

ЛЕКЦІЯ 5

МОДЕЛЬ РОЗПОДІЛЕНОГО ПРОГРАМУВАННЯ АСИНХРОННІ РОЗПОДІЛЕНІ СИСТЕМИ

Модель розподільного програмування.

Для вивчення різних аспектів функціонування розподільних систем часто використовують декілька моделей розподільної обробки даних. При цьому та чи інша модель обирається в залежності від задачі, яка досліджується, з області розподільних чи паралельних обчислень.

Розподільні обчислення зручно розглядати як сукупність дискретних подій, кожна з яких викликає невелику зміну в загальній системі. Ми будемо розглядати асинхронні розподільні системи, в яких процеси взаємодіють за допомогою асинхронного механізму обміну повідомленнями.

Модель розподільної системи.

Розподільна система складається зі скінченної множини N незалежних процесів

$$\{P_1, P_2, \dots, P_N\}$$

Якщо процес P_i може напряму відправляти сигнал процесу P_j , то ми кажемо, що між ними існує канал зв'язку C_{ij} . Для зручності всі канали будемо вважати однаковими. Щоб два процеси мали можливість обмінюватися повідомленнями, між ними потрібно встановити два різнонаправлених канали, кожен з яких передає повідомлення в одному напрямку. Структуру розподільної системи можна представити у вигляді орієнтованого графу, вершини якого відповідають процесам, а ребра – каналам зв'язку. Такий граф також називається топологією мережі розподільної системи. В нашому курсі говорячи про топологію мережі розподільної системи будемо мати на увазі, що вона є повнозв'язною.

Стан каналу C представляється сукупністю повідомлень відправлених по цьому каналу, але ще не отриманих отримуючим процесом. Внутрішній стан процесу P по-своєму формалізується в контексті кожної задачі. Стан процесу – це характеристика, що однозначно визначати його майбутню поведінку, всі подальші реакції на зовнішні події. В загальному випадку стан процесів визначається значеннями його лічильника команд, реєстрів та змінних.

Теоретична модель процесу визначається у вигляді множини його можливих станів, деякі підмножини цієї множини, елементи якої називаються початковими станами та множини дискретних подій. Подія e , що відбувається в процесі P , представляє собою атомарну дію, яка може змінити стан самого процесу та стан каналу C , що ініціалізований цьому процесу: стан C буде змінено при відправці повідомлення по цьому каналу, або при отриманні повідомлення з цього каналу. Тому всі події можуть бути класифіковані як внутрішні події відправки та отримання повідомлень. Відмітимо, що в якості однієї атамарної події часто розглядається відправка декількох повідомлень по декільком каналам зв'язку, інцидентним P .

Подія e визначається процесом P , я кому вона відбулася. Стан s процесу P безпосередньо перед подією e ; стан s' процесу P безпосередньо після події e ; каналом C , стан якого змінюється при настанні події e ; повідомленням m , що було відправлено по каналу або отримано з каналу C . Тому повідомлення e будемо визначати як п'ятірку (P, s, s', m, C) .

Множина подій процесу лінійно впорядкована згідно того, в якому порядку вони настають. Через e_i^x позначають x -ту подію у процесі P_i , $1 \leq i \leq N$. Ми будемо часто пропускати написання верхнього чи нижнього індексу, якщо його значення очевидно з контексту. Через $e_i^x(s)$ будемо позначати стан s' в який переходить процес P_i зі стану s після події e_i^x .

Виконанням процесу P_i називається послідовність подій

$$e_i^0, e_i^1, e_i^2, \dots, e_i^x, e_i^{x+1}, \dots$$

де e_i^0 - початковий стан процесу, та виконується співвідношення

$$s_i^{x+1} = e_i^x(s_i^x)$$

Виконання процесу P_i будемо позначати через R_i

$$R_i = (E_i \rightarrow_i)$$

E_i - множина подій, \rightarrow_i - лінійний порядок, заданий на цій множини згідно порядку подій відносно часу.

Але нас у більшості випадків буде цікавити виконання не 1 процесу, а всієї системи в цілому.

Глобальний стан S розподільної системи визначається станом всіх процесів та всіх каналів зв'язку, що входять до розподільної системи.

Визначимо через $E = \cup_i E_i$ - множину всіх подій, що відбувається при виконанні розподільної системи. Подія $e \in E$ вважається допустимою в глобальному стані S тоді і тільки тоді, коли:

1. В глобальному стані S процес P знаходиться в стані s .
2. Якщо ребро, відповідне каналу C направлене до вершини з процесом P та повідомлення m знаходиться в каналі C може бути прийнято з нього.

Стан $e(S)$ визначен тільки тоді, коли подія e допустима у глобальному стані S . Тоді $e(S)$ буде відрізнятися від глобального S тільки станом процесу P та, можливо, станом каналу C .

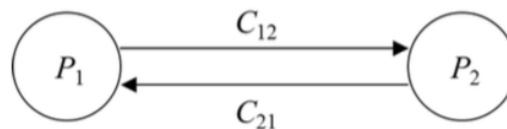
1. В глобальному стані $e(S)$ процес P буде знаходитися у стані s'
2. Якщо ребро, що відповідає каналу C , направлене до вершини s процесом P , то у глобальному стані $e(S)$ стан каналу C буде співпадати з його станом в S , за виключенням повідомлення m , що буде видалено з цього каналу.

3. Якщо ребро, що відповідає каналу C , направлене з вершини s процесом P , то у глобальному стані $e(S)$ стан каналу C буде співпадати з його станом в S , за виключенням повідомлення m , що буде додано у цей канал.

Послідовність подій $e_i^0, e_i^1, e_i^2, \dots, e_i^x, e_i^{x+1}, \dots$ будемо називати виконанням розподільної системи тоді і тільки тоді, коли подія e^x допустима у глобальному стані S^x , S^0 - початковий глобальний стан розподільної системи, тоді виконується співвідношення:

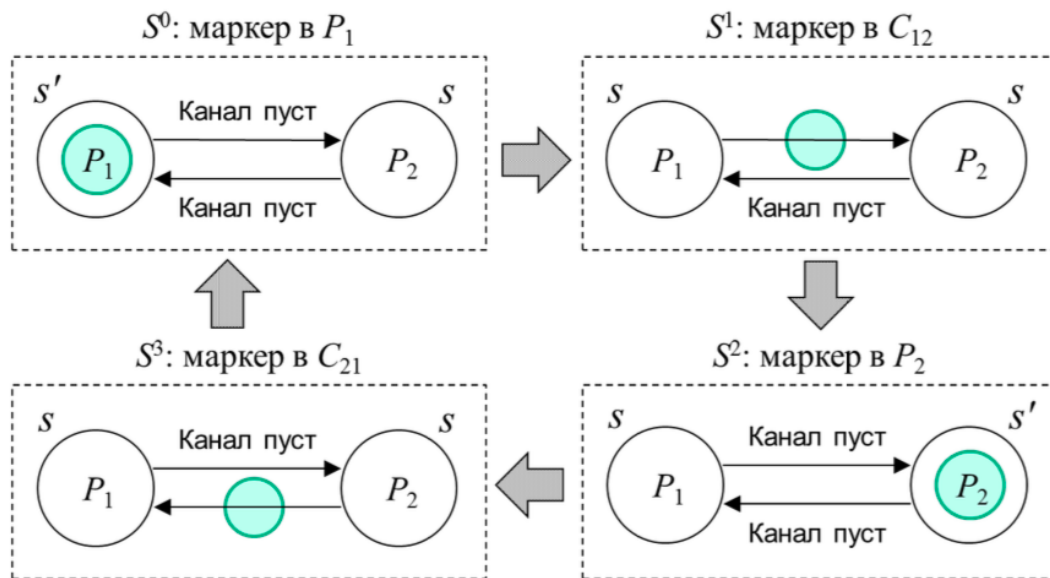
$$S^{x+1} = e^x(S^x)$$

Для ілюстрації виконання розподільної системи розберемо наступний приклад передачі токена. Нехай розподільна система складається з 2 процесів P_1, P_2 та двох каналів C_{12}, C_{21}



Токен – це унікальний об’єкт, що процеси передають один одному за допомогою повідомлень. Кожен процес може або володіти токеном (s') в даний момент часу, або ні (s).

Глобальний стан розподільної системи та переходи між ними можна проілюструвати так:



Асинхронні розподільні системи.

Більшість розподільних систем по своїй природі є асинхронними. В асинхронних системах не накладається ніяких обмежень на розглядаємі характеристики, що ми зараз перерахуємо.

- По-перше швидкість виконання будь-кого окремого процесу кінцева, але не має відомої верхньої границі.
- Повідомлення може бути доставлено через довільний, але кінцевий час після його відправки. Повідомлення від процесу А до процесу В може бути відправлено за декілька мілісекунд, а від процесу В до процесу С за декілька хвилин.
- Швидкість відхилення годинників від стандарту також може бути довільною.

Основна причина використання асинхронних розподільних систем є те, що багато процесів використовують 1 процесор, а декілька логічних каналів зв'язку використовують одне фізичне з'єднання. Відповідно, коли багато різних процесів, різної природи та поведінки працюють на одному й тому самому процесорі, результуюча швидкість та ефективність кожного з

них не може бути гарантована. Це в точності описує роботу веб-додатків, де, по суті, навантаження на сервери та мережеві з'єднання не може бути обмеженою. Саме тому не можна передбачити скільки часу потребує передача файлу по протоколу FTP з серверу на комп'ютер користувача. Необхідно також відмітити, що найбільший вклад у затримку доставки повідомлень робить не час поширення сигналу, а час, витрачений різними рівнями стека протоколів на обробку операцій відправки та отримання, а також затримка, що вноситься проміжними вузлами на шляху передачі повідомлення.

Перевагою асинхронної системи це те, що побудовані на її основі розподільні алгоритми є більш стійкими до зміни умов їх функціонування, та, як наслідок, є більш універсальними. Однак, існує багато архітектурних та алгоритмічних проблем, що не можуть бути вирішеними в умовах асинхронної системи.

Є класичний приклад – задача двох генералів, що ілюструє одну з проблем в асинхронних системах.

Дві армії, кожна має свого генерала готується до штурму міста. Табори обох армій розташовані на двох пагорбах, що розділені степом. Єдиний спосіб зв'язку між генералами є надсилання повідомлень через степ. Припустимо, що посильні, що переносять повідомлення не можуть бути перехоплені, або вбиті. Для успішного штурму генерали повинні атакувати місто разом. При цьому необхідно узгодити хто буде головою штурму та коли необхідно його розпочати.

Легко побачити, що навіть в асинхронній системі генерали можуть узгодити хто буде головою штурму (наприклад головою буде той, в кого більше солдатів, або той, хто старший і так далі). Але ось питання: коли починати штурм? В асинхронних системах посильні можуть йти до іншого

штабу скільки завгодно часу, хоч 10 хвилин, хоч 3 години. В синхронних системах проблема все ще актуальна, але її можна вирішити.

Наприклад, генерал 1 армії висилає повідомлення «Атакуймо!», в синхронній системі він не знає точно, скільки часу повідомлення буде доходити, але знає, що воно буде йти від *min* до *max* хвилин.

Упорядкування подій.

Як правило, точний час не настільки важливий, як порядок, в якому події відбувалися.

Розберемо це на прикладі розсилки повідомлень між користувачами А, В, С, D. В спільному чаті відбувається наступна переписка:

А: Давайте всі зберемося та підемо гуляти.

В: Гарна ідея! Але я не зможу, у мене прорвало труби в будинку(

С: Це жахливо! Але може ми зможемо піти без тебе?

Однак, у зв'язку з затримкою відправку повідомлень, для користувача А повідомлення перемішалися:

А: Давайте всі зберемося та підемо гуляти.

С: Це жахливо! Але може ми зможемо піти без тебе?

В: Гарна ідея! Але я не зможу, у мене прорвало труби в будинку(

В розподільних системах годинники не можуть бути синхронізовані з абсолютною точністю. Якщо розсинхронізація годинників велика, то якщо для кожної події вказувати реальний час, коли вона відбулася, через розсинхронізацію локальних годинників, ситуація, яку ми тільки що описали, може виникати дуже часто. Для уникнення проблеми, пов'язаної з використанням реального часу, Лемпорт запропонував модель логічного годинника, що відраховують логічний час, ця модель може бути застосована для впорядкування подій, що відбуваються у різних процесах розподільних та паралельних систем. Логічний час дозволяє розуміти

порядок подій, без знання про час цих подій. Питання, що пов'язані з логічним часом, ми розберемо на наступних лекціях.