Søgemaskiner på Internettet

Gerth Stølting Brodal og Rolf Fagerberg

BRICS, Daimi, Aarhus Universitet

Foredrag i Datalogforeningen, 29. januar 2003

Internettet

- Meget stor mængde information.
- Ustruktureret.

Hvordan finder man relevant info?

Internettet

- Meget stor mængde information.
- Ustruktureret.

Hvordan finder man relevant info?

Søgemaskiner!

Internettet

- Meget stor mængde information.
- Ustruktureret.

Hvordan finder man relevant info?

Søgemaskiner!

94: Lycos,. . .
96: Alta Vista: mange sider.
99: Google: mange sider *og* god ranking.

Moderne Søgemaskiner

Imponerende performance. F.eks. Google™:

- Søger i $3 \cdot 10^9$ sider (jan 03).
- Svartider $\approx$ 0,1 sekund
- 1000 brugere i sekundet.
- Finder relevante sider (*Do you feel lucky. . . ?*)

Moderne Søgemaskiner

Imponerende performance. F.eks. Google™:

- Søger i $3 \cdot 10^9$ sider (jan 03).
- Svartider $\approx$ 0,1 sekund
- 1000 brugere i sekundet.
- Finder relevante sider (*Do you feel lucky. . . ?*)

Hvem bruger bookmarks mere?

Umoderne Søgemaskiner

Ikke alle er (var) lige gode…

`princess diana`

Engine 1

Princess Diana Memorial WebRing
Follow the WebRing for a tour of memorial site
87% http://www.geocities.com/RainForest/Vines/1009/diana
1998
Grouped results from http://www.geocities.com

FOR DIANA, PRINCESS OF HEART - Dr. K
...
Dr. Kate Wachs Comments on Princess Diana T
84% http://www.therelationshipcenter.com/diana.shtml (Si

Princess Diana Editorial Cartoons! Cartoons
The Professional Cartoonists Index is the most
cartoonists o
daily cartoo
82% http://ww

Relevant and high quality

Diana, Princess of Wales
1 July 1961 - 31 August 1997 The BBC Web sit
Camera Press/Snowdon
79% http://www.royal.gov.uk/start.htm (Size 2.3K) Doc
Grouped results from http://www.royal.gov.uk

Engine 2

1. **Re: Lost in the shadow of Princess Diana**
[**URL:** www.spiceisle.com/talkshop/messages/6232.htm]
The Spicelslander TalkShop. [Follow Ups] [Pos
The Spicelslander TalkShop] Date: September
00:54:03 From: Sno,...
Last modified 12-Sep-97 - page size 4K - in English [Tran

2. **Re: Princess Diana's gown auction**
[**URL:** www.elle.com/textes/blablabla/forum/messages1/15
Re: Princess Diana's gown auction. [Follow Ups
Followup] [Elle International - Blablabla] Posted
September 07, 1997 at 02:15:26:...
Last modified 30-Mar-98 - page size 2K - in English [Tran

3. **Re: Princess Diana**
[**URL:** spicyhot.com/gaynet/messages/1053.html]
Re: Prince
Maine Ga
Novembe
Last modifie

Relevant but low quality

4. **Re: Princess Diana - Queen of Hearts**
[**URL:** www.elle.com/textes/blablabla/forum/messages1/28
Re: Princess Diana - Queen of Hearts. [Follow U
Followup] [Elle International - Blablabla] Posted
on August 31, 1997 at...
Last modified 30-Mar-98 - page size 4K - in English [Tran

Engine 3

1. Free Passwords To Adult Sites ...
99% - Articles & General info: Free Passwords
Sites ..... warez princess diana demi moore
magazine kathy ireland lingerie jennifer aniston cook
warez princess diana demi moore... 03/09/98
Commercial site: http://www.prurient.com /warez

2. SEX CHAT XXX NUDE PORNO PLAYBOY P...
[redacted]
Personal page: http://www.connix.com /~wgonzo
/sex/slidesuperall.htm

3. Ro...
[redacted]
Personal page: http://www.octet.com /~gonzo/jy

Not relevant index pollution

4. Sunday, 18-Jan-98
99% - Articles & General info: Sunday, 18-Jan-
CHAT XXX NUDE PORNO PLAYBOY PAME

Dette foredrag

Teknologien bag effektive søgemaskiner

Dette foredrag

Teknologien bag effektive søgemaskiner

NB: Søgemaskine $\neq$ Emnekatalog (Yahoo, Jubii,. . .)

Emnekataloger er stadig mest manuelt arbejde.

Behandles ikke her.

Dette foredrag

Teknologien bag effektive søgemaskiner

NB: Søgemaskine $\neq$ Emnekatalog (Yahoo, Jubii,. . .)

Emnekataloger er stadig mest manuelt arbejde.

Behandles ikke her.

Hvis tid: andre forskningsemner relateret til Internettet

Dette foredrag

Uddrag af kurset:

Algorithms for Web Indexing and Searching

Daimi, efteråret 2002

Dette foredrag

Uddrag af kurset:

Algorithms for Web Indexing and Searching

Daimi, efteråret 2002

For litteratur med yderligere detaljer, se:

`www.brics.dk/~gerth/webalg02/index.html`

Overblik

- ✓ Indledning
- Google facts
- Information Retrieval generelt
- Teknisk mellemspil
- En søgemaskines dele
  - Crawling
  - Indeksering
  - Søgning og ranking
- Gør-det-selv
- Yderligere emner

Google

- Startet i 1995 som forskningsprojekt ved Stanford University af ph.d. studerende **Larry Page** og **Sergey Brin**
- Privat firma grundlagt 1998
- 500 medarbejdere, 50 med en ph.d.
- Ansvarlig for ca. halvdelen af alle internet-søgninger
- 3 datacentre, to i Silicon Valley og et på USA's østkyst

$$\text{google} \approx \text{googol} = 10^{100}$$

Google™ – Services

Søgemaskine

- Hurtig
- Relevante links
- Opdateret
- Cache
- GoogleScout (lignende sider)
- Automatisk stavekontrol
- Interface til WAP og PDA
- Produktsøgninger (froogle)
- Billed søgning
- Aktiekurser, kort, ordbøger, nyheder, telefonbøger ...

Stemmestyret teknologi

AdWords

- tekstbaseret reklame
- query afhængig

Licenser

- AOL/Netscape, Red Hat, Virgin Group, YAHOO, The Washington Post ...

Web API

Hardware + Software

Google™ – Fakta

- +3.000.000.000 web sider
- +35.000.000 ikke HTML sider
- +700.000.000 USENET beskeder (20 år)
- +2 Terabyte index, opdateres en gang om måneden
- +2.000.000 termer i indeks
- +150.000.000 søgninger om dagen (2000 i sekundet)
- +200 filtyper: HTML, Microsoft Office, PDF, PostScript, WordPerfect, Lotus ...
- 28 sprog

Google™ – Hardware

- Cluster af +10.000 Intel servere
  - Single-processor
  - 256 MB–1 GB RAM
  - 2 IDE diske med 20-40 Gb
- Fejl-tolerance: Redundans
- Hastighed: Load-balancing
- Netværk
  - Gigabit ethernet backbone
  - Round-robin DNS for at dirigere traffiken til data centrerne (fremtiden Border Gateway Protocol)
  - Internt i data centrerne bruges egenudviklet load-balancing software

Google™ – Software

Red Hat Linux

- Pris: “bought 50 copies ... goodwill gesture”
- Adgang til kildetekst
- In-house support
- Stort rekruteringsgrundlag
- Mest udbredte linux

Fjernet

- X windows
- Apache Web Server
- Telnet ...

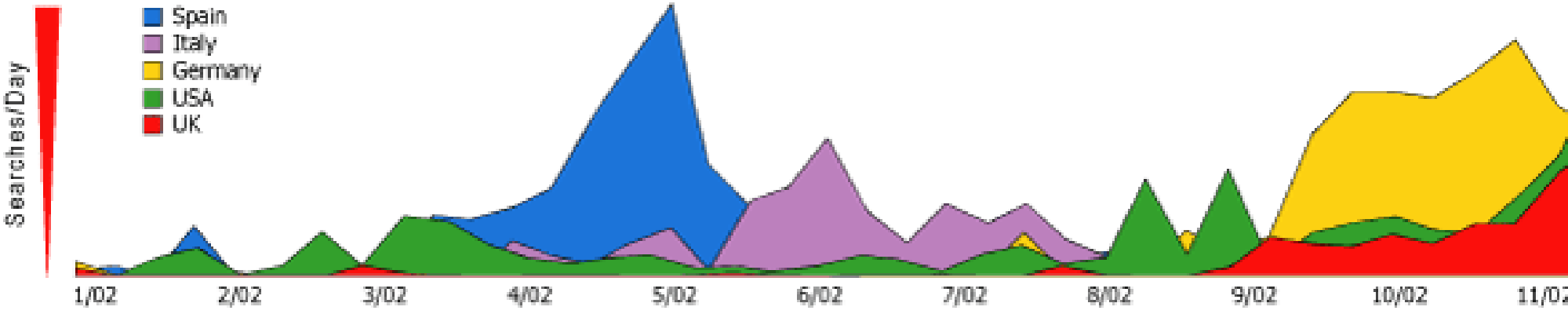
Bibeholdt

- Emacs

- Alle maskiner har identisk grundkonfiguration (kan bruges til både webserver, indeks, ...)
- Stateless services
- Egenudviklet automatisk installations værktøjer (Linux halter efter SUN)

Google™ – Zeitgeist

Las Ketchup



Overblik

- √ Indledning
- √ Google facts
- Information Retrieval generelt
- Teknisk mellemspil
- En søgemaskines dele
  - Crawling
  - Indeksering
  - Søgning og ranking
- Gør-det-selv
- Yderligere emner

Information Retrieval

Generelt:

- Lav indeks over data.
- Søg i indekset:
  - Find alle relevante dokumenter.
  - Rank (orden) dokumenterne efter relevans, vis mest relevante først.

Klassisk Information Retrieval (IR):

Metoder til homogene samlinger af tekst dokumenter. Moderat antal dokument.

Eksempler: Biblioteker, nyhedsarkiver, videnskabelige dokumentsamlinger.

IR på Internettet

Vanskeligheder:

- Dokumenter er ikke lokale.
- Dokumenter er meget forskellige.
- Dokumentsamling ikke statisk (dokumenter ændrer sig, tilføjes, forsvinder).
- Meget stort antal dokumenter (milliarder af dokumenter, samlet størrelse måles i Terabytes).
  - Pladsforbrug og svartider kritiske. Distribution og parallelisme er nødvendigt.
  - Mange (f.eks. 100.000) relevante dokumenter for mange søgninger. God ranking er essentiel.

Fordele:

- Ekstra struktur: links.

IR på Internettet

Yderligere udfordringer:

- Mange næsten ens dokumenter (30%)
- Bruger meget forskellige, men utålmodige. Avancerede søgemuligheder bruges ikke.
- Indexering af og søgning efter ikke-tekstuelle dokumenter.
  - Multimedie.
  - Databaser.

Dette foredrag omhandler kun tekstdokumenter.

Overblik

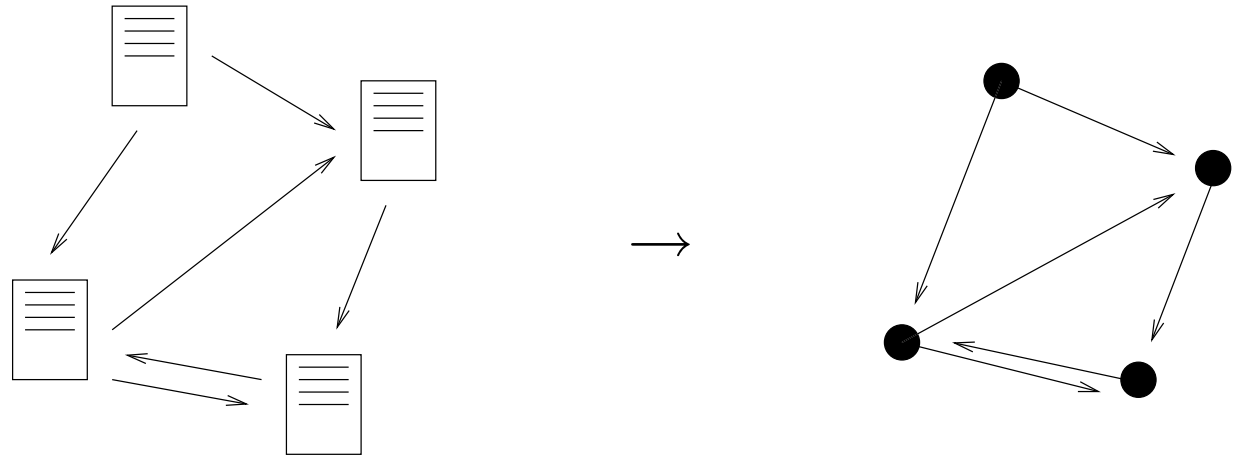
- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- Teknisk mellemspil
- En søgemaskines dele
  - Crawling
  - Indeksering
  - Søgning og ranking
- Gør-det-selv
- Yderligere emner

Internetgrafen

Internettet = en orienteret graf

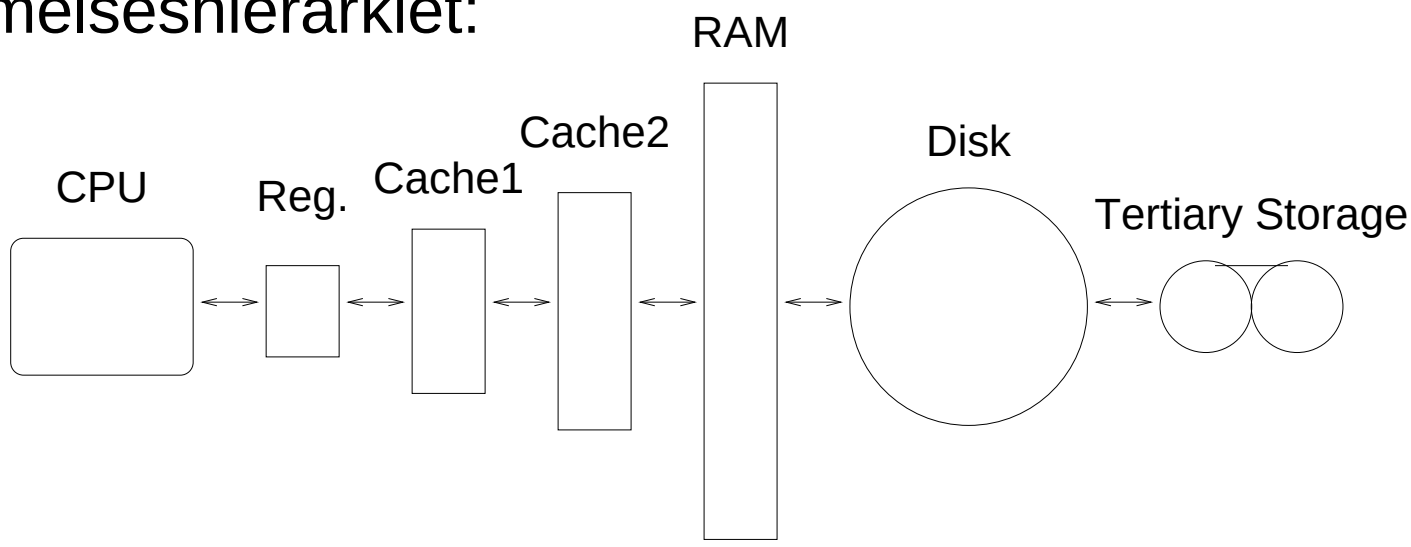
knuder = sider (URL'er)

kanter = links



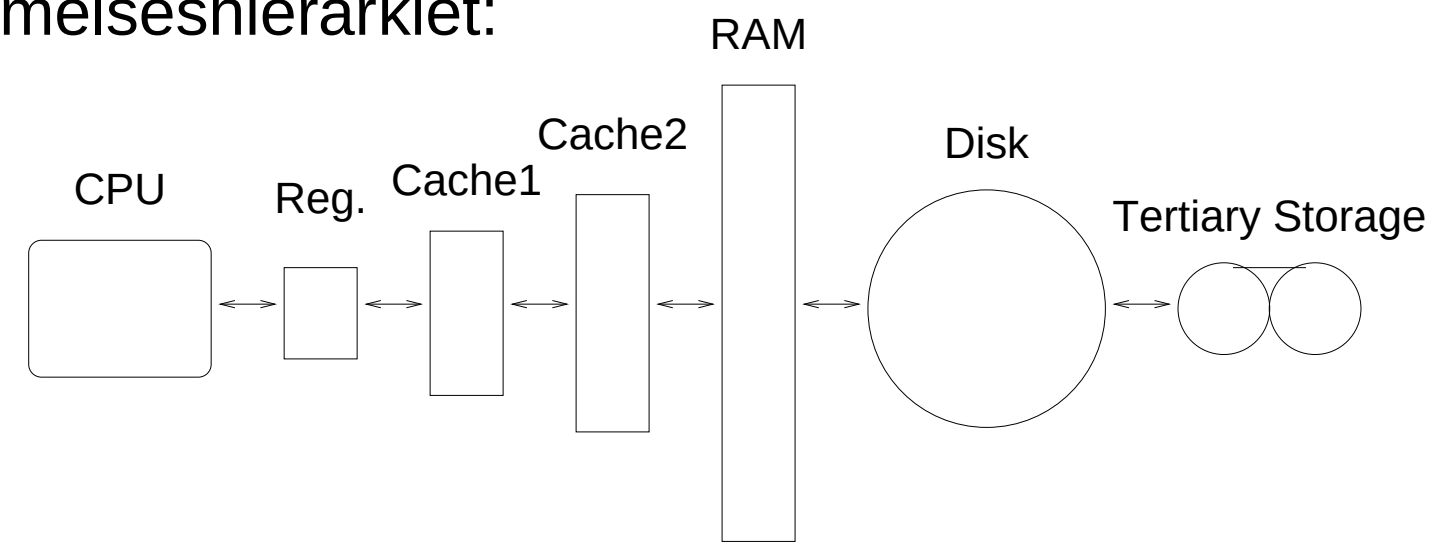
Massive datamængder

Hukommelseshierarkiet:



Massive datamængder

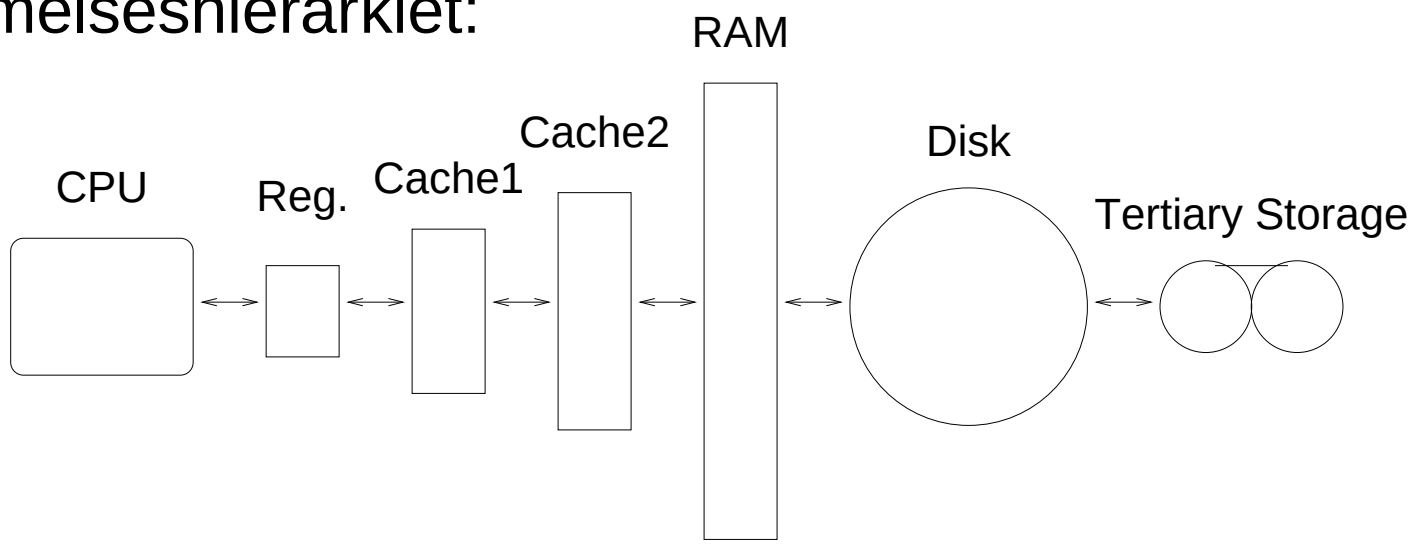
Hukommelseshierarkiet:



	Access time	*Volume*
Registers	1 cycle	1 Kb
Cache2	5 cycles	512 Kb
RAM	50 cycles	256 Mb
Disk	2,000,000 cycles	80 Gb

Massive datamængder

Hukommelseshierarkiet:



	Access time	*Volume*
Registers	1 cycle	1 Kb
Cache2	5 cycles	512 Kb
RAM	50 cycles	256 Mb
Disk	2,000,000 cycles	80 Gb

For datamængder > > RAM:

Disk access er flaskehalsen

⇓

Disk access skal minimeres

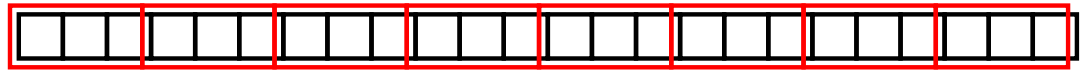
(ikke CPU tid)

Generisk eksempel

Én disk-access (I/O) overfører blok (page) med B elementer.

To $O(n)$ algoritmer:

1. Virtuel huk. tilgås tilfældigt $\Rightarrow$ page fault ved hver tilgang.
2. Virtuel huk. tilgås sekventiel $\Rightarrow$ page fault hver B'te tilgang.

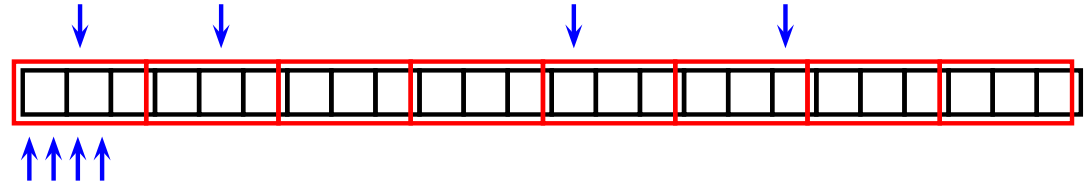


Generisk eksempel

Én disk-access (I/O) overfører blok (page) med B elementer.

To $O(n)$ algoritmer:

1. Virtuel huk. tilgås tilfældigt $\Rightarrow$ page fault ved hver tilgang.
2. Virtuel huk. tilgås sekventiel $\Rightarrow$ page fault hver B'te tilgang.

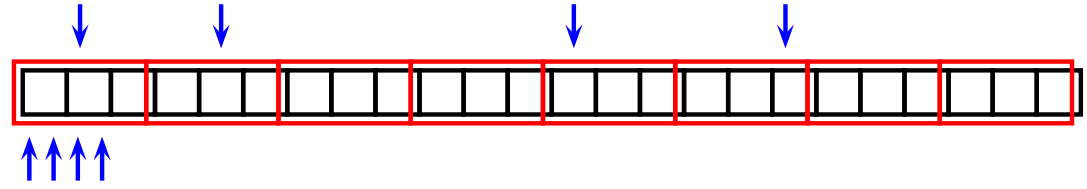


Generisk eksempel

Én disk-access (I/O) overfører blok (page) med B elementer.

To $O(n)$ algoritmer:

1. Virtuel huk. tilgås tilfældigt $\Rightarrow$ page fault ved hver tilgang.
2. Virtuel huk. tilgås sekventiel $\Rightarrow$ page fault hver B'te tilgang.



$O(N)$ I/O'er *vs.* $O(N/B)$ I/O'er

Typically, $B \sim 10^3$.

Specifikt Eksempel

QuickSort $\sim$ sekventiel tilgang

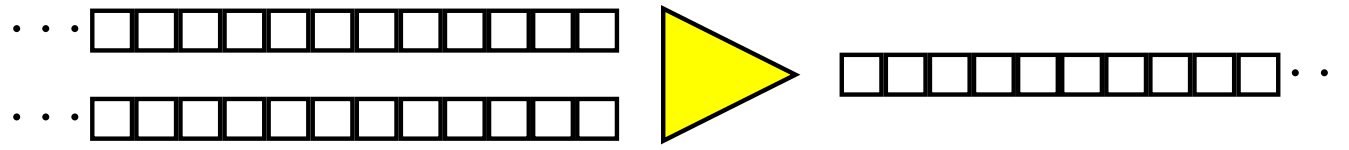
vs.

HeapSort $\sim$ tilfældig tilgang

QuickSort:	$O(N \log_2(N/M)/B)$
HeapSort:	$O(N \log_2(N/M))$

Standard MergeSort

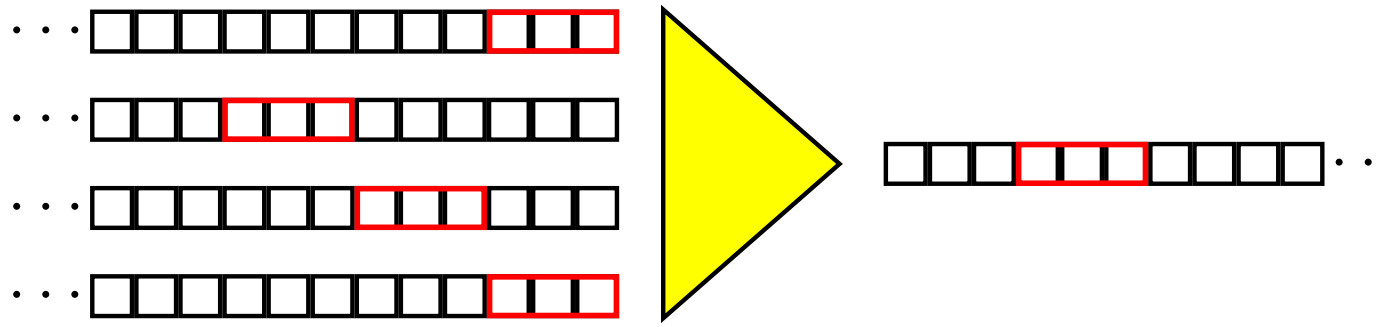
Merge af to sorterede sekvenser $\sim$ sekventiel tilgang.



MergeSort: $O(N \log_2(N/M)/B)$ I/O'er

Multiway MergeSort

Merge af M/B sorterede sekvenser (M = RAM størrelse).



Multiway MergeSort: $O(N \log_{M/B}(N/M)/B)$ I/O'er

Overblik

- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- √ Teknisk mellemspil
- En søgemaskines dele
  - Crawling
  - Indeksering
  - Søgning og ranking
- Gør-det-selv
- Yderligere emner

En søgemaskines dele

Indsamling af data:

- Webcrawling (gennemløb af internetgrafen).

Indeksering data:

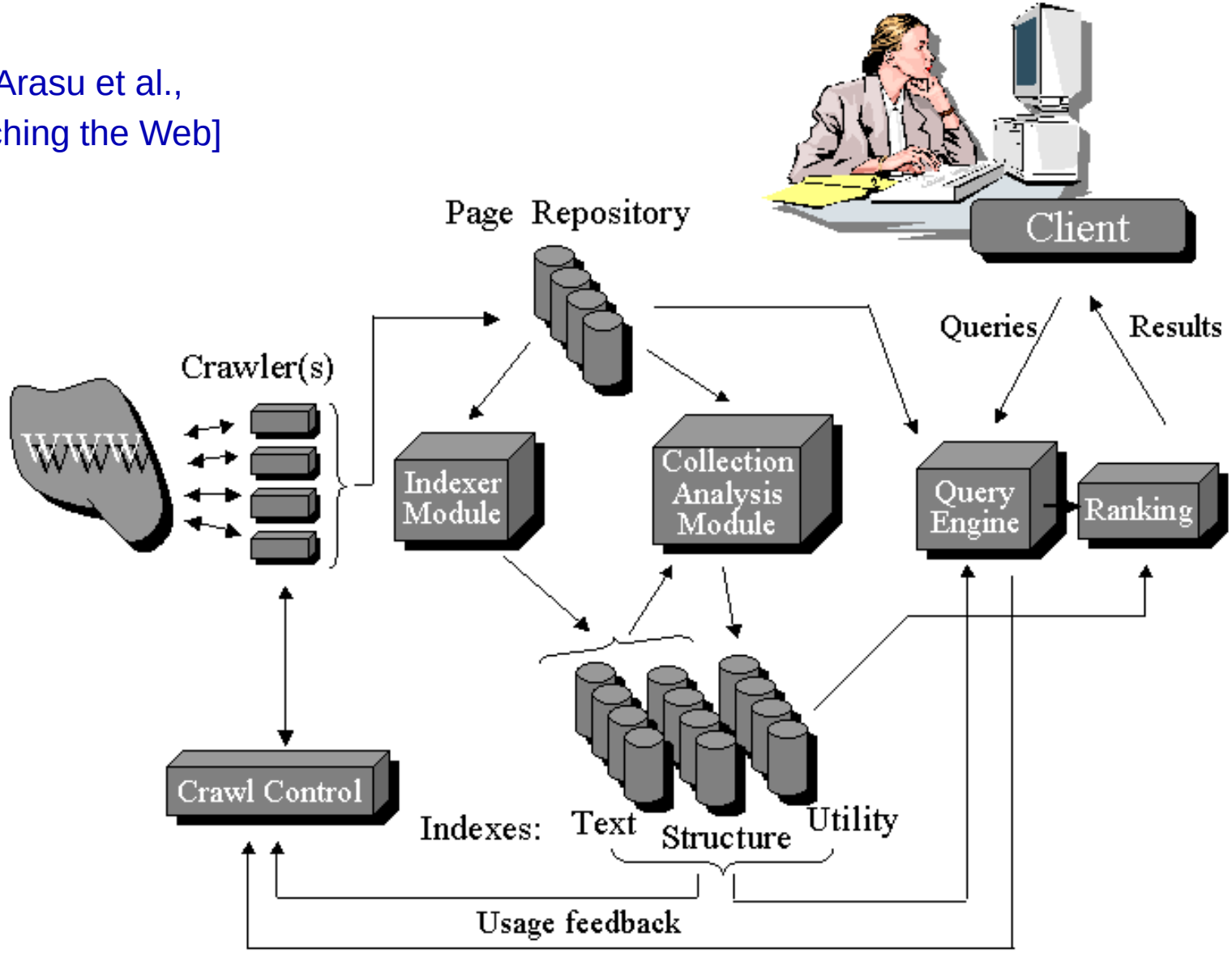
- Parsning af dokumenter.
- Lexicon: indeks (ordbog) over alle ord mødt.
- Inverted file: for alle ord i lexicon, angiv i hvilke dokumenter de findes.

Søgning i data:

- Find alle dokumenter med søgeordene.
- Rank dokumenterne.

Generel opbygning

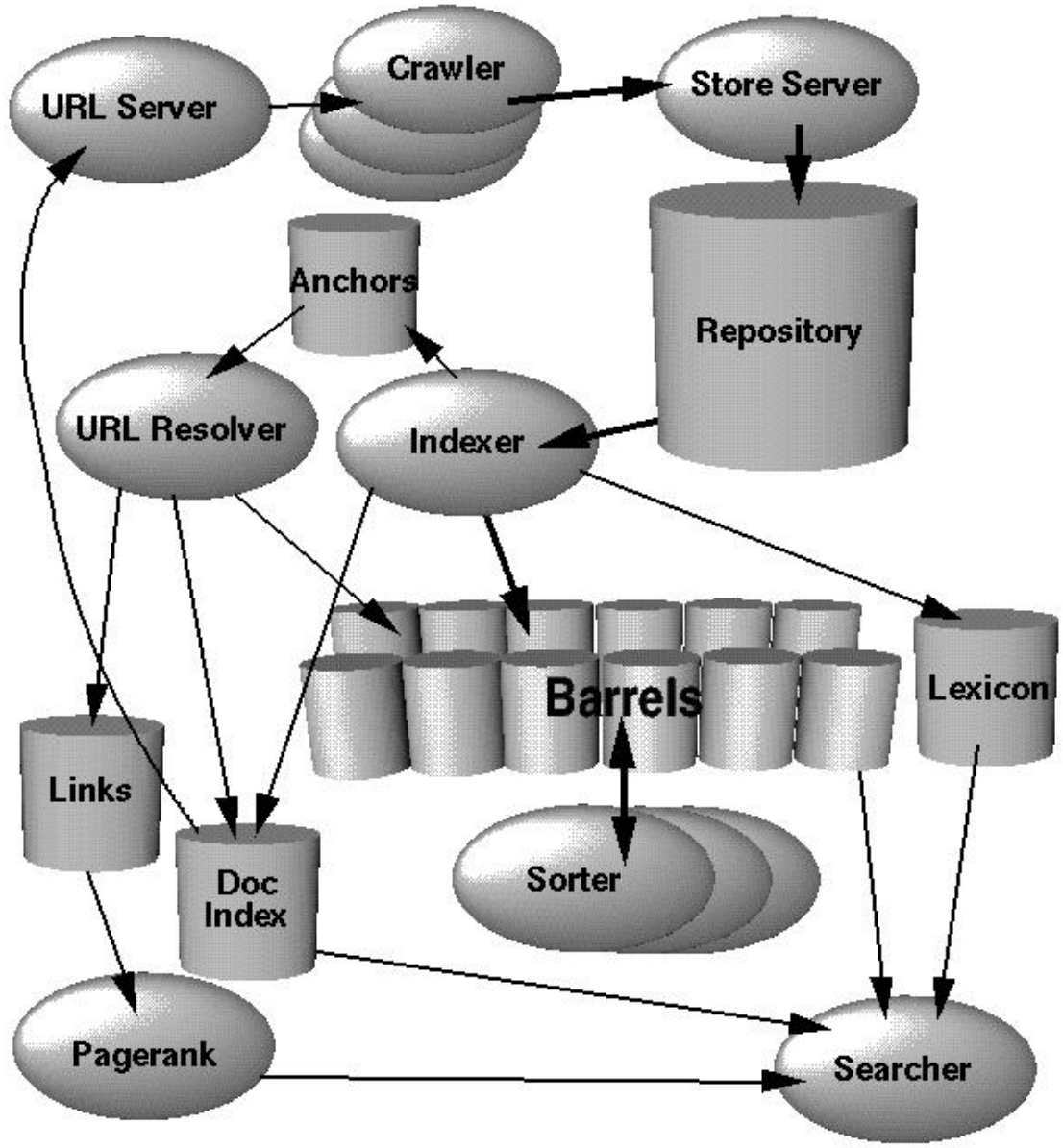
[Fra: Arasu et al., Searching the Web]



Konkret eksempel

Google:
(1998)

[Fra:
Brin and Page,
Anatomy of. . .]



Overblik

- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- √ Teknisk mellemspil
- √ En søgemaskines dele
  - Crawling
  - Indeksering
  - Søgning og ranking
- Gør-det-selv
- Yderligere emner

Crawling

Webcrawling = Grafgennemløb

S = {startside}
repeat
    fjern en side s fra S
    parse s og find alle links (s, v)
    foreach (s, v)
        if v ikke besøgt før
            indsæt v i S

Designovervejelser

- Startpunkt (initial S).
- Crawl-strategi (valg af s).
- Mærkning af besøgte sider.
- Robusthed.
- Ressourceforbrug (egne og andres ressourcer).
- Opdatering. Kontinuert vs. periodisk crawling.

S = {startside}
repeat
    fjern en side s fra S
    parse s og find alle links (s, v)
    foreach (s, v)
        if v ikke besøgt før
            indsæt v i S

Output: DB med besøgte dokumenter.
DB med links i disse (kanterne i Internetgrafen)
DB med DokumentID–URL mapning

Crawl-strategier

- Breath First Search
- Depth First Search
- Random
- Priority Search

Mulige prioriteter:

- Sider som opdateres ofte (kræver metode til at estimatere opdateringsfrekvens).
- Efter vigtighed (kræver metode til at estimere vigtighed, f.eks. PageRank).

BFS virker godt

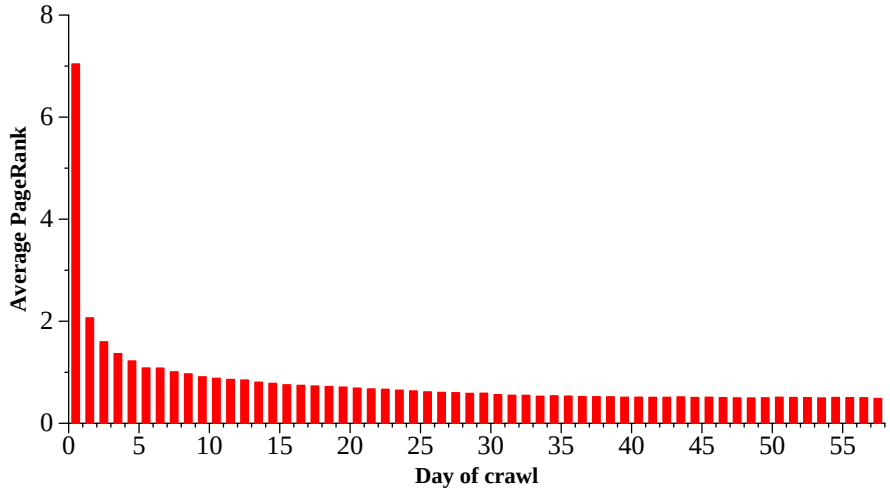


Figure 1: Average PageRank score by day of crawl

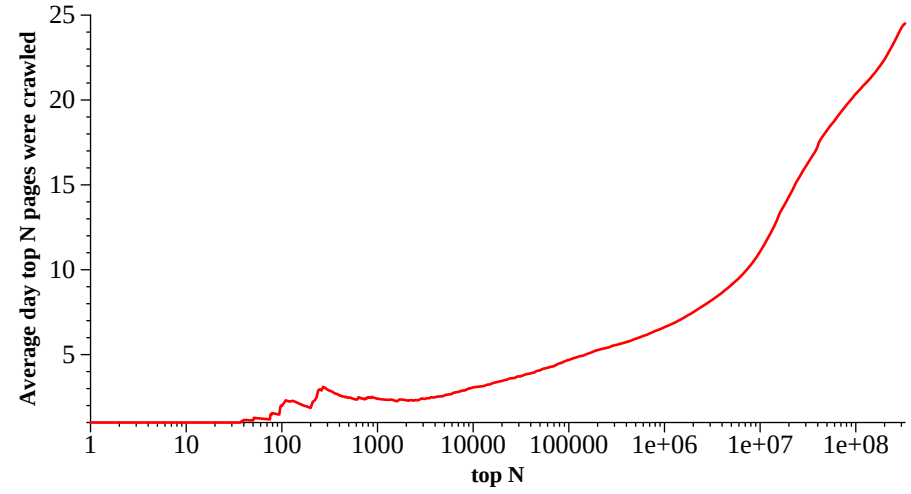


Figure 2: Average day on which the top N pages were crawled

[Fra: Najork and Wiener, 2001]

Fra et crawl af 328 millioner sider.

PageRank prioritet er endnu bedre

(men mere beregningstung. . .)

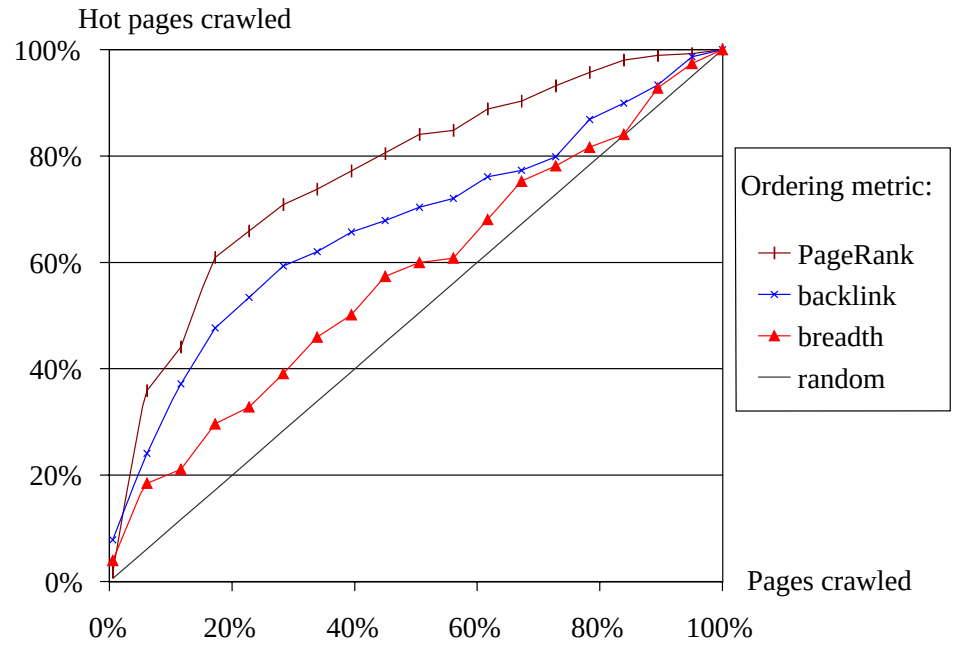


Figure 2: The performance of various ordering metrics for $IB(P)$; $G = 100$

[Fra: Arasu et al., 2001]

Fra et crawl af 225.000 sider på Stanford University.

Mærkning af besøgte sider

$|S|$ < RAM:

- Hashtabel over besøgte URL'er.

$|S|$ > RAM:

- Fil med sorteret liste af (hashværdier af) besøgte URL'er.
- Ryd S op en gang i mellem:
  - Sorter S.
  - Scan S og listen af besøgte URL'er. Opdatér begge under scan.

Robusthed

- Normalisering af URLer.
- Parsning af malformet HTML.
- Mange filtyper.
- Forkert content-type fra server.
- Forkert HTTP response code fra server.
- Enorme filer.
- Uendelige URL-løkker (crawler traps).

⋮

Robusthed

- Normalisering af URLer.
- Parsning af malformet HTML.
- Mange filtyper.
- Forkert content-type fra server.
- Forkert HTTP response code fra server.
- Enorme filer.
- Uendelige URL-løkker (crawler traps).

⋮

Vær konservativ – opgiv at finde alt.
Crawling tager måneder – brug checkpoints.

Ressourceforbrug

Egne ressourcer

- Båndbredde (global request rate)
- Lagerplads (brug kompakte representationer)
- Distribuér på flere maskiner (opdel f.eks. rummet af ULR’er)

Ressourceforbrug

Egne ressourcer

- Båndbredde (global request rate)
- Lagerplads (brug kompakte representationer)
- Distribuér på flere maskiner (opdel f.eks. rummet af ULR'er)

Andres ressourcer (politeness)

- **Båndbredde** (lokal request rate). Tommelfingerregel: 30 sekunder mellem request til samme site.
- Robots Exclusion Protocol (`www.robotstxt.org`).
- Giv kontakt info i HTTP-request.

Erfaringer ang. effektivitet

- Brug caching (DNS opslag, `robots.txt` files, senest mødte URL'er).
- Flaskehals er ofte I/O under tilgang til datastrukturerne
- CPU cycler er ikke flaskehals (Java og scripting languages er OK).

Erfaringer ang. effektivitet

- Brug caching (DNS opslag, `robots.txt` files, senest mødte URL'er).
- Flaskehals er ofte I/O under tilgang til datastrukturerne
- CPU cycler er ikke flaskehals (Java og scripting languages er OK).

En tunet crawler (på een eller få maskiner) kan crawle

200-400 sider/sek $\sim$ 35 mio sider/dag.

Konkret eksempel: Mercator

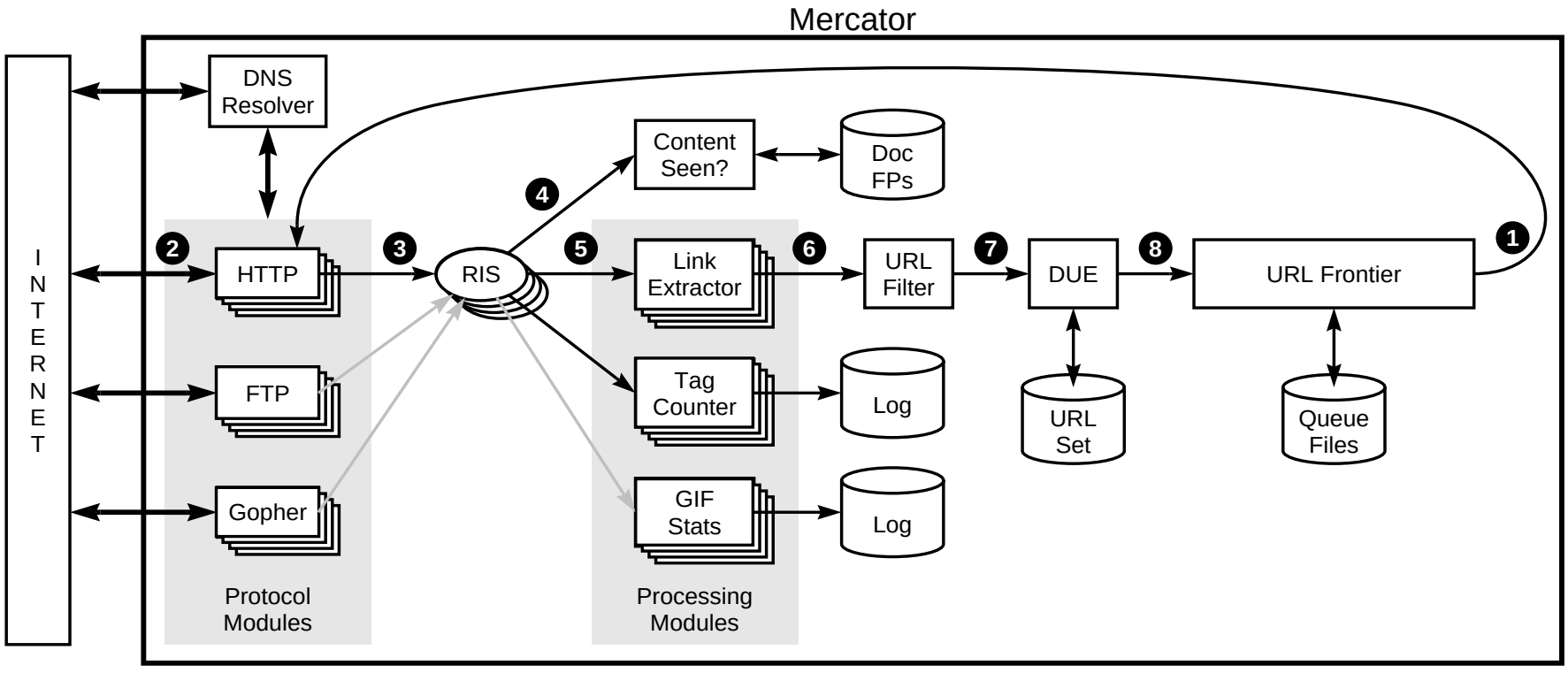


Figure 1: Mercator's main components.

[Fra: Najork and Heydon, 2001]

Detaljer: Politeness

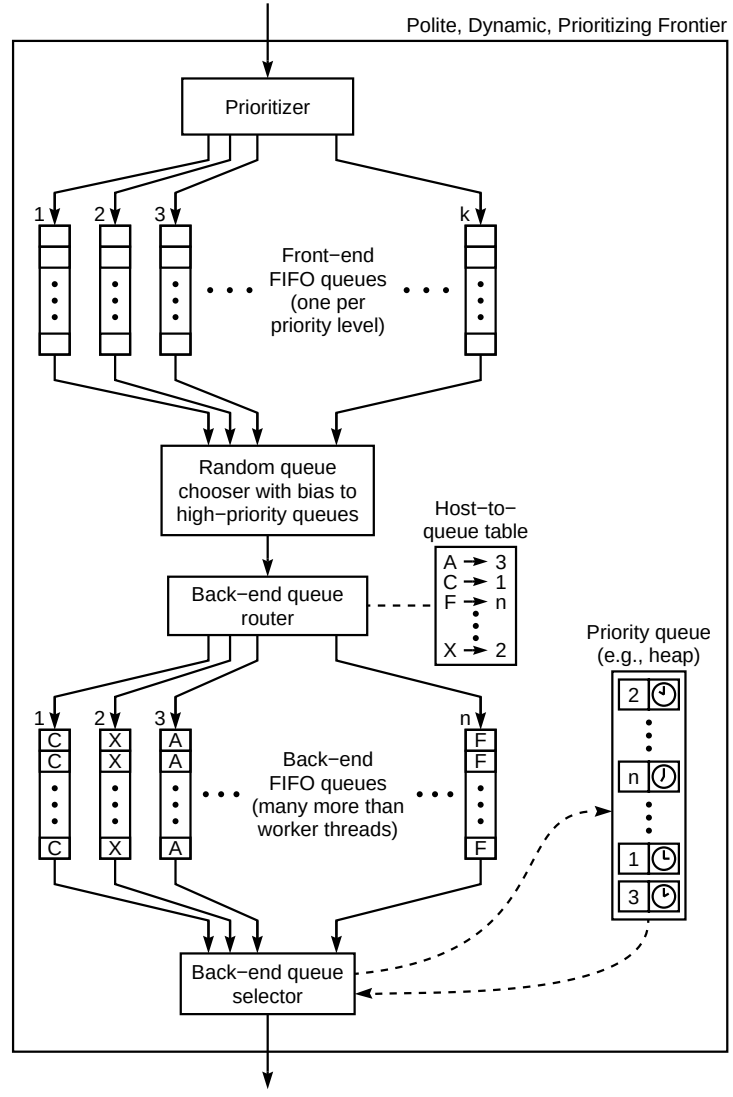


Figure 3: Our best URL frontier implementation

[Fra: Najork and Heydon, 2001]

Statistik

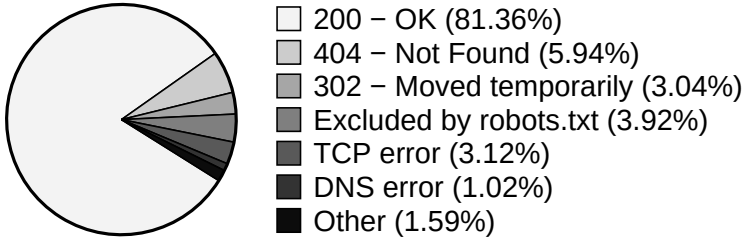


Figure 6: Outcome of download attempts

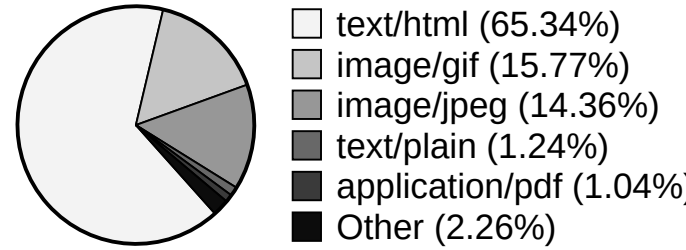


Figure 7: Distribution of content types

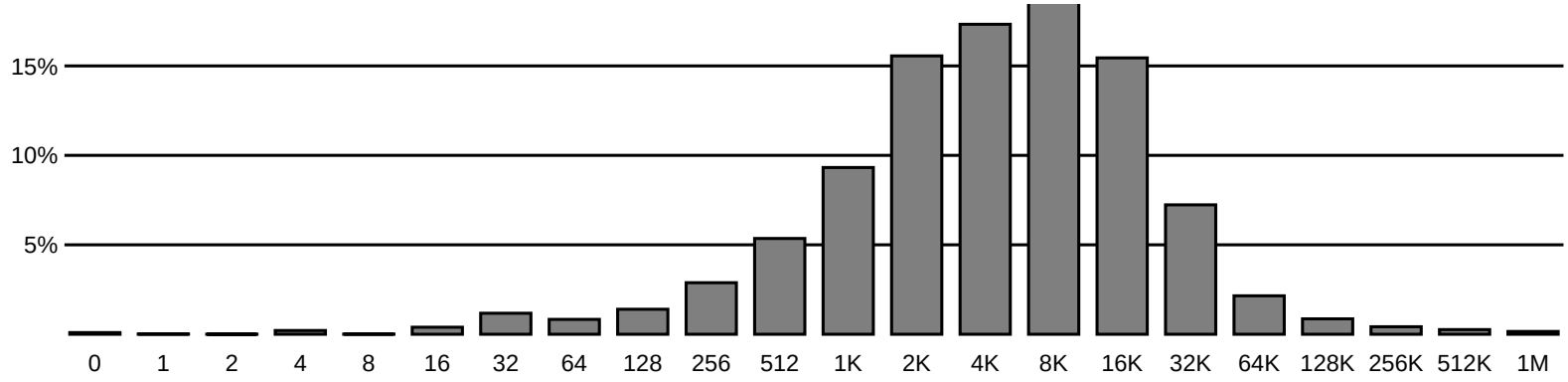
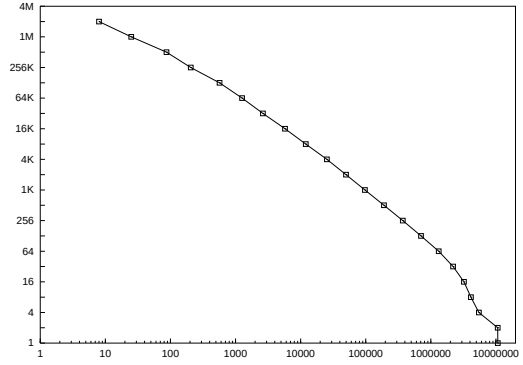
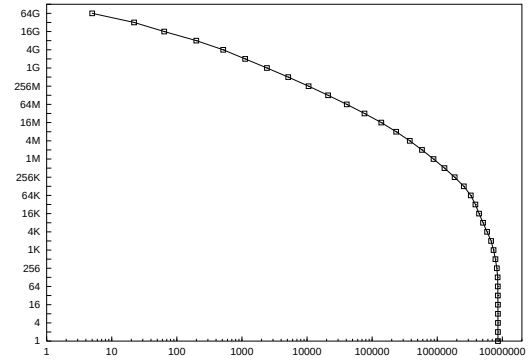


Figure 8: Distribution of document sizes

Statistik

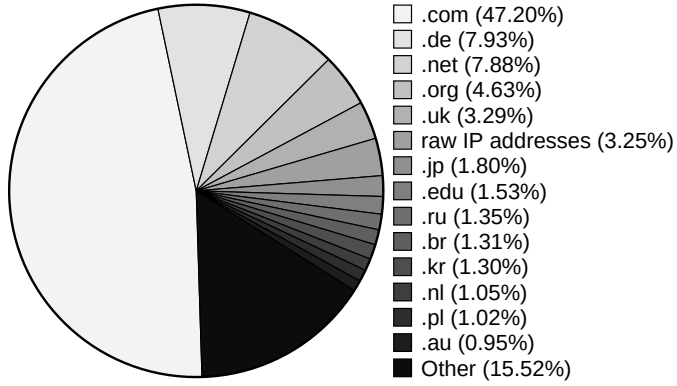


(a) Distribution of pages over web servers

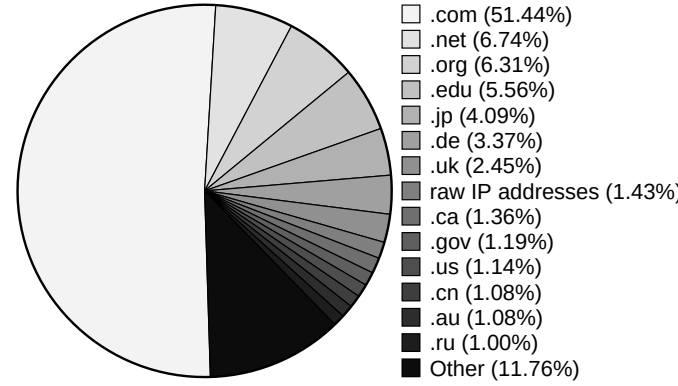


(b) Distribution of bytes over web servers

Figure 9: Document and web server size distributions



(a) Distribution of hosts over



(b) Distribution of pages over

[Fra: Najork and Heydon, 2001]

Overblik

- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- √ Teknisk mellemspil
- √ En søgemaskines dele
  - √ Crawling
  - Indeksering
  - Søgning og ranking
- Gør-det-selv
- Yderligere emner

Indeksering af dokumenter

Opgave:

> Preprocessér en dokumentsamling så dokumenter med et givet søgeord kan blive returneret hurtigt.

Input: dokumentsamling.

Output: søgestruktur

Løsning

Inverted file + lexicon

- Inverted file = for hvert ord w en liste af dokumenter indeholdende w.
- Lexicon = ordbog over alle ord i dokumentsamlingen.

  (key = ord, value = pointer til liste i inverted file + evt. ekstra info for ordet, f.eks. længde af listen)

Løsning

Inverted file + lexicon

- Inverted file = for hvert ord w en liste af dokumenter indeholdende w.
- Lexicon = ordbog over alle ord i dokumentsamlingen.

  (key = ord, value = pointer til liste i inverted file + evt. ekstra info for ordet, f.eks. længde af listen)

For en milliard dokumenter:

Inverted files $\sim$ totalt antal ord $\geq$ 100 mia — Disk

Lexicon $\sim$ antal forskellige ord $\sim$ 1 mio — RAM

Lexicon

Kan være i RAM, så almindelige ordbogs-datastrukturer er OK. F.eks.:

- Binær søgning i sorteret liste af ord.
- Hash tabeller.
- Tries, suffix træer, suffix arrays.

Inverted File

- Simpel (forekomst af ord i dokument):

  ord_1: DocID, DocID, DocID
  ord_2: DocID, DocID
  ord_3: DocID, DocID, DocID, DocID, DocID,...
  ⋮

- Detaljeret (*alle* forekomster af ord i dokument):

  ord_1: DocID, Position, Position, DocID, Position...
  ⋮

- Endnu mere detaljeret:

  Forekomst annoteret med info (heading, boldface, anchor text,...). Kan bruges under ranking.

Komprimer inverted file

- Specifikke metoder
  - Gem differencen mellem DocID'er (ikke absolutte DocID'er).
  - Kod denne difference effektivt.
- Generiske værktøjer (zip,. . .)
  - Komprimer hver liste.
  - Opdel lister i blokke, komprimer hver blok.

Parsning af dokumenter

- Find ord
  - Fjern mark-up, scripts,. . .
  - Definition af ord? (sekvens af alfanumeriske tegn, længde max 256, max 4 digits).
  - Lowercase
  - Tegnsæt? ascii, latin-1, Unicode,. . . .
- Stemming? (“funktion”, “funktionalitet”,. . . $\longrightarrow$ “funktio”).
- Stop ord? (udelad hyppige ord som “og”, “er”,. . .).

Bygning af index

foreach dokument D i samlingen
    Parse D og identificér ord
    foreach ord w
        Udskriv (DocID, w)
        if w ikke i lexicon
            indsæt w i lexicon

$\Downarrow$

$(1,2),(1,37),\ldots,(1,123)$, $(2,34),(2,37),\ldots,(2,101)$, $(3,486),\ldots$

External Sorting $\surd$ $\Downarrow$ Hashing $\div$

$(22,1),(77,1),\ldots,(198,1)$, $(1,2),(22,2),\ldots,(345,2)$, $(67,3),\ldots$

$\approx$ inverted file

Overblik

- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- √ Teknisk mellemspil
- √ En søgemaskines dele
  - √ Crawling
  - √ Indeksering
  - • Søgning og ranking
- • Gør-det-selv
- • Yderligere emner

Søgning og Ranking

Query: computer AND science:

1. Slå computer og science op i lexicon. Giver adresse på disk hvor deres lister starter.
2. Scan disse lister og "flet" dem (returnér DocID'er som er med i begge lister).

   computer: 12, 15, 117, 155, 256,. . .
   science: 5, 27, 117, 119, 256,. . .
3. Udregn rank af fundne DocID'er. Hent de 10 højst rank'ede i dokumentsamling og returnér URL samt kontekst fra dokument til bruger.

OR og NOT kan laves tilsvarende. Hvis lister har ord-positioner kan frase-søgninger ("computer science") og proximity-søgningner ("computer" tæt på "science") også laves.

Tekstbaseret ranking

Vægt forekomsten af et ord med f.eks.

- Antal forekomster i dokumentet.
- Ordets typografi (fed skrift, overskrift,. . .)
- Forekomst i META-tags.
- Forekomst i tekst ved links som peger på siden

Forbedring, men ikke nok på Internettet (rankning af f.eks. 100.000 relevante dokumenter).

Let at spamme (fyld siden med søge-ord).

Linkbaseret ranking

Idé 1: Link til en side $\approx$ anbefaling af den.

Vægt en side med dens indgrad i webgrafen.

MEN: Let at spamme (opret en masse links til din side).

Linkbaseret ranking

Idé 1: Link til en side $\approx$ anbefaling af den.

Idé 2: Anbefalinger fra vigtige sider skal vægte mere.

PageRank

Rekursiv def: $$r(j) = \sum_{i \in B(j)} r(i)/N(i)$$

$r(j)$ = pagerank af side j,
$B(j)$ = sider som peger på side j,
$N(i)$ = udgrad af side i.

$$\vec{r} = \vec{r}A$$

A = nabomatricen for webgrafen
(normaliseret, d.v.s. indgangene i række i divideret med $N(i)$).

Beregning af PageRank

PageRank vektoren $\vec{r}$ er egenvektor for A.

$$\vec{r} = \vec{r}A$$

Matematisk teori (ergodisk sætning om random walks):

For vilkårlig startvektor x:

$$\vec{x}A^k \to r \quad \text{for} \quad k \to \infty$$

hvis A opfylder visse betingelser.

Beregning af PageRank

For at opfylde betingelser i PageRank: erstat A med

$$0.85A + 0.15E \ ,$$

hvor E er en (normaliseret) nabomatrice som indeholder kanter fra alle sider til alle sider. Vægtningen 85–15% er valgt ud fra at den har vist sig god i praksis.

Beregning: Gentag

$$\vec{r}_{\text{ny}} = \vec{r}_{\text{gl.}}(0.85A + 0.15E)$$

I praksis: 20-50 iterationer er nok.

PageRank $\approx$ websurfer

PageRank beregning kan opfattes som en websurfer som (i uendelig lang tid) i hver skridt

- med 85% ss. vælger at følge et tilfældigt link fra nuværende side,
- med 15% ss. vælger at gå til en tilfældig side i hele internettet.

PageRank for en side x er lig den procentdel af hans besøg som er til side x.

Med andre ord: PageRank-vektoren er den stationære fordeling for ovennævnte Random Walk.

Overblik

- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- √ Teknisk mellemspil
- √ En søgemaskines dele
  - √ Crawling
  - √ Indeksering
  - √ Søgning og ranking
- Gør-det-selv
- Yderligere emner

Gør-det-selv

MIA – Searching the Danish Web

Programmeringsprojekt i kurset Algorithms for Web Indexing and Searching.

- Opgave: lav en søgemaskine for domæne `.dk` .
- 15 studerende.
- 4 parallelt arbejdende grupper (crawling, indexing, PageRank, søgning/brugergrænseflade).
- Google $\approx$ googol, Mia $\approx$ milliard.

Se resultatet på:

`http://mia.lir.dk/`

Overblik

- √ Indledning
- √ Google facts
- √ Information Retrieval generelt
- √ Teknisk mellemspil
- √ En søgemaskines dele
  - √ Crawling
  - √ Indeksering
  - √ Søgning og ranking
- √ Gør-det-selv
- • Yderligere emner

Yderligere Emner

F.eks.:

- Andre link-baserede ranking metoder (Kleinbergs “HITS”).
- Egenskaber ved webgrafen.
- Graf-modeller som kan forudsige disse egenskaber.
- Identifikation af mirrors/dubletter (websider og hele websites).
- Spil-teori anvendt på Internettet.

Se litteraturlisten på

`ww.brics.dk/~gerth/webalg02/index.html`