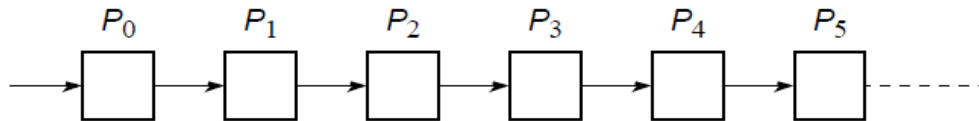


# Lab 8 Parallel&Distributed Computing Techniques

## *Pipelined Computations*

### 1 Giới thiệu

Trong kỹ thuật pipeline, bài toán được chia thành một chuỗi các công việc, các công việc này kết thúc nối tiếp lẫn nhau. Việc song song hóa bài toán theo pipeline cũng được xem như là một dạng của song song theo chức năng (functional decomposition). Hình sau minh họa hoạt động của mô hình pipeline:

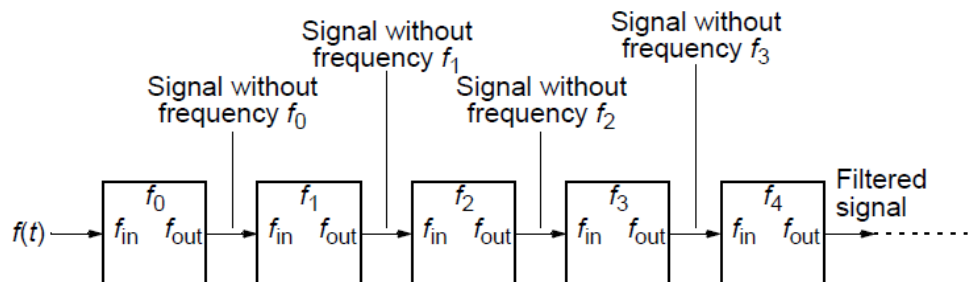


Kỹ thuật lập trình pipeline được sử dụng rộng rãi trong các bài toán thực tế. Các trường hợp ứng dụng kỹ thuật pipeline có thể chia làm 3 loại:

- “ 1. If more than one instance of the complete problem is to be executed
2. If a series of data items must be processed, each requiring multiple operations
3. If information to start the next process can be passed forward before the process has completed all its internal operations ”

Barry WilKinson & Michael Allen

Một ví dụ về sử dụng kỹ thuật pipeline đó là việc lọc tần số của một tín hiệu  $f(t)$  như minh họa ở hình bên dưới, mỗi một giai đoạn của pipeline sẽ loại bỏ một tần số  $f_i$  nào đó. Kết quả đạt được của quá trình là tần số cần sử dụng.



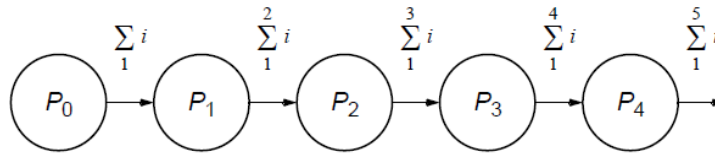
☺ SV tìm hiểu cách song song hóa bài toán theo mô hình pipeline.

### 2 Nội dung

Một số bài tập minh họa cho mô hình lập trình pipeline, tương ứng với 3 lớp bài toán đã trình bày ở trên, sv chọn và hiện thực một bài tập cho phần lab này !

## 2.1 Tính tổng

Xét bài toán tính tổng một dãy số theo mô hình pipeline :



Tính toán pipeline loại 1

Chương trình hiện thực có dạng như sau:

```
if ( rank > 0 ) { // rank là id của mỗi process
```

```
    Nhận sum từ Pi-1
```

```
    sum += number;
```

```
}
```

```
if (rank < size - 1) {
```

```
    Gửi sum đã tính đến Pi+1
```

```
}
```

☺ SV nhận xét xem bài toán này giải bằng mô hình pipeline thì khác biệt thế nào với phương pháp phân chia dữ liệu đã thực hiện ở lab 4.

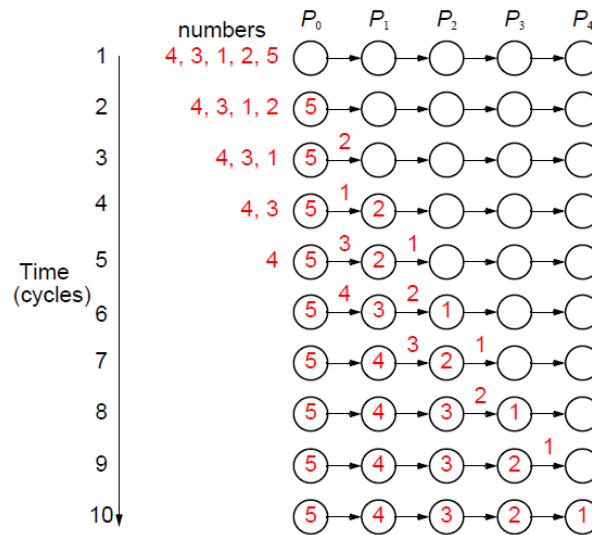
## 2.2 Sắp xếp theo phương pháp insertion sort

Kỹ thuật pipeline là một trong các kỹ thuật được sử dụng để sắp xếp khi dữ liệu đầu vào lớn, các giải thuật tuần tự không còn hiệu quả. Ý tưởng của giải thuật có thể mô tả đơn giản như sau: P<sub>0</sub> sẽ nhận một chuỗi dữ liệu vào trong mỗi lần thực thi, lưu giữ lại giá trị lớn nhất (hay nhỏ nhất) và chuyển các số còn lại sang P<sub>1</sub>. Các P<sub>i</sub> kế tiếp sẽ thực hiện kiểm tra giá trị mà nó đang lưu - x, nếu x lớn hơn giá trị nhận được thì sẽ lưu lại x. Ngược lại sẽ gán giá trị mới cho x và gửi giá trị cũ đến P<sub>i+1</sub>.

Hiện thực code của chương trình sẽ có dạng như sau:

```
recv(&n, Pi-1)
if( n > x ) {
    send (&x, Pi+1);
    x = n ;
}
else send (&n, Pi+1);
```

Hình bên dưới minh họa các bước sắp xếp theo mô hình pipeline:



Các bước sắp xếp với 5 số

☺ Bài toán sắp xếp theo mô hình pipeline có đảm bảo các processor luôn bận rộn (busy) ?

### 3 Bài tập

3.1 Viết chương trình tính  $x^{16}$  theo mô hình pipeline ( $x > 1$ )

3.2 Viết chương trình song song theo mô hình pipeline tính :

$$\sin \theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \frac{\theta^9}{9!} - \dots$$

Với một chuỗi các giá trị đầu vào  $\theta_1, \theta_2, \theta_3, \dots$

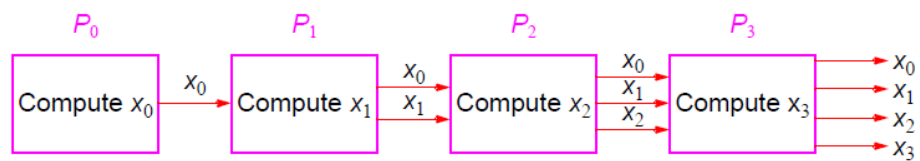
3.3 SV tìm hiểu về bài toán giải hệ phương trình tuyến tính có ma trận hệ số là ma trận tam giác trên theo mô hình pipeline.

*Hướng dẫn:*

Một ma trận tam giác trên (upper-triangular) sẽ có dạng như sau:

$$\begin{array}{rcl}
 a_{n-1,0}x_0 + a_{n-1,1}x_1 + a_{n-1,2}x_2 & \dots & + a_{n-1,n-1}x_{n-1} \\
 & & \cdot \\
 & & \cdot \\
 a_{2,0}x_0 + a_{2,1}x_1 + a_{2,2}x_2 & & = b_2 \\
 a_{1,0}x_0 + a_{1,1}x_1 & & = b_1 \\
 a_{0,0}x_0 & & = b_0
 \end{array}$$

Các biến  $x_0, x_1, x_2 \dots$  sẽ được tính theo thứ tự từ dưới lên theo mô hình pipeline :



☺ SV phải nộp mã nguồn chương trình của một bài tập lên sakai.