Sotto il cofano del ragno

Considerazioni generali e dettagli implementativi per uno small scale crawler

Massimo Santini

msantini@unimo.it

Dipartimento di Scienze Sociali, Cognitive e Quantitative

Università degli Studi di Modena e Reggio Emilia

Introduzione

Passo per passo

Seguiamo nel tempo il comportamento del crawler illustrato nell’introduzione:

Passo per passo

Seguiamo nel tempo il comportamento del crawler illustrato nell'introduzione:

PRELEVARE

aspetti di rete, buona educazione ed efficienza,

Passo per passo

Seguiamo nel tempo il comportamento del crawler illustrato nell'introduzione:

PRELEVARE

aspetti di rete, buona educazione ed efficienza,

ELABORARE

estrazione delle URL, distribuzione del carico,

Passo per passo

Seguiamo nel tempo il comportamento del crawler illustrato nell'introduzione:

PRELEVARE

aspetti di rete, buona educazione ed efficienza,

ELABORARE

estrazione delle URL, distribuzione del carico,

MEMORIZZARE

efficienza ed affidabilità.

Una questione di dimensione!

Possiamo distinguere i crawler in base all’ordine di grandezza del numero di pagine che intendiamo raccogliere con essi:

Una questione di dimensione!

Possiamo distinguere i crawler in base all'ordine di grandezza del numero di pagine che intendiamo raccogliere con essi:

SMALL SCALE (migliaia di pagine)

strutture dati semplici, uso della memoria centrale, ridotta necessità di ottimizzazione,

Una questione di dimensione!

Possiamo distinguere i crawler in base all'ordine di grandezza del numero di pagine che intendiamo raccogliere con essi:

SMALL SCALE (migliaia di pagine)

strutture dati semplici, uso della memoria centrale, ridotta necessità di ottimizzazione,

MEDIUM SCALE (milioni di pagine)

strutture dati sofisticate, uso memoria esterna, necessità di ottimizzazione e algoritmi efficienti,

Una questione di dimensione!

Possiamo distinguere i crawler in base all'ordine di grandezza del numero di pagine che intendiamo raccogliere con essi:

SMALL SCALE (migliaia di pagine)

strutture dati semplici, uso della memoria centrale, ridotta necessità di ottimizzazione,

MEDIUM SCALE (milioni di pagine)

strutture dati sofisticate, uso memoria esterna, necessità di ottimizzazione e algoritmi efficienti,

LARGE SCALE (miliardi di pagine)

strutture dati ed algoritmi molto sofisticati, grandi problemi di ingegnerizzazione e sistemistici.

Quale linguaggio?

In linea di principio, le ricette implementative presentate saranno indipendenti dalla scelta del linguaggio di programmazione.

Però. . .

Quale linguaggio?

In linea di principio, le ricette implementative presentate saranno indipendenti dalla scelta del linguaggio di programmazione.

Però. . .

Java sembra essere una buona scelta:

Quale linguaggio?

In linea di principio, le ricette implementative presentate saranno indipendenti dalla scelta del linguaggio di programmazione.

Però. . .

Java sembra essere una buona scelta:

- è particolarmente adatto ad applicazioni di rete,
- aiuta ad evitare errori difficili da scovare (*bound checking*),
- mette a disposizione un grande numero di librerie di utilità,
- consente di realizzare rapidamente prototipi,
- è portabile (dipende poco dal sistema operativo),
- è efficiente!

Programmare per il mondo reale

Prima di iniziare, un viatico dettato dall’esperienza:

Programmare per il mondo reale

Prima di iniziare, un viatico dettato dall'esperienza:

ROLLBACK

crash *inevitabili* richiedono un salvataggio continuo dei dati e la possibilità di ripartire dal momento del malfunzionamento,

Programmare per il mondo reale

Prima di iniziare, un viatico dettato dall'esperienza:

ROLLBACK

crash *inevitabili* richiedono un salvataggio continuo dei dati e la possibilità di ripartire dal momento del malfunzionamento,

RICONFIGURAZIONE A CALDO

il carico di CPU e banda deve poter essere modificato nel corso del crawl, così come altri parametri,

Programmare per il mondo reale

Prima di iniziare, un viatico dettato dall'esperienza:

Rollback

crash *inevitabili* richiedono un salvataggio continuo dei dati e la possibilità di ripartire dal momento del malfunzionamento,

Riconfigurazione a caldo

il carico di CPU e banda deve poter essere modificato nel corso del crawl, così come altri parametri,

Test

costruite da subito una infrastruttura per i test, che non richieda l'uso di banda "esterna".

Prelevare

Questioni generali

Quale che sia la dimensione, i principali problemi in fase di prelevamento sono:

Questioni generali

Quale che sia la dimensione, i principali problemi in fase di prelevamento sono:

QUALI PAGINE PRELEVARE, IN CHE ORDINE

per interesse/argomento, popolarità/qualità o posizione,

Questioni generali

Quale che sia la dimensione, i principali problemi in fase di prelevamento sono:

QUALI PAGINE PRELEVARE, IN CHE ORDINE

per interesse/argomento, popolarità/qualità o posizione,

COME RIPARTIRE IL CARICO

in maniera centalizzata, parzialmente o totalmente distribuita,

Questioni generali

Quale che sia la dimensione, i principali problemi in fase di prelevamento sono:

QUALI PAGINE PRELEVARE, IN CHE ORDINE

per interesse/argomento, popolarità/qualità o posizione,

COME RIPARTIRE IL CARICO

in maniera centalizzata, parzialmente o totalmente distribuita,

COME SFRUTTARE AL MEGLIO LE RISORSE

CPU e banda, propria e degli altri,

Questioni generali

Quale che sia la dimensione, i principali problemi in fase di prelevamento sono:

QUALI PAGINE PRELEVARE, IN CHE ORDINE

per interesse/argomento, popolarità/qualità o posizione,

COME RIPARTIRE IL CARICO

in maniera centalizzata, parzialmente o totalmente distribuita,

COME SFRUTTARE AL MEGLIO LE RISORSE

CPU e banda, propria e degli altri,

COME MANTENERE AGGIORNATE LE PAGINE

in modo uniforme, o adattativo.

come stimare il tasso di cambiamento?

Quali e in che ordine

SCEGLIERE LE PAGINE (caso elementare)

Quali e in che ordine

SCEGLIERE LE PAGINE (caso elementare)

- espressioni regolari basate su URL,
- tipo MIME [RFC 2046, IANA Media-Types].

Quali e in che ordine

SCEGLIERE LE PAGINE (caso elementare)

- espressioni regolari basate su URL,
- tipo MIME [RFC 2046, IANA Media-Types].

ORDINE DI VISITA

Quali e in che ordine

Scegliere le pagine (caso elementare)

- espressioni regolari basate su URL,
- tipo MIME [RFC 2046, IANA Media-Types].

Ordine di visita

- ampiezza: buona qualità, ma cattiva efficienza e *politeness*,
- profondità: scarsa qualità, ma meno problemi,
- strategie miste, visite per priorità.

Parallelismo

La latenza della comunicazione impone l’uso del parallelismo.

Parallelismo

La latenza della comunicazione impone l’uso del parallelismo.

Il parallelismo introduce vari problemi di sincronizzazione, in particolare riguardo alla suddivisione del carico, che influiscono sulle possibili modalità della parallelizzazione stessa.

Parallelismo

La latenza della comunicazione impone l'uso del parallelismo.

Il parallelismo introduce vari problemi di sincronizzazione, in particolare riguardo alla suddivisione del carico, che influiscono sulle possibili modalità della parallelizzazione stessa.

Alcuni concetti base:

- *overlap* vs. *coverage*,
- costo di intercomunicazione,
- scalabilità,
- *fault-tolerance*.

Parallelismo

La latenza della comunicazione impone l'uso del parallelismo.

Il parallelismo introduce vari problemi di sincronizzazione, in particolare riguardo alla suddivisione del carico, che influiscono sulle possibili modalità della parallelizzazione stessa.

Alcuni concetti base:

- *overlap* vs. *coverage*,
- costo di intercomunicazione,
- scalabilità,
- *fault-tolerance*.

Suggerimento: ordine ibrido (ampiezza inter-host, profondità in-host) con un thread per host.

Buona educazione

PRESENTARSI!

usare l'header `User-Agent` dell'HTTP [W3C],

Buona educazione

PRESENTARSI!

usare l'header `User-Agent` dell'HTTP [W3C],

RISPETTARE LA CPU E BANDA DEGLI ALTRI

una pagina per volta per sito (e i virtualhost?), regolare la banda a seconda delle ore del giorno,

Buona educazione

Presentarsi!

usare l'header `User-Agent` dell'HTTP [W3C],

Rispettare la CPU e banda degli altri

una pagina per volta per sito (e i virtualhost?), regolare la banda a seconda delle ore del giorno,

Rispettare le scelte degli altri

rispettare il *Robot Exclusion Standard* [www.robotstxt.org], essere pronti a escludere manualmente pagine/siti,

Buona educazione

Presentarsi!

usare l'header `User-Agent` dell'HTTP [W3C],

Rispettare la CPU e banda degli altri

una pagina per volta per sito (e i virtualhost?), regolare la banda a seconda delle ore del giorno,

Rispettare le scelte degli altri

rispettare il *Robot Exclusion Standard* [www.robotstxt.org], essere pronti a escludere manualmente pagine/siti,

Rispettare le propria banda

adattare il carico del crawl al carico di rete, essere responsivi alle richieste di chi gestisce l'infrastruttura.

La frontiera

In ogni caso, osserviamo che è necessario tenere traccia sia delle pagine già visitate che dell'insieme di pagine da visitare (la *frontiera*, nel gergo degli algoritmi di visita dei grafi).

La frontiera

In ogni caso, osserviamo che è necessario tenere traccia sia delle pagine già visitate che dell'insieme di pagine da visitare (la *frontiera*, nel gergo degli algoritmi di visita dei grafi).

Vedremo nella sezione sulla memorizzazione una possibile soluzione per il problema di decidere se una pagina sia stata visitata o meno.

La frontiera

In ogni caso, osserviamo che è necessario tenere traccia sia delle pagine già visitate che dell'insieme di pagine da visitare (la *frontiera*, nel gergo degli algoritmi di visita dei grafi).

Vedremo nella sezione sulla memorizzazione una possibile soluzione per il problema di decidere se una pagina sia stata visitata o meno.

L'istanza più generale della frontiera può essere rappresentata con una *coda di priorità* (di cui code e pile sono casi particolari), ma con qualche ulteriore accorgimento. . .

La frontiera (continua)

Dato il numero medio di URL per pagina può avere grande dimensione, tale da richiedere l'uso della memoria esterna.

La frontiera (continua)

Dato il numero medio di URL per pagina può avere grande dimensione, tale da richiedere l'uso della memoria esterna.

Oltre alle primitive di inserimento ed estrazione dalla coda possono essere necessarie primitive per l'implementazione del "parallelismo" e della "buona educazione".

La frontiera (continua)

Dato il numero medio di URL per pagina può avere grande dimensione, tale da richiedere l'uso della memoria esterna.

Oltre alle primitive di inserimento ed estrazione dalla coda possono essere necessarie primitive per l'implementazione del "parallelismo" e della "buona educazione".

Suggerimento: nel caso *small scale* la frontiera può essere memorizzata come una collezione di liste (una per host), in memoria centrale.

Elaborare

Estrarre le URL

Da pagine HTML

Estrarre le URL

Da pagine HTML

- parsing completo (*context-free*):
  - più preciso se il documento è valido
    ma raramente lo sono!,
  - meno efficiente (per uso di memoria e CPU),
  - consente di raccogliere maggiori informazioni,
  - esiste *free software* che lo implementa.

Estrarre le URL

Da pagine HTML

- parsing completo (*context-free*):
  - più preciso se il documento è valido
    ma raramente lo sono!,
  - meno efficiente (per uso di memoria e CPU),
  - consente di raccogliere maggiori informazioni,
  - esiste *free software* che lo implementa.
- parsing elementare (basato su espressioni regolari):
  - meno preciso,
    ma funziona anche per documenti non validi,
  - è basato su euristiche, non banale ottenerne di corrette,
  - molto efficiente.

Estrarre le URL (continua)

Da pagine di altro formato

Estrarre le URL (continua)

Da pagine di altro formato

- parsing “diretto”:
  - dipendente dal formato,
  - manca software che lo implementi, anche per i formati più standard (PostScript e Portable Document Format).

Estrarre le URL (continua)

Da pagine di altro formato

- parsing “diretto”:
  - dipendente dal formato,
  - manca software che lo implementi, anche per i formati più standard (PostScript e Portable Document Format).
- trasformazione in testo, seguita da parsing elementare:
  - esiste *free software* per la trasformazione,
  - il parsing può essere derivato da quello per l’HTML,
  - il parsing elementare del testo è comunque necessario.

Estrarre le URL (continua)

Da pagine di altro formato

- parsing “diretto”:
  - dipendente dal formato,
  - manca software che lo implementi, anche per i formati più standard (PostScript e Portable Document Format).
- trasformazione in testo, seguita da parsing elementare:
  - esiste *free software* per la trasformazione,
  - il parsing può essere derivato da quello per l’HTML,
  - il parsing elementare del testo è comunque necessario.

Alcune informazioni dalla pagina del corso “Text Information Retrieval, Mining, and Exploitation” [Stanford, CS276B].

URL. . . uniformi?

Quale che sia il processo di estrazione delle URL dalla pagina, in genere è necessario manipolarle ulteriormente.

URL. . . uniformi?

Quale che sia il processo di estrazione delle URL dalla pagina, in genere è necessario manipolarle ulteriormente.

In particolare, sono necessarie almeno le seguenti operazioni:

URL. . . uniformi?

Quale che sia il processo di estrazione delle URL dalla pagina, in genere è necessario manipolarle ulteriormente.

In particolare, sono necessarie almeno le seguenti operazioni:

- de-relativizzazione (a partire dall'URL della pagina da cui sono state estratte, o dell'elemento BASE dell'HTML [HTML, RFC 1808]),

URL. . . uniformi?

Quale che sia il processo di estrazione delle URL dalla pagina, in genere è necessario manipolarle ulteriormente.

In particolare, sono necessarie almeno le seguenti operazioni:

- de-relativizzazione (a partire dall'URL della pagina da cui sono state estratte, o dell'elemento BASE dell'HTML [HTML, RFC 1808]),
- normalizzazione [RFC 1738]

URL. . . uniformi?

Quale che sia il processo di estrazione delle URL dalla pagina, in genere è necessario manipolarle ulteriormente.

In particolare, sono necessarie almeno le seguenti operazioni:

- de-relativizzazione (a partire dall'URL della pagina da cui sono state estratte, o dell'elemento BASE dell'HTML [HTML, RFC 1808]),
- normalizzazione [RFC 1738]
- soluzione delle redirezioni (per effetto dell'header `Location` dell'HTTP [W3C],

URL. . . uniformi?

Quale che sia il processo di estrazione delle URL dalla pagina, in genere è necessario manipolarle ulteriormente.

In particolare, sono necessarie almeno le seguenti operazioni:

- de-relativizzazione (a partire dall'URL della pagina da cui sono state estratte, o dell'elemento BASE dell'HTML [HTML, RFC 1808]),
- normalizzazione [RFC 1738]
- soluzione delle redirezioni (per effetto dell'header `Location` dell'HTTP [W3C],

Osservate inoltre che per memorizzare le URL nelle varie strutture dati facendo uso di una quantità costante di memoria può essere comodo calcolarne una qualche funzione di hash.

Analisi durante il crawl

Spesso le finalità del crawl sono raccogliere pagine per effettuarvi, in seguito, analisi di vario tipo.

Analisi durante il crawl

Spesso le finalità del crawl sono raccogliere pagine per effettuarvi, in seguito, analisi di vario tipo.

Il crawl è una attività di tipo *I/O bound* pertanto può essere sensato sfruttare la CPU durante il crawl per precomputare alcune di tali analisi, immagazzinandone i risultati assieme alle pagine stesse.

Analisi durante il crawl

Spesso le finalità del crawl sono raccogliere pagine per effettuarvi, in seguito, analisi di vario tipo.

Il crawl è una attività di tipo *I/O bound* pertanto può essere sensato sfruttare la CPU durante il crawl per precomputare alcune di tali analisi, immagazzinandone i risultati assieme alle pagine stesse.

Questo è particolarmente sensato quando le analisi che verranno fatte nelle fasi successive sono le stesse che sono necessarie nella fase di crawl, come, ad esempio, il parsing.

Intermezzo: Consistent Hashing

Parallelismo in UbiCrawler

Vediamo una possibile implementazione del parallelismo adottata da UbiCrawler, un *medium scale crawler* usato per progetti di ricerca.

Parallelismo in UbiCrawler

Vediamo una possibile implementazione del parallelismo adottata da UbiCrawler, un *medium scale crawler* usato per progetti di ricerca.

Consideriamo un certo insieme C di *compiti* e un insieme A di possibili *agenti* che devono svolgerli. Poiché intendiamo tener conto di malfunzionamenti, considereremo in particolare gli agenti *vivi*, che sono un sottoinsieme $V \subset A$ di tutti gli agenti possibili.

Parallelismo in UbiCrawler

Vediamo una possibile implementazione del parallelismo adottata da UbiCrawler, un *medium scale crawler* usato per progetti di ricerca.

Consideriamo un certo insieme C di *compiti* e un insieme A di possibili *agenti* che devono svolgerli. Poiché intendiamo tener conto di malfunzionamenti, considereremo in particolare gli agenti *vivi*, che sono un sottoinsieme $V \subset A$ di tutti gli agenti possibili.

Cerchiamo una *funzione di assegnamento*:

$$f : \mathcal{P}(A) \times C \to A$$

che soddisfi alcune proprietà...

La funzione di assegnamento

L'assegnamento $f_V(c) = f(V, c)$ dipende dagli agenti vivi e deve garantire:

La funzione di assegnamento

L'assegnamento $f_V(c) = f(V, c)$ dipende dagli agenti vivi e deve garantire:

- correttezza

$$f_V(c) = a \in V$$

La funzione di assegnamento

L'assegnamento $f_V(c) = f(V, c)$ dipende dagli agenti vivi e deve garantire:

- correttezza

$$f_V(c) = a \in V$$

- bilanciamento

$$f_V^{-1}(a) \approx |C|/|V|$$

La funzione di assegnamento

L'assegnamento $f_V(c) = f(V, c)$ dipende dagli agenti vivi e deve garantire:

- correttezza

$$f_V(c) = a \in V$$

- bilanciamento

$$f_V^{-1}(a) \approx |C|/|V|$$

- monotonicità

$$f_V(c) \in V' \subset V \Rightarrow f_{V'}(c) = f_V(c).$$

La funzione di assegnamento

L'assegnamento $f_V(c) = f(V, c)$ dipende dagli agenti vivi e deve garantire:

- correttezza

$$f_V(c) = a \in V$$

- bilanciamento

$$f_V^{-1}(a) \approx |C|/|V|$$

- monotonicità

$$f_V(c) \in V' \subset V \Rightarrow f_{V'}(c) = f_V(c).$$

Le prime due proprietà possono essere ottenute semplicemente con una funzione tipo "hash modulo $|V|$".

La funzione di assegnamento

L'assegnamento $f_V(c) = f(V, c)$ dipende dagli agenti vivi e deve garantire:

- correttezza

$$f_V(c) = a \in V$$

- bilanciamento

$$f_V^{-1}(a) \approx |C|/|V|$$

- monotonicità

$$f_V(c) \in V' \subset V \Rightarrow f_{V'}(c) = f_V(c).$$

Le prime due proprietà possono essere ottenute semplicemente con una funzione tipo "hash modulo $|V|$".

Ma una soluzione così banale non garantirebbe la monotonicità se la cardinalità di V cambiasse!

Consistent Hashing [Leighton *et al.*, ’97]

Consideriamo due mappe da A e C al cerchio unitario.

Consistent Hashing [Leighton *et al.*, '97]

Consideriamo due mappe da A e C al cerchio unitario.

Usiamo la prima mappa per far corrispondere ad ogni agente $a \in V$ un certo numero di punti sul cerchio (detti *repliche*).

Consistent Hashing [Leighton *et al.*, '97]

Consideriamo due mappe da A e C al cerchio unitario.

Usiamo la prima mappa per far corrispondere ad ogni agente $a \in V$ un certo numero di punti sul cerchio (detti *repliche*).

Usiamo quindi la seconda mappa per far corrispondere a $c \in C$ un solo punto del cerchio.

Consistent Hashing [Leighton *et al.*, '97]

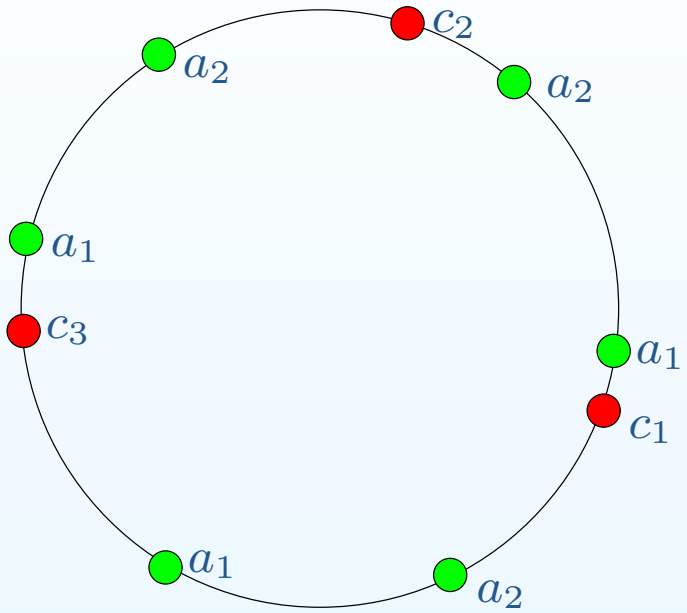
Consideriamo due mappe da A e C al cerchio unitario.

Usiamo la prima mappa per far corrispondere ad ogni agente $a \in V$ un certo numero di punti sul cerchio (detti *repliche*).

Usiamo quindi la seconda mappa per far corrispondere a $c \in C$ un solo punto del cerchio.

Definiamo $a = f_V(c)$ come l'agente a "più vicino" a c sul cerchio.

Esempio



$V = \{a_1, a_2\}$ (3 repliche)

$f_V^{-1}(a_1) = \{c_1, c_3\}$ e $f_V^{-1}(a_2) = \{c_2\}$

Osservazioni

- Si può provare che, sotto opportune ipotesi, la funzione definite come sopra soddisfano: correttezza, bilanciamento e monotonicità,

Osservazioni

- Si può provare che, sotto opportune ipotesi, la funzione definite come sopra soddisfano: correttezza, bilanciamento e monotonicità,
- le mappe possono essere realizzate con funzioni di hash e randomizzazione,

  (usando opportuni accorgimenti per rendere deterministico l’insieme di repliche),

Osservazioni

- Si può provare che, sotto opportune ipotesi, la funzione definite come sopra soddisfano: correttezza, bilanciamento e monotonicità,
- le mappe possono essere realizzate con funzioni di hash e randomizzazione,

  (usando opportuni accorgimenti per rendere deterministico l'insieme di repliche),
- Il calcolo di $f_V(a)$ può essere implementato con opportune strutture dati in modo molto efficiente, richiedendo una minima quantità di comunicazione tra gli agenti vivi.

Osservazioni

- Si può provare che, sotto opportune ipotesi, la funzione definite come sopra soddisfano: correttezza, bilanciamento e monotonicità,
- le mappe possono essere realizzate con funzioni di hash e randomizzazione,

  (usando opportuni accorgimenti per rendere deterministico l'insieme di repliche),
- Il calcolo di $f_V(a)$ può essere implementato con opportune strutture dati in modo molto efficiente, richiedendo una minima quantità di comunicazione tra gli agenti vivi.

Suggerimento: se la cardinalità di V non cambia (ossia se si trascura la tolleranza ai malfunzionamenti), la soluzione basata su hash e modulo è molto semplice da realizzare!

Immagazzinare

Modalità di accesso

La memorizzazione e successiva lettura delle pagine scaricate avviene usualmente secondo due modalità:

Modalità di accesso

La memorizzazione e successiva lettura delle pagine scaricate avviene usualmente secondo due modalità:

INCREMENTALE (append)

durante il crawl, le pagine vengono semplicemente aggiunte a quelle memorizzate in precedenza

se non è un crawl di “rinfresco”, non avvengono accessi casuali, o cancellazioni

Modalità di accesso

La memorizzazione e successiva lettura delle pagine scaricate avviene usualmente secondo due modalità:

INCREMENTALE (append)

durante il crawl, le pagine vengono semplicemente aggiunte a quelle memorizzate in precedenza

se non è un crawl di “rinfresco”, non avvengono accessi casuali, o cancellazioni

A BLOCCHI (batch)

dopo il crawl, i moduli per l’analisi dei dati (indicizzazione e altro) accedono solitamente a grosse porzioni dei dati per volta (per ottimizzare l’I/O).

Due possibilità

Viste le modalità di accesso, le soluzioni più affermate sono due:

Due possibilità

Viste le modalità di accesso, le soluzioni più affermate sono due:

INDEXED-LOGFILE

un file sequenziale a cui le pagine vengono accodate più un indice in memoria centrale degli offset di inizio pagina,

usato anche per tener traccia delle pagine sono state già prelevate

Due possibilità

Viste le modalità di accesso, le soluzioni più affermate sono due:

INDEXED-LOGFILE

un file sequenziale a cui le pagine vengono accodate più un indice in memoria centrale degli offset di inizio pagina,

usato anche per tener traccia delle pagine sono state già prelevate

facile da implementare, resistente ai malfunzionamenti, ma meno versatile in fase di analisi,

Due possibilità

Viste le modalità di accesso, le soluzioni più affermate sono due:

INDEXED-LOGFILE

un file sequenziale a cui le pagine vengono accodate più un indice in memoria centrale degli offset di inizio pagina,

usato anche per tener traccia delle pagine sono state già prelevate

facile da implementare, resistente ai malfunzionamenti, ma meno versatile in fase di analisi,

DATABASE

sia esterno (basato su un DMBS SQL), oppure *embedded*,

Due possibilità

Viste le modalità di accesso, le soluzioni più affermate sono due:

INDEXED-LOGFILE

un file sequenziale a cui le pagine vengono accodate più un indice in memoria centrale degli offset di inizio pagina,

usato anche per tener traccia delle pagine sono state già prelevate

facile da implementare, resistente ai malfunzionamenti, ma meno versatile in fase di analisi,

DATABASE

sia esterno (basato su un DMBS SQL), oppure *embedded*,

in generale difficile da implementare e poco efficiente, ma molto versatile per le fasi successive.

Dettagli implementativi

Suggerimenti su Java

- Attenzione alle prestazioni:

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),
  - configurate bene il *garbage collector* [doc]

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),
  - configurate bene il *garbage collector* [doc]
  - usate *object* e soprattutto *thread pooling* [doc],

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),
  - configurate bene il *garbage collector* [doc]
  - usate *object* e soprattutto *thread pooling* [doc],

- usate le libreria adatte ai vari scopi:

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),
  - configurate bene il *garbage collector* [doc]
  - usate *object* e soprattutto *thread pooling* [doc],

- usate le libreria adatte ai vari scopi:
  - RMI per la comunicazione intreprocesso [doc],

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),
  - configurate bene il *garbage collector* [doc]
  - usate *object* e soprattutto *thread pooling* [doc],

- usate le libreria adatte ai vari scopi:
  - RMI per la comunicazione intreprocesso [doc],
  - le nuove API per l'I/O (`java.nio`) [doc],

Suggerimenti su Java

- Attenzione alle prestazioni:
  - dimensionate bene la memoria (heap, stack. . .),
  - configurate bene il *garbage collector* [doc]
  - usate *object* e soprattutto *thread pooling* [doc],
- usate le libreria adatte ai vari scopi:
  - RMI per la comunicazione intreprocesso [doc],
  - le nuove API per l'I/O (`java.nio`) [doc],
- Usate Ant come *build tool* [Apache Ant Project],

Librerie

Ci sono molte librerie rilasciate come free software che possono essere utilizzate sia a “scatola chiusa” (evitandovi, più o meno leggittimamente, di implementare certe parti del codice) che a “scatola aperta”, fornendo guida ed ispirazione nella soluzione di problemi difficili.

Librerie

Ci sono molte librerie rilasciate come free software che possono essere utilizzate sia a “scatola chiusa” (evitandovi, più o meno leggittimamente, di implementare certe parti del codice) che a “scatola aperta”, fornendo guida ed ispirazione nella soluzione di problemi difficili.

Quella che segue è una minima antologia delle librerie più interessanti in relazione alla costruzione di un crawler (e di un motore di ricerca, più in generale):

Librerie

Ci sono molte librerie rilasciate come free software che possono essere utilizzate sia a “scatola chiusa” (evitandovi, più o meno leggittimamente, di implementare certe parti del codice) che a “scatola aperta”, fornendo guida ed ispirazione nella soluzione di problemi difficili.

Quella che segue è una minima antologia delle librerie più interessanti in relazione alla costruzione di un crawler (e di un motore di ricerca, più in generale):

- Colt,
- Fastutil
- WebGraph,
- Managing Gigabytes for Java (MG4J).

Fastutil

http://fastutil.dsi.unimi.it

Fastutil provides type-specific maps, sets and lists with a small memory footprint and much faster access and insertion.
The classes implement their standard counterpart interface (e.g., Map for maps) and can be plugged into existing code.
Moreover, they provide additional features (such as bidirectional iterators) that are not available in the standard classes.
Besides objects and primitive types, fastutil classes provide support for references, that is, objects that are compared using the equality operator rather than the equals() method.

Colt

http://hoschek.home.cern.ch/hoschek/colt/

This distribution provides an infrastructure for scalable scientific and technical computing in Java.
It contains, among others, efficient and usable data structures and algorithms for Off-line and On-line Data Analysis, Linear Algebra, Multi-dimensional arrays, Statistics, Histogramming, Monte Carlo Simulation, Parallel and Concurrent Programming.
It summons some of the best concepts, designs and implementations thought up over time by the community, ports or improves them and introduces new approaches where need arises.

WebGraph

http://webgraph.dsi.unimi.it/

WebGraph is a framework to study the web graph. It provides simple ways to manage very large graphs, exploiting modern compression techniques.
With WebGraph you can access and analyse a very large web graph, even on a PC with as little as 256 Mbytes of RAM. Using WebGraph is as easy as installing a few jar files and downloading a data set. This makes studying phenomena such as PageRank, distribution of graph properties of the web graph, etc. very easy.

MG4J

http://mg4j.dsi.unimi.it/

MG4J is a collaborative effort aimed at providing a free Java implementation of inverted-index compression techniques; as a by-product, it offers several general-purpose optimised classes, including fast and compact mutable strings, bit-level I/O, fast unsychronised buffered streams, (possibly signed) minimal perfect hashing, etc.
MG4J tries to make the techniques described in the book Managing Gigabytes, by Ian Witten, Alistair Moffat and Timothy Bell, accessible without having to deal with bit-level operations in a clean, object-oriented environment.
At this point, MG4J lets you build compressed full-text indices for large collections of documents using sophisticated techniques such as interpolative coding. Moreover, it provides utility classes that are essential in any serious text-processing activity.

Articoli interessanti

In generale, sui motori di ricerca, una bella rassegna:

Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, Sriram Raghavan. “Searching the Web”. *ACM Transactions on Internet Technology*, 1(1): August 2001.
[preprint]

Articoli interessanti

In generale, sui motori di ricerca, una bella rassegna:

Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, Sriram Raghavan. “Searching the Web”. *ACM Transactions on Internet Technology*, 1(1): August 2001. [preprint]

Riguardo al solo aspetto del crawling:

Junghoo Cho and Hector Garcia-Molina “Parallel Crawlers”. *In Proceedings of the 11th World Wide Web conference (WWW11)*, Honolulu, Hawaii, May 2002. [preprint]

Marc Najork and Allan Heydon. “High-Performance Web Crawling”. *Compaq SRC Research Report 173*, Report #173, September 26, 2001. [preprint]