Термінологія та базові алгоритми

У цій главі ми спершу ознайомимось із методичною базою за допомогою якої зможемо класифікувати та проаналізувати розподілені алгоритми, розглянемо декілька простих розподілених алгоритмів на графах, дізнаємось про синхронізатори, які забезпечують абстракцію синхронних систем над асинхронними системами. З рештою, подивимось на деякі практичні проблеми, щоб зрозуміти необхідність побудови ефективних розподілених алгоритмів.

Топологічна абстракція та накладання

Топологія розподілених систем можна розглядати як неорієнтований граф у якому вершини означають процесори, а ребра відповідають зв’язкам, що ці процесори об’єднують. Ваги на ребрах можуть відповідати за функцію оцінки, якої потребує наша програма. Зазвичай, виділяють три (хоча і не завжди різні) рівні топологічної абстракції, які можна використати для аналізу розподіленої системи або розподіленої програми. Розглянемо малюнок 5.1. Покажемо тільки найголовніші кінцеві хости, що використовуються програмою. WAN-мережі показані у формі пунктирних овалів. Комутаційні елементи всередині WAN-мереж, та інші кінцеві хости не вимальовані, незважаючи на їх належність до топологічного представлення. Так само, як і всі ребра, що з’єднують всі кінцеві вузли та всі ребра, що об’єднують всі комутатори всередині глобальних мереж відносяться до топологічного представлення, хоча показані лише деякі з них.

* Фізична топологія Вузли в цій топології представляють всі вузли мережі разом із комутаторами (або роутерами) в мережі та кінцевими вузлами, незалежно від їх ролі у додатку. Ребра в топології показують всі комунікаційні зв’язки у мережі в доповнення до всіх прямих зв’язків між кінцевими хостами.

На малюнку 5.1(а), фізична топологія не показана для збереження простоти малюнку.

Малюнок 5.1 Два приклади топологічного представлення на різних рівнях абстракції.

* Логічна топологія Цей вид топології зазвичай визначається у контексті конкретної програми. Вершини представляють всі кінцеві вузли, на яких виконується програма. Ребра в цій топології це логічні канали (або логічні посилання) серед усіх вузлів. Таке представлення розміщене на вищому рівні абстракції, ніж фізична топологія. До того ж, немає потреби додавати сюди вершини та ребра фізичної топології.

Зазвичай, логічні зв’язки моделюються між конкретними парами кінцевих вузлів, що приймають участь у роботі програми для отримання логічної топології з корисними властивостями. На малюнку 5.1(б) видно, що всі пари вузлів об’єднані задля отримання повністю зв’язаної мережі. Кожна пара вузлів може напряму звертатись до іншого учасника мережі використовуючи інцидентні логічні зв’язки на рівні абстракції топології. Однак, логічні зв’язки можуть задавати випадкову зв’язність (окіл-зв'язок) з вузлами у цій абстракції. На малюнку 5.1 (б) логічне представлення надає конкретному вузлу частковий топологічний вигляд, а отримана зв’язність забезпечується зв’язність в околах. Для того, щоб комунікувати з іншим вузлом системи, що не є логічним сусідом, вузол може зробити стрибок через декілька логічних зв’язків на цьому рівні абстракції топології.