Web Crawler

Índice

- Introducción
- Taxonomía
- Arquitectura e implementación
- Algoritmos de planificación
- Conclusiones

Introducción

Introducción

- Web crawler / Web Spider / Web Robot / Bot
  - Software automático para la descarga de web
  - Envía peticiones a los servidores web
- El primer crawler fue creado en 1993 por Matthew Gray, un estudiante del MIT
  - Recuperaba las URL de una web
  - Mantenía un registro de las visitadas
  - Producía un índice que permitía búsquedas
- Los primeros buscadores ya comenzaron a utilizar crawlers
  - Lycos (1994)
  - Excite (1995)
  - AltaVista (1995)
  - HotBot (1996)

Introducción

- En los 90s, los buscadores competían con los servicios de directorios (AOL, Yahoo, etc.)
  - La web era muy pequeña y los servicios de directorios eran intuitivos
- Actualmente, todos los buscadores hacen uso de crawlers
  - Gran parte del éxito de un buscador recae en el crawler que utilice
  - Aspecto crucial en el desarrollo de un sistema RI
- Un 10% de las peticiones recibidas por los servidores son realizadas por crawlers

Introducción

- Algoritmo básico
  - Entrada: conjunto de páginas (semillas)
  - Las semillas son descargadas, analizadas. Se extraen los links
  - Los links pertenecientes a páginas no visitadas se almacenan en una cola central para su posterior procesamiento
  - El crawler selecciona una nueva página de la cola de descargas
  - El proceso se repite hasta que se da alguna condición de parada

Introducción

```
Crawler(semillas S)

1. ColaURLs ← S
2. hacer {
3.   p ← Seleccionar-URL(ColaURL)
4.   contenido ← descargar(p)
5.   (texto, enlaces, estructura) ← parser(contenido)
6.   ColaURLs ← AñadirNuevosEnlaces(ColaURLs, enlaces)
7.   Procesar texto, estructura etc.
8. } hasta (criterio_parada)
```

Introducción

- Aplicaciones de los crawlers:
  - Búsqueda web
    - Crawler vertical: agregación de datos desde fuentes diferentes de materiales relacionados
      - Shop-bot: crawler para la descarga de catálogos de tiendas de ropa y ofrecer una plataforma para comparar precios
      - Recopilador de e-mails
    - Crawler vertical de formatos específicos
      - Descarga sólo objetos con un formato específico: audio, imágenes, videos, pdf, etc.
      - Ejemplo: CiteseerX

Introducción

- Crawler temático
  - Descarga contenido relacionado con un tema en particular
  - Se centra en páginas con un contenido específico
  - Recibe como entrada la descripción del tema (consulta o documentos de ejemplo)
  - La salida será una lista de páginas relacionadas con el tema
  - Puede funcionar en modo por lotes o bajo demanda
- Caracterización de la web: obtener estadísticas válidas de la web
- Mirroring: almacenamiento total o parcial de un sitio web, actualizando el contenido

Introducción

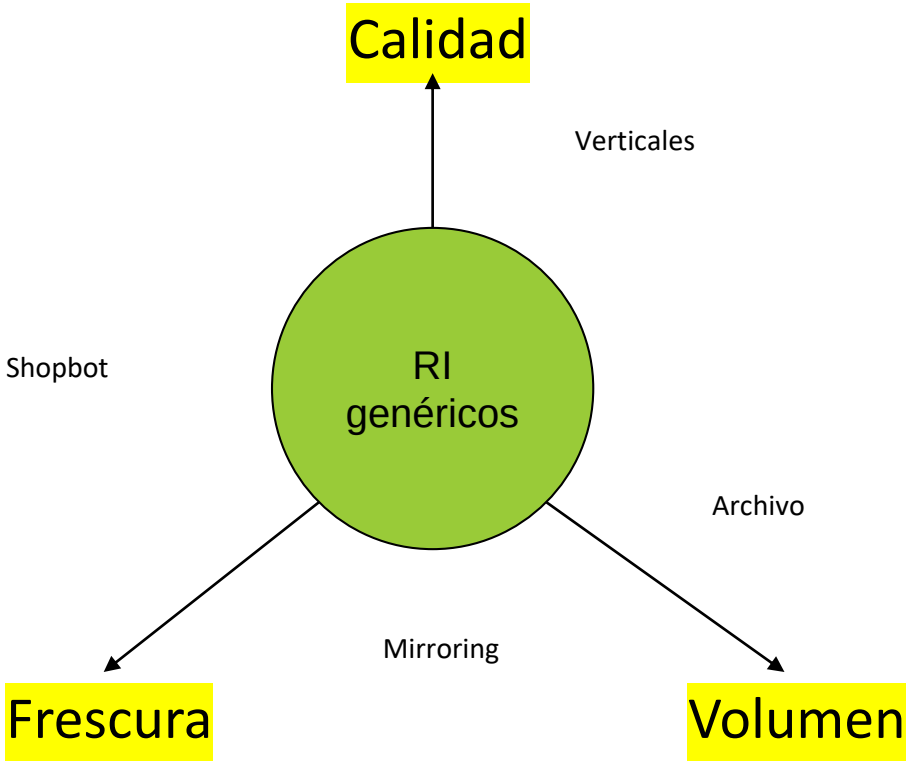
- Archivo: almacenamiento de páginas web sin descartar copias antiguas
- Análisis de sitios web
  - Dada una web o conjunto de páginas comprueba si se ha producido algún cambio de acuerdo a un criterio predefinido
  - Ejemplos
    - Validación de enlaces: comprobación periódica de enlaces “rotos”
    - Análisis de directorios para comprobar sitios que ya no están disponibles
    - Búsqueda de vulnerabilidades
    - Detección de imágenes sin la propiedad “alt”
    - Etc.

Taxonomía

Taxonomía

- Características de los crawlers:
  - Frescura
    - Es importante que la copia obtenida sea lo más reciente posible
    - La antigüedad de la copia no es problema
  - Calidad
    - Porción de la web, pero con una calidad alta
    - Amplia cobertura incluyendo diferentes niveles de calidad
  - Volumen
    - Almacenar una gran fracción de la web
    - Almacenar una porción más pequeña

Taxonomía


Calidad
Verticales
Shopbot
RI
genéricos
Archivo
Mirroring
Frescura
Volumen

Taxonomía

- Desde el punto de vista de un crawler, las web se pueden clasificar en:
  - Privadas: no son indexables → no pueden descargarse mediante un crawler
    - Es necesario una contraseña para acceder
    - Se ocultan detrás de un cortafuegos (Intranet)
    - Datos de redes sociales
      - Sólo disponibles para los amigos
      - Sólo disponibles para amigos de amigos
      - Sólo disponibles para las personas que cumplan los criterios de privacidad establecidos por el usuario
  - Públicas: públicas y accesibles por todo el mundo
    - Se pueden indexar
    - Un crawler puede acceder y descargarlas

Taxonomía

- Estáticas
  - El servidor web las almacena y las sirve bajo una petición
  - Contenido estático
  - HTML/XHTML
  - Imágenes, videos, documentos, etc.
- Dinámicas
  - Se crean bajo demanda
    - Cuando el servidor recibe una petición, interpreta el código y genera la web resultado
  - Contenido dinámico
  - PHP, JSP, Perl, etc.
  - Se pueden generar miles de diferentes web dinámicas dependiendo de los parámetros de la petición
  - Algunas sólo son accesibles tras rellenar un formulario
  - Otras pueden accederse fácilmente siguiendo un enlace

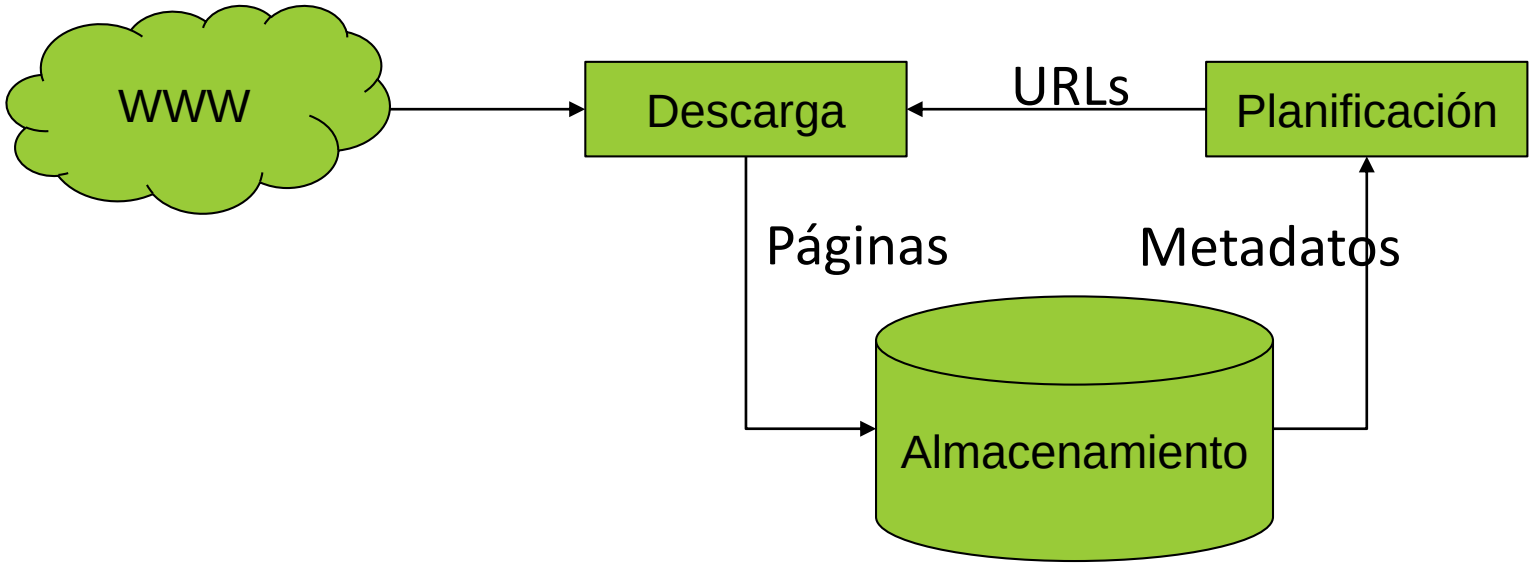
Taxonomía

	Estáticas	Dinámicas	
		Alcanzables por enlaces	Alcanzables por formulario
Privadas	No indexable		
Públicas	Indexable		Web oculta

Arquitectura e Implementación

Arquitectura e Implementación

- Un crawler se compone de tres módulos principales
  - Descarga
  - Almacenamiento
  - Planificación



Arquitectura e Implementación

- El módulo de descarga es el más importante
  - Mantiene la cola de URLs por visitar (frontera) y las envía al módulo de descarga (pueden ser varios módulos)
- El módulo de descarga se encargará de recuperar el contenido de una determinada URL y enviarlo al módulo de almacenamiento
- El módulo de almacenamiento guarda una copia del contenido para su posterior indexación
  - Provee de metadatos al planificador útiles en la configuración de la política a llevar a cabo

Arquitectura e Implementación

- Aspectos prácticos a tener en cuenta en la implementación de un crawler
  - Resolución DNS
  - Canonización de URLs
  - Análisis (parsing)
  - Errores 404 y páginas 404
  - Duplicados
  - Sistema paralelo o distribuido

Arquitectura e Implementación

RESOLUCIÓN DNS

Arquitectura e Implementación Resolución DNS

- Fallos DNS
- Registros DNS erróneos o mal formados
- Eficiencia en la resolución
- Los crawler pueden saturar DNS locales
  - DNS caching: almacenar las direcciones IP de los servidores a los que se acceden más frecuentemente
    - Más eficiente que esperar que el servidor DNS resuelva la petición

Arquitectura e Implementación

CANONIZACIÓN DE URLS

Arquitectura e Implementación Canonización de URLs

- La web contiene numeras URLs que apuntan hacia el mismo contenido:
  - http://x.ejemplo.com/
  - http://x.ejemplo.com/index.html
  - http://ejemplo.com/x/
  - http://ejemplo.com/x/index.html
  - http://ejemplo.com/x?sessionid=000001
- Es preferible detectar URLs similares antes que detectar contenido duplicado
  - Mediante reglas sintácticas podemos detectar URLs similares
    - Eliminación de index.html
    - Eliminación de parámetros de sesión
    - Reglas que cumplen los servidores más populares

Arquitectura e Implementación

ANÁLISIS

Arquitectura e Implementación Análisis

- Muchas páginas web están mal codificadas
  - Etiquetas erróneas
  - Etiquetas abiertas que no son cerradas
  - No se adhieren a la codificación HTML
- El usuario no percibe dichos errores al navegar ya que los navegadores son muy tolerantes a esos fallos
  - No quieren interrumpir la experiencia de usuario por un error de sintaxis en la página
- El análisis estricto de una página no es posible
  - El crawler tiene que ser tolerante a dichos errores
    - Construir el DOM solucionando inconsistencias

Arquitectura e Implementación Análisis

- En el análisis o parsing tiene por objetivo extraer información:
  - Etiquetas HTML: títulos, cabeceras (H1, H2,), etc.
  - Extracción de entidades mediante técnicas de análisis del lenguaje natural
    - Nombres de personas u organizaciones
    - Fechas
    - Localizaciones geográficas
    - Teléfonos
    - Parejas atributos-valor: producto-precio
    - Opiniones

Arquitectura e Implementación

PÁGINAS 404

Arquitectura e Implementación Páginas 404

- Cuando se intenta acceder a una URL determinada, el servidor puede actuar de varias formas:
  - Error 404: Page not found
  - Redireccionar a una página con un mensaje de error indicando qué la página no existe
  - Redireccionar a una página valida (index)
- Distinguir entre una URL válida y no válida es una tarea compleja en el caso de que el servidor redireccione en lugar de enviar un error 404
- Solución
  - Realizar una petición de una URL aleatoria al servidor para comprobar su comportamiento
  - Algoritmos de clasificación de texto

Arquitectura e Implementación

DUPLICADOS

Arquitectura e Implementación Duplicados

- Un gran porcentaje de las páginas web tienen contenido duplicado
  - Intencionado
    - Mirroring
    - Plagio
  - No intencionado: URLs apuntando al mismo documento
- En los foros o sitios de respuestas, el hilo principal muestra todos los mensajes
  - Para cada mensaje existe una página dedicada
  - La conversación global es lo que tiene valor
- Se debe primar el contenido original

Arquitectura e Implementación

PARALELISMO

Arquitectura e Implementación Paralelismo

- Para lograr escalabilidad y tolerancia a fallos, se deben utilizar múltiples hilos
  - El ancho de banda del crawler suele ser mayor que el ancho de banda del servidor
    - El crawler no puede detenerse a la espera de que una petición individual termine
- En colecciones extremadamente grandes, se debe diseñar un sistema distribuido
  - Evitar descargar la misma página varias veces
  - La coordinación e intercambio de URLs debe hacerse con máxima precaución
  - Cada página debe descargarse en un proceso único
  - El proceso encargado de descubrir los enlaces dentro de una web, no puede ser el mismo encargado de la descarga del contenido

Arquitectura e Implementación Paralelismo

- La decisión de que proceso debe realizar la descarga se realiza mediante una función de asignación
  - Conocida desde el principio
- Las URLs pertenecientes al mismo dominio las debe descargar un mismo proceso
- Características de las funciones de asignación
  - Propiedad de balanceo: cada proceso crawler tiene que tener el mismo número de host
  - Propiedad contra-varianza: si el número de procesos crece, el número de host asociados a cada uno debe disminuir
  - Dinamismo: capacidad de añadir y eliminar procesos de forma dinámica
- Las URLs se tienen que intercambiar por lotes
  - Intercambiar una única URL consume muchos recursos

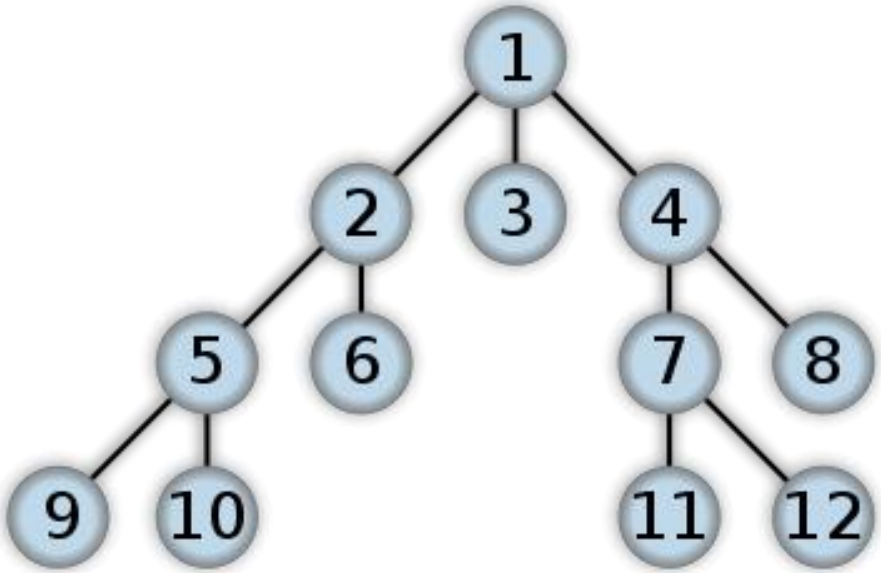
Algoritmos de planificación

Algoritmos de planificación

- Un crawler tiene que equilibrar muchos objetivos de forma simultánea
  - Objetivos contradictorios
- Algunos objetivos:
  - Necesita frescura en el contenido, por lo que tiene que revisitar frecuentemente las páginas
  - Descubrir nuevas páginas
  - Utilizar el ancho de banda de forma eficiente
  - No malgastar los recursos
- Problema: la web es dinámica
  - Las páginas se crean, destruyen y actualizan al instante

Algoritmos de planificación

- Se suele seguir una estrategia de búsqueda en anchura
  - Se parte de u nodo inicial
  - Se visitan los vecinos
  - Se visitan los hijos y sus vecinos



Algoritmos de planificación

- Un crawler se basa en tres políticas con objetivos diferentes:
  1. Política de Selección
  2. Política de revisitas
  3. Política de cortesía

Algoritmos de planificación

POLÍTICA DE SELECCIÓN

Algoritmos de planificación Política de selección

- La web es enorme → un crawler no puede acceder a todas las páginas
  - Se tienen que centrar en un conjunto limitado
    - Páginas principales
    - Páginas importantes
    - Páginas de actualidad
  - La decisión de la siguiente página a descargar no puede tomarse al azar
- El ancho de banda es limitado y no es gratuito
  - El crawler tiene que ser escalable y eficiente
  - El coste de una petición puede considerarse como marginal, pero la web es enorme por lo que el coste es directamente proporcional

Algoritmos de planificación Política de selección

- Límite off-line: son preestablecidos a priori
  - Un número máximo de host
  - Una profundidad máxima
    - Número máximo de enlaces a seguir que empiecen en una URL determinada
  - Número máximo global de páginas
    - El espacio de almacenamiento es finito
  - Límites por host o por dominio
    - Número máximo de páginas
    - Bytes
  - Lista de mime-types aceptados
    - text/html
    - Text/plain
  - Límite por página
    - Si es una página grande, indexar solo las primeras palabras

Algoritmos de planificación Política de selección

- Selección on-line
  - Dependiente de una métrica de importancia
    - Calidad
    - Popularidad en número de visitas o enlaces
    - Tipo de URL
  - Técnica OPIC (on-line Page Importance Computation)
    - Cada página tiene una “suma de dinero”
    - Dicho dinero se reparte de forma equitativa entre los enlaces a los que apunta
    - Similar a PageRank pero se computa en una sola iteración
    - Se descargan primero las páginas con más “efectivo”

Algoritmos de planificación

POLÍTICA DE REVISITA

Algoritmos de planificación Política de revisita

- La web es dinámica
- Realizar un crawling de una porción de la web puede llevar meses
  - Cuando la descarga ha finalizado, la captura está anticuada
- Tipo de eventos posibles
  - Creación
  - Actualización
    - Menor: a nivel de párrafo o frase. La página en general no ha cambiado
    - Mayor: el contenido antiguo ya no es valido
  - Borrado: la página ya no está accesible

Algoritmos de planificación Política de revisita

- Frescura: indica si la copia local está actualizada o no

$$F_p(t) = \begin{cases} 1 & Si\ p\ es\ igual\ a\ la\ copia\ local\ en\ el\ instante\ t \\ 0 & en\ otro\ caso \end{cases}$$

- Edad: indica como de antigua es la copia local

$$A_p(t) = \begin{cases} 1 & Si\ p\ no\ ha\ sido\ modificada\ desde\ la\ ultima\ actualización \\ t - t_{ultima\_actua} & en\ otro\ caso \end{cases}$$

Algoritmos de planificación Política de revisita

- Los objetivos del crawler pueden ser:
  - Mantener una media de frescura en las páginas en la colección local
    - ¿Cuántas páginas están desfasadas?
  - Mantener la edad media de las páginas en la colección local
    - ¿Cómo de antiguas son las copias locales?

Algoritmos de planificación

POLÍTICA DE CORTESÍA

Algoritmos de planificación Política de cortesía

- Los crawlers realizan multitud de peticiones a los servidores en intervalos de tiempo cortos
  - Pueden causar una caída del servidor
- Aunque las páginas de un servidor sean alcanzables sin contraseña, o cortafuegos, no tienen por qué ser públicas
  - Los administradores tienen que poder decidir qué páginas serán públicas y cuáles no
- Las copias almacenadas localmente pueden infringir ciertos permisos de autor (copyright)
- Un crawler mal diseñado y que no siga unas guías de estilo puede ser censurado desde los proveedores de servicios

Algoritmos de planificación Política de cortesía

- Reglas básicas de un crawler
  1. Se tiene que identificar asimismo como un crawler y no hacerse pasar por un usuario normal
     - Utilizar el campo user-agent del protocolo HTTP
       - El nombre debe tener la palabra robot o crawler
     - Proveer de e-mail y web donde se detalle el próposito
  2. Obedecer a los protocolos de exclusión que indican qué parte no debe ser accedido por el crawler
     - Archivo robots.txt

```
User-agent: *
Disallow: /datos
Disallow: /cgi-bin
```

  3. Mantener un uso de ancho de banda pequeño para un sitio web determinado
     - Mantener un intervalo de tiempo entre peticiones (10 segundos)