article is to create and implement application software of the process of adaptive integrated content filtering on the Internet. Based on the proposed Adaptive complex system of content filtering (ACSCF), which differs from the existing ones, due to the constant updating of the lists and user profile, the quality of content filtering is more profitable, the software product «Proxy Bloker» was developed, for which the software was chosen as the development platform. Debian shell. The sequence of passing the content filtering procedure is shown. Testing of the «Proxy Blocker» software was performed, which showed that the more rarely the sorting is done, the faster processing of the request and vice versa. The mathematical model of the dependence of the processing time of the query on the sorting interval, the prolongation of which will allow to determine the further dynamics of changes in the processing time of the query with the growth (reduction, change of step) of the sorting interval. The simulation results indicate that, with probability $R^2 = 0,9576$, the model of the dependence of the processing time of the query on the sorting interval has a polynomial dependence of the fourth degree. The simulation was carried out by the approximation method using the standard Microcoft Office Excel software. A comparison of the work of the proposed ACSCF and the «Proxy Bloker» product software with the existing one, based on the example of one of Rejik+Squid most popular programs, has been made. It is determined that the average time of sorting when applying Rejik+Squid program increases with increasing the number of requests. However, similar rates for the «Proxy Bloker» program are lower in most of the experiments conducted. In particular, only in the case of a single query processing time by Rejik+Squid is 66% better than when using «Proxy Blocker» software. Also 17,1% faster processing requests in the case when their number is 100000. In other cases, the developed software «Proxy Bloker» shows a better average query processing time, which is on average 2,5% lower than the similar indicators of the product Rejik+Squid. The «Proxy Bloker» software program, in contrast to the existing ones, shows the best results in terms of content filtering, which proves the benefit of its use in ACSCF. Further research will be aimed at determining the speed of the filtration procedure developed by the program in cases of random requests.

Keywords: *content filtering, software, adaptive complex system, sorting, content.*

Вступ. Розвиток інформаційно-комунікаційних технологій та імплементація мережі Інтернет майже в усі сфери життєдіяльності суспільства призводить як до позитивних, так і негативних наслідків. Серед останніх слід визначити несанкціоноване проникнення до користувачів небажаної чи забороненої інформації, яка не була затребувана. Тобто існує проблема розповсюдження в мережі Інтернет та проникнення до користувачів небажаного контенту (заборонених ресурсів, політичної та іншої пропаганди, торгівельної реклами, вірусних програм тощо), який може нанести користувачам як моральну, так і матеріальну шкоду (моральні травми, крадіжки інформації та коштів з рахунків тощо). Одним із шляхів вирішення цієї проблеми є фільтрація контенту, що поступає до користувача. Це призводить до формулювання науково-прикладного завдання щодо формування системи фільтрації контенту в мережі Інтернет, а також прикладного програмного забезпечення реалізації цього завдання, що обумовлює актуальність даного дослідження.

Аналіз. У практиці фільтрації контенту розроблена та функціонує низка продуктів, спрямованих на запобігання несанкціонованого доступу до ресурсів чи недопуску до небажаних ресурсів. Так, Воробієнко П., Каптур В. та Коляденко В. наводять теоретичні та прикладні механізми обмеження доступу до нецільових ресурсів мережі Інтернет [1]. Козевич О. та Комарова Н. розглядають фільтрацію контенту як технологію комплексного контролю Інтернет-ресурсів [2, 3]. Отт А. висвітлює сучасні тенденції в сфері контентної фільтрації [4]. Кузьміч А. визначає законодавчі аспекти регулювання методів фільтрації шкідливого інтернет-контенту [5]. У [6] досліджено низку існуючих програмних продуктів, що можуть бути застосовані для фільтрації контенту.

До продуктів, які мають широке розповсюдження серед користувачів мережевих послуг, можна віднести Content Protect, Cyber Patrol, Cyber Sentinel, Filter Pak, McAfee Parental Controls, Cyber Sitter, Net Nanny, Norton Parental Controls та інші. Наведені та інші програмні продукти розроблені на підставі інтелектуального алгоритму аналізу та можливості пропускати чи затримувати певні ресурси в режимі динамічної фільтрації, перелік яких визначає чи виробник, чи користувач. Втім, означені продукти недосконалі, оскільки більшість з них здійснюють попередження про небажаний контент замість заборони його потрапляння, мають низку чутливість та можуть біти налаштовані на ігнорування усіх чи деяких заборон. До того ж лише деякі з продуктів здійснюють аналіз об'єктів, більшість – забороняють чи пропускають контент без аналізу. Також до недоліків існуючих продуктів можна віднести відсутність у більшості з них функції фільтрації та моніторингу чатів, через які приховано також може потрапляти заборонений чи небажаний контент.

Також існують контент-фільтри для роутерів (наприклад, SkyDNS), які більш зручні у користуванні, оскільки дозволяють фільтрувати контент на всіх пристроях, що використовують доступ до Інтернету через роутер. Тобто відпадає необхідність контролювати та обслуговувати кожний пристрій у домашній чи корпоративній мережі, що суттєво економить час та кошти. Втім, ці фільтри потребують системного адміністрування та, як правило, мають суттєву вартість.

Наявність низки недоліків та проблемних аспектів в діяльності існуючих програмних продуктів обумовлює актуальність розробки такого програмного продукту, який би мав за мету подолати найбільш проблемні аспекти функціонування існуючих програм.

Автором у попередніх працях запропоновано (у співавторстві) бачення вирішення науково-прикладної задачі фільтрації контенту шляхом формування адаптивної комплексної системи фільтрації контенту в мережі Інтернет [7]. Втім, запропонована теоретико-прикладна розробка потребує відповідного програмного забезпечення, яке надає можливість практичного застосування розробленої системи фільтрації контенту.

Мета статті пов'язана із формуванням та реалізацією прикладного програмного забезпечення процесу адаптивної комплексної фільтрації контенту в мережі Інтернет.

Основна частина. Сьогодні вітчизняні провайдери активно розвивають власні програми, які базуються на низці найбільш розповсюджених механізмів фільтрації web-контенту в мережі Інтернет. Одним з значущих елементів впровадження та реалізації системи фільтрації контенту в мережі Інтернет є розробка, тестування та формування базових принципів, розробка відповідного алгоритму та програмної оболонки системи фільтрації контенту.

У попередніх роботах [7] сформовано теоретичне бачення процесу фільтрації контенту та прикладні засади роботи адаптивної комплексної системи фільтрації контенту (АКСФК) в мережі Інтернет, суть якого полягає у знаходженні системою потоку запитів від користувачів (груп користувачів), визначення типу запиту (URI, IP-адресу тощо), перевірці запиту на відповідність «чорному» та «білому» спискам та прийняття рішення щодо блокування або надання доступу до ресурсу. АКСФК має зворотний зв'язок з користувачем для інформування його про блокування запиту та/чи ресурсу, якщо система визначила запитуваний ресурс шкідливим (на основі збігу в «чорному списку»). Розроблена АКСФК відрізняється від існуючих тим, що завдяки постійному оновленню списків та профілю користувачів має більш викосу якість фільтрації контенту, а завдяки оптимізації процедури фільтрації (скасування зайвих процедур) зменшує час оброблення запитів користувачів.

Послідовність роботи АКСФК полягає в такому: в АКСФК надходить потік запитів від користувачів (груп користувачів), після чого система визначає тип запиту (URI, IP-адресу тощо). Далі здійснюється перевірка на наявність користувача в базі даних системи. В разі відсутності профілю конкретного

користувача відбувається його додавання в систему та починається фільтрація небажаного контенту за умовчанням.

Якщо ж профіль користувача присутній в базі даних, то для нього застосовується механізм, який підбирає найкращу (у даному випадку) послідовність використання способів і процедур фільтрації за умови найбільшої ймовірності спрацьовування системи. Вибір оптимальної послідовності стає можливим завдяки збереженню в базі даних інформації про засоби і процедури, які для кожного конкретного користувача (або групи користувачів) спрацьовували кращим чином.

Далі відбувається процедура фільтрації небажаного контенту шляхом застосування оптимальної послідовності засобів і процедур для кожного конкретного випадку, в результаті чого приймається рішення про блокування або ж надання доступу до запитуваного ресурсу. Одночасно в процесі роботи АКСФК відбувається постійне оновлення даних як про кожного користувача, так і про прийняті рішення щодо блокування або надання доступу. Ця інформація надалі використовується для прийняття рішення про вибір найкращої послідовності засобів і процедур фільтрації, що, як було вказано вище, дозволяє прискорити процес фільтрації.

Для реалізації прикладного програмного забезпечення функціонування адаптивної комплексної фільтрації контенту в мережі Інтернет відповідно до наведеної послідовності в якості платформи розробки було обрано програмну оболонку Debian, оскільки вона надає змогу провести подальше тестування та порівняння двох різних програм (розробленої і тієї, яку біде обрано для порівняння) в єдиній оболонці.

Розглянемо детально процес проходження процедури фільтрації контенту за допомогою розробленого програмного продукту, що має робочу назву «Проксі Блокер», що складається в виконанні послідовності наступних кроків.

1. Процес запуску програмного продукту супроводжується підключенням користувача до програмної оболонки Debian через віртуальну машину VirtualBox. Для запуску програмної оболонки необхідно здійснити її попереднє налаштування. Для цього у файлі (application.properties) відбувається налаштування необхідних параметрів роботи програми. Задля цього прописуються такі параметри:

application.proxy.test.port=8182 (підключення програми «Проксі Блокер»);

banlists.root.path=./banlists (шлях до бан-листів);

iphistory.persisting.interval.minutes=5 (команда, завдяки якій кожні 5 хвилин здійснюється збереження історії відвідування);

banlists.sorting.enabled=true (включення чи відключення процедури сортування, де **true** – сортування включено, а **false** – відповідно сортування вимкнено);

banlists.sorting.interval.secords=300 (час, протягом якого буде здійснюватися включення сортування бан-листів (у даному випадку сортування вмикається кожні 5 хвилин));

webclient.connect.timeout=5000 (час на з'єднання із сайтом, який запитує користувач);

webclient.read.timeout=10000 (час на зчитування контенту з сайту (якщо час зчитування буде перевищувати той, який передбачено програмою, запит буде скасовано).

2. Після налаштування файлу application.properties відбувається перехід до браузера. У меню налаштування браузера необхідно встановили параметр Proxy (**8182** для тестування програми «Проксі Блокер»).

3. Далі програма переходить до оболонки Debian, де відбувається запуск терміналу. У цьому терміналі здійснюється перехід в режим адміністратора за допомогою команди: ***su***. Після цього для отримання доступу до подальших налаштувань необхідно ввести пароль для адміністратора. У результаті отримання доступу на терміналі буде відображена команда: **root@debian:**

4. Наступним кроком є перехід до каталогу, де знаходиться програма, за допомогою команди **/cd.** Після цього потрібно здійснити запуск програми за допомогою команди **./start.sh.** У терміналі буде відображений процес запуску програмного продукту. Закінчення запуску програми буде супроводжуватися рядком: ***Starting FilteringApplication in 30,053 seconds*** (де 30,053 – час запуску програми).

5. Далі у браузері за допомогою наведених нижче команд потрібно здійснити обчислення часу обробки запитів, що надходять, у різних варіантах використання системи, створюючи при цьому імітацію її працездатності різними користувачами:

http://localhost:8080/statistics – (збирання статистичних даних щодо часу обробки запитів)

http://localhost:8080/?calls=10 – (10 випадкових запитів на випадковий бан-лист)

http://localhost:8080/?calls=10&banlistName=news (10 запитів на бан-лист під назвою ***news***)

Результати проведеної процедури можна побачити в терміналі, а також в загальному консольному меню, де наводиться уся статистична інформація. Для цього в браузері необхідно прописати команду ***http://localhost:8080/h2-console***

6. В процесі переходу в загальне консольне меню можна спостерігати появу різноманітної статистики, а саме:

BAN_LIST:

NAME **–** найменування бан-листів

IP_HISTORY:

IP – IP-адреса, з якої було здійснено запит
COUNTER – кількість запитів
BAN_LIST_NAME – ім'я бан-листа, на яке було здійснено запит

STATISTICS:
CALLS – кількість запитів
ELAPSED – час обробки запитів
SORTING_ENABLED – (True – сортування включено; False – сортування вимкнено)
START_DATE – (дата та час запуску запитів, що надходять)

IP_TO_BAN_LIST_ORDER:
IP – IP-адреси, з яких відбувалися запити

IP_TO_BAN_LIST_ORDER_BAN_LISTS:
IP_TO_BAN_LIST_ORDER_IP – IP-адреси, з яких відбувалися запити
BAN_LIST_NAME – найменування бан-листа, що відвідувався

7. Надалі процес виводу статистики в терміналі буде супроводжуватися наступними командами:
Calls to process: 100 (кількість запитів, що обробляються)
Time spend= [1,879025] (час обробки 100 запитів)

При включенні сортування в файлі **application.properties**, а також налаштування часового параметра її включення (***banlists.sorting.enabled=true; banlists.sorting.interval.secords=300***) в терміналі буде відбиватися включення сортування кожні 5 хвилин за допомогою наступних команд:
Sorting started
Sorting completed

8. Для того, щоб здійснити тестування іншого програмного продукту, необхідно зупинити програму, що працює. Для цього в терміналі прописується команда: **./stop.sh**
Результат зупинки буде відображений у терміналі: ***Stopping server***
Далі здійснюється налаштування файлу application.properties за процедурою, що описана вище.
9. При спробі входу через браузер на ресурс, що перебуває в бан-листах, в результаті роботи АКСФК буде виведена наступна інформація (рис. 1).

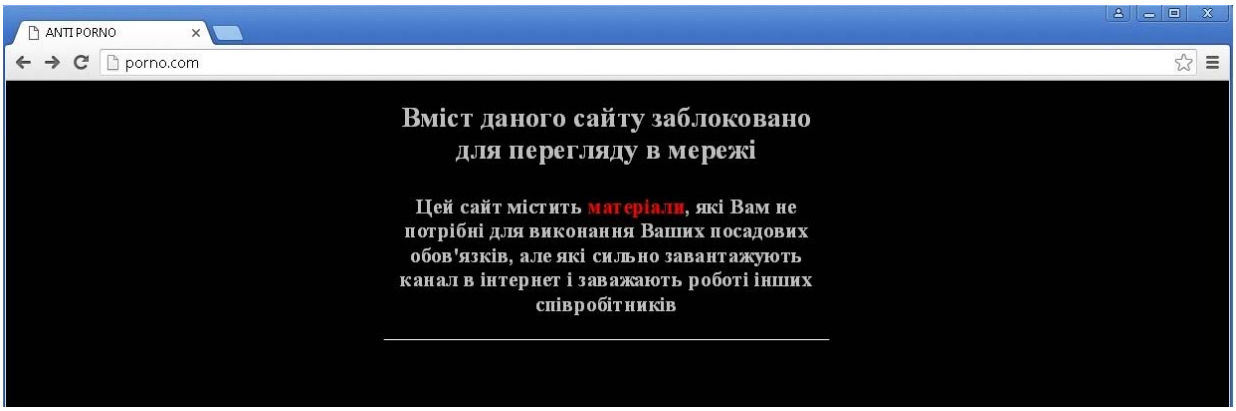

Рис. 1. Приклад повідомлення про заборону доступу до ресурсу (результат роботи АКСФК через програму «Проксі Блокер»)

На рис. 1 наведено приклад інформування користувача про заборону доступу до ресурсу, який він запитував. Текст, який наводитися на екрані, може бути змінено на будь-яке інше інформаційне повідомлення аналогічного змісту (тобто містити заборону доступу), а також наводитися будь-якою мовою, зручною для користувача та адміністратора ресурсу. Ці параметри змінюються в процесі налаштування програми «Проксі Блокер». Таке або аналогічне повідомлення буде застосовуватися до усіх заборонених ресурсів, на які буде здійснюватися спроба доступу від користувача.

В якості тестування розробленого програмного продукту було проведено експеримент, який полягав у розташуванні списку заборонених ресурсів на перші позиції «Проксі блокеру» при однаковій кількості запитів (2000 запитів) із інтервалом сортування у п'ять хвилин.

Параметри сортування сформовані в такий спосіб:
banlists.sorting.enabled=true
banlists.sorting.interval.secords=300)

Через ці команди здійснюється включення сортування. Інтервал включення сортування з подальшим перерозподілом досліджуваного бан-листа становить 5 хвилин. Кількість запитів, що надходять,

як вже визначалося вище, фіксуємо на числі 2000 запитів наступною командою:

http://localhost:8080/?calls=2000&banlistName=webtv (2000 запитів на бан-лист з іменем webtv).

В результаті проведення 12 спроб отримано результати, які свідчать, що за цих умов час обробки запиту скорочується при зростанні інтервалу сортування (табл. 1).

Таблиця 1

Результати експерименту з підняття списку заборонених ресурсів на перші позиції «Проксі Блокеру» при однаковій кількості запитів (2000 запитів)

Інтервал сортування, хвилини	Час обробки запиту, секунди	Скорочення часу обробки запиту (порівняно із попередніми значеннями), %
5	46,387	-
10	27,329	1,69
15	25,532	1,07
20	23,584	1,08
25	23,344	1,01
30	21,444	1,08
35	21,419	1,001
40	20,756	1,03
45	3,409	6,1
50	2,892	1,17
55	2,693	1,07
60	2,595	1,03

Джерело: власні розрахунки

Дані табл. 1 свідчать, що чим рідше здійснюється сортування, тим швидше відбувається обробка запиту і навпаки. Це пояснюється навантаженням на сервіси. Втім, не визначено прямої лінійної залежності часу обробки запиту від інтервалу сортування. Як видно, між інтервалом сортування 5 та 10 хвилин спостерігається стрибок скорочення часу обробки запиту майже на 70%, а в інтервалі 20 та 45 хвилин стрибок скорочення часу обробки запиту більше ніж у 6 разів (із 20,75 до 3,4 секунд).

Означені стрибки мають місце тому, що відбувається перерозподіл положення бан-листа (із заданим інтервалом) на вищі позиції до моменту, поки досліджуваний бан-лист не буде знаходитися на першому місці у списку. Так, у зазначених інтервалах часу між 5–10 і 40–45 хвилинами сортування, досліджуваний бан-лист перерозподілиться на декілька (чотири чи п’ять) позицій уперед.

Сформуємо математичну модель залежності часу обробки запиту від інтервалу сортування, пролонгація якої надасть змогу визначити подальшу динаміку змін часу обробки запиту при зростанні (скороченні, зміні кроку) інтервалу сортування. Моделювання здійснимо методом апроксимації за допомогою стандартного програмного забезпечення Microcoft Office Excel (рис. 2).

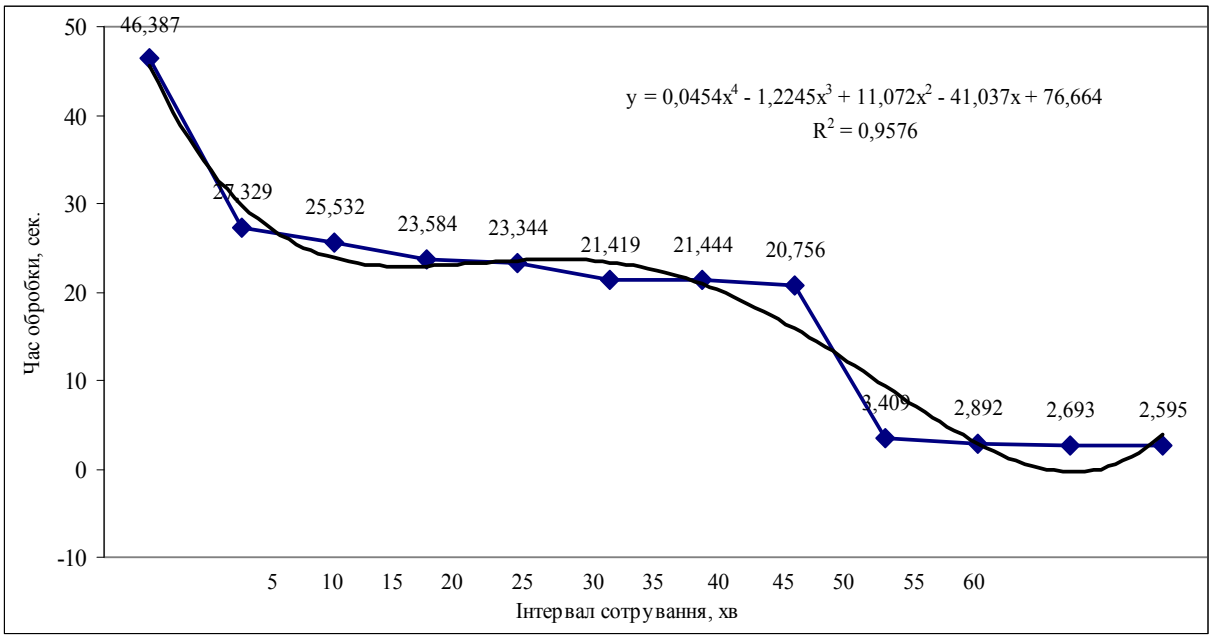

Рис. 2. Залежність часу обробки запиту від інтервалу сортування

Результати моделювання свідчать, що із вірогідністю $R^2 = 0,9576$ модель залежності часу обробки запиту від інтервалу сортування має поліноміальну залежність четвертого ступеню виду:

$$y = 0{,}0454x^4 - 1{,}2245x^3 + 11{,}072x^2 - 41{,}037x + 76{,}664 \quad (1)$$

На графіку наочно видно, що побудована за формулою (1) крива достатньо адекватно відбиває реальні дані, отримані в результаті експерименту. Також, відповідно до рис. 2, слід визначити, що в разі, коли досліджуваний бан-лист на початку знаходиться на останньому місці, час його обробки завжди буде максимальний (в межах даних, отриманих в експерименті).

Також в роботі проведено порівняння розробленого програмного продукту «Проксі Блокер» із існуючими. Для порівняння обрано одну з найбільш розповсюджених програм Rejik+Squid [8], яка є у вільному доступі та дозволяє проводити маніпуляції із операторами.

Порівнюючи роботу запропонованої АКСФК та програмного продукту із існуючими (тобто обраною програмою Rejik+Squid), в ході тестування отримано низку результатів експериментів (табл. 2).

Таблиця 2

Результати експериментів із порівнянням швидкості обробки запитів розробленого програмного продукту «Проксі Блокер» та існуючих продуктів (на прикладні програми Rejik+Squid)

Експеримент	Середній час обробки запитів* (секунд), при фіксованій кількості запитів, що надходять					
	Кількість запитів, що надходять					
	1	100	1000	10000	100000	500000
Імітація запитів, що надходять на останній бан-лист в списках, що використовуються у програмному продукті Rejik+Squid	0,00328	0,6093	3,66996	32,549	308,179	1588,32
Імітація запитів, що надходять на останній бан-лист в списках, що використовуються у програмному продукті «Проксі Блокер»	0,0053	0,577	3,652	32,316	361,149	1548,67
Абсолютна зміна часу обробки у «Проксі Блокер» порівняно із Rejik+Squid, +/-	+0,00212	-0,03234	-0,0179	-0,233	+52,97	-39,65
Відносна зміна часу обробки у «Проксі Блокер» порівняно із Rejik+Squid, %	+66%	-5,3%	-0,3%	-0,7%	+17,1%	-2,5%
Імітація запитів, що надходять на перший бан-лист в списках в результаті повного сортування на базі програмного продукту «Проксі Блокер»	0,00206	0,1863	1,6022	14,856	124,132	620,21
Відносна зміна часу обробки «Проксі Блокер» у випадку, коли запит потрапляє на перший бан-лист в списках	157%	209%	127%	117%	190%	149%

Джерело: отримано автором

** В ході експериментів здійснено по десять пробних тестів роботи програмного продукт. Середній час обробки запитів (секунд) отримано як проста середньоарифметична.*

Так, можна побачити, що при запиті на останній лист списку заборонених ресурсів, які використовуються у програмному продукті Rejik+Squid, середнє значення часу сортування природно збільшується при збільшенні кількості запитів. Втім, аналогічні показники для програми «Проксі Блокер», що розроблена автором, у більшості проведених експериментів менші за результати експериментів з існуючими продуктами. Зокрема, лише у випадку одиничного запиту час обробки продуктом Rejik+Squid на 66% кращий ніж при використанні програмного продукту «Проксі Блокер». Також на 17,1% швидше відбувається обробка запитів у разі, коли їх кількість становить 100000. В інших випадках розроблений програмний продукт «Проксі Блокер» показує кращій середній час обробки запитів, який в середньому на 2,5% менший за аналогічні показники продукту Rejik+Squid.

Також визначено, що у випадку, коли запит потрапляє на перший бан-лист в списках в результаті повного сортування на базі програмного продукту «Проксі Блокер», середній час обробки запиту скорочується порівняно із тим, коли запит потрапляє на останній бан-лист в списках за обома програмними продуктами. Зокрема, час обробки запитів «Проксі Блокер» у випадку, коли запит потрапляє на перший бан-лист в списках (порівняно із тим, коли запит потрапляє на останній лист тієї ж програми) наведено в останньому рядку табл. 2. У середньому час скорочується на 158,16%.

Таким чином, розроблений програмний продукт, на відміну від існуючих, показує кращі за часом результати фільтрації контенту, що свідчить на користь його застосування у АКСФК.

Висновки

Існуюча проблема проникнення до користувачів через мережу Інтернет небажаного чи забороненого контенту породжує необхідність формування систем захисту від несанкціонованого чи забороненого втручання в особистий інформаційний простір користувачів та в діяльність організацій. Одним зі шляхів вирішення цієї проблеми є формування систем фільтрації контенту та прикладного програмного забезпечення цієї фільтрації.

В роботі надано основні теоретичні засади розробленої адаптивної комплексної системи фільтрації контенту (АКСФК), наведено процедуру проходження фільтрації в системі та запропоновано розроблений автором програмний продукт «Проксі Блокер», який реалізує відповідний алгоритм АКСФК. Наведено основні процедури та оператори програми, проведено її тестування та сформовано математичну модель залежності часу обробки запиту від інтервалу сортування при однаковій кількості запитів. Встановлено, що не існує прямої лінійної залежності часу обробки запиту від інтервалу сортування. При перерозподілі положення бан-листа (із заданим інтервалом) на позиції спостерігаються стрибки скорочення часу обробки запиту (в деяких випадках більш, ніж у шість разів – із 20,75 до 3,4 секунд).

Проведено порівняння розробленого програмного продукту «Проксі Блокер» з іншими (на прикладі застосування програмного продукту Rejik+Squid), в ході чого встановлено, що середній час сортування у розробленому програмному продукті «Проксі Блокер» за результатами експериментів на 2,5% швидше. У випадку, коли запит потрапляє на перший бан-лист в списках в результаті повного сортування на базі програмного продукту «Проксі Блокер», середній час обробки запиту скорочується у середньому на 158,16% порівняно із тим, коли запит потрапляє на останній бан-лист в списках за обома програмними продуктами.

Таким чином, розроблений програмний продукт «Проксі Блокер», на відміну від існуючих, показує кращі за часом результати фільтрації контенту, що свідчить на користь його застосування у АКСФК.

Подальші дослідження будуть спрямовані на визначення швидкості проведення процедури фільтрації розробленою програмою у випадках рандомних запитів.

Література

1. Воробієнко П.П. Єдина система обмеження доступу до нецільових ресурсів мережі Інтернет в освітніх закладах України / П.П. Воробієнко, В.А. Каптур, В.А. Коляденко, В.О. Самодід // Комп'ютер у школі та сім'ї. – 2009. – № 8. – С. 30–34.

2. Козевич О. П. Контентна фільтрація – технологія комплексного контролю Інтернет-ресурсів. Основні підходи і проблеми / О. П. Козевич // Вісник Національного університету «Львівська політехніка». – 2013. – № 774: Автоматика, вимірювання та керування. – С. 138–145.

3. Комарова Н. Технология комплексной безопасности интернет-контента / Н. Комарова // Information Security. – 2008. – № 4. – С. 32–33.

4. Отт А. Современные тенденции в области контентной фильтрации / А. Отт // Информационный бюллетень “JET INFO”. – 2012. – С. 3–23.

5. Кузьміч А. Законодавче регулювання та способи фільтрації шкідливого Інтернет-контенту: міжнародний досвід [Електронний ресурс]. – Режим доступу : http://telpu.com.ua/archives/1872

6. Программы контроля Internet-контента [Електронний ресурс]. – Режим доступу : https://itc.ua/articles/programmy_kontrolya_internet-kontenta_18021/

7. Каптур В.А. Метод адаптивної оцінки URL в комплексних системах фільтрації контенту / В.А. Каптур, О.А. Князєв // Наукові праці ОНАЗ. – Одеса, 2016. – № 1. – С. 35–45.

8. Блокування за допомогою Rejik+Squid [Електронний ресурс]. – Режим доступу: Rejik.ru

References

1. Vorobiyenko P. Uniform system of restriction of access to no-purpose resources of a network the Internet in educational institutions of Ukraine / P. Vorobiyenko, V. Kaptur, V. Koliadenko, V. Samoid // Computer in a family and school. – 2009. – № 8. – P. 30-34.

2. Kozevich O. Content a filtration - technology of complex control of Internet resources. The basic approaches and problems / O. Kozevich // Visnyk of Lviv university. - 2013. - № 774: Series of automation, measurement and control. - P. 038-145.

3. Komarova N. Technology of complex safety the content Internet / Komarova N. // “Information Security”. – 2008. – № 4. – P. 32–33.

4. Ott A. Modern lines in area контентной filtrations / A. Ott // The Newsletter “JET INFO”. – 2012. – P. 3–23.

5. Kuzmich A. Legislative regulation and ways of a filtration of a content harmful the Internet: the international [Electron resource]. – Access mode: http://telpu.com.ua/archives/1872

6. Programs of control content of Internet [Electronic resource]. – Access mode: https://itc.ua/articles/programmy_kontrolya_internet-kontenta_18021/

7. Kaptur V. Method of adaptive estimation URL in complex systems of a filtration of a content / V. Kaptur, O. Kniaziev // Proceedings of the O. S. Popov ONAT, 2016. – № 1. – P. 35-45.

8. Blocking on a basis Rejik+Squid [Electron resource]. – Access mode: Rejik.ru

Рецензія/Peer review : 3.5.2019 р. Надрукована/Printed : 2.6.2019 р.

Прорецензовано редакційною колегією