

Tìm kiếm thông tin

Chương 10. Thu thập dữ liệu Web và chỉ mục ngược quy mô lớn

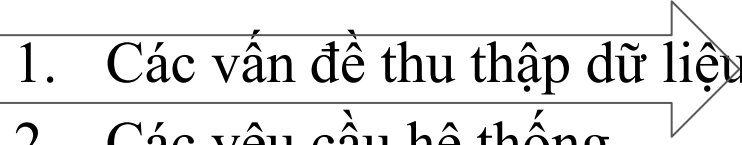
Soạn bởi: TS. Nguyễn Bá Ngọc

2021

Nội dung

1. Các vấn đề thu thập dữ liệu Web
2. Các yêu cầu hệ thống
3. Tổng quan hệ thống thu thập dữ liệu
4. Phân giải DNS
5. Hàng đợi URL
6. Máy chủ liên kết
7. Chia nhỏ và phân tán chỉ mục ngược
8. Lưu trữ tài liệu quy mô lớn

Nội dung

- 
1. Các vấn đề thu thập dữ liệu Web
 2. Các yêu cầu hệ thống
 3. Tổng quan hệ thống thu thập dữ liệu
 4. Phân giải DNS
 5. Hàng đợi URL
 6. Máy chủ liên kết
 7. Chia nhỏ và phân tán chỉ mục ngược
 8. Lưu trữ tài liệu quy mô lớn

Giải thuật loang trên đồ thị Web

Ý tưởng: Bắt đầu với 1 tập trang Web và từng bước mở rộng với các URL trong các trang hiện có.

Giải thuật:

- Khởi tạo hàng đợi với tập mầm URLs
- Lặp nếu hàng đợi không rỗng:
 - Lấy URL từ hàng đợi;
 - Nạp trang Web
 - Đọc nội dung trang Web;
 - Tách nội dung và các URLs từ trang Web;
 - Thêm URLs mới vào hàng đợi.

(Có lẽ đây là giải thuật thu thập thô sơ nhất)

Giải thuật loang trên đồ thị Web (2)

url_queue := (tập URLs ban đầu)

while url_queue is not empty:

 cur := url_queue.get_last_and_delete()

 page := fetch(cur)

 fetched_urls.add(cur)

 urls := extract_urls(page)

 for u in urls:

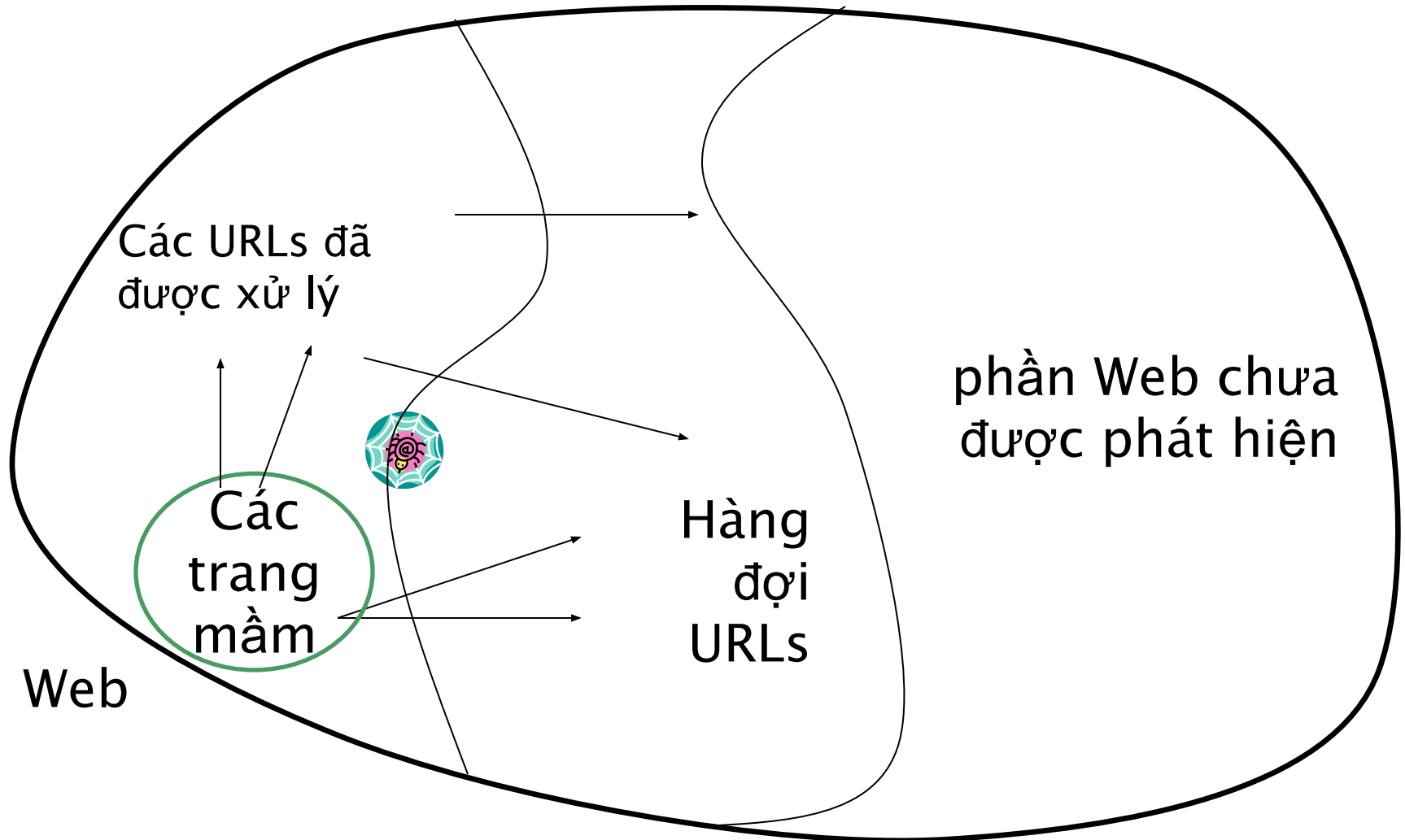
 if u not in fetched_urls and not in url_queue:

 url_queue.add(u)

 add_to_inverted_index(page)

Các hạn chế của giải thuật này?

Toàn cảnh thu thập



Các vấn đề thu thập dữ liệu Web

- Không thể thu thập nhiều dữ liệu Web chỉ với 1 máy
 - => Cần phân tán tiến trình thu thập
- Không phải tất cả dữ liệu Web đều hữu ích
 - Cần phát hiện các trang gian lận,
 - Bẫy thu thập,
 - Các nội dung trùng lặp, các bản sao của một trang Web
 - v.v..
- Các vấn đề kỹ thuật
 - Độ trễ/băng thông, kết nối mạng
 - Các giới hạn đối với nội dung
 - Được thu thập những nội dung nào?

Nội dung

1. Các vấn đề thu thập dữ liệu Web
2. Các yêu cầu hệ thống
3. Tổng quan hệ thống thu thập dữ liệu
4. Phân giải DNS
5. Hàng đợi URL
6. Máy chủ liên kết
7. Chia nhỏ và phân tán chỉ mục ngược
8. Lưu trữ tài liệu quy mô lớn

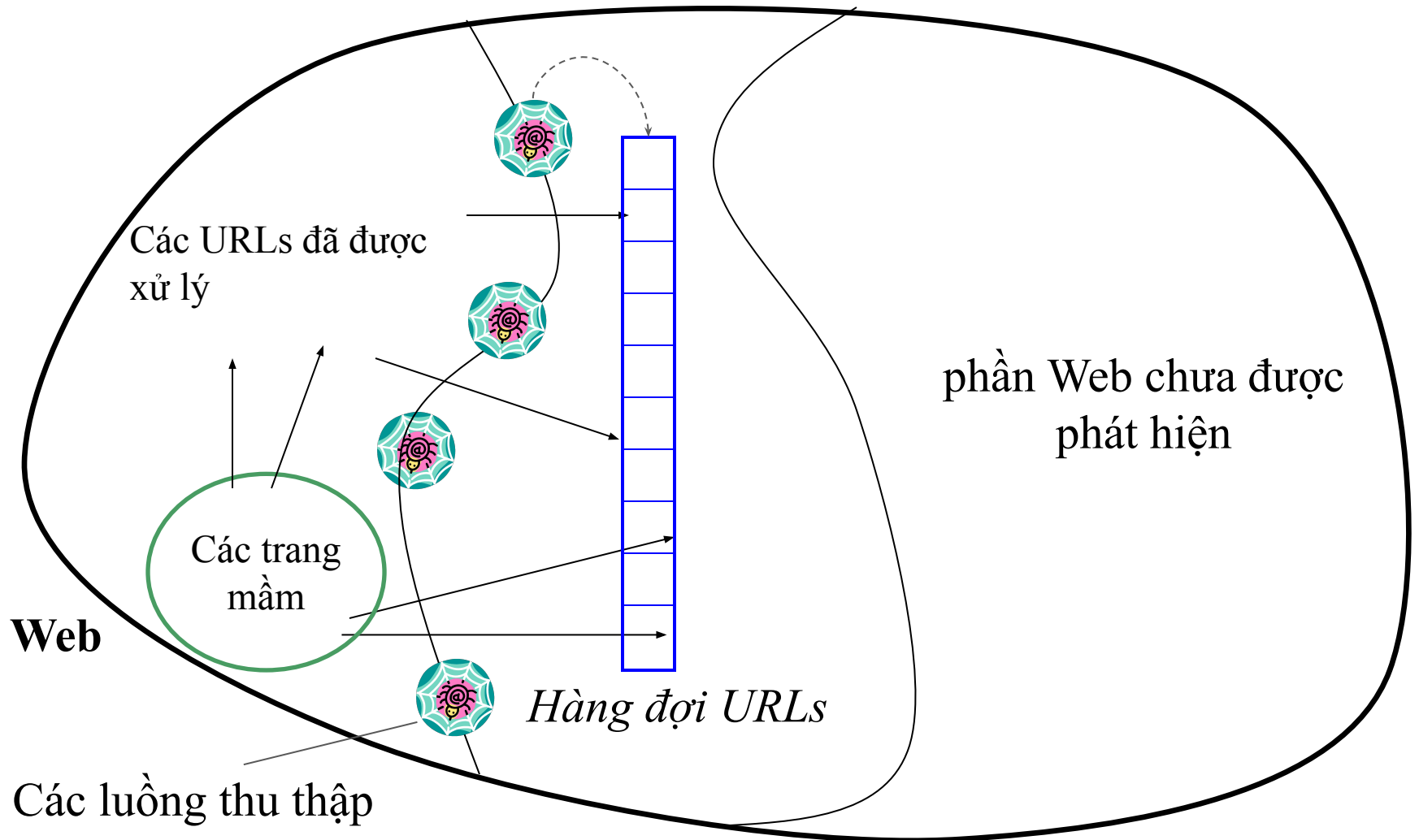
Các tính năng phải có

- Lọc nội dung theo giới hạn truy cập
 - Giao thức robots.txt
 - Chỉ thu thập các trang được phép.
- Kiểm soát tần suất truy cập
 - Không gửi yêu cầu quá thường xuyên tới 1 máy chủ
 - Tránh làm ảnh hưởng (làm chậm) dịch vụ của máy được thu thập.
- Có khả năng phát hiện và thoát các bẫy thu thập và các hành vi gian lận khác từ phía máy chủ Web.
- Có khả năng khôi phục và tiếp tục tiến trình thu thập sau khi bị gián đoạn
- v.v..

Các tính năng nên có

- Khả năng xử lý phân tán: Hoạt động trong hệ thống máy tính phân tán, trong phạm vi 1 trung tâm dữ liệu hoặc phân tán trên cả phương diện địa lý.
- Sử dụng tối đa tài nguyên: Có thể khai thác hết các tài nguyên tính toán và băng thông mạng
- Khả năng co-giãn: Có thể tăng hoặc giảm tốc độ thu thập bằng cách bổ xung hoặc giảm tài nguyên.
- Cơ chế ưu tiên: Ưu tiên nạp những trang chất lượng cao và thay đổi thường xuyên
- Vận hành liên tục: Liên tục nạp những bản sao mới nhất của các trang đã nạp trước đó
- Khả năng mở rộng tính năng: Bổ xung khả năng xử lý các định dạng dữ liệu mới, các giao thức mới

Toàn cảnh thu thập phân tán



Hàng đợi URL

- Có thể bao gồm nhiều trang từ cùng 1 máy chủ
- Giải thuật lựa chọn các URLs có thể kiểm soát:
 - Tần suất gửi yêu cầu tới 1 máy chủ
 - Không nạp đồng thời quá nhiều URLs từ 1 máy chủ
 - Ưu tiên nội dung tải về
 - Hệ số tải cao cho các luồng thu thập

Chi tiết được trình bày sau!

robots.txt

- Giao thức giới hạn truy cập tự động được thiết lập từ 1994

<http://www.robotstxt.org/robotstxt.html>

- Các mô tả được đưa ra trong tệp robots.txt thường được đặt trong thư mục gốc của trang Web

Ví dụ: <https://hust.edu.vn/robots.txt>

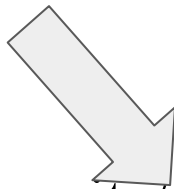
Ví dụ 10.1. robots.txt

User-agent: *

Disallow: /yoursite/temp/

User-agent: searchengine

Disallow:



bot không được mở "/yoursite/temp/", ngoại trừ searchengine

Tách nội dung

- Nhiều trang Web chứa văn bản, liên kết, và các hình ảnh không liên quan với nội dung chính của trang Web
- Những thông tin không liên quan được coi là nhiễu và có thể làm sai lệch xếp hạng của nội dung
- Các kỹ thuật đã được phát triển để xác định các khối nội dung trong 1 trang Web
 - Các thông tin không thuộc nội dung được bỏ qua hoặc thiết lập trọng số nhỏ trong tiến trình đánh chỉ mục

Ví dụ 10.2. Nội dung vs. Thông tin nhiều

Con chip M1 trong MacBo x

← → ↺

← → ↺

META.vn

BẢO SALE ĐỒ BỘ

12.12

DEAL SỐC HÀNG NGÀY

Xem ngay

Nội dung

Công nghệ › macOS

Vào tháng 6/2020, Apple thông báo sẽ không sử dụng chip Intel cho các dòng máy tính Mac nữa. M1 là con chip đầu tiên dựa trên ARM, được thiết kế bởi Apple và dành riêng cho các sản phẩm máy tính của Apple. Dưới đây là tất cả những gì bạn cần biết về con chip M1 mới nhất của Apple.

Chip M1 là gì?

M1 là hệ thống silicon tùy chỉnh đầu tiên của Apple trên chip được sử dụng trong dòng máy tính Mac của hãng. Kể từ năm 2006, tất cả các máy Mac đều được trang bị chip Intel. Chúng sử dụng cấu trúc x86 (và mới hơn, x86_64), cũng có mặt trên các PC của Windows.

Tuy nhiên, M1 lại khác. Nó sử dụng cấu trúc ARM, thường xuất hiện trên điện thoại hoặc các thiết bị nhỏ như iPhone và iPad. ARM sử dụng tập lệnh đơn giản hơn so với x86, do đó mức tiêu thụ điện năng thấp hơn.

M1 mang đến lợi ích nhiều hơn rõ ràng so với chip Intel nhưng cũng còn tồn tại một số nhược điểm. Tuy nhiên, Apple cho rằng hầu hết mọi người sẽ không nhận thấy sự khác biệt lớn khi chuyển từ máy Intel sang máy có chip ARM tùy chỉnh.

Model	Year	Chip	RAM	Storage	Price	Price/GB	Price/Storage	Price/Chip
Mac mini	2020	M1	8GB	256GB	699	87.375	27.750	699
Mac mini	2020	M1	16GB	512GB	1.199	74.937	23.125	1.199
Mac mini	2020	M1	32GB	1TB	1.699	53.125	16.875	1.699
Mac mini	2020	M1	64GB	2TB	2.199	34.375	10.937	2.199
Mac mini	2020	M1	128GB	4TB	2.699	21.250	6.750	2.699
Mac mini	2020	M1	256GB	8TB	3.199	12.500	3.750	3.199
Mac mini	2020	M1	512GB	16TB	3.699	7.187	2.250	3.699
Mac mini	2020	M1	1024GB	32TB	4.199	4.047	1.250	4.199
Mac mini	2020	M1	2048GB	64TB	4.699	2.250	0.625	4.699
Mac mini	2020	M1	4096GB	128TB	5.199	1.250	0.312	5.199
Mac mini	2020	M1	8192GB	256TB	5.699	0.687	0.156	5.699
Mac mini	2020	M1	16384GB	512TB	6.199	0.375	0.078	6.199
Mac mini	2020	M1	32768GB	1024TB	6.699	0.203	0.039	6.699
Mac mini	2020	M1	65536GB	2048TB	7.199	0.109	0.020	7.199
Mac mini	2020	M1	131072GB	4096TB	7.699	0.057	0.010	7.699
Mac mini	2020	M1	262144GB	8192TB	8.199	0.031	0.005	8.199
Mac mini	2020	M1	524288GB	16384TB	8.699	0.016	0.002	8.699
Mac mini	2020	M1	1048576GB	32768TB	9.199	0.008	0.001	9.199
Mac mini	2020	M1	2097152GB	65536TB	9.699	0.004	0.000	9.699
Mac mini	2020	M1	4194304GB	131072TB	10.199	0.002	0.000	10.199
Mac mini	2020	M1	8388608GB	262144TB	10.699	0.001	0.000	10.699
Mac mini	2020	M1	16777216GB	524288TB	11.199	0.000	0.000	11.199
Mac mini	2020	M1	33554432GB	1048576TB	11.699	0.000	0.000	11.699
Mac mini	2020	M1	67108864GB	2097152TB	12.199	0.000	0.000	12.199
Mac mini	2020	M1	134217728GB	4194304TB	12.699	0.000	0.000	12.699
Mac mini	2020	M1	268435456GB	8388608TB	13.199	0.000	0.000	13.199
Mac mini	2020	M1	536870912GB	16777216TB	13.699	0.000	0.000	13.699
Mac mini	2020	M1	1073741824GB	33554432TB	14.199	0.000	0.000	14.199
Mac mini	2020	M1	2147483648GB	67108864TB	14.699	0.000	0.000	14.699
Mac mini	2020	M1	4294967296GB	134217728TB	15.199	0.000	0.000	15.199
Mac mini	2020	M1	8589934592GB	268435456TB	15.699	0.000	0.000	15.699
Mac mini	2020	M1	17179869184GB	536870912TB	16.199	0.000	0.000	16.199
Mac mini	2020	M1	34359738368GB	1073741824TB	16.699	0.000	0.000	16.699
Mac mini	2020	M1	68719476736GB	2147483648TB	17.199	0.000	0.000	17.199
Mac mini	2020	M1	137438953472GB	4294967296TB	17.699	0.000	0.000	17.699
Mac mini	2020	M1	274877906944GB	8589934592TB	18.199	0.000	0.000	18.199
Mac mini	2020	M1	549755813888GB	17179869184TB	18.699	0.000	0.000	18.699
Mac mini	2020	M1	1099511627776GB	34359738368TB	19.199	0.000	0.000	19.199
Mac mini	2020	M1	2199023255552GB	68719476736TB	19.699	0.000	0.000	19.699
Mac mini	2020	M1	4398046511104GB	137438953472TB	20.199	0.000	0.000	20.199
Mac mini	2020	M1	8796093022208GB	274877906944TB	20.699	0.000	0.000	20.699
Mac mini	2020	M1	17592186044416GB	549755813888TB	21.199	0.000	0.000	21.199
Mac mini	2020	M1	35184372088832GB	1099511627776TB	21.699	0.000	0.000	21.699
Mac mini	2020	M1	70368744177664GB	2199023255552TB	22.199	0.000	0.000	22.199
Mac mini	2020	M1	140737488355328GB	4398046511104TB	22.699	0.000	0.000	22.699
Mac mini	2020	M1	281474976710656GB	8796093022208TB	23.199	0.000	0.000	23.199
Mac mini	2020	M1	562949953421312GB	17592186044416TB	23.699	0.000	0.000	23.699
Mac mini	2020	M1	1125899906842624GB	35184372088832TB	24.199	0.000	0.000	24.199
Mac mini	2020	M1	2251799813685248GB	70368744177664TB	24.699	0.000	0.000	24.699
Mac mini	2020	M1	4503599627370496GB	140737488355328TB	25.199	0.000	0.000	25.199
Mac mini	2020	M1	9007199254740992GB	281474976710656TB	25.699	0.000	0.000	25.699
Mac mini	2020	M1	18014398509481984GB	562949953421312TB	26.199	0.000	0.000	26.199
Mac mini	2020	M1	36028797018963968GB	1125899906842624TB	26.699	0.000	0.000	26.699
Mac mini	2020	M1	72057594037927936GB	2251799813685248TB	27.199	0.000	0.000	27.199
Mac mini	2020	M1	144115188075855872GB	4503599627370496TB	27.699	0.000	0.000	27.699
Mac mini	2020	M1	288230376151711744GB	9007199254740992TB	28.199	0.000	0.000	28.199
Mac mini	2020	M1	576460752303423488GB	18014398509481984TB	28.699	0.000	0.000	28.699
Mac mini	2020	M1	1152921504606846976GB	36028797018963968TB	29.199	0.000	0.000	29.199
Mac mini	2020	M1	2305843009213693952GB	72057594037927936TB	29.699	0.000	0.000	29.699
Mac mini	2020	M1	4611686018427387904GB	144115188075855872TB	30.199	0.000	0.000	30.199
Mac mini	2020	M1	9223372036854775808GB	288230376151711744TB	30.699	0.000	0.000	30.699
Mac mini	2020	M1	18446744073709551616GB	576460752303423488TB	31.199	0.000	0.000	31.199
Mac mini	2020	M1	36893488147419103232GB	1152921504606846976TB	31.699	0.000	0.000	31.699
Mac mini	2020	M1	73786976294838206464GB	2305843009213693952TB	32.199	0.000	0.000	32.199
Mac mini	2020	M1	147573952589676412928GB	4611686018427387904TB	32.699	0.000	0.000	32.699
Mac mini	2020	M1	295147905179352825856GB	9223372036854775808TB	33.199	0.000	0.000	33.199
Mac mini	2020	M1	590295810358705651712GB	18446744073709551616TB	33.699	0.000	0.000	33.699
Mac mini	2020	M1	1180591620717411303424GB	36893488147419103232TB	34.199	0.000	0.000	34.199
Mac mini	2020	M1	2361183241434822606848GB	73786976294838206464TB	34.699	0.000	0.000	34.699
Mac mini	2020	M1	4722366482869645213696GB	147573952589676412928TB	35.199	0.000	0.000	35.199
Mac mini	2020	M1	9444732965739290427392GB	295147905179352825856TB	35.699	0.000	0.000	35.699
Mac mini	2020	M1	18889465931478580854784GB	590295810358705651712TB	36.199	0.000	0.000	36.199
Mac mini	2020	M1	37778931862957161709568GB	1180591620717411303424TB	36.699	0.000	0.000	36.699
Mac mini	2020	M1	75557863725914323419136GB	2361183241434822606848TB	37.199	0.000	0.000	37.199
Mac mini	2020	M1	151115727451828646838272GB	4722366482869645213696TB	37.699	0.000	0.000	37.699
Mac mini	2020	M1	302231454903657293676544GB	9444732965739290427392TB	38.199	0.000	0.000	38.199
Mac mini	2020	M1	604462909807314587353088GB	18889465931478580854784TB	38.699	0.000	0.000	38.699
Mac mini	2020	M1	1208925819614629174706176GB	37778931862957161709568TB	39.199	0.000	0.000	39.199
Mac mini	2020	M1	2417851639229258349412352GB	75557863725914323419136TB	39.699	0.000	0.000	39.699
Mac mini	2020	M1	4835703278458516698824704GB	151115727451828646838272TB	40.199	0.000	0.000	40.199
Mac mini	2020	M1	9671406556917033397649408GB	302231454903657293676544TB	40.699	0.000	0.000	40.699
Mac mini	2020	M1	19342813113834066795298816GB	604462909807314587353088TB	41.199	0.000	0.000	41.199
Mac mini	2020	M1	38685626227668133590597632GB	1208925819614629174706176TB	41.699	0.000	0.000	41.699
Mac mini	2020	M1	77371252455336267181195264GB	2417851639229258349412352TB	42.199	0.000	0.000	42.199
Mac mini	2020	M1	154742504910672534362390528GB	4835703278458516698824704TB	42.699	0.000	0.000	42.699
Mac mini	2020	M1	309485009821345068724781056GB	9671406556917033397649408TB	43.199	0.000	0.000	43.199
Mac mini	2020	M1	618970019642690137449562112GB	19342813113834066795298816TB	43.699	0.000	0.000	43.699
Mac mini	2020	M1	1237940039285380274899124224GB	38685626227668133590597632TB	44.199	0.000	0.000	44.199
Mac mini	2020	M1	2475880078570760549798248448GB	77371252455336267181195264TB	44.699	0.000	0.000	44.699
Mac mini	2020	M1	4951760157141521099596496896GB	154742504910672534362390528TB	45.199	0.000	0.000	45.199
Mac mini	2020	M1	9903520314283042199192993792GB	309485009821345068724781056TB	45.699	0.000	0.000	45.699
Mac mini	2020	M1	19807040628566084398385987584GB	618970019642690137449562112TB	46.199	0.000	0.000	46.199
Mac mini	2020	M1	39614081257132168796771975168GB	1237940039285380274899124224TB	46.699	0.000	0.000	46.699
Mac mini	2020	M1	79228162514264337593543950336GB	2475880078570760549798248448TB	47.199	0.000	0.000	47.199
Mac mini	2020	M1	158456325028528675187087900672GB	4951760157141521099596496896TB	47.699	0.000	0.000	47.699
Mac mini	2020	M1	316912650057057350374175801344GB	9903520314283042199192993792TB	48.199	0.000	0.000	48.199
Mac mini	2020	M1	633825300114114700748351602688GB	19807040628566084398385987584TB	48.699	0.000	0.000	48.699
Mac mini	2020	M1	1267650600228229401496703205376GB	39614081257132168796771975168TB	49.199	0.000	0.000	49.199
Mac mini	2020	M1	2535301200456458802993406410752GB	79228162514264337593543950336TB	49.699	0.000	0.000	49.699
Mac mini	2020	M1	5070602400912917605986812821504GB	158456325028528675187087900672TB	50.199	0.000	0.000	50.199
Mac mini	2020	M1	10141204801825835211973625643008GB	316912650057057350374175801344TB	50.699	0.000	0.000	50.699
Mac mini	2020	M1	20282409603651670423947251286016GB	633825300114114700748351602688TB	51.199	0.000	0.000	51.199
Mac mini	2020	M1	40564819207303340847894502572032GB	1267650600228229401496703205376TB	51.699	0.000	0.000	51.699
Mac mini	2020	M1	81129638414606681695789005144064GB	2535301200456458802993406410752TB	52.199	0.000	0.000	52.199
Mac mini	2020	M1	162259276829213363391578010288128GB	5070602400912917605986812821504TB	52.699	0.000	0.000	52.699
Mac mini	2020	M1	324					

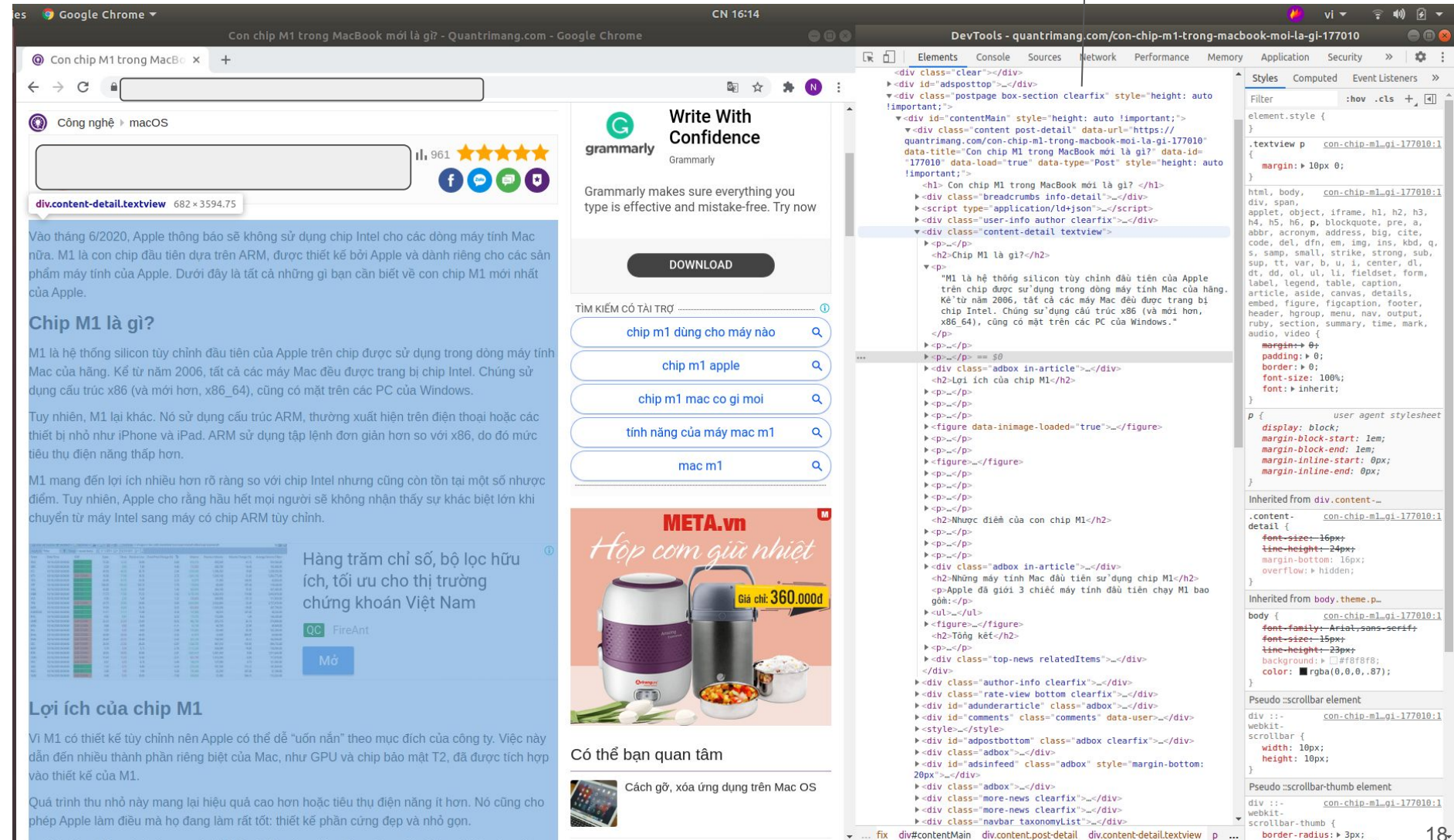
Tìm khối nội dung

- Biểu diễn một trang Web như một dãy bits, trong đó $b_i = 1$ nếu từ thứ i là một thẻ và $b_i = 0$ nếu ngược lại.
- Bài toán tối ưu hóa: Tìm cặp giá trị i và j để cực đại hóa số lượng thẻ trước i và sau j và số lượng từ không phải là thẻ giữa i và j .
- Cực đại hóa:

$$\sum_{n=0}^{i-1} b_n + \sum_{n=i}^j (1 - b_n) + \sum_{n=j+1}^{N-1} b_n$$

Tìm khối nội dung₍₂₎

Xử lý thủ công trên cấu trúc DOM với các biểu thức XPath hoặc CSS



Cập nhật giải thuật thu thập dữ liệu

- Lặp nếu hàng đợi không rỗng:

- Lấy URL từ hàng đợi;

Theo cơ chế nào?

- Nạp trang Web

- Đọc nội dung trang Web;

- Tách nội dung và các URLs từ trang Web;

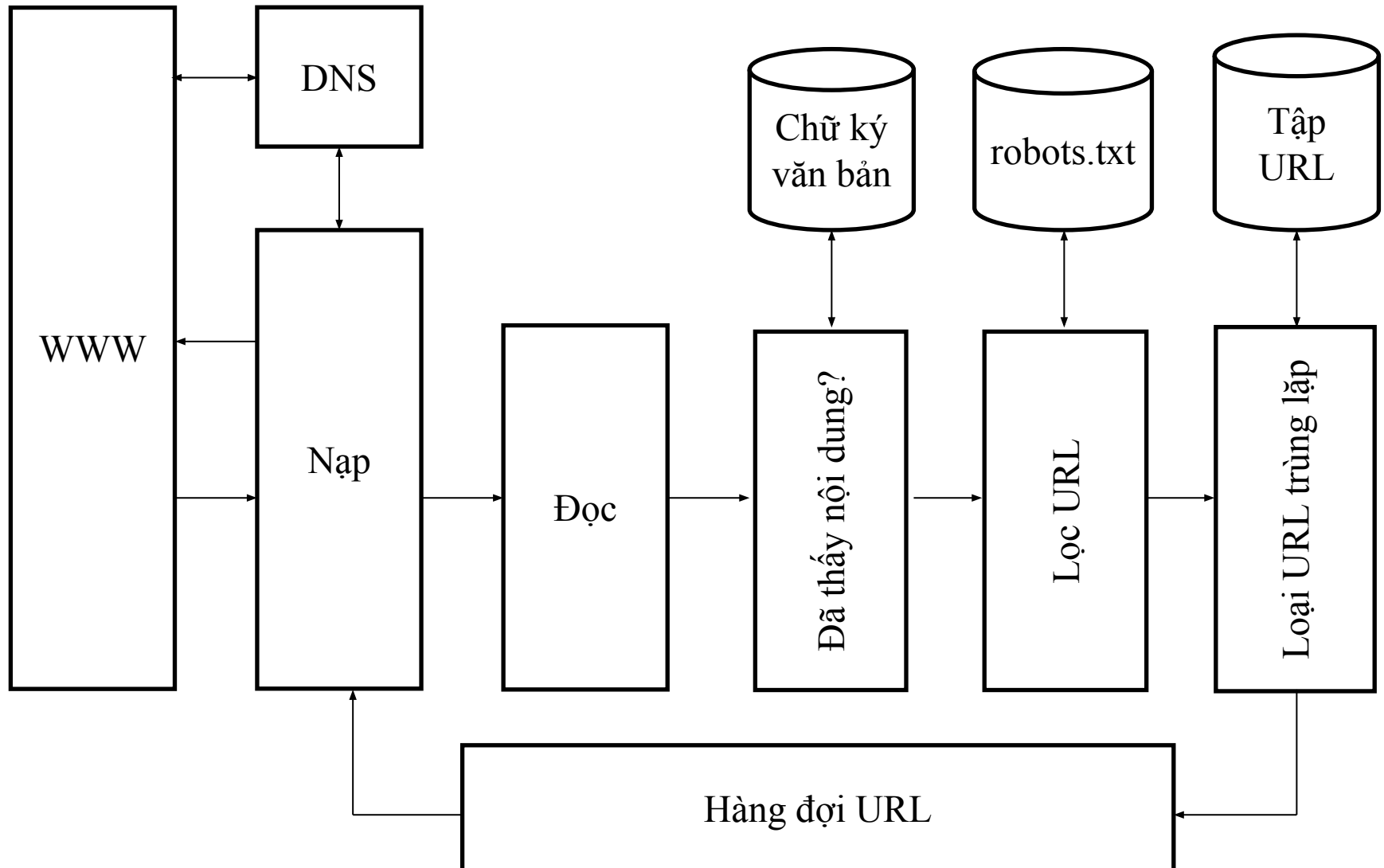
- Thêm URLs mới vào hàng đợi.

Lọc URL trước khi thêm vào hàng đợi: Ví dụ, giới hạn tên miền, robots.txt, v.v.

Nội dung

1. Các vấn đề thu thập dữ liệu Web
2. Các yêu cầu hệ thống
3. Tổng quan hệ thống thu thập dữ liệu
4. Phân giải DNS
5. Hàng đợi URL
6. Máy chủ liên kết
7. Chia nhỏ và phân tán chỉ mục ngược
8. Lưu trữ tài liệu quy mô lớn

Kiến trúc thu thập cơ bản



Phân giải tên miền

DNS - Domain Name System

- Phân giải tên miền là 1 dịch vụ nền tảng của Internet
 - Cho một URL, hệ thống trả về IP của máy chủ Web.
 - Dịch vụ được cung cấp bởi một hệ thống máy chủ phân tán, có thể được thực hiện qua nhiều bước, thời gian phân giải có thể lớn.
- Phân giải DNS trong các HĐH phổ thông thường được đồng bộ theo cơ chế khóa với số lượng yêu cầu đồng thời giới hạn
 - Không đủ đáp ứng các yêu cầu về hiệu năng cho hệ thu thập
- Giải pháp
 - Bộ nhớ đệm DNS, phân giải DNS theo gói v.v.

Nội dung chi tiết được trình bày sau

Đọc dữ liệu: Chuẩn hóa URL

- Liên kết tách được từ trang Web thường là liên kết tương đối
- Ví dụ, trang
[https://vi.wikipedia.org/wiki/Trang_Ch%C3%ADnh](https://vi.wikipedia.org/wiki/Trang_Ch%C3%ADnh_chứa_liên_kết_tương_đối)
chứa liên kết tương đối
[/wiki/Wikipedia:Gi%E1%BB%9Bi_thi%E1%BB%87u](https://vi.wikipedia.org/wiki/Wikipedia:Gi%E1%BB%9Bi_thi%E1%BB%87u)
có URL tuyệt đối tương đương là
https://vi.wikipedia.org/wiki/Wikipedia:Gi%E1%BB%9Bi_thi%E1%BB%87u
- Cần chuẩn hóa: Mở rộng các URLs ở dạng tương đối về dạng tuyệt đối.

Nội dung đã có?

- Các nội dung trùng lặp rất phổ biến trên Web
- Nếu nội dung của 1 trang mới được tải về đã có trong chỉ mục, thì có thể bỏ qua.
- Được kiểm soát bằng tổng đại diện hoặc các đại lượng khác (chữ ký văn bản) tùy theo giải thuật kiểm tra trùng lặp.

[Chương 9]

Bộ lọc và robots.txt

- Các biểu thức chính quy cho các URL's cần được thu thập/hoặc không
 - Kiểm tra URLs là của tài liệu có định dạng (doc, pdf, ...), hình ảnh, âm thanh, 1 trang Web, v.v...
 - Kiểm tra quyền truy cập
- Lưu robots.txt trong các bộ nhớ đệm
 - Tải robots.txt 1 lần từ 1 miền Web
 - Không cần tải lại mỗi khi thu thập 1 trang cùng miền
 - Giảm lưu lượng và số lượng yêu cầu tới máy chủ

Loại URL trùng lặp

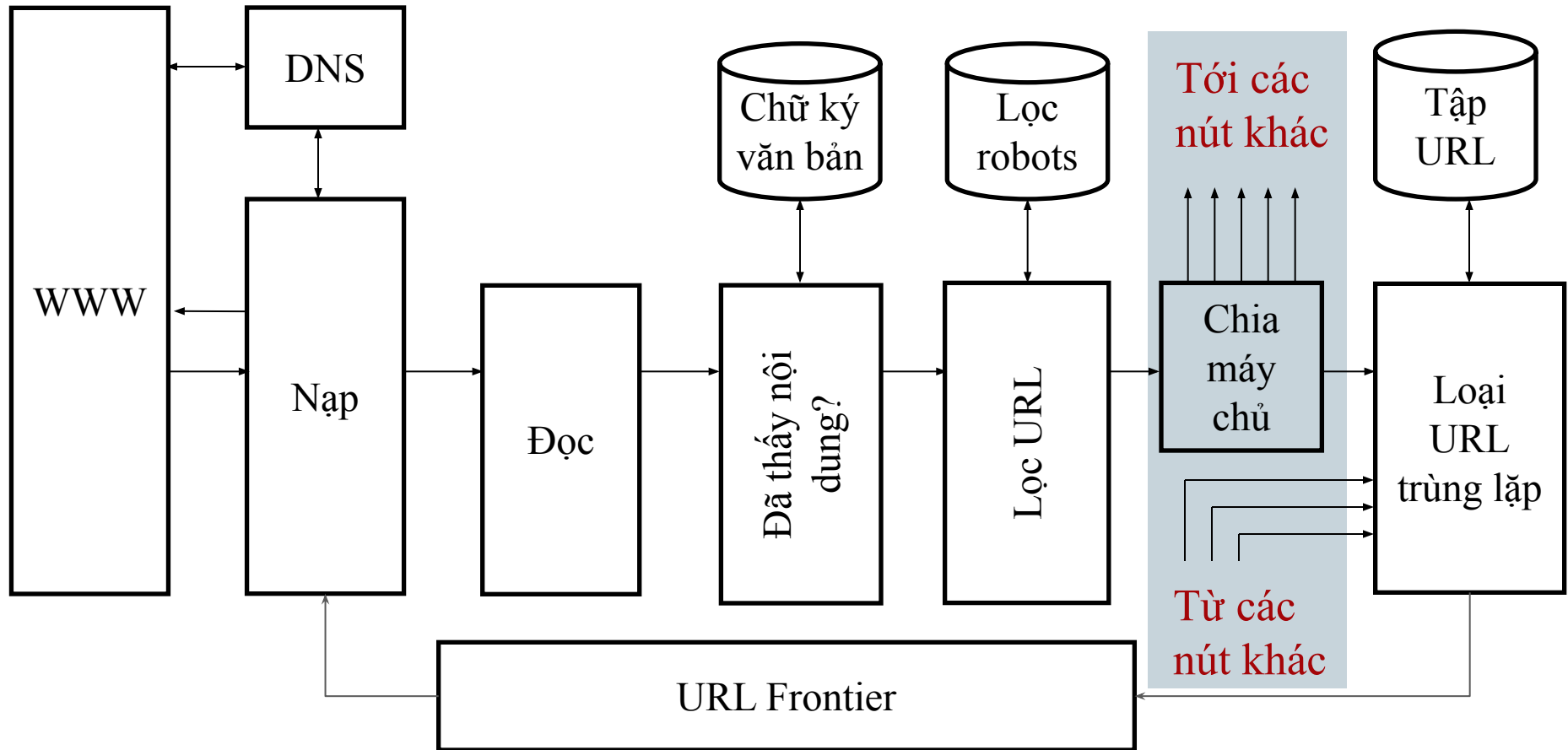
Bài toán đối với các thu thập 1 lượt (không liên tục): Kiểm tra URL đi qua bộ lọc có phải là URL mới hay không?

Phân tán các bộ thu thập

- Chạy nhiều luồng thu thập trong nhiều tiến trình khác nhau, có thể trong nhiều nút khác nhau của hệ phân tán
 - Các nút có thể được phân tán theo phương diện địa lý
- Phân chia các trang cần thu thập cho các nút theo máy chủ
 - Có thể sử dụng các hàm băm.


Các nút giao tiếp và chia sẻ URLs như thế nào?

Giao tiếp giữa các nút

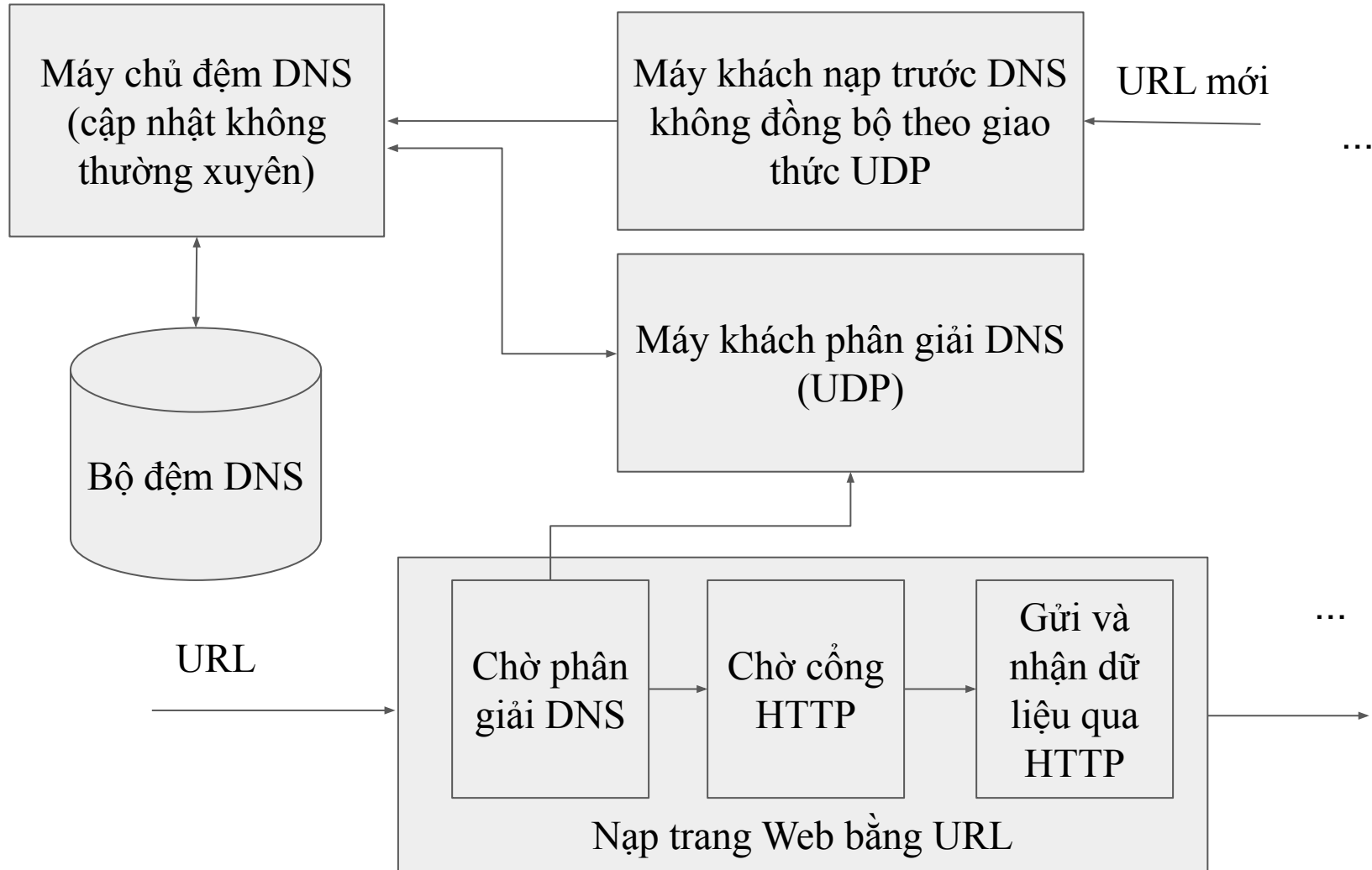


Các URLs đã đi qua mô-đun lọc URL được chuyển tới mô-đun loại URL trùng lặp tập trung.

Nội dung

1. Các vấn đề thu thập dữ liệu Web
 2. Các yêu cầu hệ thống
 3. Tổng quan hệ thống thu thập dữ liệu
 4. Phân giải DNS
 5. Hàng đợi URL
 6. Máy chủ liên kết
 7. Chia nhỏ và phân tán chỉ mục ngược
 8. Lưu trữ tài liệu quy mô lớn
- 

Giải pháp DNS cho hệ thu thập



DNS riêng cho hệ thu thập₍₂₎

- Tăng lưu lượng DNS => giảm thời gian thu thập:
 1. Máy khách riêng cho phân giải tên miền
 2. Máy khách nạp trước
 3. Máy chủ đệm DNS

Máy khách phân giải tên miền

- Có khả năng xử lý xung đột của các yêu cầu đồng thời
- Cho phép gửi đồng thời nhiều yêu cầu phân giải
 - Sau đó kiểm tra trạng thái của từng yêu cầu riêng lẻ
- Phân bổ tải giữa nhiều máy chủ DNS

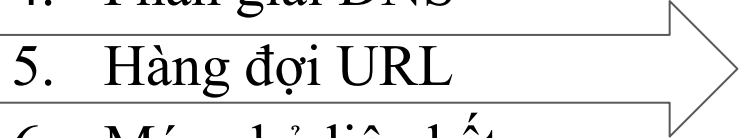
Máy chủ đệm

- Duy trì bộ nhớ đệm lớn, sử dụng lưu trữ cố định để khôi phục lại sau mỗi lần khởi động lại
- Lưu giữ tối đa dữ liệu trong bộ nhớ chính

Máy khách nạp trước

- Các bước:
 1. Đọc một trang mới được tải về
 2. Xuất tên máy chủ từ các thuộc tính href
 3. Gửi các yêu cầu phân giải DNS tới các máy chủ đệm
- Thường được triển khai theo giao thức UDP
 - (User Datagram Protocol)
 - Giao thức dựa trên gói, không yêu cầu kết nối
 - Không đảm bảo chuyển giao gói
- Không chờ cho tới khi yêu cầu phân giải được hoàn thành

Nội dung

1. Các vấn đề thu thập dữ liệu Web
 2. Các yêu cầu hệ thống
 3. Tổng quan hệ thống thu thập dữ liệu
 4. Phân giải DNS
 5. Hàng đợi URL
 6. Máy chủ liên kết
 7. Chia nhỏ và phân tán chỉ mục ngược
 8. Lưu trữ tài liệu quy mô lớn
- 

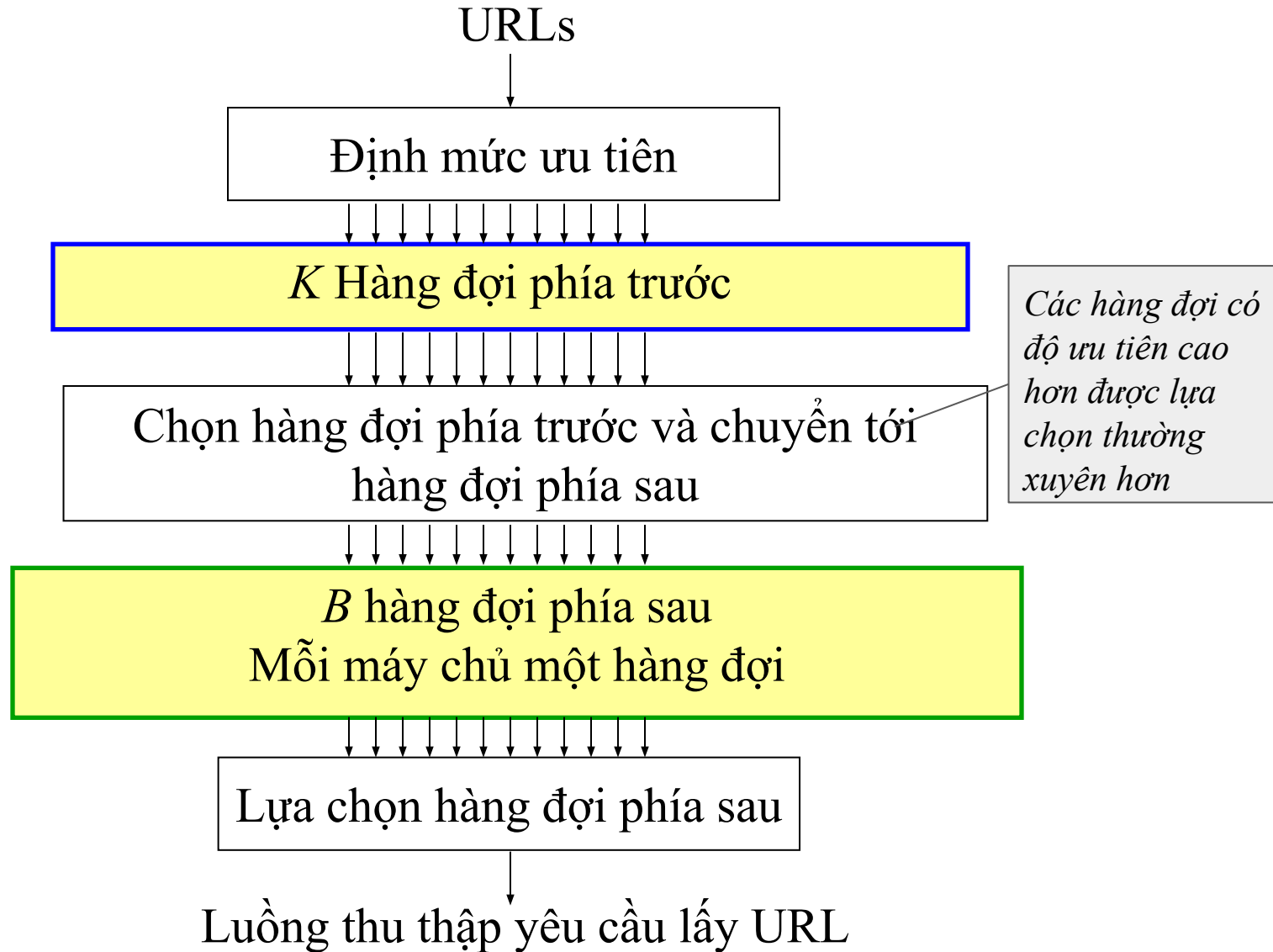
Hàng đợi URL: Hai yêu cầu chính

- Tính lịch thiệp: Không yêu cầu 1 máy chủ Web quá thường xuyên
- Tính cập nhật: Thu thập 1 số trang thường xuyên hơn các trang khác
 - Ví dụ, các trang (như các trang tin tức) có nội dung thay đổi thường xuyên
- Hai yêu cầu này có thể có mâu thuẫn:
 - Đảm bảo tính lịch thiệp có thể làm giảm tốc độ thu thập, ảnh hưởng đến tính cập nhật.
 - Các trang được ưu tiên có thể chứa nhiều liên kết đến các trang khác cùng miền, tạo ra một số lượng lớn yêu cầu thu thập tới 1 miền được ưu tiên.

Đảm bảo tính lịch thiệp

- Không chỉ trong điều kiện phân tán, kể cả trong trường hợp giới hạn 1 luồng thu thập dữ liệu chạy trên 1 máy, vẫn có thể có rất nhiều yêu cầu được gửi tới 1 máy chủ Web
- Giải pháp phổ biến: Thiết lập 1 khoảng thời gian chờ kể từ sau lần thu thập cuối cùng đến lần gửi yêu cầu tiếp theo.

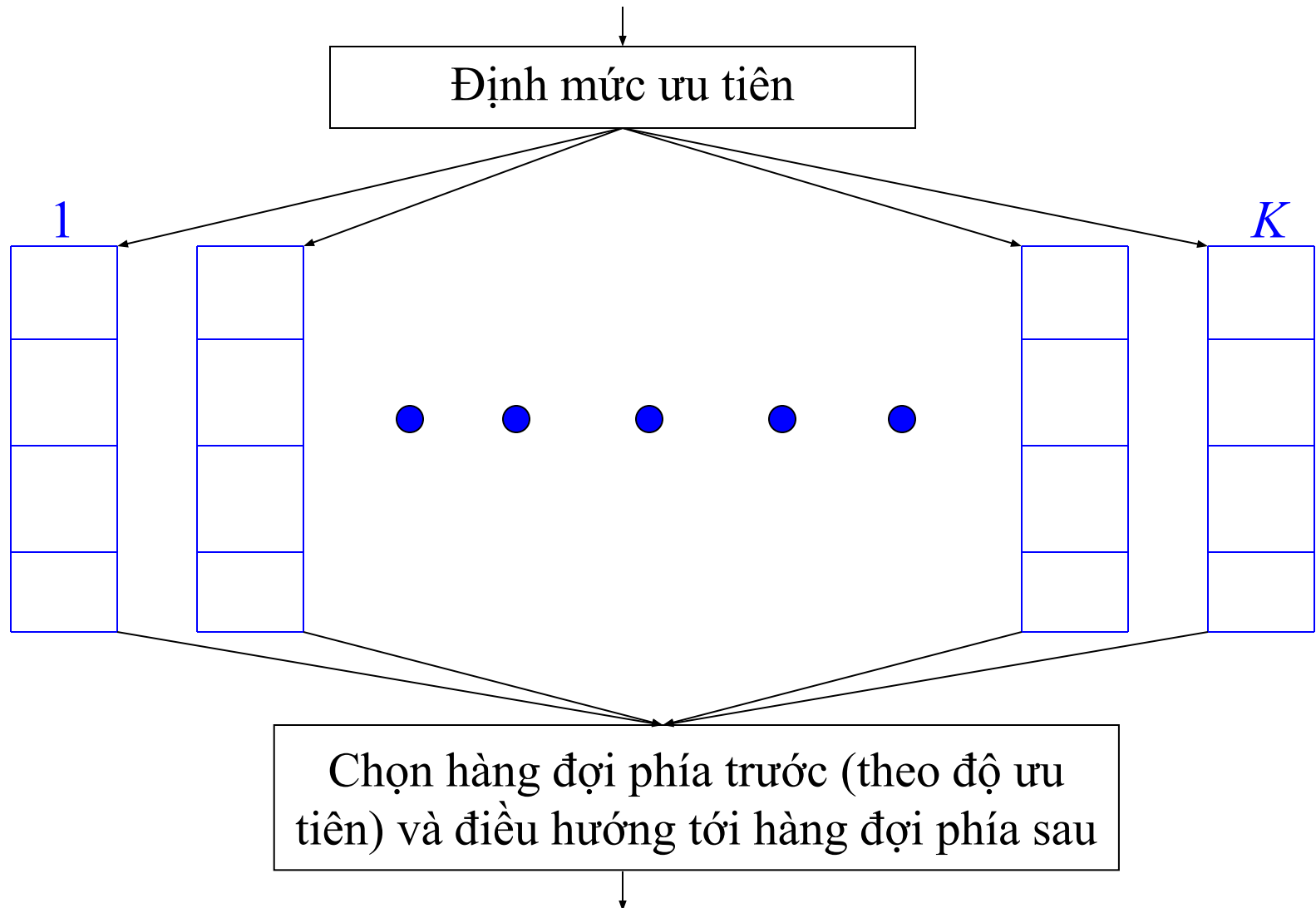
Hàng đợi URL: Triển khai trong Mercator



Hàng đợi của Mercator

- Các URLs di chuyển theo luồng từ trên xuống dưới (theo sơ đồ), trước khi đi tới luồng thu thập phải đi qua hai lớp hàng đợi: Các hàng đợi phía trước và các hàng đợi phía sau.
- Các hàng đợi phía trước đảm bảo cơ chế ưu tiên
- Các hàng đợi phía sau cho phép kiểm soát tần suất truy cập
- Tất cả các hàng đợi cơ bản trong các tập hàng đợi đều là FIFO

Các hàng đợi phía trước



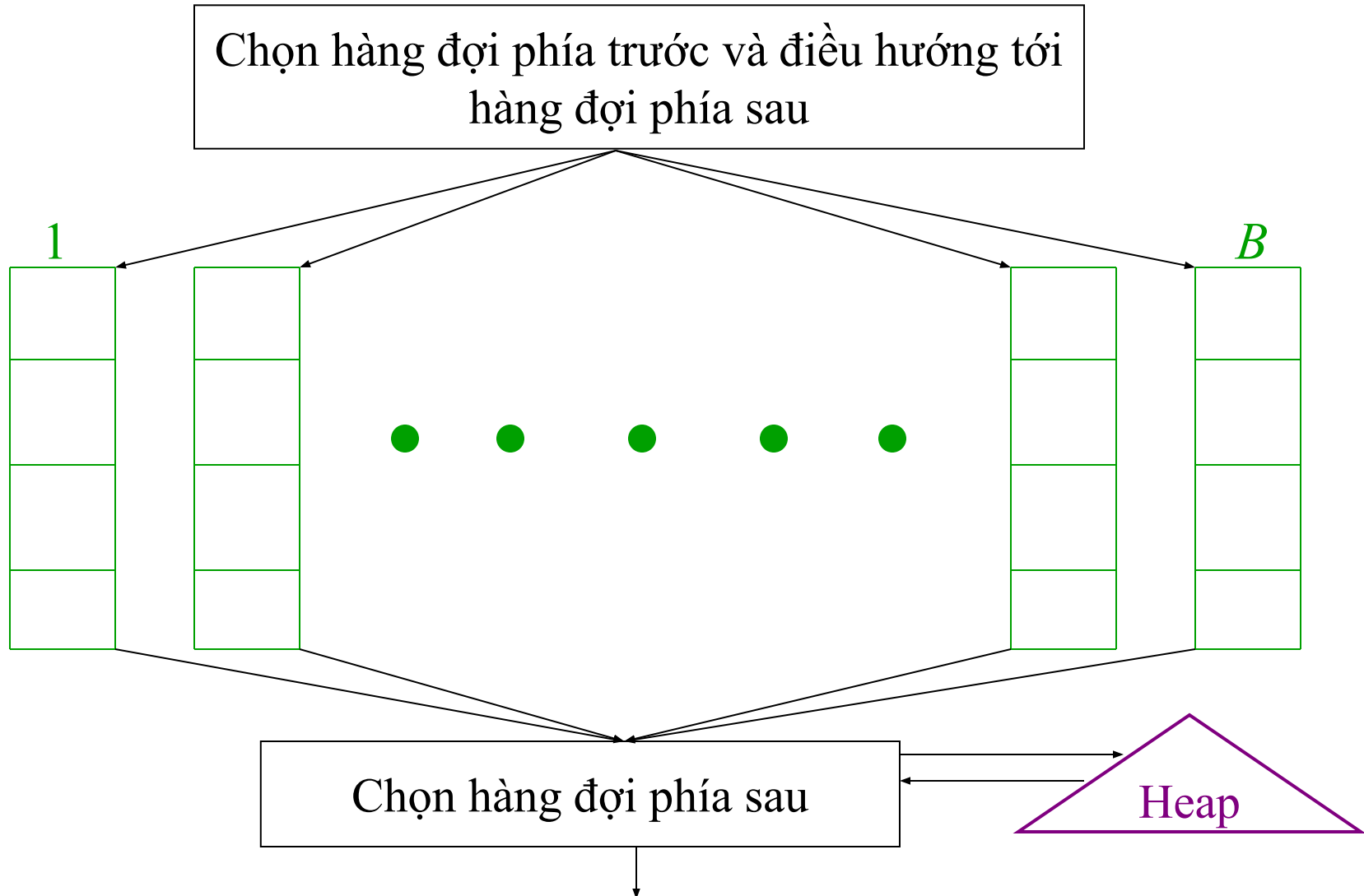
Các hàng đợi phía trước₍₂₎

- Mô-đun định mức ưu tiên gán cho mỗi URL 1 mức ưu tiên là số nguyên trong khoảng từ 1 đến F.
 - Sau đó thêm URL vào hàng đợi tương ứng với mức ưu tiên
- Mức ưu tiên được xác định bằng các chỉ số:
 - Tốc độ tải về trong lần tải về cuối cùng, PageRank
 - Theo từng ứng dụng cụ thể
 - Ví dụ, thu thập các trang tin tức thường xuyên hơn đối với hệ thống tập trung vào tin tức.

Chọn hàng đợi phía trước

- Khi 1 hàng đợi phía sau yêu cầu 1 URL:
 - 1 hàng đợi phía trước cần được chọn để cung cấp URL
- Có thể theo lô-gic xoay vòng nhưng tập trung hơn vào các hàng đợi có độ ưu tiên cao hơn
 - Hoặc có thể sử dụng các giải thuật ngẫu nhiên và các giải thuật khác

Các hàng đợi phía sau



Các bất biến đối với hàng đợi phía sau

- Các hàng đợi phía sau được đảm bảo không rỗng trong suốt thời gian thu thập.
- Mỗi hàng đợi phía sau chỉ lưu các URLs từ 1 máy chủ duy nhất

Máy chủ	Hàng đợi phía sau
example.org	3
search.vn	1
...	B

Cấu trúc Heap cho các hàng đợi phía sau

- Mỗi cặp máy chủ Web & hàng đợi phía sau được gắn với 1 thời điểm t_e , là thời điểm gần nhất có thể gửi yêu cầu tiếp theo tới máy chủ Web
- Thời điểm t_e gần nhất được xác định dựa trên
 - Lần truy cập cuối cùng tới máy chủ
 - Khoảng thời gian chờ trước khi gửi yêu cầu tiếp theo
- Các bản ghi được lưu theo 1 cấu trúc đống (Heap) cực tiểu theo thời gian t_e

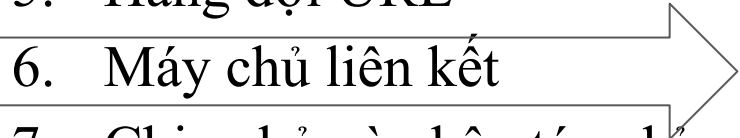
Lấy URL từ các hàng đợi phía sau

- Khi một luồng thu thập yêu cầu 1 URL để xử lý
 - Xuất gốc của Heap (có t_e nhỏ nhất)
 - Lấy ra 1 URL từ hàng đợi phía sau tương ứng (hàng đợi q , sử dụng bảng tra cứu)
- Kiểm tra xem q rỗng hay không sau khi lấy URL. Nếu rỗng thì lấy 1 URL từ các hàng đợi phía trước (v)
 - Nếu đã có hàng đợi phía sau cho máy chủ của v , thì thêm v vào đó, và lặp lại yêu cầu lấy URL từ hàng đợi phía trước
 - Nếu ngược lại thì thêm v vào q
- Nếu q không rỗng thì tạo 1 bản ghi mới trong Heap cho nó

Số lượng hàng đợi phía sau B

- Sử dụng số lượng hàng đợi phía sau hợp lý có thể giữ cho tất cả các luồng thu thập hoạt động liên tục, đồng thời vẫn đảm bảo tính lịch thiệp
- Mercator sử dụng số lượng hàng đợi phía sau nhiều gấp 3 lần số lượng luồng thu thập.

Nội dung

1. Các vấn đề thu thập dữ liệu Web
 2. Các yêu cầu hệ thống
 3. Tổng quan hệ thống thu thập dữ liệu
 4. Phân giải DNS
 5. Hàng đợi URL
 6. Máy chủ liên kết
 7. Chia nhỏ và phân tán chỉ mục ngược
 8. Lưu trữ tài liệu quy mô lớn
- 

Máy chủ liên kết

- Hỗ trợ xử lý nhanh các truy vấn trên đồ thị Web
 - Lấy URLs trở tới 1 URL?/Láng giềng theo chiều đi vào
 - Lấy URLs được trở tới bởi 1 URL?/Láng giềng theo chiều đi ra
- Lưu các tham chiếu trong bộ nhớ
- Ứng dụng
 - Phân tích liên kết
 - Phân tích đồ thị Web
 - Tính liên thông, tối ưu hóa thu thập
 - Điều khiển thu thập

Biểu diễn danh sách kề

- Danh sách các láng giềng của một nút
- Giả sử có thể biểu diễn mỗi URL bằng 1 số nguyên
- Ví dụ, với 4 tỉ trang Web, chúng ta cần 32 bits cho 1 nút
- Với cách biểu diễn thông thường, sẽ cần đến 64 bits để biểu diễn 1 liên kết (gồm nút nguồn và đích đến)
- Giải thuật nén có vai trò quan trọng để lưu một lượng lớn liên kết trong bộ nhớ
 - Boldi/Vigna giảm xuống khoảng ~ 3 bits/liên kết

Nén danh sách kề

- Các tính chất được khai thác trong giải thuật nén:
 - Tính tương đồng (giữa các danh sách)
 - Tính cục bộ (nhiều liên kết từ 1 trang đi tới những trang gần nó)
 - Mã hóa các khoảng cách trong các danh sách được sắp xếp
 - Phân bố các giá trị khoảng cách

Các ý tưởng chính của Boldi/Vigna (BV)

- Xét trật tự chữ cái của một danh sách URLs, ví dụ,
 - <https://www.hust.edu.vn/dai-hoc-chinh-quy>
 - <https://www.hust.edu.vn/hoatdongchung>
 - <https://www.hust.edu.vn/nganh-dao-tao>
 - <https://www.hust.edu.vn/nghien-cuu-sinh>
 - <https://www.hust.edu.vn/su-kien-sap-dien-ra>
 - <https://www.hust.edu.vn/thong-bao-moi>
 - <https://www.hust.edu.vn/tuyen-sinh-cao-hoc>

Boldi/Vigna

- Mỗi URL có 1 danh sách kề
- Ý tưởng chính: Danh sách kề của 1 nút tương tự với 1 trong số 7 danh sách URLs trước nó theo trật tự chữ cái
- Biểu diễn danh sách kề thông qua một danh sách đứng trước nó
- Ví dụ:

1, 2, 4, 8, 16, 32, 64

1, 4, 9, 16, 25, 36, 49, 64

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144

1, 4, 8, 16, 25, 36, 49, 64



Được mã hóa là (-2), xóa 9, thêm 8

Mã hóa khoảng cách

- Cho một danh sách số nguyên x, y, z đã được sắp xếp
- Chuyển danh sách đã sắp xếp thành danh sách khoảng cách $x, y-z, z-y$
- Nén các giá trị khoảng cách bằng:
 - mã y - số lượng bits = $1 + 2 \lceil \lg x \rceil$
 - ...
 - Giới hạn theo lý thuyết thông tin: $1 + \lceil \lg x \rceil$

Các ưu điểm của BV

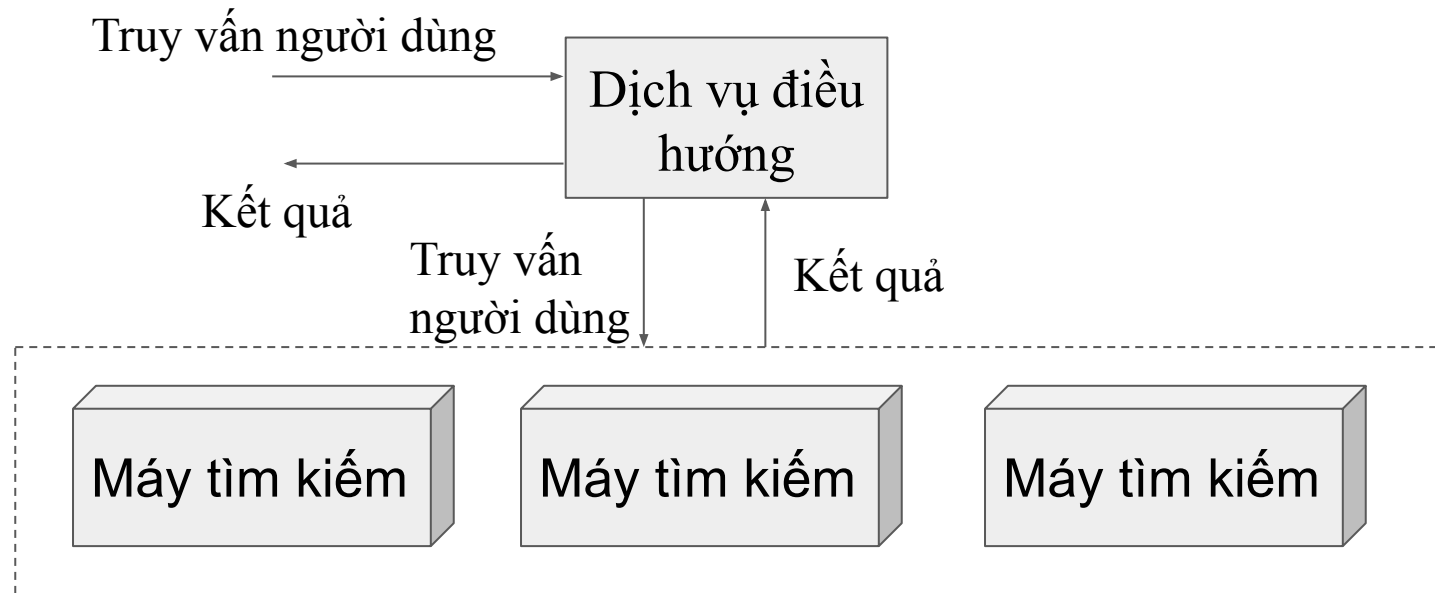
- Chỉ phụ thuộc vào tính cục bộ trong danh sách được sắp xếp theo trật tự ký tự
 - Trật tự ký tự hoạt động tốt cho các URL
- Các truy vấn tìm láng giềng có thể được đáp ứng rất hiệu quả
 - Để lấy 1 danh sách có thể phải xử lý một chuỗi các bước giải mã
 - Chuỗi mã hóa thường ngắn trong thực tế
 - Có thể đặt giới hạn độ dài cho chuỗi mã hóa

Nội dung

1. Các vấn đề thu thập dữ liệu Web
2. Các yêu cầu hệ thống
3. Tổng quan hệ thống thu thập dữ liệu
4. Phân giải DNS
5. Hàng đợi URL
6. Máy chủ liên kết
7. Chia nhỏ và phân tán chỉ mục ngược
8. Lưu trữ tài liệu quy mô lớn

Phân tán chỉ mục

- Trường hợp đầu tiên là lặp lại chỉ mục nhiều lần trên các máy tìm kiếm: Để xử lý đồng thời nhiều truy vấn
- Một dịch vụ điều hướng gửi câu truy vấn tới các máy tìm kiếm và kiểm soát phân bố tải:



Phân chia theo văn bản

- Khi có 1 lượng lớn văn bản mà 1 máy chủ không thể xử lý hết
 - Chia toàn bộ văn bản thành nhiều khối nhỏ
 - Xây dựng cho mỗi phần văn bản 1 chỉ mục ngược riêng
- Cách đơn giản nhất là phân chia ngẫu nhiên các văn bản
- Tuy nhiên các kết quả phù hợp vẫn có thể được tìm kiếm trong phạm vi tất cả văn bản
 - Dịch vụ điều hướng gửi câu truy vấn tới tất cả các máy chủ tìm kiếm và tích hợp các kết quả trước khi trả về cho người dùng

Phân chia theo văn bản₍₂₎

- Cũng có thể phân chia các văn bản thành các phần độc lập
- Các văn bản được gom lại thành các khối theo nội dung
 - Ví dụ, theo các chủ đề, mỗi chủ đề là 1 khối
- Phương án phân chia lý tưởng là phương án phân chia sao cho mỗi truy vấn gắn với 1 khối văn bản duy nhất
 - Các kết quả phù hợp với truy vấn tập trung trong 1 khối văn bản

Tham khảo thêm phân chia cụm văn bản

Lựa chọn khối văn bản

- Hệ quả của việc phân chia văn bản là cần thực hiện lựa chọn đúng khối văn bản có khả năng chứa nhiều kết quả phù hợp trong thời gian xử lý truy vấn
- Một cách tiếp cận đơn giản là coi khả năng chứa kết quả phù hợp của tất cả các khối như nhau
 - Kém chính xác
- Khi các văn bản được phân chia theo nội dung, các khối có thể được xếp hạng dựa trên khả năng chứa văn bản phù hợp

Lựa chọn khối văn bản₍₂₎

- Kỹ thuật cơ bản là coi mỗi khối văn bản như 1 văn bản lớn
- Xử lý truy vấn trên các biểu diễn của các khối văn bản để có được một danh sách xếp hạng các khối văn bản:
 - Ví dụ độ tương đồng cosine, khoảng cách Euclide trong không gian vec-tơ
 - Các phương pháp xác suất
 - v.v.

Các đại lượng thống kê

- Cần các giá trị thống kê toàn cục cho từ để có thể tính đúng điểm xếp hạng cho các văn bản
- Có hai cách tiếp cận để thu thập các đại lượng thống kê toàn cục cho từ
 - Cách thứ nhất là tính các đại lượng thống kê toàn cục ở thời điểm đánh chỉ mục và lưu cùng với từng khối văn bản
 - Cách thứ hai là tính các đại lượng thống kê ở thời điểm xử lý truy vấn:
 - Các đại lượng thống kê từ các chỉ mục con được tổng hợp lại thành đại lượng thống kê toàn cục
 - Dịch vụ điều hướng gửi các đại lượng thống kê cho các mô-đun xử lý truy vấn

Phân chia theo từ khóa

- Các chỉ mục ngược được chia theo chiều ngang và phân tán trên các máy tìm kiếm
 - Mỗi máy tìm kiếm chứa các danh sách thể định vị cho 1 nhóm từ
- Câu truy vấn được phân rã thành nhiều thành phần và gửi đến máy tìm kiếm có từ truy vấn
- Các máy tìm kiếm trả về các kết quả với một phần điểm xếp hạng của các văn bản
- Dịch vụ điều hướng sau đó thực hiện hợp nhất các danh sách kết quả

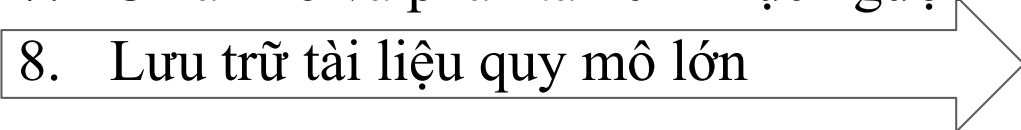
Phân chia theo từ khóa₍₂₎

- Các truy vấn có thể được xử lý song song, các máy tìm kiếm khác nhau có thể trả lời các câu truy vấn thành phần khác nhau
- Tuy nhiên, tải của các máy tìm kiếm có thể không cân bằng do phân bố không đều của từ truy vấn
 - Một từ truy vấn được sử dụng thường xuyên dẫn đến các yêu cầu thường xuyên gửi đến 1 máy tìm kiếm có từ đó
 - Có thể phân tích lịch sử truy vấn để tìm cách phân chia tài nguyên tính toán phù hợp với tải của các máy

Phân chia theo văn bản vs. từ khóa

- Giả sử mỗi máy tìm kiếm được triển khai trên 1 hệ thống máy tính độc lập, phân chia theo văn bản có thể hiệu quả hơn
 - Các chỉ mục nhỏ hơn, dễ quản lý hơn
- Tuy nhiên trong điều kiện từ truy vấn được phân bố đồng đều, phân chia theo từ khóa có thể hiệu quả hơn
- Nhược điểm chính của phân chia theo văn bản
 - Yêu cầu tìm kiếm có thể được thực hiện trên các chỉ mục con chứa ít kết quả phù hợp => có thể không tìm thấy kết quả phù hợp
- Nhược điểm chính của phân chia theo từ khóa
 - Cần xây dựng và quản lý chỉ mục toàn cục đầy đủ ở quy mô lớn => Hạn chế khả năng mở rộng
 - Thời gian xử lý truy vấn có thể dao động đáng kể và cơ chế cân bằng tải phức tạp hơn do các vấn đề liên quan đến phân bố từ

Nội dung

1. Các vấn đề thu thập dữ liệu Web
 2. Các yêu cầu hệ thống
 3. Tổng quan hệ thống thu thập dữ liệu
 4. Phân giải DNS
 5. Hàng đợi URL
 6. Máy chủ liên kết
 7. Chia nhỏ và phân tán chỉ mục ngược
 8. Lưu trữ tài liệu quy mô lớn
- 

Lưu trữ các văn bản

- Có nhiều lý do để lưu các nội dung văn bản thu thập được
 - Tiết kiệm thời gian thu thập khi trang chưa cập nhật
 - Sử dụng nội dung văn bản để sinh trích đoạn cho kết quả tìm kiếm, trích rút thông tin, v.v.
- Các hệ quản trị CSDL thông dụng có thể đáp ứng nhu cầu lưu trữ văn bản cho 1 số tình huống ứng dụng
 - Máy tìm kiếm Web sử dụng hệ thống lưu trữ văn bản chuyên dụng, tự phát triển

Lưu trữ các văn bản₍₂₎

- Các yêu cầu đối với hệ thống lưu trữ văn bản:
 - Truy cập ngẫu nhiên
 - Lấy nội dung văn bản bằng URL của nó
 - Thường được triển khai dựa trên các hàm băm URL
 - Nén các tệp lớn
 - Giảm dung lượng lưu trữ đồng thời giảm thời gian truy cập
 - Khả năng cập nhật
 - Xử lý 1 lượng lớn văn bản mới và được cập nhật liên tục
 - Bổ xung thêm các văn bản liên kết mới vào biểu diễn văn bản

Các tệp lớn

- Lưu nhiều văn bản trong 1 tệp lớn, thay vì lưu mỗi văn bản trong 1 tệp
 - Giảm chi phí mở và đóng tệp
 - Giảm tỉ lệ thời gian định vị so với thời gian đọc
- Định dạng văn bản tổng hợp
 - Được sử dụng để lưu nhiều văn bản trong 1 tệp
 - Ví dụ, TREC Web

Định dạng TREC Web

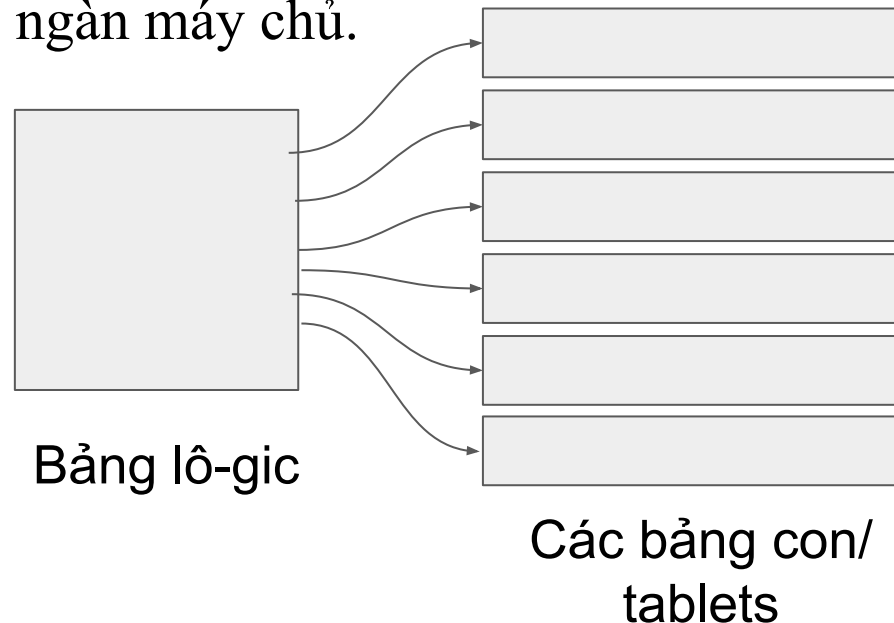
```
<DOC>
<DOCNO>WTX001-B01-10</DOCNO>
<DOCHDR>
http://www.example.com/test.html 204.244.59.33 19970101013145 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:13 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Tropical Fish Store</TITLE>
Coming soon!
</HTML>
</DOC>
<DOC>
<DOCNO>WTX001-B01-109</DOCNO>
<DOCHDR>
http://www.example.com/fish.html 204.244.59.33 19970101013149 text/html 440
HTTP/1.0 200 OK
Date: Wed, 01 Jan 1997 01:21:19 GMT
Server: Apache/1.0.3
Content-type: text/html
Content-length: 270
Last-modified: Mon, 25 Nov 1996 05:31:24 GMT
</DOCHDR>
<HTML>
<TITLE>Fish Information</TITLE>
This page will soon contain interesting
information about tropical fish.
</HTML>
</DOC>
```

Nén

- Dữ liệu văn bản có tính dư thừa cao (ví dụ, 1 từ có thể được sử dụng nhiều lần)
- Các giải thuật nén khai thác các đặc điểm dư thừa để tạo các biểu diễn nhỏ gọn hơn mà không làm mất nội dung
- Các giải thuật nén phổ biến có thể nén văn bản HTML và XML tới 80%
 - Ví dụ, DEFLATE (zip, gzip) và LZW (được sử dụng nhiều trong môi trường UNIX, định dạng Gif, PDF)
 - Có thể nén các tệp lớn thành nhiều khối để truy cập nhanh hơn

BigTable

- Hệ thống lưu trữ văn bản của Google
 - Được tùy chỉnh để lưu, tìm kiếm, và cập nhật các trang Web
 - Quản lý dữ liệu dung lượng lớn bằng hệ thống máy tính phổ thông
- Mỗi CSDL chỉ bao gồm 1 bảng
 - Kích thước có thể rất lớn, nhiều Peta Bytes (PB)
 - Bảng được nhỏ thành nhiều phần, các bảng con được lưu phân tán trong hàng ngàn máy chủ.

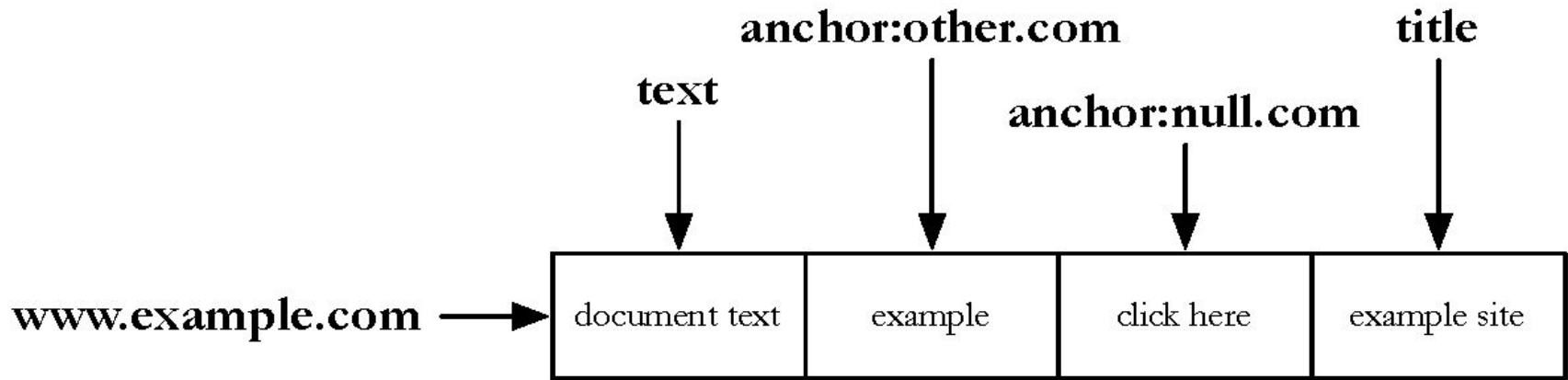


BigTable

- Không có ngôn ngữ truy vấn, không có câu truy vấn phức tạp để tối ưu hóa
- Các giao dịch ở mức dòng
- Các bảng con được lưu trong 1 hệ thống tệp dư thừa có thể được truy cập bởi tất cả các máy chủ BigTable
- Bất kỳ thay đổi nào đối với 1 bảng con trong BigTable đều được lưu trong lịch sử giao dịch cùng trong hệ thống tệp dùng chung.
- Nếu bất kỳ máy chủ chứa bảng con nào gặp sự cố, các bảng con được lưu trong đó có thể ngay lập tức được thay thế bằng các bảng con tương đương đang được lưu ở những máy khác.

BigTable

- Dữ liệu theo lô-gic được tổ chức thành các dòng
- Mỗi dòng lưu dữ liệu cho 1 trang Web



- Tổ hợp khóa dòng, tên cột, và 1 mốc thời gian xác định 1 ô duy nhất trong dòng

BigTable

- BigTable cho phép có 1 số lượng lớn cột trong 1 dòng
 - Các cột được gom lại thành các nhóm
 - Các nhóm cột cho các dòng là giống nhau
 - Nhưng các cột cho các dòng có thể khác nhau
 - Quan trọng để giảm số lần đọc ổ đĩa khi truy cập dữ liệu
- Các dòng được phân chia vào các bảng con dựa trên khóa dòng
 - Giúp máy khách có thể xác định máy chủ đang lưu dòng cần truy cập

