Università degli Studi di Catania
Dipartimento di Matematica e Informatica
Corso di laurea in Informatica Magistrale
Compilatori A.A. 2014-15

Spiderino: web crawler

Abstract

Un crawler (detto anche web crawler, spider o robot), è un software che analizza i contenuti di una rete (o di un database) in un modo metodico e automatizzato, in genere per conto di un motore di ricerca.
Un Web crawler si basa su una lista di URL da visitare fornita dal motore di ricerca (il quale, inizialmente, si basa sugli indirizzi suggeriti dagli utenti o su una lista precompilata dai programmatori stessi).
Durante l'analisi di un URL, identifica tutti gli hyperlink presenti nel documento e li aggiunge alla lista di URLs da visitare. Il processo può essere concluso manualmente, dopo che un determinato numero di collegamenti è stato seguito oppure dopo un determinato intervallo di tempo.
Spiderino è un web crawler scritto in PHP nato da un progetto universitario all' Università di Catania.

Introduzione

Lo scopo del nostro progetto è stato quello di realizzare un crawler per l' analisi di URLs, a partire da seeds iniziali, memorizzando solamente le pagine contenenti almeno una delle keywords date in input dall'utente.
Per lo sviluppo del software è stato utilizzato il linguaggio PHP per l'analisi degli URLs e un database mySQL per la loro memorizzazione.
Il codice sorgente è rilasciato sotto una licenza GNU GPL v3 ed è disponibile su GitHub all'indirizzo https://github.com/filirnd/spiderino

Funzionamento

Il software prende in input :

- uno o più seeds iniziali,
- un tempo di esecuzione espresso in minuti,
- una o piu keyWords,

come mostrato di seguito:

`php ./spiderino SEED1 [SEED2 SEED3 .. ] -t TIME_SIM KEYWORD1 [KEYWORD2 KEYWORD3 .. ]`

Nella fase iniziale viene inizializzata all' interno del database una tabella contenente i seguenti campi:

- **id** (identificatore dell' URL salvato)
- **filename** (nome della pagina salvata su disco in caso di KeyWord presente)
- **url** (nome dell'URL trovato durante l'analisi)
- **father** (nome dell'URL padre da cui è stato trovato)
- **depth** (profondità dell'URL nell'albero di analisi)
- **discoveredurls** (numero di URLs trovati nella pagina)

I campi filename e discoveredurls assumono valore -1 nel caso in cui l' URL è rimasto in coda senza essere stata analizzato, mentre assume valore 0 se non viene trovata nessuna KeyWords all' interno della pagina.

Ogni URL trovato a partire dai seeds iniziali viene inserito all' interno di questo database, creato automaticamente, se non esiste già, all'avvio dello spider.

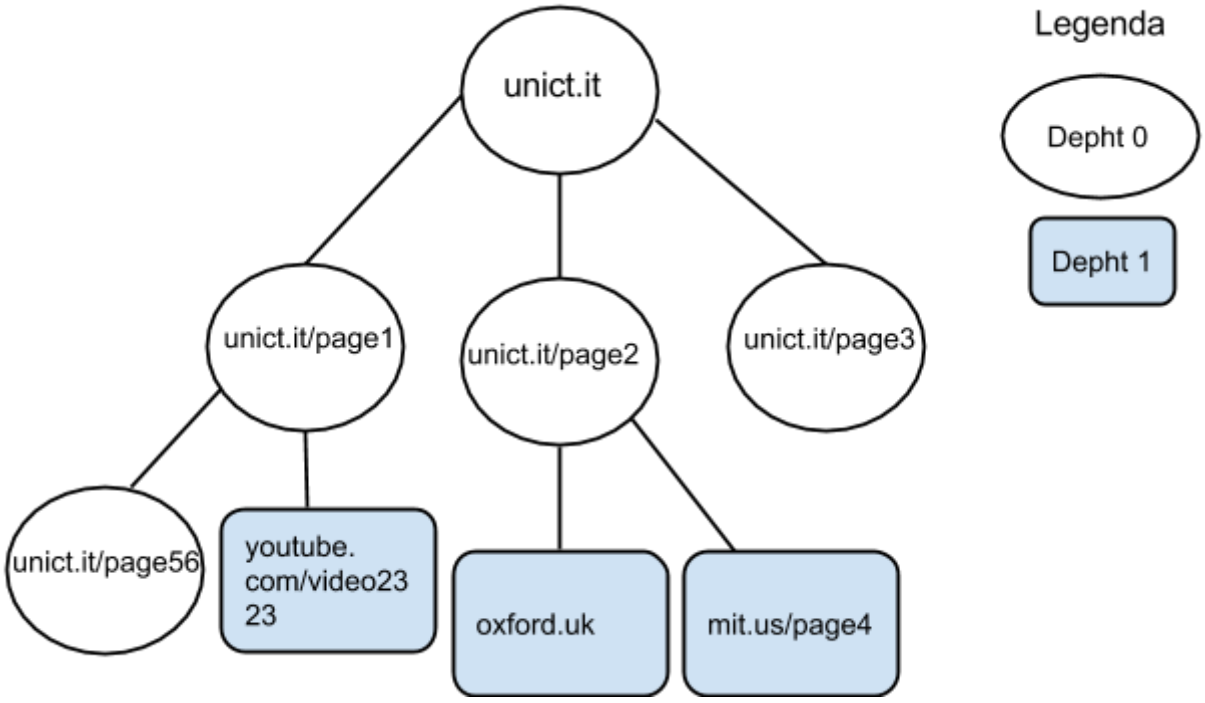
Per ogni ricerca viene creata in automatico una tabella apposita il cui nome corrisponde alla data di inizio della simulazione. In questo modo è possibile effettuare diverse simulazioni senza dover stare attenti a svuotare la tabella dei risultati.

Nei dati di input è necessario che ci siano sempre almeno un URL di partenza, un tempo di simulazione espresso in minuti e almeno una parola chiave. Se manca uno di questi prerequesiti, viene mostrato un errore.
Tutti gli URLs passati come seeds vengono subito controllati (per verificare la loro validità sintattica) e quindi inseriti all'interno del database.
Inoltre viene creata una cartella, con lo stesso nome della tabella all'interno del database, dove verranno memorizzate le pagine che contengono almeno una delle parole chiave date in input.
A questo punto tramite un ciclo while, si iniziano ad analizzare tutti gli URLs inseriti nel database. Questo verrà fatto finchè ci saranno URLs da analizzare oppure finchè non si esaurisce il tempo di simulazione.

La funzione principale dello spiderino è la **readUrls($siteUrl)**.
Questa funzione si occupa di analizzare ogni URL per trovare nuovi URLs da inserire nel database e salvare tutte quelle pagine che contengono almeno una delle parole chiave.

La pagina viene subito scaricata e tramite una regex vengono estratti tutti gli URLs validi. Dall'URL padre vengono estratti il dominio principale e il dominio fino alla cartella attuale, in modo tale da poter concatenarli ai successivi URLs relativi trovati nella pagina.

Per ogni URL trovato, questo viene pulito, portato in una forma standard (simile a http://sito.com/cartella/file.html) e aggiunto al database se ritenuto valido. La sua profondità sarà uguale a quella del padre, cioè la pagina dove è stato trovato, aumentata di uno. Si creerà così un albero n-ario nella quale i figli di un nodo rappresentano tutti gli URLs trovati in quella pagina.



Una volta analizzati gli URLs e inseriti nel database, la fase successiva è quella di cercare nella pagina attuale le parole chiave date in input.
Questo viene fatto tramite una semplice regex per ognuna delle parole chiave da ricercare. Se ne viene trovata una, la ricerca viene subito fermata e la pagina viene salvata in un file, dentro

la cartella di output, con nome uguale ad un indice progressivo che verrà inserito anche nel database nella riga corrispondente all'URL della pagina appena analizzata.

Finito anche questo processo, vengono aggiornate le informazioni nel database, inserendo numero di URLs trovati in quella pagina ed eventuale ID del file salvato in memoria.

Esempio concreto

Per testare spiderino sono state eseguite delle simulazioni utilizzando i seguenti seeds:

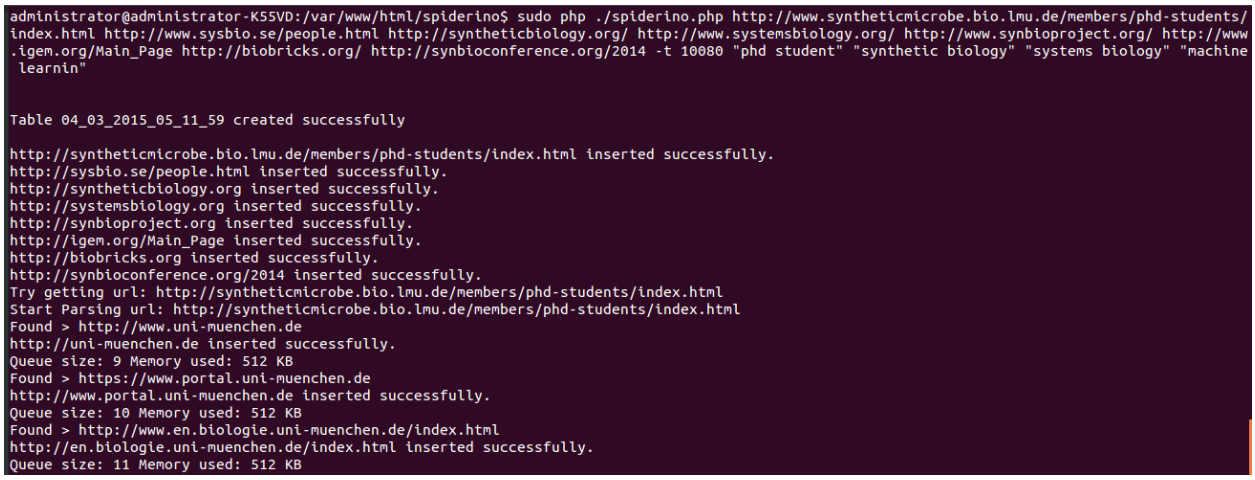
- http://www.syntheticmicrobe.bio.lmu.de/members/phd-students/index.html
- http://www.sysbio.se/people.html
- http://syntheticbiology.org/
- http://www.systemsbiology.org/
- http://www.synbioproject.org/
- http://www.igem.org/Main_Page
- http://biobricks.org/
- http://synbioconference.org/2014

Su questi particolari seeds sono state ricercate 4 parole:

1. "phd student"
2. "synthetic biology"
3. "systems biology"
4. "machine learning"

Lo spider è statodunque avviato nel seguente modo:

sudo php ./spiderino.php http://www.syntheticmicrobe.bio.lmu.de/members/phd-students/index.html http://www.sysbio.se/people.html http://syntheticbiology.org/ http://www.systemsbiology.org/ http://www.synbioproject.org/ http://www.igem.org/Main_Page http://biobricks.org/ http://synbioconference.org/2014 -t 10080 "phd student" "synthetic biology" "systems biology" "machine learning"



Statistiche

Sono state effettuate tre simulazioni da 1 ora, 24 ore e 7 giorni.
I risultati ottenuti per ogni singola simulazione sono stati ricavati dalle seguenti operazioni:

1. L'output ottenuto dall'esecuzione del crawler spiderino è stato passato al MailParser, che ha permesso di estrarre tutte le emails valide presenti nei diversi file.txt, che corrispondono ai vari URL analizzati nella quale è presente almeno una delle parole chiavi, e sono state inserite in un database;

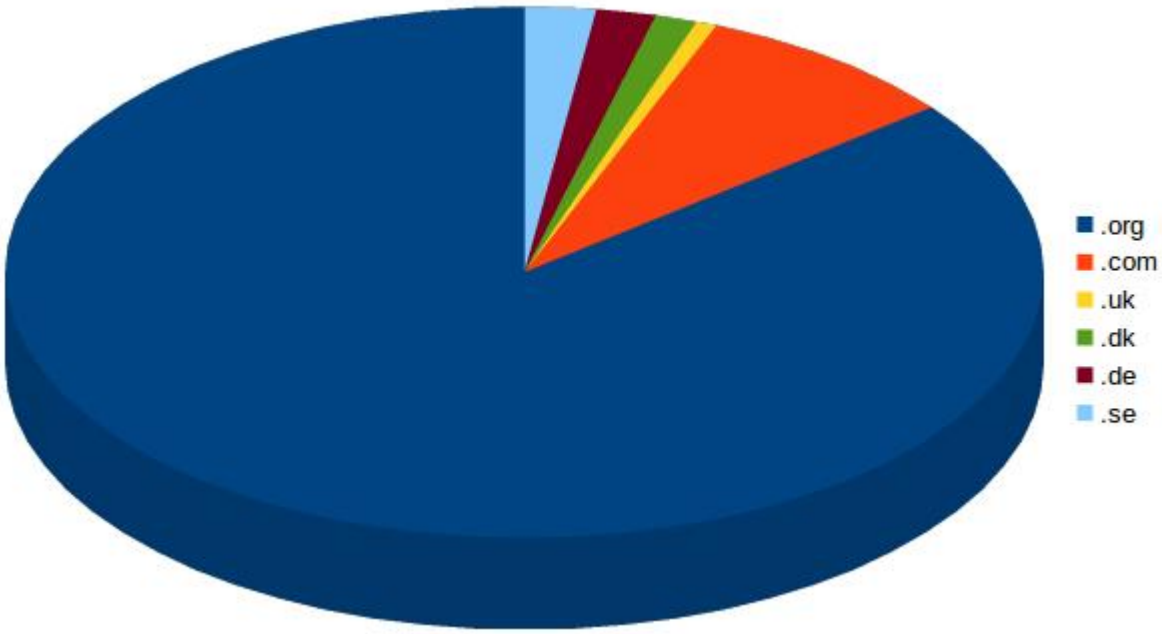
2. Sono state poi effettuate le connessioni sia al database contenenti tutte le emails e sia al database creato dal crawler spiderino. In questo modo, tramite delle query, sono state fatte le statistiche relative alla profondità massima (degli URLs trovati ed URLs analizzati), il numero di URLs trovati ed analizzati per ogni dominio di 1°livello, e il numero di emails trovate per ogni URL analizzato.

id	filename	url	father	depth	discoveredurls
0	1	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	Initial seed Url	0	10
1	2	http://sysbio.se/people.html	Initial seed Url	0	92
2	3	http://syntheticbiology.org	Initial seed Url	0	68
3	4	http://systemsbiology.org	Initial seed Url	0	52
4	5	http://synbioproject.org	Initial seed Url	0	9
5	6	http://igem.org/Main_Page	Initial seed Url	0	49
6	7	http://biobricks.org	Initial seed Url	0	54
7	8	http://synbioconference.org/2014	Initial seed Url	0	25
8	-1	http://uni-muenchen.de	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	-1
9	0	http://www.portal.uni-muenchen.de	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
10	0	http://en.biologie.uni-muenchen.de/index.html	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	16
11	9	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
12	10	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
13	11	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
14	12	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
15	13	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
16	14	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	0
17	15	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	http://syntheticmicrobe.bio.lmu.de/members/phd-stu...	1	3
18	-1	http://sysbio.se/BioMet	http://sysbio.se/people.html	1	-1

Simulazione 1 ora

I risultati ottenuti dalla simulazione di 1 ora sono riportati nei file *statistiche_email_1h.php* e *statistiche_url_1h.php.*

I domini più comuni sono stati:

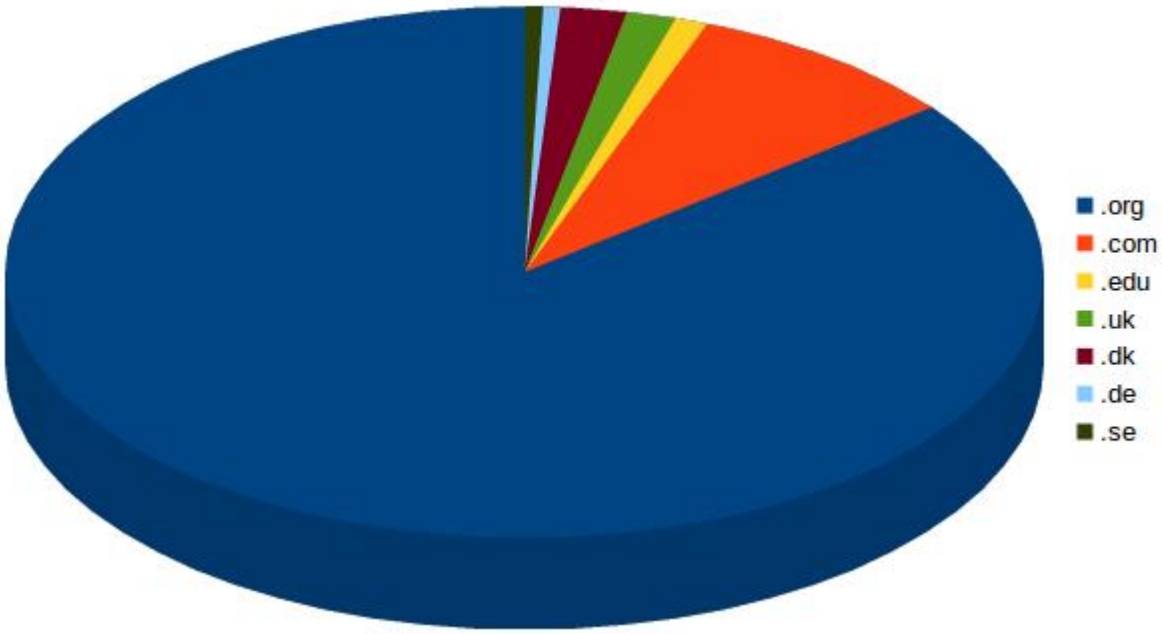


Sono state trovate in totale 56 emails.

Simulazione 24 ore

I risultati ottenuti dalla simulazione di 24 ore sono riportati nei file *statistiche_email_24h.php* e *statistiche_url_24h.php.*

I domini più comuni sono stati:

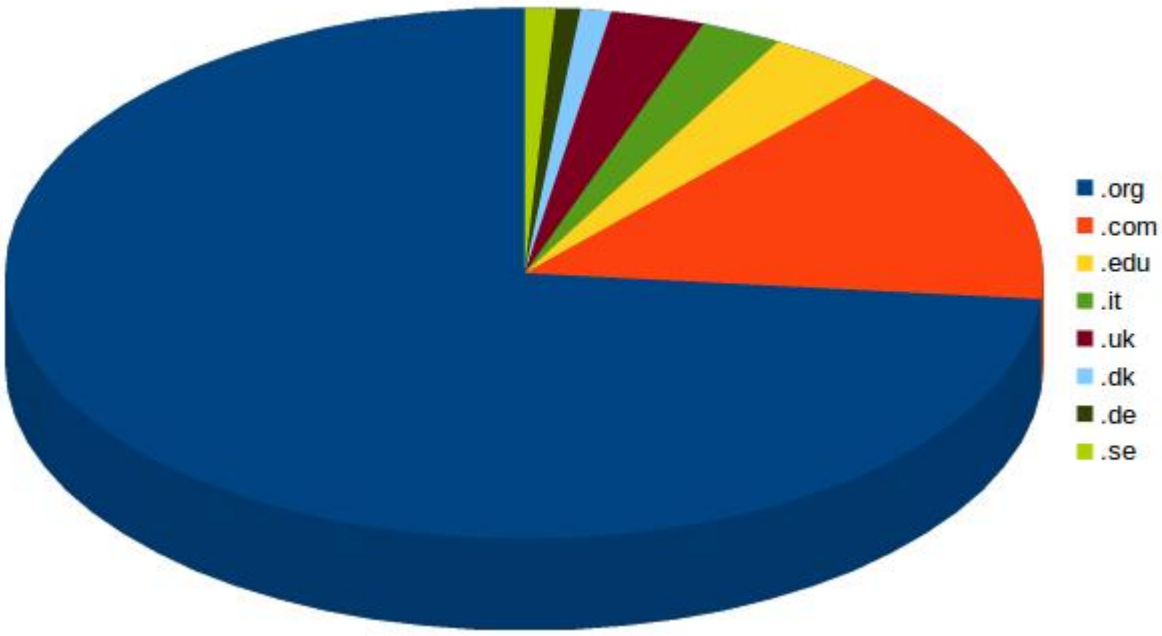


Sono state trovate in totale 56 emails.

Simulazione 7 giorni

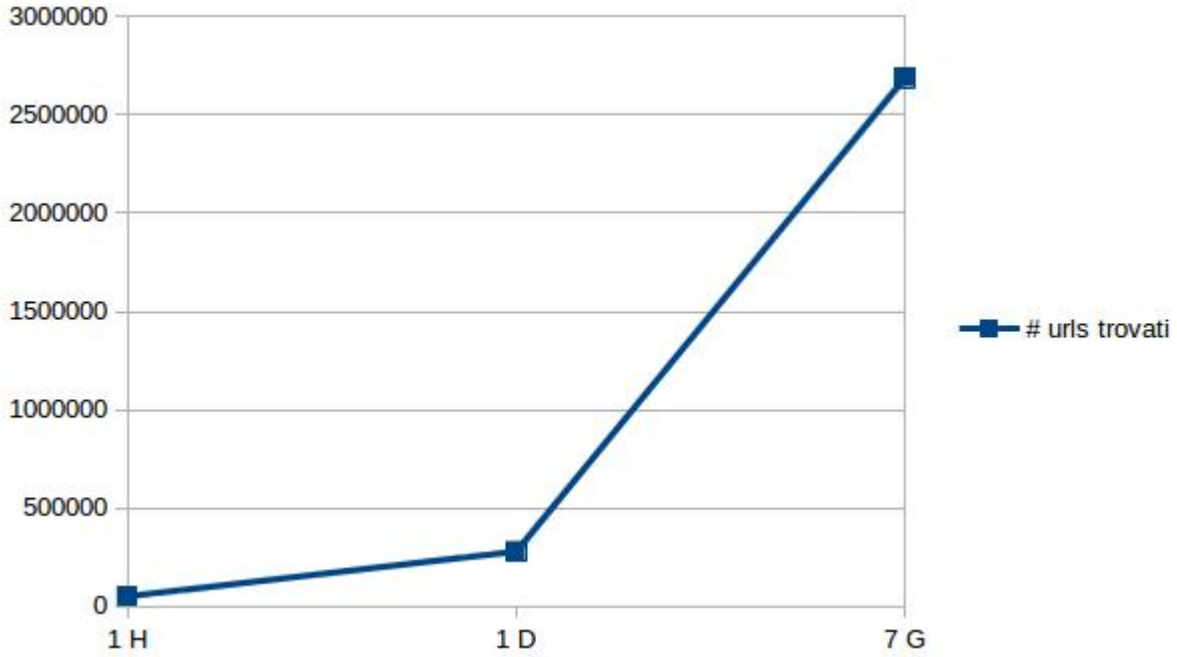
I risultati ottenuti dalla simulazione di 7 giorni sono riportati nei file *statistiche_email_7g.php* e *statistiche_url_7g.php.*

I domini più comuni sono stati:

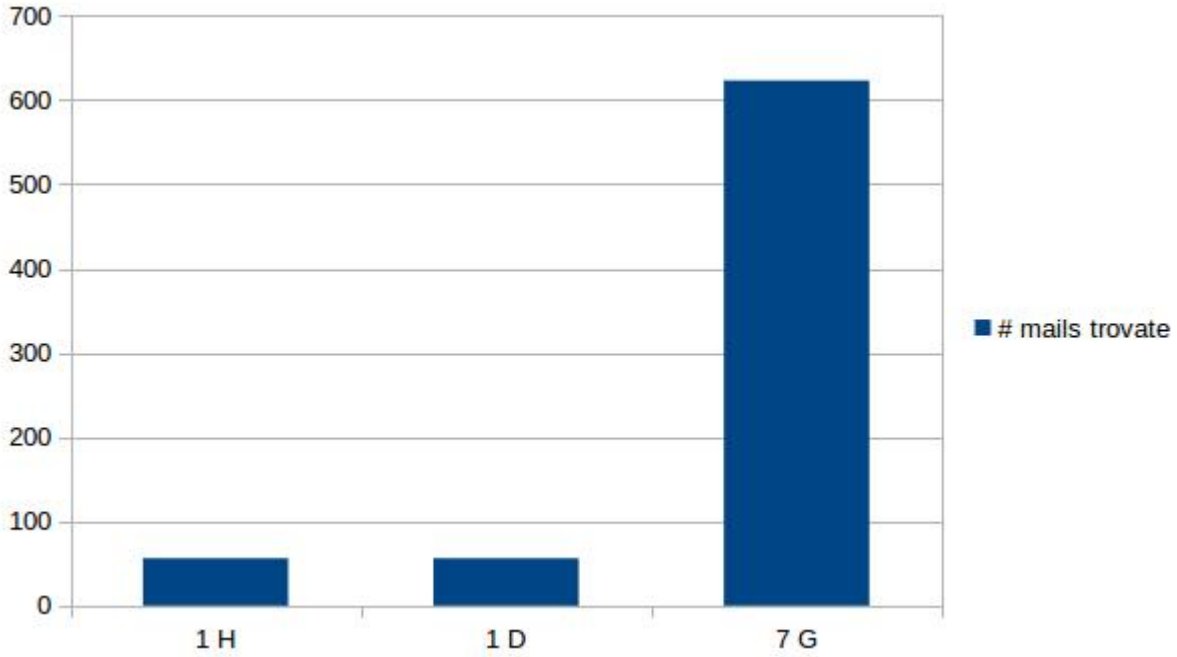


Sono state trovate in totale 623 emails.

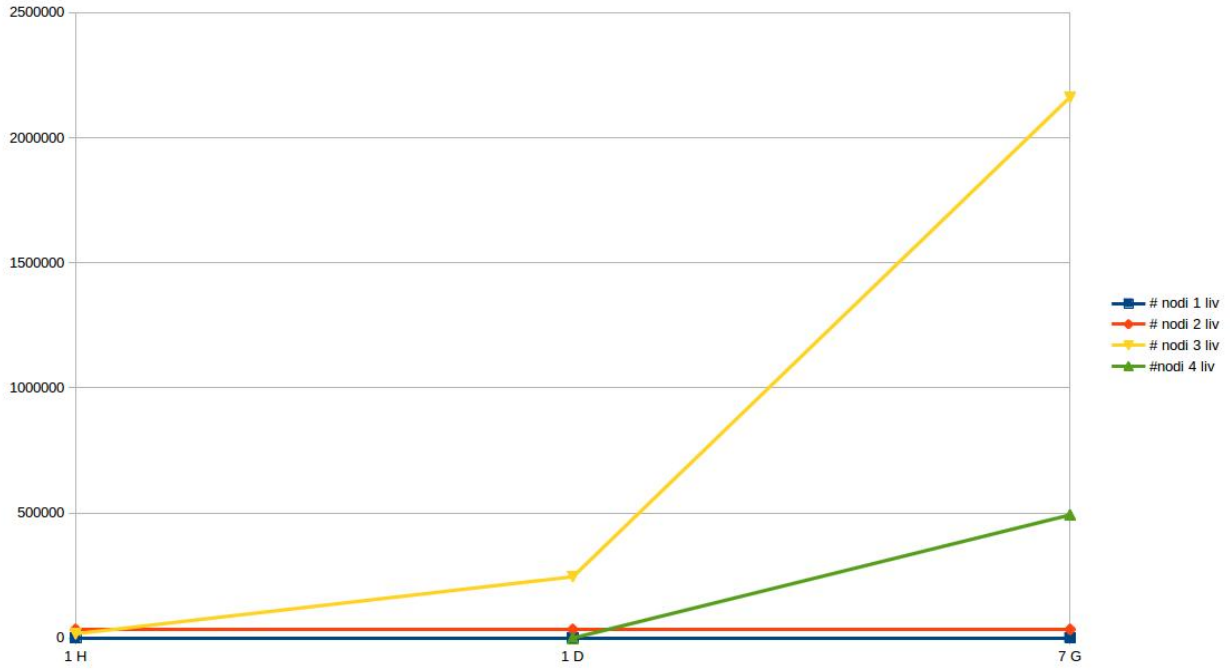
Numero di URLs trovati



Numero di emails trovate



Numero di nodi ad ogni profondità



Credits

Hanno partecipato al progetto (in ordine alfabetico):

- Cantarella Danilo (http://cantarelladanilo.com)
- Maccarrone Roberta (http://robertamaccarrone.altervista.org)
- Parasiliti Parracello Cristina (http://parasiliticristina.altervista.org)
- Randazzo Filippo (http://randazzofilippo.com)
- Safarally Dario (http://dariosafarally.altervista.org)
- Siragusa Sebastiano (http://sebastianosiragusa.altervista.org/)
- Vindigni Federico (http://federicovindigni.altervista.org)