УДК 004.75

А. М. Волокита

ЗАСІБ ВИЯВЛЕННЯ АНОМАЛІЙ ПОКАЗНИКІВ ПРОДУКТИВНОСТІ РОЗПОДІЛЕНИХ КОМП’ЮТЕРНИХ СИСТЕМ

Анотація: В даній роботі розглянуто питання виявлення аномалій показників продуктивності розподілених комп’ютерних систем. Проведено порівняльний аналіз існуючих програмних рішень для виявлення аномалій, запропоновано засіб виявлення аномалій продуктивності, який за рахунок використання алгоритмів виявлення аномалій на часових рядах та кореляційного аналізу, дозволяє автоматично виявляти аномальні показники продуктивності.

Ключові слова: показники продуктивності, виявлення аномалій.

Вступ

Розподілені комп'ютерні системи (РКС) забезпечують масштабованість, однак досягнення високої продуктивності, з точки зору високої пропускної здатності і малого часу відгуку, залишається складною і важливою проблемою. Тому важливо постійно аналізувати продуктивність РКС для своєчасного визначення і усунення вузьких місць [1]. Для спостереження за роботою РКС використовують системи моніторингу показників продуктивності, які дозволяють відслідковувати діяльність системи. Моніторинг — це комплекс швидкого знаходження проблеми, оповіщення про неї адміністраторів та діагностики, що дає повну і точну інформацію про проблеми [2]. Для збору інформації про систему на кожному сервері встановлюється агент, який відправляє показники продуктивності до системи моніторингу, в якій поточні показники порівнюються з граничними, у разі відхилення система сигналізує про проблеми адміністратору. Постійний моніторинг допомагає уникнути простоїв в роботі, підтримувати всі сервіси в робочому стані і зберігати необхідний рівень їх якості, а також спланувати її модернізацію [3]. При появі і поширенні віртуалізації виникла необхідність відстежувати стан і фізичних серверів і віртуальних серверів. Серед основних функцій систем моніторингу можна виділити наступні [2]: стеження; зберігання бази даних; побудова звітів; візуалізація; пошук вузьких місць.

Дана робота присвячена актуальній задачі розробки засобу автоматичного виявлення аномалій продуктивності в РКС. Виявлення аномалій є задачею знаходження закономірностей у даних, які не відповідають моделі «нормальної» поведінки.

Для досягнення вищезазначених цілей необхідно розробити засіб, який здатен виконувати:

- виявлення аномальних значень показників продуктивності;
- кореляційний аналіз для автоматичного виявлення аномальних показників між різними системами.

Показники продуктивності

Кожен комп'ютер має апаратні ресурси: процесори, оперативна пам'ять, дисковий простір, мережеві інтерфейси і т. д. Використання кожного ресурсу можна контролювати [4]. Для контролю використання CPU застосовується «середнє завантаження», яке засновано на середній довжині черги планувальника на протязі короткого періоду часу. Для оперативної пам'яті контролюється загальна кількість фізичної пам'яті в системі та показник використаної пам’яті. Для жорстких дисків та інших сховищ даних може вимірюватися завантаженість наявних ресурсів зберігання даних. Для мережевих інтерфейсів потрібно контролювати показники передачі даних. Крім цього, можливе урахування додаткових показників безпеки розподілених комп’ютерних систем.

Огляд існуючих програмних рішень для виявлення аномалій продуктивності

Microsoft Azure Anomaly Detection API. Anomaly Detection API — це хмарний сервіс(SAAS) від корпорації Microsoft для виявлення аномалій в даних часових рядів, які рівномірно розташовані в часі. Працює на основі платформи Azure Machine Learning [5], запускає детектори аномалій на даних, що були завантажені і повертає результати виявлення аномалій в кожній точці часу. Можливість потокової відправки даних та отримання результатів перевірки відсутня, тому цей сервіс на підходить для моніторингу в реальному часі.

Numenta Grok Grok — сервіс аналітики та виявлення аномалій в часових рядах від компанії Numenta, який знаходить складні шаблони в потоках даних і генерує прогнози в режимі реального

 ISSN 1560-8956

часу. Grok використовує алгоритм навчання на основі ієрархічної тимчасової пам’яті (HTM). HTM є біометричною моделлю, що заснована на теорії пам'яті прогнозування функції мозку, описаної Джеффом Хокінс [6].

AppDynamics Server Monitoring [7] — хмарний сервіс для моніторингу серверів. Алгоритми виявлення аномалій основані на динамічному базисі. Збір показників продуктивності відбувається пропрієтарним агентом, що доступний для Windows та Linux.

Prelert Anomaly Detective [8] — хмарний сервіс аналітики для системи зберігання та аналізу логів Splunk, Hadoop та Elasticsearch. Він автоматично встановлює моделі нормальної поведінки і використовує статистичний аналіз для виявлення аномалій та кореляції між різними метриками. Алгоритми роботи не розкриваються.

Sematext SPM [9] — сервіс для моніторингу продуктивності серверів та додатків з можливостями виявлення аномалій. Для збору метрик з сервера використовується додаток з відритим вихідним кодом collectd. Алгоритми виявлення аномалій заснований на машинному навчанні, точні методи не розкриваються.

RRDtool — набір утиліт з відкритим вихідним кодом для роботи з кільцевими базами даних. В базу даних вносяться консолідовані підсумки, старі дані при цьому затираються новими, реалізовано виявлення аномальної поведінки. Використовується алгоритм Холта-Вінтерса, який адаптивно пророкує майбутні спостереження часового ряду. Передбачення може також розглядатися як «згладжене» значення для часового ряду [10].

Skyline [11] — система виявлення аномалій в реальному часі з відкритим вихідним кодом, яка створена для пасивного спостереження за метриками. Система автоматично аналізує дані і приймає рішення про «аномальність» цих даних. На даний момент реалізовані наступні алгоритми: середнє відхилення; тест Граббса; середнє за першу годину; середньоквадратичне відхилення від середнього; середньоквадратичне відхилення від ковзаючого серед-нього; метод найменших квадратів; викиди на гістограмі; критерій згоди Колмогорова.

Порівняємо описані програмні рішення за типом, використовуваними алгоритмами, наявності оповіщень та наявності функцій пошуку кореляцій (табл. 1).

Таблиця 1.

Порівняння програмних рішень для виявлення аномалій

Рішення	Тип	Алгоритми	Оповіщення	Пошук кореляцій
Microsoft Azure Anomaly Detection API	SAAS	Машинне навчання – пропрієтарні алгоритми	-	-
Numenta Grok	SAAS	Машинне навчання — Ієрархічна тимчасова пам'ять	+	-
AppDynamics Server Monitoring	SAAS	Пропрієтарні	+	-
Prelert Anomaly Detective	SAAS	Пропрієтарні	+	+
Sematext SPM	SAAS та standalone	Машинне навчання – пропрієтарні алгоритми	+	+
RRDTool	Opensource	Алгоритм Холта-Уінтерса	-	-
Skyline	Opensource	Середнє відхилення; тест Граббса; середнє за першу годину; середньоквадратичні відхилення; викиди на гістограмі; критерій згоди Колмогорова.	+	-

З таблиці ми бачимо, що усім вимогам задовольняють тільки сервіси Prelert Anomaly Detective та Sematext SPM, але вони є комерційними. Серед програмного забезпечення з відкритим вихідним кодом жодне з рішень не покриває в повній мірі усі функціональні вимоги. Так, у RRDTool відсутні оповіщення та засоби кореляційного аналізу, Skyline має функцію оповіщення при виявленні аномалій, але кореляційний аналіз відсутній. Таким чином, актуальною є задача розробки засобу для моніторингу показників продуктивності розподілених комп’ютерних систем.

 ISSN 1560-8956

Вибір алгоритмів для виявлення аномалій показників продуктивності розподілених комп’ютерних систем

Алгоритм виявлення аномалій на часових рядах

В [12] був описаний алгоритм для виявлення аномалій, який заснований на принципі кластеризації та не вимагає навчання з учителем. Це забезпечить виявлення як відомих, так і невідомих аномалій. Алгоритм працює у режимі онлайн та веде безперервне спостереження. Щоб уникнути ручного маркування даних і виявити аномалії використвується метод навчання без вчителя, а саме самоорганізовані карти Кохонена (SOM). SOM здатна захоплювати складні поведінки системи, будучи дешевою для обчислення, ніж інші підходи, такі, як метод k найближчих сусідів.

Для роботи з SOM збираємо вектор вимірювань D(t) = [x_1, x_2, …, x_n] безперервно з кожної системи, де x_i позначає один показник продуктивності системи (наприклад, завантаження процесора, пам'яті, дискового вводу/виводу, або мережевий трафіку), і використовуємо ці вектори, як входи для навчання саоморганізованої карти Кохонена.

Алгоритм кореляційного аналізу

Багато моделей для виявлення кореляцій є надійними і точними, за винятком випадку, коли вони застосовуються до часового ряду, який характеризується високою мінливістю, що характерно для показників продуктивності розподілених комп’ютерних систем. Відповідно до [13], висока мінливість це явище, при якому множина спостережень приймає значення, які відрізняються на порядки величини. При цьому більшість спостережень приймає значення навколо тенденції часового ряду, деякі значення з помітною частотою відс тупають від тренду, а іноді приймають екстремально великі значення.

В [13] представлена нова кореляційна модель, а саме CoHiVa (кореляція для сильно мінливих даних — Correlation for Highly Variable data), яка здатна оцінити схожість між часових рядів, які характеризуються високою мінливістю. Ця модель може розглядатися як поліпшення LoCo алгоритму, який пропонує оцінку кореляції на основі аналізу подібності образів. CoHiVa розширює цю ідею до кореляційного аналізу для більш варіабельних доменів. На відміну від моделі

Пірсона, яка добре працює, тільки якщо часові ряди пов'язані лінійною залежністю, CoHiVa здатна виявляти як лінійні, так і нелінійні залежності [13].

Крім того, CoHiVa не виконує розподіл даних, як того вимагають моделі Спірмена і Кендалла. Алгоритм CoHiVa грунтується на наступних чотирьох основних етапах [13]:

1) виділення з x і y тренду та збурень;

2) видалення помилок забруднюючих часовий ряд;

3) вибір тренду за допомогою видалення збурень;

4) обчислення кореляційного індексу CoHiVa між двома часовими рядами по оцінці подібності між їх формою тренда.

Вибір агентів збору та відправки метрик. В якості агентів збору та відправки метрик можуть виступати програми, які працюють по протоколу Carbon, наприклад StatsD та collectd. StatsD — це мережевий демон з відкритим вихідним кодом, який постійно отримує показники продуктивності по протоколу UDP від інших програм, що встановлені на локальній системі, і періодично посилає сукупні показники до інших сервісів. Collectd — це мережевий Unix демон з відкритим вихідним кодом, який збирає, передає і зберігає дані про продуктивність комп'ютерів і мережевого устаткування. Збирання та зберігання даних обробляється плагінами у вигляді загальних об'єктів.

Вибір бази даних. Для зберігання часових рядів будемо використовувати базу даних InfluxDB. База, що написана мовою Go, і позиціонується як база даних для зберігання часових рядів, метрик та інформації про події. Має SQL-подібний програмний інтерфейс для взаємодії [14].

СХЕМА ВЗАЄМОДІЇ КОМПОНЕНТІВ СИСТЕМИ ВИЯВЛЕННЯ АНОМАЛІЙ ПОКАЗНИКІВ ПРОДУКТИВНОСТІ

Система виявлення аномалій показників продуктивності складається з трьох компонентів:

- агентів збору та відправки метрик, які встановлюються на сервери РКС;
- система аналізу метрик та виявлення аномалій;
- бази даних для зберігання показників продуктивності та виявлених аномалій.

 ISSN 1560-8956

Адміністратор має змогу конфігурувати роботу системи, переглядати графіки та діаграми для моніторингу показників продуктивності, переглядати виявлені аномалії у веб-інтерфейсі.

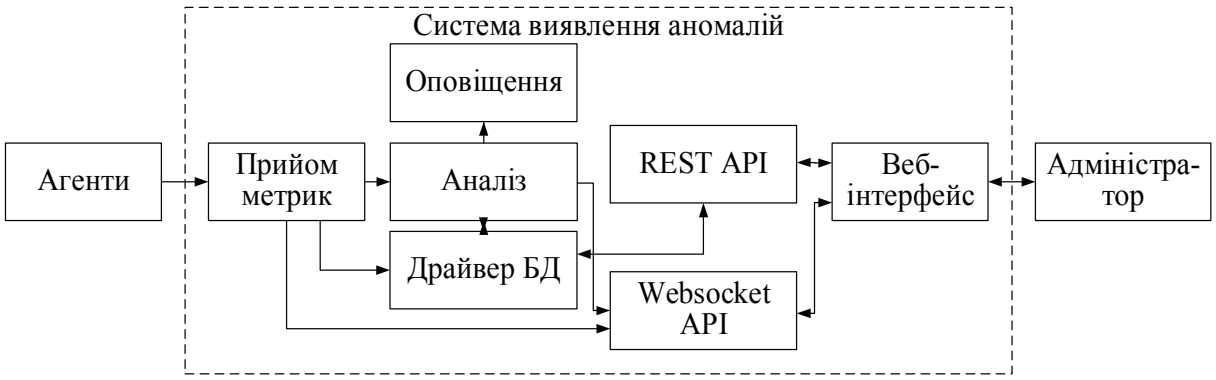

Схема взаємодії компонентів системи

Система складається з 6 підсистем:

- підсистема прийому метрик — приймає метрики від агентів по протоколу Carbon, передає їх до підсистеми аналізу, додає показники в базу даних для зберігання та передає через Websocket API для відображення графіків у режимі онлайн;
- підсистема аналізу – проводить аналіз показників, записує у БД результати аналізу, сповіщає підсистему оповіщення та передає дані через Websocket APi у разі виникнення проблем;
- підсистема оповіщення – відправляє оповіщення через сервіс групових чатів Slack;
- REST API – надає користувачу програмний для взаємодії з системою;
- Websocket API – створює програмний інтерфейс для отримання показників продуктивності в онлайн режимі;
- веб-інтерфейс адміністратора.

Також система має засоби роботи в кластерному режимі для забезпечення масштабування та підвищення надійності роботи.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СЕРВЕРНОГО ЗАСТОСУНКУ ВИЯВЛЕННЯ АНОМАЛІЙ ПОКАЗНИКІВ ПРОДУКТИВНОСТІ

Для розробки серверного застосунку використано наступні технології: мова програмування Java, інструмент для збірки проекту Apache Maven, фреймворк Spring для прискорення розробки та стандартизації рішень, метод взаємодії компонентів розподіленого додатка REST для

управління сервером, в якості формату передачі повідомлень використано JSON, для відправки повідомлень адміністратору у режимі онлайн використано протокол Websocket. Для написання логіки роботи використано мову програмування Javascript з фреймворком AngularJS.

В якості бази даних обрано спеціалізовану базу даних для зберігання часових рядів Influxdb, яка дозволяє працювати в кластерному режимі та забезпечує горизонтальне масштабування для зберігання великої кількості часових рядів. Для збору метрик з застосунків обрано StatsD, який приймає показники по протоколу UDP, збирає їх, та відправляє на обробку по протоколу Carbon [15].

Висновки

В роботі розглянуто необхідність використання систем моніторингу з виявленням аномалій, показані показники продуктивності систем та їх представлення на часових рядах. Розглянуто існуючі програмні засоби для виявлення аномалій показників продуктивності в РКС. Описано можливість використання кореляційного аналізу для виявлення аномалій. Виявлення кореляції між даними з високою мінливістю, які зібрані під час моніторингу ресурсів, вимагає використання спеціалізованих алгоритмів.

В результаті розроблено засіб, що може забезпечити ефективне виявлення аномалій показників продуктивності в розподілених комп’ютерних системах. Використання даного способу дозволить підвищити ефективність та зручність обслуговування РКС. Розроблено архітектуру системи виявлення аномалій показників продуктивності, описано можливість масштабування та резервування системи. Запропонований засіб має кілька переваг: він не потребує будь-яких припущень про статистичні властивості; йому не потрібен попередній аналіз характеристик часових рядів; він здатний адаптувати свої параметри до отриманих даних і виявляти лінійні та нелінійні залежності.

Список використаних джерел

1. Practical fault detection & alerting. You don’t need to be a data scientist – Ht;bv ljcnegeˆ httpˆ‖dieter/plaetinck/be|practical-fault-detection-alerting-dont-need-to-be-data-scientist/html – Lfnf ljcnegeˆ 23/06/2015 – Yfpdf p trhfye/
2. Veasey T., Dodson S. Anomaly Detection in Application Performance Monitoring Data // IJMLC. 2014. Vol. 4. pp. 120–126.

3. Zhang Q., Cherkasova L., Mathews G., Greene W., and Smirni E. R-Capriccio: A Capacity Planning and Anomaly Detection Tool for Enterprise Services with Live Workloads // HPL. 2012. Vol. 7. pp. 129–149.
4. Peiris M., Hil J.H., Thelin J., Bykov S., Kliot G., and Konig C. PAD: Performance Anomaly Detection in Multi-Server Distributed Systems // CLOUD 2014: 7th IEEE International Conference on Cloud Computing. Alaska. 2014.
5. Microsoft Azure Anomaly Detection. – Режим доступу: http://datamarket.azure.com/dataset/aml_labs/anomalydetection – Дата доступу: 23.06.2015 – Назва з екрану.
6. Numenta Grok – Режим доступу: URL: http://numenta.com/grok/ – Дата доступу: 23.06.2015 – Назва з екрану.
7. Appdynamics: Behavior Learning and Anomaly Detection – Режим доступу: URL: https://docs.appdynamics.com/display/PRO14S/Behavior+Learning+and+Anomaly+Detection – Дата доступу: 23.06.2015 – Назва з екрану.
8. Prelert – Режим доступу: URL: http://info.prelert.com – Дата доступу: 23.06.2015 – Назва з екрану.
9. SPM Performance Monitoring, Alerting, & Anomaly Detection – Режим доступу: URL: http://oss.oetiker.ch/rrdtool/ – Дата доступу: 23.06.2015 – Назва з екрану.
10. Notes on RRDTOOL implementation of Aberrant Behavior Detection – Режим доступу: URL: http://cricket.sourceforge.net/aberrant/rrd_hw.htm. – Дата доступу: 23.06.2015 – Назва з екрану.
11. Skyline – Режим доступу: https://github.com/etsy/skyline. – Дата доступу: 23.06.2015 – Назва з екрану.
12. InfluxDB – Режим доступу: URL: https://influxdb.com/ – Дата доступу: 23.06.2015 – Назва з екрану.
13. Tosia S., Casolaria S., and Colajannia M. Detecting correlation between server resources for system management // Journal of Computer and System Sciences, Vol. 8, No. 4, June 2014. pp. 821–836.
14. Dean D. J., Nguyen H., and Gu X. ICAC '12 // UBL: Unsupervised Behavior Learning for Predicting Performance Anomalies in Virtualized Cloud Systems. 2012. pp. 191–200.
15. Carbon – Режим доступу: URL: https://github.com/graphite-project/carbon – Дата доступу: 23.06.2015 – Назва з екрану.