Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Cezary Kaliszyk
Nr albumu: 189400

Site Improving Engine

Praca licencjacka
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
Krzysztofa Ciebiery
Instytut Informatyki

Wrzesień 2003

Pracę przedkładam do oceny

Data Podpis autora pracy:

Praca jest gotowa do oceny przez recenzenta

Data Podpis kierującego pracą:

Streszczenie

Niniejszy dokument jest opisem systemu *Site Improving Engine*, stworzonego przez grupe Cek (w składzie Cezary Kaliszyk, Grzegorz Andruszkiewicz, Katarzyna Bozek, Ewa Łyszczek) jako praca licencjacka. System nasz jest podstawą projektów przystosowujących strony WWW do potrzeb indywidualnego użytkownika (ang. *adaptive web*). Programista, pisząc własny moduł, dostaje od naszego systemu obsługę wszelkich niezbędnych protokołów sieciowych, obsługę pracy w klastrze, identyfikację użytkowników, interpretację dokumentów *html*, bazę danych i obsługę błędów. Ssystem jest napisany w funkcyjnym języku oCaml, co dodatkowo ułatwia pisanie nowych modułów, przy zachowaniu jednocześnie wysokiej wydajności.

Słowa kluczowe

adaptive web, oCaml, WWW, html

Klasyfikacja tematyczna

H - Information Systems
H.3 - Information Storage and Retrieval

Spis treści

Wstęp

Współczesny świat charakteryzuje się niespotykanym dotąd w historii rozwojem technologii i komunikacji, zachodzącymi dynamicznie procesami globalizacji.

Z tym łączy się ogrom informacji, z którym współczesny człowiek, a w szczególności użytkownik sieci Internet, styka się na codzień.

Niewątpliwym pozytywem wymienionych procesów jest możliwość wyszukania danych niedostępnych (trudno dostępnych) w inny sposób, ale negatywną stroną jest zwiększający się koszt i nakłady czasowe potrzebne do znalezienia nawet najbardziej podstawowej informacji.

Internetowe serwisy informacyjne prześcigają się w stworzeniu klarownej struktury udostępnianych stron, umożliwiającej hierarchiczny i logicznie uporządkowany dostęp dodanych.

Nie można jednak dogodzić wszystkim na raz - sztywna struktura powiązań pomiędzy stronami jest stworzona pod „przeciętnego użytkownika", nietypowe preferencje nie mogą zostać uwzględnione.

Zaproponowany przez nas system jest kolejnym krokiem naprzód. Analizuje sposób korzystania przez użytkownika ze stron internetowych, zapamiętuje kolejno wykonywane czynności i tworzy system linków dopasowany do jego indywidualnych preferencji. *Site Improving Engine* optymalizuje i personalizuje dynamicznie strukturę przekazywanych danych. Dodatkowo, oferujemy użytkownikom wyszukiwarkę (dobierającą strony także na zasadzie preferencji indywidualnych) oraz możliwość nagrania i odtworzenia przebiegu sesji. Sesje mogą być odtwarzane wielokrotnie, równolegle, mogą być parametryzowane.[1]

[1] np. Za każdym odtworzeniem w formularzu zostanie wpisane inne nazwisko - wtedy nazwisko nazywamy *parametrem odtworzenia sesji*

Rozdział 1

Założenia projektu

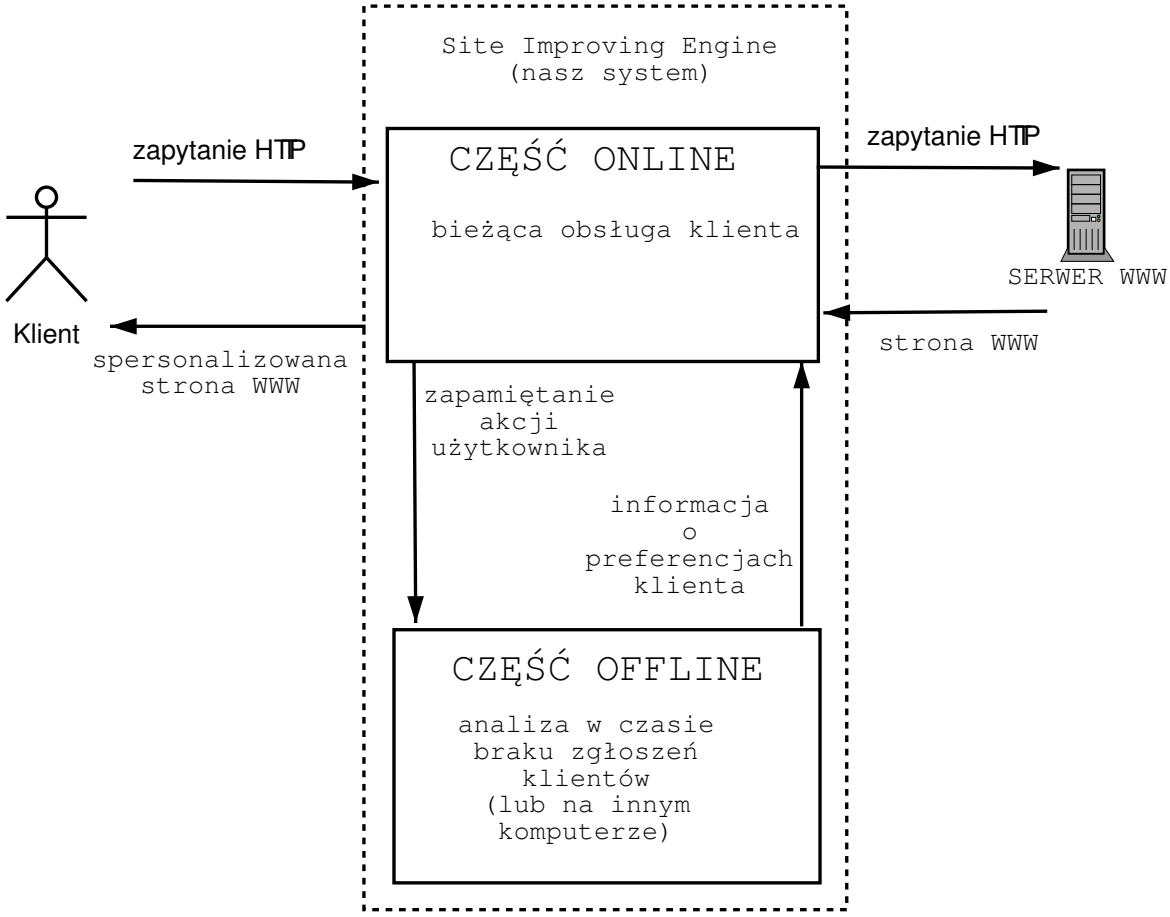
- analiza działań użytkownika - wybieranych przez niego stron z serwisu WWW, informacji zawartych na tych stronach, czasu oglądania każdej strony, kolejności klikanych linków, etc.
- dynamiczne dodawanie spersonalizowanej struktury linków - przy przesyłaniu każdej strony nasz system stara się „domyślić" jakie strony użytkownik będzie chciał zobaczyć w następnej kolejności i dodaje do nich linki.
- udostępnienie wyszukiwarki - wyszukiwarka pokazuje strony zawierające słowa kluczowe podane przez użytkownika. Stara się przeanalizować, które strony będą ciekawsze dla konkretnego klienta.
- przezroczystość dla użytkownika - użytkownik nie musi instalować specjalnego oprogramowania, używa zwykłej przeglądarki i łączy się z serwerem WWW. Nasz system przechwytuje komunikację w obie strony pomiędzy serwerem a klientem i wzbogaca strony WWW. Gdyby nie nowe linki, użytkownik w ogóle by nie wiedział o pośrednictwie naszego systemu.
- skalowalność - poprzez dokupienie nowych komputerów można prawie dowolnie zwiększać wydajność naszego systemu.
- wysoka wydajność pracy - nasz system, gdy jest zainstalowany na wystarczającej liczbie komputerów, w sposób niezauważalny opóźnia otrzymanie strony WWW przez użytkownika.
- niezawodność - w nasz system są wbudowane mechanizmy pozwalające na wyłączenie całości lub części systemu, która źle działa (*Watchdog* - monitoruje i ew. wyłącza system automatycznie).
- modularność i łatwość modyfikacji - podzieliliśmy system na dobrze wyspecyfikowane części z ustalonym interfejsem. Można bez problemu modyfikować oraz dopisywać nowe moduły poprawiające strony WWW.

Rozdział 2

Zasada działania

Architektura systemu SIE (Site Improving Engine) skonstruowana jest z myślą o jak najlepszym spełnieniu kryteriów z poprzedniego rozdziału.

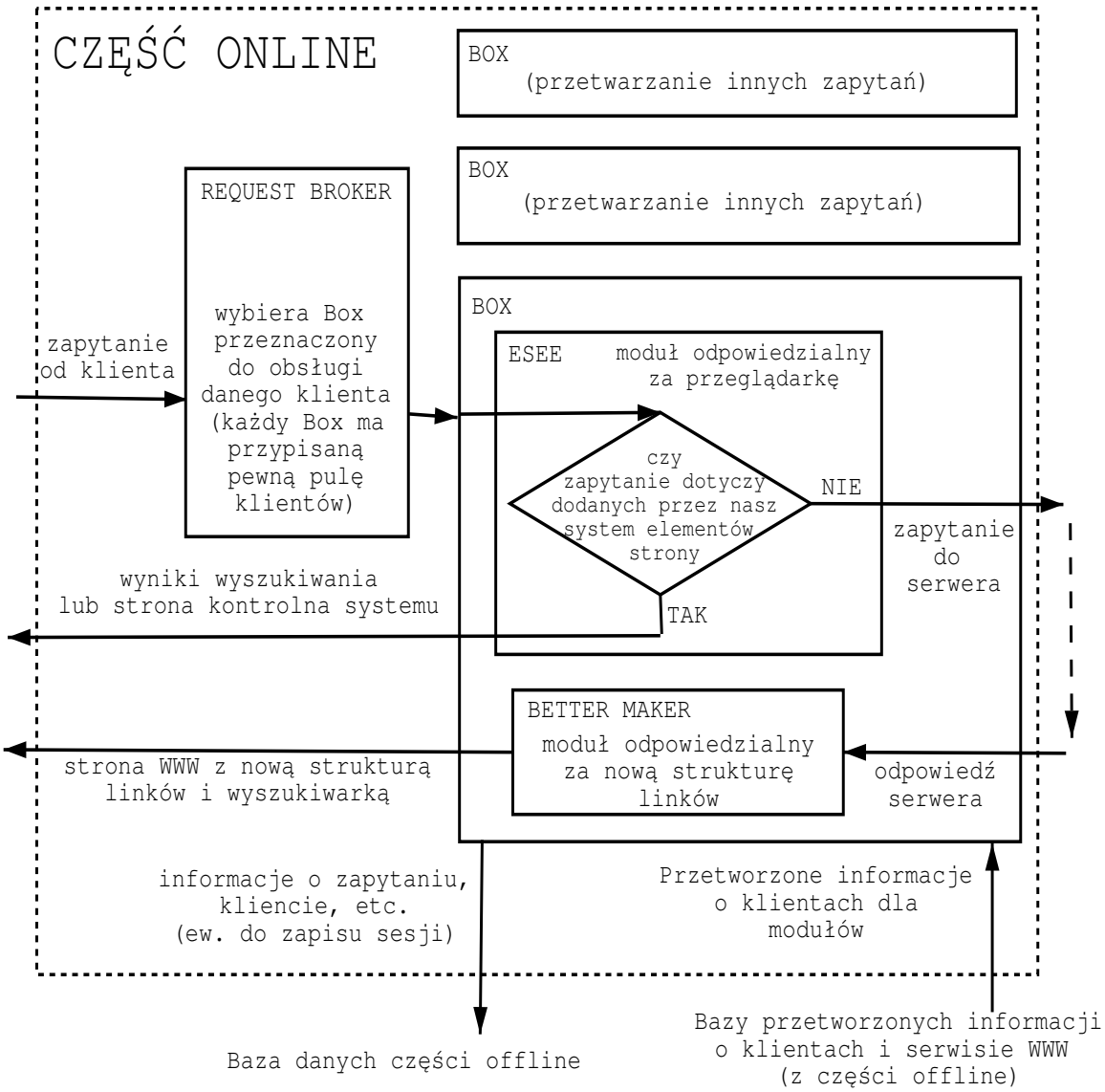
Najlepiej widać to na schemacie:



System nasz dzieli się na dwie funkcjonalne części:

- *część Online* - na którą składają się wszystkie czynności wykonywane w trakcie obsługi klienta i niezbędne do tego celu
- *część Offline* - w której wykonywane są obliczenia i analizy możliwe do wykonania a priori. Wyniki tych działań są następnie wykorzystywane później w części *Online*.

2.1. Część Online



Kiedy klient, chce pobrać stronę, jego przeglądarka wysyła *zapytanie http* do serwera WWW. Nasz system, w sposób niewidoczny dla użytkownika, przechwytuje to zapytanie zanim dotrze do serwera.

Zapytanie to trafi najpierw do *Request Broker'a*, który przekazuje je dalej do *Box'a*. *Box'ów* może być kilka i są to równolegle działające aplikacje, każda na innym komputerze. Zastosowaliśmy tutaj tzw. *architekturę klastrową*, czyli podzieliliśmy pracę na kilka równolegle działających maszyn, wykonujących dokładnie tę samą pracę. Zwiększa to wydajność i jednocześnie bezpieczeństwo. Zapewnia także skalowalność rozwiązania (można praktycznie dowolnie rozbudować system nie napotykając na problem maksymalnej wydajności pojedynczej maszyny).

Box najpierw zapisuje informacje o zapytaniu do *Bazy Danych*, co jest potem niezbędne do przeanalizowania działań użytkownika. Następnie uruchamia pierwszy moduł przetwarzający informację - *ESEE*.

ESEE sprawdza, czy zapytanie rzeczywiście skierowane jest do serwera, czy może odnosi się do części strony obsługiwanej wyłącznie przez nasz system (np. przeszukiwarka lub strona

kontrolna). W tym pierwszym przypadku przekazujemy zapytanie dalej do serwera, modyfikując je nieznacznie (podmieniamy linki i pola adresowe), a w drugim przesyłamy odpowiedź bezpośrednio klientowi.

Box umożliwia także użytkownikowi wybranie opcji zapisania sesji. Jest to bardzo przydatna funkcja, umożliwia szybkie odtworzenie skomplikowanej sekwencji kliknięć i wypełniania formularzy. Zapisaną sesję (ściągniętą przez przeglądarkę) można wykorzystywać w aplikacji Edytor.

Dalej zapytanie trafia do serwera, który myśli, że to zapytanie pochodzi od naszego serwera i wysyła mu odpowiedź, nie zdając sobie sprawy, że jesteśmy jedynie pośrednikiem. (uzyskujemy taką sytuacje dzięki podmianie linków)

Powracające zapytanie znowu trafia do tego samego *Box'a*, który tym razem przekazuje je do drugiego modułu przetwarzającego - *Better Maker*. Wcześniej jednak modyfikuje wszystkie linki znajdujące się na tej stronie tak, aby wskazywały na nasz serwer. Gdy klient kliknie na tak zmieniony link, jego przeglądarka połączy się z naszym serwerem, zamiast z serwerem docelowym. Mechanizm ten nazywamy *Podmianą linków*. Dzięki niemu kontrolujemy komunikację pomiędzy klientem a serwerem WWW i możemy zapisywać dodatkowe szczegółowe informacje, jak np. z jakiej strony pochodzi dane kliknięcie, czas czytania strony etc., które zapisujemy w bazie danych.

Better Maker „przystraja" stronę stworzoną przez nas strukturą połączeń i dodaje linki do *Przeszukiwarki* i *Panelu kontrolnego*. Informację niezbędną do dostosowania struktury połączeń do wymagań indywidualnych czerpie z analiz przygotowanych uprzednio w części *Offline* przez *eXPperimenter'a*. Zmodyfikowana odpowiedź trafia do klienta.

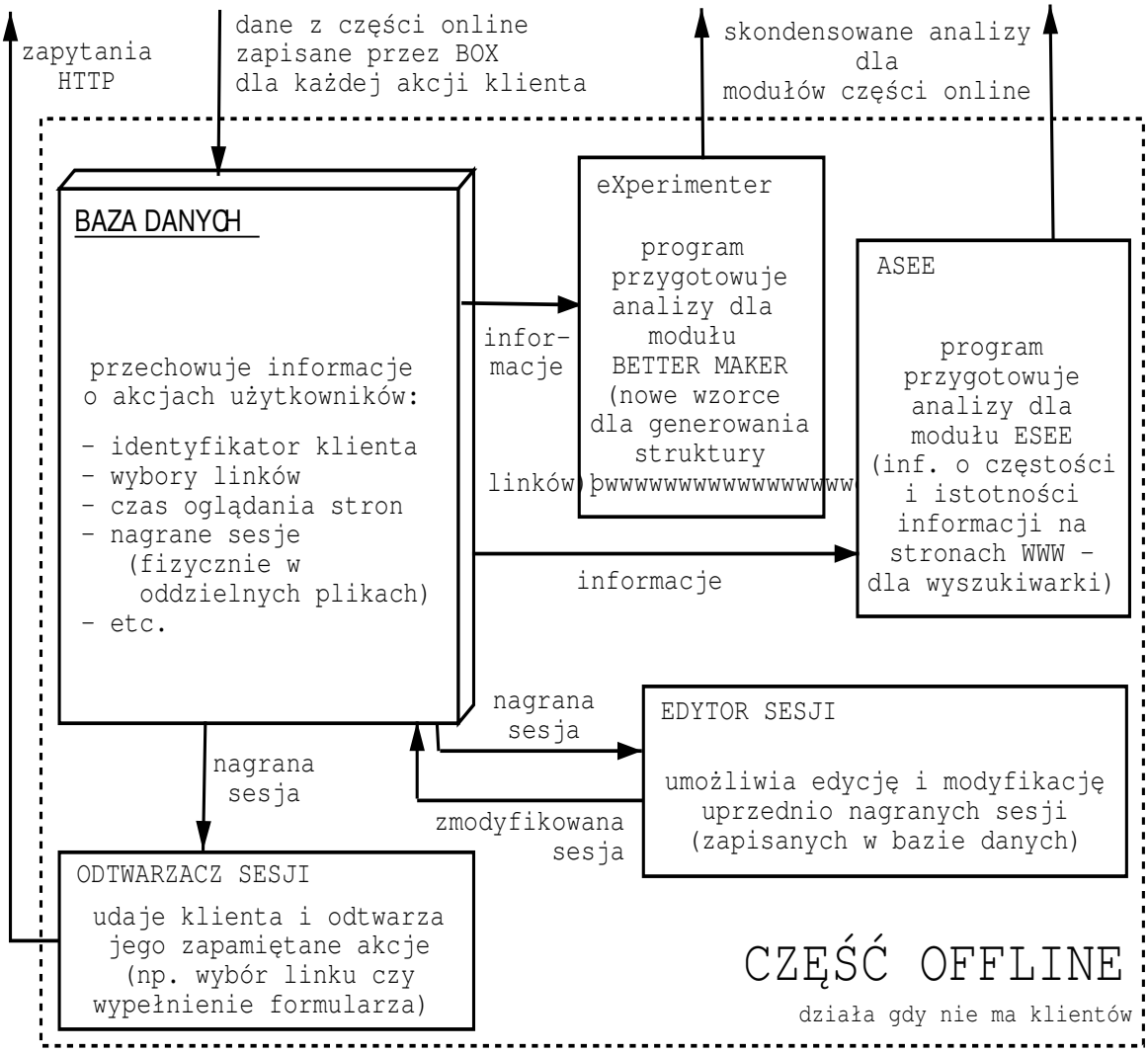
Klientów identyfikujemy dzięki tzw. mechanizmowi ciasteczek[1]. Każdemu nowemu klientowi przydzielamy losowo wygenerowany numer. Przeglądarka klienta pamięta od tej pory ciasteczko[2] i przy komunikacji z naszym serwerem przekazuje nam jego kopię, jednoznacznie identyfikując klienta.

Istotną częścią podsystemu *Online* jest *Watchdog* - system monitorowania i automatycznego odłączania nieprawidłowo działających części (*Box'ów*) lub wyłączenia systemu jako całości i przepuszczania zapytań bezpośrednio do serwera WWW.

[1] ang. cookie

[2] Niektóre starsze przeglądarki nie obsługują mechanizmu ciasteczek - w nowszych można ten mechanizm wyłączyć - wtedy nasz system za pomocą mechanizmu *Podmiany linków* potrafi rozpoznawać danego klienta przez czas trwania danej sesji, ale w następnej sesji klientowi zostanie przydzielony inny numer

2.2. Część Offline



Wszystkie operacje zawierające się w tej części systemu wykonywane zazwyczaj na innym komputerze, aby nie obciążać części *Online* zajętej bezpośrednią obsługą zapytań klientów.

Przeprowadzana jest tutaj analiza zebranych danych. Generuje ona wzorce mające później służyć części *Online*:

- *eXPerimenter* - program dokonujący analizy przejść użytkowników. Wylicza informacje niezbędne do późniejszego stworzenia zindywidualizowanej struktury stron internetowych, zaspokajającej potrzeby każdego użytkownika[3]. Analizator ten współpracuje z modułem części *online Better Maker*.

- *ASEE* - Program analizuje zbiory stron na serwerze i ich ocenę przez użytkowników, aby uzyskać informacje umożliwiające sprawne działanie wyszukiwarki. Moduł ten współpracuje z *ESEE* - modułem części *Online*.[4]

- *Edytor* - moduł ten umożliwia edycję nagranych sesji pracy na serwerze[5] - dzięki temu

[3]Szczegóły można znaleźć w dodatkowej dokumentacji tego modułu
[4]Szczegóły można znaleźć w dodatkowej dokumentacji tego modułu
[5]Sesja jest sekwencją przejść przez strony które wybierał użytkownik

mechanizmowi można wyróżnić istotne dla pewnego działania przejścia.[6]

- *Odtwarzacz sesji* - umożliwia odtwarzanie sesji nagranych przez system. Dzięki temu można testować działanie systemu w sytuacjach gdy wielu użytkowników będzie próbowało wykonać pewne zestawy czynności (takie jak założenie konta w systemie). Odtwarzacz współpracuje z edytorem sesji. Główną zaletą mechanizmu nagrywania i odtwarzania sesji jest możliwość automatycznego wielokrotnego i równoległego (też z różnych komputerów jednocześnie) [7] symulowanie działań użytkowników. Może to być wykorzystane np. do testowania wydajności serwera WWW. Poza tym bardzo duże możliwości daje parametryzowanie odtworzenia sesji (np. dla dowolnego użytkownika zapisujemy sesję będącą założeniem konta na nowym serwerze, a następnie odtwarzamy tę sesję wielokrotnie podstawiając za każdym razem nazwę kolejnego użytkownika.) Może to bardzo przyspieszyć procesy wielokrotnego wykonywania żmudnych czynności, niewiele się od siebie różniących.

[6] Szczegóły można znaleźć w dodatkowej dokumentacji tego modułu

[7] Dzięki mechanizmowi tzw. *tuneli SSH* możemy „oszukiwać" serwer, który myśli, że zapytania przychodzą z wielu komputerów, a tak na prawdę pochodzą tylko z jednego

Rozdział 3

Rzeczywiste zastosowanie systemu

System został uruchomiony w połowie marca na systemie olimpiady informatycznej (sio.mimuw.edu.pl:8080). Od początku kwietnia oferował już swoją wyszukiwarkę i podpowiadał linki, miał jednak tyle niedociągnieć, że nie nadawał się do sensownego wykorzystania. Od połowy kwietnia posiada całą funkcjonalność opisaną w tym dokumencie. Od tego czasu zarejestrowaliśmy 113 sesji, w których brało udział 86-ciu użytkowników. W sumie klienci zadali 473 zapytania HTTP (nie licząc robotów[1]). Z tego 63 do naszej wyszukiwarki.

3.1. Podobne systemy

Istotną kwestią jest istnienie podobnych systemów do naszego, mających taką samą funkcjonalność. Istnieje wiele systemów zawierających pewne elementy naszego programu, ale żaden z nich nie spełnia podstawowej jego funkcjonalności - daje API modułom użytkownika, zapewniając im obsługę protokołów sieciowych, analizę i parsowanie drzewa HTML, identyfikację poszczególnych użytkowników oraz bazę danych ich działań w serwisie WWW.

[1] Robot to inny program działający w sieci, który zbiera w sposób zautomatyzowany informacje o stronach WWW Żeby rozpoznać roboty, umieszczamy na każdej stronie jednopunktowy obrazek będący specjalnym linkiem. Człowiek nie jest w stanie dostrzec tego obrazka i nigdy nie wybierze tego linku - robot natomiast przechodzi najczęściej wszystkie linki na stronie - a więc też ten ukryty.

Rozdział 4

Zalety zaproponowanej architektury systemu

W rozdziale tym opiszę, w jaki sposób nasza architektura systemu i przyjęte rozwiązania przyczyniają się do zrealizowania wymienionych na początku tego dokument założeń projektu.

analiza działań użytkownika - nasz system zapisuje w części *Online* wszystkie działania użytkownika i analizuje je potem w części *Offline*

dynamiczne dodawanie spersonalizowanej struktury linków - moduł *Better Maker* dodaje do każdej strony strukturę linków, skonstruowaną w oparciu o analizę działań użytkowników

udostępnienie wyszukiwarki - wyszukiwarka jest dostępna z poziomu panelu kontrolnego na każdej stronie (moduł *ESEE*) - jej działanie jest także oparte na analizie działań użytkowników

przezroczystość dla użytkownika - użytkownik wie o istnieniu naszego systemu tylko dzięki efektom jego pracy (nowe linki, etc.), nie musi natomiast go „instalować" lub specjalnie aktywować.

skalowalność - dzięki zastosowaniu architektury klastrowej (wiele współbieżnie[1] działających *Box'ów*) można rozbudowywać i zwiększać wydajność naszego systemu praktycznie dowolnie.

wysoka wydajność pracy - uzyskana została głównie dzięki architekturze klastrowej.

niezawodność - jest to też cecha architektury klastrowej - uszkodzenie jednego komputera wchodzącego w skład klastra, nie przeszkadza w dalszym funkcjonowaniu systemu - zadania tego komputera są przejmowane przez pozostałe sprawne jednostki. *Watchdog* automatycznie zarządza uszkodzonymi elementami systemu, w razie konieczności zupełnie odłącza system umożliwiając pracę użytkownikom serwisu WWW.

modularność i łatwość modyfikacji - nasz system jest podzielony na dobrze zdefiniowane części - poszczególne moduły przetwarzania *Online* i analizatory pracujące *Offline*. Każda z tych części ma dużą niezależność od pozostałych i może być modyfikowana oddzielnie lub wykorzystana w innym projekcie.

[1] jednocześnie

Rozdział 5

Podsumowanie

Przedstawiliśmy działający system, dopasowujący strukturę informacji do potrzeb konkretnego użytkownika, podpowiadający mu często wykorzystywane przez niego strony i informacje, których z dużym prawdopodobieństwem w tym momencie potrzebuje. Dodaliśmy wyszukiwarkę, oceniającą strony też na podstawie ich przewidywanej użyteczności dla konkretnego klienta, oraz możliwość zapisywania całych sekwencji posunięć, co dodatkowo przyspieszy długie, a często powtarzane sekwencje czynności dostępu do danych (np. sprawdzanie stanu konta w banku internetowym - można przyspieszyć akcje wpisywanie identyfikatora klienta i hasła, a następnie wybierania np. opcji wyświetlenia ostatnich operacji dokonanych na rachunku).

Uważam, że projekt *SIE* jest udaną próbą wykorzystania technik komputerowych i statystycznych do zmniejszenia kosztów i nakładów czasowych, niezbędnych do znalezienia pożądanej informacji. Mam nadzieję, że nasz system przyczyni się do powstania profesjonalnych programów ułatwiających użytkownikowi poruszanie się po głębokich wodach ogromu informacji znajdującego się w zbiorach Internetu.

Bibliografia

[http] *Http 1.1 Specification, RFC 2616*

[html] *HTML 4.01 Specification, http://www.w3.org/TR/html4*

[cookies] *Persistent client state HTTP cookies, http://wp.netscape.com/newsref/std/cookie_spec.html*