NetCensus
Medição da evolução dos conteúdos na web

Leopoldo O. e Silva   Joaquim Macedo   António Costa
Orlando Belo   Alexandre Santos

29 de Maio de 2002

Resumo

Neste documento, apresenta-se um sistema automático para obtenção de indicadores estatísticos a respeito dos conteúdos da Web publicamente acessíveis. O sistema percorre periodicamente a Web, colecta conteúdos, processa-os e armazena a informação extraída numa Base de Dados. Esta BD pode ser interrogada para geração dos mais diversos relatórios.

Embora já se tenham recolhido dados bastante valiosos com este sistema, subsistem falhas. Colectar dados periódicos à escala da Web Portuguesa não é um problema simples.

Com base na experiência resultante da operação do sistema, apresentam-se algumas soluções simples para os problemas encontrados e propostas de alterações a introduzir no modelo inicial.

1 - Motivação e objectivos

Nos dias de hoje, a Web é uma das mais importantes fontes de informação disponíveis. A sua principal diferença relativamente às fontes de informação convencionais é a facilidade de publicação da informação. Trata-se, sem dúvida, duma das características que constitui a chave do seu sucesso e popularidade e do seu crescimento exponencial.

O grande número e volume de dados disponibilizados e sua natureza distribuída e descentralizada, transformam operações simples como contagem, localização de informação de interesse e arquivo, em problemas bastante complexos e de muito difícl solução.

Entretanto quer pela sua crescente importância económica quer pelo papel que desempenha na Sociedade de Informação, têm sido mobilizados muitos recursos humanos e financeiros, publicos e privados para tentar encontrar soluções para esses problemas.

Que degradação teria para nós a utilidade do Web senão existissem motores de busca como o Google[8] e o Altavista[2] ou classificados como o

Yahoo![19] ou o Sapo?[16]. Existe um trabalho bastante avançado na adaptação da tecnologia tradicional de busca de informação e de classificação de conteúdos à escala do Web. Estes serviços assumem uma importância tal que já fazem parte do próprio Web.

O Internet Archive[9] tem a difícil missão de tentar manter um arquivo histórico do Web. Manter uma cópia em arquivo de todos documentos disponibilizados na Web para posterior utilização. Tal como não podemos estudar a História de Portugal do século XX sem recorrer aos arquivos da RTP ou da Emissora Nacional, certamente nao poderemos estudar a História do século XXI sem recorrer ao arquivo da Web.

Outros sistemas têm objectos de estudo mais específicos como certas características da web. O NetCraft[12] é um bom exemplo de um destes sistemas.

Da consulta da literatura é óbvio que os serviços existentes à escala mundial não têm cobertura global. Como não conseguem colectar toda a Web têm que definir uma prioridade dos conteúdos a visitar, muitas vezes ditada por leis de mercado ou outros interesses. Uma outra constatação importante são as restricções colocadas pelas empresas à divulgação dos seus resultados de investigação e os custos elevadíssimos para a compra da sua tecnologia. Estas razões justificam plenamente a pretensão, concretizada no Projecto NetCensus[1], de desenvolver uma solução própria para obtenção de indicadores estatísticos dos conteúdos Web disponibilizados em Portugal.

Os indicadores estatísticos pretendidos são baseados em contagens simples de conteúdos ou em somas dos respectivos volumes e a sua evolução ao longo do tempo. Todavia, em face da grande dimensão do espaço em análise trata-se duma tarefa bastante complexa e que antecipa (nomeadamente ao nível de descarga dos conteúdos) os problemas de busca e de arquivo histórico. De qualquer forma, há alguns requisitos próprios que não são disponibilizados pelos robots convencionais.

2 - Trabalho prévio

Na origem deste projecto esteve um sistema por nós desenvolvido, para o extinto Observatório das Ciências e Tecnologias, que visava obter indicadores estatísticos da web Portuguesa de uma forma periódica.

O sistema encontra-se em estado avançado de desenvolvimento conceiptual mas está implementado em hardware símples (pcs) com baixa largura de banda. Assim quando aplicado a uma rede relativamente grande os tempos de visita da mesma e as diversas operações realizadas sobre a informação recolhida são incomportáveis. Neste momento usam-se os prencípios e algorítmos dos motores de busca que, embora realizem uma tarefa distínta a nível de análise de informação, percorrem partes extensas

[1]Financiado no ano de 2001 pelo então Observatório da Ciência e Tecnologia

da web, tal como nós pretendemos.

A informação recolhida é vasta e compreende tanto informação proveniente dos protocolos usados como dos conteúdos recolhidos. Sobre esta é possível realizar os mais variados estudos que vão desde o volume ou quantidade de dados presentes numa determinada máquina, servidor ou domínio, até à distribuição do software servidor ou quantidade de subdomínios por domínio.

Uma das falhas de implementação do modelo actual é o facto de considerar as distíntas recolhas como conjuntos de informação também distíntos. Tal não é grave quando se olha para os indicadores estatísticos obtidos numa recolha como representando uma amostra da web, mas o facto é que entre recolhas, que representam a mesma web, existem diferenças nos valores obtidos difíceis de explicar.

Nomeadamente temos duas extensas recolhas realizadas à web Portuguesa, com as mesmas técnicas e condições de partida, que apresentam resultados bastante diferentes. Na necessidade de explicar estas diferenças, surgiu um problema novo e distínto do original, não só interessa medir os conteúdos de uma web mas também perceber como estes evoluiem.

3 - Modelo do sistema actual

O modelo actual (figura 1) actua da seguinte forma:

Um crawler percorre a web e recolhe conteúdos, estes são processados para obter indicadores estatísticos. O conteúdo original é descartado e os indicadores calculados armazenados num formato normalizado. Neste formato são também armazenados, pelo crawler, dados relativos à transação efectuada com a web.

Estes dados são carregados numa base de dados temporária (BDT) e daí numa base de dados relacional sobre a qual se realizam diversas interrogações e produzem relatórios.

O sistema encontra-se distribuído em duas máquinas, uma delas dedicada às bases de dados e a outra aos restantes módulos.

3.1 - Crawler

O crawler é responsável por percorrer a web e recolher informação. No sistema implementado usamos um que segue o algorítmo básico dos motores de busca[5].

O robot mantém uma lista de urls a visitar que é iniciada com um conjunto de urls semente. A estes aplica-se o seguinte algorítmo:

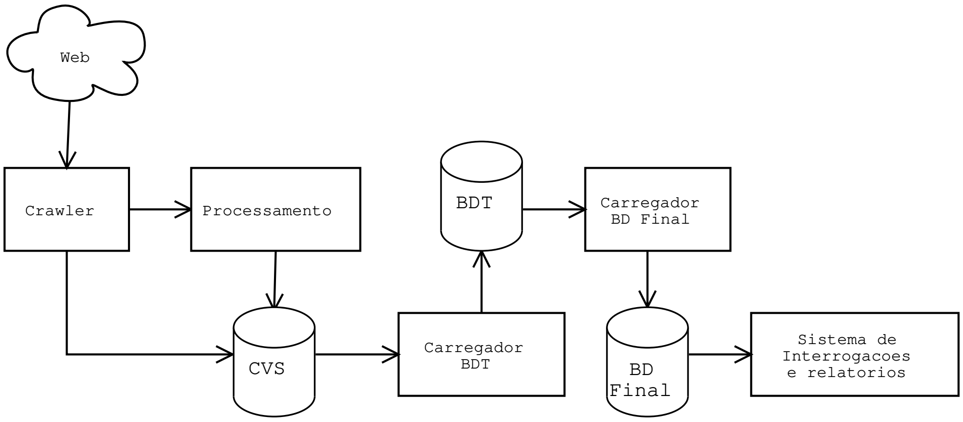

Figura 1: Modelo actual do NetCensus

- Descarrega o respectivo conteúdo.
- Extrai todos os links contidos nesse conteúdo.
- Os links extraidos que ainda não tenham sido vistos nesta recolha são adicionados à lista de urls a visitar.
- Se é a primeira vez que se vê este conteúdo envia-se-o aos módulos de processamento.

Este algorítmo simples tenta percorrer a web o mais exaustivamente possível. A porção da web percorrida depende largamente do conjunto de sementes iniciais e de alguns filtros aplicados aos links recolhidos, como por exemplo limitar a visita à web Portuguesa ou a um domínio particular.

O crawler armazena informação relativa às transações realizadas com a web. Entre esta informação encontra-se o url, o endereço ip, tipo e tamanho do conteúdo e grande parte dos restantes dados obtidos directamente dos protocolos usados.

3.2 - Módulo de Processamento

Este módulo recebe os conteúdos descarregados e processa-os com o intuito de guardar informação útil sobre os mesmos. Neste momento o módulo implementado processa conteúdos HTML, com o intuito de guardar alguma informação como o título, autor ou versão HTTP e um conjunto de somatórios de certas características como o número de frames, applets, areas e forms.

3.3 - Armazenamento temporário normalizado

Ao armazenar a informação obtida num formato normalizado, os módulos de crawling e processamento criam uma abstração simples para os módulos seguintes, permitindo que estes deixam de se preocupar com o funcionamento e implementação concreta do crawler.

O formato escolhido na implementação foi o cvs, um formato simples que se limita a representar registros de dados como texto separado por vírgulas. Este é facilmente tratável e simples de implementar, o suficiente para permitir uma rápida transação de informação para a base de dados temporária. O seu maior problema é a sua estrutura rígida onde cada campo dum registo cvs têm um significado concreto. Assim torna-se impossível representar estruturas mais complexas, como listas, num único registro cvs, o que leva a que existam vários registros associados à informação obtida por uma transação com a web. Por exemplo os links presentes numa página HTML apresentam-se como um conjunto de registros cvs.

3.4 - Base de dados temporária e seu carregador

A base de dados temporária (BDT) é practicamente uma cópia dos registros cvs. Existe uma tabela por tipo de registro cvs, preenchida com os respectivos registros.

De momento a única utilidade desta base de dados é facilitar o preenchimento da base de dados final, pois permite a marcação dos registros, como processados ou em erro, à medida que estes são carregados, algo que não era trivial realizar sobre o armazenamento temporário.

Teoricamente também se torna mais fácil paralelizar o carregador da base de dados final recorrendo a esta base de dados, no entanto tal ainda não está concretizado.

3.5 - Base de dados final e seu carregador

A base de dados final armazena a informação de forma relacional normalizada. É constituida por um conjunto de tabelas que representam as várias entidades existentes na informação recolhida, como as transações realizadas com a web, os servidores e urls visitados e, claro, os dados obtidos das páginas HTML pelo módulo de processamento.

Os dados são tratados durante a transferência da base de dados temporária para a final. Estes, representados sempre como strings na base de dados temporária são ajustados ao seu tipo real. As datas, como exemplo, passam de string ao tipo DATE.

O processo de carregamento da base de dados final é lento e pode durar, com o sistema existente e dependendo da escala da web estudada, dias, semanas ou meses. Ainda estamos a estudar formas de reduzir este tempo, tendo já identificado algumas causas possíveis tais como a existência de muita indexação e tempos de acesso a disco elevados.

3.6 - Módulo de interrogações e relatórios

Sobre a base de dados final podem-se realizar interrogações complexas, visto que todas as entidades estão, directa ou indirectamente, relacionadas.

Este módulo é o responsável por realizar essas interrogações e construir com a informação recolhida relatórios. Neste momento os relatórios são simples ficheiros de texto que resultam quase directamente dos dados retornados pelas interrogações.

Várias são as interrogações desenvolvidas sobre a base de dados, entre as mais interessantes encontram-se as seguintes:

- Quantidade e volume de dados por tipo MIME. Estas interrogações podem ser reduzidas a um determinado universo como um hostname ou domínio.
- Distribuição dos conteúdos por data de última modificação. Esta interrogação pretendia estudar a idade dos conteúdos na web. Como se verá existe uma falha na concepção do crawler que impede que esta interrogação seja fiável.
- Ocorrências de dados em páginas HTML, como títulos, autores, contagem de certas tags, etc....
- Distribuição das versões dos servidores visitados, versões HTTP e versões HTML.

Com a informação recolhida muitos outros dados podem ser obtidos como page ranks[14], grafos de links e outras medidas concretas como a fiabilidade, visibilidade e localidade dos conteúdos e referências[13].

4 - Resultados obtidos

O sistema foi desenvolvido sobre um domínio local e testado sobre toda a web estática Portuguesa, um teste ambicioso dada a escala do problema e a simplicidade do sistema e dos recursos em que ele opera. Sobre a web Portuguesa realizaram-se duas recolhas, que foram interrompidas quando se julgou, por análise do número de urls na lista a visitar, que o processo estava perto do fim.

Os dados aqui apresentados foram calculados sobre a base de dados temporária, pois a final demora um tempo incomportável a carregar. Assim estes resultados são experimentais e não confirmados.

Os gráficos da figura 2 mostram a distribuição dos recursos e seu volume por tipo MIME. De notar que as aplicações cuja quantidade é ínfima quando comparada com os recursos do tipo texto, representam a maior parte do volume de dados recolhidos.

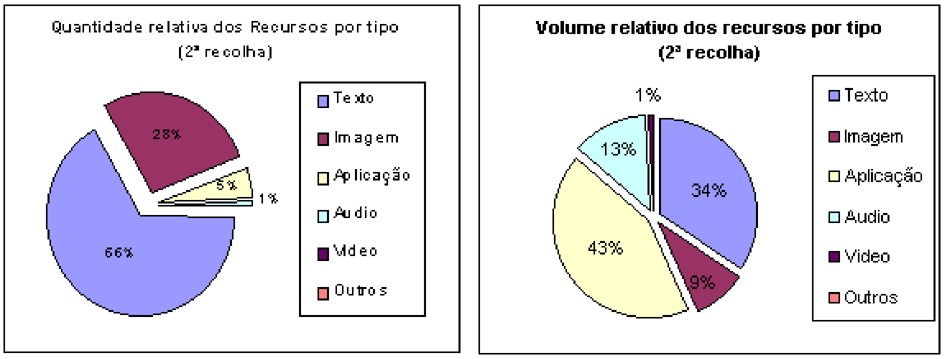

Figura 2: Distribuição dos conteúdos e do seu volume, por tipo MIME

Usando a data de última modificação de um documento construimos gráficos (figuras 3 e 4) que pretendem mostrar qual a idade dos conteúdos na web. Esta data nem sempre é obtida durante uma transação, principalmente se tratarem-se de conteúdos dinâmicamente gerados e é infiável como discutido num dos pontos seguintes. Nos gráficos apresentados não se contabilizam estes conteúdos sem data.

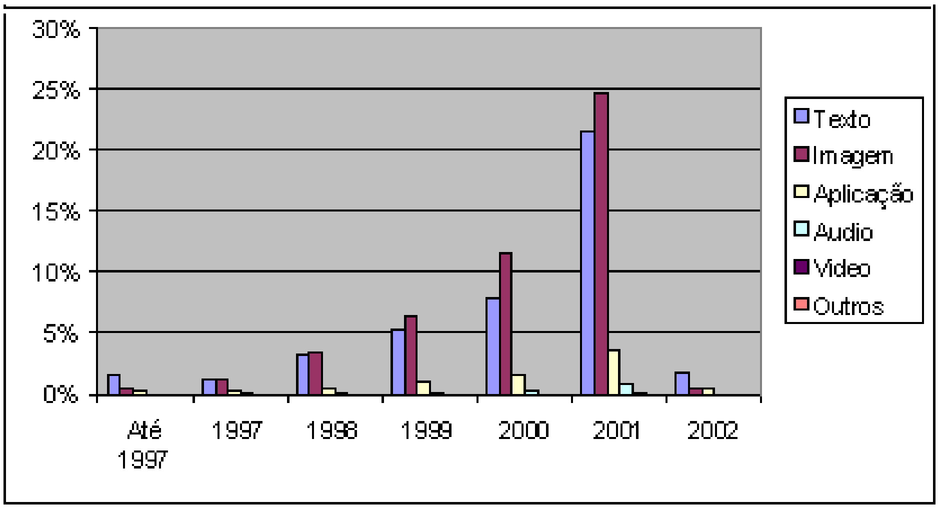

Figura 3: Distribuição dos conteúdos por tipo MIME e data de última modificação

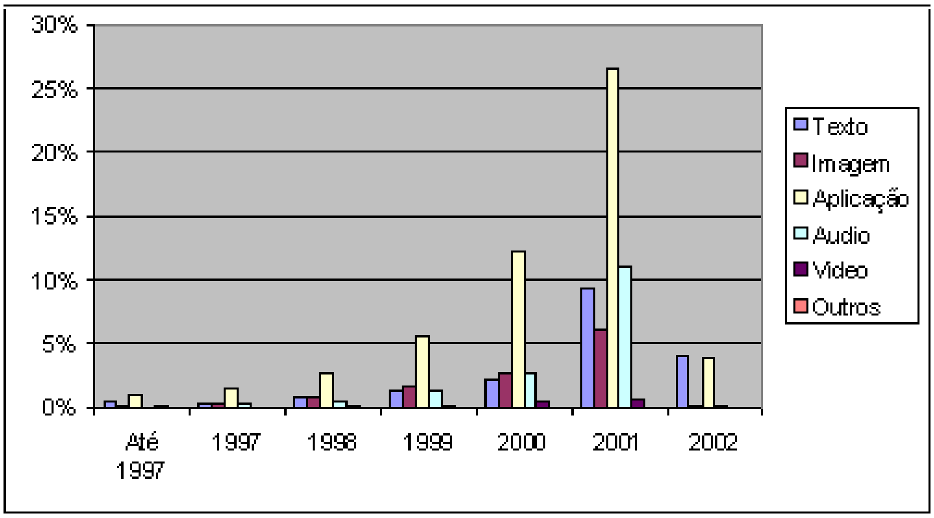

Figura 4: Distribuição do volume dos conteúdos por tipo MIME e data de última modificação

A análise aos conteúdos obtidos permite-nos obter dados sobre as características das páginas HTML recolhidas. Estas encontram-se sintetizadas na figura 5.

Os dados gerais sobre ambas as recolhas, quantos endereços IP, hostnames, urls e conteúdos únicos[2] foram vistos apresentam-se de seguida (figuras 6 e 7), dando uma ideia da escala do problema e revelando algumas falhas do sistema actual.

Embora o sistema tenha tido, em ambas as recolhas, as mesmas condições de partida (urls semente), pode-se verificar nestas tabelas que os universos visitados a nível de urls e conteúdos obtidos são dístintos.

Tal pode resultar de se terem percorrido universos distíntos ou a uma evolução da web, certos urls desaparecem e outros são criados (mudando os seus conteúdos). Numa tentativa de explicar estas diferenças concluímos que com a metodologia actual é impossível saber o que aconteceu aos urls por visitar na segunda recolha, sendo esta uma falha a resolver futuramente.

5 - Problemas do sistema actual

O modelo actual apresenta algumas falhas que, não nos impedindo de realizar os objectivos iniciais, obter indicadores estatísticos, nos impossibilitam de usar certos dados fiávelmente e relacionar correctamente as recolhas

[2]Conteúdos com mais de uma cópia são contabilizados uma única vez

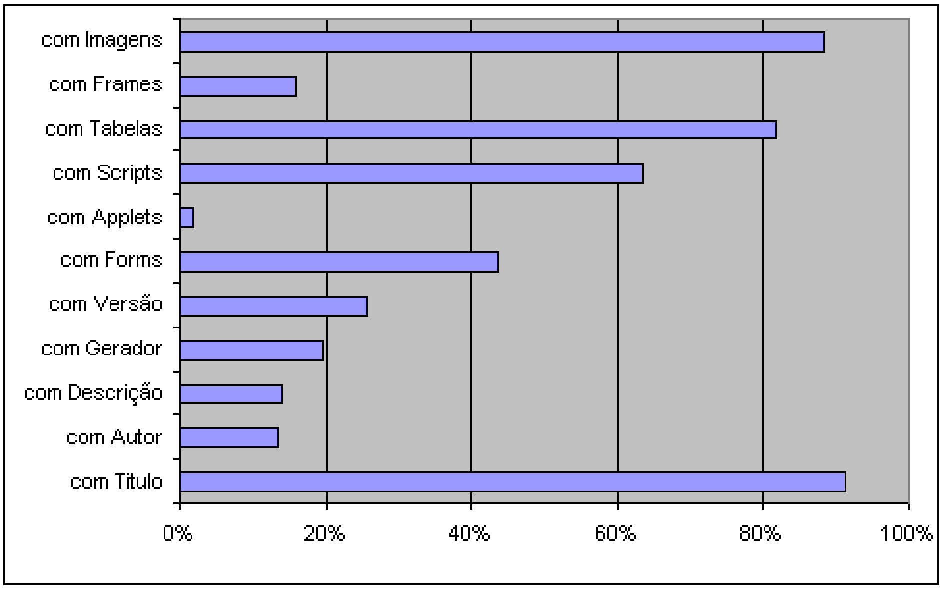

Figura 5: Ocurrências em páginas HTML.

.	1 recolha	2 recolha	Intersecção	União	1 \ 2	2 \ 1
IP	5.757	6.156	4.884	7.029	873	1.272
Hostnames	21.742	28.133	20.558	29.317	1.184	7.575
Urls	11.055.167	15.830.609	6.009.646	20.876.130	5.045.521	9.820.963
Conteúdos únicos	6.797.616	9.047.502	2.810.044	13.035.074	3.987.572	6.237.458

Figura 6: Universos percorridos nas recolhas

A data de última modificação de um recurso, informação que pretendiamos usar para determinar a idade dos documentos na web, é associada, pelo crawler, ao conteúdo único. Visto que podem existir conteúdos replicados, alguns deles armazenados à mais tempo que outros, associar a data ao conteúdo único torna-se um pouco arbitrário pois a que fica associada ao mesmo é a do primeiro que o crawler recolhe. O correcto seria associar a data à transação, garantindo que todos os conteúdos, mesmos os replicados, tivessem uma data associada. As interrogações teriam de ser remodeladas para não consultarem somente os conteúdos únicos mas também as transações que os tinham recolhido, no sentido de indicar qual a idade de todos os conteúdos, mesmo os replicados, na web.

Uma outra falha na implementação actual do modelo é o facto de as distíntas recolhas serem carregadas em base de dados finais também

.	Intersecção da 1 recolha	Intersecção da 2 recolha	Diferença entre 1 e 2	Diferença entre 2 e 1
IP	84.84%	79.34%	15.16%	20.66%
Hostnames	94.55%	73.07%	5.45%	26.93%
Urls	54.36%	37.96%	45.64%	62.04%
Conteúdos únicos	41.34%	31.06%	58.66%	68.94%

Figura 7: Comparação entre universos percorridos nas recolhas

distíntas. Tal levanta sérios problemas temporais quando se tenta relacionar as várias amostras, pois os cálculos realizados, principalmente os que envolvem unificações e intersecções, são morosos.

O algoritmo simples que o crawler segue também impede um correcto relacionamento das amostras, pois não garante que todos os urls conhecidos de uma recolha anterior sejam revistos numa recolha futura. Isto impede-nos de saber o que aconteceu a um determinado url não visto numa recolha. Terá desaparecido ou simplesmente não se encontrava acessível atravéz das sementes iniciais?

Muitas das diferenças observadas entre amostras podem também provir da forma simplista como o crawler lida com erros da camada de rede. Tal leva a que certos urls sejam registados em erro. Ora considerando que a não visita de um único url pode levar à não visita de uma parte significativa da rede, facilmente constatamos que as várias amostras podem representar porções da web totalmente diferentes. A distribuição dos erros obtidos na 2? recolha à web Portuguesa é apresentada na figura 8

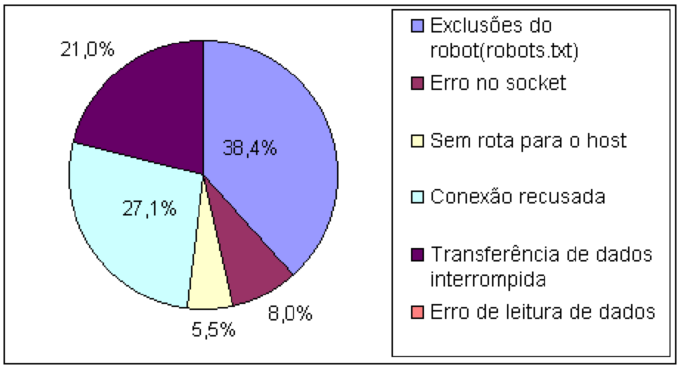

Figura 8: Erros durante a 2 recolha.

Os dois últimos problemas, a não revisita de urls e uma abordagem simplísta a erros de rede, leva a que seja impossível, no modelo actual, discernir se as diferenças registadas nas amostras se devem a uma evolução da web ou se representam partes diferentes desta.

Outros tipos de problemas surgem ao abordar-se uma rede extensa. Com os poucos recursos que temos, as recolhas, cargas e interrogações dificilmente são realizáveis em tempo útil.

A primeira recolha à web Portuguesa durou 45 dias e a segunda, com a largura de banda reduzida, 90 dias, tempos que não consideramos preocupantes. O que de facto nos preocupa é que deviamos carregar as bases de dados num tempo compatível com o das recolhas e isso não está a acontecer.

O formato cvs permite um rápido carregamento da base de dados temporária. Estas foram carregadas com os dados referentes a toda uma recolha em poucas horas. No entanto o carregamento é uma transação SQL de registros cvs para registros de uma tabela, todos eles sem indexação, ou seja uma transação rápida. Isto leva a que a base de dados temporária seja um conjunto de tabelas relacionadas por campos não indexados (urls na maioria dos casos).

Ora isto acarreta um problema grave para o carregador da base de dados final pois este, para ser eficiente, têm de carregar a base de dados final por partes. Primeiro a informação referente às transacções, depois a referente aos conteúdos examinados, a resultante do sístema de resolução de nomes, etc....

Torna-se assim impossível realizar interrogações completas à base de dados final antes de esta estar completamente carregada. No sistema implementado preencher a base de dados final, bastante complexa, com as recolhas à web Portuguesa não é possível em tempo útil.

Numa tentativa de paralelizar o processo usando distíntas bases de dados temporárias referentes a partes das recolhas, descobrímos uma falha no carregador da base de dados final que impede que vários corram fiávelmente sobre distíntas bases de dados temporárias e uma única base de dados final.

As próprias interrogações à base de dados são lentas quando a escala é a web Portuguesa. De facto quais os indíces a criar durante o preenchimento da base de dados, que contribuiem para atrasar o tempo de carregamento mas contribuiem para melhorar os tempos de resposta a interrogações, têm sido uma questão delicada e que precisa de ser revista.

6 - Soluções apontadas ao sistema actual

Sem alterar os objectivos iniciais, obter indicadores estatísticos da web, não são necessárias muitas alterações ao modelo actual para o tornar mais eficiente.

O crawler têm de garantir a revisita de todos os urls que ainda se consideram válidos, ou seja cuja última transação realizada com o url não resultou num erro de conteúdo inexistente.

Na realidade pequenas adaptações ao algorítmo actual permitem garantir a revisita dos urls válidos. Se, após duas recolhas distíntas, calcular-mos os urls que ficaram por visitar e alimentar-mos a próxima recolha com estes, garantimos que estes são visitados. Os urls cuja transação falhe devido a erros de falta de conectividade, também devem ser alimentados, pois na realidade um erro destes equivale a uma não revisita.

Com esta técnica simples é possível manter uma base de dados com um universo de urls conhecidos, cujo erro depende da velocidade do sistema ou, em última análise, da velocidade da recolha[6].

No entanto existem outros interesses na forma como o crawler percorre a web, nomeadamente deve-se tentar classificar as páginas de alguma forma de modo a visitar as mesmas a um ritmo diferente e adaptado ao ritmo de variação da página. As regras de delicadeza do robot também devem ser revistas para garantir que o crawler não sobre-carrega em demasia os servidores visitados.

A indexação de certos campos na base de dados temporária permitiria carregar a base de dados final com outras técnicas, nomeadamente carregar toda a informação referente a uma transação e respectivo conteúdo obtido, eliminando o carregamento por partes. Isto permitiria fazer interrogações completas à base de dados final durante o processo de carregamento.

A base de dados final têm de ser revista e adaptada ao problema, no sentido de tornar o carregamento mais eficaz.

Tornar a base de dados final única permitiria manter um conhecimento estável sobre os conteúdos existentes na web, no entanto, supondo que o carregador alterava a informação sobre um url aquando da sua revisita, não permitiria saber quais os conteúdos num tempo passado.

Para tal têm de se tornar a base de dados final histórica, garantindo que esta mantém informação relativa a todas as transações alguma vez realizadas com a web e, a estas, associe um prazo de validade. Este servirá para construir vistas da base de dados que representem o que era válido numa determinada data, permitindo assim estudar a evolução dos dados ao longo do tempo.

7 - Modelo proposto

Com tempo pensamos desenvolver e implementar um modelo aperfeiçoado do NetCensus. O modelo proposto tenta generalizar o anterior de modo a permitir outro tipo de tratamento da informação recolhida. Embora a escalabilidade do modelo seja um factor importante é difícil prever como será a implementação final. Alguns dos problemas levantados no modelo actual são, no entanto, abordados numa tentativa de permitir paralelismo entre os vários módulos.

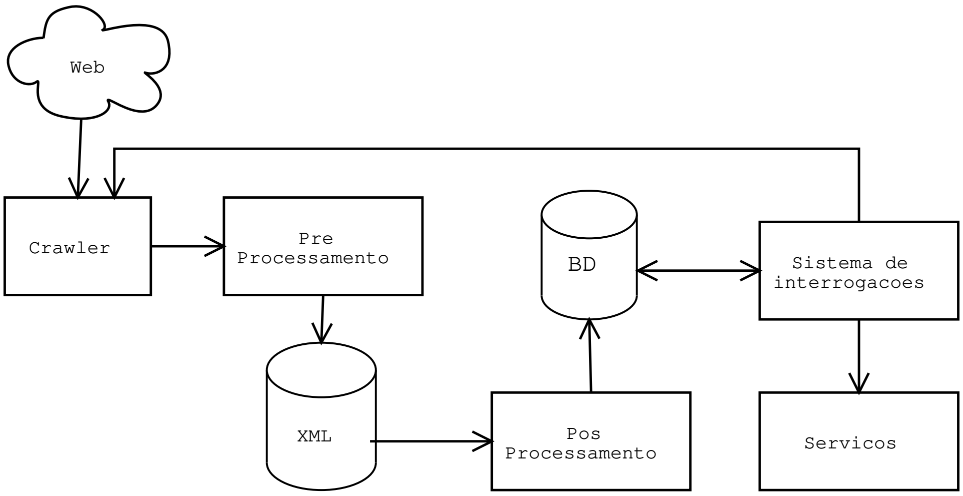

Figura 9: Modelo proposto para o NetCensus

O novo modelo (figura 9) apresenta dois módulos de processamento de dados. O de pré-processamento trata a informação recebida do crawler e o de pós-processamento encarrega-se de a carregar na base de dados. Cada um destes módulos pode conter vários sub-módulos dedicados a tratar a informação de forma distinta, como calcular indicadores estatísticos ou obter informação geográfica dos dados recolhidos.

Os dados obtidos pelo crawler e pré-processamento são armazenados num formato estruturado, provavelmente um formato XML, que garanta um único registro para toda a informação obtida duma transação. Isto permite que os módulos de pós-processamento encarem os registros como estruturas de dados referentes a uma única transação, facilitando o seu processamento.

O crawler deve garantir uma visita apropriada e extensiva ao universo pretendido, recolhendo conteúdos que envia, conjuntamente com dados provindos do protocolo usado, ao módulo de pré-processamento.

O modelo não apresenta uma base de dados temporária, tal deve-se ao facto de esta estar a ser usada por motivos bastante pragmáticos, garante-nos uma fácil consulta aos dados obtidos através do uso de SQL, tornando o processo de carga da base de dados final mais robusto, e permite calcular indicadores estatísticos sem que os dados estejam na base de dados final. Ora a introdução de um formato estruturado nos dados provenientes do módulo de pós-processamento também traz esta funcionalidade pois torna-se expedito desenvolver pequenos programas que consultem directamente esta informação.

Os módulos de pós-processamento carregam os dados obtidos em bases de dados. Neste caso uma base de dados histórica que registra os indicadores estatísticos obtidos.

O módulo de interrogações fornece uma abstração à base de dados, deve ser o único módulo que consulta a mesma directamente, todos os outros o devem usar quando necessitam de informação, o crawler, por exemplo, deve usá-lo para obter a lista de urls a revisitar. Este módulo também pode inserir informação na base de dados, especialmente se esta for morosa de calcular ou de especial interesse.

Os serviços usam o módulo de interrogações para produzirem relatórios. Um site é um dos serviços que pretendemos implementar.

Para lidar com a escala têm-se, necessariamente, de recorrer a mais recursos físicos, nomeadamente mais poder de computação e largura de banda. Tais recursos poderiam ser computadores melhores com imensa largura de banda, mas tal abordagem é controversa pois além de congestionar a web, torna-se ineficaz à medida que a web cresce[10]. As novas abordagens à distribuição de crawlers que começam a aparecer[7][11][18] podem solucionar este problema mas o facto é que o nosso modelo contém muitos outros módulos, além do crawler, que necessitam de ser eficazes.

Provavelmente a solução passará por um sistema com partes massivamente distríbuidas que, com um mecanismo de comunicação em grupo, consegue responder a interrogações globais. Outras partes ou serviços serão mais centralizados para impedir que a largura de banda usada pela comunicação em grupo exceda um limite que, eventualmente, seria comparável à usada pelos sistemas centralizados, um exemplo claro são os pageranks que caso tivessem de ser calculados, disseminando informação, sempre que necessário eram impraticáveis.

8 - Trabalho relacionado

(IN PROGRESS)

Serviços online:

Medidas da web[3][4][15]

crescimento da web. [17] uso da web (cliente side). [17] evolução dde certas características da web [12]

Motores de busca: Google[8], Alexa[1] , InternetArchive[9]
RIPE Geocensus. SIRE.

9 - Conclusões

(IN PROGRESS)

- Focar a necessidade de distribuir de novo, política centralizada além de ineficaz é polémica.
- Muito mais que um estudo à evolução dos conteúdos, o modelo apresentado permite estudar a evolução da própria web.
- Cópia da web histórica, teoricamente é possível aplicar todos os processos actuais.

Olhando para o sistema actualmente implementado, na sua simplicidade, julgamo-lo interessante pois permite obter dados sobre os conteúdos da web que, aparentemente, não existem ou não estão publicados. O objectivo de construir um histórico é um desafio a enfrentar e de valor científico considerável pois permite estudos sobre os conteúdos que vão muito além da obtenção pontual de indicadores estatísticos.

Referências

[1] Alexa. http://www.alexa.com.

[2] AltaVista. http://www.altavista.com.

[3] T. Bray. Measuring the web. http://www5conf.inria.fr/fich_html/papers/P9/Overview.html.

[4] O. W. Characterization. http://wcp.oclc.org/.

[5] J. Cho. Crawling the web, discovery and maintenance of large-scale web data.

[6] J. Cho and H. Garcia-Molina. The evolution of the web and implications for an incremental crawler. Technical report, Computer Science Department, Stanford University, 2000.

[7] J. Cho and H. Garcia-Molina. Parallel crawlers. Technical report, Stanford University, 2002.

[8] Google. http://www.google.com.

[9] InternetArchive. http://www.archive.org.

[10] M. Koster. Robots in the web: threat or treat? http://robotstxt.org/wc/threat-or-treat.html.

[11] M. Najork and A. Heydon. High-performance web crawling. Technical report, Compaq, Systems Research Center, 2001.

[12] NetCraft. http://www.netcraft.net.

[13] M. J. Nicolau, J. Macedo, and A. Costa. Caracterização da informação www na rccn. Technical report, Universidade do Minho, 1997.

[14] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[15] W. W. C. Repository. http://repository.cs.vt.edu/.

[16] Sapo. http://www.sapo.pt.

[17] I. Statistics. http://www.mit.edu/people/mkgray/net/.

[18] P. B. S. Vigna, B. Codenotti, and M. Santini. Trovatore: Torwards a highly scalable distributed web crawler.

[19] Yahoo! http://www.yahoo.com.