|  |  |
| --- | --- |
| Національний університет  "Києво-Могилянська Академія" | |
|  | Департамент комп‘ютерних технологій  Кафедра інформатики |
| **Проблеми** **використання роботів  в Інтернеті** | |
|  | Реферат з курсу "Програмне забезпечення інтелектуальних систем" студента ДКТ-5 Підгорного Свєтослава |
|  | Викладач: Гороховський С. С. |
| Київ 2011 | |

Вступ 3

Використання роботів 3

Статистичний аналіз 4

Підтримка 4

Дзеркала 5

Пошук ресурсів 6

Комбіноване використання 7

Ціна використання та застереження 7

Ресурси мережі та завантаження серверів 7

Оновлення 8

Погані реалізації 10

Ведення каталогу 11

Забагато матеріалу, що є забагато динамічним. 11

Визначення що індексувати, а що ні. 11

Статистичний аналіз 13

Етика 13

Альтернативи для пошуку ресурсів 15

Висновок 16

Література 18

# Вступ

Всесвітня павутина стала дуже популярною за останні декілька років, і є зараз основним засобом розміщення інформації в інтернет. Коли розмір WWW збільшився від декількох десятків сайтів, стало зрозуміло, що людина не в змозі самостійно переглянути таку кількість сайтів, тим паче знайти нові.

Тому почалися розробки програм, що автоматично переглядають WWW. Така програма-робот передивляється Інтернет сторінка за сторінкою, йдучи за посиланнями на інші сторінки. Роботів також називають павуками (spiders) або черв‘яками (worms), але ці назви дають хибне враження, що роботи пересуваються самі, або, що вони копіюють себе, як відомі інтернет-черви. Насправді, робот – це одна програмна система, що отримує інформацію з Інтернет сайтів використовуючи стандартні протоколи WWW.

# Використання роботів

Переглядаючи Інтернет, роботи можуть виконувати такі функції:

## Статистичний аналіз

Найперші роботи були створені для підрахунку кількості WWW-серверів. Сучасні роботи можуть також підраховувати іншу статистичну інформацію, як то, середню кількість документів на один сервер, пропорцію заданих типів файлів, середній розмір веб-сайту, ступень міжсполучення між сайтами.

## Підтримка

Одна з основних проблем підтримки Інтернет виникає, коли перестають працювати посилання на інші сторінки. Це відбувається тоді, коли сторінка, на яку посилаються, перемістилася на іншу адресу або взагалі перестала існувати. Нажаль, натепер не існує автоматичного механізму, що повідомляв би веб-мастерів про такі зміни.

Деякі сервери, наприклад CERN HTTPD, записують у файл журналу, запити до сторінки, що не існує, з адресою сторінки, звідки йшло посилання, що дозволяє виправити ситуацію пізніше. Але це не практично, і, насправді, автори дізнаються, про неправильні посилання на своїх сторінках, коли помітять самі, або коли користувачі повідомлять їх по пошті.

Робот, що перевіряє посилання, такий як MOMspider, допомагає автору знайти такі неправильні посилання, тому допомагає підтримувати сайт. Роботи можуть допомагати підтримувати зміст так само, як і структуру, первіряючи правильність HTML коду, регулярність оновлення інформації, але це використовується не дуже часто. Цей вид функціональності часто є вбудованою частиною HTML редакторів, але, роботи можуть повторювати ці перевірки при кожній модифікації сайту, і будь-які проблеми можуть бути вирішені негайно.

## Дзеркала

Дзеркальність – популярна техніка для підтримки FTP архівів. Дзеркало містить копію усього дерева каталогів FTP сайту і регулярно оновлює ті документи, що додаються до оригінального сайту. Це дозволяє зменшити трафік до кожного з вузлів, їх завантаженість, працювати з архівом, якщо оригінальний сервер не працює, мати швидкий та дешевий локальний доступ та доступ без підключення до інтернет.

Дзеркальність може бути легко забезпечена роботом. Вже існують роботи, що отримують дерево каталогів веб-вузлу та записують його локально, але вони не мають можливості оновлювати тільки ту інформацію, що змінилася. Необхідність таких засобів зменшилася з появою складних кеш-серверів, які можуть робити вибіркове оновлення, та гарантувати, що документ з кешу є оновленим.

## Пошук ресурсів

Мабуть, найбільш захоплюючим використанням роботів є пошук нових ресурсів. Там, де люди не можуть впоратись з величезною кількістю інформації, комп‘ютер зробить цю задачу швидко й якісно. Існує багато роботів, що оглядають великі частини WWW та дають доступ до зібраної інформації за допомогою пошукової системи.

Це значить, що користувач може одночасно використовувати перегляд та пошук для знаходження потрібної інформації. Навіть, якщо база даних не містить саме ту інформацію, що вам потрібна, ймовірно вона містить сторінки на яких є посилання на потрібну.

Другою перевагою є те, що ці бази даних можуть оновлюватися періодично, і посилання на сторінки, що не існують, можуть бути знайдені та видалені. Це добра альтернатива ручній підтримці документів, де перевірки бувають рідкі та не глибокі.

## Комбіноване використання

Один робот може виконувати декілька задач. Наприклад RBSE Spider робить статистичний аналіз отриманих документів та видає базу даних знайдених ресурсів.

# Ціна використання та застереження

В цьому розділі ми побачимо, як роботи можуть завдавати шкоди, та як цю шкоду можна мінімізувати.

## Ресурси мережі та завантаження серверів

Роботи потребують чималого трафіку. Перші роботи працювали протягом тривалого часу, інколи місяцями. Щоб прискорити виконання завдання, деякі роботи працюють у паралельному режимі, чимало завантажуючи трафік. Навіть віддалені сервери можуть відчути заповільнення роботи, якщо робот отримує велику кількість документів за короткий час. Це призводить до зменшення швидкості роботи інших користувачів, особливо на повільних лініях.

Деякі корпоративні користувачі можуть вбачати пряму залежність між видатками на Інтернет та завантаженістю мережі. Ці видатки окуплять себе для потенційних користувачів, але не окупить використання мережі роботами.

Крім ставлення високих вимог до мережі, роботи можуть ставити високі вимоги серверам. У залежності від частоти отримування документів з серверу, одночасне завантаження багатьох документів може призвести до відчутної завантаженості серверу, що призведе, у свою чергу, до зниження рівню сервісу для інших користувачів, цього ж серверу. У випадку, коли сервер використовується також з іншими цілями, це може бути взагалі не допустимо.

Тому, робот не може часто отримувати документи з одного серверу. Навіть сучасний браузер Netscape страждає на цю проблему, отримуючи вбудовані у сторінку малюнки одночасово. HTTP протокол не пристосований до паралельного отримування документів, тому ведуться розробки нових протоколів.

## Оновлення

Було сказано, що бази даних про документи Інтернет мусять оновлюватися регулярно, але ж не існує механізмів контролю за змінами у WWW. Не існує єдиного запиту, що визначав би які з заданих URL‘ів знищені, модифіковані або переміщені.

Але, HTTP протокол разом з запитом даних дозволяє задати дату цього ж документу з кеша. Сервер передасть документ тільки тоді, коли документ було змінено з часу коли він був записаний у кеш.

Ця можливість може бути використана роботом лише тоді, коли він зберігає зв‘язок між даними, які він отримує з документу, його URL та часом отримування. Це ставить додаткові вимоги до розміру та складності бази даних, і застосовується не часто. Завантаженість мережі грає особливу роль, коли роботи використовуються кінцевими користувачами, що працюють на повільних лініях.

Інший небезпечний аспект клієнтських роботів: коли робот розповсюджений — ніякі помилки не можуть бути виправлені, ніякі інші дані не можуть бути додані і нові можливості не можуть бути реалізовані тому, що не всі оновлять версії своїх роботів.

Але найбільш небезпечним є те, що хоча деякі люди будуть використовувати роботів розумно, тобто не перевищувати який-небудь максимум, будуть люди, що зловживатимуть своїми можливостями, через незнання чи самонадійність.

"Розумні агенти" або "персональні асистенти" користувача, є зараз популярною темою для дослідження. Очевидно, що автоматизація безцінна для пошуку ресурсів, потрібно ще багато досліджень, щоб пошук став більш ефективним. Прості роботи користувача ще дуже далекі від розумних мережних агентів.

## Погані реалізації

Навантаження на мережу та сервери іноді збільшується за рахунок погано написаних нових роботів. Навіть якщо вхідні та вихідні дані правильні, є багато менш очевидних проблем.

Наприклад, іноді один робот звертається декілька раз до одного й того ж ресурсу. Іноді робота просто тестують на цьому ресурсі, а у інших випадках є помилки у програмі робота. Повторні звертання трапляються, коли не записуються історія завантажених ресурсів (що обов‘язково потрібно робити), або коли робот не розпізнає випадки коли використовуються різні DNS-назви для однієї IP адреси.

Деякі роботи отримують файли таких типів, які вони не можуть обробити (GIF, PS), і ігнорують, хоча їх можна було б і не завантажувати.

# Ведення каталогу

Бази даних ресурсів, що знайдені роботом, безумовно, популярні. Однак, є декілька проблем, що обмежують використання роботів для пошуку ресурсів.

## Забагато матеріалу, що є забагато динамічним.

Виміром ефективності заданого підходу отримання інформації є відсоток релевантних документів серед усіх знайдених. Пошук інформації відбувається не у самому Інтернеті а у локальній базі даних конкретного робота, що може не містити інформації, яка насправді існує у Інтернеті, тому що розмір Інтернет велетенський і зміни дуже часті.

## Визначення що індексувати, а що ні.

Робот не може автоматично визначити, чи потрібно дану веб-сторінку включати до свого списку чи ні. Веб-сервери можуть містити документи, які використовуються тільки локально, наприклад, список внутрішньої бібліотеки, або ті, які існують тимчасово. До деякої міри, рішення про те, що необхідно, а що ні, залежить від уподобань користувачів, але вони можуть бути невідомими для робота. Практично роботи зберігають майже все, що вони знаходять. Помітьте, що навіть, якщо робот зможе визначити, що дану сторінку не треба включати у базу даних, він уже отримав цей непотрібний файл з Інтернет. Навпаки, робот, що ігнорує великий відсоток документів є не дуже корисним.

Robot Community спробувало змінити цю ситуацію, створивши спеціальний стандарт "A standard for robot exclusion". Цей стандарт описує використання простого структурованого текстового файлу, щоб задати, які частини даного серверу не потрібно оглядати роботам. Цю можливість можна також використовувати, аби повідомити робота про чорні дірки, коли заглиблюючись у все глибші сторінки робот не може з них повернутися. Кожному роботу можна давати особливі інструкції, бо кожен з них спеціалізується в певній окремій галузі. Цей стандарт не загальноприйнятий, але вважається, що кожен робот повинен його дотримуватися.

Визначити порядок обходу сайтів це велика проблема. Більшість сайтів організовані ієрархічно, тому обхід в ширину, з вершини до заданої глибини, дає більший набір документів, ніж обхід в глибину. Проте обхід в глибину швидше віднайде сторінки користувачів, в яких є посилання на інші сторінки, і тому швидше знайде нові сайти для обходу.

|  |
| --- |
| # /robots.txt for http://www.site.com/  User-agent: \* # attention all robots:  Disallow: /cyberworld/map # infinite URL space  Disallow: /tmp/ # temporary files |

Приклад файлу robots.txt

## Статистичний аналіз

Дуже складно проіндексувати звичайний веб-документ. Перші роботи просто зберігали назви документів та тексти посилань, але сучасні роботи використовують більш складні механізми, і як правило аналізують весь зміст документу.

Ці методи можна автоматично застосовувати для всіх документів, але вони не можуть бути настільки ефективні, як ручне індексування автором. HTML дає можливість додати метаінформацію до документів, яка спрощує роботу пошук по документу.

# Етика

Зрозуміло, що роботи дуже корисні, але вони накладають високі вимоги на трафік і спричинюють багато проблем. Тому автори роботів мусять обирати золоту середину між користю та шкодою, коли створюють та випускають робота. І тут є етична проблема: "Чи шкоду від роботів можна виправдати їх корисністю". Люди мають різні думки щодо цього.

Деякі з проблем стали очевидними лише тоді, коли роботи збільшили вдвічі завантаження серверів. Мартин Костер виробив набір правил для авторів роботів, дотримуючись яких, можна мінімізувати шкоду від роботів:

* подумайте, чи насправді вам потрібен новий робот;
* зробіть так, щоб адміністратори серверів могли легко ідентифікувати робота та при необхідності зв‘язатися з автором;
* ретельно протестуйте робота локально;
* керуйте використанням ресурсів, не допускайте кількох послідовних скачувань з одного серверу та не допускайте непотрібних скачувань;
* дотримуйтесь стандарту "for Robot Exclusion";
* регулярно переглядайте log файли робота;
* діліться результатами своєї роботи з іншими.

Девід Ейхмант розділяє роботів, які створюють загальнодоступні інформаційні бази, та роботів для користувачів, результат роботи яких, використовується лише однією людиною.

Той факт, що більшість авторів роботів використовують поради Мартина Костера показує, що вони свідомо ставляться до можливих проблем, та намагаються мінімізувати будь-який негативний вплив.

# Альтернативи для пошуку ресурсів

Існує альтернативний підхід до пошуку ресурсів, коли сумарна індексна інформація про сервер вже зібрана на ньому. Це інформація лише про локальні ресурси. Вона може бути створена вручну, а може автоматично з заголовків, або тегів META. Ця інформація додається до пошукової бази даних за допомогою звичайних WWW протоколів. Це не робот, тому що він не отримує рекурсивно документи які є в цьому індексі.

У цього способу є переваги. Якість індексу, створеного людьми, поєднується з ефективністю автоматичного оновлення. Цілісність інформації у цього способу вище, ніж у звичайних індексів, тому, що підтримувати потрібно тільки локальну індексну інформацію. Дуже низькі вимоги до мережі, індексна інформація менша ніж весь сайт і отримується тільки один раз.

Є також декілька недоліків. Ручна підтримка індексної інформації може дати додаткові проблеми провайдеру інформації, але практично, індексна інформація для основних документів змінюється не часто. Іншим обмеженням є те, що інформаційні провайдери мусять записувати інформацію у заданому індексному форматі, що обмежує використання додаткових можливостей. На кінець, оновлення індексу є не дуже ефективними, бо увесь індексний файл треба прочитати заново, навіть якщо змінився лише один запис.

Така система, як описано вище, - ALIWEB вже працює з жовтня 1993 і дає не погані результати. Але подальший розвиток йде повільно, тому що це персональний проект, над яким працюють у вільний час, і який не отримує капіталовкладень.

Harvest — це інша система пошуку інформації, що була недавно випущена IRTF-RD, що пропонує програмні системи для автоматичного індексування змісту документів, ефективної реплікації та кешування такої індексної інформації на віддалених хостах, і на кінець пошук цієї інформації через інтерфейс у WWW. Реакції на цю систему були дуже позитивними.

# Висновок

Роботи є дуже корисними та перспективними програмами для Інтернет, але при їх написанні потрібно враховувати, що вони можуть суттєво зменшити трафік для інших користувачів. Щоб цього не сталося, при написанні роботів потрібно користуватися методологією Мартіна Костера та підтримувати "Standard for Robot Exclusion".

# Література

1. Martijn Koster, "A Standard for Robot Exclusion," Nexor Corp., http://web.nexor.co.uk/mak/doc/robots/norobots.html.
2. Martijn Koster, "Robots in the Web: threat or treat?", NEXOR
3. David Eichmann, "Ethical Web Agents"