Tarântula – Sistema de Recolha de Documentos da Web

Daniel Gomes, Mário J. Silva

Faculdade de Ciências da Universidade de Lisboa, Departamento de Informática
1749-016 Lisboa
{dcg@di.fc.ul.pt, mjs@di.fc.ul.pt}

Palavras chave: Recolha de informação, Spider, Crawler, Robot, TUMBA, World Wide Web.

Resumo
A recolha de colecções de documentos é uma tarefa cada vez mais comum nos sistemas que actuam sobre a WWW. Este artigo apresenta um novo sistema de recolha de documentos da Web, com características de configuração, integração e distribuição que permitem a sua fácil adaptação e integração em aplicações de diferentes naturezas. Descrevemos a sua arquitectura, funcionamento e os resultados obtidos ao efectuarmos uma recolha da Web Portuguesa.

I. Introdução

Nos últimos anos a Internet vulgarizou-se como forma privilegiada de comunicação e ferramenta de trabalho ou lazer, fazendo com que cada vez existam mais documentos online. Em paralelo com a vulgarização da Internet, a comunidade científica e a indústria Informática empenham-se em criar novos sistemas que permitam gerir e tirar proveito de toda esta informação disponível [5, 6]. Neste contexto, têm sido desenvolvidos, num processo que tenta acompanhar o crescimento da *web*, os sistemas de recolha, vulgarmente conhecidos como *robots*, *spiders* ou *crawlers*. Estes sistemas têm sido incluídos como subsistema em aplicações de diversas naturezas, mas o investimento no desenvolvimento de *crawlers* cada vez mais sofisticados, tem sido feito principalmente pelo mercado dos motores de busca mundiais, onde são componentes cruciais. No entanto, o aumento dos investimentos não se tem reflectido, de forma proporcional, na evolução dos sistemas de *crawling*.

Devido às características competitivas deste mercado, existem poucos artigos publicados sobre o assunto, sendo normalmente bastante vagos e não constituindo uma base de conhecimento suficiente para o desenvolvimento de novos *crawlers*. Este interesse motivou-nos para a construção do sistema Tarântula [9]. Este possui a capacidade de ser integrado em várias aplicações *Web* que realizem um conjunto de funções complexas que são comuns a esta variedade de aplicações.

O desenvolvimento do Tarântula teve como objectivo a criação de um módulo de recolha de documentos da *Web* genérico, com características de configuração e escalabilidade que permitam a sua fácil integração como subsistema de uma aplicação, independentemente das suas características particulares. O Tarântula guarda informação relativa às suas acções e aos documentos recolhidos, permitindo a sua visualização e tratamento consoante a finalidade a que se destinam na aplicação em que se integram. As acções do Tarântula podem ser monitorizadas e controladas, sendo possível interrompê-las e retomá-las de forma simples e eficiente.

O Tarântula foi integrado com sucesso como módulo de recolha em dois projectos com objectivos e requisitos distintos, o TUMBA [30] e o DROP [18]. O TUMBA é um projecto de investigação que consiste no desenvolvimento de um motor de busca de documentos na *Web* Portuguesa. O DROP é um sistema de recolha e armazenamento de publicações online, para depósito digital na Biblioteca Nacional, à semelhança do depósito legal que é efectuado para as publicações tradicionais impressas em papel.

Este artigo está organizado da seguinte forma: na secção II, introduzimos o tema dos sistemas de recolha de informação da Web e algumas das suas aplicações; na secção III, apresentamos a arquitectura, o funcionamento e a concretização do Tarântula; na secção IV, os resultados da integração do sistema no projecto TUMBA. Finalmente, na secção V apresentamos as conclusões e indicamos trabalho futuro.

II. Sistemas de Recolha de Informação da Web

Um sistema de recolha de informação da *Web*, vulgarmente designado por *crawler*, é um componente de *software* conceptualmente simples, existente praticamente desde o início da Internet [17, 20, 29]. A sua complexidade pode variar muito, consoante os requisitos do sistema onde estará integrado, podendo ser uma simples aplicação com carácter académico [28] ou um elaborado projecto de investigação como o Mercator [11], o Harvest [10] ou o *crawler* do Google [3]. Se pretendermos construir um motor de busca à escala mundial, o *crawler* deverá ter capacidade para recolher uma grande quantidade de documentos num intervalo de tempo relativamente curto, sendo a sua

escalabilidade um dos seus requisitos mais importantes, face ao constante crescimento da WWW. No entanto, ao integrarmos um *crawler* num sistema como o DROP, em que se pretende fazer um depósito legal de publicações online, o seu principal requisito é o de conseguir fazer a cópia mais fiel possível da publicação original, lidando com a diversidade de documentos e estrutura das publicações online.

A. Processo de recolha

A actividade de um *crawler* consiste num processo iterativo de recolha de documentos da *Web* [5]. Este processo inicia-se a partir de um conjunto de URLs que referenciam páginas *Web*. Estes URLs, que são vulgarmente denominados de *raízes*, são inseridos no sistema através de uma entidade externa. A partir das *raízes*, é feita a recolha do conteúdos dos documentos por elas referidos. Por último, é feita uma análise dos conteúdos, a fim de extrair as ligações para novos documentos, que serão recolhidos numa próxima iteração.

A complexidade de um *crawler* surge no tratamento da diversidade de protocolos, tipos de ficheiros ou sistemas de segurança que encontramos na Internet. As maiores dificuldades são devidas ao crescente desrespeito pelas normas estabelecidas que obrigam o sistema a ser tolerante a este tipo de situações e ao mesmo tempo robusto a informações erróneas. Por vezes, estas informações são induzidas voluntariamente na Internet por indivíduos mal-intencionados com o objectivo de sabotar o funcionamento destes sistemas.

Embora à partida se pretenda que um *crawler* recolha a maior quantidade de documentos possível, esta não pode ser feita de forma indiscriminada [34], devendo respeitar normas de bom comportamento [15], tais como:

- Identificar o *crawler*, usando os campos disponibilizados pelo protocolo HTTP [32] para esse fim;
- Não sobrecarregar servidores *Web*, evitando pedidos simultâneos ou sequenciais a um mesmo servidor;
- Não visitar servidores ou partes de servidores que não pretendam ser visitados por *crawlers*, respeitando o protocolo REP (*Robot Exclusion Protocol*) [14].

O *crawler* ao mesmo tempo que se mostra cordial com os servidores *Web* deve ser robusto a situações nefastas para o seu desempenho, como por exemplo:

- Evitar recolher documentos repetidos.
- Não recolher informação fútil para os propósitos do sistema.
- Evitar *spider traps*. As *spider traps* são URLs que fazem com que o *crawler* faça recolhas infinitas num dado sítio da *Web*.

Um *crawler* deverá manter, de forma eficiente e acessível, informação acerca das suas acções de recolha de documentos da *Web*. Esta informação destina-se a permitir a monitorização permanente das suas acções e uma análise estatística da WWW, ou parte dela, à posteriori. A monitorização das acções do *crawler* é crucial para a sua operação, uma vez que dada a vastidão e diversidade da WWW, é impossível prever ou testar todas as situações que possam surgir. Nas recolhas efectuadas com o Tarântula detectámos situações inesperadas que nos permitiram melhorar o sistema, tais como: servidores *Web* que devolvem respostas distintas consoante a identificação do cliente, documentos repetidos devido a redirecções, perda de documentos devido a má configuração ou URLs sintacticamente incorrectos aceites pelos servidores *Web* [33].

III. Tarântula

Nesta secção apresentamos o Tarântula, começando por descrever a sua arquitectura na subsecção A. Na subsecção B, descrevemos como é feita a distribuição do trabalho pelos componentes (*Coleccionadores*) que efectuam a recolha de documentos da *Web* e como controlamos a carga que incute, às máquinas que os alojam. Na subsecção C, descrevemos a estrutura de execução de um *Coleccionador*. Na subsecção D, exemplificamos o funcionamento do sistema ao recolher um documento da *Web*. Finalmente, na subsecção E, apresentamos as tecnologias utilizadas e principais dificuldades encontradas na concretização do sistema.

No Tarântula, o utilizador fornece ao sistema dados de parametrização das recolhas, que especificam o comportamento pretendido. Os parâmetros das configurações determinam a *raiz* (URL a partir do qual se inicia a recolha), os servidores que poderão ser visitados, os tipos aceites, o tamanho máximo e a profundidade máxima dos documentos. No contexto do nosso trabalho, convencionámos que a profundidade seria o número mínimo de saltos (*links*) entre dois documentos. Assim sendo, a profundidade máxima determina a distância máxima entre a *raiz* e os restantes documentos da recolha. Cada configuração contém ainda informação relativa a quais as datas, prazos e periodicidade com que deverão ser efectuadas as recolhas.

A. Arquitectura

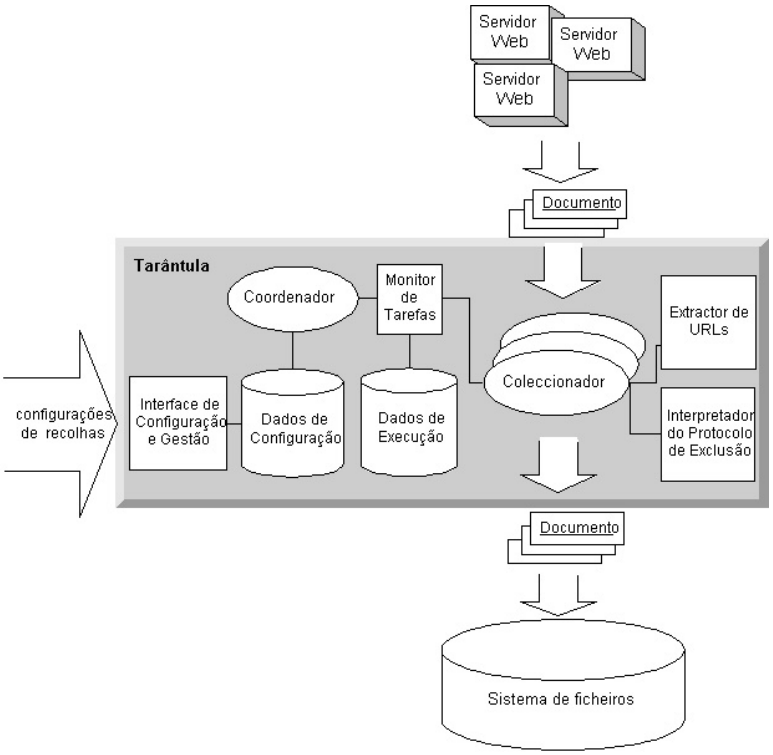

Figura 1 Arquitectura do Tarântula. O sistema recebe configurações de recolha, a partir das quais rege o processo de recolha dos documentos online, transferindo-os da WWW para o sistema de ficheiros.

A Figura 1 representa a arquitectura de *software* do Tarântula. Trata-se de um conjunto de componentes de *software* que se encontram interligados através de interfaces bem definidas. O processo de recolha dos documentos desenvolve-se através da interacção entre os componentes do Tarântula:

- A *Interface de Configuração e Gestão* recebe configurações de recolha fornecidas pelos utilizadores. Sendo o Tarântula um sistema de recolha genérico, destinado a ser usado em aplicações de diferentes naturezas, este componente deverá ser reconstruído consoante as necessidades do sistema que o vier a incorporar.
- Os *Dados de Configuração* armazenam as configurações de recolha fornecidas pelos utilizadores. Os *Dados de Execução* armazenam os dados relativos às tarefas a executar nos processos de recolha dos documentos. O *Monitor de Tarefas* tem a função de controlar o acesso dos restantes componentes a estes *Dados de Execução*. O *Coordenador* tem a responsabilidade de identificar recolhas a efectuar através da

análise periódica dos *Dados de Configuração* e agendá-las no *Monitor de Tarefas*.

- O *Coleccionador* recolhe documentos da *Web* através do protocolo HTTP e armazena-os no sistema de ficheiros, guardando informações relativas à recolha. No Tarântula podem existir vários *Coleccionadores* em execução simultânea. Cada *Coleccionador* recorre a dois componentes para efectuar as recolhas, o *Interpretador do Protocolo de Exclusão* e o *Extractor de URLs*. O *Interpretador do Protocolo de Exclusão* é um componente, que interpreta as restrições à recolha automática de documentos por *robots*, impostas pelos administradores dos sítios da *Web* através do REP. O *Extractor de URLs*, analisa os conteúdos dos documentos HTML recolhidos e extrai as ligações (URLs) para outros documentos, tolerando os erros de sintaxe do HTML mais comuns.

B. Controlo de carga e distribuição

A arquitectura modular do Tarântula permite fazer escalar o seu desempenho, através da distribuição dos seus componentes por várias máquinas, podendo cada uma delas ter características diferentes, desde que permitam a execução de programas numa base de *software* comum estabelecida pela arquitectura.

O controlo da carga que um *Coleccionador* pode incutir à máquina em que está alojado, é feito através da parametrização do número máximo de servidores que o *Coleccionador* pode recolher simultaneamente. Esta funcionalidade permite optimizar o funcionamento do sistema consoante as características do *hardware* disponível, permitindo a utilização de estações de trabalho comuns para efectuar recolhas sem prejudicar a interactividade com o seu utilizador.

A capacidade de recolha do Tarântula pode ser aumentada através da expansão do sistema, acrescentando novas máquinas que alojem *Coleccionadores*. A divisão de trabalho pelos *Coleccionadores* é feita através de uma função de partição por nós definida. Consideramos que cada um servidor *Web* que aloja documentos a recolher corresponde a uma unidade de trabalho. A partição do trabalho pelos *Coleccionadores* é feita da seguinte forma:

> Numa configuração com *N Coleccionadores*, cada um deles recebe um identificador $I=\{0, 1, ..., N-1\}$. Um dado *Coleccionador* $i \in I$ recolherá os documentos do servidor com URL *U*, se $i=comprimento\_string(U) mod(N)$.

Escolhemos este mecanismo, por ser simples e facilmente concretizável, permitindo distribuir os documentos a recolher por diversos *Coleccionadores* sem gerar conflitos. Ao mesmo tempo, atribui automaticamente a recolha de novos servidores descobertos durante a recolha e garante que cada servidor *Web* não é visitado simultaneamente por mais do que um *Coleccionador*.

C. Níveis de execução de um Coleccionador

O *Coleccionador* tem a função de recolher os documentos da *Web*, armazená-los no sistema de ficheiros e guardar os dados gerados durante o processo de recolha. No entanto, é imprescindível que possua mecanismos que permitam controlar a carga de pedidos a cada servidor e dar robustez ao sistema perante situações de excepção. A recolha de documentos deve progredir concorrentemente, mesmo quando um documento referenciado fica inacessível ou tem um tempo de acesso muito elevado. Para satisfazermos estes requisitos concretizámos cada *Coleccionador* segundo uma hierarquia de fios de execução mostrada na Figura 2.

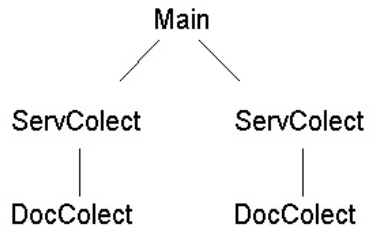

Figura 2. Hierarquia de fios de execução do *Coleccionador*. O *Main* lança um *ServColect* que será responsável pela recolha dos documentos de cada servidor. O *ServColect* lança um *DocColect* para recolha de cada um documento.

O fio de execução *Main* é o processo principal do *Coleccionador* e encontra-se em permanente execução, lançando um fio de execução do tipo *ServColect*, por cada servidor *Web* a visitar pelo *Coleccionador*. Cada *ServColect*, por sua vez, tem a responsabilidade de recolher todos os documentos alojados no servidor *Web*, lançando um fio de execução *DocColect* por cada documento a recolher.

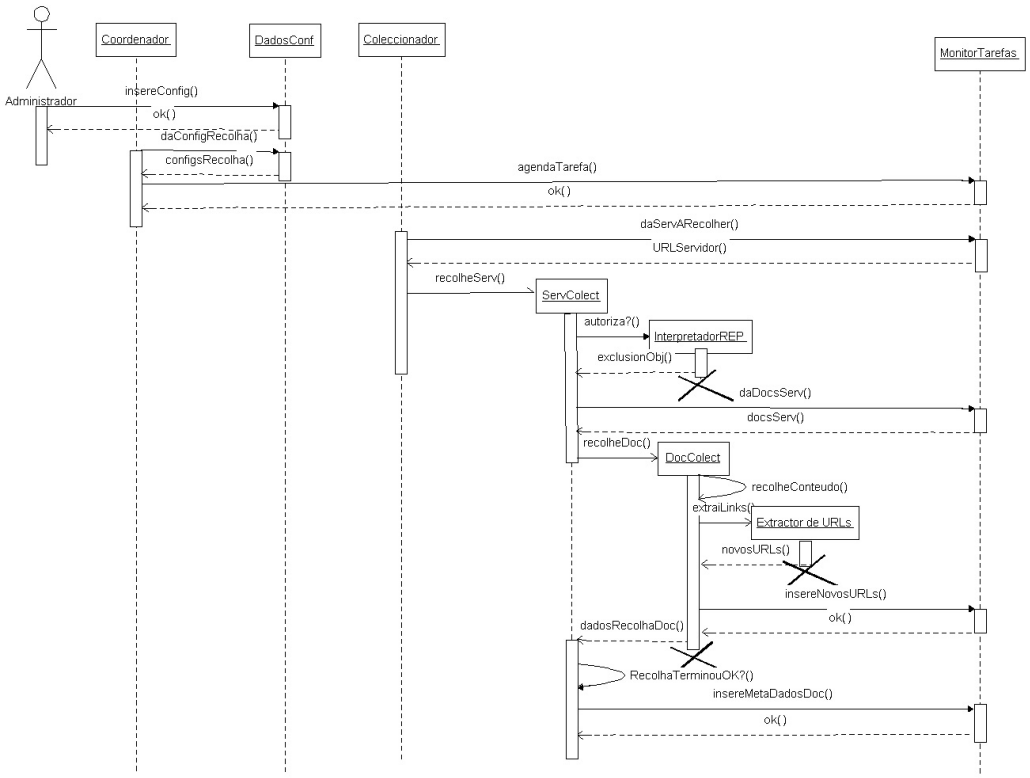

Figura 3. Diagrama de Sequência UML representativo da estrutura de execução do Tarântula ao efectuar a recolha de um documento.

D. Fluxo de execução

O funcionamento do Tarântula é exemplificado na Figura 3. O processo de recolha é desencadeado após a inserção pelo utilizador de uma configuração de recolha nos *Dados de Configuração*. O *Coordenador* periodicamente analisa as todas as configurações de recolha obtidas através da invocação do método *daConfigRecolha()* e quando detecta que deve ser feita uma nova recolha, agenda uma tarefa no *Monitor de Tarefas* invocando o método *agendaTarefa().*

O *Coleccionador* consulta periodicamente o *Monitor de Tarefas*, a fim de obter servidores *Web* que contenham documentos a recolher (*daServARecolher()*). Uma vez obtidos, invoca o método *recolheServ()*, que lança assincronamente fios de execução *ServColect* destinados a efectuar a recolha dos documentos de cada servidor.

Um fio de execução do *Coleccionador*, *ServColect* inicia a sua execução invocando o *Interpretador do REP*, para obter o objecto de controlo de acesso ao servidor. Em seguida, o *ServColect* pede ao *Monitor de Tarefas*, por invocação do método *daDocsServ()*, uma lista de todos os documentos que deve

recolher no servidor. *ServColect* lança sincronamente um novo fio de execução, *DocColect* que efectuará a recolha de um documento, e espera que este termine no prazo máximo estipulado pelos dados de configuração da recolha. Caso isto não aconteça, *DocColect* é terminado pelo *thread* pai (*ServColect*).

O *ServColect* após executar a recolha de um documento, dá preempção a outro fio de execução, só efectuando um novo pedido ao servidor passado um intervalo de tempo configurável (1 segundo por omissão). Os pedidos são assim distribuídos pelos servidores *Web* de modo a não os sobrecarregar com pedidos sucessivos.

Um fio de execução *DocColect* recolhe o documento da *Web* respeitando as restrições impostas pelas configurações e guarda o seu conteúdo em memória. Sobre este o *Extractor de URLs* efectua a extracção de ligações e insere-as para recolha no *Monitor de Tarefas* através do método *insereNovosURLs()*. Em seguida, o *DocColect* armazena o documento no sistema de ficheiros e termina a sua execução devolvendo ao *ServColect* os dados relativos à recolha do documento (*dadosRecolhaDoc()*).

O *ServColect* verifica se o documento foi recolhido dentro do limite de tempo estabelecido e insere os seus dados de recolha nos *Dados de Execução* através da invocação do método *insereMetaDadosDoc()*. O *ServColect* lança ciclicamente fios de execução *DocColect***,** respeitando o intervalo mínimo entre pedidos ao mesmo servidor, até não existirem mais documentos a recolher no servidor.

E. Concretização do sistema

1) Tecnologias

Os componentes do Tarântula foram realizados com recurso às seguintes tecnologias:

Os *Dados de Configuração* e *Dados de Execução* são armazenados num sistema de gestão de base de dados PostgreSQL [21] e acedidos por SQL. O *Monitor de Tarefas* foi concretizado através de uma classe Java [24] que estabelece uma ligação JDBC [23] com os *Dados de Execução*. *A Interface de Configuração e Gestão* foi desenvolvida recorrendo a HTML e a Java Servlets [22], executadas no contentor de *Servlets* TOMCAT [27] e disponibilizadas através de um servidor *web* Apache [26]. O *Interpretador do Protocolo de Exclusão* é um analisador lexical, gerado a partir de uma versão modificada do JLex [1]. O *Extractor de URLs* é um analisador lexical, gerado a partir do JFLex [2]. O *Coordenador e Coleccionador* são aplicações *multi-threaded* desenvolvidas em Java. Todo o código escrito na linguagem Java foi desenvolvido recorrendo ao JDK1.3 [25].

2) Principais dificuldades

Em face da permanente evolução da *Web* e a escassa documentação existente acerca da construção de *crawlers*, foi difícil estabelecer à partida quais os maiores obstáculos a ultrapassar. Embora a arquitectura do Tarântula estivesse definida desde o inicio do projecto, a construção do seu *software* foi progredindo incrementalmente, evoluindo à medida que superávamos as dificuldades encontradas. Apresentamos em seguida algumas delas:

- Estabelecimento de quais os parâmetros de configuração a definir para as recolhas, de forma a que fossem específicos e compreensíveis, permitindo que sistemas com necessidades de recolha diferentes pudessem facilmente utilizar ao Tarântula.
- Adaptação de configurações dos *Coleccionadores* a máquinas com características de *hardware* diferentes. As máquinas com poucos recursos que executam um *Coleccionador*, podem lançar um número excessivo de fios de execução e não conseguir cumprir os prazos de recolha configurados, perdendo documentos. Esta situação embora de solução simples, é difícil de identificar, uma vez que apesar da máquina não operar na sua carga máxima, por vezes o tempo máximo para recolha de um documento é atingido, enquanto o fio de execução responsável por efectuar a recolha do documento se encontra à espera de obter o CPU.
- Construção do *Extractor de URLs*. Teoricamente, o funcionamento do *Extractor de URLs* deveria ser determinado pela sintaxe do HTML [31]. No entanto, o número de documentos que os servidores *web* indicam como sendo do tipo *text/html* que não respeitam a definição (DTD) do formato HTML é muito elevado. Há por isso que adaptar o extractor a um número muito elevado de situações de erro, de modo a conseguir extrair o máximo de ligações contidas nestes documentos.
- Minimização da congestão no acesso aos *Dados de Execução*. O acesso rápido a esta base de dados é crucial para o funcionamento do sistema e foi durante muito tempo uma grande limitação ao seu desempenho. Foram necessárias várias alterações ao código desenvolvido e às configurações do *PostgreSQL* para a superar.

IV. Resultados

Esta secção descreve o desempenho do Tarântula, e os resultados da recolha de documentos alojados sob o domínio .PT, efectuada na construção do motor de busca TUMBA.

A. Ambiente de produção

A recolha foi efectuada a partir de 12781 *raízes* (uma por servidor) obtidas a partir de um *mirror* do servidor de nomes da FCCN. O Tarântula foi configurado para recolher todos os documentos do tipo MIME [13] *text/html*, com tamanho inferior a 200 KB que se encontrassem alojados num servidor do domínio .PT e a uma profundidade máxima de 3. O tempo máximo permitido para a recolha de cada documento foi de 10 minutos, ao fim do qual a recolha era cancelada.

O ambiente de produção utilizado na recolha foi constituído por 7 máquinas com características distintas, desde um Pentium a 200MHz com 128 MB de memória a um Pentium 3 a 1 GHz com 256 MB de memória, ligadas entre si através de uma rede local a 100 Mbit/s. Esta rede encontrava-se ligada à FCCN através de uma ligação a 4 Mbit/s, no momento da realização desta recolha (Agosto de 2001).

B. Desempenho

A recolha teve a duração de 4 dias e 5 horas. Foram efectuados 796629 pedidos HTTP, dos quais resultou a recolha de 676261 documentos, alojados em 14068 servidores *Web*. Os documentos HTML recolhidos foram armazenados em 14 GB de disco (não comprimidos), tendo um tamanho médio de 23,9 KB.

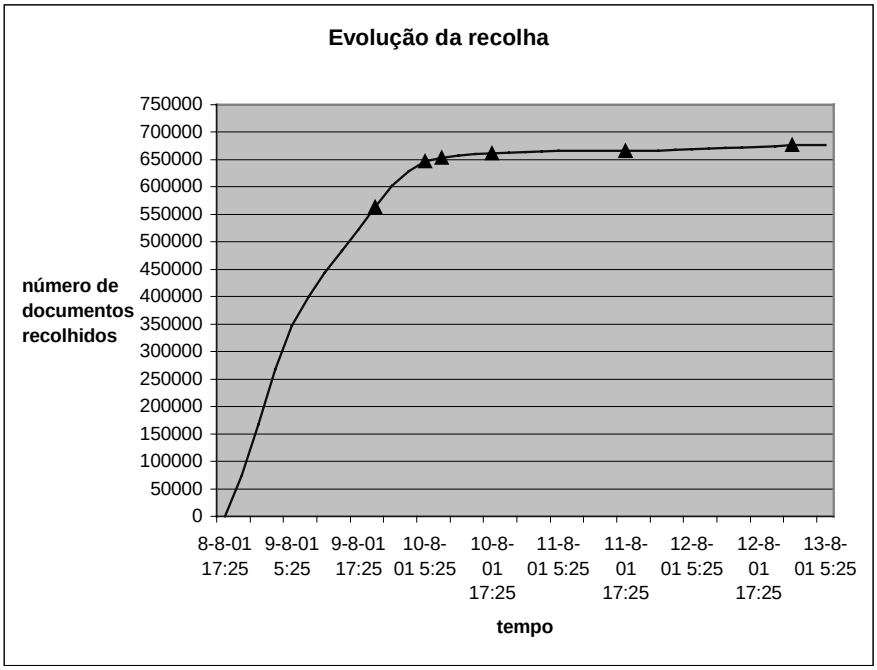

Figura 4. Evolução do número de documentos recolhidos ao longo do tempo. Cada triângulo na curva do gráfico indica a data de término da recolha por parte de um Coleccionador.

A Figura 4 mostra a evolução da recolha ao longo do tempo. O sistema teve o seu pico de desempenho, medido em termos do débito de documentos recolhidos por segundo, nas primeiras 12 horas de recolha. Durante este período efectuou a recolha de 51% do total de documentos, a uma média de 8 documentos por segundo e 947 Kbit/s de informação (excluindo os *overheads* gerados pelo tráfego na rede).

Observa-se que o débito do sistema diminui à medida que os *Coleccionadores* terminam a recolha dos documentos alojados nos servidores que lhes foram atribuídos. Como descrevemos na subsecção III.C, cada *Coleccionador* possui uma estrutura de fios de execução responsáveis por efectuar a recolha dos documentos alojados nos servidores *Web*. Estes fios de execução encontram-se frequentemente bloqueados, enquanto esperam respostas de um servidor *Web*, de uma base de dados ou fazem um compasso de espera até efectuar um novo pedido a um servidor *Web*. Para rentabilizar a utilização de recursos de cada máquina que aloja um *Coleccionador*, quando um fio de execução fica bloqueado, outro fio de execução é executado. Assim sendo, à medida que o número de servidores *Web* com documentos por recolher diminui, existem menos fios de execução em cada máquina, ocorrendo frequentemente períodos de tempo em que todos os fios de execução se encontram bloqueados, consequentemente reduzindo o número de documentos recolhidos por unidade de tempo.

C. *Estatísticas da Web Portuguesa*

Estado	Significado	Numero de documentos	%
200	HTTP: OK	676261	85%
302	HTTP: Redirecção temporária	50263	6%
-11	Erro sistema	31086	4%
-3	Tipo interdito	12691	2%
404	HTTP: Ficheiro não encontrado	11624	1%
301	HTTP: Redirecção permanente	7212	1%
-10	Tempo de máximo de recolha excedido	4819	1%
-4	Tamanho máximo excedido	1142	0%
-7	Excluido pelo REP	414	0%
403	HTTP: Acesso Proibido	329	0%
400	HTTP: Pedido mal formado	276	0%
-1	Resposta HTTP inválida	270	0%
401	HTTP: Não autorizado	180	0%
500	HTTP: Erro interno ao servidor	40	0%
300	HTTP: Redirecção com escolhas múltiplas	15	0%
204	HTTP: Resposta sem conteudo	3	0%
502	HTTP: Erro de gateway	2	0%
503	HTTP: Serviço indisponivel	2	0%
	Total	796629	100%

Tabela 1. Respostas HTTP devolvidas pelos servidores *Web* durante a recolha (códigos com valores positivos) e códigos de erro gerados pelo Tarântula (códigos com valores negativos).

A Tabela 1 sumariza os resultados obtidos ao efectuarmos os pedidos HTTP para a recolha do domínio .PT. Através da sua análise, verificamos que 85% dos pedidos HTTP foram bem sucedidos. As redirecções totalizam 7% dos pedidos. Este número de redirecções dá-nos conta das características dinâmicas da estrutura da *Web* Portuguesa. Os erros de sistema representam 4% dos pedidos, mas após consulta dos ficheiros de *log*, concluímos que estes se deveram principalmente a dificuldades de comunicação com os servidores. O número de documentos de tipo interdito relativamente reduzido (3%), é devido à filtragem dos URLs efectuada durante a recolha. Esta filtragem consiste na exclusão dos ficheiros cuja extensão evidencie não se tratarem de documentos do tipo *text/html*. O número de documentos cuja recolha não foi autorizada pelo REP foi de apenas 414 documentos. Concluímos assim que a maior parte dos servidores *Web* não publicam o ficheiro *robots.txt* indispensável à implementação do REP.

URL Servidor	Número de documentos
freethemes.ip.pt	27568
www.terravista.pt	22964
foruns.star.pt	16253
gameover.forum.sapo.pt	14842
seguro.nortenet.pt	13625
www.medicosnainternet.pt	13341
enciclopediaverbo.clix.pt	11313
tucows.ipb.pt	10629
tucows.telepac.pt	10387
pdacentral.ip.pt	10240
pdacentral.KPNQwest.pt	9848
www.star.pt	8740
www.fotografia.em.pt	8678
mundial.sapo.pt	8607
linuxberg.ip.pt	8038
www.diariodigital.pt	7515
www.mni.pt	7038
www.instituto-camoes.pt	6177
www.infordesporto.pt	5055
mapas.netc.pt	4960
www.bragavirtual.pt	4648
planeta.clix.pt	3649
www.anedotas.com.pt	3538
www.saudenainternet.pt	3439
www.novaguarda.pt	3294

Tabela 2. Lista dos 25 servidores do domínio .PT que alojam maior número de documentos recolhidos.

A Tabela 2 mostra os 25 servidores mais visitados pelo Tarântula e os respectivos números de documentos recolhidos. Ao efectuarmos uma análise mais profunda da distribuição dos documentos pelos servidores, verificámos que não é uniforme: 80% dos documentos estão alojados em apenas 8% dos servidores visitados.

V. Conclusões e trabalho futuro

A recolha de documentos da *Web* é hoje em dia uma tarefa comum a várias aplicações. Neste artigo apresentámos um sistema de recolha de documentos *Web* genérico denominado Tarântula, dotado de características que permitem a sua fácil integração em qualquer sistema que necessite de efectuar recolhas de conteúdos online.

O Tarântula apresenta uma arquitectura modular e distribuída que permite aumentar a sua capacidade de recolha através da adição de máquinas ao sistema.

Os resultados obtidos ao efectuarmos uma recolha da *Web* Portuguesa, mostraram que o Tarântula tem capacidade para efectuar recolhas em grande escala, sem prejudicar os servidores *Web* que visita. Dos servidores 14068 visitados, apenas um *webmaster* se mostrou descontente com o comportamento do nosso *crawler*.

O trabalho futuro consistirá em transformar o Tarântula, num sistema de recolha especificamente direccionado para integração no motor de busca TUMBA. As alterações a efectuar visarão principalmente a sua adaptação para funcionar em estrita colaboração com um repositório de dados[4, 12]. Este repositório permitirá guardar versões dos documentos recolhidos, à medida que são actualizados com novos conteúdos, permitindo assim a possibilidade de manter várias recolhas em paralelo e ao longo do tempo. O Tarântula efectuará também recolha de informação a fim aplicar algoritmos de *ranking* [8, 19] sobre os documentos recolhidos. O Tarântula incluirá um novo módulo de detecção de páginas HTML escritas em Português [16], que permitirá estender a recolha da *Web* Portuguesa além do domínio .PT. A performance do Tarântula será também melhorada através do desenvolvimento de algoritmos de distribuição de trabalho mais sofisticados.

VI. Referências

[1] BERK E. and ANANIAN, C.S. *JLex: A Lexical Analyzer Generator for Java(TM)*. http://www.cs.princeton.edu/~appel/modern/java/JLex/, acedido em Maio de 2001.

[2] BERK, E. *JFlex – The Fast Scanner Generator for Java*. http://www.jflex.de/, acedido em Maio de 2001.

[3] BRIN, S. and PAGE, L. *The anatomy of a large-scale hypertextual Web search engine*. In Proceedings of the 7th International World Wide Web Conference, 1998, páginas 107-117.

[4] CAMPOS, J. P. and SILVA, M. J. *Versus: A model for a Web repository*. Submetido à 4ª Conferência de Redes de Computadores, 2001.

[5] CHAKRABARTI, S., BERG, M. and DOM, B. *Focused crawling: a new approach to topic-specific Web resource discovery*. In Proceedings of the 8th International World Wide Web Conference, 1998.

[6] CHO, J. and GARCIA-MOLINA H. *The evolution of the web and implications for an incremental crawler*. In Proceedings of the 26th International Conference on Very Large Data Bases, 2000, páginas 200-219.

[7] CHO, J.and GARCIA-MOLINA, H. and PAGE, L. *Efficient Crawling Through URL Ordering*. In Proceedings of the 7th International World Wide Web Conference, 1998, páginas 161-172.

[8] COSTA, M. e SILVA M. J. *Ranking no Motor de Busca Tumba*. Submetido à 4ª Conferência de Redes de Computadores, 2001.

[9] GOMES, D. *Tarântula – Sistema de Recolha de Documentos na Web*. Relatório de estágio, Faculdade de Ciências da Universidade de Lisboa, 2001.

[10] HARVEST. http://harvest.transarc.com, acedido em Janeiro de 2001.

[11] HEYDON, A., and NAJORK, M. *Mercator. A Scalable, Extensible Web Crawler*. World Wide Web, 1999, páginas 219-229.

[12] HIRAI J., RAGHAVAN S., GARCIA-MOLINA, H. and PAEPCKE. A. *WebBase: A repository of web pages*. In Proceedings of 9th International World Wide Web Conference, 2000, páginas 277-293.

[13] HOOD, E. *MIME. (Multipurpose Internet Mail Extensions).* http://www.nacs.uci.edu/indiv/ehood/MIME/MIME.html, acedido em Maio de 2001.

[14] KOSTER, M. *A Standard for Robot Exclusion.* http://info.webcrawler.com/mak/projects/robots/norobots.html, acedido em Março de 2001.

[15] KOSTER, M. *Guidelines for Robot Writers*. http://info.webcrawler.com/mak/projects/robots/guidelines.html, acedido em Março de 2001.

[16] MARTINS, B. and SILVA M. J. *Is it Portuguese? Language detection in large document collections* . Submetido à 4ª Conferência de Redes de Computadores, 2001.

[17] MCBRYAN, O. A. *GENVL and WWWW: Tools for taming the Web.* In Proceedings of the First International World Wide Web Conference, 1994, páginas 79-90.

[18] NORONHA, N., CAMPOS, J. P., GOMES, D., SILVA, M. J. and BORBINHA, J. *A Deposit for Digital Collections*. In Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries, 2001, páginas 200-211.

[19] PAGE, L., BRIN, S., MOTWANI, R. and WINOGRAD, T. *The PageRank citation ranking: Bringing order to the Web*. http://google.stanford.edu/~backrub/pageranksub.ps, acedido em Julho de 2001.

[20] PINKERTON, B. *Finding What People Want: Experiences with the WebCrawler*. In Proceedings of the Second International World Wide Web Conference, 1994.

[21] PostgreSQL . *A sophisticated Object Relational DBMS.* http://www.postgresql.org, acedido em Agosto de 2001.

[22] SUN MICROSYSTEMS INC. *Java (TM) Servlet Technology*. http://java.sun.com/products/servlet/, acedido em Janeiro de 2001.

[23] SUN MICROSYSTEMS INC. *JDBC (TM) Technology*. http://java.sun.com/products/jdbc/, acedido em Janeiro de 2001.

[24] SUN MICROSYSTEMS, INC. *What is the Java Platform?- Here's the quick guide*. http://java.sun.com/nav/whatis, acedido em Janeiro de 2001.

[25] SUN MICROSYSTEMS. *Java 2 Platform SE v1.3.1*. http://java.sun.com/j2se/1.3/docs/api/, acedido em Janeiro de 2001.

[26] THE APACHE SOFTWARE FOUNDATION. *Apache Project* http://httpd.apache.org/, acedido em Fevereiro de 2001.

[27] THE APACHE SOFTWARE FOUNDATION. *The Jakarta Site – Jakarta Tomcat*. http://jakarta.apache.org/tomcat/, acedido em Fevereiro de 2001.

[28] *The Java Program: Crawler.java.* http://cs.fit.edu/~ryan/java/programs/crawler/Crawler-java.html, acedido em Julho de 2001.

[29] *The Web Robots Pages.* http://info.webcrawler.com/mak/projects/robots/robots.html, acedido em Setembro de 2001.

[30] TUMBA. http://xldb.fc.ul.pt/tumba, acedido em Julho de 2001.

[31] W3C. *HTML 4.01 Specification*. http://www.w3.org/TR/html401/, acedido em Agosto de 2001.

[32] W3C. *HTTP- Hypertext Transfer Protocol Overview*. http://www.w3.org/Protocols/, acedido em Agosto de 2001.

[33] W3C. *Uniform Resource Locators: BNF*. http://www.w3.org/Addressing/URL/5_BNF.html, acedido em Agosto de 2001.

[34] WILLS, C. and MIKHAILOV M. *Towards a better understanding of web resources and server responses for improved caching*. In Proceedings of the 8th International World Wide Web Conference, 1999, páginas 153-165.