

Розподілені системи.

Асинхронне програмування та його застосування в розподілених системах.



Підготував
Студент 4-курсу
Комар Іван

Вступ

- Розподілена система - це система, в якій різні компоненти працюють на різних фізичних машинах та обмінюються повідомленнями для досягнення спільної мети.
- Асинхронне програмування - це підхід до програмування, при якому виконання деяких операцій може затримуватися, але не блокувати інші операції.



Розподілені системи



- Розподілені системи можуть бути різної природи, наприклад, розподілені бази даних, розподілені обчислення, розподілені додатки та ін.
- У розподілених системах можуть виникати проблеми, пов'язані зі збоєм мережі, відмовою обладнання, низькою швидкістю передачі даних тощо.
- Для забезпечення надійності та масштабовності розподілені системи можуть використовувати різні підходи, наприклад, реплікацію даних, шарування додатків, мікросервісну архітектуру та ін.

Асинхронне програмування



- Асинхронне програмування може бути корисним у розподілених системах, оскільки дозволяє не блокувати виконання інших операцій під час очікування результатів виконання деякої операції.
- У мовах програмування, які підтримують асинхронність, для роботи з асинхронним кодом використовуються спеціальні конструкції, наприклад, `async/await` у C#.
- Асинхронне програмування може бути складнішим для розуміння та написання, ніж синхронне програмування, тому потребує додаткового навчання

Застосування асинхронного програмування в розподілених системах



- Асинхронне програмування може бути корисним у розподілених системах, де часто виникають ситуації, коли треба чекати на результат виконання деякої операції.
- Наприклад, у веб-додатках зазвичай використовують асинхронне програмування для обробки запитів користувачів, оскільки запити можуть затримуватися на деякий час.
- Для забезпечення ефективності та масштабовності розподілених систем можуть використовуватися різні підходи до асинхронного програмування, такі як non-blocking I/O, event-driven programming та ін.




Non-blocking I/O:

- Non-blocking I/O - це підхід до обробки введення/виведення, при якому операції введення/виведення не блокують потік виконання програми.
- Замість цього, програма може продовжувати виконання інших операцій, доки введення/виведення не буде готово.
- Non-blocking I/O може бути корисним у розподілених системах, де виконання деяких операцій може затримуватися на деякий час, тому дозволяє підтримувати ефективність та масштабовність системи.



Event-driven programming

- Event-driven programming - це підхід до програмування, при якому програма реагує на події, які відбуваються у системі.
- Замість циклічного опитування системи на предмет нових подій, програма очікує на сповіщення про нову подію та реагує на неї відповідним чином.
- Event-driven programming може бути корисним у розподілених системах, де потрібно реагувати на події, що відбуваються в інших компонентах системи.




Асинхронне програмування та його застосування в мовах програмування

- Асинхронне програмування можна застосовувати в більшості сучасних мов програмування, особливо тих, які мають вбудовану підтримку для паралельної обробки даних.
- Javascript
- Python
- C#
- Java
- Ruby



Асинхронне програмування у мові C#

- У мові C# існують спеціальні ключові слова `async` та `await`, які дозволяють реалізовувати асинхронне програмування.
- Ключове слово `async` вказує на те, що метод є асинхронним, тобто може виконуватися в іншому потоці виконання програми.
- Ключове слово `await` вказує на те, що метод може чекати на результат виконання деякої асинхронної операції, такої як введення/виведення або запит до бази даних.



Асинхронне програмування у мові Javascript

- В Javascript асинхронність реалізується за допомогою функцій зворотного виклику (callback functions), обіцянок (promises) та async/await функцій.
- Зазвичай це використовується для обробки подій (event handling) в браузері, взаємодії з сервером (наприклад, отримання даних з сервера) та інших асинхронних операцій.
- Функції зворотного виклику - це функції, які викликаються після того, як асинхронна операція закінчилась. Вони передаються як аргументи у функції, яка ініціює асинхронну операцію.
- Проте функції зворотного виклику мають певні недоліки, зокрема складність управління послідовністю виконання функцій, що може призвести до проблем з часом виконання та пам'яттю. Ці недоліки можна уникнути за допомогою обіцянок.
- Обіцянки - це об'єкти, які можуть бути використані для отримання результату асинхронної операції. Обіцянка приймає функції зворотного виклику для обробки результату операції. Обіцянки забезпечують більш чистий та зрозумілий код у порівнянні з функціями зворотного виклику.

Технології для розробки розподілених систем



- У розробці розподілених систем можуть використовуватися різні технології та фреймворки, такі як .NET Framework, Java Enterprise Edition, Apache Hadoop, Apache Kafka, Apache Spark, та багато іншого.
- Технології для розробки розподілених систем дозволяють розробникам швидко та ефективно розробляти та масштабувати розподілені системи.

Розподілена обробка даних



- Розподілена обробка даних може бути використана для обробки великих обсягів даних, що не можуть бути оброблені на одному комп'ютері.
- Розподілена обробка даних може бути реалізована з використанням різних технологій та фреймворків, таких як Apache Hadoop, Apache Spark, MapReduce, та ін.
- Розподілена обробка даних дозволяє ефективно використовувати ресурси багатьох комп'ютерів для обробки великих обсягів даних.

Застосування асинхронного програмування в розподілених системах



- Асинхронне програмування може бути використане в розподілених системах для поліпшення продуктивності та ефективності роботи системи.
- Асинхронне програмування може бути використане для зменшення часу очікування на результати роботи системи, зменшення навантаження на сервер, та поліпшення реактивності системи.
- Асинхронне програмування може бути використано в системах зі зменшеною пропускною здатністю, наприклад у мобільних додатках, де асинхронність дозволяє забезпечити відгук користувача на дії без очікування довгих запитів до сервера.



Висновки

- Розподілені системи та асинхронне програмування є важливими технологіями, які можуть бути використані для розробки ефективних та продуктивних систем.
- Розподілені системи можуть бути використані для обробки великих обсягів даних, розподіленої обробки та аналізу даних, розподіленої обробки транзакцій та багато іншого.
- Щоб досягти успіху у розробці розподілених систем та використанні асинхронного програмування, необхідно детально вивчати технології, фреймворки та принципи розробки таких систем.
- Розподілені системи та асинхронне програмування є важливими технологіями, які знаходять застосування у різних галузях, таких як фінанси, медицина, транспорт, телекомунікації та багато іншого.