

УДК 621.3.087.44

РОЗРОБКА СИСТЕМИ ІНТЕРНЕТУ РЕЧЕЙ НА ОСНОВІ ПЛАТФОРМО-НЕЗАЛЕЖНИХ ТЕХНОЛОГІЙ

Токаренко О. В., Богомазов С. А.

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

E-mail: tokarenko36@gmail.com

Протягом довгого часу домінуюче місце в розробці серверних додатків для Інтернету речей (Internet of Things, IoT) займала мова програмування Java [1]. Але не дивлячись на велику популярність даної мови, вона має ряд недоліків. В цілому засновані на Java веб-фреймворки не дозволяють швидке внесення змін в роботу. Ще один фактор, що сповільнює веб-розробку – недостатня гнучкість мови Java. В цьому випадку її статична типізація є недоліком, а не перевагою. Прискорити процес веб-розробки, прототипування, написання сценаріїв дозволяє використання JVM-сумісних програмних засобів, що підтримують динамічну типізацію, зокрема мови Groovy та фреймворку Grails.

Grails – це фреймворк для швидкої веб-розробки, в основі якого лежить динамічна мова Groovy. В першу чергу Grails допомагає заощадити час при роботі над автоматичним створенням структури проекту. Даний фреймворк активно використовує концепцію програмування за угодами, що застосовується в аналогічному фреймворці Rails для мови Ruby. Якщо код відповідає угодам, то фреймворк виконує багато завдань, що вирішуються на рівні шаблонного коду [2]. Groovy – це об'єктно-орієнтована динамічно типізована мова, яка працює на віртуальній машині Java (JVM), що забезпечує платформонезалежність технології. Її динамічні можливості доповнюють статично типізовану мову Java. Використання динамічної типізації дозволяє скоротити код, тому що зникає оголошення «очевидних» типів, забезпечується більш швидкий зворотній зв'язок і гнучкість при присвоєнні об'єктів різних типів єдиній змінній, з якою виконуються операції. Groovy заснований на ідеях Smalltalk, Ruby і Python і підтримує функціональні літерали, вбудовані колекції, регулярні вирази. В Groovy значно спрощується робота з XML завдяки вбудованій підтримці XML-обробки.

На базі фреймворку Grails з використанням мови Groovy була розроблена серверна частина демонстраційної системи Інтернету речей. Програмне забезпечення серверної частини системи розроблено для запуску у віртуальній машині Java. Такий підхід дозволив отримати платформонезалежне програмне рішення, що не залежить від конкретних архітектур. Демонстраційна система складається з інтелектуальних датчиків, CoAP сервера і сервера додатків з СУБД PostgreSQL. Дані про температуру та вологість надходять з інтелектуальних датчиків в одноплатний комп'ютер Raspberry Pi. За допомогою протоколу CoAP (Constrained Application Protocol) виконуються запити на отримання даних.

CoAP – це веб-протокол, оптимізований для мереж з обмеженими ресурсами, типових для IoT і M2M (Machine-to-Machine) додатків [3]. Протокол базується на REST архітектурі, в якій ресурси доступні і ідентифіковані за допомогою уніфікованих ідентифікаторів ресурсів (URI). З ресурсами можна взаємодіяти за допомогою тих самих методів, які використовуються в HTTP: GET, PUT, POST і DELETE. Протокол складається з підмножини функціональних можливостей HTTP, які були перероблені з урахуванням обмежених обчислювальних можливостей і необхідності споживання мінімальної енергії. Транзакція CoAP має в 10 разів менше за розміром повідомлення, ніж HTTP. Це є наслідком значного стиснення заголовку, який реалізовано в протоколі CoAP.

В якості хмарної платформи було використано Amazon EC2. За допомогою налаштування конфігурації iptables реалізовано можливість обміну даними з веб-сервером як за протоколом HTTP, так і за протоколом CoAP. Для обміну даними за протоколом CoAP використовувалась бібліотека Californium, для роботи через UART – бібліотека jssc, що дозволило абстрагуватись від низькорівневих деталей. Для дослідження роботи сервера використовувався додаток Corper для браузеру Firefox та програма для аналізу мережових пакетів Wireshark.

Запит до CoAP-сервера типу GET coap://name:port/.well-known/core повертає відповідь в форматі xml, що містить перелік URI усіх ресурсів. Запит до CoAP-сервера типу observe для ресурсу має вигляд coap://192.168.2.1:5683/SensorNode/ObjectTemperature. В результаті сервер періодично посилає повідомлення з експериментальними даними при кожному оновленні CoAP ресурсу. У випадку, коли метод не підтримується ресурсом, відправляється повідомлення з кодом 4.05- method not allowed.

Застосування протоколу CoAP дозволило зменшити вимоги до пропускну здатності комунікаційного каналу, що дозволяє використовувати навіть низькошвидкісне модемне з'єднання. Структура системи є горизонтально масштабованою, тобто дозволяє розширити конфігурацію на будь-якому рівні. Суттєвою перевагою розробленої структури є можливість перенесення складних обчислювальних операцій на рівень “хмарних” сервісів.

Використання фреймворку Grails та мови Groovy дало змогу значно скоротити кількість коду, надати гнучкості і стабільності роботи при внесенні змін при збереженні платформонезалежності програмного забезпечення.

Ключові слова: Інтернет речей, обробка даних, протокол CoAP, веб-додаток.

Література

- [1] О. В. Токаренко, та С. А. Богомазов, “Система збору експериментальних даних на основі Java-фреймворків”, на *XV Всеукр. наук.-практ. конф. студ., асп. та мол. вч. Ефективність інженерних рішень у приладобудуванні*, Київ, 2019, с.532-534.
- [2] Б. Эванс, и М. Вербург, *Java. Новое поколение разработки*. СПб, Россия: Питер, 2014.
- [3] Z. Shelby, K.Hartke, and C. Bormann, The Constrained Application Protocol (CoAP). RFC 7252, 2014. [Online]. Available: <http://www.rfc-editor.org/info/rfc7252>. Accessed on: June 1, 2018.