

Побудова наближених до реального часу розподілених систем обробки даних із застосуванням Apache Storm

Виконав: студент групи ДА-22
Кривокінь Сергій Миколайович
Керівник: асистент, к.т.н.
Свірін Павло Володимирович

Актуальність: обчислення у реальному часі широко застосовуються у сучасних інформаційних галузях. Фремворки, одним з яких є Storm, дозволяють пришвидшити процес розробки системи для обробки даних в реальному часі завдяки реалізованим механізмам відмовостійкості, перебалансування, транспортування даних, та ін., які розробникам не доведеться реалізовувати з нуля.

Мета роботи:

- Розглянути підхід Storm до побудови систем обробки даних
- Розглянути інструменти та механізми, що надає Storm
- Порівняти Storm з аналогами та виокремити слабкі та сильні сторони
- Розробити систему для обробки даних з застосуванням Storm

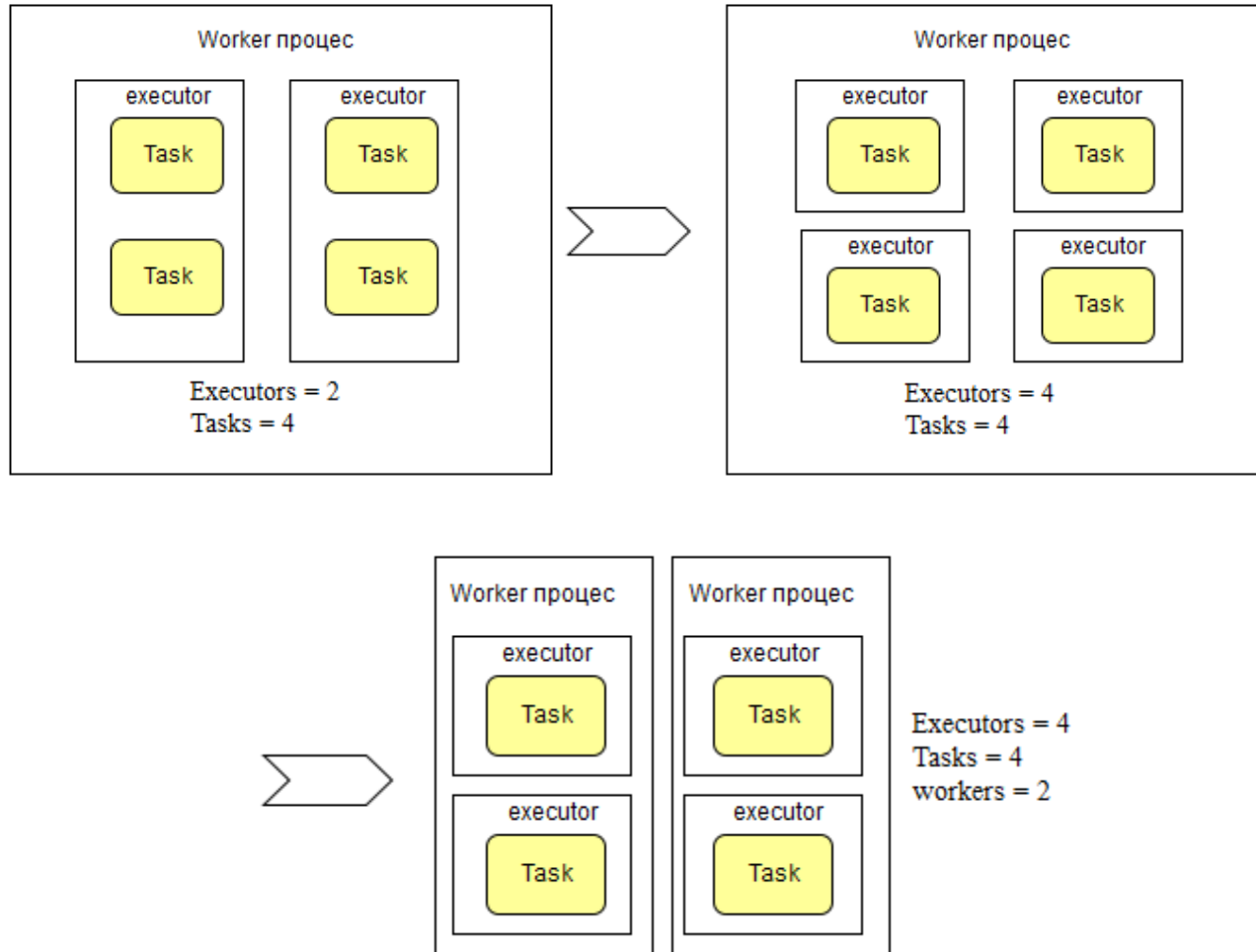
Основні сутності Storm

- Stream – потік кортежів(Tuple) даних
- Spout – джерело потоку даних
- Bolt – обробник потоку даних
- Топологія – напрямлений граф з елементів Spout та Bolt
- Worker – процес, що виконує певну частину логіки топології
- Executor – потік, викликаний worker'ом, що виконує логіку одного елемента топології
- Task – логічна одиниця, містить логіку елемента та виконується executor'ом
- Nimbus – головний процес, що розподіляє роботу між машинами в кластері
- Supervisor – процес, що запускає/зупиняє worker-процеси на машині в кластері

Паралелізм в Storm



Перезбалансування топології



Механізми відмовостійкості

Якщо worker процес зупиняється, supervisor перезапустить його. Якщо не вдається перезапустити, nimbus перенесе процес на іншу машину в кластері.

Якщо машина в кластері виходить з ладу, nimbus переносить усі worker процеси на інші машини в кластері.

Якщо nimbus або supervisor зупиняються, то відбувається їх перезапуск.

Гарантованість обробки повідомлення

Storm за вмовчуванням має гарантійну семантику 'at least once'

Storm Trident має семантику 'exactly once'

Для забезпечення гарантованості обробки повідомлення, джерело повідомлень має мати можливість відправити повідомлення повторно у разі помилки при обробці.

Порівняння з аналогами

- Spark Streaming – подібний до Storm Trident, завдяки пакетній обробці даних збільшує пропускну здатність системи, але також збільшує затримки на обробку. Має розвинуту бібліотеку machine-learning.
- Java Akka – більш базовий фреймворк, який має перевагу над Storm в плані гнучкості при розробці системи. Зокрема, на Akka можна розробити Storm-подібну систему. Головний мінус полягає в затратах на реалізацію механізмів, які в Storm уже є реалізованими.

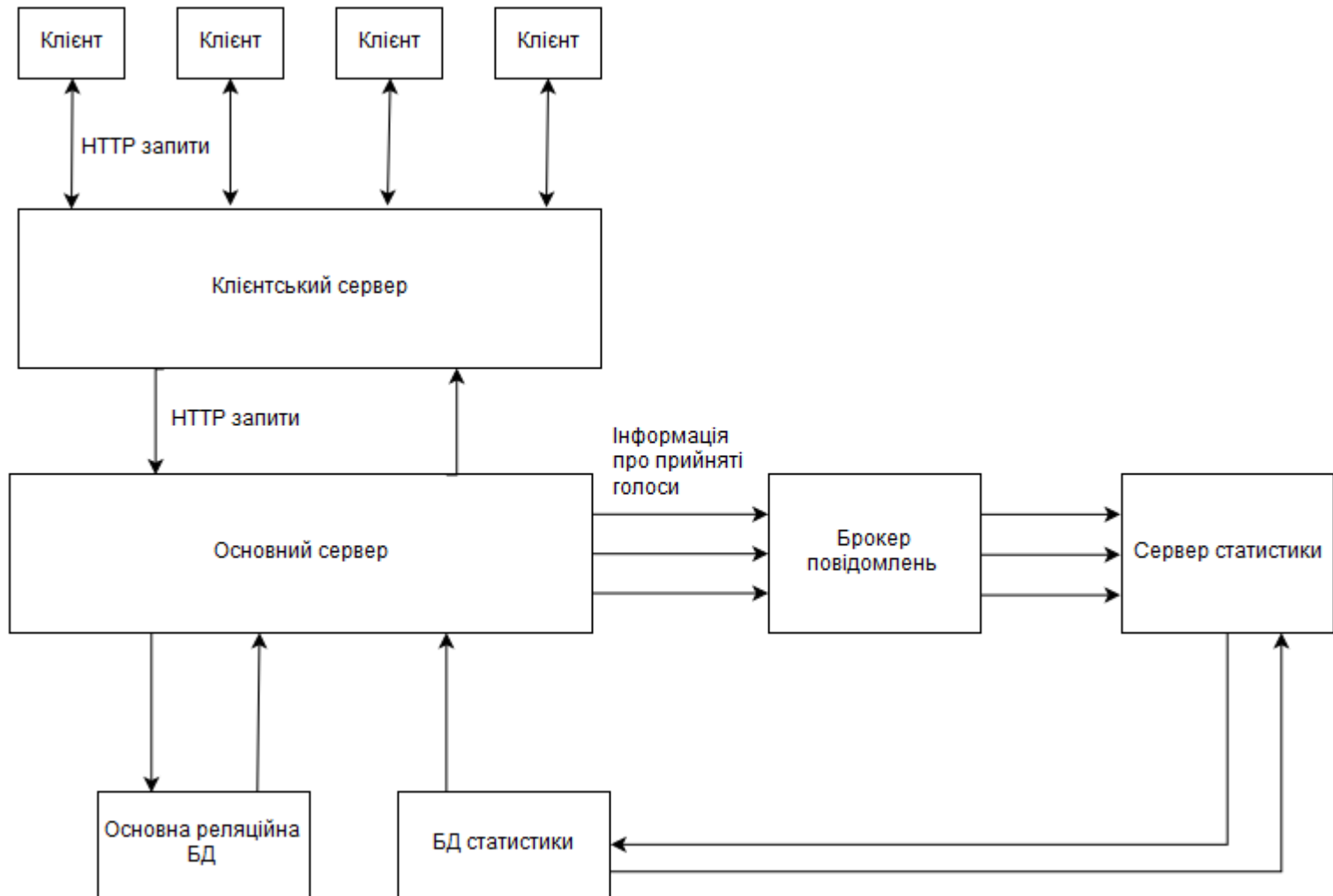
Розробка системи з використанням Storm

Предметною областю розробки було обрано ведення статистики в реальному часі, а конкретніше – статистики під час виборів.

Така система дозволить:

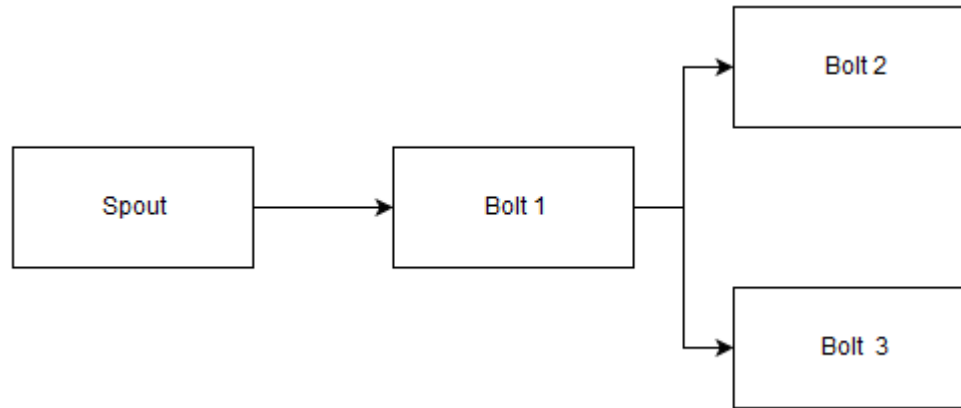
- Підвищити явку виборців завдяки зацікавленості
- Виявляти аномальні активності в процесі голосування
- Надати виборцям доступ до статистичних даних

Загальна архітектура системи



Архітектура компоненти статистики

Компонент статистики реалізований у вигляді Storm топології:



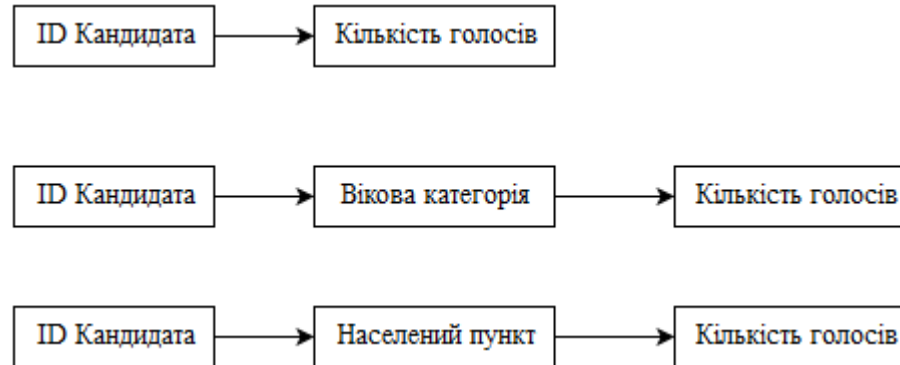
Spout – елемент який відповідає за передачу пар <Виборець, Кандидат> з джерела у топологію

Bolt1 – додає голос кандидату, та передає далі пари <Кандидат, Вік виборця> та <Кандидат, Локація виборця>

Bolt2 – оброблює передану з Bolt1 пару <Кандидат, Вік виборця>

Bolt3 – оброблює передану з Bolt1 пару <Кандидат, Локація виборця>

Структура сховища даних



Результати тестування системи

Для тестування було згенеровано 10^7 записів, які подавались на вхід у топологію

потоків Spout	потоків Bolt	час обробки 10^7 записів
1	1	120 с.
2	4	78 с.
4	8	90 с.

Інші підходи до реалізації

- Тригери в БД. Мінусом є збиткове навантаження головної БД, а також відсутність можливості виконувати додаткову логіку.
- Розрахунок по записам БД через інтервал часу. Мінусом є дуже нерівномірне розподілення навантаження для виконання задачі. Коли БД міститиме мільйони записів, такий підхід стане неможливим.
- Виконання пакетної обробки даних. Мінусом є більша затримка на отримання результату. Плюсом є більша пропускна здатність системи. Такий підхід є досить раціональним.

Дякую за увагу