MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY

Identifikace a omezení přístupu k „nevhodným“ stránkám ve webovém archivu

BAKALÁŘSKÁ PRÁCE

Filip Kusalík

Brno, 2009

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: Ing. Petr Žabička

Poděkování

Rád by som sa poďakoval tímu WebArchiv.cz, ktorí so mnou spolupracovali a konzultovali návrh a implementáciu systému. Menovite predovšetkým pánom A. Brokešovi a Mgr. Bc. L. Matějkovi. Ďalej moja vďaka patrí pánovi Ing. P. Žabičkovi za vedenie práce, ochotu a podporu.

Shrnutí

Práca si kladie za cieľ navrhnúť a zrealizovať systém pre identifikovanie „nevhodných“ záznamov podľa zákonnej úpravy vo webovom archíve *webarchiv.cz* [2] a ich zneprístupnenie. Programové riešenie systému je implementované v jazyku Java. Program využíva funkcie viacerých externých knižníc. Zvolenou platformou bol Windows s vývojovým prostredím NetBeans.

Práca bola vysádzaná v LaTeX

Klíčová slova

Závadní stránky, identifikace nevhodnosti, WebArchiv, modulární systém, archivace webu, internetový filter, pornografia, rasizmus, násilie, omezení přístupu, WayBack, ImpEval, ARCWayBack, arcrepos.

Obsah

Úvod

Webový archív obsahuje rozmanité portfólio dokumentov ako po typovej stránke, tak po stránke kategorického obsahu. S vytváraním, uchovávaním a rozširovaním archívu vznikla potreba rozlišovať v archíve záznamy na legislatívne neregulované (nezávadné, vhodné) a regulované(závadné, nevhodné). Táto práca si kladie za cieľ navrhnúť a naimplementovať nástroj na rozlišovanie týchto záznamov, ktorý sa snaží určiť hranicu závadnosti.

V 1 kapitole popíšem teoretické východiská identifikácie „nevhodných" stránok. Budem sa zaoberať kategorizovaním nevhodnosti a možnosťami ako ju identifikovať a ako ju spracovávať. V kapitole 2 navrhnem nástroj *ImpEval*[1], proces spracovávania záznamov v archíve, štruktúru nástroja vychádzajúcu z tohto procesu a popíšem úpravy v stávajúcom databázovom modeli webového archívu. Zaoberať sa samotnou programovou implementáciou budem v kapitole 3.

Myšlienka celého systému bude v zabalení každého záznamu do obalového rozhrania, ktoré bude sebestačne obsahovať všetky údaje vzťahujúce sa k danému záznamu, z dôvodu snahy o vysokú úroveň *granularity*[2] systému. Toto obalové rozhranie bude vložené do modulárneho systému, ktorý približujem v kapitole 4, z ktorého sa postupným volaním modulov definovaných v úložnej štruktúre a ich aplikovaním na obalové rozhranie vráti záznam s nastaveným ohodnotením od každého modulu, v ktorom došlo k určitej identifikácii nevhodnosti (ale i vhodnosti). Záznam bude následne opatrený príslušným príznakom miery ohodnotenia nevhodnosti. Odpoveďou na otázku obmedzenia prístupu sa budem zaoberať v kapitole 5, kde načrtnem ideu zabránenia prístupu k nevhodným záznamom.

Výsledkom celej práce bude ukážková evaluácia ktorú popíšem v kapitole 6. Idea tejto evaluácie bude spočívať vo vyhodnotení ukážkových domén v zastúpení nevhodnosti, vhodnosti a spornosti. Nad týmito doménami spustím nástroj *ImpEval* a výsledky evaluácie zhrniem do zrozumiteľnej podoby. V kapitole 7 v krátkosti popíšem inštalačné, konfiguračné možnosti a projektové riadenie nástroja *ImpEval*.

---

1. Impropriety Evaluator – ohodnocovač nevhodnosti
2. Úroveň podrobnosti identifikácie jednotlivých zložiek informačného systému

Kapitola 1

Identifikácia „nevhodných“ stránok

1.1 Aké sú nebezpečenstvá na internete? [6]

Rozšírenosť a rozmanitosť Internetu poskytuje množstvo informácii v globálnom merítku, ktoré je základom schopnosti otvorenosti a prístupnosti ku každému a preto je jeho obsah výrazne neregulovaný. Otázka riešenia *vhodnosti* (ďalej i *nezávadnosti*) obsahu je náročný proces, parciálne kvôli existencii vysokého stupňa anonymity ako pre užívateľov, tak pre poskytovateľov obsahu. Ďalšia prekážka je legislatíva, pretože Internet nemusí nutne rešpektovat štátne hranice a právne systémy.

Z druhého pohľadu, ako na Internete pribúda množstvo užitočných informácií, tak bujnie rast materiálu, ktorý zvykne byť považovaný za *nevhodný* (ďalej i *závadný*). Bežné vnímanie tohto názoru je vyhranené na materiál erotickej povahy, ktorý môže byť prehliadaný deťmi. Fakt je, že problém zahŕňa typovo väčší počet skupín materiálu, ktorý je často považovaný za nechcený všetkými členmi spoločnosti.

Nasledujúci zoznam analyzuje *nevhodný* obsah do základných skupín:

Spoločenský – nenávisť, netolerancia;

Náboženský – netolerancia, rúhanie, okultizmus, satanizmus;

Nemravný – pornografia, nahota, erotický materiál;

Násilnícky – zbrane, drogy, bomby, ilegálna aktivita, anarchia, násilie;

Počítačový – *warez*[1], lámanie proprietárnych šifier, kradnutý software;

Politický – terorizmus, organizovaný odboj.

V neposlednej rade sú tu okrajové záležitosti týkajúce sa komunikácie cez Internet – chatovacie miestnosti a online komunity sú známe teritóriá

1. Ukradnutý software, u ktorého typicky bola potlačená ochrana proti kopírovaniu, alebo nutnosť registračného procesu – `http://isp.webopedia.com/TERM/W/warez.html`

pre pedofilov a ostatných patologických úchylov. Ďalej je tu skupina jemnejšie ladeného materiálu, ktorý môže byť *závadný* hlavne pre deti a mládež:

- Online nakupovanie.
- Online zoznamka a hry.
- Ovplyvnenie nevyžiadanou reklamou.

Internet je ale na druhej strane výborný nástroj pre výuku, riešenie úloh, zjednodušovanie komunikácie a procesov. Pravdepodobnosť kontaktu s *nevhodným* obsahom môže byť minimalizovaná používaním zopár základných preventívnych techník, takže šanca naraziť na zlú skúsenosť bude veľmi malá.

1.2 Filtre a blokátory obsahu [6]

Filtre obsahu patria do kategórie softwarových nástrojov, ktorých úlohou je filtrovanie *nevhodných* webových stránok, alebo podobne ladeného materiálu pred zobrazením. Blokátory obsahu patria do kategórie softwarových nástrojov, ktorých úlohou je kompletné, alebo parciálne zablokovanie určitého druhu aktivít ako napr. chat.

Filtre obsahu musia odstrániť *nevhodné* informácie internetových stránok so zachovaním informácií užitočných a overených. Tento proces je náročný hlavne kvôli množstvu variácií obsahu, rozličným úrovniam slušnosti pre rôznorodých užívateľov, takisto ako kultúrnym a právnym zvyklostiam a predpisom pre väčšinu audiencií požadujúcich kontrolu obsahu. A navyše, na Internet je denne umiestnené množstvo čerstvého obsahu [18, s. 39], ktorý sa môže svojou povahou *závadnosti* vymykať zo stávajúceho portfólia nechcených skupín.

Filtre musia preto na jednej strane podporovať výraznú úroveň flexibility, ktorá na strane druhej vyžaduje rozsiahle nastavenie pre každú individuálnu situáciu. Funkčnosť inteligentného analyzátora by mala byť zakotvená v zásade dobrého rozpoznania bližšie neurčenej množiny *nechcených* webových stránok oproti stránkam *chceným*, obsahovo nevinným, napr. výukové, informačné stránky o návykových látkach, sexuálnej výchove, alebo stránkam z dvojznačným obsahom. Misiou moderného filtrovacieho software je dosiahnutie tohto stavu s akceptovateľnou úrovňou úspechu.

Kapitola 2

Návrh systému

WebArchiv.cz[1] je projekt [13, kapitola 1], ktorý zastrešuje snahu o dlhodobé uchovanie a sprístupnenie online dostupných elektronických informačných zdrojov. Skrátený popis procesu archivácie internetových stránok môže byť zhrnutý do nasledujúcich postupných krokov:

1. Sklízení obsahu v digitálnej forme nástrojom *Heritrix*[2], ktorý zo sklízených internetových stránok (ďalej záznamy) vytvára rovnomerne veľké archívne súbory[3].

2. Na uložených archívnych súboroch je spustený proces *ARCWayBack* popísaný v [16, strana 42], prechádzajúci archívne súbory v repozitári a v nich obsiahnuté záznamy a uloží ich do databázového modelu *arcrepos*, ktorý je bližšie popísaný v [16, strana 34].

3. Na týchto záznamoch je spustený nástroj *ImpEval* ktorý popisujem v 2.1 v tejto práci, ktorý prejde každý jednotlivý záznam a vyhodnotí u neho, či je *vhodný*, alebo *nevhodný* a uloží o miere *nevhodnosti* príznak do databáze.

4. Jednotlivé záznamy sú sprístupnené pomocou nástroja *WayBack* [10], ktorý na základe zadaného URL a času vyberie a zobrazí odpovedajúci záznam. V prípade, že miera nevhodnosti záznamu prekračuje hranicu nastavenú v 3.3, záznam sa nezobrazí.

---

1. `http://www.webarchiv.cz/`
2. `http://crawler.archive.org/`
3. Vo verzii 1.14.1 Heritrix vytvára súbory ARC [1], v budúcnosti je plánovaný prechod na súbory WARC [3].

2.1 Projekt Impeval

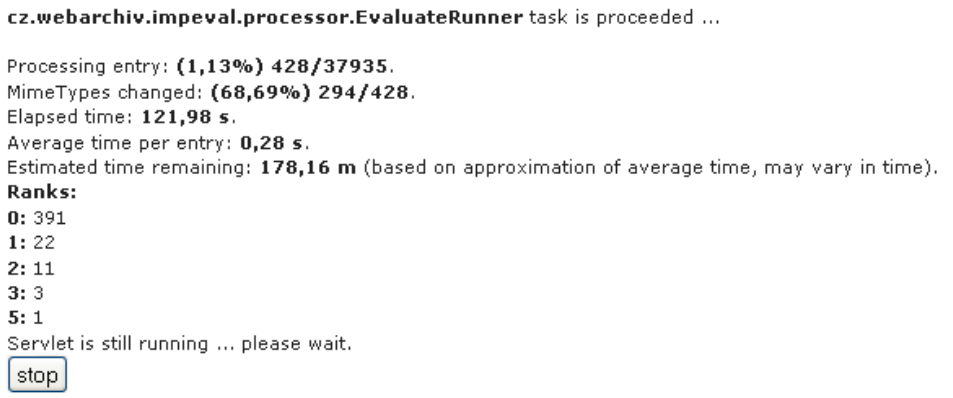
Nástroj *ImpEval* funguje vo forme servletu jazyka *Java*[4], ktorý je umiestnený v kontajneri na webovom serveri *Apache Tomcat*[5]. Používa databázový model *arcrepos* navrhnutý, implementovaný a popísaný v [16, strana 34]. Aplikáciu som vytvoril ako Netbeans[6] projekt, ktorý používa nasledovnú štruktúru projektu:

Web Pages – v ktorých je umiestnený konfiguračný súbor kontextu, v ktorom je servlet spustený, konfiguračný súbor servletu, ďalej JSP stránky, HTML, CSS, obrázky a súbory obsahujúce licenčné ujednanie popísané v 7.3;

Source Packages – v ktorých sú umiestnené zdrojové kódy aplikácie 2.2;

Libraries – v tomto adresári sa nachádzajú knižnice tretích strán, potrebné pre korektný preklad a beh aplikácie.

Aplikácia je po umiestnení (*deploy*) na server dostupná cez webové rozhranie. Po spustení zobrazuje v pravidelných intervaloch priebeh operácie, ktorý je naznačený na obrázku 2.1.



Obrázok 2.1: Ukážka priebehu spracovávania archívu.

4. http://java.sun.com/javaee/technologies/
5. http://tomcat.apache.org/
6. http://www.netbeans.org/

2.2 Balíky

Zdrojové kódy aplikácie sú rozdelené do viacerých balíkov (package):

⟨*default package*⟩ – balík obsahujúci súbory properties;

cz.webarchiv.impeval – základný balík;

cz.webarchiv.impeval.exception – balík obsahujúci výnimky;

cz.webarchiv.impeval.libs – zahŕňa knižnice;

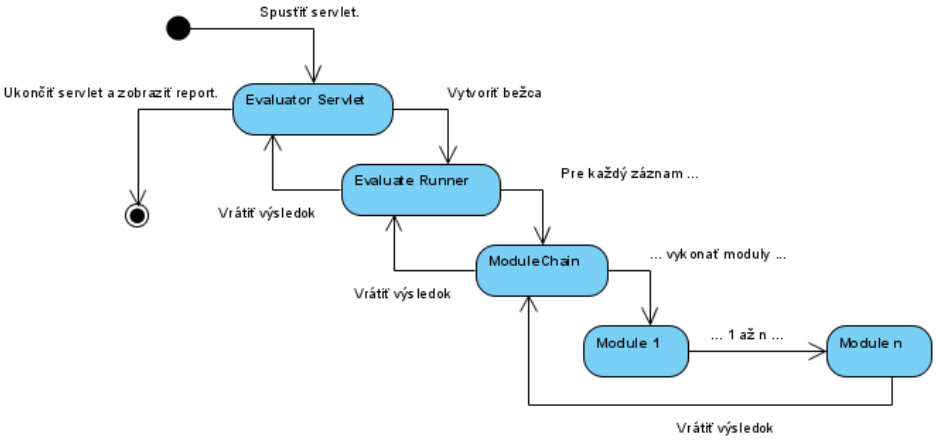
cz.webarchiv.impeval.modules – moduly;

cz.webarchiv.impeval.processor – balík obsahujúci bežce nad bázou dát;

cz.webarchiv.impeval.servlets – servlety.

Balíky *exception*, *processor* a *servlets* sa snažia zachovať štruktúru aplikácie ARCWB [16, strana 41]. Každá trieda má definovaný privátny atribút typu *java.util.Logger* [8], pomocou ktorého zapisuje užitočné informácie, ale i prípadné chyby a výnimky.

2.3 Beh programu



Obrázok 2.2: Beh programu.

Beh aplikácie sa dá zovšeobecniť do nasledovných krokov zobrazených na obrázku 2.2:

1. Užívateľ zobrazí servlet popísaný v 3.1 a spustí program;
2. servlet vytvorí inštanciu bežca popísaného v 3.2 nad bázou dát popísanou v 2.4;
3. ktorý pre každý záznam spustí modulárny vyhodnocovací systém popísaný v kapitole 4;
4. a vyhodnotí záznam podľa 3.3.

2.4 Databázový model

Databázový model *arcrepos* [16, strana 34] som upravil pre potrebu uloženia príznaku závadnosti pridaním atribútu *rank* do tabuľky *Docs*. Jedná sa o atribút typu SMALLINT(1). Ďalej bol vytvorený index *i_rank* typu INDEX. Pre testovacie potreby bol vytvorený i atribút *testrank* typu DOUBLE a k nemu index *i_testrank* typu INDEX. Atribút *rank* obsahuje ohodnotenie škálované podľa 3.3, atribút *testrank* ohodnotenie neškálované – tento atribút odporúčam pre produkčné prostredie z aplikácie odstrániť z kapacitných dôvodov – pracuje s rádovo miliónmi záznamov.

Implementáciu databázovej vrstvy som prvotne navrhol a naimplementoval v nástroji *Apache Cayenne* [7]. *Apache Cayenne* je open-source ORM[7] nástroj poskytujúci perzistentné a cacheovacie možnosti pre jazyk Java a viaceré databázové systémy. S pomocou nástroja *CayenneModeler*[8] som vytvoril XML súbor namapovaný na databázový model vrátane kľúčov, atribútov a vzťahov. *CayenneModeler* následne vygeneruje dva Java balíky:

model.arcrepos.auto – v ňom sú automaticky vytvorené a spravované triedy pre každú tabuľku. V každej tejto triede sú definované atribúty a relačné vzťahy. Tieto triedy by sa nemali upravovať ručne, ale výlučne s pomocou nástroja *CayenneModeler*.

model.arcrepos – v ktorom sú explicitne vytvorené triedy dediace od tried z balíku *model.arcrepos.auto*. Príkladom takejto triedy môže byť trieda *model.arcrepos.Docs* dediaca od triedy *model.arcrepos.auto._Docs*.

Po vygenerovaní oboch balíkov som zaviedol do kódu nasledovné zmeny do triedy *Docs*:

1. Metóda *getContent()* vracajúca obsah daného záznamu z archívneho súboru;

7. Object Relational Mapping
8. Nástroj s grafickým rozhraním na definíciu databázového modelu – `http://cayenne.apache.org/doc20/modeler-guide.html`.

2. *Gettery* a *settery* pre atribúty *id* a *mimeType*.
3. Metóda *getURL()* vracajúca celú URL zloženú zo záznamu v tabuľke *Hosts* a *RestHosts*.

Ďalej *CayenneModeler* vygeneruje nasledovné konfiguračné a mapovacie XML súbory:

ARCReposDomainNode.driver.xml – tu sa definujú prihlasovacie údaje (meno, heslo, URL databáze) k databázi a *connection pool*[9];

ARCReposMap.map.xml – kde sú namapované tabuľky, ich atribúty a vzťahy na objekty v balíku *model.arcrepos.auto*;

cayenne.xml – kde sú uvedené cesty k horeuvedeným dvom súborom.

Je vidieť, že je možné triedy v balíku *model.arcrepos.auto* a konfiguračné a mapovacie XML súbory vytvoriť aj ručne, ničmenej práca s nástrojom *CayenneModeler* tento proces spríjemňuje, zefektívňuje a urýchľuje. Tento prístup, byť rýchly a pohodlný, sa neosvedčil z dôvodu obrovskej spotreby operačnej pamäte a časovej náročnosti vykonávania úlohy. Preto som vybral pôvodnú databázovú vrstvu [16, strana 42].

Tabuľka 2.1 ukazuje prehľad porovnania[10] *Cayenne* a *DBUtils*:

Porovnávaná veličina	**Cayenne**	**DBUtils**
Iniciačná pamäť	8.0 MB	0.5 MB
Pamäť spotrebovaná jedným záznamom	5.0 MB	0.1 MB
Priemerná rýchlosť spracovania jedného záznamu	Rádovo *s*	Rádovo desiatky *ms*
Pohodlnosť práce s nástrojom	Veľká	Stredná
Veľkosť zdrojového kódu	Malá	Veľká
Logika objektového návrhu	Veľká	Stredná

Tabuľka 2.1: Porovnanie databázových nástrojov *CayenneModeler* a *DBUtils*

9. `http://java.sun.com/developer/onlineTraining/Programming/JDCBook/conpool.html`
10. Namerané údaje boli získané na vzorke 10,000 záznamov na hardwarovej konfigurácii Intel Core2 Duo T5500 1.66GHz, pamäť 1024MB RAM.

Kapitola 3

Implementácia systému

Pôvodne som program naimplementoval spôsobom, ktorý ako databázová vrstva bol podobne z časového hľadiska a z hľadiska správy zdrojov veľmi nevyhovujúci, lebo bol veľmi náročný ako na výpočetný čas, tak operačnú pamäť. V nasledujúcom texte budem pôvodnú verzia s novou zrovnávať. Sekvenčný diagram systému môže byť dohladaný v prílohe A.

3.1 Servlet evaluatorservlet

Evaluatorservlet nachádzajúci sa v balíku *cz.webarchiv.impeval.servlets* predstavuje interakciu užívateľa s aplikáciou. Po spustení aplikácie zobrazuje priebeh vykonávania príkazu. Tento servlet si drží spojenie k databázi v inštancii triedy *Connector* [16, strana 42], bežca triedy *EvaluateRunner* (3.2) a konfiguráciu *ResourceBundle* [9], ktorá obsahuje hodnotu *path_repository* ktorá predstavuje relatívnu cestu k adresáru s archívnymi súbormi. Kompletné konfiguračné možnosti aplikácie sú uvedené v 7.2. Servlet používa dva HTML formuláre: jeden na spustenie, druhý na zastavenie obnovovania stránky. Servlet sa každé 3 sekúnd obnoví a zobrazí aktuálny priebeh spracovania. Obnovovanie stránky je možné zastaviť tlačítkom stop. Po spustení servletu štartovacím formulárom sa získa inštancia *TaskProcessFactory* [16, strana 43], ktorá slúži na registráciu a držanie si informácii o bežcoch a zaregistruje sa do nej nový bežec triedy *EvaluateRunner*. Servlet vypisuje nasledujúce informácie:

- Počet záznamov celkovo.
- Počet spracovaných záznamov absolútne i percentuálne.
- Počet nesprávnych (a následne opravených) mime typov.
- Uplynulý a odhadovaný zostávajúci čas.
- Priemerný čas spracovávania jedného záznamu.
- Hodnoty ohodnotenia a počet záznamov nimi ohodnotených.

3.2 Bežec EvaluateRunner

Bežec *EvaluateRunner* je potomkom triedy *stepHandler* [16, strana 45] a implementuje rozhranie *Runnable*[1]. Trieda *stepHandler* je určená pre sekvenčné prechádzanie vybranej množiny záznamov, ktoré sú spracovávané v dávkach a je na ne uplatňovaná nejaká akcia (metóda *action* ktorá môže byť prekrytá). Množina záznamov sa pokiaľ možno delí na rovnako veľké podmnožiny (*Batch* – trieda starajúca sa o dávkove spracovanie príkazov [16, strana 42]) ktorých veľkosť je explicitne vo verzii popísanej v [16] nastavená na hodnotu 1000. Implementuje i povinnú metódu rozhrania *Runnable* run.

Trieda *EvaluateRunner* iba prekrýva metódu *action*. V tejto metóde si získa aktuálny spracovávaný záznam, k nemu prislúchajúce hodnoty *host*[2], *mimetype*, názov archívneho súboru. Vytvorí objekt rozhrania *IEntry* (3.3), ktorému predá základné parametre a tým ho zabalí do prenosnej štruktúry, s ktorou vie pracovať modulárny systém. Následne si získa evaluačnú hodnotu a náležite upraví záznamy *rank* a *testrank* v tabuľke *Docs*.

Tento bežec si drží záznamy o počte vykonaných záznamov, celkovom počte záznamov, uplynulom čase, počte nesprávnych (a neskor opravených mime-type) a o hodnotách ohodnotení záznamov a ich počtoch.

3.3 Obalová rozhranie IEntry

Modulárny systém pracuje s rozhraním *IEntry*, ktoré obsahuje všetky potrebné informácie, aby bolo nezávislé a samostatné, čoho som dosiahol zapuzdrením všetkých údajov. Rozhranie definuje potrebné metódy na získanie a uloženie údajov. Pre zvýšenie funkcionality a miery abstrakcie som navrhol triedu *ArchiveEntry*, ktorá toto rozhranie sčasti implementuje. V podstate núti konkrétnu implementáciu naimplementovať iba triedu *getContent()*, ktorá vracia konkrétny záznamu odpovedajúci obsah archívneho súboru:

```
public String getContent();
```

Táto potreba vznikla kvôli predpokladanému prechodu archívu z ukladania dát do archívneho súboru ARC [1] na nový typ archívneho súboru WARC [3], v ktorom sú dáta uložené odlišným spôsobom.

Základná implementácia rozhrania *ARCEntry* pri získaní obsahu ARC

1. `http://java.sun.com/j2se/1.5.0/docs/api/java/lang/Runnable.html`
2. Doména v tvare: názov-domény.tld

súboru získa údaje z hlavičky[3] a uloží do atribútu *headers*, a uloží obsah do atribútu *content*. Pri ďalšom dotaze vráti tento obsah, takže nie je potreba pristupovať do ARC súboru viac krát.

Ohodnotenie záznamu funguje podobne ako vyhodnocovanie dôležitosti stránky u *Google Pagerank*™ [12, kapitola 2.1] – stupnica je v intervale $\langle 0, ..., 10 \rangle$, ale škálovanie nie je lineárne, ale logaritmické. Pri faktore útlmu 0.85 a tým pádom minimálneho ohodnotenia záznamu 0.15 a na určenom logaritmickom základe 3 je škálovanie nasledovné:

0	0	–	0.15
1	0.15	–	0.45
2	0.45	–	1.35
3	1.35	–	4.05
4	4.05	–	12.15
5	12.15	–	36.45
6	36.45	–	109.35
7	109.35	–	328.05
8	328.05	–	984.15
9	984.15	–	2952.45
10	2952.45	–	∞

Tabuľka 3.1: Škálovanie ohodnotenia záznamov.

Priebeh logaritmickej funkcie môže byť dohľadaný v prílohe B. Odporúčanú hodnota závadnosti som stanovil nasledovne: **Stránky s ohodnotením väčším ako 6 sú *závadné*.** Metóda *getEntryRank()* vráti ohodnotenie záznamu v intervale $\langle 0, ..., 10 \rangle$ pre uloženie príznaku do databáze. Transformácia do tohto intervalu prebieha práve podľa tejto logaritmickej funkcie. Pre zvýšenie konfigurovateľnosti je možné jednotlivé hranice intervalov nastaviť v súbore *limits.properties* popísanom v 7.2.

3. Typu org.apache.commons.httpclient.Header

3.4 Knižnice

3.4.1 Text2Words

Zo zadaného textu je trieda *Text2Words* schopná vyextrahovať a vrátiť slová vyskytujúce sa v texte a ich počty v kolekcii *Map* ⟨*String*, *Integer*⟩. Dokáže pracovať s holým textom. Pri spracovaní textu sú ignorované tzv. *stop slova*[4] a to české, ale i slovenské a anglické. Stop slová sú uložené v zozname *stopwords.list*. Ukážka je ukázaná v prílohe C. Vyextrahované slová sú zároveň čistené od rôznych znakov[5].

3.4.2 HTML2Words

HTML2Words zo zadaného HTML najprv vyextrahuje text z tagov, potom zavolá knižnicu *Text2Words*, ktorá vráti slová. Extrahovanie textu z HTML prebieha v troch krokoch pomocou regulárnych výrazov:

1. Vyextrahovanie obsahu atribútov content u meta tagov

```
stripped = html.replaceAll(
    "\\<meta(.*)content=\"?'?([^\"]*)\"?'?(.*)\\>",
        " $2 ");
```

2. Vyextrahovanie textu z tagov

```
stripped = stripped.replaceAll("\\<.*?\\>", "");
```

3. Odstránenie html entít

```
stripped = stripped.replaceAll("&[#a-zA-Z0-9]*;", "");
```

3.4.3 Výnimky

Program používa výnimky z balíku *cz.webarchiv.impeval.exception*:

XMLDBException – výnimky hádzané knižnicou XMLDB;

ImpevalException – výnimky hádzané modulárnym systémom.

---

4. Slová ktoré nemajú žiadnu vypovedajúcu hodnotu a zbytočne by zdržovali spracovanie dôležitejších slov.
5. Jedná sa o znaky ako , . ! ? "( ) ; { } [ ] –.

Kapitola 4

Modulárny systém

Pôvodne som modulárny systém založil na myšlienke prechádzania definovanej štruktúry stromu modulov do hĺbky. Hierarchia a štruktúra stromu bola definovaná v XML súbore *modules.xml*. Moduly museli implementovať rozhranie *IModule*, ktoré predpisovalo metódy spôsobom, ktorý umožňoval pracovať s modulom univerzálne, takže bolo možné moduly jednoducho pridávať, zanorovať do seba a presúvať.

Verzia ktorá nahradila pôvodnú verziu, pracuje s modulárnym systémom lineárne, nie hierarchicky – na jednu stranu tak prichádza o možnosť zanorovať moduly, získava však na ušetrení výpočetného času.

4.1 Programové riešenie

Všetky moduly dedia od triedy *AbstractModule*, ktorá implementuje vyššie zmieňované jednotné rozhranie *IModule*, ktoré zapuzdruje komunikáciu s objektami inštancií *IModule* do unifikovaného procesu hromadnému spracovávaniu modulov.

```
public interface IModule {
    public IEntry execute(IEntry entry)
        throws ImpevalException;
    public boolean isLast();
    public IEntry process(IEntry entry);
}
```

Pôvodné rozhranie obsahovalo ešte 2 metódy:

```
public IEntry mark(IEntry entry, Double appendRank);
public int getEntryRank(IEntry rank);
```

ktoré som ale neskôr preniesol do rozhrania *IEntry*. Metóda modulu *execute()* prevedie vlastný kód. Pôvodne spúšťala aj kód všetkých submodulov.

Odporúčam dediť moduly od triedy *AbstractModule*, nie len implementovať rozhranie *IModule*. Metóda *mark(Double appendRank)* prinásobí k hodnoteniu aktuálne spracovávaného záznamu zadanú hodnotu. Metóda *isLast()* určuje, či je daný modul posledný. Táto možnosť sa využije napríklad v module *Blacklist*, ktorý príslušnosťou záznamu na čiernu listinu určí jeho maximálnu závadnosť a označí sa za posledný a tak je ušetrený výpočetný čas. Metóda *process()* prevádza samotnú evaluáciu aktuálneho záznamu. Konečne, metóda *getEntry()* vráti aktuálny záznam.

Abstraktná trieda *AbstractModule* z rozhrania definovaných metód implementuje metódu *execute()* ako *final*, čo znamená, že metóda nesmie byť ďalej prekrytá. Trieda poskytovala ďalej základnú funkcionalitu metód *isLast()*, *mark(Double appendRank)* a *getEntry()*. V novšej verziu ale zostala len metóda *isLast()*, ostatné dve som presunul do abstraktnej triedy *ArchiveEntry*. Tieto metódy nie je spravidla potrebné implementovať v dediacich triedach, ale je tu ponechaná možnosť v prípade potreby metódy preťažiť. Metódu *process()* označuje ako *abstract*, takže vzniká nutnosť túto metódu preťažiť. Tato metóda je v podstate základom každého modulu a je jediná nutná pre korektný preklad zdrojového kódu.

```
public abstract class AbstractModule
        implements IModule {

    protected boolean isLast = false;

    public final IEntry execute(IEntry entry)
            throws ImpevalException {
        isLast = false;
        return process(entry);
    }

    public abstract IEntry process(IEntry entry);

    public boolean isLast() {
        return this.isLast;
    }
}
```

Pôvodne abstraktná trieda *AbstractModule* v konštruktori nastavila vnútorný atribút rozhrania *IEntry* a názvy dcérskych modulov načítala pomocou objektu *XMLDB* popísanej v 4.2 do fronty *Queue⟨String⟩*[1].

1. http://java.sun.com/docs/books/tutorial/collections/interfaces/queue.html

Bližší pohľad na metódu *execute()* prezentuje algoritmus založený na myšlienke modulárneho systému:

1. Spracovanie obsahu aktuálnym modulom pomocou metódy *process*;
2. overenie finálnosti modulu;
3. a následné zavolanie a spracovanie kódu každým dynamicky zavolaným submodulom.

Celá metóda *execute()* aj pôvodná myšlienka modulárneho systému bola nevýhodná z viacerých dôvodov. Trieda *AbstractModule* čítala dcérske moduly pri vytváraní každého modulu. Vznikali tak zbytočné prístupy k XML a XSD súborom a zbytočne sa vytvárali nové inštancie modulov. Existoval rodičovský modul *Evaluator*, ktorý sám o sebe nič nerobil, iba zastrešoval ďalšie submoduly. Preto bola metóda *execute()* sčasti presunutá do nového objektu *ModuleChain*, ktorý implementuje návrhový vzor *singleton* [15], a vytvára si postupnosť modulov iba raz. Načítavanie modulov prebieha pomocou tzv. *classloading* [5]:

```
public Queue<IModule> loadModules(
        Queue<String> moduleNames) {
    Queue<IModule> mods = new LinkedList<IModule>();
    for (String mod : moduleNames) {
        String className =
            "cz.webarchiv.impeval.modules." + mod;
        Class<?> myClass = Class.forName(className);
        Method myMethod = myClass.getDeclaredMethod(
            "getInstance", new Class[]{});
        IModule module = (IModule) myMethod.invoke(
            null, new Object[]{});
        mods.add(module);
    return mods;
}
```

Záznamy potom spracováva pomocou metódy *traverse(IEntry entry)*.

```
public IEntry traverse(IEntry entry) {

    for (IModule module : modules) {
        entry = module.execute(entry);
        if (module.isLast()) {
            break;
        }
    }
    return entry;
}
```

4.2 XMLDB

XMLDB zaobaľuje XML súbory *modules.xml* (4.2.1) a *modules.xsd* (4.2.2) a poskytuje metódy na prístup k elementom – podľa zadaného mena vráti dcérske moduly s daným menom. Po vytvorení si vytvorí reprezentáciu XML súboru pomocou *DocumentBuilderFactory*[2] a XSD súboru pomocou *SchemaFactory*[3]. Následne overí validitu XML súboru a uloží jeho obsah do privátneho atribútu *doc*. Po zavolaní funkcie:

```
public Queue<String> get(String name) { ... }
```

sa vykoná nad XML súborom *XPath* [17] dotaz:

```
String expression = "//modules/module/@name";
```

ktorý vráti mená dcérskych modulov uložené v kolekcii rozhrania *Queue*, ktorá je navrhnutá na držanie elementov podľa priebehu spracovania a zoraďuje elementy chovaním *FIFO*[4].

4.2.1 Modules.xml

Tento XML súbor uchovával a vyjadroval postupnosť spracovania modulov v hierarchickom strome, v novšej verzii v lineárnej postupnosti. Každý záznam o module v elemente module má atribút *name* s názvom modulu a pôvodnej verzii i dcérsky element *modules* ktorý obsahoval elementy modulov logicky príslušiacich k rodičovskému elementu.

Ukážka zdrojového kódu:

```
<modules>
    <module name="IsBig"></module>
    <module name="Blacklist"></module>
    <module name="Whitelist"></module>
    <module name="MimeType"></module>
    <module name="IsValid"></module>
    <module name="Secure"></module>
    <module name="Keywords"></module>
</modules>
```

---

2. http://java.sun.com/j2se/1.5.0/docs/api/javax/xml/parsers/DocumentBuilderFactory.html
3. http://java.sun.com/j2se/1.5.0/docs/api/javax/xml/validation/SchemaFactory.html
4. *First In First Out* – spracovávaný je vždy najskôr vložený záznam

4.2.2 Modules.xsd

Jednoduchá účinná XSD schéma k súboru *modules.xml*, ktorého validácia prebieha pri každom prístupe do neho. V prípade ručnej editácie súboru z dôvodu manipulácie s elementom modulu (pridanie, odobranie popr. editácia modulu) sa tak rýchlo odhalí chyba.

```
<xsd:element name="modules">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="module" minOccurs="0"
                    maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence></xsd:sequence>
                    <xsd:attribute name="name"
                        type="xsd:string"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

4.3 Moduly

4.3.1 Modul Evaluator

Modul *Evaluator* v pôvodnej verzii obsahoval a spúšťal všetky ostatné moduly. V novšej verzii sa nenachádza.

4.3.2 Modul IsBig

V prípade, že veľkosť záznamu prekročí 2 MB, záznam sa nespracováva z dôvodu časovej náročnosti.

4.3.3 Modul IsValid

Modul *IsValid* vyhodnocuje záznam po stránke schopnosti byť valídny. Vylúčené sú záznamy podľa rôznych pravidiel, v základnej konfigurácii sú to:

- Záznam je binárny, vylúčený z kontroly. Záznamy binárne sú typicky obrázky, videá, multimediálny obsah.
- Záznam označený identifikátorom 4 – *future delete* [16, strana 37].

- Bloky určitých sekvencií bytov (napr. vymazaný obsah – záznamy sa z archívu nevymazávajú, ale prepisujú určitými sekvenciami bytov z dôvodu zachovania konzistentnosti archívu)

4.3.4 Modul Secure

Skontroluje záznam pomocou antivírusového programu tretej strany. Tento modul nie je vo verzii dodávanej s touto bakalárskou prácou naimplementovaný.

4.3.5 Modul MimeType

Modul *MimeType* iba nastaví korektný mime-type daného dokumentu. Prítomnosť tohto modulu je spôsobená vysokým percentom falošných údajov o mime-type uložených v archívoch, primárne kvôli funkcionalite *crawlera Heritrix*, ktorý mime-type zisťuje zo sklízených stránok a nekontroluje ich pravosť. Podľa výsledku ukážkovej evaluácie (6.5) je priemerné percento nesprávnych mime-type približne 44%.

Zisťovanie mime-type je netriviálna záležitosť. Modul *MimeType* poskytuje až trojfázové vyhodnocovanie.

1. Vyhodnotenie pomocou *MimeMagic* [4]. Toto vyhodnotenie je výpočetne veľmi náročné, ale zato veľmi spoľahlivé.
2. Vyhodnotenie pomocou *Headers* (3.3). *Headers* sú získavané priamo z *ARC* súboru, ale udávajúce hodnotu mime-type sa v zázname často nenachádzajú.
3. Vyhodnotenie pomocou *MimetypesFileMap* [11], ktoré iba háda mime-type podľa prípony súboru. Veľmi rýchle, ale nespoľahlivé.

Ďalšie metódy môžu byť nájdené napríklad v [14].

4.3.6 Modul Blacklist

Modul *Blacklist* iteratívne porovnáva host daného záznamu so zoznamom *závadných* stránok.

4.3.7 Modul Keywords

Modul Keywords za pomoci knižnice *Text2Words* popísanej v 3.4.1, poprípade *HTML2Words* 3.4.2 získa slová a ich počty a porovnáva ich so zoznamom kľúčových slov definovaných v súbore *keywords.properties* (viď 7.2).

V prípade, že slovo nachádzajúce sa na stránke sa vyskytuje v zozname kľúčových slov, vynásobí sa počet výskytov na stránke ohodnotením v zozname kľúčových slov. V prípade výskytu viacerých slov, jednotlivé ohodnotenia sa sčítajú. S celkovým hodnotením sa metódou *mark(Double appendRank)* vynásobí stávajúci rank uložený v *IEntry*.

4.3.8 Modul Whitelist

Modul *Whitelist* iteratívne porovnáva host daného záznamu so zoznamom *vhodných* stránok. Funkcionalita je rovnaká ako u modulu *Blacklist*.

Kapitola 5

Obmedzenie prístupu

Webový archív je sprístupnený cez upravený nástroj *WayBack* [10]. Obmedzenie prístupu som realizoval pridaním filtru záznamov do triedy *MYSQLResourceIndex*[1] a metódy:

```
public SearchResults query(final WaybackRequest r)
    throws ResourceIndexNotAvailableException,
    ResourceNotInArchiveException,
    BadQueryException, AccessControlException {

    ...
    filter.add(new MySQLFilterImpeval(limit);
    ...
}
```

Implementácia filtru je jednoduchá a vyzerá nasledovne:

```
public class MySQLFilterImpeval implements MArcFilter {
    private int limit;
    public MySQLFilterImpeval(int lim){
        limit = lim;
    }
    public int getLimit(){
        return limit;
    }
    public String filter() {
        return " AND rank < " + limit + " ";
    }
}
```

Takto bude zaručené zobrazenie iba záznamov s ohodnotením menším ako zadaný limit popísaný v 3.3.

---

1. `http://raptor.webarchiv.cz:8000/trac/browser/projects/WA-CZ/trunk/wayback/src/java/cz/webarchiv/wayback/mysqlcz/MYSQLResourceIndex.java`

Kapitola 6

Ukážková Evaluácia

Vízia spočíva v zosklízení nasledujúcich domén nástrojom *Heritrix*, ktorý ich uloží do ARC súborov:

freefoto.cz – ako doménu ktorú by mal nástroj *ImpEval* vyhodnotiť celú ako *závadnú*, tzv. *závadná doména*;

vlada.cz – ako doménu ktorú by mal vyhodnotiť celú ako *nezávadnú*, tzv. *nezávadná doména*;

sexus.cz – ktorú by mal vyhodnotiť ako *závadnú* iba na určitých stránkach, tzv. *sporná doména*.

A následným spracovaním ARC súborov nástrojom *arcinsert* [16, strana 45], ktorý iteratívne prechádza všetky ARC súbory a ukladá informácie o nich do repozitára *arcrepos* a vyhodnotením pomocou nástroja *ImpEval*, v ktorom sú deaktivované moduly *Blacklist* (viď 4.3.6) a *Whitelist* (viď 4.3.8). Doména *freefoto.cz* by sa s veľkou pravdepodobnosťou nachádzala v zozname *blacklist*, doména *vlada.cz* v zozname *whitelist*. Doména *sexus.cz* by sa pravdepodobne nenachádzala ani v jednom.

6.1 Freefoto.cz

Freefoto.cz je server zaoberajúci sa explicitným pornografickým materiálom ako videá a fotografie. Tento materiál má členený do množstva kategórii, takže je veľmi vhodným kandidátom na zástupcu webových sídiel veľmi nevhodných.

6.1.1 Predpoklad

Bude zosklízených max. 2 GB dát. Vyhodnotenie u textových elementov nad hranicou závadnosti je očakávané 95–100%. Je očakávaný vysoký výskyt binárnych súborov (videá, obrázky), tým pádom nízky výskyt textových záznamov. Cieľom je vyhodnotiť textové elementy (primárne HTML)

ako závadné, nie je nutné vyhodnotiť ako závadné samotné obrázky, pretože ak bude závadná HTML stránka, tak sa nebude dať zobraziť, a tým sa nezobrazia ani obrázky na nej umiestnené.

6.1.2 Výsledok

Zosklízených bolo 20 súborov ARC o celkovej veľkosti 1.86 GB. Po spracovaní nástrojom *ARCWB* bolo do databázového modelu *arcrepos* vložených 10,145 dokumentov sídliacich dohromady na šiestich *hostoch*[1]. Hosts boli zistení SQL dotazom:

```
SELECT * FROM `hosts` WHERE `host` LIKE  '%freefoto.cz';
```

Počet dokumentov bol zistený jednoduchým SQL dotazom (čísla v zátvorke sú identifikátory cudzích kľúčov):

```
SELECT COUNT(*) FROM `docs` WHERE
    `host` IN (2,3,5,622,662,1206);
```

Pri evaluácii bolo zistené, že záznamy o mime-type konkrétneho dokumentu nesúhlasia. Tento fakt dal dôvod vzniku modulu *MimeType* popísanom v 4.3.5, ktorý vyhodnocuje mime-type znova. U sklizne domény *freefoto.cz* bolo zistených 51% nesprávnych mime-type (5173 záznamov). Trvanie evaluácie bolo 5,2 minút na hardwarovej konfigurácii Intel Core2 Duo T5500 1.66GHz, pamäť 1024MB RAM. Priemerný čas spracovania jedného záznamu bol 0,03 s. Hodnoty a počty ohodnotení sú zhrnuté v tabuľke 6.1.

0: 9897	**1:** 72	**2:** 0	**3:** 10	**4:** 0	**5:** 0	**6:** 4	=	9983
7: 3	**8:** 5	**9:** 4	**10:** 150				=	162

Tabuľka 6.1: Hodnoty a počty ohodnotení *freefoto.cz*

Pretože sa do úvahy z predpokladu neberú iné ako textové elementy, bola potreba vyfiltrovať binárny obsah od textového. Dosiahnuté toho bolo pomocou SQL dotazu:

1. freefoto.cz, sk.files.freefoto.cz, img.freefoto.cz, sk.img.freefoto.cz, veronika.freefoto.cz, th.freefoto.cz

```
SELECT * FROM `docs` WHERE host IN (2,3,5,622,662,1206)
    AND mimetype IN (2,5,15);
```

Tento dotaz vrátil 8144 záznamov, čo je 80%. Po odpočítaní duplicitného obsahu u obrázkov, kde namiesto binárneho obsahu sa vyskytoval text informačnej stránky *302 Moved Permanently*, zostalo 247 záznamov schopných byť ohodnotené. Z týchto záznamov boli vybrané len záznamy s ohodnotením rank väčším ako 6 – uvažované v 3.3. Počet týchto záznamov bol 162, čo je 65% (viď obrázok 6.1), takže závadnosť sa oproti predpokladu líšila o 30-35%. Najvyšší dosiahnutý testovací neškálovaný rank bol 4.37087942611823e+099[2].

Obrázok 6.1: Percentuálna závadnosť *freefoto.cz*

2. http://freefoto.cz/czech

6.2 Vlada.cz

Vlada.cz je oficiálna štatutárna stránka, takže je veľmi nepravdepodobné, že bude obsahovať nejaký nevhodný obsah. Z tohto dôvodu bola vybraná ako zástupca webových sídiel s minimálnou nevhodnosťou.

6.2.1 Predpoklad

Bude zosklízených max. 2 GB dát. Vyhodnotenie u textových elementov nad hranicou závadnosti je očakávané 0-2%. Je očakávaný stredne vysoký až vysoký výskyt textových záznamov, tým pádom nízky výskyt binárnych súborov (videá, obrázky). Cieľom je vyhodnotiť textové elementy (primárne HTML) ako nezávadné, lebo nie je predpoklad, že sa na stránkach verejnej inštitúcie budú nachádzať nevhodné údaje, prípadne ak áno, tak v minimálnom množstve.

6.2.2 Výsledok

Zosklízených bolo 21 súborov ARC o celkovej veľkosti 1.37 GB. Po spracovaní nástrojom *ARCWB* bolo do databázového modelu *arcrepos* vložených 37,935 dokumentov sídliacich dohromady na dvadsaťjedna *hostoch*[3].

Hosts a počet dokumentov boli zistení obdobne ako u domény *freefoto.cz* SQL dotazom. U sklizne domény *vlada.cz* bolo zistených 39% nesprávnych mime-type (14851 záznamov). Trvanie evaluácie bolo 48 minút na hardwarovej konfigurácii Intel Core2 Duo T5500 1.66GHz, pamäť 1024MB RAM. Priemerný čas spracovania jedného záznamu bol 0,08 s. Hodnoty a počty ohodnotení sú zhrnuté v tabuľke 6.2.

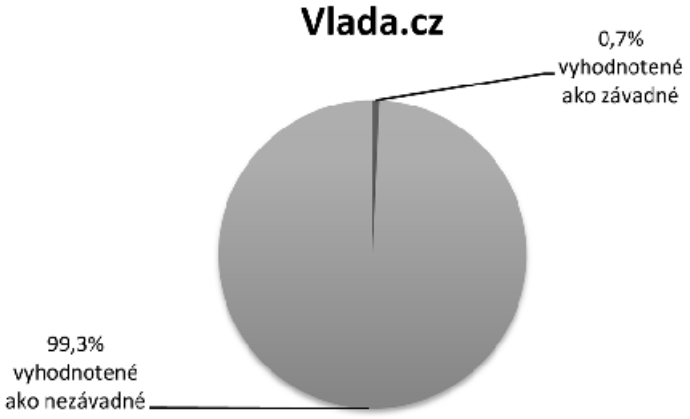
0: 33737	**1:** 2512	**2:** 893	**3:** 405	**4:** 170	**5:** 86	**6:** 27	= 37830
7: 45	**8:** 15	**9:** 5	**10:** 40				= 105

Tabuľka 6.2: Hodnoty a počty ohodnotení *vlada.cz*

Po vyfiltrovaní textových záznamov od binárnych podobne ako u domény *freefoto.cz* bolo zistených 8,953 textových záznamov, čo je 23%. Z tých-

3. vladaprovas.vlada.cz, vlada.cz, www.vlada.cz, kormoran.vlada.cz, eu.vlada.cz, racek.vlada.cz, eklep.vlada.cz, icm.vlada.cz, vanoce.vlada.cz, isap.vlada.cz, web2009.vlada.cz, web2006.vlada.cz, wtd.vlada.cz, www.vvzpo.vlada.cz, lidskaprava.vlada.cz, rvkpp.vlada.cz, 2009.cadros.vlada.cz, 2008.cadros.vlada.cz, www.vladaprovas.vlada.cz, denceskestatnosti.vlada.cz, www.rvnno.vlada.cz

to záznamov boli vybrané len záznamy s ohodnotením rank väčším ako 6 - uvažované v 3.3. Počet týchto záznamov bol 69, čo je 0,7%, takže závadnosť sa oproti predpokladu nelíšila. Najvyšší dosiahnutý testovací neškálovaný rank bol 61152351.624[4].



Obrázok 6.2: Percentuálna závadnosť *vlada.cz*

6.3 Sexus.cz

Sexus.cz je diskusné fórum a magazín o ľudskej sexualite, takže je veľmi pravdepodobné, že bude obsahovať určité množstvo nevhodného materiálu, preto bola vybraná ako zástupca webových sídiel sporných.

6.3.1 Predpoklad

Bude zosklízených max. 2 GB dát. Vyhodnotenie u textových elementov nad hranicou závadnosti je očakávané 15-20%. Je očakávaný veľmi nízky výskyt binárnych súborov (videá, obrázky).

6.3.2 Výsledok

Zosklízených bolo 5 súborov ARC o celkovej veľkosti 82.3 MB. Po spracovaní nástrojom *ARCWB* bolo do repozitára *arcrepos* vložených 3,766 doku-

4. http://isap.vlada.cz/Dul/radaevr.nsf/24b80e5e18d2086d80256dd500544b4f/30194e110600b8a0c125632700727572?OpenDocument

mentov sídliacich dohromady na dvanástich *hostoch*[5].

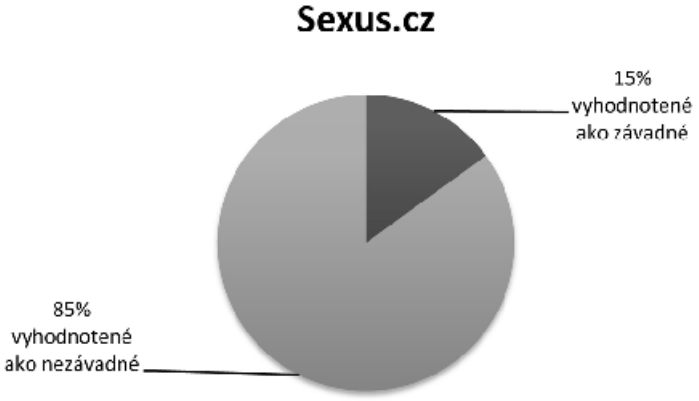
Hosts a počet dokumentov boli zistení obdobne ako u domén *freefoto.cz* a *vlada.cz* SQL dotazom.

U sklizne domény *sexus.cz* bolo zistených 42% nesprávnych mime-type (1587 záznamov). Trvanie evaluácie bolo 2.8 minúty na hardwarovej konfigurácii Intel Core2 Duo T5500 1.66GHz, pamäť 1024MB RAM. Priemerný čas spracovania jedného záznamu bol 0,05 s. Hodnoty a počty ohodnotení sú zhrnuté v tabuľke 6.3.

0: 3198	**1:** 19	**2:** 20	**3:** 15	**4:** 26	**5:** 36	**6:** 37	=	3351
7: 46	**8:** 56	**9:** 58	**10:** 255				=	415

Tabuľka 6.3: Hodnoty a počty ohodnotení *sexus.cz*

Po vyfiltrovaní textových záznamov od binárnych podobne ako u domén *freefoto.cz* a *vlada.cz* bolo zistených 2,638 textových záznamov, čo je 70%. Z týchto záznamov boli vybrané len záznamy s ohodnotením rank väčším ako 6 – uvažované v 3.3. Počet týchto záznamov bol 415, čo je 15%, takže závadnosť sa oproti predpokladu nelíšila. Najvyšší dosiahnutý testovací neškálovaný rank bol 1.91487145398605e+016[6].



Obrázok 6.3: Percentuálna závadnosť *sexus.cz*

5. www.sexus.cz, penis.sexus.cz, materstvi.sexus.cz, sexus.cz, slovnik.sexus.cz, poradna.sexus.cz, bondage.sexus.cz, coitus-per-os.sexus.cz, blog.sexus.cz, masturbace.sexus.cz, antikoncepce.sexus.cz, amateri.sexus.cz
6. http://www.sexus.cz/clanek/1347-zeny-a-orgasmus-2-dil-jak-na-to.html

6.4 Porovnanie a zhrnutie

Doména	**Freefoto.cz**	**Vlada.cz**	**Sexus.cz**
Maximálne zosklízeno	2 GB	2 GB	2 GB
Reálne zosklízeno	1.86 GB	1.37 GB	82.3 MB
Počet ARC súborov	20	21	5
Počet záznamov	10,145	37,935	3,766
Počet použiteľných	247	8,953	2,638
% použiteľných	2.5	23	70
% nesprávnych mime-type	51	39	42
Čas trvania	5.2 min	48 min	2.8 min
Priemerný čas na záznam	0.03 s	0.08 s	0.05 s
% závadných odhad	95–100	0–2	15
% závadných skutočnosť	65	0.7	15
% rozdiel závadných	-30	0	0

Tabuľka 6.4: Zhrnutie výsledkov

Priemerný čas na záznam	0.053 s
Priemerný % nesprávnych mime-type	44

Tabuľka 6.5: Výsledková tabuľka

Z výsledkovej tabuľky vyplýva, že program vyhodnocuje záznamy prijateľne rýchlo (1132 záznamov za minútu, približne 50 miliónov záznamov mesačne), ale je nakonfigurovaný nevyrovnane, lebo u domény *freefoto.cz* nevyhodnotil 30% záznamov ako *nevhodných*, zatiaľ čo u domény *vlada.cz* vyhodnotil ako záznamy *závadné* aj záznamy ktoré očividne závadné nie sú, viď napr. najvyšší rank 6.2.2. Týchto záznamov bolo však prijateľné percento – 0.7. V konečnom dôsledku je teda zrejmé, že je nástroj nakonfigurovaný príliš *jemne*. Pripadá do úvahy možnosť automatického navrhovania celej domény do *Blacklistu* 4.3.6 po prekročení určitej hranice percentuálnej závadnosti napr. 85%.

Kapitola 7

Projektové riadenie

7.1 Inštalácia

7.1.1 Získanie ImpEval

DVD – na dodanom DVD sa nachádza kópia programu popisovaná v tejto práci. Obsah DVD je popísaný v prílohe D.

SVN Checkout – aktuálna verzia môže byť stiahnutá z SVN repozitára na `http://my-svn.assembla.com/svn/impeval`.

7.1.2 Inštalácia

Splnenie požiadavok na systém – je potreba mať nainštalovaný server *Apache Tomcat*[1] a databázu *MySQL*[2].

Vytvorenie databázového modelu – Nad databázou je potreba spustiť inštalačný SQL súbor z *ARCWB*[3], ktorý vytvorí databázový model *arcrepos*.

Úpravenie arcrepos – Nad týmto databázovým modelom je potreba spustiť inštalačný SQL súbor *docs.sql* z *ImpEval* (dostupný z adresára *install*).

Naplnenie dátami – Rozbaliť nástroj *ARCWB* na server (inštalácia popísaná v [16, strana 51]) a spustiť bežca *ArcInsert* nad ARC súbormi. Tie je možné buď zosklízet nástrojom *Heritrix*[4], alebo použiť ARC súbory dodané na DVD (D) s touto prácou.

Evaluovanie – Rozbaliť na server nástroj *ImpEval* z WAR súboru v adresári *war*, nakonfigurovať ho podľa 7.2 a spustiť. .

---

1. `http://tomcat.apache.org/`
2. `http://www.mysql.com/`
3. `http://sourceforge.net/projects/arcwayback/`
4. `http://crawler.archive.org/`

7.2 Konfigurácia

blacklist.properties – zoznam *nevhodných* hosts;

evaluatorservlet.properties – cesta k ARC repozitáru;

isbig.properties – maximálna dovolená veľkosť na záznam;

isvalid.properties – id *future delete*;

keywords.properties – zoznam ohodnotených kľúčových slov.

limits.properties – jednotlive hranice škálovania ohodnotenia;

mime_mark.properties – zoznam mime-typov značkovacích jazykov;

mime_plain.properties – zoznam mime-typov holých textov;

stopwords.properties – zoznam stop slov;

whitelist.properties – zoznam *vhodných* hosts;

7.3 Vývoj

Táto sekcia je nad rámec, ale môže pomôcť, poprípade motivovať v ďalšom vývoji aplikácie.

SVN/Trac – aplikácia používa Trac[5] s integrovaným SVN. SVN repozitár je dostupný na `http://my-svn.assembla.com/svn/impeval` a Trac na `http://my-trac.assembla.com/impeval`.

JavaDoc – je vyvíjaná snaha o úplné popísanie aplikácie.

LGPL – aplikácie je distribuovaná pod licenciou GNU LGPL[6].

5. `http://trac.edgewall.org/`
6. GNU Lesser General Public License – `http://www.gnu.org/licenses/lgpl.html`

Záver

Záverom by som chcel zhrnúť výsledok práce a vyhodnotenia nástroja *ImpEval*, ktorým by som chcel priniesť úžitok do evaluácie vhodnosti záznamov vo webovom archíve. Ukážková evaluácia sa ukázala ako vcelku odpovedajúca predpokladom na ňu kladeným, ale určite existuje nutnosť nakonfigurovať nástroj ekvivalentnejšie úrovni nevhodných záznamov, aby ich dokázal presnejšie odhaliť medzi záznamami vhodnými.

Ďalej sa budem zaoberať hlavne možnými rozšíreniami programu. Tento projekt je zaujímavý, prospešný a potrebný, ale i komplexný a komplikovaný v natoľkej miere, ktorá by ďaleko presiahla rozsah tejto práce. Takže je zrejmé, že sa nachádza ešte len na svojom začiatku, preto som sa tomu snažil podriadiť celú architektúru nástroja, hlavne rozšíritelným modulárnym systémom a projektovým riadením od začiatku vývoja. Navrhoval by som do nástroja začleniť modul na fulltextové vyhľadávanie v obsahu záznamu (využívajúci napr. nástroj *Apache Lucene*[7]), poprípade modulárny systém upraviť do možnosti *real-time* vypínať a zapínať moduly podľa potreby – takto by sa mohlo podľa povahy záznamu evaluovať buď modulom *Keywords*, alebo *Fulltext*. Je možné takisto zaviesť do modulu *Keywords* funkcionalitu na skloňovanie kľúčových slov pre lepšie výsledky vo vyhľadávaní nad slovami v stránke. Ďalej by boli vhodné moduly na vyhodnocovanie netextových záznamov, primárne PDF, PS, RTF a binárne záznamy ako obrázky a videá. Na vyhľadávanie v obrázkoch existujú nástroje[8] vyhodnocujúce podobnosť, takže nie je vylúčená možnosť použiť ich. Celú problematiku pokročilých technológii vyhľadávania zastrešuje projekt CHORUS[9], ktorý by mohol byť dobrým zdrojom teoretickej a znalostnej báze. Určite je ale potrebné v budúcnosti naimplementovať triedu *WARCEntry* pracujúcu s archívnym súborom WARC, pre ktorú som pripravil rozhranie *IEntry*.

Projekt je možné v širšom ponímaní rozšíriť nielen na evaluáciu závadnosti v závadných kategóriach, ale i na evaluáciu v nezávadných ka-

7. `http://lucene.apache.org/`
8. `http://mufin.fi.muni.cz/imgsearch/`
9. `http://www.ist-chorus.org/`

tegóriach – udržiaval by si sémantický popis dát, čím by sa zároveň stal by sa z neho istý druh meta popisovača webového archívu použiteľný aj pre iné účely. Je možné takisto vytvoriť administračné rozhranie, kde by sa editovali autorizovanými osobami záznamy v *blackliste*, *whiteliste*, kľúčové slová. Snažil som sa týmto ďalším rozšíreniam pripraviť podklad, na ktorom je možné ďalej budovať tento nástroj – možností je mnoho a myslím, že sú to veľmi interesantné, užitočné a významné rozšírenia.

Literatúra

[1] Internet Archive ARC files. [online]. Available from World Wide Web: http://crawler.archive.org/articles/developer_manual/arcs.html. [cit. 2008-12-20].

[2] WebArchiv - archiv českého webu. [online]. Available from World Wide Web: http://www.webarchiv.cz/. [cit. 2008-12-20].

[3] Information and documentation of The WARC File Format. [online], 2006. Available from World Wide Web: http://archive-access.sourceforge.net/warc/WARC_ISO_28500_final_draft%20v018%20Zentveld%20080618.doc. [cit. 2008-12-20].

[4] Java Mime Magic Library. [online], 2008. Available from World Wide Web: http://sourceforge.net/projects/jmimemagic/. [cit. 2008-12-20].

[5] Understanding Extension Class Loading. [online], c1995–2008. Available from World Wide Web: http://java.sun.com/docs/books/tutorial/ext/basics/load.html. [cit. 2008-2-14].

[6] Guidelines for consumers in choosing an Internet filter. [online], c1996–2009. Available from World Wide Web: http://www.iwf.org.uk/public/page.28.33.htm. [cit. 2005-11-17].

[7] Apache Cayenne – object Relational Mapping, Persistence and Caching for Java. [online], c2001-2008. Available from World Wide Web: http://cayenne.apache.org/. [cit. 2008-12-20].

[8] Class Logger. [online], c2004. Available from World Wide Web: http://java.sun.com/j2se/1.5.0/docs/api/java/util/logging/Logger.html. [cit. 2008-12-20].

[9] Class Resource Bundle. [online], c2004. Available from World Wide Web: http://java.sun.com/j2se/1.5.0/docs/api/java/util/ResourceBundle.html. [cit. 2008-12-20].

[10] The Internet Archive Wayback Machine. [online], c2005–2008. Available from World Wide Web: `http://archive-access.sourceforge.net/projects/wayback/`. [cit. 2008-12-20].

[11] Class MimetypesFileTypeMap. [online], c2008. Available from World Wide Web: `java.sun.com/javase/6/docs/api/javax/activation/MimetypesFileTypeMap.html`. [cit. 2008-12-20].

[12] S. Brin and L. Page. The Anatomy of a Large–Scale Hypertextual Web Search Engine. [online]. Available from World Wide Web: `http://infolab.stanford.edu/~backrub/google.html`. [cit. 2008-12-20].

[13] A. Brokeš. Projekt Webarchiv - archiv českého webu. *Zpravodaj ÚVT MU*, roč. XVIII(4):s. 10–13, 2008. Available from World Wide Web: `http://www.ics.muni.cz/zpravodaj/articles/578.html`.

[14] R. Gagnon. Get the Mime Type from a File. [online], c1998–2007. Available from World Wide Web: `http://www.rgagnon.com/javadetails/java-0487.html`. [cit. 2008-12-20].

[15] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, Michigan, 2004. 395 s.

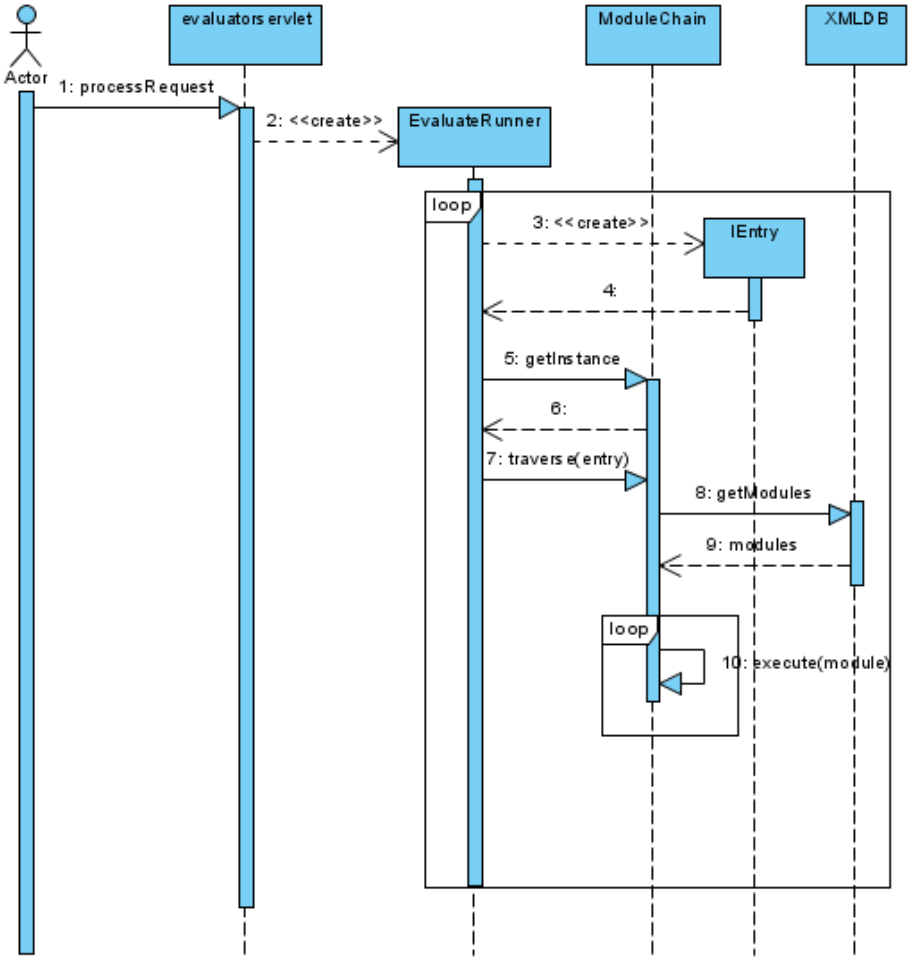
[16] L. Matějka. Zpřístupnění archivu českého webu. Master's thesis, Masarykova Univerzita, Fakulta Informatiky, 2006.

[17] M. Nič and J. Jirát. Xpath tutorial. [online], c2000. Available from World Wide Web: `http://www.zvon.org/xxl/XPathTutorial/General_cze/examples.html`. [cit. 2008-12-20].

[18] D. Tapscott. *Growing Up Digital : The Rise of the Net Generation*. McGraw-Hill, New York, 1998. 338 s.

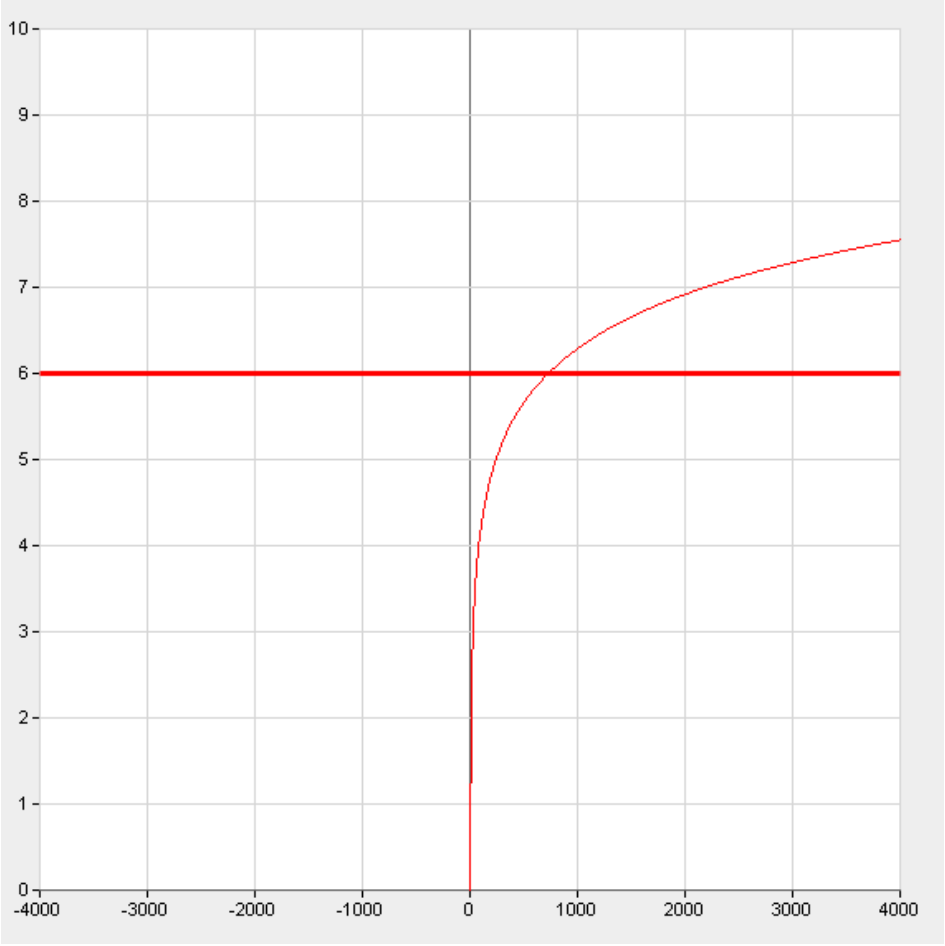
Dodatok A

Sekvenčný diagram systému



Dodatok B

Logaritmická funkcia pre škálovanie ohodnotení



Táto funkcia neodpovedá presne funkcii použitej v programe, iba demonštruje škálovanie. Na vertikálnej osi sú škálované *ranky*, na horizontálnej osi neškálované. Horizontálna tučná línia pri čísle 6 znamená hranicu *nevhodnosti*.

Dodatok C

Stop slová

C.1 České stop slová

a a sice a to aby aj ale ani aniž ano asi až bez bude budem budeš by byl byla byli bylo být co což cz či článek článku články další dnes do ho i já jak jako je jeho jej její jejich jen jenž ještě ji jiné již jsem jseš jsme jsou jšte k kam každý kde kdo když ke která které kterou který kteři ku ma máte mezi mi me mě mně mnou mít můj může my na ná nad nám napište náš naši ne nebo nechť nejsou není než nic ní nové nový o od ode on pak po pod podle pokud pouze práve pro proč proto protože první před přede přes při pta re s se si strana své svůj svých svým svými ta tak také takže tato tedy těma te tě ten tento této tím tímto tipy to tohle toho tohoto tom tomto tomuto toto tu tuto tvůj ty tyto u už v vám váš vaše ve více však všechen vy z za zda zde ze zpět zprávy že

Zdroj: `http://seo-servis.cz/libs/stopwords.txt.cz`

C.2 Slovenské stop slová

a a síce a to aby aj ale ani že by áno asi až bez bude budem budeš by bol bola boli bolo byť čo či článok článku články další dnes do ho i ja ak ako je jeho jej ich náš len lenže ešte iné už som si sme sú ešte k kam každý kde kdo keď ku ktorá ktoré ktorú ktorý ktorí ma máte medzi mi me mňe mnou mať môj môže my na ná nad nám napíšte náš naši ne lebo alebo nech nisú neni než nič nové nový o od odo on potom po poď podľa pokiaľ iba práve pre prečo preto pretože prvý prví pred predo cey pri s so si strana svoje svoj svojich svojím svojími ta tak tiež takže táto vtedy t?my ťa t? ten tento tieto tým týmto tipy to toto toho tohoto tomu tomuto tu tuto tvoj ty tieto u už v vám váš vaše vo viac viacej však všetok vy z za že správy že

Zdroj: `http://seo-servis.cz/libs/stopwords.txt.sk`

Dodatok D

Obsah DVD

Priložené DVD má nasledovnú adresárovú štruktúru:

arcs – ARC súbory použité pri ukážkovej evaluácii.

install – inštalačný SQL skript.

javadoc – dokumentácia k *ImpEval*.

src – zdrojové kódy k *ImpEval*.

thesis – táto práca.

war – WAR súbor obsahujúci nástroj *ImpEval*.