Corso di Programmazione Concorrente

Tesine 2017-2018

Valter Crescenzi

http://crescenzi.inf.uniroma3.it

Tesine vs Homework

- Gli homework rappresentano un percorso di studio guidato
  - La soluzione è già ampiamente delineata nel testo
- Le tesine rappresentano un momento di studio “libero” e non vincolato, di problemi non banali e di chiarissima utilità pratica contingente
- Ma *ostinatamente* allineati agli obiettivi formativi di questo corso sulla programmazione concorrente
- Lo scopo *dichiarato* è *avvicinare*, *senza vincolare*, lo studente ad un argomento su cui svolgere una tesi di laurea
- Senza per questo voler rinunciare all’aspetto formativo che ogni contenuto, di ogni corso, dovrebbe *sempre* avere

Mi conviene fare una tesina?

- Se l'obiettivo è massimizzare il voto…
  - ...*forse*
- Se l'obiettivo è anche minimizzare lo sforzo
  - ...sicuramente no
  - sicuramente l'operazione che ha il migliore rapporto costi/benefici è preparsi bene per l'orale
- Se l'obiettivo è considerare una potenziale tesi di laurea...
  - ...sicuramente si
- E se non intendo comunque svolgere un tesi su questi argomenti o correlati?
  - sicuramente l'operazione che ha il migliore rapporto costi/benefici è preparsi bene per l'orale
  - Sono ancora in dubbio. Che fare?
    - sicuramente l'operazione che ha il migliore rapporto costi/benefici è preparsi bene per l'orale

Sommario

- Tesine
  - Speed-up Contests
    - XPath Evaluation
    - XPath Fixed Point
  - Lettura di articoli scientifici (*JACM)*
    - Su algoritmi *non-bloccanti*
    - Biased Locking
    - Swarm Locking
  - Structure-Driven Crawling
    - *Incremental Crawling*
    - *Parallel Corpora (DiffWeb)*
  - Compressione ed Archiviazione di pagine su larghissima scala (Web Archive)
  - Utilizzo pratico di un framework per la scrittura di codice ad attori (*Akka*)
    - Risolvere HWC2 con Attori (Homework *HWA*)
    - Progettare un crawler su larga scala basato su Akka
  - Hypertextual Logging
    - Un sistema di logging ipertestuale pensato per codice concorrente
  - Extraction+Linkage+Matching=Fusion

Speed-up Contest: Tesina *XPath*

- Si vuole parallelizzare un algoritmo per il calcolo massivo di espressioni XPath su collezioni di pagine Web rispondenti al medesimo HTML template
- Obiettivo: massimo speed-up
- Valutatore di N~10^3 XPath su M~10^3 pagine
- Utilizzando Cyberneko per caricare i DOM di pagine Web
  - *http://nekohtml.sourceforge.net/*
- A sua volta questo utilizza una libreria nota come Xerces
  - *https://xerces.apache.org/xerces2-j/faq-dom.html*
- Che però non è thread safe!
  - *https://xerces.apache.org/xerces2-j/faq-dom.html#faq-1*

Speed-up Contest: Tesina *FixedPoint*

- Un algoritmo innovativo per l'inferenza automatica di espressioni XPath
- Data una collezione di pagine che rispondono allo stesso HTML template
- Max speed-up per un algoritmo che itera queste fasi:
  - Generazione di un insieme di regole XPath che estraggono tutti i valori di un campione di pagine
  1) Estrazione di vettori di valori tramite le regole
  2) Rigenerazione delle regole per i valori estratti
  - Ripeti 1-2 sino alla convergenza (dell'ins. regole/vettori/valori estratti)

Tesina *Swarm*

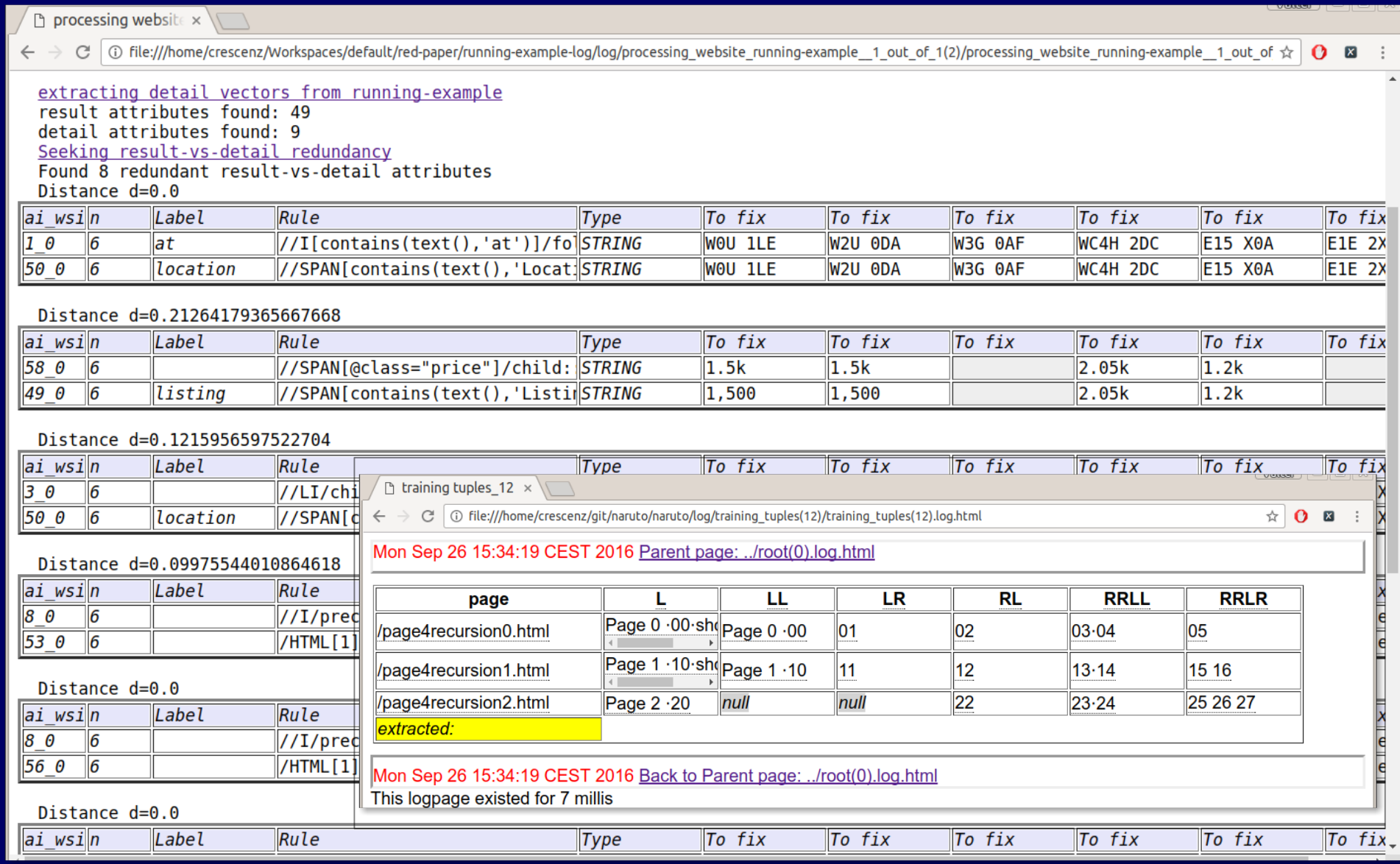
Le grosse collezioni (dinamiche) non vengono mai protette da un solo lock

- *Lock-splitting:* molteplici lock per proteggere *regioni* diverse della collezione.
- Al crescere del numero di lock:
  - Migliora il livello di parallelismo consentito
  - Peggiora l'uso di memoria
- Alberi, Grafi, ed in generale enormi collezioni dinamiche possono crescere in direzioni non prevedibili creando squilibri in ogni distribuzione inizialmente equa dei lock
- Vediamo i lock come membri di uno swarn per "proteggere" l'intera collezione
- Gli accessi concorrenti possono perturbare l'equilibrio:
  - più si concentrano nel tempo e nello spazio, più dovrebbe essere aumentata la granularità, per non minare i livello di concorrenza
  - più si dirano nel tempo e nello spazio più i lock, più dovrebbero rimuoversi perché sostanzialmente inutili
- I lock possono:
  - *muovere* verso le regioni sino a raggiungere un punto di equilibrio
  - *sdoppiarsi/morire* in funzione della dimensione e forma della struttura

Tesina *HLog*

- La consultazione e di log è un’attività
  - molto onerosa
  - Log di codice multi-thread spesso illegibili
- HLog: un framework per il log ipertestuale
  - Il log è organizzato in pagine HTML
  - Le pagine formano un sito intero
  - Le chiamate di metodi possono facilmente essere loggate con link pagina padre/figlia
  - Diversi thread possono loggare in distinte pagine/percorsi del sito di log
- Specializzare HLog per il supporto del logging di codice multi-thread

Esempio di Log Ipertestuale



processing website

file:///home/crescenz/Workspaces/default/red-paper/running-example-log/log/processing_website_running-example__1_out_of_1(2)/processing_website_running-example__1_out_of

extracting detail vectors from running-example
result attributes found: 49
detail attributes found: 9
Seeking result-vs-detail redundancy
Found 8 redundant result-vs-detail attributes
Distance d=0.0

ai_wsi	n	Label	Rule	Type	To fix	To fix	To fix	To fix	To fix	To fix
1_0	6	at	//I[contains(text(),'at')]/fo	STRING	W0U 1LE	W2U 0DA	W3G 0AF	WC4H 2DC	E15 X0A	E1E 2X
50_0	6	location	//SPAN[contains(text(),'Locat	STRING	W0U 1LE	W2U 0DA	W3G 0AF	WC4H 2DC	E15 X0A	E1E 2X

Distance d=0.21264179365667668

ai_wsi	n	Label	Rule	Type	To fix	To fix	To fix	To fix	To fix	To fix
58_0	6		//SPAN[@class="price"]/child:	STRING	1.5k	1.5k		2.05k	1.2k	
49_0	6	listing	//SPAN[contains(text(),'Listi	STRING	1,500	1,500		2.05k	1.2k	

Distance d=0.1215956597522704

ai_wsi	n	Label	Rule	Type	To fix	To fix	To fix	To fix	To fix	To fix
3_0	6		//LI/chi							
50_0	6	location	//SPAN[c							

Distance d=0.09975544010864618

ai_wsi	n	Label	Rule
8_0	6		//I/prec
53_0	6		/HTML[1]

Distance d=0.0

ai_wsi	n	Label	Rule
8_0	6		//I/prec
56_0	6		/HTML[1]

Distance d=0.0

ai_wsi	n	Label	Rule	Type	To fix	To fix	To fix	To fix	To fix	To fix

training tuples_12

file:///home/crescenz/git/naruto/naruto/log/training_tuples(12)/training_tuples(12).log.html

Mon Sep 26 15:34:19 CEST 2016 Parent page: ../root(0).log.html

page	L	LL	LR	RL	RRLL	RRLR
/page4recursion0.html	Page 0 ·00·sh	Page 0 ·00	01	02	03·04	05
/page4recursion1.html	Page 1 ·10·sh	Page 1 ·10	11	12	13·14	15 16
/page4recursion2.html	Page 2 ·20	null	null	22	23·24	25 26 27
extracted:						

Mon Sep 26 15:34:19 CEST 2016 Back to Parent page: ../root(0).log.html
This logpage existed for 7 millis

Tesina *JACM*

- Leggere, commentare, (e farmi capire!) uno tra gli articoli scientifici scelto tra i seguenti, eventualmente preparando una breve presentazione
  - *Split-ordered lists: Lock-free extensible hash tables*
    Ori Shalev, Nir Shavit

    http://dl.acm.org/citation.cfmdoid=1147954.1147958
  - *DomLock: a new multi-granularity locking technique for hierarchies.*

    Saurabh Kalikar, Rupesh Nasre

    http://dl.acm.org/citation.cfm?id=2851164
  - *Fine-grained adaptive biased locking*

    Filip Pizlo, Daniel Frampton, Antony L. Hosking

    http://dl.acm.org/citation.cfm?id=2093157.2093184

Tesina *DiffWeb*

- E' stato ideato un algoritmo di crawling innovativo specializzato per trovare e catturare parallel-corpora da siti web strutturati e multi-lingua

- Parallel corpora: collezione di *documenti* di medesimo contenuto tradotti in più lingue

ZH	天下大勢，分久必合，合久必分。
EN	Long divided, the world will unite; long united, it will fall apart.
PT	O império, unir-se-á após a divisão e dividir-se-á após a união

Perché Accumulare Parallel Corpora?

- Motivazioni
  - Esiste una fiorente industria, tuttora in forte espansione, per la traduzione professionale

- Esistono diversi algoritmi pubblici ML per la MT
  - Machine-Translation
- Si possono scaricare gratuitamente da internet, non sono considerati un asset da proteggere! Perché?
- Perché si tratta di algoritmi *statistici* **affamati** di **enormi** quantità di esempi
- Il vero asset sono i *training data*!
  - che difatti nessuno pubblica...

Crawling di Parallel Corpora

- Non è un dettaglio: per essere competitivi c'è bisogno di esempi per *miliardi di parole*
- Google-Translate si basa su un *crawling esaustivo del Web*
  - in presenza di risorse di calcolo praticamente illimitate e dello snapshot del Web…

- E' possibile collezionare i siti che offrono i parallel-corpora anche senza essere Google
  - Basta ultra-specializzare gli algoritmi di crawling
  - Crawling efficiente e mirato della *sola* porzione di interesse del Web

Tesina *DiffWeb* (max 2 persone)

- Progettare un'architettura ad attori dell'intero sistema
  - ovvero definire quali siano gli attori principali e cosa comunicano
- Sviluppare alcuni di questi attori>>

Structure Driven Crawling

- Indipendentemente dalla quantità di risorse (tempo, memoria, bandwidth) disponibili un crawler su larga scala rimane sempre e comunque a corto di risorse
- Se l'obiettivo è tanto ambizioso quanto coprire l'intero Web, usare meno risorse significa
  - coprirne una parte più significativa
  - disporre di snapshot più aggiornati
- Progettazione e sviluppo di crawler specializzati per
  - News
  - Companies

  ad uso limitato di risorse e guidati dalla struttura del sito e delle sue pagine

Generazione Automatica di Wrapper

- E’ disponibile un generatore di wrapper allo stato dell’arte
  - Completamente automatico (unsupervised)
  - N log N
- Date N pagine (stesso template) produce
  - una relazione (nel senso di BD) di N tuple (una per ciascuna delle pagine usate per il training
- In breve
  - *Un complicato Diff specializzato per codice html*
  - Riesce a capire qual’è il template sottostante per poi separarlo dai contenuti che sono estratti e riorganizzati in formato relazionale

	RLRLLLRLLLLLRRRLLI	RLRLLLRLLLRL	RLRLLLRLLLRLI	
	null	</th id:header15/>·</td headers/>·81186·</td	81186	</th
	null	</th id:header15/>·</td headers/>·89179·</td	89179	</th
21	0657333213	</th id:header15/>·</td headers/>·87356·</td	87356	</th
21	0657333214	</th id:header15/>·</td headers/>·81614·</td	81614	</th
	null	</th id:header15/>·</td headers/>·81135·</td	81135	</th

	RLRRLRLRLLI	RLRRLRLRLRLLLLLRI	
/>·m	Mobile·computing	*null*	*null*
/>·o	Programmazione·orientata·agli·oggetti	Programmazione·concorrente	</h
atze	Basi·di·dati·i	Basi·di·dati·ii	</h
>·pa	*null*	*null*	*null*
>·ga	*null*	*null*	*null*

Crawling Incrementale

- Pensiamo alle pagine di contenuto dinamico
- Collezioniamo una serie di snapshot in istanti diversi dello stesso sito (versioni di pagina)
- Le diverse *versioni* di una stessa pagina, possono essere viste come elementi di una collezione di pagine con stesso template
- Separiamo la parte che statica da quella dinamica
- Studiamo le correlazioni tra queste informazioni per capire come conviene navigare il sito per tenersi aggiornati sui suoi cambiamenti ed estrarne efficientemente i contentuti

Web Archive

- Siete *Web Archive*
  - Una organizzazione che spera e pretende di archiviare tutte le pagine del Web per sempre, allo scopo di non perderne mai memoria
- Supponete di usare *Amazon Glacier* per archiviare le pagine e fate due conti…
- Comprimere un qualche punto % in più significa usare risparmiare molti, ma *molti!* soldi
- Usiamo il generatore di wrapper per fattorizzare il template ed omogenizzarne i contenuti prima di passarli ad un compressore standard
- Alcuni esperimenti preliminari molti incoraggianti

Extraction+Linkage+Matching=Fusion

- Date N collezioni di pagine pubblicanti informazioni sugli stessi oggetti (ad esempio stocks, people, prodotti) e prelevanti da M siti
- Un algoritmo completamente automatico permette di
  - Trovare le regole di estrazione corrette
  - Allineare le pagine provenienti da siti distinti ma che parlano degli stesso oggetti
  - Capire quali regole (di siti diversi) finiscono per estrarre lo stesso attributo
- Contestualmente!