

CS 347:
Distributed Databases and
Transaction Processing

**Distributed
Information Retrieval**

Hector Garcia-Molina
Zoltan Gyongyi

CS 347

Distributed IR

1

Web Search Engine

- Crawling
- Indexing
- Computing ranking features
- Serving queries

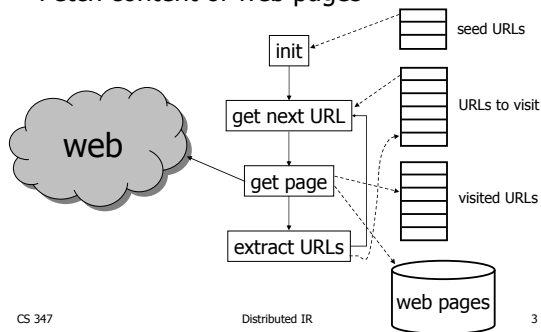
CS 347

Distributed IR

2

Crawling

- Fetch content of web pages



CS 347

Distributed IR

3

Issues

- Scope and freshness
 - Not enough space/time to crawl “all” pages
 - Page importance, quality, and update frequency
 - Site mirrors and (near) duplicate pages
 - Dynamic content and crawler traps
- Load at visited web sites
 - Rules in robots.txt
 - Limit number of visits per day
 - Limit depth of crawl

CS 347

Distributed IR

4

Issues

- Load at crawler
 - Variance of fetch latency/bandwidth
 - **Parallelization and scalability**
 - Multiple agents
 - Partitioning URL lists
 - Communication between agents
 - Recovering from agent failure

CS 347

Distributed IR

5

Crawl Partitioning

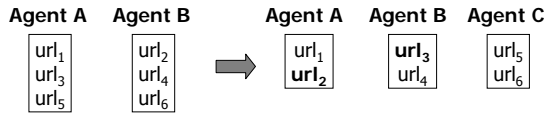
- Requirements
 - Each URL assigned to a single agent
 - Locally computable URL-to-agent mapping
 - Balanced distribution of URLs across agents
 - Contravariance

CS 347

Distributed IR

6

Contravariance

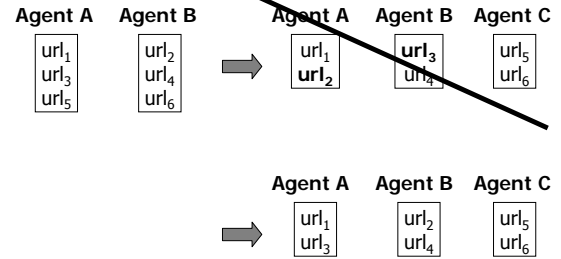


CS 347

Distributed IR

7

Contravariance



CS 347

Distributed IR

8

Assignment

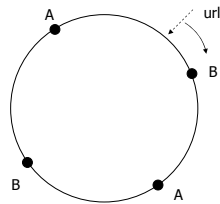
- Consistent hashing
 - Hash function: URL \rightarrow agent
 - Each agent “replicated” k times
 - Each replica mapped randomly on unit circle
 - Mapping persistent across agent restarts
 - Lookup: map URL on unit circle; find closest live replica

CS 347

Distributed IR

9

Assignment

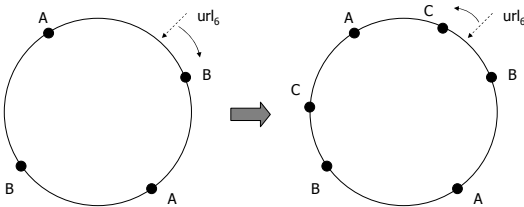


CS 347

Distributed IR

10

Assignment



- Balancing ✓
- Contravariance ✓

CS 347

Distributed IR

11

Crawl Partitioning

- Ideas
 - URL normalization
 - E.g., relative to absolute URL
 - Host-based partitioning
 - Reduces communication between agents
 - Small vs. large hosts
 - Geographic distribution

CS 347

Distributed IR

12

Fault Tolerance

- Repartitioning ✓
- Permanent failure
 - Recovering list of URLs to visit
 - Checkpoints
 - Communication logs
- Transient failure
 - Avoiding re-visiting URLs
 - Before fetch, check with near neighbor agents

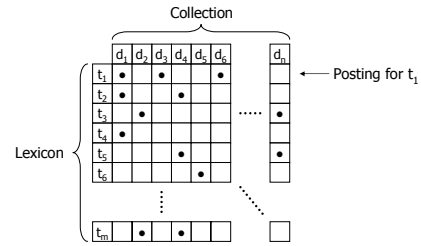
CS 347

Distributed IR

13

Indexing

- Build term-document index

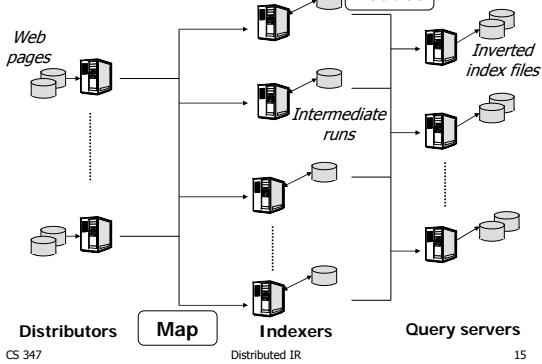


CS 347

Distributed IR

14

Architecture



CS 347

Distributed IR

15

Issues

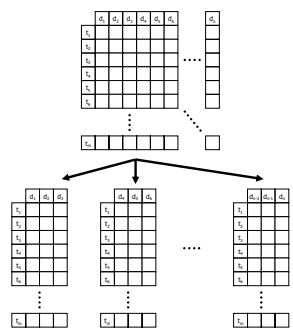
- Index partitioning
 - Efficient query processing
 - Query routing
 - Result retrieval

CS 347

Distributed IR

16

Document Partitioning



CS 347

Distributed IR

17

Document Partitioning

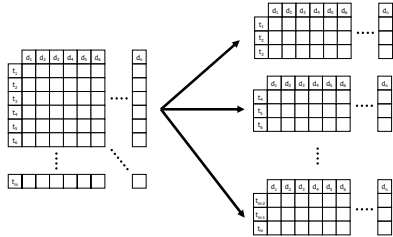
- Split the collection of documents
- Advantages
 - Easy to add new documents
 - Load balanced
 - High processing throughput
- Disadvantages
 - Communication with all query servers

CS 347

Distributed IR

18

Term Partitioning



CS 347

Distributed IR

19

Term Partitioning

- Split the lexicon
- Advantages
 - Reduced communication with query servers
- Disadvantages
 - More processing before partitioning
 - Adding new documents is hard
 - Load balancing is hard
 - Processing throughput limited by query length

CS 347

Distributed IR

20

Advanced Partitioning

- Topical partitioning using clustering
 - Documents clustered by term-similarity
 - Partitions made up of one or more clusters
- Usage-induced partitioning
 - Queries extracted from logs
 - Documents clustered by query-similarity
 - Partitions made up of one or more clusters

CS 347

Distributed IR

21

Ranking Feature Computation

- Parallel/distributed computation tasks
 - Text/language processing
 - Document classification/clustering
 - Web graph analysis

CS 347

Distributed IR

22

Example: PageRank

- Link-based global (query-independent) importance metric
 - Random surfer model
 - Start at a random page
 - With probability d , navigate to new page by following a random link on current page
 - With probability $(1 - d)$, restart at a random page
- ⇒ PageRank score = expected fraction of time spent at a page

CS 347

Distributed IR

23

Formula

$$p(x) = d \cdot \sum_{y \rightarrow x} p(y) / \text{out}(y) + (1 - d) / n$$

CS 347

Distributed IR

24

Formula

Probability of random restart at x

Out-degree of page y

$$p(x) = d \cdot \sum_{y \rightarrow x} p(y) / \text{out}(y) + (1 - d) / n$$

PageRank of page x

PageRank of y, where y links to x

CS 347 Distributed IR 25

Algorithm

```

i = 0
p[0](x) = (1 - d) / n
repeat
  i += 1
  p[i](x) = (1 - d) / n
  for all y → x
    p[i](x) += d · p[i-1](y) / out(y)
until | p[i] - p[i-1] | < ε
  
```

CS 347 Distributed IR 26

Implementation

- Two vectors, current and next
- Initialize vectors
- Iterate over all pages y, distribute PageRank from current(y) to next(x) for all links y → x
- current = next, re-initialize next
- Go back to iteration over pages or stop

CS 347 Distributed IR 27

Distribution

- MapReduce for each iteration i
- Map
 - Take <y, (current(y), edges(y))>
 - For each y → x in edges(y) emit <x, current(y) / |edges(y)|>
 - Also emit <y, edges(y)>
- Reduce
 - Take <x, val> and <x, edges(x)>
 - Sum (d · val) into next(x), add (1 - d) / n
 - Emit <x, (next(x), edges(x))>

CS 347 Distributed IR 28

Distribution

CS 347 Distributed IR 29

Query Processing

- Locate, retrieve, process, and serve query results

CS 347 Distributed IR 30

Architecture

- Multiple sites connected by WAN
 - Site = coordinator + servers + cache
- Partitioning
 - Parallel processing
 - Distributed storage of data
 - E.g., index partitioning
- Replication
 - Availability
 - Throughput
 - Response time

CS 347

Distributed IR

31

Issues

- Routing the query
 - To sites
 - E.g., identical sites + routing by dynamic DNS lookup
 - Within sites
- Merging the results
- Caching

CS 347

Distributed IR

32

Issues

	Routing	Merging
Document partition	All servers	Results selected by servers; ranking by coordinator
Term partition	Servers containing query terms	Selection and ranking by coordinator

CS 347

Distributed IR

33

Caching

- What to cache?
 - Query answers
 - Term postings

CS 347

Distributed IR

34

Caching

Query terms repeated more frequently than whole queries

- What to cache?
 - Query answers
 - Faster response
 - Term postings ✓
 - More hits

CS 347

Distributed IR

35

Caching Policy

- Terms most frequent in queries
 - high hit ratio
- Terms most frequent in documents
 - require more cache space (longer postings)
- Use static caching based on query/document frequency ratio

CS 347

Distributed IR

36

Summary

- **Crawling**
 - Partitioning: balancing and contravariance
 - Consistent hashing
- **Indexing**
 - Document, term, topical, and usage-induced partitioning
- **Computing ranking features**
 - PageRank with MapReduce
- **Serving queries**
 - Routing queries, merging results, and caching postings