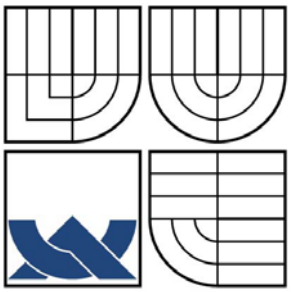
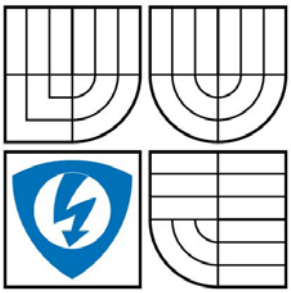


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKACNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

EFEKTIVNÍ VYHLEDÁVÁNÍ DAT NA INTERNETU

EFECTIVE DATA SEARCHING ON THE INTERNET

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — BC. MILOŠ TLUSŤÁK
AUTHOR

VEDOUCÍ PRÁCE — ING. PETRA LAMBERTOVÁ
SUPERVISOR

BRNO 2008

ZADANI

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Bc. Miloš Tlusťák

Bytem: Palackého 693, Bystřice pod Hostýnem 76861

Narozen/a (datum a místo): 13.01.1984 v Přílepech

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

Prof. Ing. Kamil Vrba, CSc.

(dále jen „nabyvatel“)

Čl. 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - □ disertační práce
   - □ diplomová práce
   - □ bakalářská práce
   - □ jiná práce, jejíž druh je specifikován jako .....

   (dále jen VŠKP nebo dílo)

Název VŠKP: Efektivní vyhledávání dat na internetu

Vedoucí/ školitel VŠKP: Ing. Petra Lambertová

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v*:

- □ tištěné formě – počet exemplářů 2
- □ elektronické formě – počet exemplářů 1

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - □ ihned po uzavření této smlouvy
   - □ 1 rok po uzavření této smlouvy
   - □ 3 roky po uzavření této smlouvy
   - □ 5 let po uzavření této smlouvy
   - □ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: …….

…….. ……
Nabyvatel Autor

ANOTACE

Tato práce se zabývá aktuální problematikou vyhledávání dat na internetu. Je zaměřena na objasnění vlastností a principů fungování centralizovaného, decentralizovaného a hybridního vyhledávání. Rozebírá výhody a nevýhody jednotlivých druhů vyhledávání. Popisuje nejpoužívanější vyhledávací stroje a jimi používanou syntaxi. Mapuje možnosti využití inteligentních vyhledávacích agentů.

Praktická část této práce je věnována tvorbě a testování vyhledávací aplikace. Ta umožňuje uživateli vyhledávat a stahovat kontakty firem podle zadaného vyhledávacího výrazu a ze získaných dat následně vytvářet databáze kontaktů.

Klíčová slova: vyhledávání, vyhledavače, P2P, vyhledávací aplikace

ABSTRAKT

This master´s thesis is concerned in problematic of data searching on the Internet which is very actual. It focuses mainly how centralized, decentralized and hybrid searching work. This paper also compares advantages and disadvantages of these methods and describes principles and syntaxes of most-used search engines. Possibilities of use intelligent search agents are shown here as well.

The practical part of this thesis is dedicated to development and testing of specialized application for searching. This program enable user to harvest contacts of companies in accordance to defined key-words and creates database file from obtained contacts.

Keywords: searching, searching engines, peer-to-peer, application for searching

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma Efektivní vyhledávání dat na internetu jsem vypracoval samostatně pod vedením vedoucí diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného semestrálního projektu dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Petře Lambertové, za velmi užitečnou metodickou pomoc a cenné rady při zpracování bakalářské práce.

V Brně dne …… ……..
(podpis autora)

Obsah

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek a pojmů

CSV	- formát souboru pro jednoduché vytvoření tabulky pomocí textového souboru, kde jeden řádek textu odpovídá jednomu řádku v tabulce, pro rozdělení řádku na sloupce se používají například čárky nebo středníky nebo tabulátory (Comma Separated Values)
Dampening faktor	- nabývá hodnot od nuly do jedné, čím nižší, tím vzorec rychleji konverguje
DOC	- velice známý a rozšířený formát dokumentů vytvořených programem Microsoft Word
Flooding	- způsob propagace dotazů
GTPR	- číslo představující aproximovanou hodnotu PageRanku (Google Toolbar PageRank)
HTML	- základní značkovací jazyk, ve kterém jsou psány internetové stránky a jenž popisuje jejich vzhled (HyperText Markup Language)
HTTP	- hypertextový přenosový protokol (HyperText Transfer Protocol)
IP	- protokol používaný v počítačových sítích (Internet Protocol)
MP3	- komprimovaný zvukový soubor (Motion Picture experts group - layer 3)
Off-page faktor	- faktor, který nelze ovlivňovat přímo na vytvářecí stránce
On-page faktor	- faktor, který lze ovlivňovat přímo na vytvářecí stránce
P2P	- počítačová síť fungující na principu rovnocenné účasti jednotlivých počítačů namísto využívání serverů (Peer-to-Peer)
PageRang	- interní hodnota kvality stránek, které obsahuje index vyhledavače Google
PDF	- univerzální formát souboru od společnosti Adobe, bývá používán například pro distribuci elektronických manuálů, dokumentace a dalších typů dokumentů (Portable Dokument Format)
PPT	- formát souboru pro prezentace vytvořené programem Microsoft PowerPoint
PS	- formát souboru, jenž je jedním ze standardů používaných pro tisk a pro vykreslování složitých obrazů, textu, grafiky, atd. (PostScript)
RDF	- standard pro modelování informací v různých syntaxích (Ressource Description Framework)
RTF	- formát souboru pro textové dokumenty, kde může být použito rozmanité formátování textu (i když v menší míře než u formátu DOC), je všeobecně uznávaný a díky své univerzálnosti bývá používán

	například pro výměnu dat (textu) mezi programem Microsoft Word a dalšími textovými editory [2]
URI	- řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci zdroje informací (Uniform Resource Identifier)
URL	- řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění zdrojů informací (Uniform Resource Locator)
XML	- je obecný značkovací jazyk umožňující snadné vytváření konkrétních značkovacích jazyků pro různé účely (eXtensible Markup Language)
XLS	- formát souboru pro tabulky vytvořené programem Microsoft Excel
WWW	- celosvětová síť postavená na komunikaci přes hypertextové odkazy (World Wide Web)

1 Úvod

Tato práce se zabývá aktuální problematikou vyhledávání dat na internetu. Je zaměřena na objasnění vlastností a principů fungování centralizovaného, decentralizovaného a hybridního vyhledávání. Rozebírá výhody a nevýhody jednotlivých druhů vyhledávání. Popisuje nejpoužívanější vyhledávací stroje a jimi používanou syntaxi. Mapuje možnosti využití inteligentních vyhledávacích agentů. Nastiňuje tvorbu vlastní vyhledávací aplikace, popisuje aplikační prostředí a práci této aplikace.

Internet se stal největší zásobárnou informací na světě. V takto velkém množství informací je proto velmi obtížné se orientovat, a proto velkým problémem internetu je vyhledávání informací. Pro vyhledávání na internetu neexistuje žádný jednotný vyhledávací systém, jako je například manažer souborů pro prohledávání pevných disků či CD/DVD. Právě naopak, existuje mnoho způsobů, jak informace na internetu vyhledávat. Zásadní otázkou tedy je, který a zdali vůbec některý z těchto způsobů je optimální.

Na tuto otázku lze těžko najít jednoznačnou odpověď, jelikož existuje velké množství různých kritérií, jejichž optimální varianty si navzájem odporují. Jako příklad lze uvést knihu. Uživatel, který potřebuje nalézt informaci v knize, má dvě možnosti, jak to udělat. Jako první možnost se nabízí přečtení obsahu, který ale obsahuje pouze názvy kapitol a hledaná informace v nich může být zahrnuta, ale také nemusí (viz kapitola 2). Druhou možností je přečtení celé knihy a hledání veškerých zmínek o hledané informaci včetně „čtení mezi řádky“. Po zhodnocení obou možností se nabízí, že první možnost je relativně rychlá, avšak kvalitativní úroveň nízká. Naproti tomu druhá možnost poskytuje vyčerpávající výčet všeho nalezeného za cenu relativně pomalé rychlosti vyhledání (viz kapitola 3).

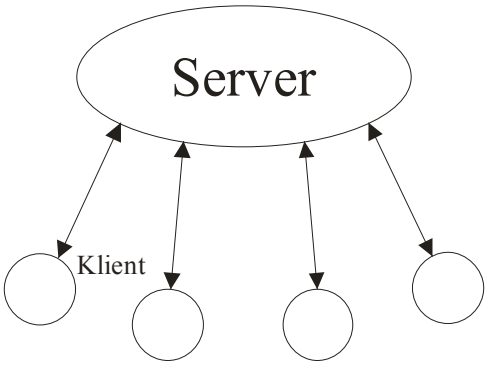
Každá z možností je tedy optimální variantou k různému upřednostňovanému kritériu výsledků vyhledávání. Ideální by tedy byla kombinace obou možností (viz kapitola 4).

Odpověď na otázku, kam se budou ubírat technologie vyhledávání dat na internetu, může napovědět kapitola 5.

V dnešní době se pro efektivní vyhledávání dat hojně využívají specializované vyhledávací aplikace. Možnost usnadnění práce a hlavně úspory času tedy představuje vytvoření vyhledávací aplikace v praktické části této diplomové práce (viz kapitola 6).

2 Centralizované vyhledávače

Centralizované jsou nazývány proto, jelikož samotné vyhledávání probíhá v centrálním indexu nebo v centrálním katalogu. Tyto vyhledavače jsou založeny na modelu Klient-Server (Obr. 1). Tento model definuje server jako entitu, která poskytuje službu a klienta jako entitu, která „konzumuje“ službu.



Obr. 1: Schéma modelu Klient-Server

Hlavní výhodou tohoto modelu je skutečnost, že informace obsažené na serveru existují pouze v jedné kopii a tím pádem se snadno zálohují a udržují. Z toho také plyne patřičná rychlost. Jelikož server má všechna data k dispozici na jednom místě (nikoliv roztroušená po síti jako u Peer-to-Peer modelu – viz kap. 3) a přesně ví, kde má potřebná data uložena.

Podle způsobu uchování a prezentace dat lze centralizované vyhledávače rozdělit na katalogové, indexové (fulltextové) a hybridní.

2.1 Katalogové vyhledávací stroje

Pokud budeme vycházet z historie, tak v dávných dobách, kdy na internetu bylo jen pár stránek a téměř všichni uživatelé internetu se snad znali osobně, stačilo při hledání nějaké informace naťukat do prohlížeče adresu příslušných stránek, na kterých jste tušili přítomnost požadované informace. Postupem doby stránek přibývalo geometrickou řadou a objevila se potřeba stránky přehledně uspořádat. Po vzoru papírových katalogů tak vznikaly první elektronické katalogy, které řadily stránky do kategorií a to buď tématicky, oborově či místně členěných.

Katalogové vyhledávací stroje obsahují velkou databázi odkazů do celého internetu uspořádanou hierarchicky a logickým způsobem do stromu. Tyto vyhledávače nám umožňují procházet stromovou strukturu od rozsáhlých témat až po specifické „předměty“, které jsou vlastně námi hledaným výsledkem (proto též předmětové katalogy). Také nám umožňují prohlížení odkazů, vyhledání odkazů podle určitého slova a v neposlední řadě přidávání odkazů do databáze. Nejznámějšími katalogovými vyhledávači u nás byly Seznam, Atlas a Centrum. Dnes již jsou všechny tři zmíněné portály vyhledávači

hybridními s možností nastavení volby vyhledávání mezi fulltextem a katalogem. V dnešní době katalogy již jen doplňují fulltextové vyhledávání.

O nových stránkách se katalogy dozvídají buď přímo od autorů stránek, kteří v katalogu své stránky zaregistrují a odkaz je poté schvalován pracovníky katalogu, kteří zároveň posuzují jeho vhodnost pro danou kategorii, nebo manuálním či automatickým sběrem dat. Manuální získávání informací o stránkách je pro provozovatele katalogu náročnější na finance, neboť se musí zaplatit práce lidí, kteří ručně vyhledávají nové stránky. Automatický sběr dat je na robotovi (malém počítačovém prográmku), který prochází internet a hledá stránky, které ještě nejsou zanesené v katalogu. I v tomto případě pak záleží na člověku, který musí vyhledanou stránku ještě zařadit do příslušné kategorie a odkaz na ni opatřit krátkým popisem.

2.1.1 DMOZ

Katalog DMOZ (někdy také Open Directory Project) je klasický katalogový vyhledávač, kde jsou odkazy zařazovány do tématických a regionálních kategorií. Tento katalog je také největší lidmi editovaný mezinárodní katalog stránek na internetu, na kterém pracují převážně dobrovolníci z řad profesionálů v dané oblasti, což zajišťuje velmi vysokou úroveň zatříděných odkazů. Jde o práci neplacenou a věnují se jí ve svém volném čase. Cílem není vytvořit katalog všech stránek na internetu, ale spíše výběr toho nejlepšího, proto se editoři snaží zařazovat jen kvalitní a informačně hodnotné stránky.

DMOZ je neziskový projekt. Tím se odlišuje od většiny katalogů a staví ho do pozice významného zdroje informací o internetových stránkách, který je neúplatný a odkaz, který se v něm vyskytne, je tedy skutečně kvalitní. V katalogu DMOZ jsou si všechny stránky rovny, nikdo není zvýhodňován ani znevýhodňován a díky tomu v něm uživatel nalezne to, co hledá a ne to, co si přejí obchodníci a propagátoři.

Při hledání v komerčním katalogu jsou na prvních místech uživateli zobrazeny odkazy, za které si jejich majitel zaplatil a teprve na dalších pozicích (někdy až na dalších stránkách) nalezne odkazy, jejichž majitel neplatí portálu žádné poplatky za lukrativnější zobrazení svého obsahu.

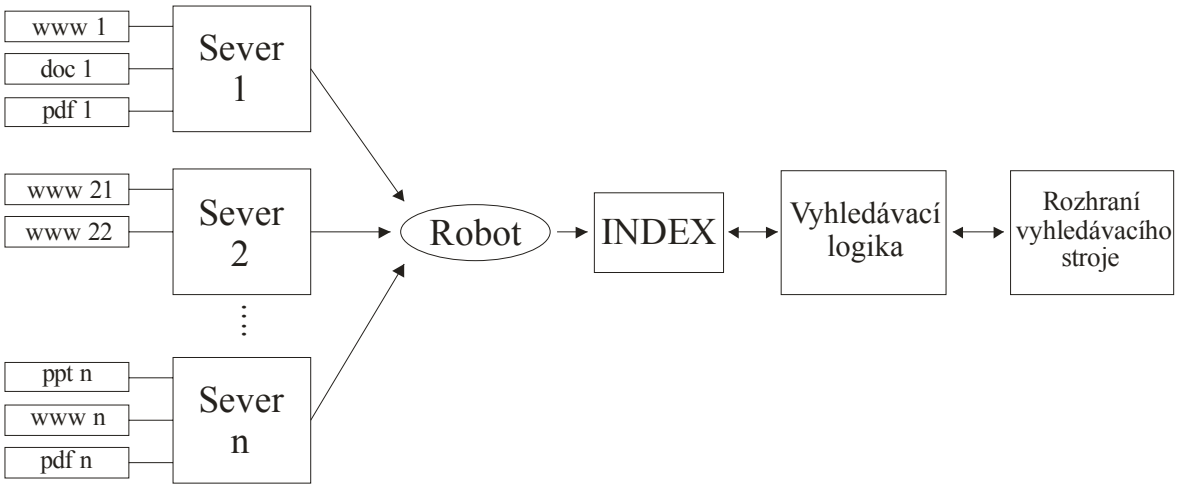
2.2 Fulltextové vyhledávací stroje

Tyto vyhledávače představují sofistikovaný způsob, jak v obrovském množství webových stránek najít na internetu to, co uživatel hledá. Protože se informace, které jej zajímají, nacházejí především v textu webových stránek, procházejí fulltextové vyhledávače celý jejich text, nikoli pouze odkazy (URL) či titulky (nadpisy) jednotlivých stránek.

Jak již bylo naznačeno výše, fulltextové vyhledávací stroje jsou založeny na prohledávání centrálního indexu (databázi), který obsahuje extrahované a upravené informace o robotem stažených webových stránkách. Tedy aby mohl vyhledávač stahovat informace a kontrolovat jejich aktuálnost, spolupracuje s jiným programem, tzv. indexovacím robotem. Ten na základě analýzy získaných informací zjišťuje změny, které byly na stránce od předchozí kontroly provedeny, vyhodnocuje je a případně provádí reindexaci.

To je časově velmi náročný úkol, protože objem dat v indexu neustále roste a indexovací robot si vlastně svou činností přidělává stále více práce. Jelikož ovšem převážná část zaindexovaných stránek zůstává relativně neměnná, poskytují vyhledávače tohoto typu velmi kvalitní služby.

Obr. 2 zobrazuje zjednodušenou strukturu fulltextového vyhledávače a princip činnosti indexovacího robota neustále stahujícího data z internetu. Tato data poté tvoří index nad kterým probíhá vyhledávání.



Obr. 2: Zjednodušený princip fulltextového vyhledavače

Průběh indexace lze popsat tak, že robot vyhledávače prochází zdrojový html kód stránky. Pokud v něm nalezne odkaz na další stránku, přeskočí na ni a v indexaci pokračuje tam. Do fulltextových vyhledávačů je možné stránky registrovat i manuálně, většinou jsou ale preferovány (lépe se umísťují ve výsledcích vyhledávání) ty stránky, které vyhledávač našel sám, přirozenou cestou, tzn. pomocí odkazu z jiného webu.

Robot při indexaci zdrojový kód stránky zpracovává a ukládá do databáze na serveru vyhledávače. Tak tedy vzniká centrální index jednotlivých vyhledavačů. Vyhledávací robot si pamatuje, na kterých stránkách už byl, a jeho program určuje, kdy se na stránku opět vrátí, aby zaindexoval případné změny. Chování robotů je plně automatické a nelze je téměř nijak ovlivnit, např. robotovi přikázat, jak často má naši stránku navštěvovat. Robotům je možné pouze zakázat indexování stránek, u kterých nechceme, aby se objevovaly ve výsledcích vyhledávání.

Důležitou roli při vyhledávání hraje klíčové slovo (keyword) neboli vyhledávací výraz, kterým se uživatel snaží vyjádřit svoji informační potřebu. Jako klíčové slovo lze použít jednotlivé slovo, část slova nebo frázi. Z jednoho nebo více vyhledávacích výrazů je pak složen dotaz (query), který reprezentuje úplný vyhledávací požadavek. Samotné vyhledávání provádí vlastní vyhledávač (search engine). To je program, který přebírá od uživatele vyhledávací dotaz. Vlastní vyhledávání je úkol pro vyhledávací logiku vyhledávače (Obr. 2), která představuje samotný „motor“ vyhledávače. Analyzuje data v indexu a vytváří nad nimi složité datové struktury, které usnadňují vyhledávání podle různých kritérií (viz. kap. 2.4). Toto vyhledávání probíhá jako hledání v obsahu na základě klíčového slova naformulovaného uživatelem. Hledání tedy neprobíhá v reálném čase na internetu, ale na serveru vyhledávače, který nám pak jako odpověď na náš dotaz odešle

do prohlížeče výsledek tohoto hledání. Jen tak je možné, abychom výsledek hledání dostali téměř okamžitě po zadání dotazu.

2.2.1 Google

Robot Google se jmenuje GoogleBot. Jádrem vyhledávače Google je algoritmus PageRank. PageRank byl navržen Larry Pagem a Sergeyem Brinem a slouží pro ohodnocení důležitosti webových stránek. Je to číslo, které si vyhledavač Google přiřazuje ke každé stránce (každému URL), kterou má v indexu, na základě její provázanosti s jinými stránkami. Vyjadřuje věrohodnost (důležitost) stránky. Postupem času tento systém přebíraly další vyhledávače (např. Seznam a Jyxo).

Google si PageRank vypočítává podle toho, kolik a jak důležitých stránek na danou stránku odkazuje (zjednodušeně řečeno). Originální dokumentace Google PageRanku udává vzorec pro výpočet PageRanku www stránky A [1]:

$$PR(A) = (1-d)/m + d * (PR(T_1)/C(T_1) + ... + PR(T_n)/C(T_n)), \qquad (2.1)$$

kde d je dampening faktor (nastavený obvykle na 0,85), m je celkový počet zaindexovaných stránek a $C(T_i)$ je počet odkazů vedoucích ze stránky T_i. Jako vstupní hodnoty $PR(T_i)$ se berou hodnoty PageRanku stránek z minulé iterace výpočtu. Vzoreček po několika iteracích konverguje (čím je nižší d tím rychleji). Hodnoty PageRanku všech stránek se pohybují těsně nad nulou [1].

Lze tedy říci, že stránka předává část svého PageRanku stránkám, na které odkazuje. Čím víc obsahuje odkazů (hodnota $C(T_i)$), tím méně každé stránce „předá“. Tento termín nejspíše není nejvhodnější jelikož stránka o svůj PageRank nepřichází a dochází zde spíše ke „kopírování“. Hodnota PageRanku dané www stránky je tedy přímo závislá na počtu zpětných odkazů ze stránek s co největším PageRankem. Čím má stránka vyšší PageRank, tím bude větší pravděpodobnost, že by mohla být výše postavena ve výsledcích vyhledávání oproti stránkám s menším PageRankem. PageRank ovšem zdaleka není jediné kritérium, které bere rozhodovací logika Googlu v potaz a které určuje výslednou pozici www stránky ve výsledcích vyhledání. Tato kritéria nejsou ovšem veřejně známa.

Google používá PageRank také pro řízení svého robota. Stránky s nízkým PageRankem navštěvuje zřídka. Některá nová URL ani nenavštíví, dokud jejich PageRank nedosáhne určité hodnoty. Zjištění přesné hodnoty PageRank není možné a lze zjistit pouze její aproximovanou hodnotu Google Toolbar PageRank (GTPR), která je zobrazována v aplikaci Google Toolbar při prohlížení stránek v prohlížeči. Rozsah hodnot GTPR se pohybuje na 11 bodové stupnici od 0 do 10. Tyto hodnoty se aktualizují najednou a pouze několikrát do roka. Vztah mezi hodnotou GTPR a hodnotou PageRank není znám. Je pravděpodobně logaritmický a hodnota GTPR šest může znamenat třeba skutečný PageRank = 0,00000008 [1].

Matematicky se na hodnotu PageRanku nějaké stránky můžeme dívat jako na pravděpodobnost, že se uživatel náhodně klikající na odkazy dostane právě na tuto konkrétní stránku. Počítá se rekurzivně podle příchozích a odchozích odkazů na celém internetu [2].

Postupem času tento systém přebíraly další vyhledávače.

2.2.2 Jyxo

Autorem technologie vyhledávání, jež využívá vyhledávač Jyxo, je Michal Illich. Robot Jyxo se jmenuje JyxoBot. Vyhledávač Jyxo, jako vůbec jediný, automaticky vyhledává i synonyma ke slovům obsaženým ve vyhledávacím výrazu.

Mezi JyxoRankem českého vyhledávače Jyxo a PageRankem světového Google existuje jeden podstatný rozdíl. JyxoRank zvýhodňuje zpětné odkazy z cizích domén (domén druhého řádu) oproti zpětným odkazům z vlastního webu. Naproti tomu Google PageRank nedělá rozdíly mezi zpětnými odkazy a přikládá jim stejnou váhu. Vysoký PageRankem tak lehce získávají velké, hustě prolinkované weby se správnou strukturou URL adres. Mají tak výhodu před weby malými. S tímto handicapem se výborně vypořádávají weblogy, které na sebe často a hodně odkazují. Získávají tak mnoho zpětných odkazů a lépe se umísťují ve vyhledávačích.

2.3 Centralizované hybridní vyhledavače

Hybridní neboli smíšené centralizované vyhledávače spojují principy obou typů výše popsaných vyhledávačů. Vzhledem k tomu, že katalogy nemohou nikdy obsáhnout všechny stránky na internetu, většina vyhledavačů začínala jako katalogové vyhledavače a katalogy obsahují cenná ručně editovaná data, snaží se většina vyhledavačů co nejvíce integrovat katalogová data s fulltextem. Proto je v současné době většina vyhledávačů hybridních. Vyhledávání může probíhat jak v katalogu daného vyhledavače tak fulltextovou technologií, kterou vyhledávač používá, nebo v oběma způsoby najednou.

2.3.1 Seznam

Seznam byl založen Ivo Lukačovičem v roce 1996 jako první katalogový vyhledávač v České republice. Dlouhodobě si udržuje pozici nejpoužívanějšího českého vyhledávače na českém internetu a v současné době jeho podíl na vyhledávání činí přibližně 50-60% [3].

Do roku 2005 byl Seznam primárně katalogovým vyhledavačem. Měl kromě katalogu firem také katalog webových stránek. V tomto katalogu se vyhledávalo primárně, jelikož technologie fulltextového vyhledávání tohoto portálu nebyla v té době na příliš dobré úrovni. Až ve zmíněném roce 2005 přišla radikální změna a Seznam spustil jako výchozí vyhledávání svou novou fulltextovou technologii. Pro české vyhledávání používá Seznam vlastní fulltextovou technologii, pro vyhledávání ve světě přebírá výsledky od Googlu. Jelikož je Seznam hybridní vyhledávač umožňuje vedle fulltextového vyhledávání hledat také v katalozích. Stačí, když nad vyhledávacím polem místo záložky Internet bude vybrána záložka Firmy, a místo na internetu proběhne vyhledávání v katalogů firem. Tento katalog firem nyní funguje i na samostatné doméně firmy.cz.

Autorem fulltextové technologie Seznamu je Dušan Janovský. Jak sám přiznává, při tvorbě se nechal značně inspirovat technologií jež využívá Google. Robot Seznamu se jmenuje SeznamBot. Podobně, jako Google využívá PageRang, fulltext Seznamu využívá algoritmus S-rank. S-rank stránky je veličina, která by měla vyjadřovat důležitost

každé stránky na českém webu. Počítá se zejména z odkazové sítě algoritmem, který zohledňuje jednak odkazy, které na stránku míří, ale i to, kam ze stránky odkazy vedou. Přesný výpočet S-ranku není veřejný. Je jen nastíněno, že S-rank se počítá váženou nelineární kombinací různých veličin, v nichž výrazně převažují off-page faktory nad on-page faktory. Výpočet hlavního zdroje S-ranku se podobá známému algoritmu Hubs & Authorities, ale je upraven tak, aby dával smysl i pro netématické množiny stránek.

2.3.2 Centrum

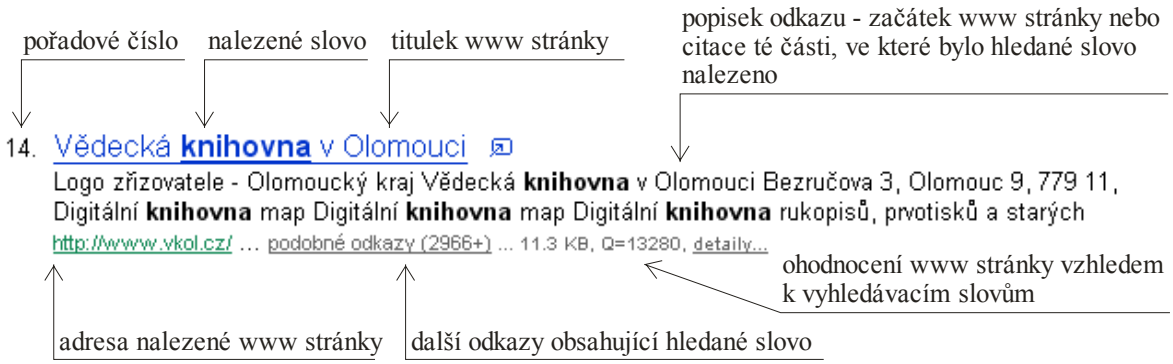
Fulltextové vyhledávání je na portálu Centrum.cz realizováno vyhledávací technologií Morfeo specializovanou na český jazyk. Tato technologie je vyvíjena samotným provozovatelem portálu Centrum ve spolupráci s akademickou sférou. Robot Morfea se jmenuje Holmes a své „pátrání“ začíná na adresách registrovaných v katalogu portálu Centrum.cz a adresách typu www.<jméno>.cz získaných z databáze domén.

Morfeo umí skloňovat a časovat hledané výrazy, nabízí možnost vyhledávání slova včetně jeho synonym a upozorňuje na nesprávně zadaná slova - překlepy, gramatické chyby apod. Klade také velký důraz na aktuálnost dat (denně přibývají do databáze nové články). Dokáže efektivně vyhledávat ve stránkách, které často mění svůj obsah. Například ve zpravodajských serverech, v blozích, ale i ve firemních www stránkách. Morfeo velmi často aktualizuje i zřídka se měnící stránky či stránky se statickým obsahem. Podle oficiálních informací uváděných v nápovědě tohoto vyhledávače, není žádná ze stránek v databázi vyhledávače starší než 28 dní [4]. Tyto informace ovšem nejsou zcela přesné, jelikož náhodným vyhledáváním byly nalezeny i stránky, jejichž indexace byla až 3 měsíce stará.

Na Obr. 3 je ukázka výpisu nalezených stránek, při fulltextovém vyhledávání realizovaném vyhledávací technologií Morfeo. Tento výpis se skládá z několika částí:

- *Pořadové číslo* – odpovídá pořadí daného výsledku ve všech nalezených výsledcích
- *Titulek www stránky* – ten je zároveň i odkazem na vyhledanou stránku. Text titulku je přebírán z titulku vyhledané stránky. Přesněji řečeno je to text uvedený ve zdrojovém kódu dané stránky mezi dvěma značkami (tagy) **title**. Pokud stránka neobsahuje tyto značky, je namísto titulku dosazena adresa nalezené stránky.
- *Popisek odkazu* – pod titulkem se nachází popisný text. Většina vyhledávačů má tendenci jej převzít přímo z popisu stránky. Tedy textu uvedeného ve zdrojovém kódu za značkou **meta**. Pokud ovšem tento text neobsahuje hledaný výraz nebo pokud daná stránka nedisponuje touto značkou, zobrazují se v popisku útržky textu samotné stránky, v nichž se hledaný výraz vyskytuje.
- *Adresa URL* – na posledním řádku každého výsledku se nacházejí technické informace. První z nich je adresa URL nalezeného výsledku.

- *Podobné odkazy* – pokud je nalezená stránka pro uživatele užitečná a chtěl by vyhledat odkazy na podobné stránky nebo na ty s podobným obsahem, stačí kliknout na tento odkaz.
- *Velikost stránky* – na tomto místě se uživatel dozvídá velikost odkazované stránky. Tato informace je důležitá nejen pro uživatele s pomalejším připojením, ale také z toho důvodu, že některé odkazy neodkazují pouze na webové stránky (tedy povětšinou soubory HTML malých velikostí), ale taký jiné datové typy (DOC, PDF, atd.), které již mohou nabývat mnohem větších velikostí.



Obr. 3: Ukázka výpisu nalezených stránek, při fulltextovém vyhledávání realizovaném vyhledávací technologií Morfeo [5]

2.3.3 Yahoo

Slovo Yahoo je zkratka tvořená prvními písmeny slov Yet Another Hierarchical Officious Oracle (volně přeloženo *„Ještě jedno nadřazené dotěrné orákulum“*). Yahoo byl založen v lednu roku 1994, kdy začal Jerry Yang se svým kolegou ze Stanfordské univerzity Davidem Filou vytvářet katalog webových stránek. Prostředí internetu bylo v té době ještě značně nepřehledné. Yang s Filou se rozhodli, že udělají v tomto všeobecném chaosu stránek pořádek. Sestavili seznam svých nejoblíbenějších internetových stránek, které podle témat systematicky roztřídili do jednotlivých kategorií. Postupně začali jejich databázi používat lidé po celém světě. Do konce devadesátých let bylo Yahoo nejoblíbenějším vyhledávačem. V té době se ovšem objevil Google a pánové Sergey Brin a Larry Page s ním dokázali během extrémně krátkého času dobýt svět a stát se nezpochybnitelnou jedničkou. Yahoo sice hodně investovalo a vyměnilo technologii katalogového vyhledávače a jako primárně nyní vyhledává fulltextově, nicméně mu to pouze pomohlo udržet si pozici světové dvojky mezi vyhledávači.

Robot Yahoo se jmenuje Slurp. Hned po Google je Yahoo druhým nejpoužívanějším vyhledávačem. Výsledky fulltextového vyhledávání od něj přebírají i další velké vyhledávače, např. Alltheweb.com, Inktomi.com, Teoma.com a Fastsearch.com. Při vyhledávání fulltextem zobrazuje Yahoo na několika málo prvních místech placené odkazy zvláštního systému Yahoo! Search Marketing (dříve Overture), a až pod nimi pak přirozené výsledky.

Největším rozdílem robota Yahoo oproti Googlebotovi je ten, že Slurp indexuje pouze prvních 500kB zdrojového kódu stránky, naproti tomu Googlebot indexuje celou stránku.

2.4 Využití příkazů rozšířeného vyhledávání

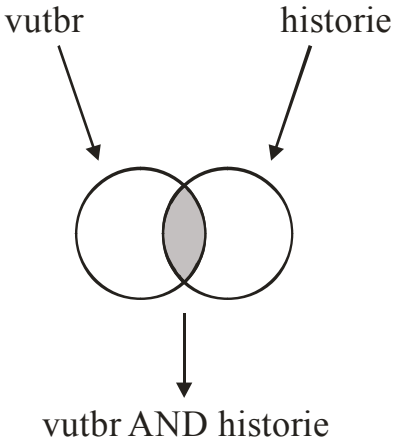
Pomocí vyhledávačů lze vyhledávat odpovědi na nejrůznější dotazy. Ty mohou být jednoslovné, víceslovné a v nejrůznějších jazycích (to záleží na nastavení nebo způsobu zadání dotazu). Zadáváním jednoduchých speciálně formulovaných dotazů lze značně zpřesnit výsledek hledání.

Problémem je, že každý vyhledávač podporuje jiný zápis podmínek nebo jej nepodporuje vůbec. Zobrazením nápovědy lze zjistit přesné definice podmínek, jejichž využití vyhledávací server podporuje. Nejpoužívanější znaky pro zadávání podmínek podporuje téměř každý vyhledávač.

2.4.1 Operátory

Základem vyhledávacího dotazu jsou klíčová slova. Vztahy mezi jednotlivými klíčovými slovy v dotazu určují operátory. Ke kombinování klíčových slov při tvorbě vyhledávacího dotazu se využívají booleovské operátory, které vyjadřují logické vztahy mezi jednotlivými klíčovými slovy, frázemi nebo množinami klíčových slov. Mezi nejpoužívanější operátory patří následující tři: AND, OR a NOT [6].

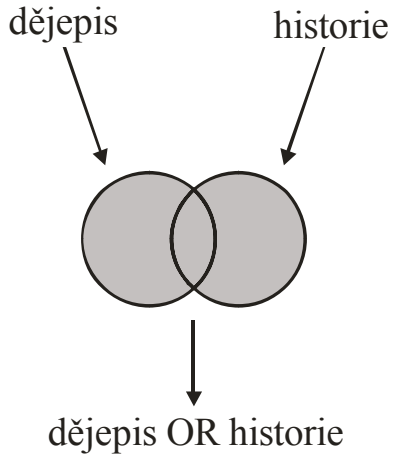
1. **AND** – operátor AND mezi dvěma klíčovými slovy znamená, že se vyhledají jen ty záznamy (dokumenty), které obsahují obě klíčová slova. Uživatel tedy říká vyhledavači aby i toto slovo zahrnul do vyhledávání. Většina vyhledávačů má tento operátor nastaven implicitně pro kombinaci více zadaných klíčových slov. Používá se hlavně k přiřazení slov, která za normálních okolností vyhledavače vypouštějí. Jsou to různá písmena a spojky, které nejsou považovány při vyhledávání za důležité. Tento operátor se zkratkově zapisuje formou znaménka těsně před slovo.



Obr. 4: Ukázka operátoru AND. Výsledná množina záznamů musí obsahovat obě klíčová slova (jde o průnik dvou množin)

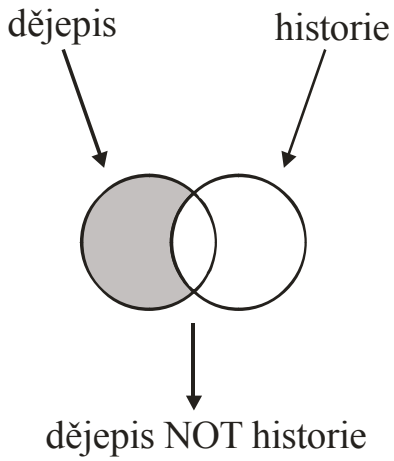
Jako příklad lze uvést uživatele, který chce vyhledat „Romeo a Julie“. Pokud takto formuluje vyhledávací dotaz a nezadá jej do uvozovek jako frázi, většina vyhledávačů zcela vypustí „a“ z hledání a výsledky již nemusí být uspokojivé. Objeví se například „Romeo Og Julie“, což uživatele moc nepotěší, jelikož je to dánsky.

2. **OR** – operátor OR mezi dvěma klíčovými slovy znamená, že se vyhledají záznamy (dokumenty), které obsahující alespoň jedno z uvedených klíčových slov. Tato volba napomáhá při rozšiřování dotazu a většinou se jím spojují synonyma. Ale dnes se převážně používá při složitějších dotazech ke spojování množin.



Obr. 5: Ukázka operátoru OR. Výsledná množina záznamů obsahuje alespoň jedno z klíčových slov nebo obě (jde o sjednocení obou množin)

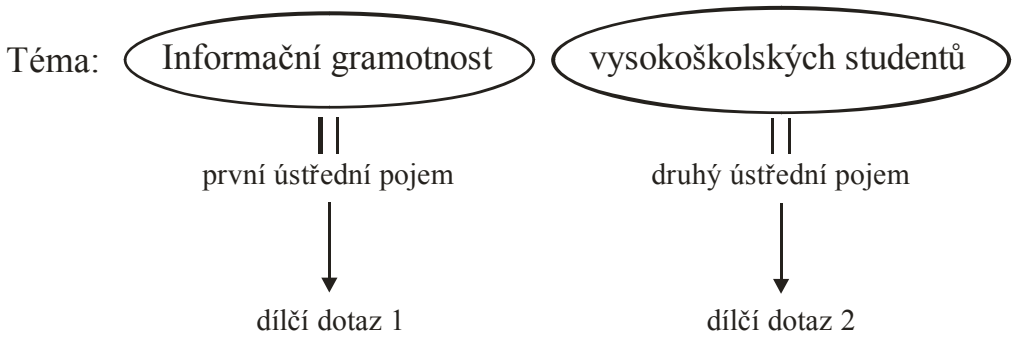
3. **NOT** – operátor NOT mezi dvěma klíčovými slovy znamená, že záznamy (dokumenty) obsahující slovo za tímto operátorem, budou z výsledků vyhledávání vyloučeny. Tento operátor tedy slouží k vyloučení záznamů obsahující dané klíčové slovo z výsledků hledání za účelem zúžení těchto výsledků.



Obr. 6: Ukázka operátoru NOT. Výsledná množina záznamů neobsahuje klíčové slovo historie, musí však obsahovat klíčové slovo dějepis (jde o rozdíl obou množin).

2.4.2 Praktický příklad využití operátorů

Uživatele zajímá a chtěl by vyhledat informace o tématu: Informativní gramotnost vysokoškolských studentů. Toto téma lze rozdělit na dva ústřední pojmy a z těch pomocí frází, operátorů a synonym sestavit výsledný dotaz (požadavek pro vyhledání určité stránky).



Obr. 7: Příklad tvorby výsledného dotazu

K formulování dílčích dotazů se využije množina frází. Tyto fráze budou jednak samotný ústřední pojem, ale také jeho ekvivalentní synonyma. Tím lze získat následující dílčí dotazy:

dílčí dotaz 1:

„informační gramotnost“ OR „informační výchova“ OR „informační vzdělávání“

dílčí dotaz 2:

„vysokoškolský student“ OR „univerzitní posluchač“ OR student

Dílčí dotazy se spojí do výsledného dotazu pomocí závorek a operátoru AND.

výsledný dotaz:
(„informační gramotnost“ OR „informační výchova“ OR „informační vzdělávání“) AND („vysokoškolský student“ OR „univerzitní posluchač“ OR student)

Při samotném hledání je pak možno s dotazem dále pracovat. Pokud bude vyhledáno příliš mnoho záznamů, pak lze odebrat související pojmy a ponechat pouze jednu variantu. Naopak, pokud bude nalezeno málo záznamů, nabízí se přidání dalších příbuzných pojmů.

2.4.3 Syntaxe

Tab. 1 ukazuje srovnání pokročilých syntaxí, jež využívají nejpoužívanější domácí vyhledavače.

Tab. 1: Srovnávací tabulka syntaxí pro jednotlivé vyhledávače:

	www.google.cz	**www.seznam.cz**	**morfeo.centrum.cz**	**www.jyxo.cz**
Nahrazení jednoho či více znaků	nepodporuje	nepodporuje	*	nepodporuje
Ohýbaní (skloňování) a časování slov	automaticky	automaticky	automaticky	automaticky
Možnost vypnutí ohýbaní a časování slov	ne (pouze "")	ne (pouze "")	ano	ano
Doplňování diaktritiky	automaticky	automaticky	automaticky	automaticky
Možnost vypnutí doplňování diaktritiky	ne (pouze "")	ne (pouze "")	ano	ano
Vyhledávání synonym	ano (pouze přes znak ~)	ne	ano	ano
Hledání fráze	"" (uvozovky)	"" (uvozovky)	"" (uvozovky)	"" (uvozovky)
Automatická volba při víceslovném dotazu	AND	AND	AND	AND
Logický operátor *a* (zástupný znak)	AND(+,&)	*nepodporuje*	AND (&)	AND(+,&)
Logický operátor *nebo* (zástupný znak)	OR (|)	*nepodporuje*	OR (?,|)	OR(*nemá*)
Logický operátor *ne* (zástupný znak)	-	*nepodporuje*	NOT(-)	NOT(-)
Proximitní operátor *blízko*	*	*nepodporuje*	*	*
Slovo se může vyskytovat, a pak má odkaz vyšší prioritu	*nepodporuje*	*nepodporuje*	MAYBE	#
Omezení doménou	site:	site:	site:, domain:	site:, domain:
Hledání v titulku	intitle:	*nepodporuje*	title:	title:
Hledání v textu	intext:	*nepodporuje*	text:	*nepodporuje*
Hledání v textu hypertextového odkazu	inanchor:	*nepodporuje*	ext:	*nepodporuje*
Hledání zpětného odkazu	link:	*nepodporuje*	link:	link:
Hledání v adresách stránek	inurl:	*nepodporuje*	urlword:	inurl:, url:
Hledání přesného typu souboru	filetype:	*nepodporuje*	filetype=""	format:

2.4.4 Speciální syntaxe – Google

Tab. 2: Speciální syntaxe vyhledavače Google

	Příklad syntaxe	Význam
Hledání archivní verze stránek	*cache:www.lupa.cz*	Zobrazí předchozí verzi stránky uloženou v archivu Googlu.
Hledání obdobných stránek	*related:www.google.com*	Vyhledá alternativní stránky, které za zabývají obdobnou činností.
Hledání informací o stránkách	*info:www.csfd.cz*	Zobrazí dostupné informace o zadané stránce.
Hledání definice výrazu	*define:router*	Vypíše definici hledaného slova nebo fráze. Použitelné pouze v angličtině.
Kalkulačka	2554/133*12,54	Vypíše výsledek matematické operace (2 554 / 133) * 12,54 = 240,805714. Nehledá výsledek!

2.4.5 Speciální syntaxe – Morfeo

Tab. 3: Speciální syntaxe vyhledávače Morfeo

	Příklad syntaxe (alternativní možnost)	Význam
Hledání obrázků	*alt:tuňák (! ALT "tuňák")*	Vyhledá slovo 'tuňák' v popisku obrázku.
Hledání zvýrazněných slov na stránkách	*emph:tuňák (! EMPH "tuňák")*	Vyhledá zvýrazněná slova - tučně, italikou atd.
Hledání výrazu v jazykových mutacích	lang:en strom (!LANG="en" "strom")	Vyhledá slovo "strom" v anglicky psaných stránkách. Dále je možno použít kódy cs, sk, en, de. Popř. jazyky kombinovat.
Hledání výrazu pouze s diakritikou	! ACCENTS 2 "tuňák"	Vyhledá pouze stránky obsahující slovo 'žena' s diakritikou, nevyhledá slovo 'zena'. Možné mody hledání: 0 - default - pokud dotaz neobsahuje diakritiku vyhledává bez diakritiky, jinak vyhledává s diakritikou v dokumentech, které nějakou diakritiku obsahují, jinde bez; 1 - bez ohledu na diakritiku; 2 - striktně dodržovat diakritiku. Při uvedení ACCENTS na začátku dotazu, platí pro celý dotaz, jinak jen pro dané slovo.
Vyhledání stránky určitého stáří	! AGE < 1209600 "tuňák"	Vyhledá stránky, na kterých se vyskytuje slovo 'tučňák' a stránka je mladší než 7 dní (uvedená hodnota 1209600 je počet sekund), lze použít operátory <, >, =, <=, >=, <>

2.5 Rozdíly fulltextového a katalogového vyhledávání

Na rozdíl od fulltextu lze v katalogu nejen vyhledávat, ale také jím procházet, neboť jednotlivé stránky jsou řazeny do hierarchie.

Při ručním zatřídění člověk (tzv. zatřiďovač) daleko lépe odhadne, zda daná stránka odpovídá určité kategorii nebo ne. Stránka je tedy správně zařazena. Pro uživatele to znamená, že při procházení jednotlivých podsekcí katalogu nalezne stránky, které tam opravdu patří a minimum stránek, které svým obsahem nezapadají do daného tématu.

Při ručním zatřiďování je možné zařadit v podstatě jen celé „weby“, tedy nikoli jednotlivé stránky. Celkový počet zatříděných položek je tak v českém Internetu ve statisících, zatímco velikost českého internetu je v mnoha milionech stránek. Zařazen a zatříděn je tedy pouze „kořen“. Tato skutečnost je vyhovující pouze tehdy, je-li celý web rozložen pouze na jedné stránce.

Fulltext vytvářený automatickým programem (robotem) obsáhne nesrovnatelně více stránek a jde do větší hloubky. Protože ale jde o „umělou inteligenci“, nikoli lidskou intuici, zatřiďuje a zařazuje velmi špatně. Pro uživatele to znamená, že vyhledávací stroj jako odpověď na jím položený dotaz často poskytne odpovědi, které jsou nevyužitelné. Naproti tomu je fulltext schopný najít i ty nejskrytější poklady internetu, které zcela určitě uniknou ručnímu zatřídění. Je vhodný pro kladení přesně specifikovaných a úzce vymezených dotazů.

2.6 Výhody a nevýhody centralizovaného vyhledávání

Jak již bylo řečeno výše, centralizované vyhledávače jsou založeny na centrálním indexu nebo centrálním katalogu. To umožňuje relativně rychlou odezvu na dotaz. Tato rychlost je ale vykoupena skutečností, že informace nalezené v indexu (katalogu) a odeslané uživateli jako výsledek hledání již nemusí být aktuální. Požadovaná stránka už na internetu nemusí vůbec existovat nebo může mít úplně jiný obsah. Tento problém způsobuje nedostatečně krátký interval, po jehož uplynutí robot nebo administrátor provádí kontrolu změn a existence stránek, o nichž již v indexu (katalogu) existují informace. Indexy (katalogy) jsou totiž již natolik rozsáhlé, že kontrola nemůže probíhat v reálném čase.

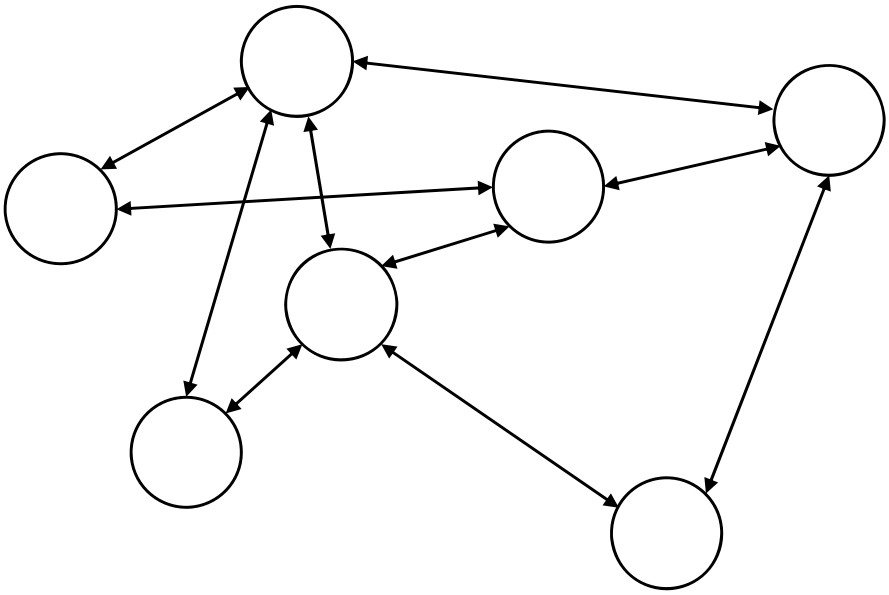
Dalším velmi významným problémem centralizovaných vyhledávačů je skutečnost, že na internetu se již nenacházejí pouze www stránky, ale nepřeberné množství forem dat. Tyto formy nemusí představovat pouze různé datové formáty, ale i systémy poskytující data různým způsobem (např. databáze přístupné přes webové rozhraní). Bohužel v současné době se pod pojmem „data“ v případě centralizovaných vyhledávačů skrývají pouze www stránky a několik málo velmi často používaných souborových formátů (např. Google vyhledává kromě HTML souborů také PDF, DOC, XLS, PPT, RTF a PS). Samozřejmě existují již i vyhledávače specializující se na internetové databáze. Ale tyto vyhledávače se specializují jen na ně a opomíjejí vše ostatní [7].

Mnoho nevýhod vyhledávačů centralizovaných řeší vyhledávače decentralizované.

3 Decentralizované vyhledávače

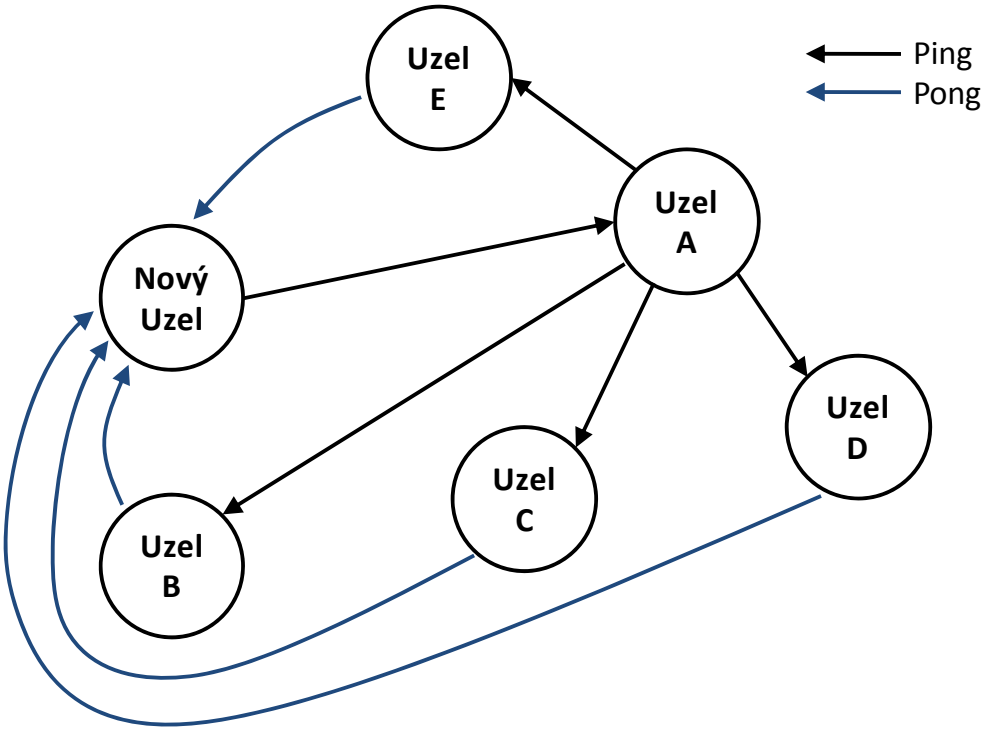
Tento typ vyhledávačů je značně odlišný od předchozích popsaných centralizovaných vyhledávačů. Je zde využito přístupu Peer-to-Peer (P2P), kde neexistuje rozlišení server a klient (Obr. 8). Peer-to-Peer v doslovném překladu znamená rovný s rovným. Jedná se tedy o počítačovou síť, kde každý počítač, respektive uzel, je klientem i serverem zároveň. Není zde potřeba serverů, které by vlastní komunikaci řídily. Přesto se zde serverů využívá, ovšem pouze pro počáteční navázání kontaktů mezi jednotlivými uživateli. Data jsou tedy přijímána i sdílena najednou. Uživatel sám tedy může plnit funkci poskytovatele dat a není pouze pasivním stahujícím. Tím pádem je možné prohledávat data ze všech uzlů v síti, respektive všechna sdílená data z těchto uzlů.

Základní výhodou P2P výměnných sítí je přenosová kapacita, jež se zvětšuje s přibývajícím počtem jejich uživatelů. U stahování z centrálních serverů je tomu přesně naopak. Uživatelé se musí dělit o kapacitu serveru a přenosová rychlost klesá. Naopak se zvyšujícím se objemem dat klesá rychlost vyhledávání.



Obr. 8: Schéma modelu Peer-to-Peer

Jelikož se jedná o decentralizovaný přístup k informacím a tudíž zde neexistuje žádná centrální část, musí být i princip vyhledávání odlišný. Vyhledávání tedy neprobíhá tak, že je poslán vyhledávací dotaz nějakému centrálnímu uzlu, jako tomu bylo u centralizovaných vyhledávačů. Zde uzel vyšle zprávu všem sousedním uzlům, které „zná“. Adresy těchto sousedních uzlů získává různými způsoby, většinou ale tak, že je dostupná nějaká vyhledávací služba orientující se přímo na správu adres. Nově příchozí uzel ke svému připojení do sítě využije této služby k získání adresy alespoň jednoho dalšího uzlu, již v síti zapojeného, k němuž se připojí. Po připojení uzel vyšle speciální zprávu (*Ping*). Příjemce, jenž obdrží tuto zprávu, ji automaticky přeposílá ostatním uzlům ve svém seznamu, aby rozšířil informaci o novém uzlu, a zpět odesílá jinou speciální zprávu (*Pong*). Tou sděluje dotazujícímu se uzlu svou IP adresu. Tímto způsobem získává uzel vazby na jiné uzly, což mu později umožní rozšířit vyhledávací dotaz rychleji a rozprostřít jej do větší šířky.



Obr. 9: Nový uzel připojující se do P2P sítě

Při vyhledávání tedy každý sousední uzel zpracuje dotaz a předá jej zase svým sousedům. Tento způsob propagace dotazů se nazývá příznačně *flooding* (v překladu zatopení). Nejzásadnější nevýhodou tohoto způsobu propagace dotazů je velké zatěžování sítě a fakt, že prohledávání musí být po určitém počtu kroků zastaveno, aby se síť nezahltila. Díky své jednoduché implementaci je to ovšem stále nejvyužívanější způsob propagace dotazů.

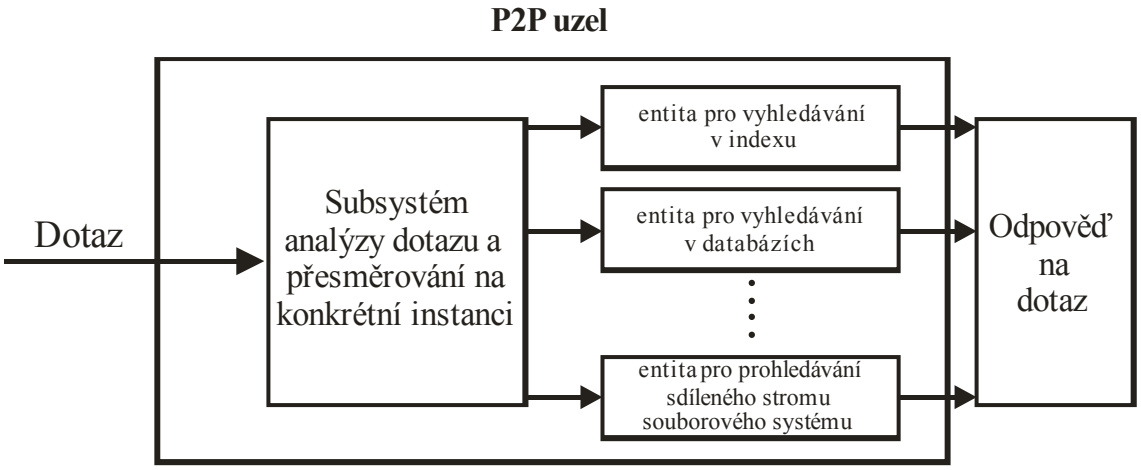
Některé ze sítí již využívají „chytřejší“ způsob propagace dotazů. Jsou v nich k tomuto účelu implementovány dvě funkce zlepšující fungování sítě. První je zabránění zpětných dotazů. Tj. dotázaný klient se dále dotazuje ostatních uzlů, s nimiž má otevřené spojení, ale jak tázající, tak i dotazovaný uzel si pamatují dotaz, a v případě, že se k nim dostane odněkud opakovaně stejný dotaz, jej již dále neposílají. Druhou funkcí je omezená životnost dotazů, tzv. *Time to Live* čítač. Aby dotaz nebloudil sítí nekonečně dlouho, je nastaveno, že po uplynutí určitého času nebo po určitém množství přeskoků vyprší jeho platnost a dále již nepokračuje. Tzn., že tento čítač je na počátku nastaven na určitou hodnotu a s každým navštíveným uzlem nebo uplynulou časovou jednotkou je dekrementován o 1. Jakmile jeho hodnota klesne na nulu, dotaz již není dále uzly přeposílán.

Jelikož jsou P2P sítě plně decentralizované, nemůže zde existovat centrální index. Ten je nahrazen lokálními indexy jednotlivých uzlů. Vzhledem k tomu, že tyto indexy odrážejí stav pouze jednoho jediného uzlu a jsou neustále aktualizovány, je vyhledávání přesné a nenastává stav častý u centralizovaného vyhledávání, kdy je odkazováno na neplatný zdroj.

Velkou výhodou P2P vyhledávání je, že každý uzel může na daný dotaz odpovědět jiným způsobem. To je způsobeno různou vyhledávací logikou jednotlivých uzlů. Např. na dotaz *cos(2pi)* mohou uzly odpovědět zasláním indexu s výpisem adres zdrojů, v jejichž textu se dané slovo vyskytuje, zasláním dokumentu, který obsahuje přímo tento text, zasláním výsledku funkce, zasláním obrázku s průběhem této funkce nebo také vším

najednou. Variabilita reakcí na stejný dotaz je velkou výhodou, neboť vhodnou odpovědí nemusí být vždy webová stránka nebo dokument. Tato možnost, kdy každý uzel může zpracovat daný dotaz naprosto odlišně, dává vyhledávání naprosto jiný rozměr.

Každý uzel může být vybaven softwarem, který bude pro každý typ dotazu používat odlišný způsob zpracování. Dotaz bude tedy nejdříve předzpracován určitým analyzátorem, který rozhodne o druhu dotazu a poté jej přesměruje na příslušnou entitu zpracovávající tento druh dotazu. Vyhodnocovací logika analyzátoru může být nastavena tak, že dotaz pak bude moci být zpracován jednou vhodnou entitou nebo několika najednou. Lze tak mít entitu např. pro zpracování dotazu jako vyhledávání v textu (entita pro vyhledávání v indexu), entita pro hledání tvarů v obrázcích, entita pro vyhledávání v databázích nebo entita pro prohledávání sdíleného stromu souborového systému. Jelikož jsou entity řešeny softwarově, záleží jen na administrátorech jednotlivých uzlů, jaké vyhledávací entity bude daný uzel používat.



Obr. 10: Princip možné modularity softwaru P2P uzlu.

Skutečnost, že nějaký uzel odpoví na dotaz zasláním konkrétních informací, se stává důvodem k tomu, aby byla jeho adresa zařazena do adresáře hledajícího uzlu pro využití při příštím hledání a zaslání dotazu i tomuto uzlu. To předpokládá existenci nějakého rozhodovacího mechanismu, který bude rozhodovat, jaký dotaz kterému uzlu poslat. Tento rozhodovací mechanismus je také velmi důležitý např. při využití floodingu jako způsobu propagace, jelikož je velmi vhodné omezit počet průchodů dotazu uzly. Pokud by toto omezení nebylo aplikováno, dotaz by „bloudil“ po síti velmi dlouho a také by byla způsobena neúměrně vysoká zátěž sítě [7],[8].

3.1 P2P systémy

Na troskách hybridního vyhledávače Napster (viz. kapitola 4.1) byla vytvořena celá technologie peer-to-peer networking. Následující podkapitoly popisují některé z těchto systémů.

3.1.1 Gnutella

P2P systém zvaný Gnutella je přímým následovníkem Napsteru (viz. 4.1). Na rozdíl od něj je již plně decentralizovaný.

Tento systém vyvinuli programátoři Justin Frankel a Tom Pepper. Důvodem ke vzniku Gnutelly byla úvaha, že jediným zákonným problémem Napsteru, díky kterému byl nucen skončit, byla centrální databáze. Proto Frankel a Pepper vytvořili systém, který žádnou centrální databázi nemá - decetralizovanou síť. Tento systém je legálně prakticky nenapadnutelný, protože není koho odsoudit a odpojit, jako to bylo v případě Napsteru.

Velmi podstatným faktorem rozšíření systému Gnutella byl fakt, že byl vytvořen jako open source software (proto GNU v názvu). K dispozici tedy byla nejen kompilovaná, funkční verze, ale také zdrojové kódy. Proto se Gnutelly, která byla v okamžiku zveřejnění vyhotovena v docela rané betaverzi, chopili další programátoři, kteří ji dále dopilovali a rozšířili.

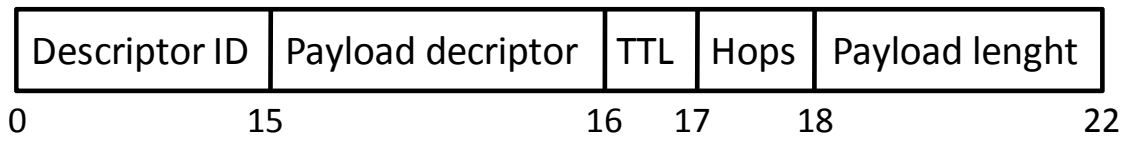
Systém Gnutella tedy pracuje tak, že jednotliví uživatelé mají na svém počítači nainstalován klientský program. Tento program plní nejen funkci klienta, ale i určité funkce serveru. A to z toho důvodu, aby byl schopný posloužit i ostatním uživatelům. Někdy je též nazýván „servent“, jako kombinace slov *server* a *klient*. Program tedy obsahuje klientské rozhraní pro vyhledávání, stahování, atd. a také pro uživatele „neviditelnou“ část, která slouží ostatním uživatelům v síti pro zpracování jejich požadavků.

Po spuštění klientského programu musí nejprve dojít k navázání spojení se sítí. To představuje první obtížný krok, jelikož neexistuje žádný centrální server, se kterým by se program spojil. Existuje ale určité množství „známých“ adres, ke kterým se klientský program na začátku automaticky připojí. Služba běžící na těchto adresách neprovádí nic nelegálního. Pouze shromažďuje adresy uzlů (uživatelů) dané sítě. Neobsahuje žádný seznam souborů, jako tomu bylo u Napsteru.

Po navázání spojení s jedním z uživatelů dané sítě jsou od něj získány adresy dalších uživatelů. Tento proces se opakuje v rychlém sledu, takže v krátké době jsou získána přímá nebo nepřímá spojení na desetitisíce dalších uživatelů dané sítě. Klientský program udržuje přímé spojení s maximálně několika desítkami dalších klientů, jelikož udržování stovek či tisíců otevřených spojení by nepřiměřeně zatěžovalo jak procesor počítače, tak propustnost sítě. Každý z klientů, se kterým je udržováno spojení, je ovšem zase připojen k desítkám dalších klientů. Takže ve výsledku je možné se přes několik spojení dostat ke všem klientům sítě, kteří jsou momentálně připojeni.

Struktura sítě Gnutella je tedy náhodný graf, kde každý uzel uchovává vlastní seznam adres sousedních uzlů, jimž předává dotazy za účelem vyhledávání dat či dalších uzlů.

Součástí každé zprávy putující v síti Gnutella je hlavička, která má strukturu, kterou lze vidět na Obr. 11.



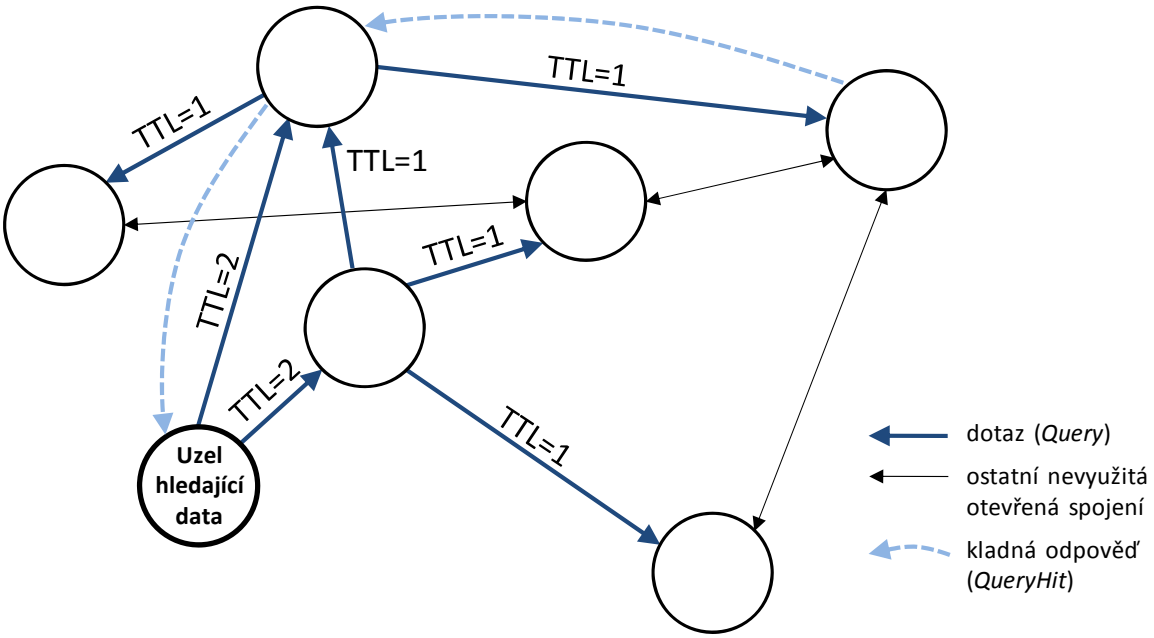
Obr. 11: Struktura hlavičky zpráv putujících v síti Gnutella [9]

Descriptor ID je 16 bytový jedinečný identifikátor dané zprávy. *Payload descriptor* obsahuje typ zprávy - dotaz (*0x80 - Query*), odpověď na dotaz (*0x81 - QueryHit*), získání adresy jiných uzlů (*0x00 - Ping*), odpověď na *Ping* (*0x01 - Pong*) nebo inicializace přenosu souboru (*0x40 - Push*). *TTL* (Time to Live) určuje kolika uzly bude ještě zpráva zpracována než zanikne. Při inicializaci obvykle nastaveno na 10 po každém zpracování zprávy v uzlu, klientský program dekrementuje tuto hodnotu o jedna. *Hops* je číslo, které určuje kolika uzly již zpráva prošla a po každém zpracování zprávy v uzlu, klientský program inkrementuje tuto hodnotu o jedna. *Payload lenght* udává počet bytů samotné zprávy, tedy dat následujících za onou hlavičkou.

Šíření zpráv v síti Gnutella se řídí čtyřmi základními pravidly:

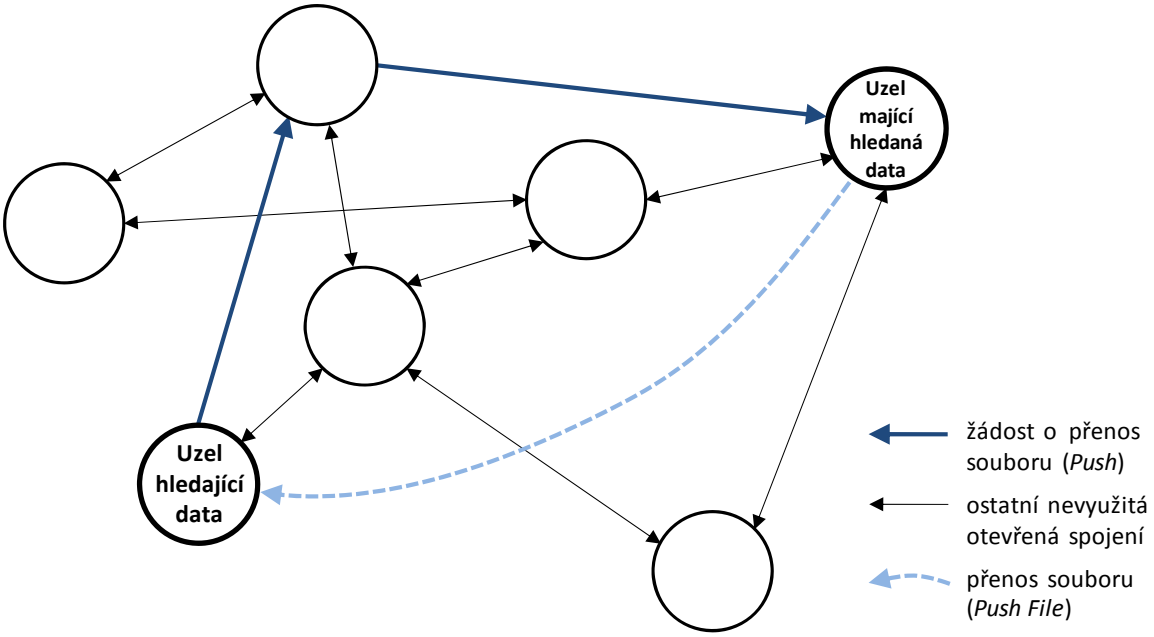
1. Uzel, jenž obdrží zprávu, ji přeposílá na všechny další uzly kromě toho, ze kterého přišla.
2. Kladná odpověď na vyhledávací dotaz (*QueryHit*) musí putovat stejnou cestou, jakou přišla původní zpráva, jen opačným směrem. Uzly předávající zprávu si po určitou dobu uchovávají ve své tabulce seznam předaných zpráv (z jakého uzlu kam byla předána), takže jsou potom schopni přeposlat tuto odpověď opačným směrem.
3. Uzel zvýší hodnotu Hops o jedna a zároveň sníží hodnotu TTL taktéž o jedna. Pokud hodnota TTL dosáhne 0, zprávu zahazuje.
4. Jestliže uzel obdrží zprávu, která se již nalézá v jeho tabulce obdržených a přeposlaných zpráv, tuto zprávu zahodí.

Samotné vyhledávání tedy probíhá v síti Gnutella tak, že na vyhledávací dotaz (zpráva typu *Query*) dostává klientský program kladné odpovědi (zpráva typu *QueryHit*). Tento průběh lze vidět na Obr. 12.



Obr. 12: Vyhledávání Gnutella - dotazy (*Query*) a kladná odpověď (*QueryHit*)

Z došlých kladných odpovědí vyhodnotí nejlepší možnou variantu a vyšle žádost o přenos souboru (zpráva typu *Push*). Tento průběh lze vidět na Obr. 13.



Obr. 13: Vyhledávání Gnutella - žádost o přenos souboru (*Push*) a samotný přenos souboru (*Push File*)

Samotný přenos soubor již potom neprobíhá přes síť Gnutella, nýbrž díky oboustranné znalosti IP adresy je navázáno přímé spojení mezi dvěma uzly.

Gnutella využívá pro přenos souborů protokolu HTTP. Důvodem použití tohoto protokolu, který jinak není pro stahování souborů nejvhodnější, je jeho univerzálnost a nemožnost jej zakázat. Každý počítač připojený k internetu má tento protokol povolený a vždy může stahovat z webu soubory. Při využití nějakého speciálního protokolu pro stahování souborů by bylo možné jej zakázat např. na proxy serverech.

Největším problémem této sítě, ale i obecným problémem P2P sítí, je samozřejmě legálnost sdíleného, vyhledávaného a stahovaného obsahu. Dalším problémem Gnutelly je nízká úroveň anonymity. Uživatel (uzel), z něhož jsou stahována data, zná přesnou IP adresu uživatele, jenž data stahuje. Ta pak může být klíčem k případnému dohledávání uživatele, respektive konkrétního počítače. Důležitým nedostatkem Gnutelly, který byl později řešen v jiných sítích, je také skutečnost, že předávání dotazů mezi jednotlivými uzly vyžaduje značnou přenosovou kapacitu. I když jde jen o krátké dotazy, je jich nesmírně mnoho, jelikož jejich propagace je řešena způsobem *flooding* [10].

3.1.2 FreeNet

Stejně jako Gnutella je FreeNet plně distribuovaný P2P systém a open source software. FreeNet tedy také postrádá jakýkoliv centralizovaný prvek, jehož výpadek by výrazně ovlivnil činnost systému.

Systém FreeNet vymyslel a implementoval Ian Clarke v červenci roku 1999, jako student na univerzitě v Edinburgu. Princip tohoto systému je do značné míry podobný Gnutelle s tím rozdílem, že je zde kladen mnohem větší důraz na anonymitu a je zde efektivněji řešeno ukládání a distribuce dat. FreeNet slučuje všechny své uživatele do jedné velké sítě, kde je každý uživatel v absolutní anonymitě a veškerá komunikace probíhá šifrovaně.

Vyhledávací proces začíná stejně jako u Gnutelly vytvořením žádosti, která je ve formě zpráv zaslána jiným uzlům v síti. Každá žádost má také své unikátní ID, které ji a všechny zprávy, které pod záštitou této žádosti vzniknou, specifikuje v celé síti. Systém FreeNet rozlišuje dva typy žádostí. K obvyklému typu, tedy žádosti o data (*DataRequest*), která slouží k vyhledání dat, je žádost typu *InsertRequest*, která slouží k ukládání dat na uzly FreeNetu. Žádost o data (*DataRequest*) v systému FreeNet se od žádosti o data v systému Gnutella (*Query*) liší především v tom, že každá data v této síti mají svůj klíč a vyhledávání probíhá na základě poskytnutého klíče, který je obsažen v této žádosti.

Zprávy putují sítí, dokud nevyprší povolený počet navštívených uzlů (Time to Live) nebo nejsou nalezena požadovaná data. Pokud vyprší tento povolený počet navštívených uzlů, je dotazujícímu se uzlu vrácena zpráva typu *TimedOut*.

Stejně jako v systému Gnutella je odpověď na dotaz zaslána dotazujícímu se uzlu zpět po stejné trase, jakou putovala. Rozdíl je zde ovšem v tom, že spolu s odpovědí putují i požadovaná data. Tento velký rozdíl má výhodu v tom, že všechny uzly, přes které odpověď putuje, si mohou tato data uložit do své cache paměti a při příštím dotazu na stejná data je poskytnout namísto přeposílání na původní zdroj dat. Tato replikace ušetří síti zasílání mnoha zpráv při opakovaném vyhledávání velmi poptávaných dat. Rozptýlení dat mezi více uzlů také napomáhá minimalizovat případy, kdy se tato velmi poptávaná data stávají téměř nedostupnými vlivem zahlcení uzlu požadavky na tato data.

Odpověď mimo dat obsahuje adresu uzlu, ze kterého byla zpráva odeslána. Tato adresa se ale pokaždé při přeposlání přes uzel mění a tudíž uživatel, který obdrží jím požadovaná data, vůbec neví, odkud mu data přišla. Pouze ví adresu bezprostředně nejbližšího uzlu, přes který data putovala. Skutečnost, že uzel zná pouze své nejbližší sousedy, tj. ty, se kterými má navázáno otevřené spojení a se kterými komunikuje, je celkem kvalitním bezpečnostním mechanismem. Dohledání zdroje je jistě teoreticky možné, ale v systému FreeNet je upřednostňován optimální poměr anonymita/efektivita oproti naprosté anonymitě, jelikož čím vyšší míra anonymity, tím nižší efektivita vyhledávání [11].

3.2 Výhody a nevýhody decentralizovaného vyhledávání

Současné P2P systémy jsou zatím bohužel proslulé pouze šířením nelegálního softwaru a zdaleka nevyužívají výhod, které P2P model poskytuje. Současný výzkum v oblasti P2P technologií ale stále pokračuje, není tedy vyloučeno, že se v brzké době může tento stav změnit.

Po shrnutí všech výše popsaných vlastností decentralizovaných vyhledávačů je jasné, že řeší většinu nevýhod vyhledávačů centralizovaných. Na druhou stranu je jejich velkou nevýhodou relativně pomalá rychlost vyhledávání. Tato nevýhoda je samozřejmě způsobena decentralizací. Při stejném objemu dat nelze provést srovnatelně rychlé vyhledávání, jaké je možné provést u centralizovaných vyhledávačů. Je samozřejmě možné

volby vhodnější topologie, účinnější využití vyhledávací logiky, použití replikace dat, cache přístupů a dalších oblastí, které napomáhají zefektivnit vyhledávání.

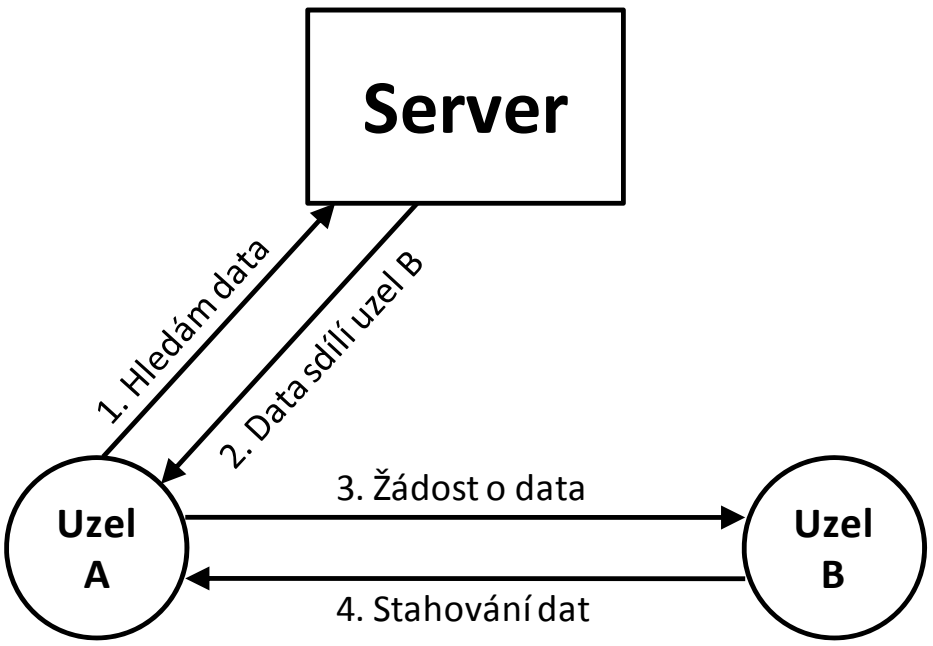
Lze tedy říci, že rychlost vyhledávání je jediným nedostatkem decentralizovaného vyhledávání. Ovšem při upřednostnění výsledku vyhledávání v podobě menšího množství vysoce relevantních před velkým množstvím méně relevantních výsledků, které lze díky decentralizaci získat, se nejeví rychlost vyhledávání jako natolik negativní faktor.

4 Hybridní vyhledávače

Jak z předešlých dvou kapitol vyplývá, ideální vyhledávač by byl ten, který by vykazoval rychlost centralizovaného vyhledávače a zároveň poskytoval výhody P2P vyhledávačů. Toto snoubení užitečných vlastností nám přináší hybridní vyhledávače.

Uzly (uživatelé) vzájemně komunikují P2P (tj. přímo jeden s druhým), avšak používají zároveň centrální server, který udržuje informace o obsahu a stavu všech těchto uzlů v síti. Server také obsahuje databázový index. Ten tvoří seznam souborů, které jsou uloženy a k dispozici na ostatních připojených uzlech. O každém souboru jsou také uloženy bližší informace, tj. na jakém uzlu se nachází, jakou má tento uzel internetovou adresu, ve kterém adresáři je daný soubor uložen, velikost tohoto souboru, jméno uživatele (nickname - přezdívka).

Tento centrální prvek vnáší do systému nevýhody, které jsou s centralitou spojeny. Na druhou stranu ovšem poskytuje podstatné zvýšení rychlosti vyhledávání. Vyhledávání v těchto systémech probíhá ve dvou fázích. Nejprve uzel, který se chce do sítě připojit, odešle serveru informace o obsahu, který hodlá sdílet. Server si tyto informace uloží. Toto nápadně připomíná obdobu vytváření indexu s tím rozdílem, že server rovnou dostane informace o místě výskytu sdílených dat a díky tomu nemusí tuto proceduru provádět sám. Zjednodušený princip hybridního vyhledávače ukazuje Obr. 14. Z obrázku je patrné, že tento proces se skládá ze 4 kroků. Uživatel nejprve odešle serveru dotaz na vyhledávání. Server ve svém indexu vyhledá odpovědi na daný dotaz a odešle uživateli informace kde se v síti hledaná data nacházejí. Poté uživatel odešle žádost o data uživateli, jenž tato data sdílí, a ten mu je následně poskytne.



Obr. 14: Zjednodušený princip hybridního vyhledávače

Fakt, že hybridní vyhledávače využívají centrálního indexu, poskytuje velkou rychlost vyhledávání spolu s P2P přístupem, jelikož konkrétní uzly již komunikují pouze mezi sebou a centrální server k tomu nepotřebují. Problém ovšem nastává při výpadku nebo

selhání serveru. V tomto případě není možné provádět vyhledávání a nastává tedy stejná situace, jako pří výpadku serveru u centralizovaného vyhledávání. Tato nevýhoda se totiž přímo pojí s modelem Klient-Server.

4.1 Napster

Napster byl typickým představitelem hybridního vyhledávání. Byl to velmi populární systém pro sdílení a vyhledávání dat, převážně ve formátu MP3.

Historie Napsteru se datuje na konec roku 1998, kdy napadlo studenta jménem Shawn Fanning, přezdívaného Napster, že by bylo možné tehdejší krkolomné metody sdílení souborů přes internet obejít a vytvořit systém, který by obsahoval celosvětovou databázi souborů pro výměnu. Fanning odešel na začátku roku 1999 ze studií a již v květnu téhož roku spustil Napster na serveru www.napster.com. Pro přístup k této službě bylo nutné mít nainstalován program Napster, který se dal zdarma stáhnout z internetu.

Popularita Napsteru se šířila obrovským tempem. Během několika málo měsíců od spuštění již měl několik miliónů aktivních uživatelů a internetem denně putovalo stovky tisíc souborů. Valná většina z nich ovšem byla nelegálních a proto vlastníci autorských práv (vydavatelské firmy) reagují podáním žaloby na Napster. Za třináct měsíců po spuštění Napsteru je vydán předběžný příkaz zakazující Napsteru provoz (červen 2000).

Jakmile začalo být zřejmé, že Napster končí, začalo mnoho dalších vývojářů a podnikatelů souběžně pracovat na jiných metodách, které rovněž umožní výměnu souborů mezi uživateli podobným způsobem jak Napster, ovšem u kterých vývojář programu službu neorganizuje a neudržuje centrální databázi. Vývojář pouze dodá klientský prográmek, který dokáže vyhledávat uživatele, spojovat je a vytvářet tak "neviditelnou" síť bez čehokoliv, co vypadá jako centrální bod [12].

5 Inteligentní vyhledávací agenti

Je možné je přirovnat k softwarovým robotům, kteří pomáhají uživatelům dosáhnout hledaných informací kvalitněji, pohodlněji a rychleji. Tito agenti by měli znát a předvídat potřeby a zájmy uživatele. Vykonávat za něj obvyklé pracovní procesy, které jsou při „lidském“ zpracování časově náročné. Agenti mohou neustále shromažďovat nové informace týkající se práce nebo zájmu uživatele, popřípadě subjektu, o nějž se starají. Uživatel si pak už jen ve zkratce projde hlavní body a zaměří se jen na relevantní informace, což mu ušetří spoustu času.

Tyto agenty ovšem nelze vytvořit na prázdné platformě. Je nutno je zasadit do určitého informačního „potrubí“, kterým je sémantický web.

5.1 Sémantický web

Webové stránky jsou navrženy pro čtení a zpracování uživatelem, tedy člověkem nikoliv strojem. Proto fulltextové vyhledávače nejsou schopny seřadit výsledky svého hledání relevantně. Datové formáty zdrojů na internetu (HTML, pdf, atd.) totiž popisují pouze formální prezentační stránku nalezené informace a nikoliv tu významovou, která pro lidského uživatele plyne z jejich obsahu. Nalezené informace tedy mohou mít spoustu významů, jejichž význam je zřejmý až z kontextu, který vyplývá ze zkušenosti uživatele.

Nejprve je tedy nutné přetransformovat informace do datové podoby, která je strojem rozpoznatelná. Nad těmito informacemi je pak nutno vytvořit určitou sémantickou strukturu, aby agenti zpracovávající dané informace nemuseli být tak „chytří“.

Sémantický web je tedy vlastně rozšířením současného webu, v kterém jsou data popsána tak, aby jejich významu rozuměly i počítače. Není tedy žádným speciálním novým typem webu, nýbrž jen souborem technologií, jež doplní stávající web o metadata popisující význam webového obsahu jako strojově čitelná informace. Skutečnost, že takto interpretované informace dávají smysl i strojům, umožní integraci dat z různých webových (ale i vnitrofiremních) zdrojů a daleko širší možnosti jejich zpracování [13].

5.2 Technologie sémantického webu

Podmínkou pro vytvoření sémantického webu byl vznik logických pravidel pro zápis metadat. Tato pravidla musela být navržena tak, aby vyhovovala globálnosti a otevřenosti webového prostředí. Není totiž problém vytvoření počítačem srozumitelné datové reprezentaci informace v uzavřeném systému a na omezené doméně. Problém je ve velikosti webu. Proto sémantický web poskytuje tak otevřený a flexibilní datový model a jazyky pro jeho zapsání, aby vyhovoval téměř nekonečným variacím webu.

Základní komponentou sémantického webu je URI (Uniform Resource Identifier). URI slouží k jednoznačnému identifikování zdrojů a představuje obecně použitelnou množinu pro všechny druhy adres. URL (Uniform Resource Locator) známé z internetu, je

podmnožinou URI. URI totiž může popisovat nejen elektronické dokumenty, ale také předměty reálného světa (např. domy, knihy a lidi). V sémantickém webu má tedy všechno své URI, a proto mohou metadata popisovat prakticky cokoliv.

Pro zachycení sémantické informace z dokumentu byl vytvořen standard RDF (Resource Description Framework) – rámec pro popis zdrojů. Je obecným mechanismem pro zápis metadat. Poskytuje kompatibilitu mezi aplikacemi, jež si na webu vyměňují strojům srozumitelné informace. Tento standard používá pro svůj zápis jazyka XML (eXtensible Markup Language). Ten ale vnáší do dokumentů jenom strukturu a neříká nic o jejich významu. Význam dokumentu vyjadřuje základní jednotka standardu RDF a to tzv. tvrzení (Statement). Tvrzení je vždy tvořeno trojicí – podmět, přísudek a předmět, takže má vlastně tvar jednoduché věty a tvoří orientovaný graf. Zjednodušeným příkladem datového modelu sémantického webu (tvrzení) může být např.: „osoba_1“ (podmět, zdroj) „je_absolventem“ (přísudek, vlastnost) „VUT Brno“ (předmět, hodnota vlastnosti).

RDF představuje univerzální vyjadřovací strukturu k zachycení významu informace počítačově čitelným způsobem. Nedefinuje ovšem termíny, se kterými pracuje. K tomuto účelu slouží v sémantickém webu ontologie. Ta je definována jako formální a jednoznačná specifikace sdíleného abstraktního modelu. Tímto modelem je myšlen abstraktní model libovolné z existujících domén. Ontologie v těchto doménách definuje slovník sdílených pojmů a termínů, jejich vlastnosti a vzájemné vztahy.

Tedy v sémantickém webu RDF používá slova (termíny) na vyjádření faktů a ontologie poskytují „výkladové slovníky“ těchto slov a „gramatiku“, která obsahuje pravidla pro jejich používání [7],[14].

5.3 Příklad inteligentního vyhledávacího agenta

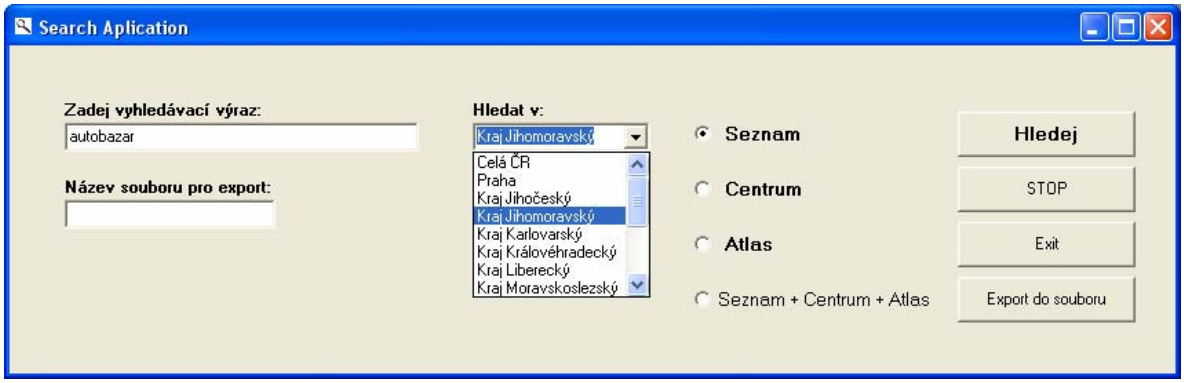
Jako příklad bude uveden jeden z nejrozšířenějších aplikačních scénářů, které by měl sémantický web v budoucnosti umožňovat. Není podstatné, jestli tomu bude přesně takto. Tento příklad slouží k objasnění možností využití potenciálu sémantického webu.

Cestovní agentura chce poskytovat svým klientům komplexní zabezpečení cesty. Bude mít tedy na starosti vystavení rezervací u poskytovatelů různých služeb (letenka, jízdenka na autobus nebo vlak, půjčení osobního vozidla, rezervace hotelu, atd.). Cestovní agentura využije ke komunikaci se svými dodavateli inteligentního softwarového agenta. Vstupem pro tohoto agenta budou požadavky klienta. Tyto požadavky mohou být různě specifické a mohou obsahovat např.: datum odjezdu, datum příjezdu, destinaci, preferovaný způsob dopravy, kategorii ubytování, návštěvu vybraných turistických aktivit, apod. Agent pak na základě těchto vstupů buď sám vyhledá dané informace (probíhá jako sběr metadat a jejich vyhodnocování) nebo kontaktuje agenty dodavatelů, jež mu poskytnou tyto informace. Agent se tedy snaží najít vyhovující služby. Data od poskytovatelů mohou být zapsána pomocí jiné ontologie, než kterou využívá daný agent. Například to mohou být různé jazykové mutace, rozdílné metrické jednotky (kilometry nebo míle), apod. V těchto případech je zapotřebí přeložit data z jedné ontologie do druhé. K tomu využije agent jiného agenta (tlumočníka), který se daným problémem zabývá. Ze vzájemných interakcí agentů vzejde několik možných návrhů na realizaci cesty. Následně je agent seřadí podle relevance k vstupním datům tak, aby co nejlépe splňovaly dané požadavky klienta. Takto zpracované informace poskytne jako výsledek své práce lidskému uživateli. Ten po výběru jedné z nabízených možností potvrdí agentovi, aby objednal u dodavatelů dané služby.

6 Vytvoření vlastní vyhledávací aplikace

Jedním z dílčích cílů této diplomové práce bylo vytvoření vyhledávací aplikace. Ta umožňuje uživateli vyhledávat a stahovat kontakty firem podle zadaného vyhledávacího výrazu a ze získaných dat následně vytvářet databáze kontaktů ve formátu CSV. Zdrojem pro vyhledávání budou firmy nacházející se ve třech největších českých katalozích firem, které provozují portály Seznam, Centrum a Atlas na samostatných doménách *www.firmy.cz*, *firmy.centrum.cz* a *firmy.atlas.cz*. Jako způsob vyhledávání je tedy použito katalogové vyhledávání. To je již dnes sice naprosto nevhodné pro hledání informací, ale na druhou stranu naprosto vhodné pro hledání kontaktů.

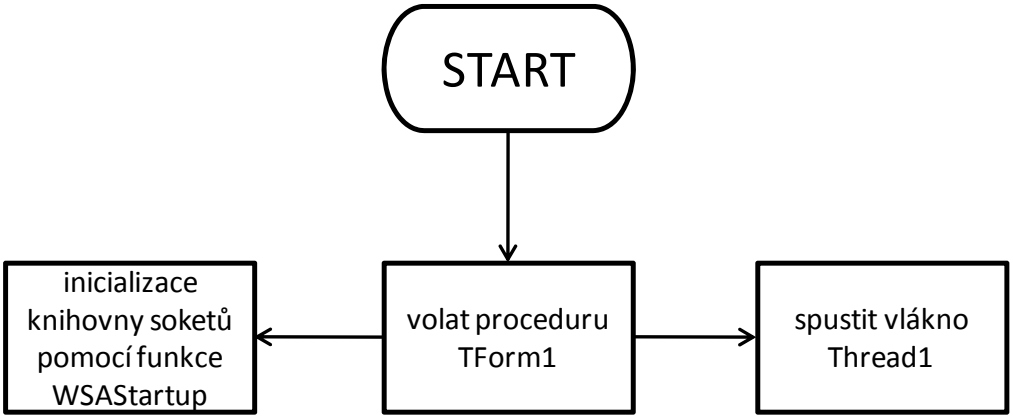
Po zadání vyhledávacího výrazu, výběru zeměpisné oblasti, ve které se mají dané firmy nacházet (implicitně nastaveno na celá ČR), a zvolení příslušného katalogu, ve kterém má být hledání provedeno, uživatel klikne na tlačítko *Hledej* (Obr. 15). Tím dává příkaz aplikaci, aby se připojila k serveru vybraného katalogu a získala zde jím požadovaná data. Tato data jsou posléze vypsána do pomocného Labelu a pomocí tlačítka *Export do souboru* si je uživatel může uložit pro pozdější využití.



Obr. 15: První ukázka z tvorby vyhledávací aplikace

6.1 Popis tvorby aplikace

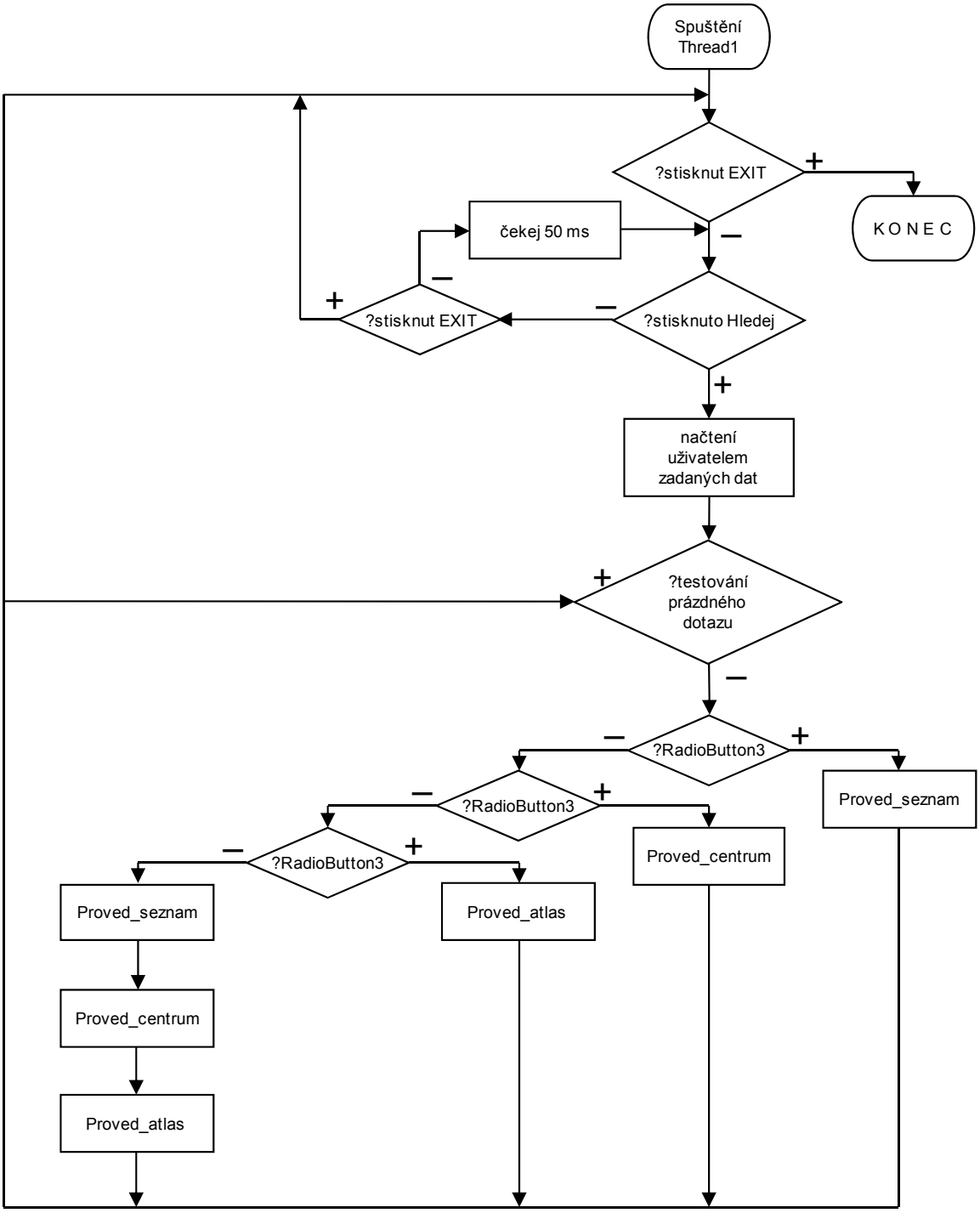
Vyhledávací aplikace byla vytvořena ve vývojovém prostředí Borland C++ Builder, který pomocí nejnovějších funkcí usnadňuje vývoj aplikací v jazycích C a C++. Jako základ vyhledávací aplikace byl využit moderní způsob správy programu v oddělených speciálních funkcích – tzv. vláknech (threadech). Aplikace využívá dvou vláken. Ta jsou provozována samostatně a paralelně, kdy jádro systému se samo stará o přidělování strojového času jednotlivým částem podle priorit jednotlivých vláken.



Obr. 16: Zjednodušený vývojový diagram vyhledávací aplikace

Ihned po startu aplikace (Obr. 16) je otevřeno hlavní okno implementované třídou TForm1. Ta je potomkem třídy TForm z VCL, jenž je předkem pro všechna okna. Tato třída vytváří grafické rozhraní aplikace a zapouzdřuje všechny prvky umístěné v okně, jako jsou tlačítka, editační pole, vypisovací labely apod. Dále je spuštěno další samostatné vlákno (Thread1), které má ovšem nižší prioritu než hlavní vlákno a je mu podřízeno. Jelikož aplikace využívá sokety, je nezbytné inicializovat knihovnu soketů pomocí funkce WSAStartup. Naproti tomu je nutné na konci běhu aplikace zavolat funkci WSACleanup, která by měla být zavolána po ukončení práce se sokety.

Vlákno Thread1 obstarává samotnou práci aplikace (Obr. 17), tzn. že po stisknutí tlačítka *Hledej* jsou nejprve načtena uživatelem zadaná data (tj. vyhledávací výraz, zeměpisnou polohu a zvolený katalog). Poté podle zvoleného katalogu volána příslušná funkce (tj. *Proved_seznam*, *Proved_centrum* nebo *Proved_atlas*, popřípadě všechny tři, pokud byla zvolena možnost vyhledání ve všech třech katalozích).



Obr. 17: Vývojový diagram Thread1

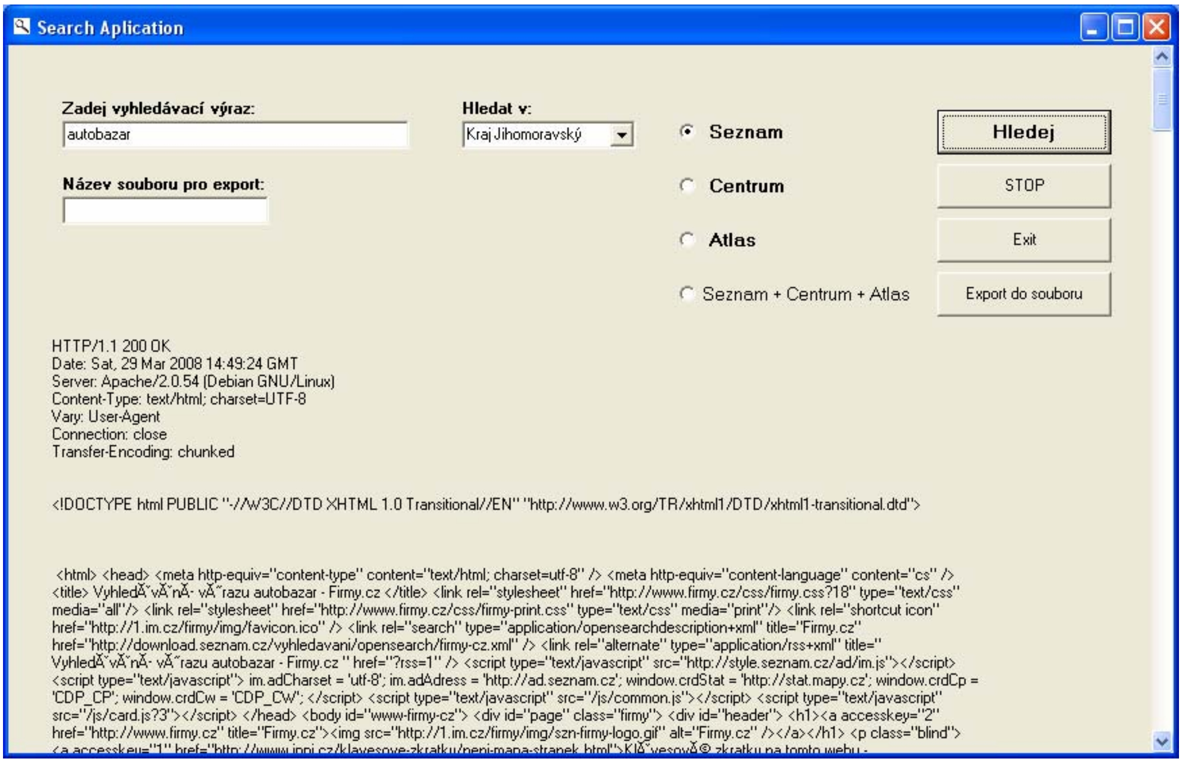
V těle samotných funkcí *Proved_seznam()*, *Proved_centrum()* nebo *Proved_atlas()* je nejprve z vyhledávacího výrazu pomocí funkcí *encodeString_seznam()*, *encodeString_centrum()* respektive *encodeString_atlas()* vytvořen vyhledávací dotaz odpovídající formátu vybraného katalogu. Dále je k vyhledávacímu dotazu pomocí funkcí *pridej_oblast_seznam()*, *pridej_oblast_centrum()* respektive *pridej_oblast_atlas()* přidána zeměpisná poloha, ve které má vyhledávání provedeno. Poté je pomocí funkce *gethostbyname()* zjištěna IP adresa odpovídající jménu domény daného katalogu. Poté je ustanoven soket, naplněna adresa a port a realizováno připojení na daný server. Nyní již

může aplikace pomocí HTTP protokolu verze 1.1 a jím podporované metody *GET* dávat požadavky na zaslání dokumentu přesně určeného pomocí URL.

Aplikace tedy zadává dotaz ve formátu:

GET /cesta/soubor HTTP/1.1

Dále je k dotazu připojena hlavička (*hearder*), ve které jsou specifikovány požadavky na daný dokument, který má být serverem poslán. Po přijetí odpovědi od serveru načte tento soubor do proměnné typu *string*. Přijatý soubor představuje zdrojový kód stránky (Obr. 18).



Obr. 18: Ukázka z tvorby vyhledávací aplikace s výpisem staženého souboru

Na Obr. 18 lze v pomocném vypisovacím Labelu vidět, že přijatý soubor je zdrojovým kódem stránky s hlavičkou na začátku. Z obrázku je také patrné, že daný server (v tomto případě Seznam) využívá kódování UTF-8 (Centrum používá Latin-2 formálně správně ISO/IEC 8859-2 a Altas rovněž UTF-8). Pro vyhledávání odkazů na stránky s kontakty na jednotlivé firmy je tedy nejprve nutné převést tato kódování do win-1250.

Samotné vyhledaní substringu, tj. adresy souboru obsahujícího kontaktní údaje firmy nalezené uživatelem, je provedeno metodou *find()*. Následně je tedy tento substring vyparserován pomocí metody *substr()*.

Dále následuje opět metoda *GET* a požadavek na soubor s kontaktními údaji nalezené firmy. Bylo nutné ošetřit přesměrování, jelikož ne vždy je udávaná cesta vždy cílovou adresou daného souboru. Po přijetí odpovědi od serveru je opět načten tento soubor do proměnné typu *string* a z ní vyparserovány hledané kontaktní údaje (název firmy, ulice, číslo popisné, město, PSČ, telefon, E-mail a adresa www stránek nalezené firmy). Pro

extrahování kontaktů firem jsou použity obdobné metody jako při specifikaci cest k souborům.

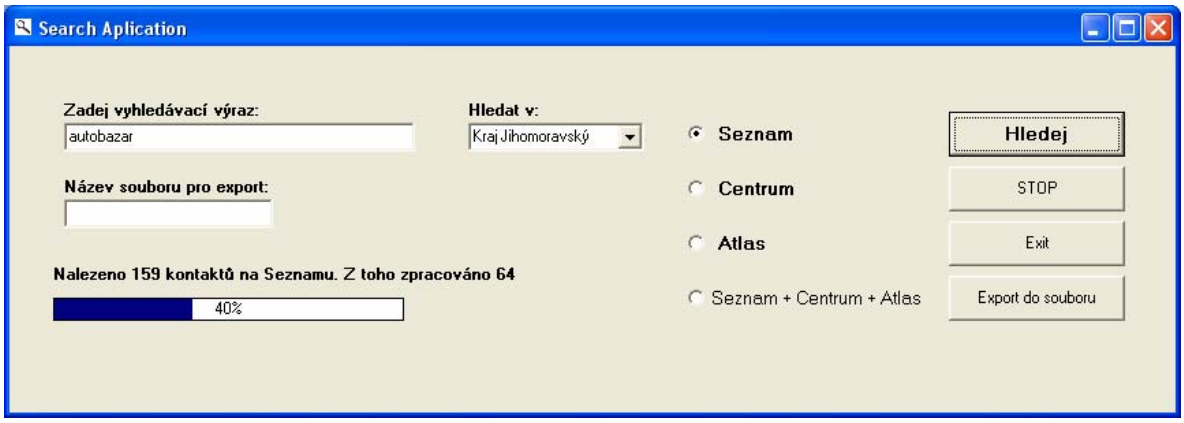
Stejný postup je následně aplikován pro získání ostatních kontaktů firem nalezených na téže stránce. Je také testováno, zda soubor neobsahuje odkaz na další stránku s nalezenými údaji. Při kladném výsledku je opět poslán požadavek na soubor s další stránkou nalezených výsledků.

6.2 Popis aplikačního prostředí a práce aplikace

Aplikační prostředí tvoří formulář. Ten je reprezentován objektem Form1, který je typu TForm1. Do formuláře je vloženo několik komponent (tlačítka, editační pole, labely a radiobuttony). Po každém vložení komponenty do formuláře je také vložena nová položka se jménem komponenty do deklarace typu formuláře. Pak všechny obsluhy událostí jsou metody třídy formuláře (TForm) a tyto metody jsou také deklarovány v samotném typu formuláře.

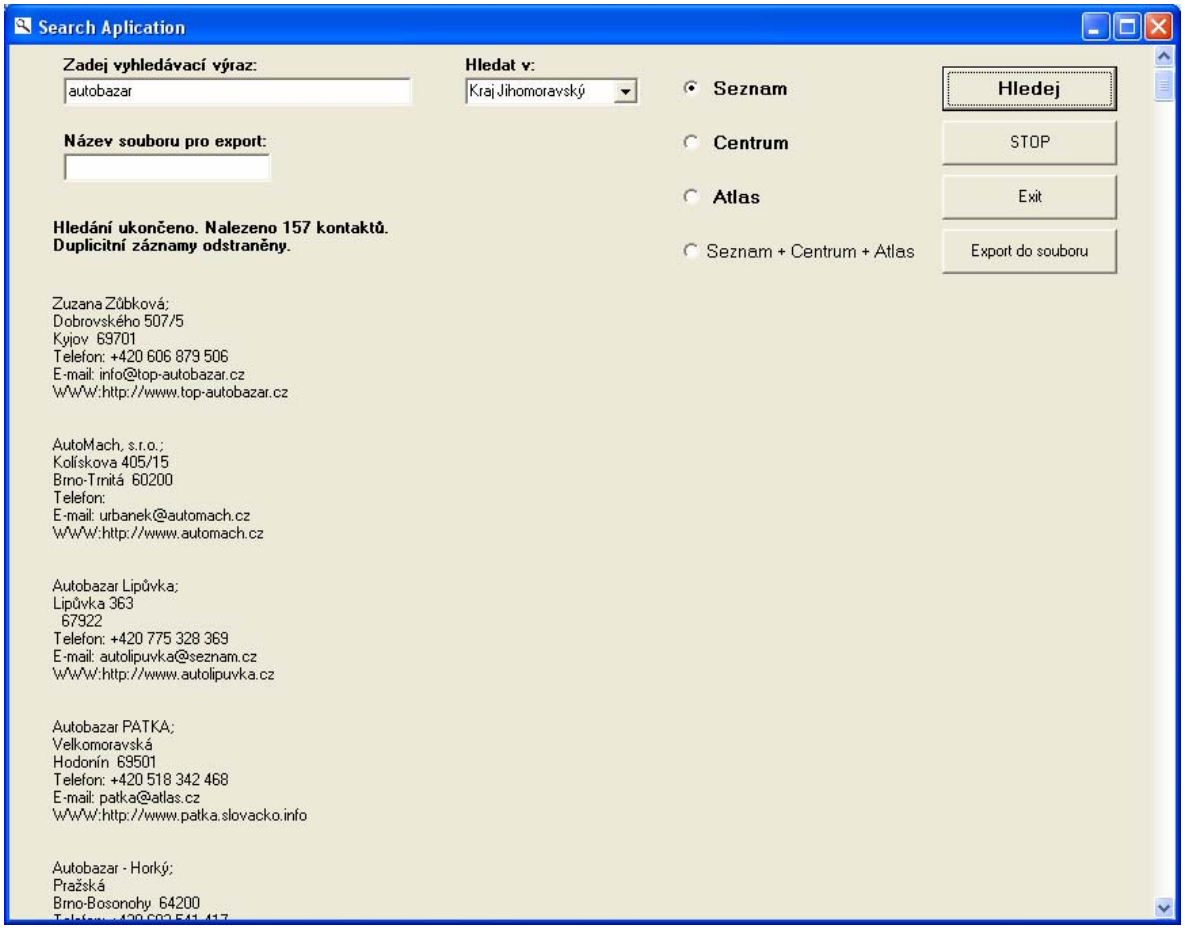
Na Obr. 19 lze vidět již hotovou aplikaci při zpracovávání nalezených kontaktů. Obsahuje tedy 4 tlačítka, 3 editační pole pro zadávání „textu“, 4 radiobuttony pro výběr katalogu, několik popisných labelů a 1 progressbar pro lepší přehled o průběhu.

Po zadání vyhledávacího dotazu a stisku tlačítka *Hledej* je vypsán počet nalezených kontaktů a začne jejich zpracování. Zpracování je možné kdykoliv přerušit stiskem tlačítka *STOP*. Poté budou vypsány a umožněno vytvoření databáze pouze z těch kontaktů, zpracovaných před stiskem tlačítka *STOP*. Lze také aplikaci kdykoliv ukončit stiskem tlačítka *Exit*. Toto ukončení neproběhne okamžitě, ale trvá cca 1 sekundu, protože ukončení je volané z hlavního řídícího vlákna TForm1, které musí počkat na ukončení podřízeného vlákna Thread1, a teprve poté ukončit samo sebe.



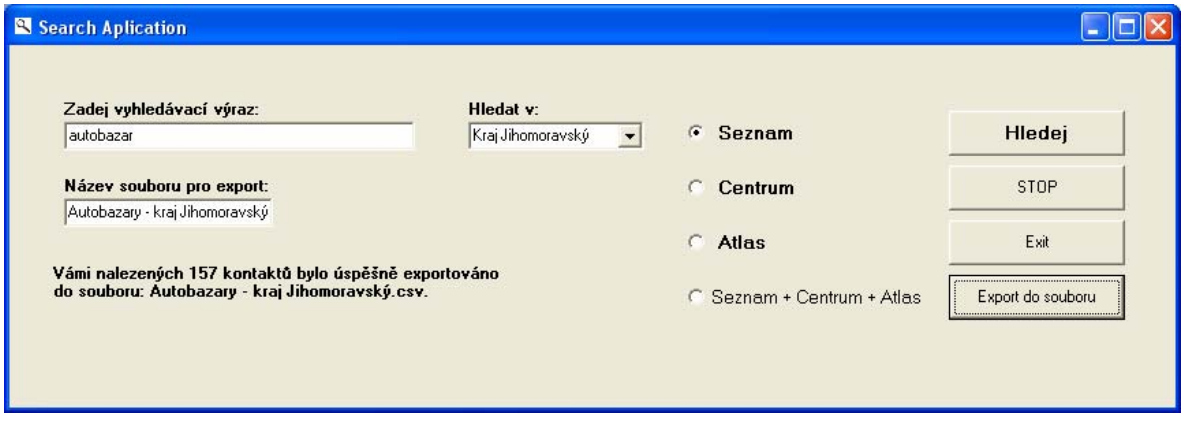
Obr. 19: Ukázka průběhu zpracování nalezených kontaktů

Po ukončení vyhledávání je nastavena viditelnost komponenty *CGauge* (ProgressBaru) na hodnotu *false*, vypsán skutečný počet nalezených kontaktů a také samotné kontakty (Obr. 20). Při samotném zpracování jsou rovnou odstraněny všechny duplicitní záznamy. Velké množství nalezených kontaktů si může uživatel prohlížet pomocí posuvníku (scroll) na pravé straně okna aplikace.



Obr. 20: Ukázka výsledků nalezených a zpracovaných kontaktů

Pokud je uživatel spokojen s výsledky hledání, má možnost stiskem tlačítka *Export do souboru* exportovat nalezená data do souboru pro vytvoření databáze kontaktů. Jak již bylo řečeno výše, vyexportovaný soubor je soubor typu CSV. Pokud uživatel nezadá jiný název souboru pro export (viz Obr. 21, políčko *Název souboru pro export*), je soubor implicitně pojmenován *databáze kontaktů*.



Obr. 21: Ukázka exportu nalezených kontaktů do databáze

S takto vytvořenou databází „si rozumí“ mnoho programů. Je možné ji otevřít v Microsoft Office Excel a zde s ní dále pracovat. Nebo pomocí několika málo jednoduchých kroků v programu Microsoft Office Word, lze z této databáze vytvořit štítky na dopisy pro hromadnou korespondenci.

6.3 Testování vytvořené aplikace

Při běhu aplikace bylo pomocí programu WildPackets Omnipeek Personal provedeno měření průběhu přenosu. Na průběh přenosu má vliv mnoho faktorů.

Kromě obligátních faktorů, jako je rychlost připojení k síti internet, průchodnost sítě a další, je to hlavně rychlost serveru, na kterém se stahovaná data nacházejí. Datový soubor nelze stahovat rychleji, než je serverem nabízeno. Servery katalogů, jež aplikace využívá, jsou nejvíce vytíženy během pracovního dne, kdy jsou jim adresovány požadavky od uživatelů ve zvýšené míře. Toto zatížení má vliv i na průběh aplikace. Tzn. že nepoměrně delší čas bude trvat aplikaci vyhledat a zpracovat stejný počet kontaktů získaných ze stejného serveru v denní špičku než v nočních hodinách, kdy je vytíženost těchto serverů minimální.

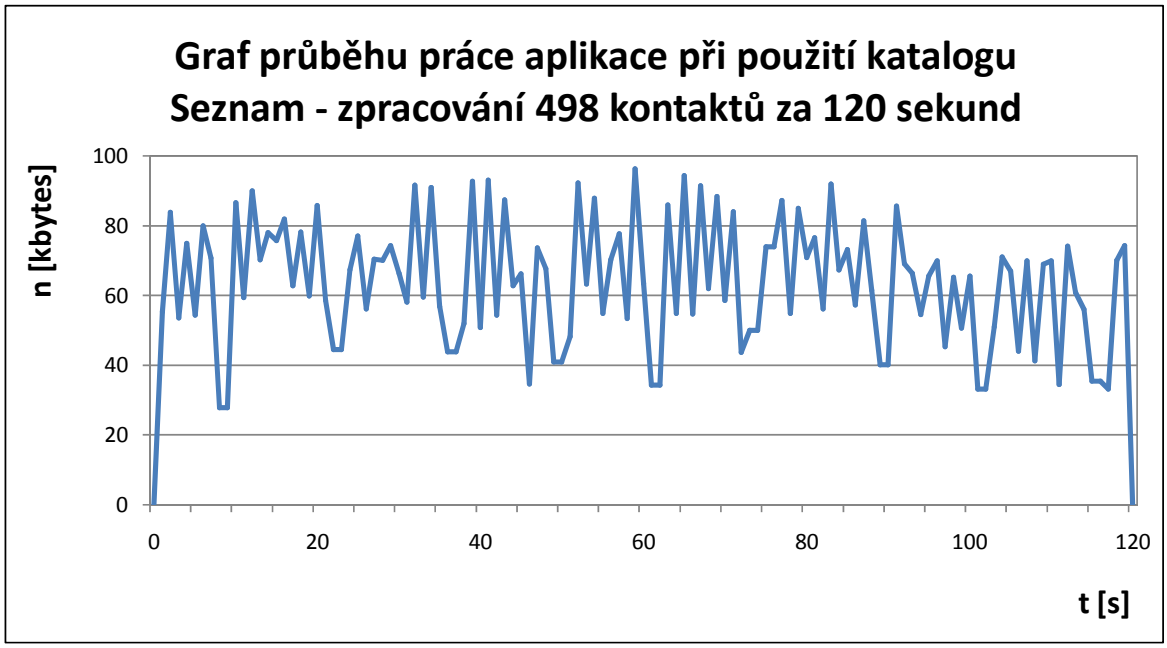
6.3.1 Testování jednotlivých katalogů

Měření, jehož výsledky jsou shrnuty v Tab. 4 a zobrazují je Obr. 22, Obr. 23 a Obr. 24, bylo provedeno okolo 19. hodiny pracovního dne, kdy vytížení serverů již není tak velké jako v denní špičce, ale není zase minimální oproti nočním hodinám.

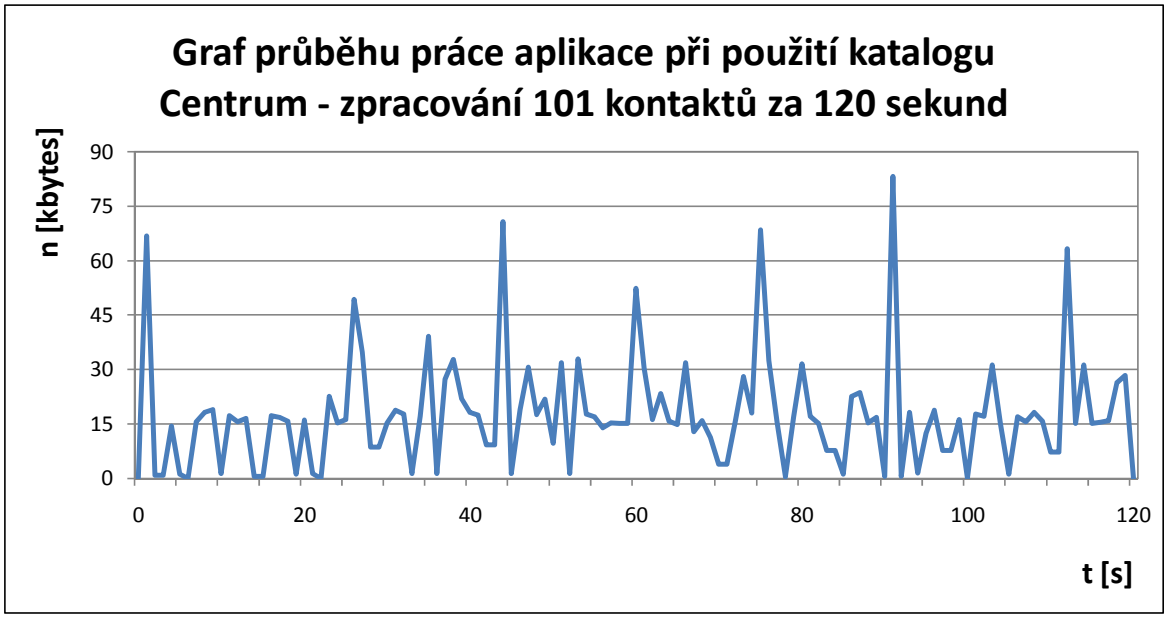
Tab. 4: Výsledky měření práce aplikace za 120 sekund pro různé katalogy

	Seznam	Centrum	Atlas
počet kontaktů [–]	498	101	445
průměrný download [kB/s]	63,825	17,933	67,651
celkový download [MB]	7,595	2,134	8,051

Z průběhů přenosů je patrné, že zatímco servery katalogů Seznam a Atlas pracovaly téměř stejně rychle a aplikace vyhledala a zpracovala 498, respektive 445 kontaktů za 120 sekund, server katalogu Centrum se ukázal mnohem pomalejší a za stejný čas umožnil aplikaci vyhledat a zpracovat jen 101 kontaktů. Tyto výsledky potvrzují zkušenosti při vývoji aplikace, kdy testování aplikace na serveru katalogu Centrum bylo velmi zdlouhavé.

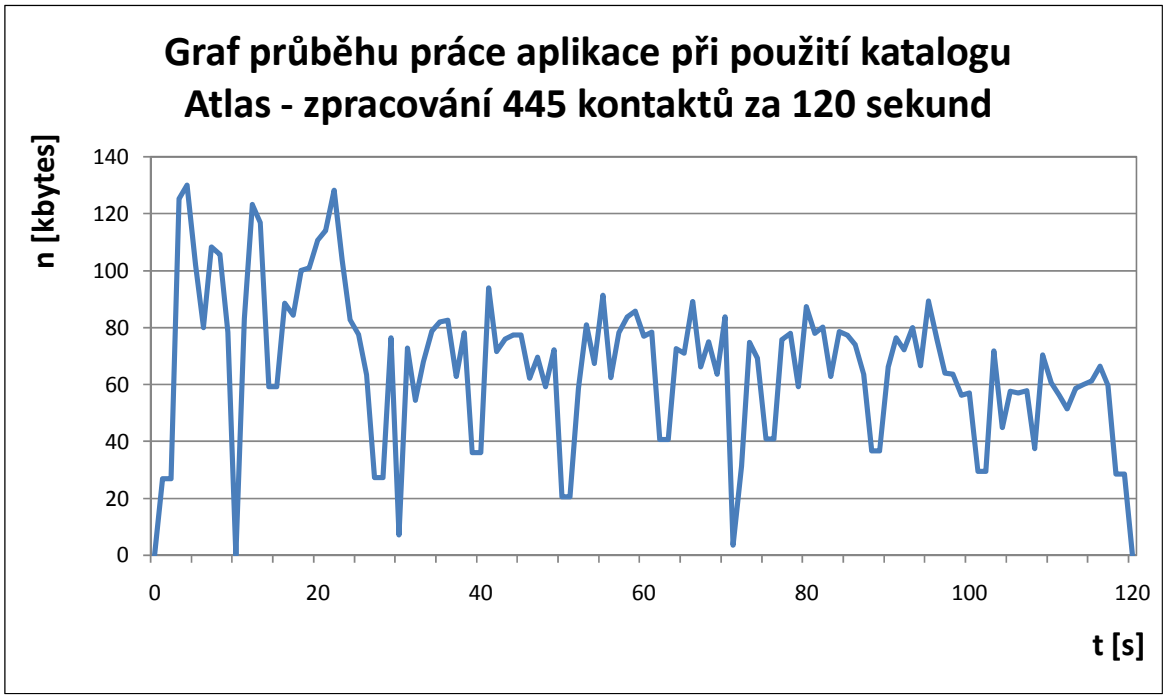


Obr. 22: Graf průběhu přenosu během práce aplikace při použití katalogu Seznam



Obr. 23: Graf průběhu přenosu během práce aplikace při použití katalogu Centrum

Z grafu průběhu práce aplikace při použití katalogu Centrum na Obr. 23 je patrné, že při downloadu každé další stránky s nalezenými kontakty je větší zatížení sítě (viz 7 největších „špiček“ v grafu). To je způsobeno tím, že zdrojový kód stránky s výsledky hledání je asi pětkrát větší než zdrojový kód stránky s kontaktními informacemi o konkrétní firmě.



Obr. 24: Graf průběhu přenosu během práce aplikace při použití katalogu Atlas

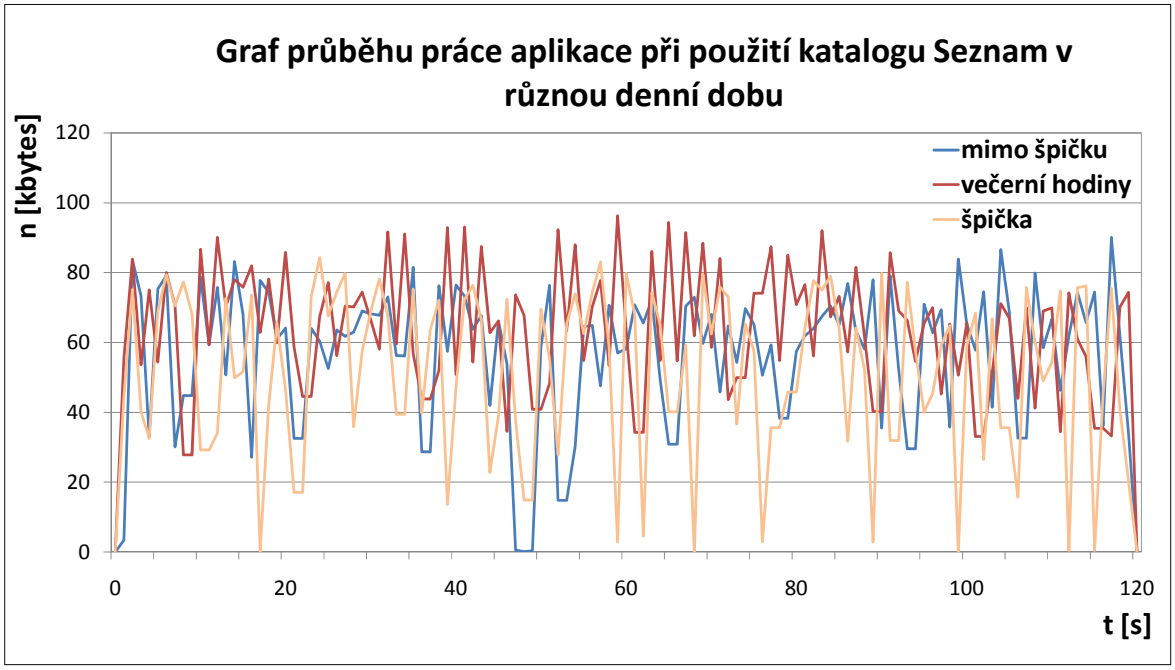
Sledováním zatížení procesoru při běhu aplikace bylo potvrzeno, že způsob správy programu v oddělených vláknech byl správnou volbou. Na zatížení procesoru má velký vliv, který z katalogů je aplikací právě využíván. Při pomalých odezvách serveru katalogu Centrum je zatížení procesoru minimální, zatímco při práci se servery katalogů Seznam a Atlas je průměrné zatížení při dlouhodobém měření 23 %.

6.3.2 Testování katalogu Seznam v různém zatížení

Měření, jehož výsledky jsou shrnuty v Tab. 5 a vyneseny v grafu na Obr. 25, ukazuje, že v porovnání s denní špičkou je aplikace schopna ve večerních hodinách zpracovat v průměru o více než 60 kontaktů za minutu více.

Tab. 5: Výsledky měření práce aplikace za 120 sekund pro katalog Seznam v různou denní dobu

	Špička pracovního dne	Mimo špičku pracovního dne	Večerní hodiny pracovního dne
počet kontaktů [–]	366	403	498
průměrný download [kB/s]	50,877	56,025	62,770
celkový download [MB]	6,156	6,779	7,595



Obr. 25: Graf průběhu práce aplikace při použití katalogu Seznam v různou denní dobu

7 Závěr

Tato práce byla vypracována k vytvoření uceleného přehledu a zmapování možností vyhledávání dat na internetu. Byly zde popsány vlastnosti a objasněny principy fungování centralizovaného, decentralizovaného a hybridního vyhledávání. Dále byly nastíněny možnosti využití inteligentních vyhledávacích agentů a popsán vývoj a práce vyhledávací aplikace, jež byla vytvořena jako praktická část této diplomové práce.

V první části této práce zabývající se problematikou centralizovaného vyhledávání, bylo nejprve provedeno rozdělení centralizovaných vyhledávačů podle způsobu uchování a prezentace dat a poté jednotlivé vyhledávací stroje dopodrobna rozebrány. Na závěr této kapitoly byly shrnuty výhody a nevýhody centralizovaného vyhledávání.

Ve druhé části práce byly popsány decentralizované vyhledávače a názorně předveden a vysvětlen princip jejich fungování. Na závěr kapitoly byly opět jako v předchozí kapitole shrnuty výhody a nevýhody tohoto vyhledávání.

Ve třetí části práce byl popsán princip hybridního vyhledávače, skloubení prvků centralizovaného a decentralizovaného vyhledávání a historicky nejstarší vyhledavač, využívající decentralizované technologie.

V poslední teoretické části práce jsou nastíněny možnosti využití inteligentních vyhledávacích agentů. Je zřejmé, že uvedení agentů a sémantických webů do praxe nelze provést naráz a má pozvolný průběh.

V závěrečné kapitole s názvem *Vytvoření vlastní vyhledávací aplikace* je naznačeno jakým způsobem probíhalo programování vyhledávací aplikace, která uživateli usnadňuje vyhledání kontaktů firem podle zadaného vyhledávacího výrazu a ze získaných dat následně vytvoření databáze kontaktů ve formátu CSV.

Po mnohém testování bylo potvrzeno, že způsob správy programu v oddělených vláknech byl správnou volbou. Při testování byly dále odhaleny chyby v korektní práci katalogů Centrum a Atlas. Konkrétně při vyhledávání v katalogu Centrum se při nalezení více jak tisíce výsledků nelze dostat za výsledky 990. Např. na vyhledávací dotaz "ubytování" nalezne v lokalitě celá ČR 10379 firem, ale při pokusu o zobrazení výsledků od 990 do 1005 (pomocí *&from=990*) katalog hlásí chybu, že na zadaný dotaz nebylo nic nalezeno. Katalog Atlas také nepracuje úplně korektně a při nalezení několika tisíc výsledků poslední výsledky stále opakuje. Např. na stejný vyhledávací dotaz "ubytování" nalezne v lokalitě celá ČR 10729 firem, ale při zobrazení výsledků od 1341 (pomocí *&p=68*) až do konce stále opakuje posledních dvacet nalezených firem. I přes opakovanou urgenci vývojářů těchto vyhledávacích katalogů nebyli schopni dané problémy za 3 měsíce odstranit. Bude to nejspíše tím, že zmiňované katalogy uvádí nepravdivé údaje o počtu nalezených relevantních odkazů, aby se alespoň přiblížily počtu výsledků nalezených na stejný vyhledávací dotaz v perfektně fungujícím katalogu firem Seznamu.

Vytvořená aplikace slouží uživateli k vyhledávání kontaktů a vytváření databází z takto nalezených dat. Je velmi efektivní z hlediska počtu zpracovaných dat v minimálním čase. Usnadňuje tedy práci při vytváření obsáhlých databází firemních kontaktů.

8 Seznam použité literatury

[1] JANOVSKÝ, Dušan. Google PageRank. [cit. 16.11.2007] Dostupný z WWW: <http://www.jakpsatweb.cz/seo/pagerank.html>.

[2] ISKRA, Jiří. *Google : Tipy a návody pro vyhledávač, Gmail, YouTube, Earth a další aplikace*. 1. vyd. Brno : Computer Press, a.s., 2008. 231 s. ISBN 978-80-251-1833-7.

[3] Seznam - nejpoužívanější český vyhledávač. [cit. 16.11.2007] Dostupný z WWW: <http://www.ataxo.cz/info/vyhledavace/seznam/>.

[4] Centrum - Nápověda. [cit. 16.11.2007] Dostupný z WWW: <http://napoveda.centrum.cz/index.php?root=93>.

[5] Morfeo - Nápověda. Dostupný z WWW: <http://morfeo.centrum.cz/index.php?napoveda=1&sec=mor>. [16.11.2007]

[6] PRESOVÁ, Silvie, PAZDERSKÝ, Michal, ŠKYŘÍK, Petr. *Jak hledat informace* [online]. 2007 [cit. 2007-11-16]. Dostupný z WWW: <is.muni.cz/elportal/estud/ff/js07/informace/materialy/pages/jak-hledat_opora.pdf>.

[7] BUREŠ, Miroslav, MORÁVEK, Adam, JELÍNEK, Ivan. *Nová generace webových technologií*. Řezníčková Michaela. 1. vyd. Praha : 1. VOX a.s., 2005. 264 s. ISBN 80-86324-46-X.

[8] ORAM, Andy. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. 1st edition. [s.l.] : [s.n.], 2001. 448 s. ISBN 0-596-00110-X.

[9] *Gnutella : Gnutella Specification 0.4* [online]. 2004 [cit. 2008-05-01]. Text v angličtině. Dostupný z WWW: <http://www.stanford.edu/class/cs244b/gnutella_protocol_0.4.pdf>.

[10] GUPTA, Indranil. *Peer-to-peer systems* [online]. 2007 [cit. 2008-05-01]. Text v angličtině. Dostupný z WWW: <www.crhc.uiuc.edu/~nhv/428/slides/p2p-I.ppt>.

[11] *The Free Network Project* [online]. 2006 [cit. 2008-05-01]. Dostupný z WWW: <http://freenetproject.org>.

[12] MATOUŠEK, Jiří, KLÍMA, Petr, KYNICKÝ, Pavel. *KaZaA, DirectConnect, eDonkey, bitTorrent a další : průvodce výměnou souborů přes internet*. Stanislav Závodný. 1. vyd. Brno : Computer Press, 2004. 190 s. ISBN 80-251-0465-6.

[13] THIRBODEAU, Patrick. *Sémantický web v kostce* [online]. 2000 [cit. 2007-11-16]. Dostupný z WWW: <http://www.scienceworld.cz/sw.nsf/0/D08DFE698E30CA80C1256E970048FCEE?OpenDocument&cast=1>.

[14] SVÁTEK, Vojtěch. *Ontologie a WWW* [online]. 2007 [cit. 2007-11-16]. Dostupný z WWW: <http://nb.vse.cz/~svatek/onto-www.pdf>.