УДК 04.65

COMPARISON OF THE METHOD OF PROCESSING AND ANALYSIS OF JSON LOG FILES WITH EXISTING SOLUTIONS

V. Bratskyi, E. Myakshylo
National University of Food Technologies

Key words:
Database
Logs
Analyzing log files
SQL
NoSQL
JSON
Relational database
Norelational database
MongoDB
Log Parser 2.2
Logstash
Web Log Exploler
Web Log Storming

Article history:
Received 04.05.2018
Received in revised form 20.05.2018
Accepted 11.06.2018

Corresponding author:
V. Bratskyi
E-mail:
vadymbratskyi@gmail.com

ABSTRACT

Log files are often and completely unfairly deprived of the attention of developers. And when programmers need to process log files by parser, sometimes from several dozen servers at a time, the task must be made by system administrators and it takes a lot of time and effort.

So when there are problems and you need to find an error in the log files of a service, it all depends on the professionalism of the administrator and the knowledge of programmers about things they must look for. And again, whether the logrotate worked out and if it did not delete the old data. In these situations, actively developing Logstash helps. On the web site https://habrahabr.ru/ there are quite a few articles about its installing and forcing to work on one server. Several similar systems (Log Parser 2.2, Logstash, Web Log Explorer, Web Log Storming), which are used as log-processing servers, are analyzed in the article. Unfortunately, these systems can not be used as agents. After several assemblies of log files from a plurality of servers, authors of the article managed to develop a log-file collection system, the description of which is proposed. Existing solutions are also reviewed and a comparative analysis of log file processing systems is conducted according to the following criteria: log file format, storage database, analysis capabilities.

As a result, it was found that log files are a very powerful tool for analyzing the functioning of client applications of a distributed system on a server. There are also many solutions for analyzing these documents today. But for each system it's quite difficult to find the right analyzer that would easily handle the documents and meet all the requirements. Therefore, we considered four systems that had been designed to solve similar problems, as well as a system based on them that is capable to:

- carry out rapid parsing of log files according to the specified criteria;
- store the analyzed data in the non-relational database MongoDB, where it is possible to analyze data from log files using NoSQL queries;
- store files where errors in the database are detected using GridFS technology;
- based on the analyzed data, automatically suggest a way to solve the problem.

DOI: 10.24263/2225-2924-2018-24-3-3

ПОРІВНЯННЯ МЕТОДУ ОБРОБКИ І АНАЛІЗУ LOG-ФАЙЛІВ У ФОРМАТІ JSON З ІСНУЮЧИМИ РІШЕННЯМИ

В.О. Брацький, О.М. М'якшило
Національний університет харчових технологій

Log-файли часто і абсолютно незаслужено обділені увагою розробників. І коли програмістам необхідно обробити парсером log-файли, іноді з декількох десятків серверів одночасно, завдання покладається на системних адміністраторів і забирає в них багато часу й сил.

У разі виникнення проблем і пошуку помилок у log-файлах сервісу все залежить від того, наскільки професійним є адміністратор і чи знають програмісти, що шукати. А ще від того, чи відпрацював logrotate і чи не видалив він старі дані.

У таких ситуаціях ефективно допомагає Logstash, що активно розвивається. У мережі https://habrahabr.ru/ є досить багато статей про те, як його встановити і змусити працювати на одному сервері. У статті проаналізовано декілька схожих систем (Log Parser 2.2, Logstash, Web Log Exploler, Web Log Storming), які використовуються як сервери обробки log-файлів. На жаль, ці системи не можна використовувати як агенти. Після кількох випадків збірки log-файлів з множини серверів, авторам статті вдалося розробити систему збирання log-файлів, опис якої пропонується. Також розглянуто існуючі рішення і проведено порівняльний аналіз систем обробки log-файлів за такими критеріями: формат log-файлів, база даних збереження, можливості аналізу.

У результаті з'ясовано, що log-файли є вельми потужним засобом для аналізу функціонування клієнтських додатків розподіленої системи на сервері. Також на сьогодні існує досить багато рішень для аналізу цих документів. Але для кожної системи досить важко підібрати потрібний аналізатор, який би легко справлявся з документами і відповідав би всім вимогам. Тому розглянуто чотири системи, які розроблені для вирішення подібних завдань. У результаті розроблено систему, яка здатна:

- проводити швидкий парсинг лог-фалів за заданими критеріями;

- зберігати проаналізовані дані в нереляційній базуйданих MongoDB, в якій можна проводити аналіз даних з лог-файлів за допомогою NoSQL запитів;

- зберігати файли, де виявлені помилки у базі даних, за допомогою технології GridFS;

- на основі проаналізованих даних автоматично запропонувати шлях вирішення проблеми.

Ключові слова: *бази даних, SQL, NoSQL, JSON, BSON, реляційні СКБД, нереляційні СКБД, MongoDB, Log Parser 2.2, Logstash, Web Log Exploler, Web Log Storming.*

Постановка проблеми. Для багатьох розробників програмних продуктів з огляду на свої обов'язки доводиться багато уваги приділяти log-файлам. Це і участь у виробленні правил і політик збору, зберігання, використання log-файлів, розбір різних інцидентів і виявлення аномалій. За добу програми, сервіси та сервери генерують дуже велику кількість log-файлів і потреба у видобутку даних з log-файлів постійно зростає. Натепер існує багато рішень для обробки log-файлів. Нам довелось попрацювати з комерційними лог-менеджмент продуктами типу Logstash., Log Parser 2.2. Але в кожному з них є свої недоліки, такі як: незрозумілий інтерфейс, мала швидкість обробки, незручний формат вихідних даних, використання бази даних без додаткових можливостей управління. Тому потрібно розробити алгоритм і програмний продукт для обробки запису в базу даних log-файлів та проаналізувати їх з метою діагностування збоїв і відмов програмного забезпечення в розподіленій системі [9].

Метою дослідження є порівняння існуючих рішень з обробки log-файлів із запропонованим авторами, виявлення недоліків в існуючих рішеннях і переваг у запропонованій методиці.

Викладення основних результатів дослідження. Нині є досить багато програмних продуктів для аналізу файлів формату *.log і кожен з них має свої особливості. Тож доцільно розглянути декілька таких програмних продуктів, надати їх повну характеристику, визначити, для вирішення яких завдань вони краще підходять. Розглянуті рішення порівнюються із запропонованим авторами програмним продуктом, який призначений для аналізу і парсингу log-файлів.

Log Parser 2.2 — це дуже потужний універсальний засіб, що надає універсальний запит доступу до текстові дані, такі як файли журналу, файли XML і CSV-файли, а також основних джерел даних операційній системі Microsoft Windows (журнал подій, реєстр, файлова система і служба каталогів Active Directory).

Log Parser був розроблений більше десяти років тому. Хоча автори намагалися працювати з іншими аналогічними інструментами, все ж доводилося повертатися до Log Parser, який більше відповідає потребам завдяки широкій підтримці типів файлів і гнучкості, наданій SQL-подібною мовою.

Log Parser — це утиліта командного рядка. Дійсно, з'явившись у пакеті IIS Resource Kit на рубежі століть, Log Parser призначалася для розбору журнальних файлів IIS з метою аналізу трафіку Web-сайтів, збору статистики відвідувань, виявлення спроб вторгнення тощо. Однак до 2005 р., зазнавши кількох оновлень (до номера 2.2.10), вона перетворилася в більш потужний інструмент, здатний отримувати інформацію з 20 типів джерел. Log Parser володіє широкими функціями розбору інформації і подання її в зручному для подальшої роботи вигляді (підтримуються 10 різних форматів).

Для утиліти передаються SQL-запит (його можна також завантажувати із заздалегідь підготовленого файлу) і параметри, що описують формати джерела й одержувача інформації. Якщо виконати команду logparser «select * from Security where EventID in (528; 540) order by TimeGenerated

DESC», то в результаті в консольне вікно буде виведений список усіх подій із системного журналу безпеки з кодами 528 і 540, відповідним успішним локальним і мережевим підключенням користувачів до ресурсів комп'ютера.

Говорячи про джерела, з якими може працювати Log Parser, варто виділити кілька найбільш важливих:

- журнали Web/FTP-сервісів IIS різних форматів, а також вельми специфічні і в той же час дуже корисні журнали помилок служб http.sys і URLScan, дослідження яких дає змогу швидко реагувати на спроби злому IIS;

- журнали подій Windows, не тільки стандартні (Application, Security, System), але й ті, що додаються штатними сервісами Windows (DNS, Active Directory) і різними додатками (наприклад, Microsoft Virtual Server). Працювати з цими джерелами Log Parser може як локально, так і дистанційно (певна річ, якщо користувач має відповідні права);

- як джерела виступають не тільки традиційні журнали. Log Parser вміє працювати з файлової системою, реєстром і навіть службою каталогів Active Directory, отримуючи таким чином доступ до інформації про різні об'єктах, скажімо, до налаштувань того ж IIS;

- за допомогою Log Parser можна досліджувати навіть Web-сайт, що містить RSS. У цьому разі достатньо лише вказати потрібний URL;

- підтримуються також журнали в широкому розумінні цього слова, тобто CSV, TSV, XML та інші структуровані текстові файли. У більшості випадків Log Parser здатний автоматично проаналізувати структуру даних і, відповідно, обробити запит.

Для вибірки даних у Log Parser застосовується мова запитів SQL, що дає змогу всім, хто знайомий з ним, максимально задіяти колишні навички, і при цьому жодним чином не обмежує дії фахівців. Більш того, в діалект SQL введені різні додаткові функції, які максимально враховують специфіку застосування Log Parser [1].

Під час роботи виявлені такі незначні недоліки:

1. Інтерфейс командного рядка не дуже інтуїтивно зрозумілий, тому знайти його можна після встановлення та запуску інструменту.

2. При аналізі великих файлів користувач може стикнутися з проблемою «Відмова доступу». Програма рекомендує, щоб клієнт використовував інший інструмент, який називається Filemon. Після завантаження Filemon і відтворення проблеми, якщо це проблема з ресурсами ACL, інструмент Filemon зможе виявити помилку. Клієнт може відправити користувачу збережений файл журналу Filemon. Однак розмір відправленого файлу величезний (Filemon записує велику кількість даних). Блокнот буде зависати і повільно знаходити рядки «Access Denied» у файлі журналу. При цьому Microsoft Office Excel відмовиться відкрити файл повністю.

Logstash — це знаряддя для збору, фільтрації та нормалізації log-файлів, яке є безкоштовним. Це open source додаток, що розподіляє дані по багатьох системах у різних форматах. Logstash підтримує різноманітні входи, які залучають події з безлічі загальних джерел в один і той же час. Легко обробляє

дані з журналів, показників, веб-програм, збережених даних і різних сервісів AWS. Усе це виконується безперервним потоковим способом.

Logstash відноситься до класу інструментів, для яких достатньо створити декілька комбінацій входів, виходів і задіяти фільтри. Цей програмний продукт пропонує попередньо конфігуровані фільтри, тому можна легко перетворювати загальні типи даних, індексувати їх в Elasticsearch і починати запити без необхідності створювати конвеєри перетворення даних. Logstash прекрасно працює з Elasticsearch, але з джерелами та з отримувачами даних може бути величезний набір сервісів, починаючи з черг повідомлень і закінчуючи TCP сокетами. Logstash аналізує stack trace (трасування коду на мові java) і абсолютно різні log-файли, не має проблем з гнучкістю, але його потрібно адаптувати до своїх вимог. Загалом, Logstash — досить продуктивний програмний продукт, який задовольняє потреби користувачів.

Але в цій системі пропущений один елемент — користувальницький інтерфейс, через що аналізувати log-файли з командного рядка не найбільш продуктивний підхід [3].

Під час роботи виявлені такі незначні недоліки:

1. Для роботи потрібно, щоб було обов'язково встановлено пакет Java на всіх серверах.

2. Високі вимоги до пам'яті, потенційно може впливати на роботу інших сервісів.

3. Досить складний у налаштуванні.

4. Незручний консольний інтерфейс.

Web Log Explorer — це інтерактивний настільний аналізатор журналів для Windows. Ця програма створює динамічні звіти миттєво, що дає змогу глибоко аналізувати дані. Натиснувши правою кнопкою миші рядок у будь-якому звіті (сторінку, хост, користувача, пошукову систему тощо), можна вибрати зі списку суб-звітів звіт, доступний для цього елемента. Наприклад, у звіті про переглядах сторінок можна вибрати певну сторінку й отримати список своїх відвідувачів, їхні шляхи по всьому сайту, країни і міста, в яких є ці відвідувачі, сайти, з яких надсилають повідомлення, ключові слова, яке ключове слово, використане в пошукових системах, щоб знайти цю сторінку тощо.

Web Log Explorer також дає змогу фільтрувати відвідування роботами пошукових систем. Можна перемикати програму в режим «Всі запити», «Без павуків» і «Тільки павуки» одним натисканням кнопки.

Web Log Explorer підтримує більше 43 форматів файлів журналу. Причому лог-файли можуть перебувати в тому вигляді, як вони зберігаються на сайті: в ZIP- і GZ-архіві. Він був протестований з усіма популярними веб-серверами, мультимедійними службами, проксі-серверами, брандмауерами тощо. Певна річ, Web Log Explorer може автоматично розпізнавати формати файлів журналу, витягувати стислі файли журналів, обробляти декілька файлів журналів і завантажувати журнали з різних джерел: локальний або мережевий шлях, Web, FTP або бази даних через ODBC.

На сайті розробника є ще один лог-аналізатор — Web Log Suite 3.43. І на відміну від Web Log Explorer, він генерує окремі HTML-звіти, які може самостійно викласти на сервер або надіслати на задану адресу електронної пошти [6].

Під час роботи виявлені такі незначні недоліки:

1. Занадто простий інтерфейс звітів, який варто було б розширити і додати більше фільтрів.

2. Витягує дані з бази даних, а проаналізований результат у базі даних не зберігає і тому не має можливості працювати з даними на рівні бази даних, щоб самому конструювати запити.

Web Log Storming є інтерактивним (на настільній основі) аналізатором веб-журналів для Windows. Дає абсолютно нову концепцію сайту статистики, явно відрізняється від будь-якого іншого програмного забезпечення журналу веб-аналітики. За допомогою Web Log Storming можна переглядати статистику, заглиблюючись у деталі аж до сесії окремих відвідувачів. Можна перевірити шаблон індивідуальної поведінки відвідувача і як вона вписується у цілі користувача. Швидке використання фільтрів (параметрів) дає можливість легко зосередитися на конкретному сегменті. Немає жодних обмежень: можна об'єднати всі звіти з будь-яким фільтром, щоб дізнатися про ефективність реклами або поведінки. Слід просто натиснути на будь-який елемент звіту, щоб переглянути список сесій по цьому пункту, або натиснути на будь-якій сесії, щоб переглянути всі деталі: IP-адреса, домен, країна, регіон, місто, агент користувача, відвідані сторінки, пошук фрази тощо.

Web Log Storming робить набагато більше, ніж просто створення спільних звітів. Він відображає детальну статистику веб-сайту з інтерактивною графікою і звітами. Повний і докладний лог-аналіз діяльності від кожного відвідувача на сайт можна отримати, клацнувши однією кнопкою миші. За допомогою цього програмного забезпечення веб-сайт статистики легко відстежує сесії, перегляди сторінок, завантаження або будь-які інші метрики, які є найбільш важливим для кожного користувача. Тобто можна побачити, які посилання, пошукові системи і ключові слова були використані, щоб привести відвідувачів на сайт.

Поведінку веб-сайту з верхніх сторінок входу та виходу до шляху, яким користувачі слідують, можна проаналізувати, щоб дізнатися, з яких країн на приходять відвідувачі, які операційні системи і браузери вони використовують. Також можна дізнатися про пропускну здатність каналу, що використовується, і скільки часу проводять користувачі на сайті [7].

Під час роботи виявлені такі незначні недоліки: не використовує жодну базу даних, тому збереження здійснюється у файловій системі комп'ютера. Файли займатимуть велику пам'ять на диску, а також потребують значного часу для обробки.

Запропонований авторами парсер log-файлів. Сучасні системи аналізу log-файлів дають змогу маніпулювати з даними практично різними способами, але мають ряд суттєвих недоліків, які були зазначені в чотирьох продуктах, розглянутих вище.

Кожен розробник так чи інакше зіткнеться з однією важливою проблемою, а саме: з величезною кількістю log-файлів, які згенерує сам програмний продукт і йому, наприклад, доведеться перебирати за датою увесь файл і шукати причину поломки. Тому авторами було вирішено написати програмний

продукт, який допоможе швидше й ефективніше аналізувати log-файли, що є необхідним як для розробників, так і для технічної підтримки. Перш за все було прийняте рішення зберігати log-файли в одному сховищі і записувати в форматі json, за яким буде дуже легко їх проаналізувати і помістити в базу даних. У нашому випадку парсер написаний на C#, дані заносяться до MongoDB. Алгоритм простий: створюємо регулярні вирази і, пробігаючи по рядках файлу, шукаємо збіг за допомогою Regex. Створюються два динамічні масиви, один тимчасовий, який буде зберігати неповні об'єкти, а інший — повні об'єкти, які в кінці будуть потрапляти до бази даних. Це все відбувається асинхронно, поки в одному з потоків відбувається парсинг, то в іншому відбувається вставка в базу даних MongoDb. Кожен об'єкт у форматі json містить інформацію про запит і відповідь із сервера, дату, коли це відбулося, і id посилання на сам файл, який також знаходиться у базі даних. За таким алгоритмом за секунду обробляється 1 400 рядків і створюється 344 повних об'єкти, які готові для збереження у базі даних. Додатково програмний продукт здатний не тільки парсити файли, а й провести аналіз існуючих даних у базі і запропонувати можливі варіанти створення запитів для користувача, йому залишається тільки ввести дані, за яким буде працювати запит для вибірки даних з бази даних, де збережені проаналізовані log-файли.

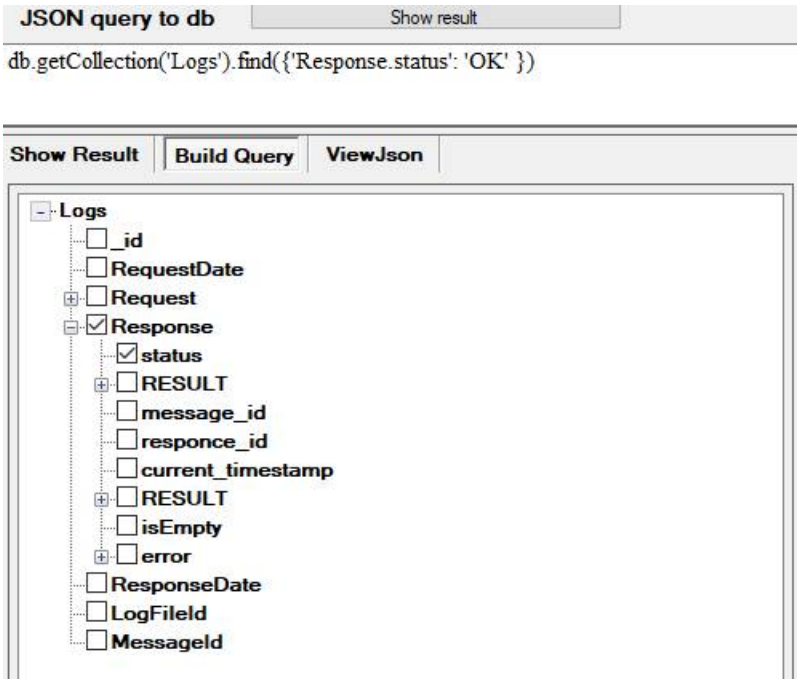

Рис. 1. Інтерфейс побудови запиту

За допомогою запиту користувач зможе легко прочитати лог-об'єкт у зручному вигляді дерева і провести аналіз за результатами запиту.

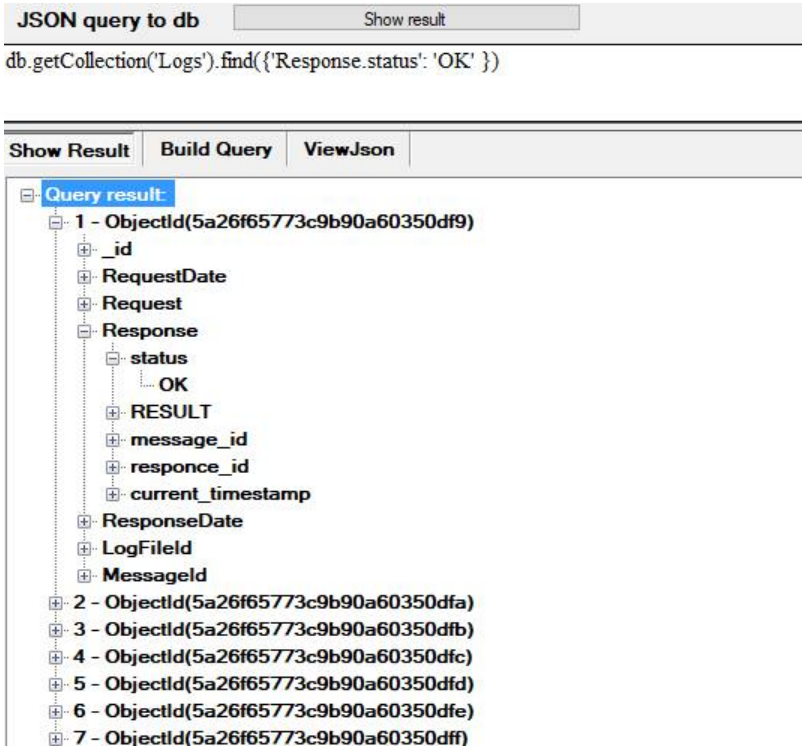

Рис. 2. Інтерфейс результату запиту

У цій програмі є ще одна велика і важлива перевага. Під час парсингу log-файлів виділяються і зберігаються об'єкти, які маю статус error, в окрему таблицю, а також інформація про помилку — це код помилки, повідомлення тощо. Потім за допомогою цих даних з легкістю можна скласти запит і виявити причину відмови або збою. В подальшому, проаналізувавши дані про помилки, в програмі передбачена можливість створення таблиці з кодами помилок і їх рішенням, щоб програма могла самостійно проаналізувати та запропонувати варіанти вирішення виявленої проблеми для користувача за прийнятими log-файлами.

Висновок

Як бачимо, log-файли є вельми потужним засобом для аналізу функціонування клієнтських додатків розподіленої системи на сервері. Також на сьогодні існує досить багато рішень для аналізу цих документів. Але для кожної системи досить важко підібрати потрібний аналізатор, який би легко справлявся з документами і відповідав би всім вимогам. Тому розглянуто чотири системи, які розроблені для вирішення подібних завдань, а також на їх основі розроблена система, яка здатна:

1) проводити швидкий парсинг лог-фалів за заданими критеріями;

2) зберігати проаналізовані дані в нереляційній базі даних MongoDB, в якій можна проводити аналіз даних з лог-файлів за допомогою NoSQL запитів;

3) зберігати файли, де виявлені помилки у базі даних, за допомогою технології GridFS;

4) на основі проаналізованих даних автоматично запропонувати шлях вирішення проблеми.

Зрозуміло, що цей програмний продукт чітко працює для визначених log-файлів, тому для вирішення інших проблем його потрібно удосконалювати.

Література

1. Microsoft Log Parser [Електроний ресурс]. — Режим доступу : https://itc.ua/articles/-microsoft_log_parser_ne_zhurnalom_edinym_21649/.

2. Средство Log parses 2.2 и ASP.NET [Електроний ресурс]. — Режим доступу : https://support.microsoft.com/ru-ru/help/910447/log-parser-2-2-and-asp-net

3. Обрабатываем логи в logstash [Електроний ресурс]. — Режим доступу : https://-dotsandbrackets.com/processing-logs-logstash-ru/.

4. Собираем и анализируем логи с помощью Lumberjack+Logstash+Elasticsearch+RabbitMQ [Електроний ресурс]. — Режим доступу : https://habrahabr.ru/company/maxifier/blog/216201./

5. Ещё одна система логирования, теперь на ElasticSearch, Logstash, Kibana и Prometheus [Електроний ресурс]. — Режим доступу : https://habrahabr.ru/company/2gis/blog/-329128/.

6. Web Log Explorer Lite - Free Log Analyzer [Електроний ресурс]. — Режим доступу : https://www.exacttrend.com/WebLogExplorer/.

7. Web Log Storming Analyzer [Електроний ресурс]. — Режим доступу : https://-www.weblogstorming.com/features.html.

8. *Брацький В.О.* Дослідження особливостей застосування реляційних і нереляційних баз даних на прикладі SQL Server та MongoDb / В.О. Брацький, О.М. М'якшило // Наукові праці Національно університету харчових технологій. — 2016. — Том 22, № 5. — С. 15—23.

9. *Брацький В.О.* Дослідження та розробка методу обробки log-файлів у розподіленій інформаційній системі з використанням нереляційної бази даних MongoDB / В.О. Брацький// Наукові праці Національно університету харчових технологій. — 2018. — Том 24, № 1. — С. 17—25.