Система за индексирано търсене в локална Windows мрежа

Христо Вълчанов, Димитър Тодоров

System for indexing and searching in local Windows network: *Sharing resources across local networks is a common practice to facilitate collaboration of multiple users. The availability of different machines on the local network leads to a very important problem - the difficult discovery of shared files and folders, which most often lies in the manual review of these folders to open certain files in them. This paper presents the architecture of distributed system for indexing and searching in a local Windows network.*

Key words: *Indexed search, Network search, Windows search.*

ВЪВЕДЕНИЕ

Развитието на редица технологии като безжични локални мрежи (WLAN) и виртуални локални мрежи (VLAN), както и ниските цени на десктоп компютрите, дадоха възможност за изграждане на локални мрежи, включващи от няколко до десетки машини. В този сегмент делът на операционната система Windows (в по-голямата си част версия 7) е изключително висок. Споделянето на ресурси в подобен род мрежи е често срещана практика, улесняваща съвместната работа на множество потребители. Наличието на различни машини в локалната мрежа обаче води до много съществен проблем - затрудненото откриване на споделените файлове и папки, което най-често се заключава в ръчно преглеждане на тези папки за откриване на определени файлове в тях. С развитието на компютърните технологии и системи, както и с нарастващия брой компютърни документи, е необходимо развитието и на системите за откриване и извличане на информация [1].

Операционната система Windows 7 разполага с вградена система за извличане на информация. Тя работи както с изградени индекси, така и без индексни файлове. Въпреки, че притежава подобрена функционалност в сравнение с Windows Vista, системата за търсене има някои недостатъци. На първо място е необходимостта от ръчно указване на папките, които да се индексират. По отношение на папките и файловете в мрежата, тяхното индексиране не се извършва автоматично, а при заявяване от потребител. Системата за индексиране и търсене работи в мрежа единствено със споделени ресурси, които за да се индексират трябва да се включат в режим offline. Тази функционалност зависи от версията на Windows 7, а при наличие на стари версии на Windows за това трябва да се инсталира допълнително пакетът Windows Search 4.0. Като факт, не е налична информация как точно Microsoft са реализирали процесите на индексиране и търсене.

Известни са редица безплатни алтернативни средства на индексиране и търсене под Windows. Някои от тях, като grepWin, MasterSeeker, FileSeek и Listary [3], въпреки подобрената им функционалност, са предназначени единствено за десктоп търсене. Съществуват и средства, които предоставят индексиране и търсене в споделени устройства в мрежа. Ускорено индексиране и търсене се предоставя от Locate32 [3]. Недостатък на софтуера е липсата на търсене в съдържанието на файл. LanHunt [8] е Java базирано приложение, използващо база данни за индекс, което ускорява процеса на търсене. Като недостатък може да се посочи търсене в съдържание единствено по единични думи. Други безплатни и гъвкави приложения са LanSearch Pro [8] и EveryThing [2]. Въпреки богатата им функционалност, търсенето при тях се свежда единствено до име или тип на файл.

В настоящия доклад е представена архитектурата на разпределена система за индексиране и търсене в локална Windows мрежа. Системата индексира освен имена на файлове и тяхното съдържание. Процесът на индексиране може да се изпълни както в оперативната, така и в дисковата памет. Системата позволява при търсене задаването на сложни булеви заявки. Достъпът до функционалността на системата е организиран на базата на потребителски групи.

АРХИТЕКТУРА НА СИСТЕМА ЗА ИНДЕКСИРАНО ТЪРСЕНЕ

Архитектурата на системата за индексирано търсене в локална Windows мрежа има разпределена организация. Тя се състои от множество независими услуги, стартирани върху отделни машини от мрежата (фиг.1-а).

Фиг. 1. *Архитектура на системата*

Услугата се състои от следните основни компоненти: подсистема за индексиране, подсистема за търсене и комуникационна подсистема (фиг.1-б). Всяка машина от мрежата съхранява само своите индекси. По този начин отпада нуждата от използване на сървърна машина, която да съхранява всички индекси на участниците в локалната мрежа. Като резултат се намалява обема на съобщенията през мрежата. Приложената стратегия позволява всеки компонент от локалната мрежа сам да менажира своите индекси. Всяка от услугите може да приема локални заявки за търсене, както и заявки от мрежата. Обменът на информация между отделните компоненти през мрежата се извършва от комуникационната подсистема.

ПОДСИСТЕМА ЗА ИНДЕКСИРАНЕ

Подсистемата за индексиране се грижи за изграждането на индексните файлове. Тя извършва обработката на заявката за индексиране, извличането на текст от документи, изпълнява алгоритъма за опростяване на думите и изграждането на индексите. Изграждането на индексите е реализирано по два начина: в оперативната и в дисковата памет.

Процесът на изграждане на индексите в оперативната памет започва с отстраняване на всички пунктуационни символи от поредния документ и като резултат връща изчистен текст. Отделянето на термините (думите) се реализира чрез генериран от Flex Lexical Analyzer код [4]. Кодът се генерира на базата на зададени условия, които трябва да бъдат удовлетворени (допускат се само символи от множествата [0-9, a-z, A-Z, а-я, А-Я]).

Преди термините да бъдат подадени на модула за опростяване, е необходимо да се провери техния размер. За да се избегне индексирането на малките думи, които са много на брой във всеки текст, не се допускат думи, които са с два или един символа. След това филтрираните термини се опростяват до техните корени посредством т.н. стеминг (stemming) алгоритъм [1]. Така опростените думи се записват в сортиран индексен списък (речник). Елемент от индексния списък съдържа термин и списък на документите, в които той се среща. Всеки елемент от списъка с документи се състои от идентификатор на документ, брой срещания на термина в този документ и списък с позициите на термина в документа.

Подсистемата за индексиране приключва като записва създадените индексни файлове в дисковата памет на машината, освобождава използваната оперативна памет и услугата преминава в режим на очакване на заявки.

Стъпките по индексация в дисковата памет са аналогични на индексирането в оперативната памет, но с някои различия. При този случай има твърдо фиксиран буфер, в който се записват формираните списъци. При запълване на буфера, той се

съхранява в дисковата памет, след което се изчиства и процесът по индексация продължава, докато не бъдат индексирани всички файлове в желаната директория и поддиректориите й. След като процесът по индексация завърши е необходимо да се сортира речника. Тъй като той може да не се побере в оперативната памет, се използва алгоритъм за външно сортиране. За да се ускори процеса на търсене, се изгражда специална карта на термините, която да съдържа първите им букви, начална и крайна позиция във файла. При постъпване на заявка за търсене се прави проверка в картата, от коя до коя позиция се намират думите, започващи с първата буква на заявения термин, след което речникът се зарежда на части в буфер.

ПОДСИСТЕМА ЗА ТЪРСЕНЕ

Подсистемата обработва получените заявки за търсене, които могат да бъдат както локално подадени, така и получени по мрежата от други машини. При постъпване на локална заявка тя се филтрира (постъпилите по мрежата заявки са предварително филтрирани) в следните стъпки:

- Премахват се всички пунктуационни символи, като се използва вътрешен списък с пунктуационни знаци. Всеки символ от заявката се проверява за наличие на някой от символите в списъка. При откриване на символ, той се изтрива и проверката продължава. Пропуска се само символът * (звезда), който е служебен и се използва при заявки с маска.
- След премахването на пунктуационните символи се проверяват всички думи. Ако се открие термин с размер по-малък или равен на два символа, той се пропуска. Изключение прави логическият оператор OR, който е служебна дума за системата.
- Всички останали думи преминават през стеминг алгоритъм за опростяване до техните корени.

Заявките могат да съдържат единични термини, фрази, дизюнкции и конюнкции от термини, сложни изрази, както и термини с маски. Всеки термин от заявката се проверява в списъка с изградените индекси.

КОМУНИКАЦИОННА ПОДСИСТЕМА

Комуникационната подсистема осигурява обмена на информация между изпълняваните на отделни машини услуги. Този обмен включва предаване на заявки за търсене към всички машини и изпращането на получения от търсенето резултат.

Реализирането на система за търсене в локална мрежа, където заявката за търсене на една машина ще се изпраща до и обработва от много машини паралелно, предполага висока натовареност на мрежата. Изборът на подходящ транспортен протокол е от голямо значение за ефективната работа на системата.

Реализацията на комуникацията в представената система е базирана на Winsock2 мултикаст [5]. Естеството на обменяните данни (думи, фрази за търсене, резултати от търсенето под формата на пътеки към файлове и самите файлове) изисква те да бъдат доставяни надеждно. Заложеният в Winsock2 мултикаст протокол за надеждна доставка е Pragmatic General Multicast (PGM) [6]. PGM е протокол от транспортното ниво за надежден мултикаст за приложения, които изискват подредено или разбъркано, без дубликати, мултикаст доставяне на данни от множество източници към множество получатели. PGM гарантира, че получател в група или получава всички пакети данни от предаване, или е способен да установи невъзстановима загуба на пакет данни.

С цел ограничаване на търсене от потребители в системата са въведени групи. Концепцията за група позволява гъвкавост в ограничаването на периметъра на търсенето, но сама по себе си не предоставя защита от прочитане на данните, които се предават от машини извън групата. За осигуряване на такава се използва концепцията за криптиране на мултикаст [7]. За разлика от класическата схема,

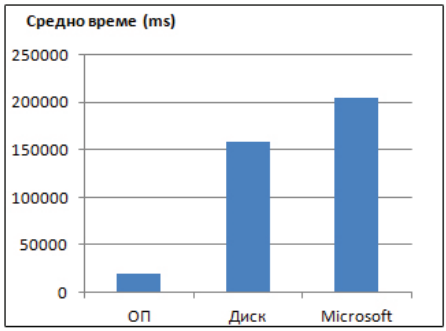
вместо да има един ключ за мултикаст групата, се въвежда по един ключ за всяка група от системата, наричан групов ключ. Всеки потребител, който желае да направи търсене, е длъжен да използва съответния групов ключ, за да криптира съобщенията, които изпраща. Поддържането на групови ключове, както и информация за членство в групи се реализира чрез специален централизиран компонент – автентикатор.

ЕКСПЕРИМЕНТАЛНИ ИЗСЛЕДВАНИЯ И РЕЗУЛТАТИ

Целта на изследванията е да се изследва функционалността на предлаганата система в реална локална мрежа и се сравни нейната ефективност с тази на вградената в Windows система за индексиране и търсене.

Експерименталната постановка включва свързани в 100 Mb Ethernet локална мрежа четири клиента и автентикатор. Клиентите са под операционна система Windows 7 Enterprise. Проведени са две групи тестове - за индексиране на зададена директория с обем 2,12 Gb и търсене на заявки на локалната машина:

- Индексиране на директория – броят на файловете е 2816, от които само 521 ще бъдат индексирани тъй като само техните формати се поддържат в текущата реализация (.txt, .html, .htm). Тестват се три типа индексиране: индексиране в оперативната памет, индексиране в дисковата памет и Windows индексиране.
- Локално търсене на заявки – извършва се търсене на три типа заявки:
  - Търсене чрез маска с тестова заявка - „app*“;
  - Търсене на единична дума - „application“;
  - Търсене чрез логически операции - „Linux AND Unix OR Windows“.

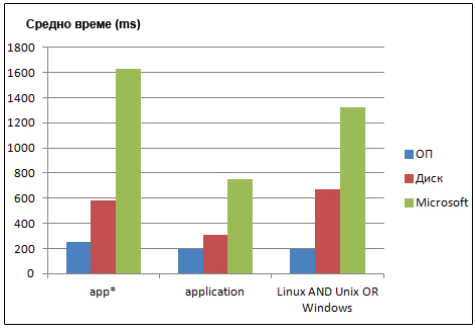


Фиг. 2. *Средно време за индексиране*

На фиг. 2 за показани резултатите от индексирането. При сравнение на средните стойности на резултатите, индексирането в оперативната памет дава най-добро време - 20087,33 ms, в сравнение с индексирането в дисковата памет - 158084,03 ms и Microsoft индексирането – 204850 ms. Натоварването на системата е много голямо при Microsoft индексирането и индексирането в оперативната памет. При индексирането в оперативната памет всички индекси се съхраняват в оперативната памет, което води до високото натоварване, докато за Microsoft няма сведения, как изграждат индексите. Трябва да се вземе и в предвид, че Microsoft индексирането е обработило много повече документи, за да изгради своите индекси, тъй като системата не позволява премахването на някои типове файлове от списъка с разширения за индексиране.

На фиг. 3 са показани резултатите от обработката за заявки. При сравнение на средните стойности на резултатите, се вижда, че обработка на заявка с маска в ОП (с дървовидна структура) се извършва два пъти по-бързо (250,98 ms) отколкото при

дискова памет (чрез многократно зареждане на векторен буфер) - 584,10 ms, а с най-голямо време е Microsoft търсенето. При търсене на единична дума и при логически изрази отново най-добри резултати дава търсене в ОП, алгоритъмът с векторен буфер е по-бавен, докато при Microsoft изпълнението е два пъти по-бавно от дисковото търсене.



Фиг. 3. *Средно време изпълнение на заявки за търсене*

ЗАКЛЮЧЕНИЕ

В настоящия доклад е представена архитектурата и базовата функционалност на система за индексирано търсене в локална Windows мрежа. Системата индексира освен имена на файлове, и тяхното съдържание. Процесът на индексиране може да се изпълни както в оперативната, така и в дисковата памет. Системата позволява при търсене задаването на сложни булеви заявки. Направени са експериментални сравнения и оценки на двата метода за индексиране на информация, както и търсене с вградената във Windows 7 система за индексиране и търсене. Резултатите показват, че е постигнато по-добро бързодействие от съществуващата реализация във Windows.

Като насоки за бъдеща работа се предвижда разработване на хибриден индексиращ алгоритъм, който включва предимствата на изграждането на индекси в оперативната и дисковата памет.

ЛИТЕРАТУРА

[1] Baeza R., Ribeiro B. Modern Information Retrieval: The Concepts and Technology behind Search. Addison-Wesley. 2011.

[2] EveryThing Search Engine. http://www.voidtools.com/ (август 2015).

[3] Five Alternative Search Tools for Windows. http://www.thewindowsclub.com/windows-search-alternative-tools (август 2015).

[4] Flex Lexycal Analyzer. http://flex.sourceforge.net (август 2015).

[5] Microsoft. Windows Socket 2 Architecture. https://msdn.microsoft.com/en-us/library/ windows/desktop/ms740650%28v=vs.85%29.aspx (август 2015).

[6] PGM Reliable Transport Protocol. https://tools.ietf.org/html/rfc3208 (август 2015).

[7] Shoufan A. High Performance Group Key Management. Akademikerverlag, 2012.

[8] Tools to Search Any Files in LAN. https://www.raymond.cc/blog/search-find-and-locate-any-files-on-local-area-network-shared-folders/ (август 2015).

За контакти:

доц. д-р инж. Христо Вълчанов, Катедра “Компютърни науки и технологии”, Технически университет-Варна, тел.: 052 383 424, e-mail: hristo@tu-varna.bg

Докладът е рецензиран.