

МЕТОДИ ОБСЛУГОВУВАННЯ ЗАПИТІВ АГЕНТОМ-БРОКЕРОМ

О.В. Шевченко

Київський національний університет імені Тараса Шевченка
03121, Київ, проспект Академіка Глушкова, 2, корп.6,
факс: 259-70-44, т.:521-33-66, sheva@unicyb.kiev.ua

Робота присвячена розробці алгоритму брокера, що обслуговує запити користувачів, на основі теорії масового обслуговування, моделі семантичного індексування інформації та методі обчислення вартості й оптимального розподілу витрат на отримання користувачем якісної відповіді.

The article describes the creation of broker's algorithm, based on queueing theory, model of semantic information indexing and method of calculating the cost and optimal distribution of expenses for high-quality reply received by user.

Вступ

Все більшої кількості людей доводиться регулярно здійснювати пошук інформації в Інтернет. У залежності від потреб користувача це може займати більше або менше часу, потребувати тих чи інших засобів автоматизації. Розподілена глобальна мережа містить усе інформаційне багатство сучасного суспільства, це, однак, створює проблему знаходження потрібних ресурсів, коли невідомі адреси їх розташування, тому ефективний пошук тематичної інформації в Інтернет за мірою зростання обсягів і кількості її джерел стає все більш складним і трудомістким. При цьому критичним є не тільки час пошуку необхідної інформації, а і відбір релевантної інформації та більш швидке формування результату пошуку [1–6]. Для багатьох людей пошук інформації в Інтернет поки що виконується разово або епізодично і користувача зовсім не цікавить, або рідко цікавить як змінилася інформація, що цікавила його раніше. Такий «разовий» пошук у мережі автоматизують відомі пошукові системи, які базуються на класичних методах розпізнавання, лінгвістичних методах і пошуку ключових слів. Разом з тим, зростання залежності людини від вірогідності, своєчасності і повноти інформації, що надходить з мережі, динамічність інформаційного середовища вимагає нових шляхів оновлення відомостей про наявність і розташування інформації у мережі. Для автоматизації довготривалого й постійного пошуку в Інтернет необхідно використовувати спеціалізовані прикладні програми, які відстежують зміни інформації, стосовно інтересів окремих користувачів.

Оскільки Інтернет використовує різноманітні ресурси й різні методи доступу до цих ресурсів, то системи пошуку документів за ключовими словами, авторами, тематикою постійно доповнювалися можливостями більш складних тематичних запитів. Такі системи надавали розподілений доступ до документів, що розташовані на Веб-серверах, на місцевих і віддалених серверах, до яких був можливий доступ через мережу. Для посилань на документи різних серверів створюються каталоги, як правило, спеціалізовані і структуровані за темами. Доступ до цих каталогів забезпечується шляхом вибору окремих тем із конкретних розділів. Складна, різноманітна структура даних і різноманіття методів доступу до них створює нові проблеми, пов'язані з необхідністю отримання за запитом більш точної тематичної інформації.

Одним з підходів для вирішення задач більш швидкого пошуку інформації є використання агентної технології [7–12]. Агенти відрізняються від існуючих поширених універсальних систем тим, що:

- спроможні самостійно виконувати завдання користувача (без уточнюючих звертань до користувача чи адміністратора) протягом тривалого часу (дні, тижні);
- один раз створений агент може бути використаний декілька разів у майбутньому, на відміну від запита універсальної системи, який збирає інформацію лише один раз.

У даний час не існує загально визнаного означення агентів. Як правило, агентами називають програми, що автоматизують пошук, розпізнавання, здобування та аналіз інформації, орієнтованої на потреби певних користувачів (груп користувачів) та демонструють автономність, реактивність (реакцію на зовнішні фактори), проактивність (дії з власної ініціативи) і, у випадку багатоагентних систем, цілеспрямовану групову поведінку. Агенти можуть бути: інформаційними, що створюють ядро тематичних індексів і запитів; такими, що виконують фільтрацію завантаженої тематичної інформації; агентами архіву документів, агентами по обслуговуванню проходження запитів за мережею, агентами інтеграції інформації про пошук, тощо [13–16]. У загальному випадку агенти отримують запити від користувачів з інформацією про необхідні інформаційні ресурси і вимоги до вартості пошуку. В даній роботі розглядаються алгоритм, принципи обслуговування, виконання запитів і оцінювання їх результатів головним агентом мережі – брокером.

Підхід до обслуговування брокером запитів користувачів

Робота брокера, що обслуговує запити користувачів при керуванні проходженням запитів за мережею, постановки їх у чергу за ресурсами мережі [18–24] базується на теорії масового обслуговування (ТМО) [17]; при функціональному виконанні запиту після виходу із черги – на використанні методу семантичного індексування інформації при формуванні індексів ядра фільтра документів і фільтра запитів для послідовного їх використання при пошуку релевантного запиту документа [4, 5, 13].

Запит до ресурсу мережі. Користувач формулює свою інформаційну потребу у вигляді запиту, який визначається на множині параметрів і має такий загальний вигляд:

$$Zap = \{ Z_p, T_p, P_d, St_p, R_p, K_p, Var \},$$

де $Z_p = z_1$ and z_2 and z_3 and z_i or z_j or, ..., ($i=1, \dots, l$) – ключові слова, зв'язані логічними операторами;
 $T_p = t_1$ or t_2 or t_3 and, ..., t_k , де t_i – терм документа для пошуку;
 P_d – параметри про документ d (розмір, дата створення, мова опису, форма збереження тощо);
 St_p – статус запиту (постійний чи одноразовий);
 R_p – пріоритет запиту;
 K_p – клас посилань;
 Var – параметр для оцінки вартості запитання тощо.

В запиті користувач задає контекст щодо деякої тематики за параметром Z_p , відомості про фіксацію документів у ядрі індексу та термів запиту в архіві запитів, а також параметр Var вартості отримання інформації із мережі.

Брокер виконує обслуговування запиту за допомогою різних агентів. Вони перетворюють формальний пошуковий вираз запиту у відповідну форму, зрозумілу всім учасникам процесу, співставляють вираз із запиту з описом тематичних документів у *ядрі індексів і запитів*, що зберігають інформацію про документи і їх посилання, ключові слова, ваги термів, обчислені за методами семантичного індексування або іншим методом, а також про дані запитів у ядрі запитів.

За допомогою індексів ядра виконується:

- розбір тексту нового документа, виділення ознак у даній предметній області та внесення їх в індекс для подальшого пошуку посилань на нові документи;
- порівняння даних із запиту з описом тематичного документа з архіву, а саме з посиланнями, ключовими словами, вагами термів документів та ін.;
- надання даних про всі доступні тематичні документи та відомості про останні зміни проіндексованих документів в ядрі індексів;
- здобування релевантного документа та інших даних, ранжирування результатів і формування відповіді.

Пошук документа релевантного заданому запиту може не дати результату або у відповідь може додаватися велика кількість релевантних запиту документів або один релевантний документ. У відповіді будуть зазначені такі дані, як ідентифікатор користувача, дата створення релевантного документа, розмір, анотація, статус, ідентифікатор цього документа та вартість пошуку. Все це користувач буде розглядати і приймати кінцеве рішення про результат пошуку.

Обслуговування запиту. Головну роль в цьому виконує брокер, який керує маршрутизацією запитів і його виконанням на основі ядер тематичних індексів і запитів. Процес обслуговування запитів складається з таких етапів:

Етап 1. Реєстрація агентом користувача, що звертається до мережі, у спеціальній таблиці;

Етап 2. Завдання необхідної інформації у запиті, відправка цього запиту в мережу для розгляду і обслуговування агентами;

Етап 3. Трансформація інформаційним агентом запиту у форму, прийняту іншим агентом процесу його обслуговування (брокером, менеджером і ін.).

Етап 4. Розгляд та просування запиту за маршрутом, визначеному агентом-брокером, та постановки запиту в чергу.

Етап 5. Обслуговування брокером черги запитів та оптимальний розподіл посилань у дереві посилань.

Етап 6. Фільтрація агентом нових документів для їх завантаження у відповідні архіви.

Етап 7. Збір інформації від усіх видів агентів про отримання результатів обслуговування запиту, передача їх агенту-інтегратору для видалення з них різних повторів і об'єднання та ранжирування посилань до тематики.

Етап 8. Розгляд користувачем отриманого результату щодо запиту та витрат на його виконання.

В обслуговуванні запиту приймають участь декілька агентів, діяльність яких повинна бути скоординованою. Кожний агент цього процесу працює за власною моделлю, що містить переконання про залежність деяких параметрів запиту до нього і даних, необхідних для виконання відповідної для нього частини запиту. До переконань входить набір концептів, відношень між ними, загальні оцінки термів документів та їх ваг, що зберігаються в архівах і до яких має доступ агент, час останнього звернення до цього архіву тощо.

Запити надходять до серверу мережі з деякою середньою швидкістю, зберігаються в буфері сервера в порядку їхнього надходження в чергу і за ними розраховуються витрати щодо прийому запиту на сервер,

здобування запрошеної інформації і відправлення результату користувачеві. Ступінь завантаження сервера впливає на витрати і визначається часткою загального часу обслуговування запиту [21].

Процес обслуговування запитів починається з маршрутизації запитів за мережею, постановку їх в чергу або безпосередньо виконання запиту. При цьому визначається час проходження і виконання запитів у мережі з урахуванням знаходження запиту у черзі. Брокер направляє маршрут запиту до сервера ресурсів, координує роботу різних агентів, що беруть участь у виборі індексів тематики, котрі відповідали б інформаційним потребам користувача [18, 21, 22]. Загальна схема обслуговування запиту і застосованих ресурсів показана на рисунку.

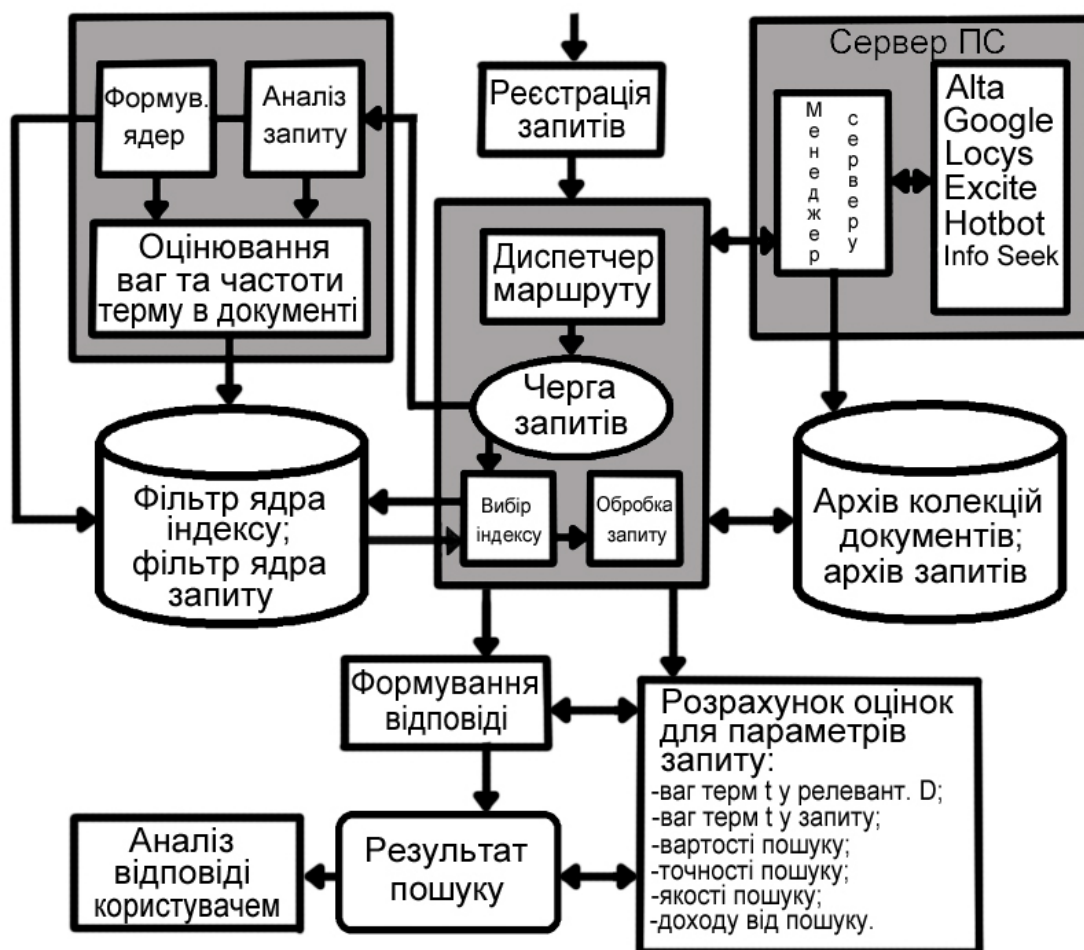


Рис. Загальна структура системи обслуговування запитів

Маршрутизація запитів користувачів здійснюється на мережному рівні за допомогою протоколів EGP (Exterior Gateway Protocol) або IP (Internet Protocol) і полягає у забезпеченні руху запитів від користувача до сервера інформаційного ресурсу. Кожен запит проходить маршрут – шлях проходження запиту від користувача до агента менеджера сервера, що забезпечує постановку запиту у чергу і розрахунок часу обслуговування [18-22].

Визначення серед різних маршрутів оптимального (найкоротшого) на даний момент відстані від одного користувача до сервера залежить від топології мережі, метрик часу і вартості шляху з урахуванням довжини, числа транзитних вузлів і виникаючих затримках при русі запиту за мережею.

При визначенні оптимального маршруту використовуються відомі методи [23, 24], за допомогою яких обчислюються розрахунок загальних витрат на маршруті і вибір оптимального маршруту за ітераційним процесом визначення маршруту від одного вузла мережі до всіх інших на шляху слідування запиту. Отримані витрати використовуються при кінцевому формуванні вартості та доходу від виконаного запиту.

При одночасному відправленні різних запитів до менеджера інформаційних ресурсів може бути створена черга запитів, для обслуговування якої запропоновано *два класи моделей обслуговування черг* запитів з одним сервером і N-серверами, які визначені на експоненційному, пуассонівському або детермінованому законах розподілу. Для обох класів моделей наведені загальні параметри: l – середнє число запитів у черзі; T_1 – середній час чекання запитів у черзі, ρ – ступінь завантаження, q – середнє число запитів усієї мережі; T_q – середній час,

протягом якого запити очікують своєї черги або знаходяться в обслуговуванні; λ (у сек.) – середня швидкість надходження до черги. Степінь завантаження ρ черзі впливає на витрати щодо запиту часткою загального часу, протягом якого обслуговується запит у черзі.

Перший клас моделі обслуговування запиту має максимальну швидкість надходження і обробки запитів на одному серверу: $\lambda_{\max} = 1/T_s$, де T_s – середній час обслуговування запиту з моменту його надходження до черги, функціонального виконання по знаходженню необхідної інформації та відправлення результату користувачеві. Розмір черги запитів зростає при $\rho = 1$ і час чекання обслуговування, як свідчать джерела, обмежується швидкістю надходження запитів у черзі понад 70 %.

Другий клас моделей характеризується наявністю N – серверів і кожний з них не завантажується до межі. Середня швидкість надходження запитів $\lambda = N/T_s$. Якщо усі сервери зайняті, то запити надходять в одну загальну чергу. При вивільненні одного із серверів, чергу залишає оброблений запит і з черги надходить наступний запит для обробки. У будь-який момент часу кількість запитів q дорівнює сумі кількості запитів, що очікують обслуговування, і кількості запитів, що знаходяться в даний момент часу в обслуговуванні, тобто $q = 1 + N \cdot \rho$.

Функціональне виконання запиту. Після свого виходу із черги запит виконується шляхом співставлення термів запиту з індексами документів ядра, що зберігаються разом з їх вагою, відповідними термами у цьому ядрі і застосовуються при знаходженні документа, релевантного запиту, а також при проведенні оцінки числа документів у колекції документів, ваг термів архіву колекції документів, точності, усередненій за усіма запитами користувачів при виборі тем та розрахунків за отриманий дохід.

Умовами пошуку тематичної інформації є мінімізація витрат на отримання посилань на документи індексу ядра індексів документів та оптимальний розподіл вартості формування відповіді на запит, яка залежить від обсягу і якості інформації в індексі, а також від стратегії забезпечення індексом пошуку. Коли є декілька індексів, тематика яких близька (схожа) до тематики запиту, брокер оцінює число релевантних документів, котрі треба отримати від кожного індексу, точність пошуку за розцінками індексів і вартістю доставки документів до користувача.

Брокер виконує розподіл доходів шляхом перерозподілу запитів між різними тематично близькими індексами. На цей розподіл впливає поведінка володарів тематичних індексів, які рекламують свої послуги, спираючись на інтереси користувачів до їхньої системи пошуку, а також можливість самостійно обслуговувати користувача поліпшеними послугами, що призводить до підвищення вартості, яка, як правило, не задовольняє користувача.

Архітектура і алгоритм брокера

Архітектура брокера базується на наявності наступних загальних об'єктів:

- користувач, що надає брокеру запит з декількох ключових слів, число посилань на документи, котрі він бажає отримати у відповідь на запит, а також дохід від отримання релевантного документа з виплатою рахунку за індекс. Умовою отримання доходу є добування релевантного документа з вартістю, що відповідає пошуку одного посилання і задовольняє користувача;
- сервер (або сервери) інформаційних ресурсів, до яких прямують запити від користувача, коли до сервера їх доходить забагато, вони ставляться в чергу і за мірою обробки попереднього запиту обслуговується наступний;
- репозитарій – сховище, де зберігається опис колекції документів, тематичні індекси та терми документів, які постійно оновлюються при аналізі нових ресурсів за запитами. Цю інформацію брокер використовує тоді, коли запит, після його маршрутизації, вийшов з черги з отриманою спеціальною ознакою типу ресурсу для пошуку в індексі необхідних документів. За кожним індексом документів репозитарій зберігає інформацію: опис індексу з множиною термів, їх ваги та деяку статистику, а також вартість посилання на один документ або витрати на індексування документів, пошук і відправку результату користувачеві.

Дохід залежить від числа направлених запитів. Брокер виконує запит з урахуванням інформації про індекс, що є у репозитарії, забезпечує контроль і співставлення результатів з отриманням вигляду опису індексу. При цьому система може завантажитися зайвою роботою, або володарі індексів можуть включати невірні дані у систему і тоді даний індекс може отримати великий дохід, а користувач – невелику якість відповіді. Але брокер повинний забезпечити якість і точність пошуку релевантного документа.

Архітектура брокера. Побудована архітектура брокера (рисунок), містить програмні компоненти та інформаційні дані, які забезпечують виконання задач пошуку релевантного документа за індексами, що зберігаються у репозитарії.

Дамо стислий опис головних компонентів побудованого брокеру.

Компонент – *аналізатор запитів* аналізує поданий запит, оцінює число посилань на документи, що релевантні запиту, котрі може дати кожний з тематичних індексів репозитарія. Для цієї оцінки використовуються параметри запиту, опис усіх індексів цього репозитарія, їх ваги та середня довжина документа, посилання на котрий зберігає індекс.

Компонент – *диспетчер* на основі інформації, отриманої від аналізатора, оцінює оптимальне число документів, котрі необхідно запитати у кожного індексу колекції документів.

Вхідними даними для диспетчера є число запитаних посилань, число релевантних запиту документів, загальне число посилань, штраф за отримання не релевантного документа, дохід від отримання посилання на релевантний документ та вартість відповіді на запит. Результатом його роботи – оптимальне число документів, що були запитані у кожного індексу.

Підсистема обробки запитів і відправки результатів встановлює зв'язок з тематичними індексами й на основі опису відповідної колекції документів отримує оптимальне число документів від диспетчера, виконує вибір необхідного індексу, розглядає документи, що запитувалися у кожного архіву тематичного індексу, після чого здійснює попередню обробку результату (ранжирування, кластеризація тощо) і формування кінцевого результату для подачі користувачеві.

Алгоритм брокера. Розроблений алгоритм брокера призначений для вирішення задач, пов'язаних з оцінками результатів пошуку тематичної інформації стосовно запитів користувачів, а саме:

- числа документів, релевантних запиту користувача в колекції документів;
- ваги термів у запиті;
- ваги термів у індексі документів;
- функції сили зв'язку документа і терму за методом семантичного індексування;
- точності пошуку тематичної інформації у колекції документів;
- якості пошуку кінцевої вартості отримання посилань із індексів колекції документів;
- загальної вартості отримання результату.

Змістовний короткий опис вирішення цих задач надається далі.

Оцінка числа документів у індексі D_i релевантних запиту q обчислюється за формулою:

$$R_i = C \sum_{t \in q} W_{t,i} \cdot V_{t,i}, \quad (1)$$

де $W_{t,i}$ – вага терму t в запиті q , C – деяка константа. Величина $V_{t,i}$ обчислюється як число входжень термів t в документ d , поділене на середнє число термів у документі колекції D_i і число документів в D_i .

Оцінка ваги терму в індексі D_i обчислюється за формулою:

$$W_{t,i} = (k' + 1) f_t^i / (k' \cdot L_i^1 + f_t^i), \quad (2)$$

де f_t^i – число входжень терму t у запит q ; L_i^1 – число термів у запиті q поділене на середнє число термів в документі D_i ; k' – константа, що забезпечує регулювання величини f_t^i , L_i^1 , зокрема при $k' = 0$, величина $W_{t,i} = 1$. При більших значеннях k' величина $W_{t,i} = f_t^i / L_i^1$.

Залежність між вагою терму запиту й індексом, куди спрямований запит зростає при зменшенні довжини запиту, коли індекс D_i , частота f_t^i терму t у запиті q фіксовані або зростає при направленні даного запиту в індекс, коли вага терма t з посиланням на документи має велику довжину від числа термів, що входять до нього.

Оцінка ваги терму в запиті обчислюється за формулою (2) і з додатковим обчисленням величини $\log((N_i - n_{t,i} + h) / (n_{t,i} + h))$, в якій N_i – число документів у колекції D_i , $n_{t,i}$ – число документів, які містять терм t в колекції D_i , $h = 1/2$. Цей логарифм пов'язаний з особливістю класичної імовірнісної моделі семантичного індексування, що використовується при обчисленні.

Якість і точність пошуку. Задача оцінки якості пошуку індексу відповідно запити орієнтована на оптимальний вибір колекцій документів, куди буде спрямовано запит користувача при виході з черги до ресурсу. Якість пошуку в індексі D_i забезпечується оцінкою для кожного індексу функції

$$EP_i(s) = P_i \cdot R_i / (R_i + s \cdot P_i), \quad (3)$$

що залежить від числа посилань R_i на релевантні документи в індексі D_i та P_i імовірності того, що перший документ з колекції D_i релевантний запити. Ця функція задає математичне очікування долі релевантних документів (точність) серед перших s документів. Коли s зростає від 0 до N_i , ця функція монотонно спадає.

Це пов'язано з тим, що кожний індекс у процесі пошуку виконується за відповідною методикою, базованою на відомих моделях індексування інформації. Взагалі кожна схема пошуку є моделлю поведінки експерта, який оцінює релевантність деякої множини документів щодо фіксованого запити з ранжуванням документів за ступенем спадання їх релевантності. Чим ближче знаходиться документ до початку ранжованого списку, тим вище імовірність релевантності.

При збільшенні довжини початкової частини в цьому списку (величина s) доля списку документів, що входять в цю частину, монотонно спадає і їх *точність* близька до 1. Коли довжина частини ранжованого списку наближається до числа усіх посилань N_i у індексу, точність практично ставиться рівною нулю.

Функція вартості пошуку обчислюється за формулою:

$$EC_i(s) = s (B_i^d + B^-) - s EP_i(B^- + B^+), \quad (4)$$

де B_i^d – вартість отримання одного посилання з індексу D_i з урахуванням всіх витрат по ньому; B^- – штраф за отримання не релевантного посилання; B^+ – дохід від отримання посилання на релевантний документ.

Ця функція є математичним очікуванням кінцевої вартості отримання s посилань з індексу D_i , і відповідних їм завантажених документів.

Оцінка вартості $var_{i,k}$ отримання k -го документа із індексу D_i має вигляд:

$$var_{i,k} = EC_i(k) - EC_i(k-1) \quad (5)$$

Твердження. Оптимальна вартість пошуку виконується за умови

$$var_{i,k} \leq var_{i,k+1} \text{ для усіх } i = 1, \dots, n \text{ і усіх } k = 1, \dots, n_d, \text{ коли } B^+ \geq B^-.$$

Зміст цього твердження полягає в отриманні оптимального рішення за критерієм отримання мінімуму вартості пошуку за формулою $var = \sum_{i=1, \dots, n} EC(s_i)$ за умовою – користувач не оцінює корисність від отримання не релевантного документа, а тільки від релевантного документа.

Таким чином, виконане співставлення двох моделей використаних для опису колекцій документів, необхідне для вибору більш оптимальної з них. Перша модель $f_{t,d}$ – частотна, сила зв'язку терму і документа якої оцінювалася числом входжень даного терму t у даний документ d . Друга модель – модель семантичного індексування оцінювалася частотою входження терму t в документі d . Результати співставлення цих двох моделей показали перевагу моделі семантичного індексування і успішне використання функції $f_{t,d}$ при оцінці точності вибору індексів і роботи брокера.

Висновки

Розроблено і досліджено алгоритм брокера, що обслуговує запити, і виконує маршрутизацію запитів для забезпечення швидкого і ефективного пошуку тематичної інформації, та фактично виконує запит. На основі теорії масового обслуговування запропоновано два класи моделей обслуговування черг запитів з одним сервером (N серверами).

Побудований алгоритм брокера базується на методі обслуговування запитів та підході, при якому індекс інформаційного ресурсу до якого був спрямований запит, повертає як результат релевантний документ за таких умов:

- *мінімізація витрат* на отримання посилань у ядрі індексу документів;
- *оптимальна вартість* доставки відповіді на запит;
- *штраф* за не релевантний документ;
- *дохід* від отримання релевантного документа.

1. Банков В.Д. Интернет: поиск информации и продвижение сайтов. – Спб. – Питер. – 2000. – 288 с.
2. Курник А.П. Поиск в Интернет. – Спб. Питер. – 2001. – 272 с.
3. Андон П.Л., Дерезький В.О. Процесори пошуку та аналізу природо мовної тематичної інформації в аналітичних системах // Проблеми програмування. – 2001. – № 3-4. – С. 144 – 165.
4. Deerwester Scott C., Dumais Susan T, Landauer Thomas R., Furnas George W. and Harshman R. A. Indexing by latent semantic analysis. – J. of The American Society of information Science. – 1990. – № 41(6). – P. 391 – 407.
5. Danzig P., Li S. and Obraczka K. Distributed indexing of autonomous Internet services. – Computing Systems. – 1992. – №5(4). – P. 433 – 459.
6. Gudivada V.N., Raghavan V.V. Information Retrieval on the World Wide Web // IEEE Internet Computing. – 1997. – 1, № 5. – P. 58 – 68.
7. Maes P. The agents Network Architecture. – SIGART Bulletin 2(4), 1991. – P. 112 – 120.
8. Huber M.J. JAM: a BDI – Theoretic mobiles agent architecture:International Conference on Autonomous Agents (Agents-99). – 1999, ACM Press, WA USA.
9. Плескач В.Л., Рогущина Ю.В. Агентні технології – К.: КНТЕУ. – 2005. – 337 с.
10. Дрейган Р. Будущее программных агентов.– PC Magazine March 25, 1997. – 190 с.
11. Riechem D. Intelligent Agents // SACM, 1994. – 37, № 7. – P. 20 – 31.
12. Трахтенгерц С.А. Взаимодействие агентов в многоагентных средах // Автоматика и телемеханика.– М.: Наука, 1998. – № 8. – С. 3 – 52.
13. Шевченко А.В. Підходи до здобування тематичної інформації із мережі Інтернет // Вісн. Київськ. ун-ту. Сер. фіз.-мат. науки, 2006.– № 1. – С. 56 – 60.
14. Tigue J., Lavinder J. WebBroker: Distributed Object Communication on the Web. – W3C NOTE 11-May-1998. – <http://www.w3.org/TR/1998/NOTE-webbroker-19980511>
15. Tomasic A., Gravano L., Lue C., Schwarz P., and Haas L. Data Structures for Efficient Broker Implementation // ACM Transactions on Information Systems, 1997. – № 15 (3). – P. 223 – 253.
16. Agent software examples.- <http://www.agents/tools/index.html>
17. Коваленко И.Н. Введение в теорию массового обслуживания и ее приложения. – М.: Наука, 1966. – 34 с.
18. Джерла М. Маршрутизация и управление потоком // Протоколы и методы управления в сетях передачи данных. – М.: Радио и связь, 1985. – 485 с.
19. Никитин А.И., Алиев А.А. Синхронизация процессов в распределенных системах обработки данных // Институт математических машин и систем НАН Украины. - Киев: 1995. – 127 с.
20. Клейнрок Л. Вычислительные системы с очередями.– М.: Мир, 1979.– 660 с.
21. Кульгин М. Технология корпоративных сетей. Энциклопедия, Питер, 2000. – 699 с.
22. Хелеби Сэм, Денни М.Ф. Принципы маршрутизации в Internet. 2-е издание. – Пер. с англ. – М.: Изд. Дом «Вильямс», 2001. – 448 с.
23. Дейкстра Э. Взаимодействие последовательных процессов // Языки программирования. – М.: 1972. – С. 9 – 86.
24. Ноаре С.А.Р. Communicating Sequential Processes. – Prentice Hall, UK, 1985. – 256 p.