

INFORMATION AND WEB TECHNOLOGIES

Диференційна методика порівняння схем розміщення стану в обробці функціонально залежних потоків подій

Максименко Андрій Олександрович¹

¹доктор філософії, старший викладач;
Національний університет «Одеська політехніка», Україна

Значний клас систем потокової обробки перевіряє кожен вхідний подію відносно поточної проекції накопиченого стану: ставку – відносно найкращої, списання – відносно залишку. Події з однаковим ідентифікатором агрегату функціонально залежні й обробляються послідовно. Класична реалізація тримає проекцію в зовнішньому сховищі й виконує на кожен подію читання та запис, тож пропускна здатність у межах агрегату обмежена оберненою величиною затримки звернення до сховища [1].

Винесення проекції в оперативну пам'ять вузла з періодичним збереженням контрольних точок [2] усуває це звернення з гарячого шляху. Проте перевага такої схеми обґрунтовується здебільшого аналітично, а спроба виміряти її настановується на методологічну перешкоду: дві незалежно написані реалізації відрізняються не лише розміщенням стану, а й кількістю з'єднань, серіалізацією, кешем і моментом підтвердження зсувів. Виміряна різниця не може бути коректно приписана архітектурному рішення.

Мета роботи – методика побудови стенда, у якому два профілі обробки відрізняються рівно одним чинником, розміщенням стану, а еквівалентність їхньої поведінки не декларується, а перевіряється в кожному прогоні.

Принцип диференційного профілювання. Стенд (рис. 1) спирається на три інваріанти. І1 (спільність правила): обидва профілі викликають один і той самий модуль, який реалізує бізнес-правило як чисту функцію – вона отримує подію та поточний стан і повертає рішення разом із новим станом, не звертаючись ані до сховища, ані до годинника, ані до генератора випадкових чисел. Детермінізм правила є класичною передумовою відтворюваності реплікованих обчислень [3].

INFORMATION AND WEB TECHNOLOGIES

I2 (спільність джерела): профілі споживають один і той самий набір подій з однаковим розподілом за розділами та порядком у межах розділу; генератор детермінований, з фіксованим зерном, а частка «гарячих» ключів задається явно.

I3 (перевіряна тотожність): кожен профіль веде журнал рішень – послідовність пар «зсув – рішення», і прогін дійсний лише тоді, коли журнали обох профілів збігаються поелементно; інакше показники непорівнянні.

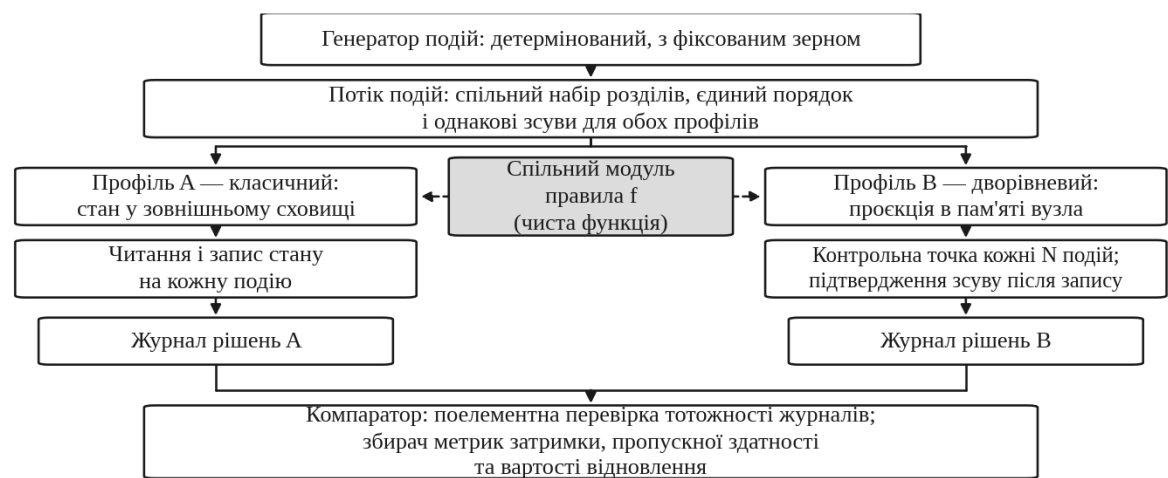


Рисунок 1
Структура диференційного вимірювального стенда

Саме інваріант I3 є ключовим: він перетворює вимірювання продуктивності на диференційне тестування [4], у якому кожен прогін одночасно є перевіркою коректності, а розбіжність журналів локалізує дефект за першим неспівпадінням.

Профілі обробки. Алгоритми наведено на рис. 2. Профіль А на кожну подію виконує читання стану зі сховища та умовний запис. Профіль В читає й оновлює стан у локальній таблиці в пам'яті, а до сховища звертається лише при досягненні порогу N прийнятих подій, записуючи контрольну точку одним пакетом [5]. Автоматичне підтвердження зсувів вимкнено: зсув підтверджується тільки після відповіді сховища [6]. Отже, звернень до сховища в профілі В приблизно в N разів менше.

Профіль В не є оптимізацією профілю А: він змінює момент, у який результат стає довговічним, бо між контрольними точками наслідки прийнятих подій існують лише в пам'яті. Тому дисципліна підтвердження зсувів входить до предмета вимірювання.

INFORMATION AND WEB TECHNOLOGIES

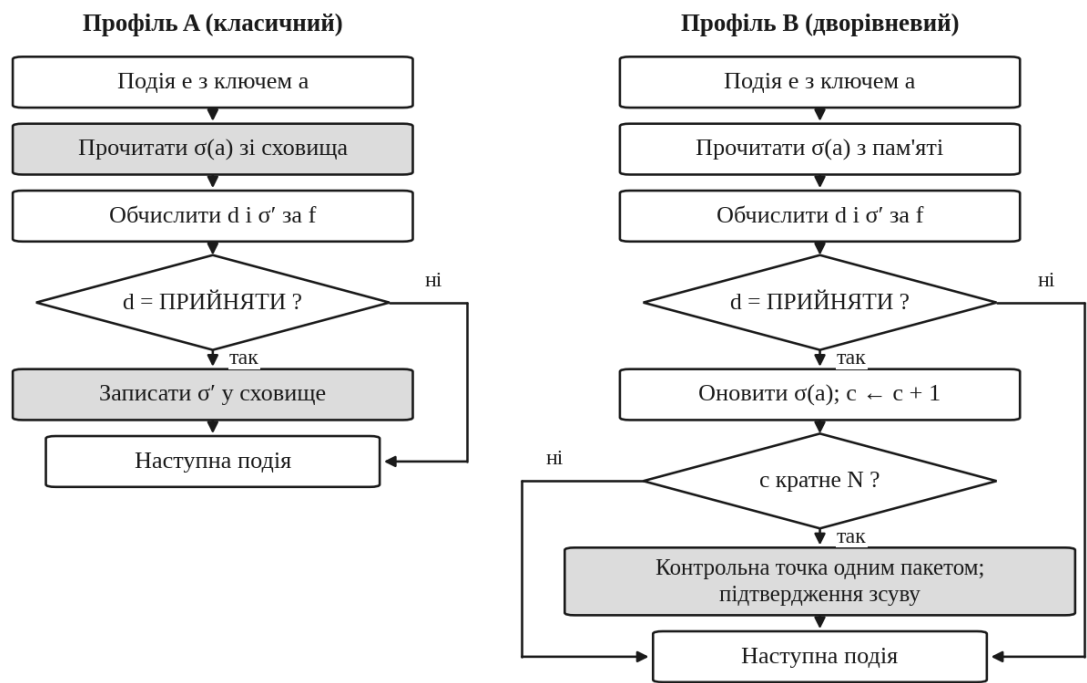


Рисунок 2
Алгоритми обробки однієї події; затінені блоки –
звернення до зовнішнього сховища

Протокол вимірjuвального прогону. Кроки подано на рис. 3. Прогрівання не входить у вибірку й усуває вплив холодних з'єднань і порожніх кешів. Ін'єкція збою – примусове завершення вузла профілю В – виконується в тому самому прогоні, після чого вузол відновлює стан із контрольної точки й відтворює події від підтвердженого зсуву. Прогін завершується порівнянням журналів.

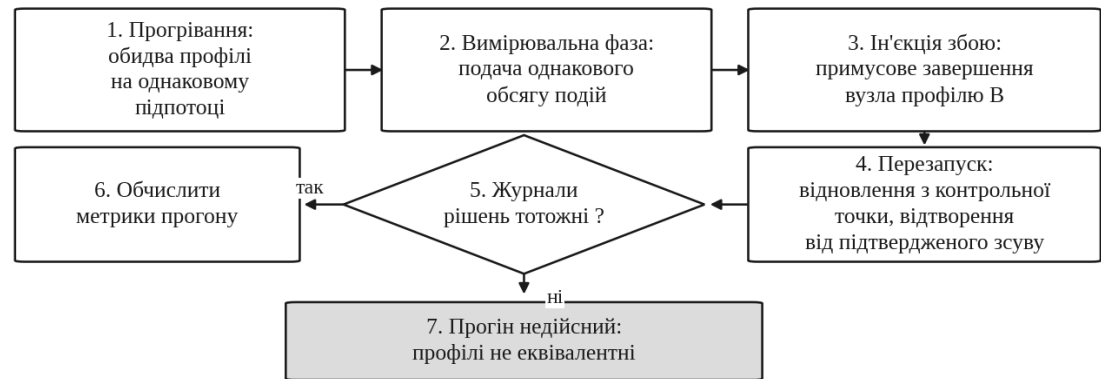


Рисунок 3
Протокол вимірjuвального прогону з ін'єкцією збою

INFORMATION AND WEB TECHNOLOGIES

Фіксуються кількість звернень до сховища на подію, затримка обробки, пропускну здатність за агрегатом, кількість повторно відтворених подій після збою та час до відновлення обслуговування. Перші три характеризують виграш, останні два – його ціну.

Контроль побічних чинників. Кеш у профілі А слід вимкнути або винести в окремий третій профіль, інакше він частково відтворює ефект пам'яті. Режим з'єднання зі сховищем має бути зіставним, інакше конвеєризація запису зараховується як перевага архітектури, хоча є перевагою транспорту.

Висновки. Запропоновано методика емпіричного порівняння схем розміщення стану при обробці функціонально залежних потоків подій. Її результат – конструкція експерименту: спільний модуль правила, спільне детерміноване джерело подій і поелементна перевірка тотожності журналів роблять різницю показників приписую саме розміщенню стану, а не деталям реалізації. Ін'єкція збою в межах того самого прогону дозволяє вимірювати виграш і вартість відновлення спільно. Подальша робота – розширення стенда на репліковану проекцію з кворумним узгодженням контрольних точок.

Список джерел:

- [1] Kreps J., Narkhede N., Rao J. Kafka: a distributed messaging system for log processing. NetDB Workshop. 2011. P. 1–7.
- [2] Fowler M. Memory Image. 2011. URL: <https://martinfowler.com/bliki/MemoryImage.html>
- [3] Schneider F. B. Implementing fault-tolerant services using the state machine approach: a tutorial. ACM Computing Surveys. 1990. Vol. 22, No. 4. P. 299–319.
- [4] McKeeman W. M. Differential testing for software. Digital Technical Journal. 1998. Vol. 10, No. 1. P. 100–107.
- [5] Carbone P., Ewen S., Fóra G. et al. State management in Apache Flink. Proceedings of the VLDB Endowment. 2017. Vol. 10, No. 12. P. 1718–1729. DOI: 10.14778/3137765.3137777
- [6] Wang G., Chen L., Dikshit A. et al. Consistency and completeness: rethinking distributed stream processing in Apache Kafka. SIGMOD. 2021. P. 2602–2613. DOI: 10.1145/3448016.3457556