

TÌM HIỂU VỀ HAI CƠ CHẾ CONCURRENCY VÀ PARALLELISM

1. Concurrency

Concurrency là thực thi các thread hoặc tác vụ cùng một khoảng thời gian nhưng không nhất thiết phải cùng một lúc. Có nghĩa là, 2 tác vụ có thể bắt đầu, chạy và hoàn thành trong một khoảng thời gian trùng với nhau. Nghĩa là tác vụ 2 có thể bắt đầu ngay cả khi tác vụ 1 chưa kết thúc. Điều này không ảnh hưởng đến kết quả của các quá trình này so với bình thường. Có nhiều mô hình toán học đã được tạo ra để tổng quát hoá mô hình như Petri nets, process calculi,...

Có nhiều cách để đạt được concurrency. Đối với các bộ xử lý đơn luồng, concurrency có thể đạt được bằng context-switching, có nghĩa là thực hiện luân phiên các tác vụ và đổi sang tác vụ khác sau một khoảng thời gian. Đối với các bộ xử lý đơn luồng, concurrency có thể đạt được bằng parallelism.

Ngoài ra, các phép tính toán trong hệ thống concurrent có thể tương tác với nhau trong quá trình thực thi, do đó số khả năng của luồng thực thi là rất lớn và kết quả xảy ra trở nên không xác định được. Việc sử dụng chung tài nguyên cũng dẫn đến tính không xác định và từ đó dẫn đến các lỗi như deadlock, starvation.

2. Parallelism

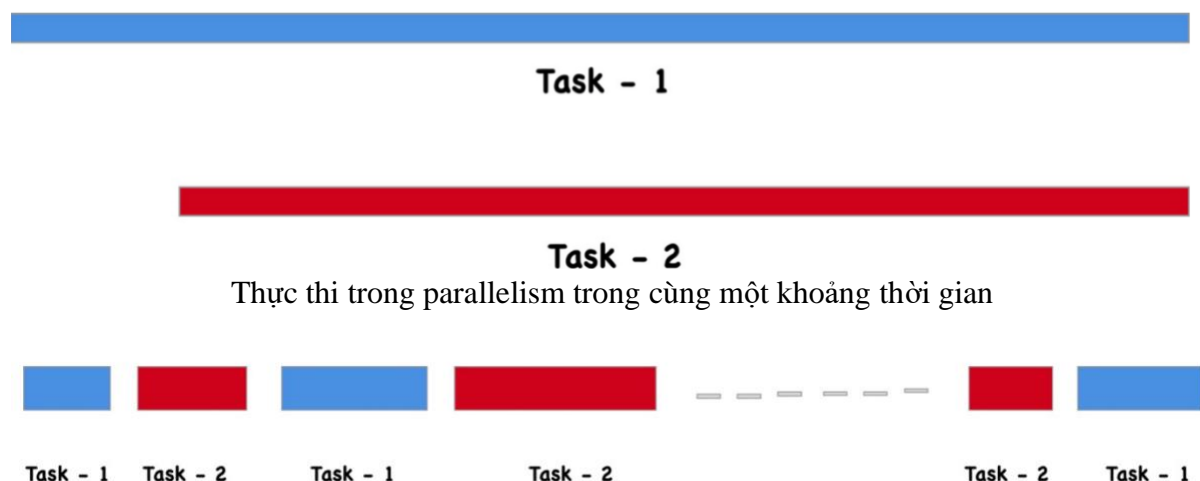
Parallelism có nghĩa là có 2 hoặc nhiều hơn các tác vụ được thực hiện trong cùng một khoảng thời gian, đây là một phần tử con của concurrency. Để đạt được parallelism đòi hỏi phải có nhiều bộ xử lý hay CPU có nhiều nhân. Một ví dụ là bạn có thể thực hiện giải phương trình trên nhân thứ nhất và thực hiện xem video trên nhân còn lại.

Việc thực hiện parallelism có thể nâng cao tốc độ xử lý kết quả, tuy nhiên, vấn đề chia sẻ dữ liệu cần phải được lưu ý để tránh thực hiện 2 lần tác vụ đó trên các bộ xử lý khác nhau. Bên cạnh đó, parallelism cũng không có sự tương tác giữa các tác vụ thực hiện trong các bộ xử lý khác nhau trong quá trình thực thi.

GPU là một ví dụ tiêu biểu cho parallelism.

3. So sánh

Concurrency so với parallelism cũng như ngã tư so với 2 làn đường. Điều này thể hiện rõ hơn đối với concurrency trên 1 bộ xử lý.



Thực thi trong concurrency trên 1 bộ xử lý

Ngoài ra, như đã nói ở trên, parallelism không tương tác với nhau trong quá trình thực thi, ngược lại với concurrency.

Cả 2 đều có thể tăng tốc độ xử lý nhiều tác vụ. Đối với parallelism, càng nhiều nhân, càng có thể thực hiện được nhiều tác vụ, cho nên rất hay được sử dụng hiện nay. Còn đối với concurrency, các tác vụ sẽ được thực hiện rất nhanh nếu ta có một bộ xử lý to và mạnh duy

nhất. Cho nên có thể xuất hiện trên các cấu trúc single instruction, multiple data (SIMD), và một số trường hợp đặc thù.

Tham khảo

- [1] <https://softwareengineering.stackexchange.com/questions/190719/the-difference-between-concurrent-and-parallel-execution>
- [2] <https://medium.com/swift-india/concurrency-parallelism-threads-processes-async-and-sync-related-39fd951bc61d>
- [3] https://wiki.haskell.org/Parallelism_vs._Concurrency