

УДК 004.42

*Картамишев Д.О., к.т.н., асистент
Донбаська державна машинобудівна академія*

МОВА ПРОГРАМУВАННЯ GO У ЗАДАЧАХ АВТОМАТИЗАЦІЇ ІНФРАСТРУКТУРНИХ ПРОЦЕСІВ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Зростання складності сучасних програмних систем безпосередньо впливає на підходи до управління обчислювальною інфраструктурою. У хмарних середовищах та мікросервісних архітектурах кількість взаємодіючих компонентів може вимірюватися сотнями сервісів, що значно ускладнює ручне адміністрування та підтримку систем. За таких умов автоматизація операційних процесів поступово переходить із допоміжного інструмента у необхідну складову функціонування інформаційної інфраструктури.

У контексті розробки інфраструктурного програмного забезпечення особливу увагу привертає мова програмування Go. Її архітектура формувалася з урахуванням потреб створення мережових сервісів та системного програмного забезпечення. Поєднання відносно простого синтаксису з компільованою моделлю виконання дозволяє досягати продуктивності, характерної для системних мов програмування, зберігаючи при цьому зручність розробки.

Однією з ключових особливостей Go є підтримка конкурентних обчислень. Механізм goroutines забезпечує можливість створення великої кількості паралельних процесів виконання з мінімальними накладними витратами. На відміну від традиційних потоків операційної системи, goroutines реалізуються у межах власного планувальника середовища виконання Go, що дозволяє більш ефективно використовувати обчислювальні ресурси під час обробки великої кількості подій.

Подібний підхід особливо добре проявляє себе у системах, де одночасно виконуються операції моніторингу, обробки запитів та управління конфігураціями сервісів. Передача даних між процесами за допомогою channels формує зрозумілу модель взаємодії компонентів та дозволяє уникнути складних механізмів синхронізації, які часто використовуються у традиційному багатопотоковому програмуванні.

Поширення Go у сфері автоматизації інфраструктури підтверджується її використанням у низці сучасних технологічних платформ. Зокрема, такі системи як Docker, Kubernetes, Prometheus та Terraform реалізовані із застосуванням цієї мови. Зазначені проекти виконують задачі контейнеризації, оркестрації сервісів та збору

телеметричних даних, що передбачає обробку значних обсягів подій у режимі реального часу.

Окремої уваги заслуговує і процес розгортання програм, написаних на Go. Після компіляції формується автономний виконуваний файл, який містить усі необхідні залежності. Такий підхід значно спрощує розгортання програмних компонентів у контейнерних та кластерних середовищах, що є типовими для сучасних інформаційних систем.

У межах проведеного дослідження було проаналізовано підходи до побудови сервісів автоматизованого моніторингу та керування інфраструктурою із використанням Go. Запропоновано архітектурну модель сервісу збору подій, що базується на асинхронній обробці повідомлень та використанні конкурентних механізмів мови. Така модель дозволяє зменшити затримки обробки подій і водночас підвищити ефективність використання обчислювальних ресурсів.

Отримані результати свідчать, що використання Go у задачах автоматизації інфраструктури є доцільним для систем, у яких важливими є висока пропускна здатність, ефективна обробка подій та простота розгортання програмних компонентів.

Варто також зазначити, що застосування Go у задачах автоматизації часто поєднується з подієво-орієнтованими архітектурами. У таких системах обробка подій відбувається асинхронно, а окремі компоненти взаємодіють через повідомлення або черги. Подібна модель дозволяє зменшити зв'язність між сервісами та підвищити стійкість системи до відмов окремих компонентів. У практичних реалізаціях це означає, що система здатна продовжувати роботу навіть у випадку часткових збоїв інфраструктури.

З технічної точки зору Go добре підходить для реалізації подібних механізмів. Вбудовані бібліотеки мови забезпечують зручні засоби для роботи з мережевими протоколами, обробки HTTP-запитів та створення сервісів, що працюють у режимі постійного обміну даними. Крім того, стандартна бібліотека містить інструменти профілювання та моніторингу продуктивності, що значно спрощує аналіз поведінки системи під навантаженням.

Список використаних джерел:

1. Donovan A. A., Kernighan B. W. The Go Programming Language. – Boston : Addison-Wesley Professional, 2016. – 380 p.
2. Pike R. Go at Google: Language Design in the Service of Software Engineering [Electronic resource]. – 2012. – Available at: <https://go.dev/talks/2012/splash.article> (accessed: 16.03.2026).