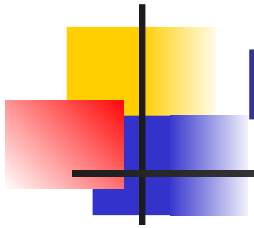
IT4853
Tìm kiếm và trình diễn thông tin

Bài 23. Thu thập dữ liệu

IIR.C20. Web crawling and indexes

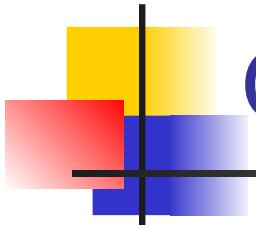
TS. Nguyễn Bá Ngọc, *Bộ môn Hệ thống thông tin, Viện CNTT & TT*
ngocnb@soict.hust.edu.vn

Hà Nội, 2016



Nội dung chính

- Các thao tác thu thập dữ liệu cơ bản
- Bộ thu thập dữ liệu Web



Các thao tác cơ bản

- Khởi tạo hàng đợi với tập mầm URLs
- Lặp:
  - Lấy URL từ hàng đợi;
  - Nạp và đọc trang web;
  - Tách URLs từ trang web;
  - Thêm URLs vào hàng đợi.

Giả thuyết cơ bản: Web là đồ thị liên thông.

Các thao tác cơ bản (2)

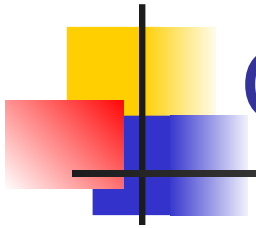
```
urlqueue := (some carefully selected set of seed urls)
while urlqueue is not empty:
    myurl := urlqueue.getlastanddelete()
    mypage := myurl.fetch()
    fetchedurls.add(myurl)
    newurls := mypage.extracturls()
    for myurl in newurls:
        if myurl not in fetchedurls and not in urlqueue:
        urlqueue.add(myurl)
    addtoinvertedindex(mypage)
```

Hạn chế của bộ thu thập này là gì?

Phương hướng cải tiến bộ thu thập đơn giản

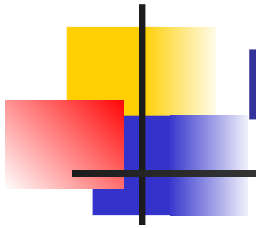
- Quy mô:
  - Cần phân tán quá trình thu thập.
- Lựa chọn nội dung:
  - Không thể đánh chỉ mục tất cả, tích hợp khả năng phát hiện trùng lặp và spam.
- Nguyên tắc lịch thiệp (politeness):
  - Không truy cập quá thường xuyên đến một máy chủ, cần thời gian nghỉ giữa những yêu cầu gửi tới một địa chỉ.
- Tính cập nhật:
  - Cần thu thập lại theo chu kỳ;
  - Web rất lớn, chỉ có thể thường xuyên thu thập một phần nhỏ.

Vấn đề xác định độ ưu tiên là cấp thiết



Quy mô của bài tóan thu thập

- Nạp 20,000,000,000 trang mỗi tháng . . .
- . . . cần nạp khoảng 8000 trang mỗi giây!
- Thực tế có thể phức tạp hơn, vì có nhiều trang thu được là trùng lặp, không tải được, spam v.v.



Robots.txt

- Giao thức hạn chế quyền truy cập đối với trình duyệt web tự động (“robots”), được thiết lập từ 1994;
- Ví dụ:
  - User-agent: *

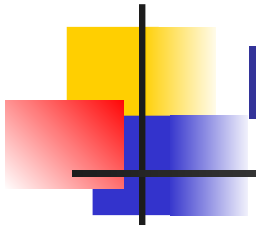
    Disallow: /yoursite/temp/
  - User-agent: searchengine

    Disallow: /

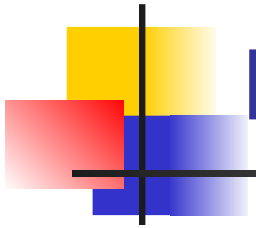
Ví dụ robots.txt (nih.gov)

```
User-agent: PicoSearch/1.0
Disallow: /news/information/knight/
Disallow: /nidcd/
...
Disallow: /news/research_matters/secure/
Disallow: /od/ocpl/wag/
User-agent: *
Disallow: /news/information/knight/
Disallow: /nidcd/
...
Disallow: /news/research_matters/secure/
Disallow: /od/ocpl/wag/
Disallow: /ddir/
Disallow: /sdminutes/
```

Yêu cầu đối với bộ thu thập dữ liệu Web



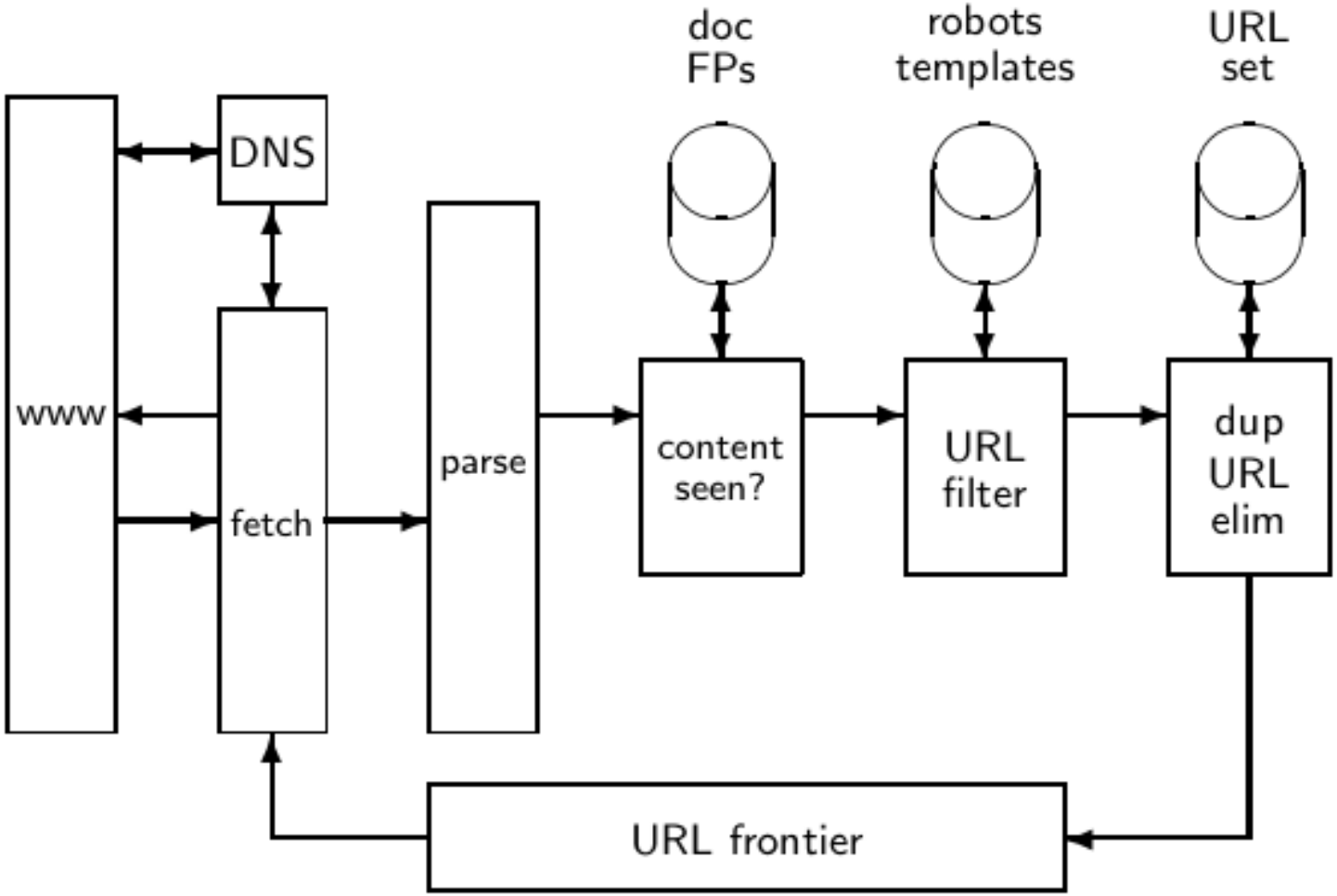
- Thiết kế hệ thống phân tán, sử dụng đồng thời nhiều luồng thu thập
- Khả mở:
  - Dễ dàng mở rộng quy mô thu thập bằng cách bổ xung thêm nhiều máy
- Nạp những trang chất lượng cao trước
- Thu thập liên tục
  - Thu thập phiên bản mới của những trang đã biết



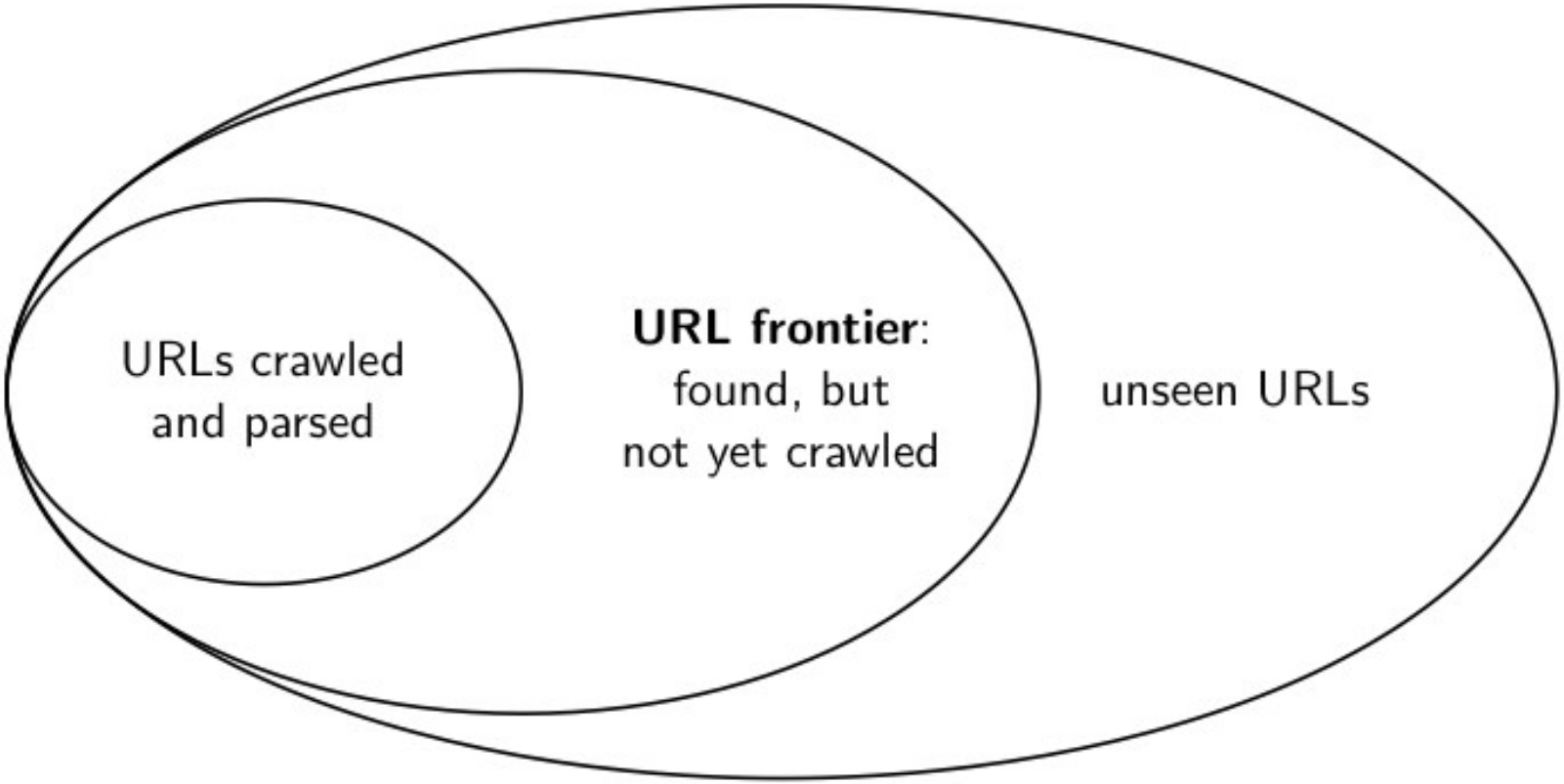
Nội dung chính

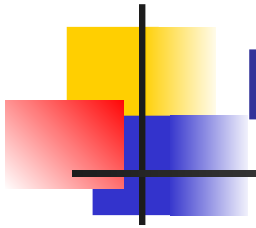
- Các thao tác thu thập dữ liệu cơ bản
- Bộ thu thập dữ liệu Web

Kiến trúc tổng quát của bộ thu thập



Hàng đợi URL

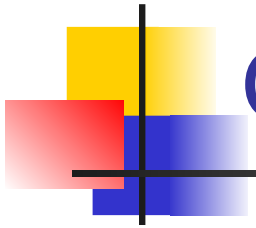

URLs crawled
and parsed
URL frontier:
found, but
not yet crawled
unseen URLs



Hàng đợi URL (2)

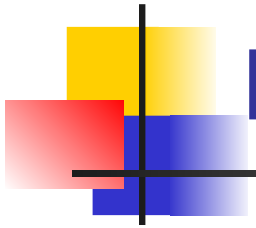
- Hàng đợi URL là cấu trúc dữ liệu lưu trữ và quản lý URLs đã phát hiện, nhưng chưa được thu thập
- Có thể bao gồm nhiều trang từ một máy chủ
  - Chánh nạp tất cả cùng lúc;
- Cần sử dụng tất cả các phân luồng thu thập

Hàng đợi URL: URL frontier



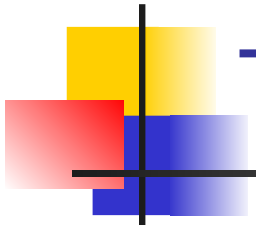
Chuẩn hóa URL

- Có nhiều URLs được trích rút từ tài liệu là những URLs tương đối.
- Ví dụ, trong http://mit.edu, địa chỉ aboutsite.html
  - Tương đương với: http://mit.edu/aboutsite.html
- Cần phải chuẩn hóa tất cả các URLs tương đối thành dạng tuyệt đối.



Nội dung đã xem

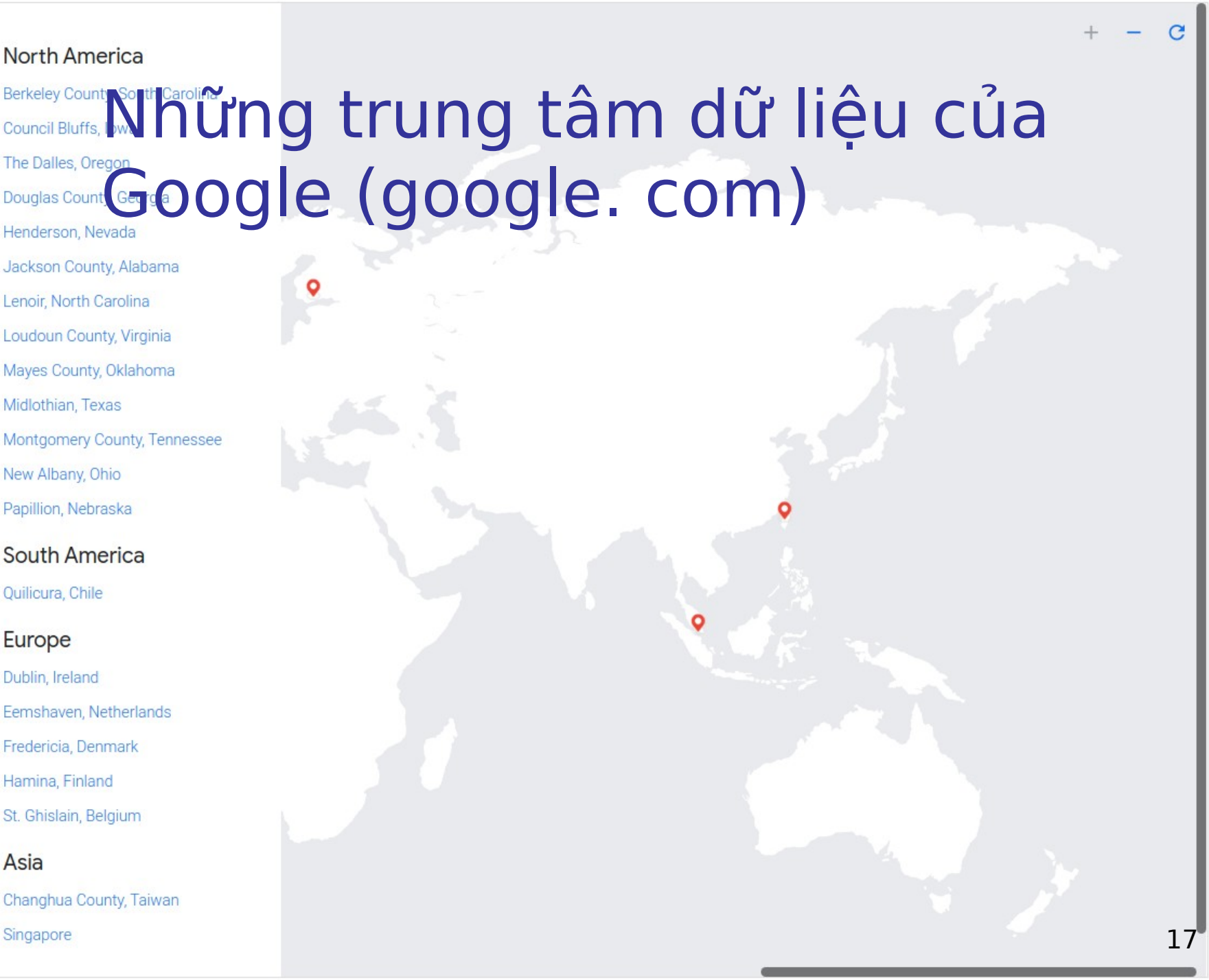
- Với mỗi trang được nạp: Kiểm tra liệu nội dung đã có trong chỉ mục
- Kiểm tra dựa trên tổng đại diện hoặc biểu diễn khung
- Bỏ qua những tài liệu có nội dung đã được đánh chỉ mục



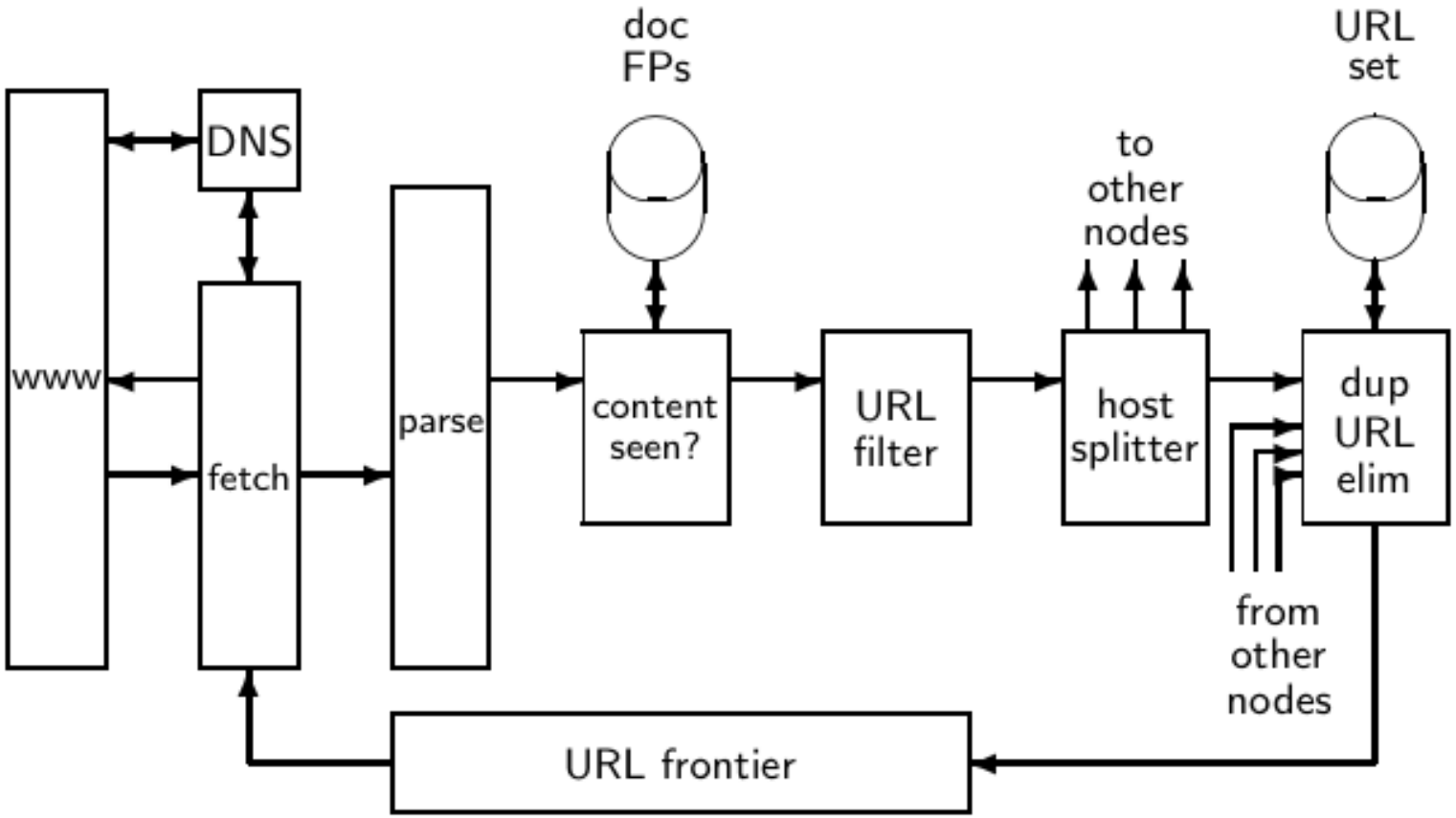
Thu gom phân tán

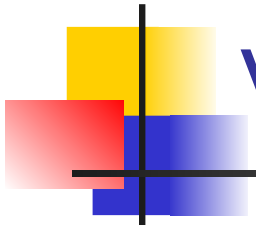
- Chạy nhiều phân luồng thu thập trên nhiều nút khác nhau đặt ở các vị trí khác nhau.
  - VD, Google thực hiện phân tán hệ thống thu thập theo vị trí địa lý
- Phân chia các máy chủ chứa dữ liệu thu thập cho các nút khác nhau
  - Mỗi nút đảm nhiệm việc thu thập từ một cụm máy chủ.

Những trung tâm dữ liệu của Google (google. com)



Thu gom dữ liệu phân tán

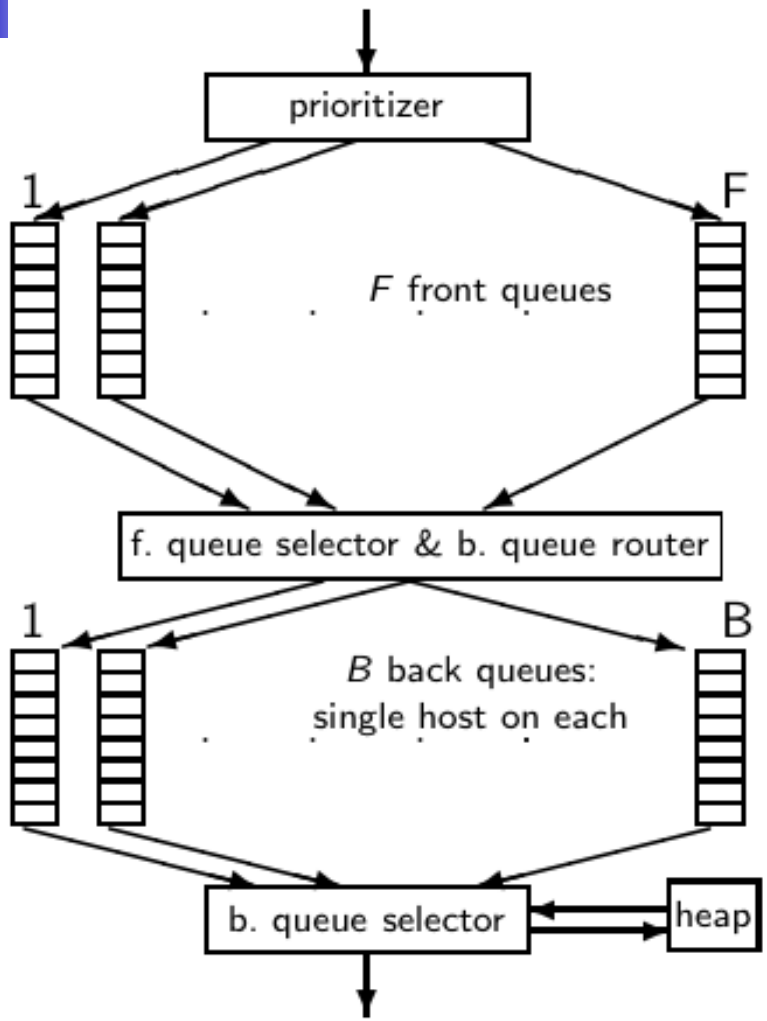




Vai trò của hàng đợi URL

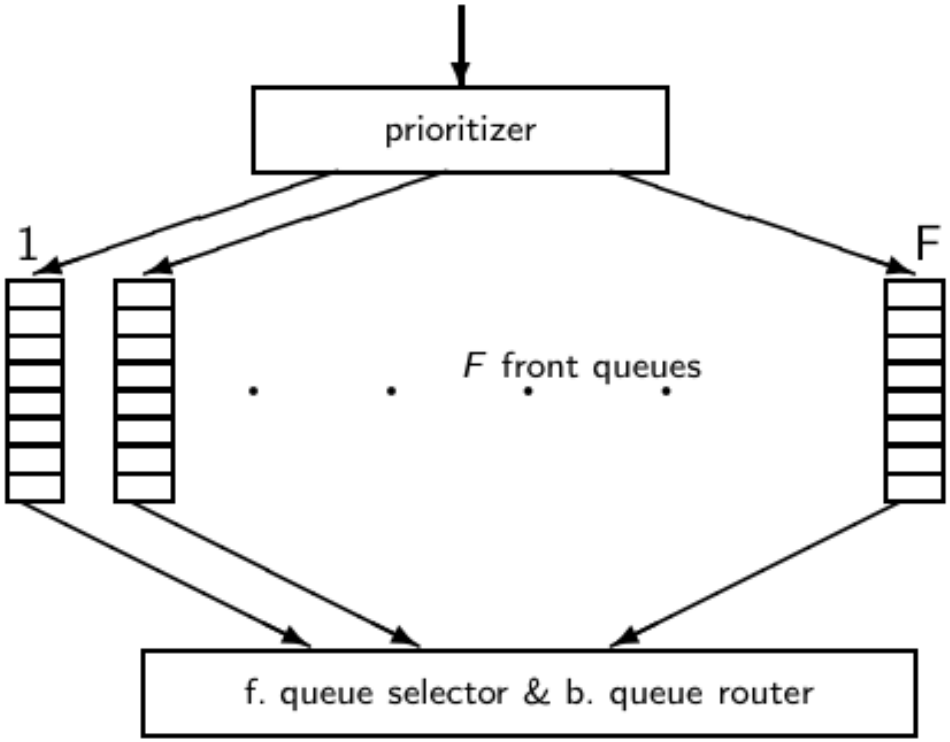
- Sự lịch thiệp: Đảm bảo không truy cập một máy chủ web quá thường xuyên
  - Ví dụ, chèn một khoảng thời gian giữa hai yêu cầu thành công được gửi đến cùng một máy chủ
- Tính cập nhật:
  - Đảm bảo tính ưu tiên cho những trang quan trọng, thường xuyên thay đổi.
    - Đây là vấn đề khó, hàng đợi thông thường không giải quyết được vấn đề này.

Hàng đợi URL của Mercator



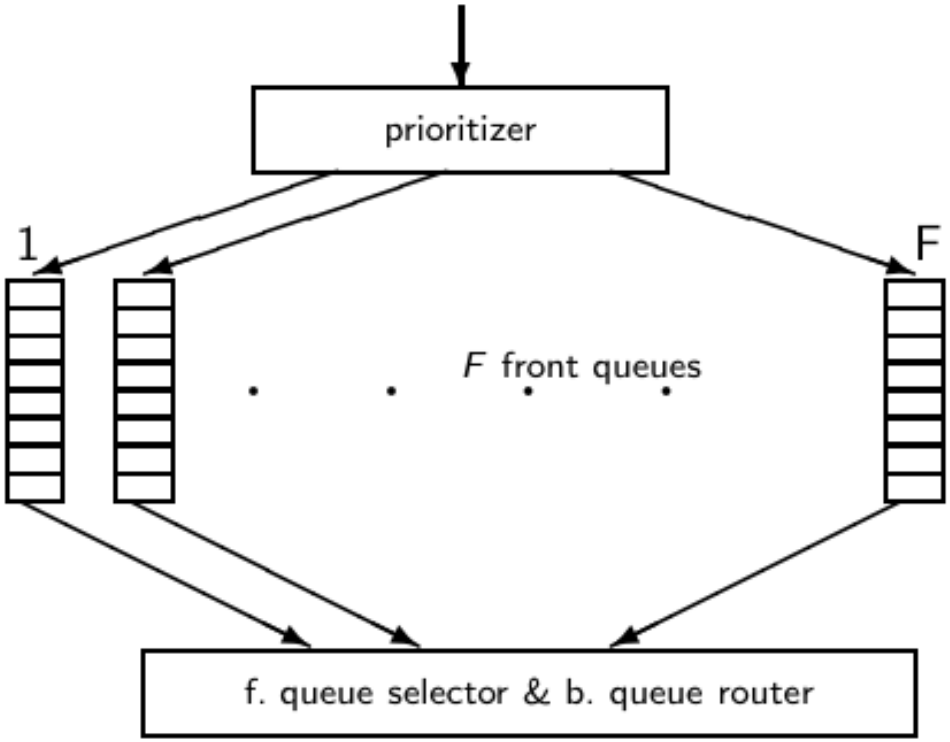
- Luồng URLs tới bộ nạp phải qua hai hàng đợi: phía trước và phía sau.
- Hàng đợi phía trước quản lý độ ưu tiên.
- Hàng đợi phía sau đảm bảo sự lịch thiệp.
- Các hàng đợi là FIFO.

Hàng đợi phía trước



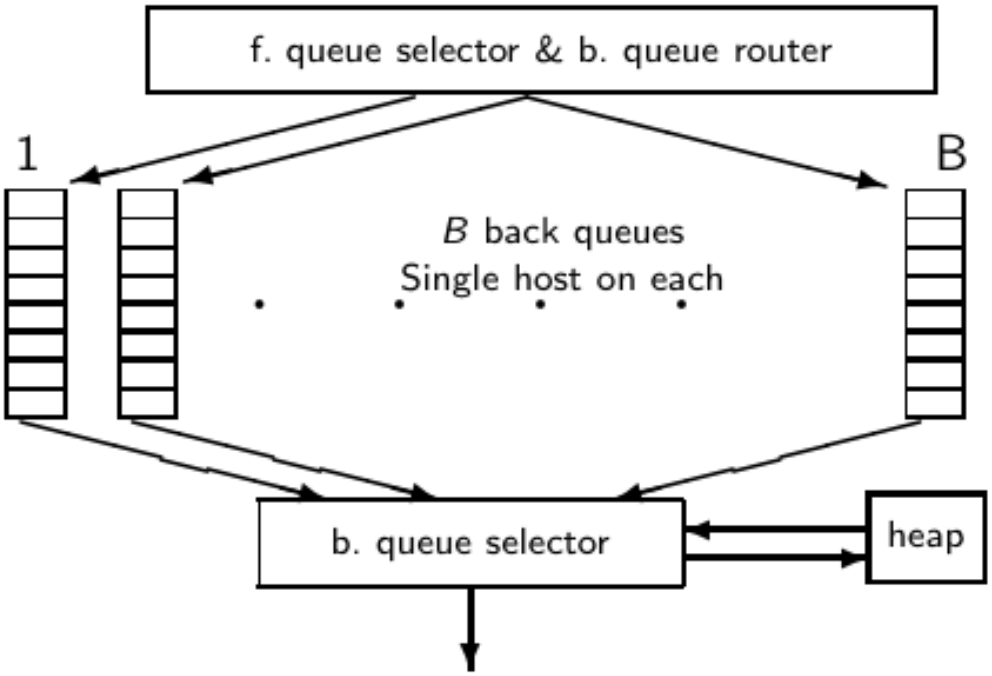
- Bộ ưu tiên gán cho mỗi URL một độ ưu tiên nguyên trong khoảng từ 1 đến F.
- Sau đó thêm URL vào hàng đợi tương ứng
- Xác định độ ưu tiên bằng giải thuật tham lam: tốc độ cập nhật, PageRank v.v.

Hàng đợi phía trước

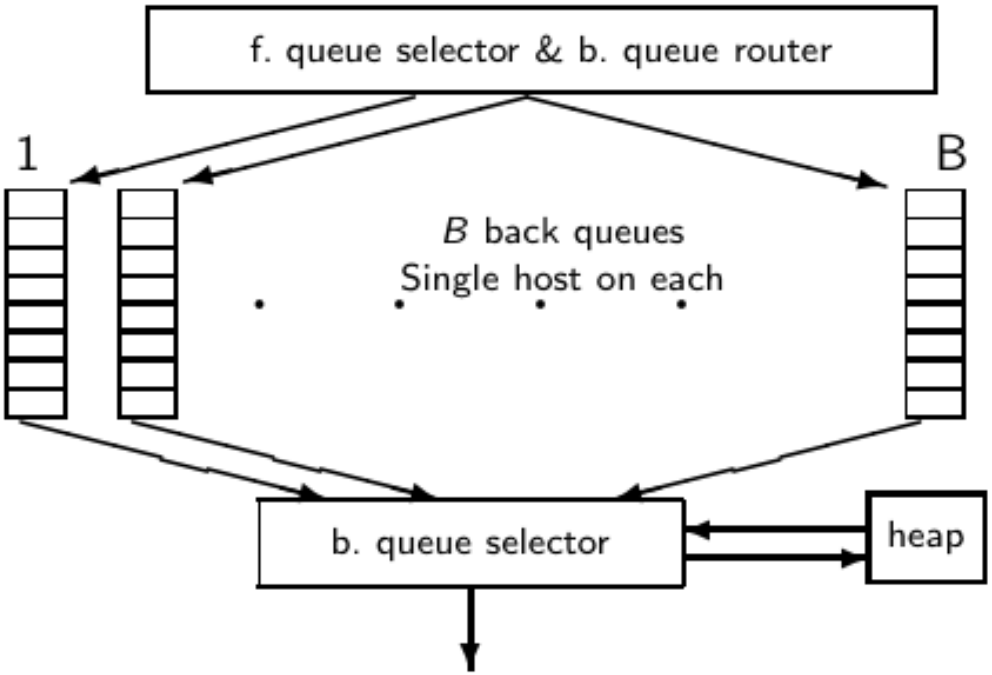


- Hàng đợi phía sau gửi yêu cầu tới hàng đợi phía trước
- Chọn một hàng đợi phía trước: Theo vòng, ngẫu nhiên, v.v. , đảm bảo sự ưu tiên đối với hàng đợi có mức ưu tiên cao
- Lấy ra URL tiếp theo

Hàng đợi phía sau

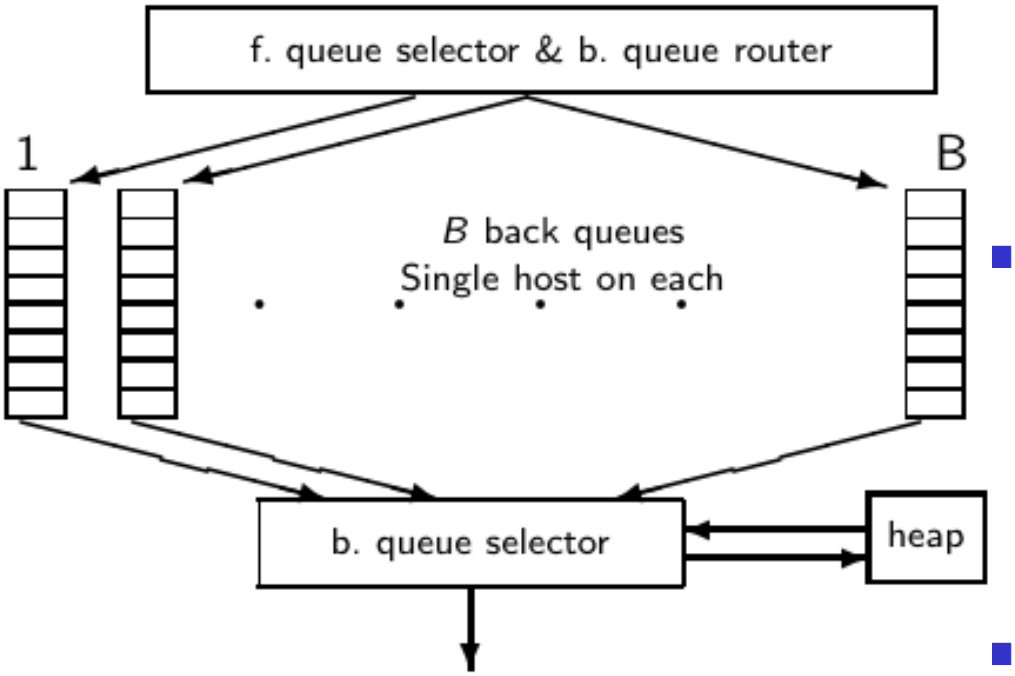


Hàng đợi phía sau



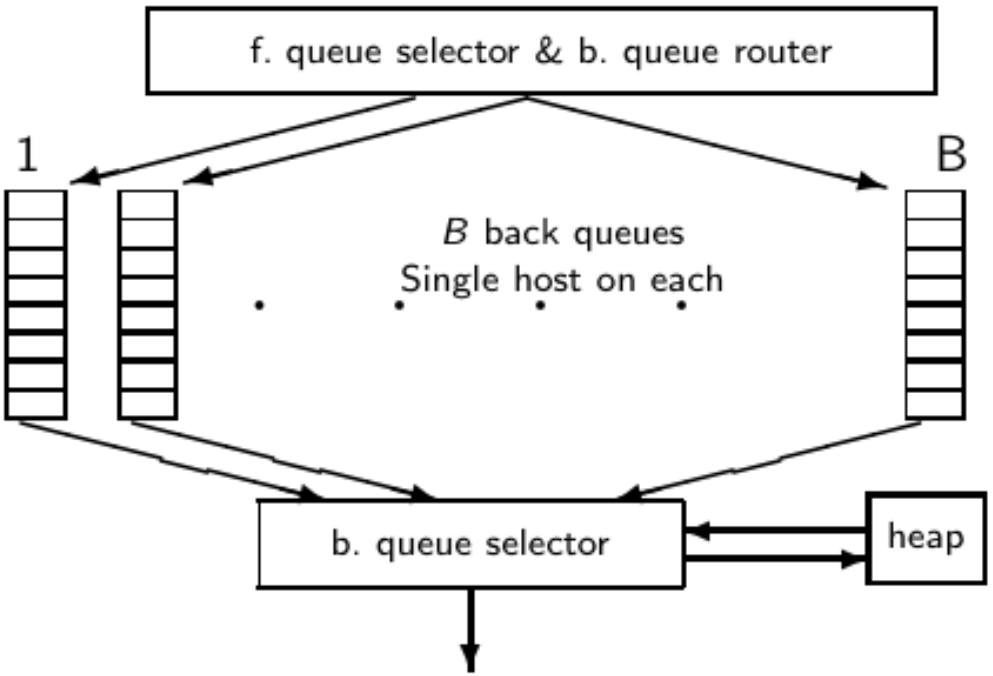
- Nguyên tắc 1. Mỗi hàng đợi phía sau được đảm bảo khác rỗng cho tới khi kết thúc thu thập.
- Nguyên tắc 2. Mỗi hàng đợi phía sau chỉ chứa những URL từ một máy chủ.
- Duy trì một bảng tham chiếu các máy chủ tới các hàng đợi

Hàng đợi phía sau



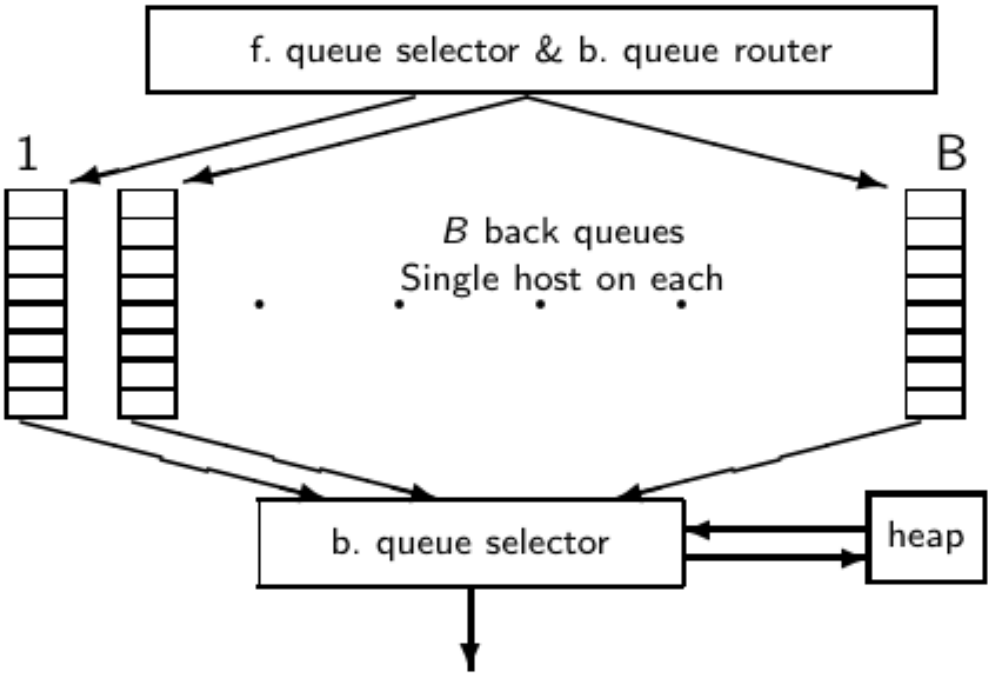
- Hệ thống còn lưu trong bộ nhớ heap một thời gian đợi cho mỗi hàng đợi phía sau
- Thời gian đợi là thời gian te sớm nhất có thể gửi yêu cầu tới máy chủ tương ứng của hàng đợi phía sau.
- Thời gian te sớm nhất được xác định dựa trên thời gian xử lý cuối cùng.

Hàng đợi phía sau

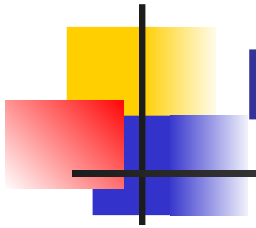


- Bộ thu thập giao tiếp với hàng đợi phía sau như thế nào?
  - Lặp (i) lấy URL từ q hiện tại (q là một hàng đợi phía sau)
  - (ii) nạp URL u vào đầu hàng đợi q

Hàng đợi phía sau



- Nếu q trở thành rỗng
  - Lặp (i) lấy những URL u từ hàng đợi phía trước và (ii) thêm u vào hàng đợi phía sau tương ứng của nó
  - Nếu u không có hàng đợi phía sau tương ứng, thì (i) tạo một hàng đợi mới, (ii) đưa u vào đó và (iii) thiết lập thời gian đợi cho hàng đợi mới tạo.



Bài tập 23.1

- Vì sao phân chia khối lượng thu thập cho các nút của hệ thống thu thập phân tán theo máy chủ (host) tốt hơn so với phân chia theo URLs?
- Tại sao bộ phân chia máy chủ nên đứng trước bộ loại bỏ trùng lặp URL trong tiến trình thu thập?

Bài tập 23.2

- Chúng ta thiết lập hằng số tăng t_e bằng 10 lần thời gian nạp lần cuối cùng, và số lượng hàng đợi phía sau bằng 3 lần số luồng thu thập. Hai hằng số này liên quan với nhau như thế nào?