Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Preservação da Web através de replicação distribuída em larga escala

Fundação para a Computação Científica Nacional

André Ricardo Lopes Nogueira

Mestrado em Engenharia Informática

2008

Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Preservação da Web através de replicação distribuída em larga escala

projecto realizado na

Fundação para a Computação Científica Nacional

André Ricardo Lopes Nogueira

Projecto orientado pelo Prof. Dr. Pedro Veiga
e co-orientado por Dr. Daniel Gomes

Mestrado em Engenharia Informática

2008

Declaração

André Ricardo Lopes Nogueira, aluno nº 29505 da Faculdade de Ciências da Universidade de Lisboa, declara ceder os seus direitos de cópia sobre o seu Relatório de Projecto em Engenharia Informática, intitulado "Preservação da Web através de replicação distribuída em larga escala", realizado no ano lectivo de 2007/2008 à Faculdade de Ciências da Universidade de Lisboa para o efeito de arquivo e consulta nas suas bibliotecas e publicação do mesmo em formato electrónico na Internet.

FCUL, 31 de Outubro de 2008

Prof. Dr. Pedro Veiga, supervisor do projecto de *André Ricardo Lopes Nogueira*, aluno da Faculdade de Ciências da Universidade de Lisboa, declara concordar com a divulgação do Relatório do Projecto em Engenharia Informática, intitulado "Preservação da Web através de replicação distribuída em larga escala".

Lisboa, 31 de Outubro de 2008

Resumo

A Web é a maior fonte de informação alguma vez construída. A tendência verificada nos últimos anos indica que a popularidade da Web vai continuar a aumentar no futuro, assim como a quantidade de informação que nela é exclusivamente publicada. No entanto, a informação publicada na Web está disponível durante um período de tempo muito curto, findo o qual, por regra se perde para sempre. Surge assim o interesse na criação de arquivos da Web que permitam preservar esta informação para gerações vindouras.

Para preservarem a informação os arquivos da Web requerem sistemas com elevada capacidade de armazenamento. Tradicionalmente, o armazenamento da informação é feito de uma forma centralizada. Contudo, esta aproximação é susceptível a perda de informação, caso ocorram falhas no sistema de armazenamento central.

O trabalho apresentado nesta tese enquadra-se no projecto de Arquivo da Web Portuguesa[1], em curso na Fundação para a Computação Científica Nacional[2]. Este trabalho tem como objectivo a criação de um sistema de replicação distribuído que permita tolerar falhas nos sistemas de armazenamento de arquivos da Web, através da replicação dos conteúdos arquivados por computadores espalhados pela Internet.

PALAVRAS-CHAVE: Arquivos da Web, Bibliotecas digitais, Sistemas distribuídos, Formato ARC, Preservação digital

[1] http://arquivo-web.fccn.pt/
[2] http://www.fccn.pt/

Abstract

The Web is the largest source of information ever built. The trend in recent years indicates that the popularity of the Web will continue to grow in the future, as well as the amount of information solely published on it. However, the information published on the Web is available for a very short period of time, after which, as a rule, is lost forever. This motivates the creation of web archives that allow the preservation of this information for future generations.

To preserve the information, web archives require high storage capacity systems. The storage of information is usually performed in a centralized manner. However, this approach is susceptible to loss of information, if failures in the central storage system occur.

The work presented in this thesis is within the scope of the Portuguese Web Archive[3], a project of the Foundation for National Scientific Computing[4]. This work aims at creating a distributed replication system that allows to tolerate failures in the storage systems of web archives, through the replication of the archived contents over computers across the Internet.

KEYWORDS: Web archives, Digital libraries, Distributed systems, Arc format, Digital preservation

[3]http://arquivo-web.fccn.pt/
[4]http://www.fccn.pt/

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A Web possibilita que cada um de nós disponibilize informação acessível a todos sem necessidade de recurso aos editores e meios de impressão tradicionais. Esta nova realidade suscita a necessidade da preservação desta informação, para que não se perca irremediavelmente e esteja acessível às gerações futuras [9]. As comunidades nacionais dos países, sensibilizadas para a urgência da preservação da informação publicada na Web, desencadearam há vários anos iniciativas formais de preservação e catalogação da informação digital [1, 8]. Os arquivos da Web são entidades que têm como objectivo preservar e disponibilizar a informação publicada na Web. A primeira iniciativa de arquivo da Web foi levada a cabo pelo Internet Archive[1]. Esta instituição sem fins lucrativos norte-americana, tem como objectivo recolher e arquivar conteúdos da Web à escala mundial [13]. Acontecimentos históricos de grande importância como o Furacão Katrina originaram acções de arquivo extraordinárias por parte do Internet Archive, para que este acontecimento que marcou a história dos Estados Unidos da América ficasse documentado o mais exaustivamente possível [12]. Contudo, é difícil para uma única organização fazer um arquivo exaustivo de todos os conteúdos publicados na Web, porque esta está em permanente mutação e muitos conteúdos desaparecem antes de poderem ser arquivados. Devido a este facto, muito países criaram os seus próprios arquivos da Web, incluindo Portugal com o projecto de Arquivo da Web Portuguesa[2].

Os arquivos da Web efectuam recolhas periódicas dos conteúdos publicados na Web. A recolha dos conteúdos é realizada por um *Batedor*. A actividade de um batedor consiste num processo cíclico. Este processo inicia-se a partir de um conjunto de endereços iniciais, denominados raízes. A partir das raízes recolhem-se os primeiros conteúdos, sendo feita a extracção das ligações para novos conteúdos, seguindo-se um novo ciclo de recolha. Em cada nova recolha, o batedor utiliza como raízes as páginas de entrada de todos os sítios recolhidos com sucesso anteri-

[1] http://www.archive.org/
[2] http://arquivo-web.fccn.pt/

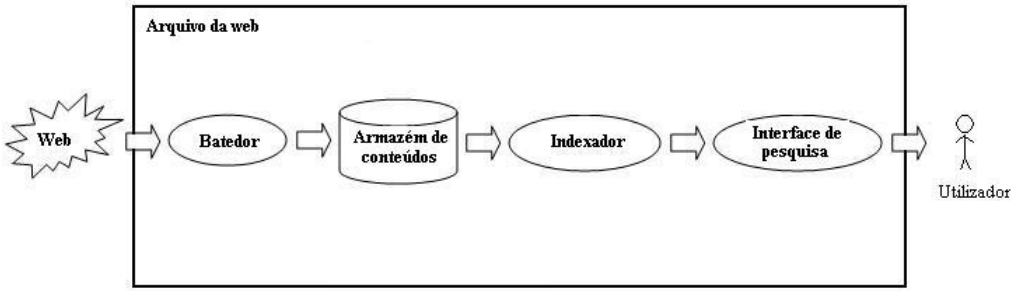


Figura 1.1: Arquitectura geral de um arquivo da Web.

ormente. Os conteúdos recolhidos são armazenados no *Armazém de conteúdos*. No fim da recolha, é realizado o processo de indexação dos conteúdos. Este processo consiste em gerar estruturas de dados que permitam realizar pesquisas rápidas e eficazes. A indexação é executada pelo *Indexador*, que pode criar índices de termos e URL. Os índices de termos permitem pesquisa por termos-chave, os índices de URL permitem a pesquisa por localização de um recurso na Internet. Terminado o processo de indexação, os utilizadores podem pesquisar na informação recolhida, utilizando uma *Interface de pesquisa*. Esta interface permite que os utilizadores possam pesquisar na informação arquivada através dos índices. A Figura 1.1 ilustra a arquitectura geral de um arquivo da Web.

A contínua recolha de conteúdos da Web produz grandes quantidades de informação, requerendo a existência de sistemas com elevada capacidade de armazenamento. Tradicionalmente, o armazenamento da informação é feito de uma forma centralizada, sendo realizadas cópias de segurança. Contudo, esta estratégia implica a duplicação dos custos de armazenamento. Apesar de os sistemas de armazenamento centralizados serem confiáveis e estáveis, caso ocorra uma falha causada, por exemplo, por uma catástrofe natural, toda a informação armazenada pode ficar comprometida, sendo possível que a informação se perca para sempre.

Surge assim o interesse na criação de sistemas que contribuam para garantir a preservação dos conteúdos arquivados a baixo custo, em caso de falha do armazém central de conteúdos dos arquivos da Web.

1.1 Objectivos

O trabalho apresentado nesta tese foca-se no desenho e desenvolvimento de um sistema de replicação distribuído, que permita a um utilizador da Internet disponibilizar espaço em disco do seu computador para armazenar uma parte da informação arquivada no armazém de conteúdos de um arquivo da Web, através da instalação de uma pequena aplicação. Pretende-se assim que em caso de falha do sistema de armazenamento central de um arquivo da Web, a colecção de conteúdos possa vir a ser

recuperada a partir da informação armazenada nos computadores dos utilizadores. O número de computadores ligados à Internet cresce rapidamente. Estima-se que até ao ano de 2015 existam mais de mil milhões de computadores ligados à rede mundial. Nos dias de hoje, um computador vulgar apresenta uma capacidade média de armazenamento de 100 GBytes. Se 100 milhões de utilizadores disponibilizarem 10% (10 GBytes) de espaço de armazenamento dos seus computadores, o total de espaço disponível seria 1 EBytes[2]. Este espaço de armazenamento ultrapassa a capacidade de qualquer sistema de armazenamento centralizado disponível actualmente. Este sistema foi denominado rArc, que significa replicador de Arcs. O Arc é a denominação do formato usado por arquivos da Web para armazenarem os conteúdos em ficheiros [6]. Este formato é utilizado pelo batedor Heritrix[3] para armazenar os conteúdos que recolhe da Web [18, 24]. Este batedor é utilizado pela maioria dos arquivos da Web. O rArc foi criado no contexto do um arquivo da Web, contudo, este sistema poderá ser também utilizado noutras áreas, que tenham como objectivo a preservação da informação digital.

O trabalho apresentado nesta tese foi realizado no âmbito do projecto Arquivo da Web Portuguesa (AWP), em curso na Fundação para a Computação Científica Nacional (FCCN). O AWP tem como objectivo a criação de um sistema que terá como missão recolher periodicamente, armazenar, preservar e disponibilizar a informação que é publicada na Web portuguesa. Para atingir este objectivo, o AWP utiliza as ferramentas disponibilizadas pelo projecto Archive-access[4] . Este projecto aglutina várias ferramentas gratuitas e de código aberto úteis para arquivar a Web. Por exemplo, o Heritrix é uma das ferramentas disponibilizadas por este projecto.

1.2 Metodologia

O rArc foi desenvolvido segundo o modelo em espiral. Este modelo de desenvolvimento de software permite lidar com requisitos tolerando alterações ao longo do tempo [20]. Este modelo organiza o desenvolvimento como um processo iterativo em que várias fases se sucedem até se obter o sistema final, como ilustra a Figura 1.2. O modelo em espiral permite que ao longo de cada iteração que se obtenham versões do sistema cada vez mais completas.

Para este trabalho o modelo espiral teve duas iterações. Na primeira iteração foi desenvolvido o processo que permite replicar os conteúdos armazenados no armazém de conteúdos do arquivo da Web por computadores espalhados pela Internet. Na segunda iteração foi desenvolvido o processo que permite recuperar os conteúdos replicados, para restaurar o armazém de conteúdos, caso este tenha sofrido uma

[3]http://crawler.archive.org/
[4]http://archive-access.sourceforge.net/

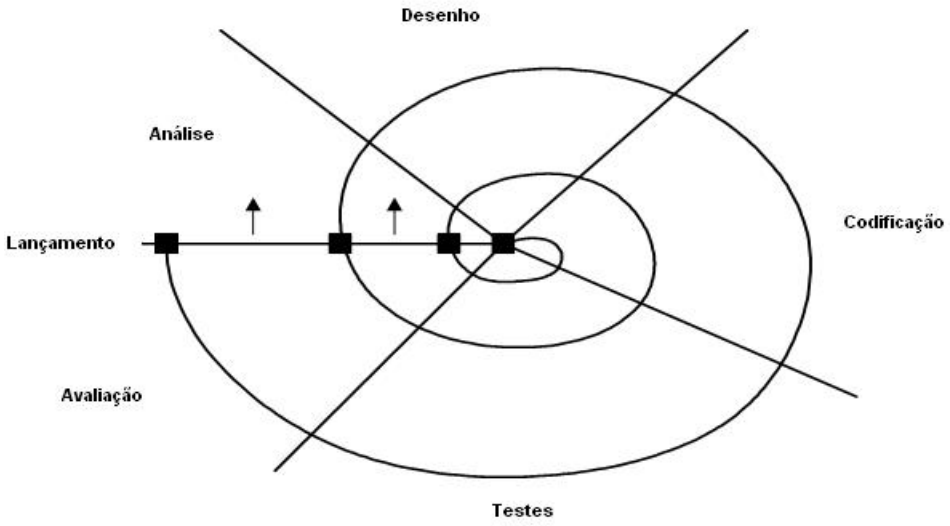


Figura 1.2: Modelo em espiral.

perda dos conteúdos que armazena.

1.3 Organização do documento

Este documento está organizado da seguinte forma: no Capítulo 2 é apresentado a arquitectura e o funcionamento do rArc; no Capítulo 3 são apresentadas as tecnologias utilizadas no desenvolvimento do trabalho. No Capítulo 4 são apresentados os resultados das experiências realizadas ao rArc; e no Capítulo 5 é apresentado trabalho relacionado. O Capítulo 6 conclui a tese e apresenta o trabalho futuro.

Capítulo 2

rArc: O Replicador de Ficheiros Arc

O rArc é um sistema de replicação distribuído que pretende replicar os conteúdos armazenados no sistema de armazenamento de um arquivo da Web, por computadores ligados à Internet. Caso ocorra uma falha no sistema de armazenamento e se verifique a perda dos conteúdos armazenados, o rArc vai recuperar os conteúdos replicados pelos computadores, para tentar recuperar os conteúdos perdidos. Poderá não ser possível recuperar todos os conteúdos, todavia é preferível tentar recuperar alguns conteúdos do que perder todos. Neste Capítulo é apresentado o desenvolvimento do rArc.

Na Secção 2.1 é apresentado o formato Arc. Este formato é utilizado por arquivos da Web para armazenar os conteúdos recolhidos da Web. Na Secção 2.2 são apresentados os requisitos do sistema. Na Secção 2.3 é apresentada a arquitectura do rArc e na Secção 2.4 é apresentado o funcionamento do sistema. Nesta secção são apresentados os vários componentes do sistema. De seguida é apresentado o funcionamento do rArc, na Secção 2.4. Nesta secção é apresentado o modo como os conteúdos são replicados e recuperados dos computadores dos utilizadores. Na Secção 2.5 são apresentados os vários mecanismos de segurança implementados. A Secção 2.6 conclui este Capítulo fazendo uma análise do desenho do sistema.

2.1 Formato Arc

O formato Arc é uma especificação de ficheiro desenvolvido pelo Internet Archive que é utilizado pelo batedor Heritrix para armazenar os conteúdos recolhidos da web. O formato Arc foi desenvolvido tendo em conta os seguintes requisitos:

- O ficheiro deve permitir a agregação de conteúdos que estão identificados e a descompressão dos mesmos sem o recurso a um índice;

- O formato deve permitir a acomodação de conteúdos recolhidos através de vários protocolos de rede, incluindo HTTP, FTP, *News*, *Gopher*, e *Mail*;
- Deve ser possível concatenar vários ficheiros de modo a criar um fluxo de dados.

Um ficheiro Arc contém um ou mais conteúdos da Web. Cada conteúdo é antecedido de um cabeçalho que contém atributos que o caracterizam e identificam, de forma ser possível a sua recuperação. A Figura 2.1 apresenta a estrutura do ficheiro Arc.

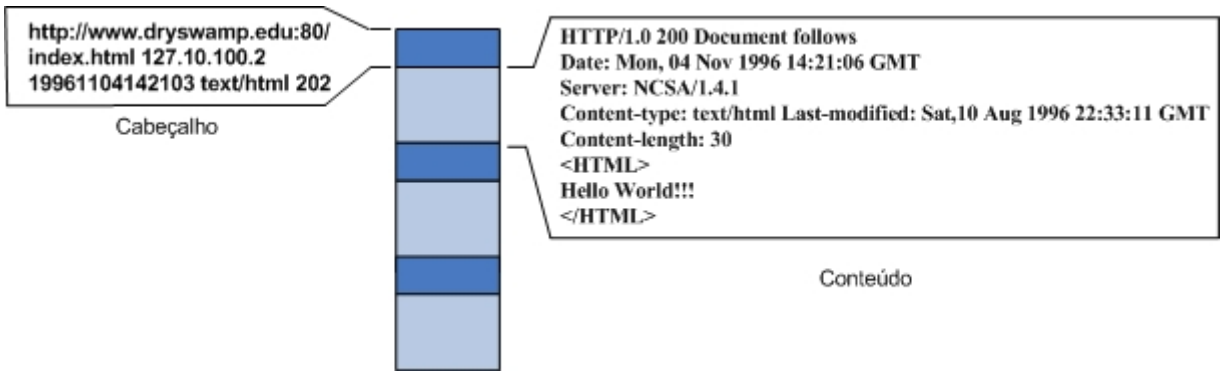


Figura 2.1: Formato do ficheiro Arc.

Cada cabeçalho contém os seguintes atributos:

- *URL* − endereço de localização do conteúdo;
- *IP-Address* − endereço IP da máquina de onde foi recolhido;
- *Archive-date* − data do arquivo;
- *Content-type* − tipo de conteúdo;
- *Archive-length* − tamanho do conteúdo.

O tamanho de um ficheiro Arc depende de dois factores: o tamanho máximo que de conteúdo recolhido pelo Heritrix e o tamanho definido por omissão no Heritrix. O tamanho definido por omissão no Heritrix é de 100 MBytes. Habitualmente os ficheiros Arc são comprimidos para optimizar a eficiência no armazenamento.

Actualmente o Internet Archive encontra-se a desenvolver um novo formato, o WARC[14]. Este novo formato é o sucessor do formato Arc. Este formato trará vantagens tais como permitir a adição de relações entre os conteúdos, tornando-se assim possível guardar por exemplo informação para gestão de conteúdos duplicados.

2.2 Requisitos

A análise de requisitos é o estudo das características que o sistema deverá ter para atender às necessidades e expectativas do cliente [20]. Nesta análise fica definido quais as propriedades e características que o sistema deverá corresponder. No caso do rArc existem vários clientes, os arquivos da Web espalhados pelo mundo. Para se definir um conjunto de requisitos que permitissem que o rArc correspondesse às necessidades dos vários arquivos, foram analisados trabalhos acerca dos requisitos de arquivos da Web. Foram definidos os seguintes requisitos:

Disponibilidade : É imperativo que o rArc esteja operacional 24 horas por dia para que novos utilizadores possam aderir à iniciativa, e porque sendo a Internet uma rede à escala mundial, estes poderão surgir a qualquer hora, de qualquer fuso horário. O rArc deverá também garantir que mesmo que exista um número elevado de contribuidores, o sistema continue a garantir o serviço.

Confidencialidade : A confidencialidade é a medida em que um serviço/informação está protegido contra o acesso em leitura de intrusos [27]. Alguns conteúdos recolhidos pelo arquivo da Web podem conter informação sensível. O *spam* consistem no envio em massa e indiscriminado de mensagens de correio electrónico. Estando os conteúdos do arquivo replicados pelos computadores dos contribuidores, um utilizador malicioso poderia extrair os endereços das páginas dos conteúdos replicados no seu computador, para efectuar *spam*. É necessário que o rArc garanta a confidencialidade dos conteúdos replicados.

Integridade : Os arquivos da Web armazenam conteúdos em sistemas de armazenamento centralizados. Estes são sistemas estáveis, controlados e confiáveis. Por sua vez, os sistemas de armazenamento dos computadores dos utilizadores são instáveis, pouco controlados e não são confiáveis. Por exemplo, um contribuidor poderá apagar os conteúdos que replicou, ou o disco poderá sofrer uma avaria física corrompendo os conteúdos armazenados. É necessário que o rArc verifique periodicamente o estado da informação replicada, para que esta esteja disponível e válida caso seja necessário.

Portabilidade : A portabilidade de um programa de computador é a sua capacidade de ser executado em diferentes plataformas (seja de hardware ou de software). Sendo a Internet uma rede mundial de computadores heterogéneos, é importante que o rARC possa ser executado no maior número possível de plataformas diferentes, para que um maior número de utilizadores da Internet possam contribuir com espaço de armazenamento.

Usabilidade : A usabilidade é uma métrica/atributo que avalia a facilidade de utilização das interfaces de utilizador [19] . A fraca usabilidade pode desmotivar os utilizadores da Internet a contribuírem. Torna-se assim imperativo que a instalação e adesão a um projecto de arquivo seja facilmente executado por qualquer utilizador da Internet.

Autenticidade : A autenticidade é a medida em que um serviço/informação é genuíno, i.e., está protegido contra a personificação por intrusos [27]. Por exemplo, se uma mensagem de correio electrónico estiver assinada digitalmente, é possível confirmar a sua autenticidade. Garantir a autenticidade da informação que está armazenada nos computadores do utilizadores é imperativo. Caso contrário um utilizador malicioso poderá alterar os conteúdos para, por exemplo, introduzir vírus informático num arquivo da Web durante um processo de recuperação dos conteúdos. Assim, o rArc terá de dispor de um mecanismo que confirme a autenticidade dos conteúdos recuperados.

2.3 Arquitectura

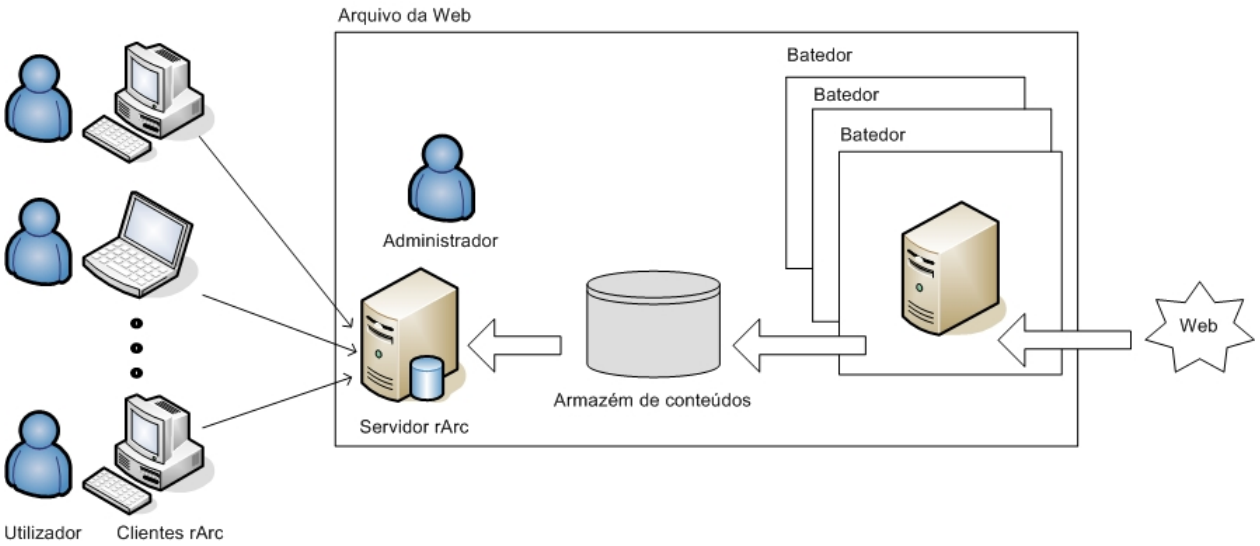


Figura 2.2: Integração do rArc com um arquivo da Web.

A Figura 2.2 apresenta a interação entre o rArc e a interacção com os outros componentes do arquivo da Web. O arquivo tem vários batedores que são responsáveis por recolher os conteúdos da Web. Os conteúdos recolhidos são armazenados no armazém de conteúdos sob a forma de ficheiros Arc. São este ficheiros que são replicados pelos computadores dos utilizadores. Contudo, estes não são replicados na seu formato original. Para cada ficheiro Arc é criado uma cápsula. No armazém de conteúdos do arquivo estão armazenadas os ficheiros Arc e as cápsulas.

Para um utilizador poder participar num projecto de arquivo terá de instalar um cliente rArc no seu computador. Cada utilizador tem uma credencial. Esta permite que o utilizador possa criar um identificador, durante a instalação do cliente. O identificador permite que o cliente se autentique perante o servidor. Durante o processo de instalação o utilizador define qual o espaço que pretende oferecer para o armazenamento das cápsulas. O espaço oferecido terá de ser superior ao tamanho de uma cápsula. Durante o processo de instalação o utilizador define também qual a largura de banda que o cliente pode utilizar na transferências das cápsulas. Tanto o espaço oferecido como a largura de banda ficam registadas num ficheiro de configuração.

O rArc apresenta uma arquitectura cliente-servidor, como ilustra a Figura 2.2. O servidor rArc está instalado no arquivo, o cliente rArc está instalado no computador do utilizador. O servidor pode estar um de dois estados: replicação ou recuperação. No estado de replicação o servidor tem como objectivo replicar as cápsulas pelos clientes. No estado de recuperação, o servidor tem como objectivo recuperar as cápsulas replicadas, para permitir o restauro do armazém de conteúdos, caso ocorra uma falha. O rArc assume que quando ocorre a falha no armazém de conteúdos, todos os seus conteúdos se perdem. Assim, os clientes têm a responsabilidade de descarregar as cápsulas, quando o servidor está no estado de replicação e enviar as cápsulas para o servidor, quando o servidor está no estado de recuperação. O servidor pode atender a vários clientes em simultâneo. Para isso o servidor lança para cada cliente uma *thread*. Assim o servidor é multi-tarefa.

Ficaram definidos os seguintes intervenientes para o rArc:

Utilizador : pessoa que instala e configura o cliente.

Administrador : pessoa responsável por administrar o servidor.

Fazem parte do rArc os seguintes componentes:

Computador do utilizador : computador do utilizador onde está instalado o cliente.

Identificador : conjunto de informações que permite ao cliente autenticar-se no servidor.

Credencial : conjunto de informações que permite que o utilizador crie identificadores para os clientes.

Ficheiro de configuração : local onde estão definidos o espaço de armazenamento oferecido e a largura de banda utilizada para a transferência das cápsulas.

Cápsula : formato em que os ficheiros Arc são replicados pelos computadores dos utilizadores.

Ficheiro de registo : local onde estão conjunto de informações contém informações sobre as cápsulas.

Base de dados : local onde o servidor armazena as informações sobre os utilizadores e as cápsulas.

Armazém de conteúdos : local onde estão armazenadas as cápsulas e os ficheiros Arc.

Depois de identificar os intervenientes e os componentes, foram definidos os seguintes pressupostos sobre os mesmos:

Computador do utilizador

- Tem no mínimo 100 MBytes de espaço livre para armazenamento.
- Tem no máximo o mesmo espaço de armazenamento do sistema central do arquivo da Web.

Cliente

- Armazena uma ou mais cápsulas.
- Não armazena cápsulas repetidas.
- É iniciado cada vez que o computador do utilizador arranca.

Cápsula

- É atómica.
- É identificada pelo seu nome.
- É constituída por um ficheiro Arc e informação de segurança.
- Tem aproximadamente 100 MBytes de tamanho.
- Pode estar armazenada em vários computadores.

Utilizador

- Pode ter um ou mais clientes instalados em computadores diferentes.
- Pode aumentar o espaço de armazenamento utilizado pelo cliente.

- Pode controlar a largura de banda utilizada pelo cliente.

Base de dados

- É replicada de forma a sobreviver à perda da informação no armazém de conteúdos.

2.4 Funcionamento

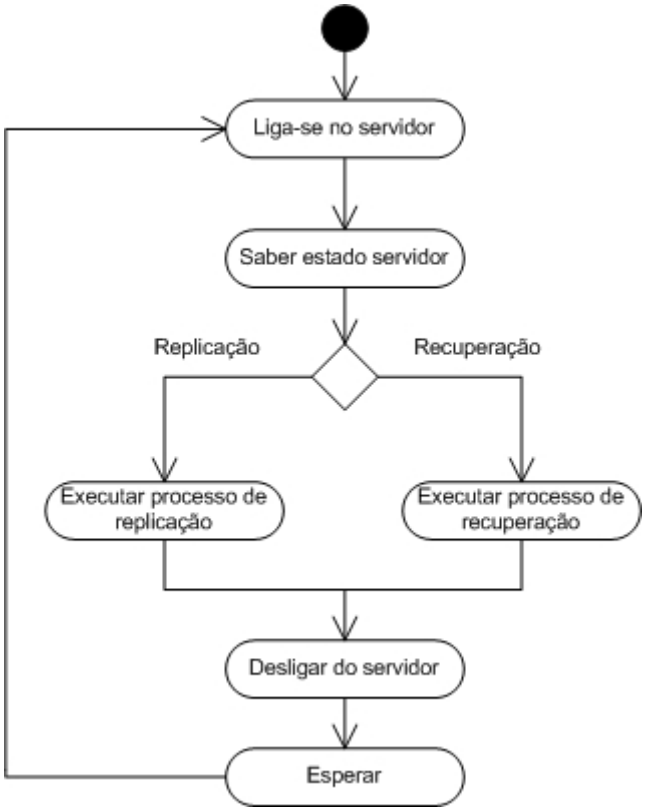


Figura 2.3: Diagrama de actividades do cliente rArc.

O funcionamento do cliente rArc é apresentado na Figura 2.3. O cliente é inicializado sempre que o computador do utilizador arranca. Este começa por ler o identificador e ligar-se ao servidor rArc. Estabelecida a ligação, o cliente vai enviar o identificador para o servidor para se autenticar. De seguida, o cliente vai verificar qual o estado está o servidor. Dependendo do estado, o cliente vai executar o processo de replicação ou de recuperação. Terminado este processo, o cliente desliga-se do servidor e espera um determinado tempo. Passado esse tempo o cliente vai repetir novamente todo o processo. Sempre que o cliente tenha um problema na ligação com o servidor, vai esperar um determinado tempo antes de ligar-se novamente.

2.4.1 Processo de replicação

Se o servidor se encontrar no estado de replicação, o cliente vai verificar se tem espaço livre para armazenar cápsulas. O espaço livre é a diferença entre o espaço oferecido pelo utilizador e o espaço ocupado pelas cápsulas que já armazenadas. Se o espaço livre for superior ao tamanho de uma cápsula, o cliente vai pedir ao servidor para lhe enviar o nome e o tamanho de uma cápsula, como ilustra a Figura 2.4. Depois de receber o nome e o tamanho, o cliente vai criar a cápsula e registar no registo o número de bytes descarregados, que inicialmente é 0. De seguida o cliente vai verificar se o espaço livre ainda é superior ao tamanho de uma cápsula. Se assim for, o cliente vai repetir este processo novamente.

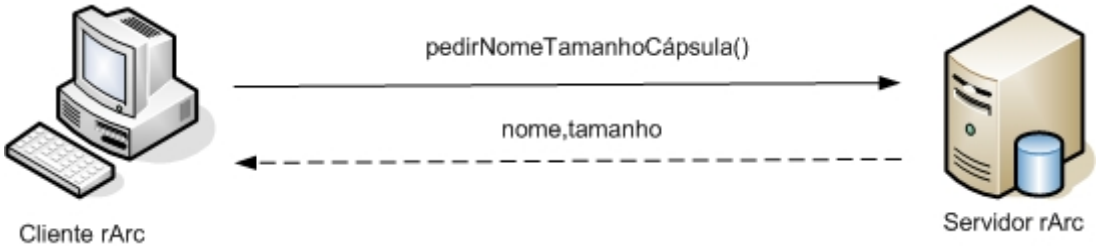


Figura 2.4: Pedido do nome e tamanho da cápsula para replicar.

Quando o espaço livre for inferior ao tamanho de uma cápsula, o cliente vai verificar qual o estado das cápsulas que armazena. Para cada cápsula o cliente vai obter os seguintes dados:

- Nome;
- Tamanho;
- Resumo criptográfico (do inglês, hash);
- Número de bytes descarregados.

Depois de obter os dados, o cliente vai enviar esta informação para o servidor, como ilustra a Figura 2.5.

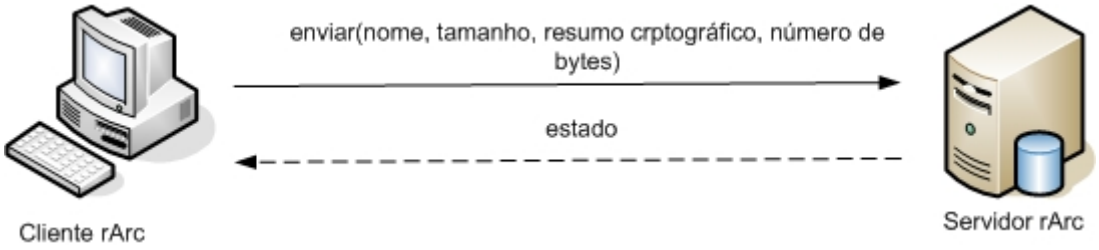


Figura 2.5: Envio dos dados da cápsula para verificação do seu estado.

Em seguida, o servidor vai devolver o estado da cápsula. Uma cápsula pode ter um dos três estados:

- Correcto - quando o nome e o tamanho estão correctos e o número de bytes descarregados é igual ao tamanho da cápsula, e o resumo criptográfico está correcto.
- Incorrecto - quando o nome ou o tamanho estão incorrectos ou o número de bytes descarregados é igual ao tamanho da cápsula, mas o resumo criptográfico está incorrecto.
- Incompleto - quando o nome e o tamanho estão correctos mas o número de bytes descarregados é inferior ao tamanho da cápsula.

Terminada a verificação dos estados das cápsulas, o cliente vai apagar as cápsulas que estão incorrectas. As cápsulas que estão incompletas, o cliente vai descarregar o restantes da cápsulas. Para descarregar a cápsula, o cliente envia para o servidor o nome e o número de bytes que já descarregou, como ilustra a Figura 2.6. Isto permite que, mesmo que um utilizador desligue o cliente, as cápsulas possam ser descarregadas a partir do momento onde estavam a ser descarregadas.

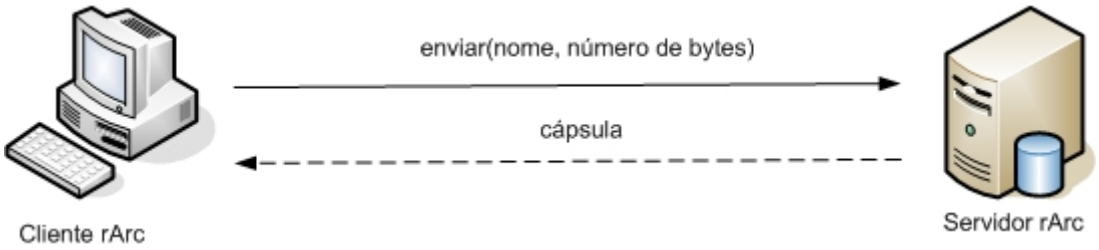
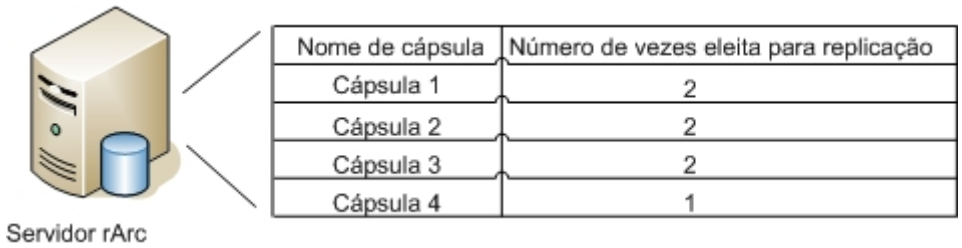


Figura 2.6: Descarga de uma cápsula.

Quando o servidor recebe o pedido de um cliente para que seja enviado o nome e tamanho da cápsula a replicar, o servidor tem de seleccionar uma cápsula. Esta selecção depende do número de vezes que esta foi eleita para replicação.



Nome de cápsula	Número de vezes eleita para replicação
Cápsula 1	2
Cápsula 2	2
Cápsula 3	2
Cápsula 4	1

Figura 2.7: Estrutura da base de dados para replicação.

O servidor associa a cada cápsula o número de vezes que esta foi eleita para replicação. Sempre que um cápsula é eleita, o servidor vai incrementar o número

de vezes de eleição. No caso da Figura 2.7, o servidor iria seleccionar a Cápsula 4. Depois de feita a selecção, o servidor vai incrementar este atributo uma unidade.

A este mecanismo foi dado o nome de replicação uniforme porque tem como objectivo tentar replicar o mesmo número de vezes as cápsulas do armazém de conteúdos.

2.4.2 Processo de recuperação

Se o servidor rArc se encontra no estado de recuperação, o cliente vai verificar o estado das cápsulas que armazena. O processo de verificação realizado durante o processo de recuperação é idêntico ao realizado no processo de replicação. Contudo, quando o estado de uma cápsula é incompleto, o cliente não vai descarregar a cápsula do servidor.

Terminada a verificação do estado das cápsulas, o cliente vai enviar os nomes das cápsulas que armazena e em que o seu estado é correcto, para o servidor, como ilustra a Figura 2.8. Para cada cápsula o servidor informa o cliente se pretende ou não que lhe seja enviada a mesma. Caso pretenda, o cliente vai enviar a cápsula para o servidor. Terminada a transferência, o cliente vai iniciar novamente o envio dos nomes das cápsulas.

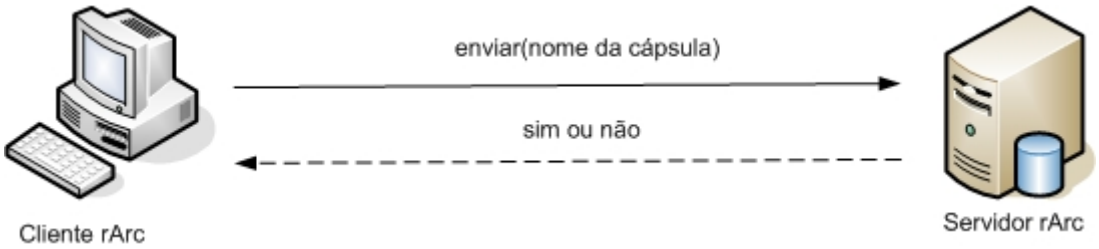


Figura 2.8: Envio do nome da cápsula a recuperar.

Quando o cliente envia os nomes das cápsulas, o servidor vai seleccionar apenas as cápsulas que ainda não foram recuperadas. Sempre que um cliente envia uma cápsula para o servidor esta é considerada temporária. Isto porque, como o cliente pode não enviar a cápsula na sua totalidade de uma única vez, o servidor classifica-a como temporária. Quando o cliente envia os nomes das cápsulas que armazena, o servidor vai seleccionar a cápsula temporária que se encontra mais próxima do fim. Assim, quando o servidor pretende uma cápsula, este vai enviar para o cliente o número de bytes que já foram enviados. O cliente vai enviar os restantes bytes da cápsula. Terminada a transferência, o servidor vai verificar se a cápsula temporária está válida ou inválida. Se estiver inválida, o servidor vai apagar a cápsula. Se estiver válida, o servidor vai registar que foi feita a recuperação da cápsula com sucesso. Terminada a transferência desta cápsula, o cliente vai novamente enviar a

lista, e o servidor vai seleccionar a próxima cápsula que se encontra mais próxima do fim.

A recuperação sequencial tem como objectivo efectuar uma recuperação rápida das cápsulas, recuperando-as de uma forma sequencial.

2.5 Segurança

Os principais mecanismos de segurança implementados tiveram como objectivo garantir a confidencialidade, a integridade e a autenticidade dos conteúdos replicados, garantir a autenticidade dos utilizadores que participam no rArc e, por último, a privacidade na comunicação entre o cliente e o servidor.

Para proteger os ficheiros Arc que estão replicados pelos computadores do utilizadores foi desenvolvido um mecanismo chamado cápsula. A Figura 2.10 apresenta uma cápsula.

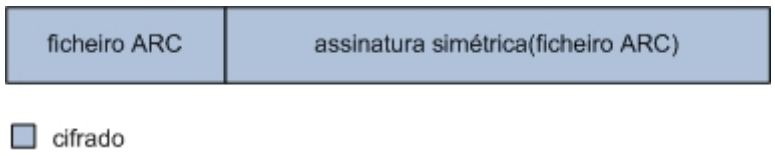


Figura 2.9: Representação de uma cápsula.

Cada ficheiro Arc é cifrado juntamente com a sua assinatura simétrica. O facto de o ficheiro Arc se encontrar cifrado permite garantir a sua confidencialidade. A assinatura simétrica permite garantir a sua autenticidade. Para garantir a integridade dos conteúdos replicados, sempre que os clientes acedem ao servidor, verificam o estado das mesmas. Sempre que o estado da cápsula é incorrecto, o cliente vai apagar a cápsula. Isto permite garantir as cápsulas replicadas pelos computadores estão correctas.

Para um utilizador poder participar no rArc terá de ter uma credencial. A credencial contém informação sobre o utilizador e permite identificá-lo perante o sistema. A Figura 2.10 apresenta a composição de uma credencial.

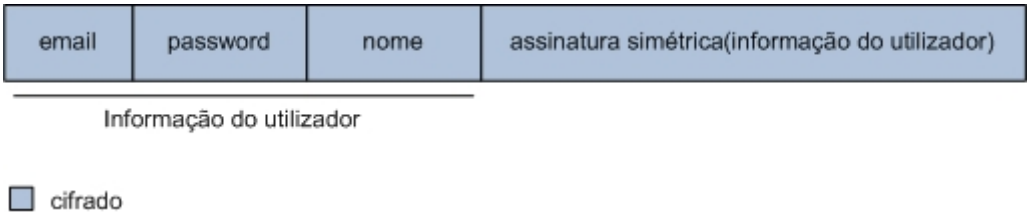


Figura 2.10: Credencial do utilizador.

A credencial contém informação sobre o utilizador e uma assinatura simétrica desta informação. A assinatura simétrica permite garantir a autenticidade da in-

formação assinada. Esta assinatura permite garantir que o utilizador é válido. Tanto a informação do utilizador e a assinatura simétrica estão cifrados, para garantir a sua confidencialidade.

O utilizador necessita de ter uma credencial para poder instalar o cliente no seu computador. Durante o processo de instalação é criado um identificador. O identificador permite que o cliente se autentique no servidor. A Figura 2.11 apresenta a composição de um identificador.

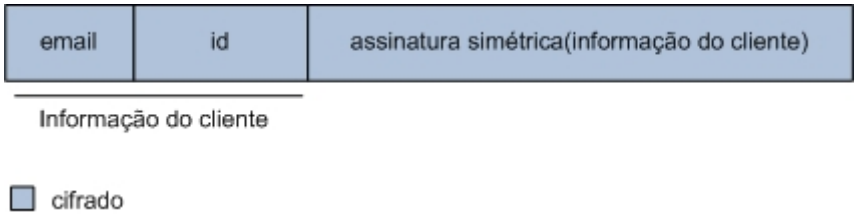


Figura 2.11: Identificador do cliente.

O identificador contém informação do cliente e uma assinatura simétrica desta informação. A assinatura simétrica permite garantir a autenticidade da informação assinada. Esta assinatura permite garantir que o cliente é válido. Tanto a informação do cliente e a assinatura simétrica estão cifrados, para garantir a sua confidencialidade.

A comunicação entre o cliente e o servidor é realizada sobre a Internet. De modo a proteger a comunicação entre estes dois componentes, foi utilizado o mecanismo de segurança *Secure Socket Layer* (SSL). Este mecanismo permite criar um canal seguro entre o servidor e o cliente. O protocolo SSL provê a privacidade e a integridade de dados entre duas aplicações que estejam se comunicando pela Internet. Isto ocorre através da autenticação das partes envolvidas e da criptografia dos dados transmitidos entre as partes. Este protocolo ajuda a prevenir que intermediários entre as duas pontas da comunicação tenham acesso indevido ou falsifiquem os dados sendo transmitidos.

Para cifrar a informação, o rArc usa um algoritmo de cifra simétrica *Advanced Encryption Standard* - AES - com uma chave de 128 bits. Para garantir a autentacidade da informação é utilizado um algoritmo de assinatura simétrica Hash Message Authentication Code - Secure Hash Algorithm - HMAC- SHA1, com uma chave de 160 bits. Por último, para garantir a integridade da informação é utilizado um algoritmo de resumo criptográfico SHA-1.

2.6 Conclusões

Neste Capítulo foi apresentado o desenvolvimento do rArc. Este foi iniciado com o estudo sobre a informação que os arquivos da Web armazenam. De seguida foram

definidos os requisitos que o rArc teria de respeitar corresponder, de forma a corresponder aos arquivos da Web. Depois foi feito o desenho do sistema, onde foi definida a sua arquitectura e seus componentes. De seguida foi explicado como o rArc faz a replicação dos ficheiros Arc e a sua recuperação dos computadores dos utilizadores. Por último, foram apresentados os mecanismos de segurança implementados no sistema. Os mecanismos de segurança implementados permitem que a segurança do rArc não seja assegurada pela obscuridade.

Capítulo 3

Tecnologias

Neste Capítulo são apresentadas a ferramentas e tecnologias usadas no desenvolvimento do rArc. Estas foram utilizadas nas fases de análise, desenho e codificação.

Na Secção 3.1 é apresentada a linguagem de modelação, utilizada na fase de análise e desenho. Na Secção 3.2 é apresentada a linguagem de programação utilizada na fase de implementação. Na Secção 3.3 é apresentada a base de dados utilizada pelo rArc. Por ultimo, na Secção 3.4 é apresentada uma conclusão sobre as tecnologias e ferramentas utilizadas.

3.1 UML

O UML[1] (Unified Modeling Language) é uma linguagem para especificação, visualização, construção e documentação de componentes de software orientados por objectos [20]. Por meio de seus diagramas é possível representar sistemas de software sob diversas perspectivas de visualização. Permite a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema.

Existem actualmente no mercado várias ferramentas que auxiliam o desenvolvimento de software recorrendo ao UML. Neste projecto foi utilizada a ferramenta: ArgoUML[2], para criação dos modelos elaborados na análise e desenho do rArc. A versão utilizada foi a 0.24.

3.2 Java

O Java é um ambiente e ao mesmo tempo uma linguagem de programação de alto nível, produzido pela Sun Microsystems[3]. Diferente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um

[1] http://www.uml.org/
[2] http://argouml.tigris.org/
[3] http://java.sun.com/

código de bytes (bytecode), independente de arquitectura de hardware ou software. Este código de bytes é executado numa máquina virtual Java.

A linguagem Java apresenta as seguintes características:

- Orientada ao objecto (metodologia de programação);
- Portabilidade (permite que o mesmo programa seja executado em vários sistemas operativos);
- Segurança (permite a execução de código remoto de uma forma segura);
- Recursos de rede (possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP).

O rARC foi desenvolvido em Java, utilizando o kit de desenvolvimento jdk 1.6, adicionando os módulos: JDBC e ftp4che.

O JDBC (Java DataBase Connectivity) é uma interface de acesso a bases de dados SQL, permitindo construir aplicações que utilizem bases de dados, mantendo a independência de API's proprietárias. Normalmente, os fabricantes de Sistemas de Gestão de Bases de Dados (SGBD's) disponibilizam *drivers* JDBC que permitem a comunicação de aplicações Java com os seus sistemas. Na fase de implementação, foi utilizado o *driver* JDBC versão 5.1.5, para a comunicação com o MySql.

O módulo ftp4che implementa o protocolo de comunicação FTP. Este módulo contém uma biblioteca que permite controlar a largura de banda da comunicação, entre processos. Na fase de implementação, foi utilizada a versão 0.7.1, para controlar a largura de banda entre cliente e servidor.

O ambiente de desenvolvimento da linguagem Java neste projecto, foi o Eclipse[4].

3.3 Mysql

O MySQL[5] é um SGBD (Sistemas de Gestão de Bases de Dados), que utiliza a linguagem SQL como interface de comunicação. O MySQL é um sistema cliente-servidor que consiste num servidor SQL multitarefa que suporta acessos diferentes, diversos programas clientes e bibliotecas, ferramentas administrativas e diversas interfaces de programação.

O MySql apresenta as seguintes características:

- Compatibilidade (existem drivers e módulos de interface para diversas linguagens de programação);

[4]http://www.eclipse.org/
[5]http://www.mysql.com/

- Portabilidade (suporte para uma vasta gama de plataformas);
- Requisitos de Hardware (pouco exigente quanto a recursos de hardware);
- Desempenho;
- Facilidade de uso;
- Estabilidade.

A versão de Mysql utilizada foi a 5.0.

3.4 Conclusões

Neste Capítulo foram apresentadas as várias tecnologias e ferramentas usadas no desenvolvimento do rArc. As tecnologias e as ferramentas adoptadas são todas disponíveis código aberto, não havendo custo na aquisição de software proprietário. Estas apresentam outras qualidades importantes, tais como qualidade, segurança, independência de fornecedor, possibilidade de adequação a necessidades específicas e estabilidade.

Capítulo 4

Avaliação

Neste Capítulo são apresentados os testes realizados ao rArc. O rArc apresenta uma arquitectura cliente-servidor. Neste tipo de arquitectura o servidor é um ponto de congestão. Os testes realizados tiveram como principal objectivo verificar o desempenho do servidor.

Na Secção 4.1 é apresentada a infra-estrutura de *hardware* do AWP, onde foram realizados os testes. Na Secção 4.2 são apresentados os testes realizados. Por último, na Secção 4.3 é feita uma conclusão sobre os testes realizados.

4.1 Infra-estrutura

A Figura 4.1 apresenta a infra-estrutura de *hardware* do AWP.

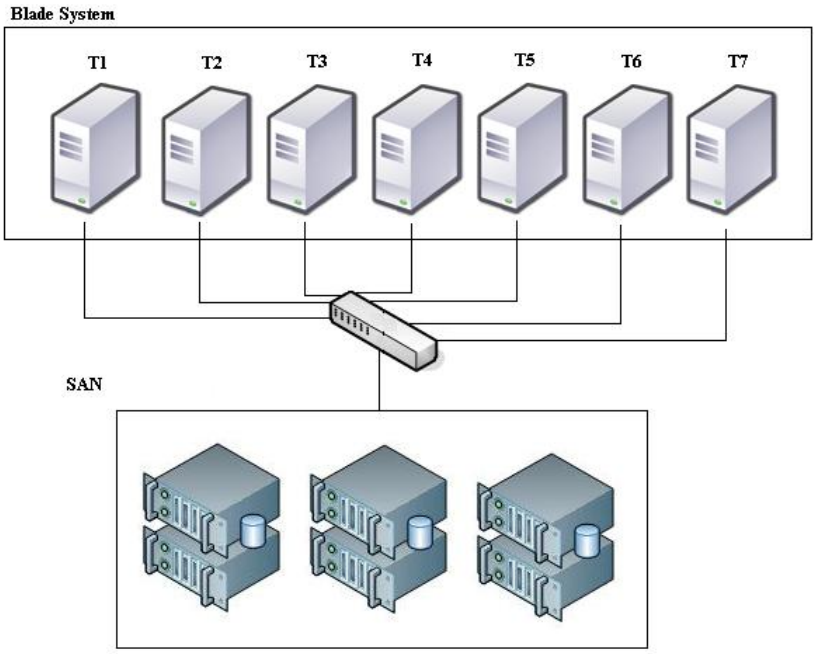


Figura 4.1: Infra-estrutura de *hardware* do AWP.

A infra-estrutura do AWP é constituída por um sistema de Blades e uma SAN. O sistema de Blades é um conjunto de computadores inseridos no mesmo compartimento [16]. Estes sistemas permitem poupar no espaço e consumir menos energia. O AWP tem 7 computadores no compartimento, tendo todos a mesma configuração física:

- Intel(R) Xeon(R) CPU 2 x Quad-core (2.33GHz);
- 8 GBytes RAM;
- Red Hat Enterprise Linux 5.

Uma SAN (Storage Area Network) é uma rede dedicada especificamente para a tarefa de transporte de dados para armazenamento e recuperação [11]. A SAN pode ser vista como uma extensão do conceito que permite que os dispositivos de armazenamento sejam compartilhados entre os computadores e interconectados entre si. Os dispositivos de armazenamento do AWP oferecem aproximadamente 24.5 TBytes de espaço de armazenamento, tendo cada computador o seu próprio espaço para armazenar os dados.

4.2 Testes

Foram realizados os seguintes testes:

1. A replicação das cápsulas existentes no sistema de armazenamento do AWP por vários clientes rArc - teste de replicação.
2. A recuperação das cápsulas armazenadas nos clientes para o sistema de armazenamento do AWP - teste de recuperação.

No primeiro teste foi medido o tempo que demorou a replicar as cápsulas pelos clientes. Foi considerado o tempo total desde o primeiro cliente a ligar-se no servidor, até ao último cliente que verificou que o estado da cápsula era correcto. No segundo teste foi medido o tempo que demorou a recuperar as cápsulas armazenadas nos clientes para o servidor. Foi considerado o tempo total de recuperação desde do momento que o primeiro cliente se ligou no servidor, até à última verificação da validade da última cápsula a ser recuperada.

Num computador foi instalado o servidor, nos restantes foram instalados vários clientes. Antes de instalar os clientes foi necessário determinar quantos poderiam ser instalados em cada computador. Para determinar o número de clientes, foi realizado o primeiro teste várias vezes, aumentando sucessivamente o número de clientes, e verificando quando o desempenho do computador ficaria afectado. Cada cliente tinha a seguinte configuração:

- Espaço para armazenar uma cápsula;
- Tempo de cortesia de 1 segundo;
- 10 MBits de largura de banda.

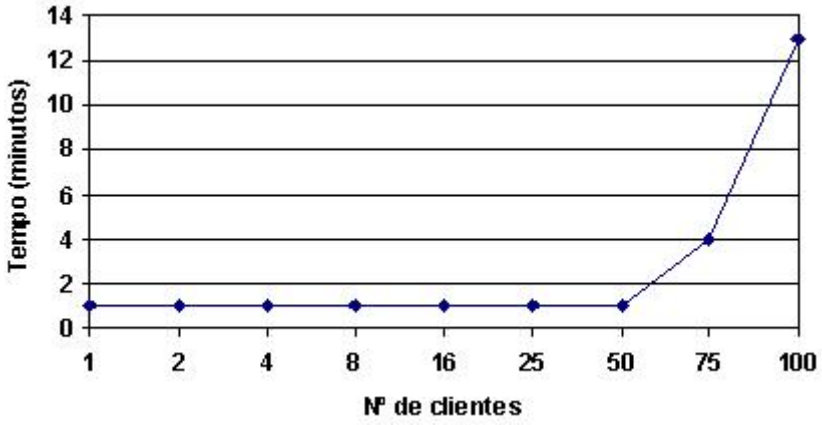


Figura 4.2: Tempo total de replicação com um computador, variando o número de clientes.

A Figura 4.2 apresenta o tempo que demorou a replicar as cápsulas pelo clientes, utilizando apenas um computador. Quando a replicação é realizada por 75 clientes, o tempo que demora a replicar é 4 vezes superior do que quando realizado por 50. Foram também verificados os tempos de acesso ao disco despendidos pelo servidor e pelos clientes. Terminado o teste, ficou definido que em cada computador iriam ser instalados 50 clientes. Assim, como existiam 6 computadores disponíveis, foi possível testar o servidor com o máximo de 300 clientes em simultâneo. Tendo cada cliente espaço de armazenamento para replicar uma cápsula, foi possível replicar aproximadamente 300 GBytes de conteúdos, armazenados no sistema de armazenamento do AWP.

4.2.1 Teste de replicação

A Figura 4.3 apresenta o tempo total que demorou a replicar as cápsulas pelos clientes. Verificou-se que com 150 clientes o tempo total é 4 vezes superior do que quando realizado por 100 clientes. Durante a replicação com 150 verificou-se que o número de clientes ligados no servidor era baixo, e o tráfego na rede também era reduzido. Verificou-se também que os tempos de acesso ao disco era reduzidos, tanto do servidor como dos clientes. Assumimos que poderia ser uma problema na comunicação entre os clientes e o servidor. Foram realizados os mesmos testes mas sem utilizar comunicação segura.

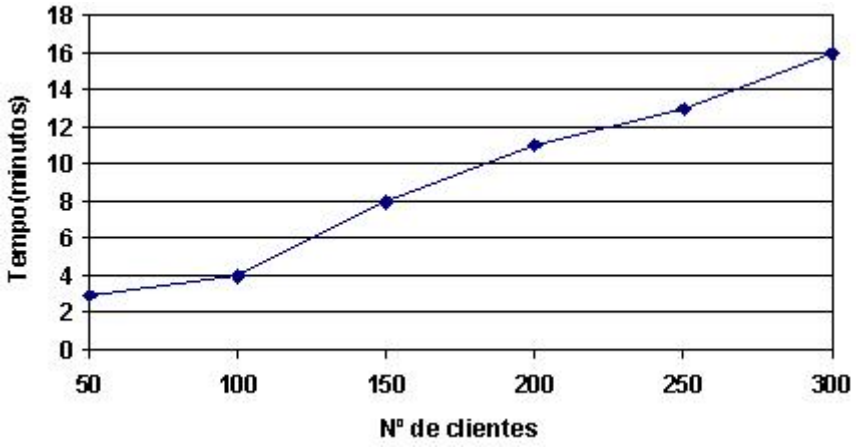


Figura 4.3: Tempo total de replicação variando o número de clientes.

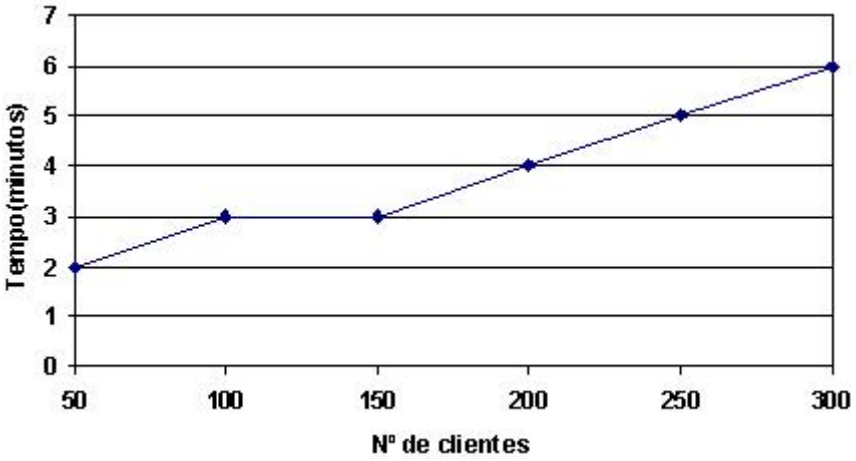


Figura 4.4: Tempo total de replicação variando o número de clientes, sem comunicação segura.

Na Figura 4.4 verifica-se que o tempo total de replicação é menor para o mesmo número de clientes. O tempo total para 300 clientes foi cerca de 6 minutos, em contraste com os 8 minutos que demorou a replicar com 150, com comunicação segura. Durante estes testes verificou-se que existe constantemente tráfego na rede e o número de clientes ligados ao servidor é sempre elevado.

4.2.2 Teste de recuperação

Na Figura 4.5 é apresentado o tempo total que demorou a realizar a recuperação das cápsulas replicadas nos clientes. Verifica-se que, com o aumento do número de clientes, o tempo total aumenta de forma linear.

Foi também realizada a recuperação das cápsulas replicadas pelos clientes mas sem comunicação segura, para verificar qual a sua influência. Na Figura 4.6 é apresentado o tempo total que demorou a realizar a recuperação das cápsulas replicadas

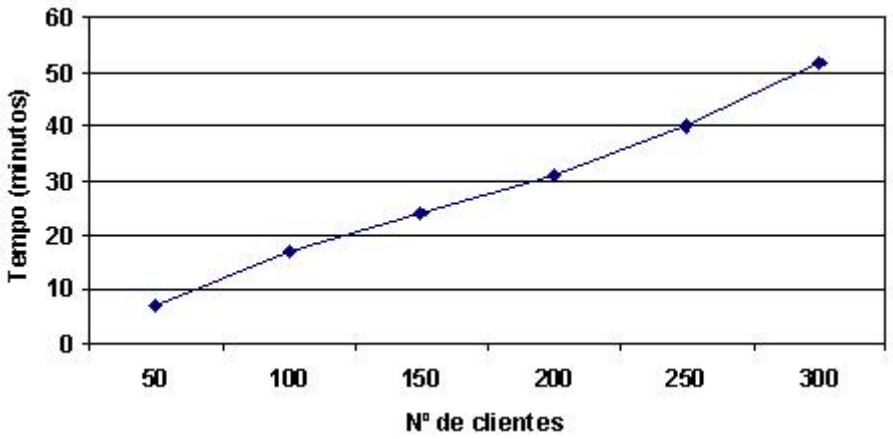


Figura 4.5: Tempo total de recuperação variando o número de clientes.

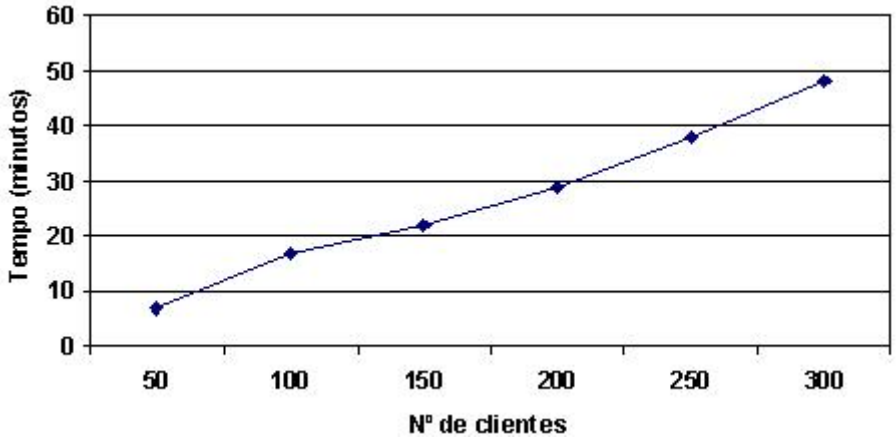


Figura 4.6: Tempo total de recuperação variando o número de clientes, sem comunicação segura.

nos clientes, sem comunicação segura. Verifica-se que com o aumento do número de clientes, o tempo de total de recuperação é menor do que com comunicação segura.

4.3 Conclusões

Neste Capítulo foram apresentados os testes realizados ao rArc. Os testes tinham como objectivo testar o desempenho do servidor. Verificou-se durante a realização dos testes que a comunicação entre os clientes e o servidor afecta os tempos de replicação e recuperação. Sem utilizar comunicação segura entre os clientes e o servidor, os tempos totais de replicação e recuperação foram inferiores do que quando testado com comunicação segura.

Capítulo 5

Trabalho Relacionado

O principal objectivo do rArc é permitir que seja feita uma replicação dos conteúdos dos arquivos da Web por computadores distribuídos pela Internet, para que seja possível a sua recuperação em caso de falha do sistema de armazenamento central. Anteriormente foram desenvolvidos sistemas que permitem a replicação de informação, como por exemplo, os sistemas de partilha de ficheiros ponto-a-ponto [23, 3], as bibliotecas digitais [29, 28] ou os *data grids* [26]. Os sistemas de partilha de ficheiros permitem que os utilizadores da Internet distribuam facilmente informação entre si. Nestes sistemas a informação encontra-se replicada pelos computadores dos utilizadores. As bibliotecas digitais fazem preservação de publicações digitais, tais como publicações científicas de universidades e instituições de investigação. Por último, os sistemas *data grid* têm como objectivo permitir o acesso de dados distribuídos para a computação *grid* [4]. Estes sistemas permitem que os investigadores possam processar dados armazenados nos repositórios de várias instituições. Contudo, embora esses sistemas façam replicação de informação, não correspondem totalmente aos requisitos impostos pelos arquivos da Web para a preservação dos seus conteúdos. Neste Capítulo são analisados sistemas pertencentes a várias classes de sistemas, por forma a avaliar a sua aplicabilidade aos objectivos de um projecto de arquivo da Web.

Na Secção 5.1 é apresentado um sistema de computação distribuída voluntária, o BOINC. Na Secção 5.2 é apresentado um sistema de preservação das publicações digitais, o LOCKSS. Na Secção 5.3 é apresentado um sistema de armazenamento global de dados, o OceanStore. Na Secção 5.4 é apresentado um sistema de partilha de ficheiros, o Napster. Na Secção 5.5 é apresentado um sistema que permite o acesso a dados em sistemas *data grid*. No fim do Capítulo, na Secção 5.6, é feita uma conclusão sobre os sistemas apresentados.

5.1 BOINC

O BOINC[1] (*Berkeley Open Infrastructure for Network Computing*) é uma plataforma que visa facilitar a implementação de sistemas de computação distribuída voluntária[2]. A computação distribuída voluntária consiste na cedência de recursos dos computadores para o processamento de dados científicos quando não estão a ser utilizados. A plataforma do BOINC tem a sua implementação disponível em código aberto e actualmente é desenvolvida por uma equipa da Universidade de Berkeley, na Califórnia. Existem vários projectos que utilizam o BOINC, como por exemplo: o SETI@home[2], o Einstein@home[3] e o Cosmology@home[4]. O SETI@home é uma experiência científica que procurar inteligência extraterrestre. O Einstein@home é um projecto que procura ondas gravitacionais emitidas por buracos negros e por estrelas. O Cosmology@home é um projecto que procura descobrir modelos que descrevem melhor o nosso universo.

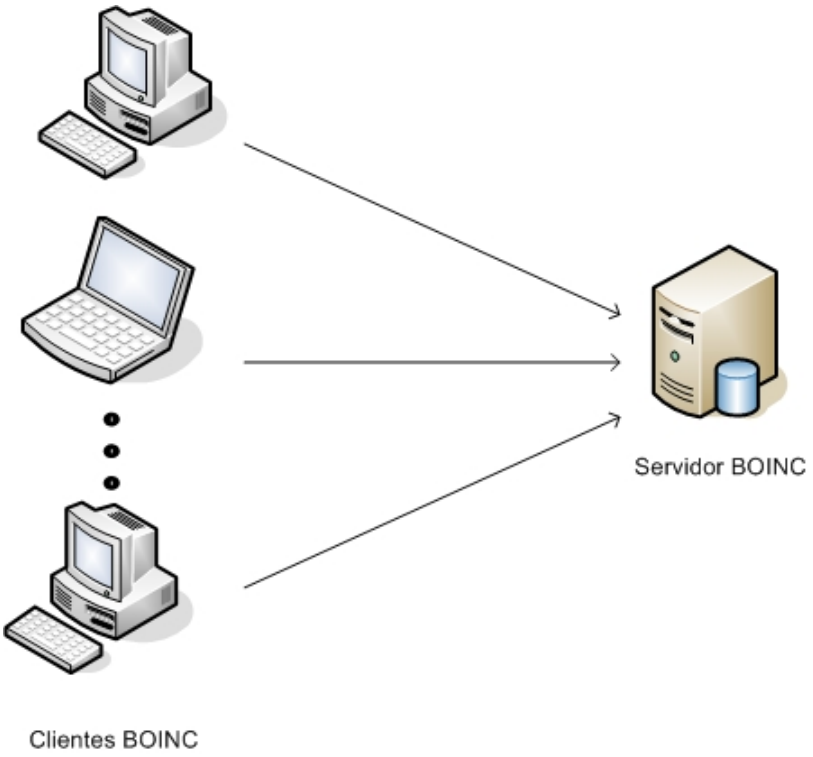


Figura 5.1: Arquitectura do BOINC.

O BOINC apresenta uma arquitectura cliente-servidor, como ilustrado na Figura 5.1. Neste tipo de arquitectura existem dois processos distintos: o cliente e o servidor. O cliente é responsável por estabelecer a conexão com o servidor, enviar men-

[1] http://boinc.berkeley.edu/
[2] http://setiathome.berkeley.edu/
[3] http://einstein.phys.uwm.edu/
[4] http://www.cosmologyathome.org/

sagens e aguardar pelas mensagens de resposta, e o servidor responde aos pedidos dos clientes[25].

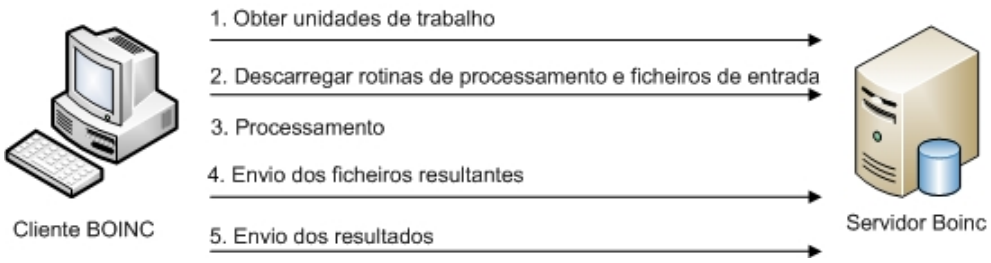


Figura 5.2: Processamento dos dados científicos.

O cliente BOINC repete o processo ciclicamente, ilustrado na Figura 5.2. Para que os dados científicos sejam processados, o cliente liga-se ao servidor para obter unidades de trabalho. Cada unidade de trabalho contém uma rotina de processamento e uma lista de ficheiros de entrada. Os clientes descarregam unidades de trabalho do servidor. Terminado este processo, é dado início ao processamento dos ficheiros de entrada. Depois de terminar o processamento, os ficheiros resultantes são enviados para o servidor. De seguida, são enviados os resultados que contêm a lista de ficheiros resultantes de uma determinada unidade de trabalho e informação acerca do processamento realizado, como por exemplo, o tempo de processamento gasto.

O BOINC foi desenhado para conseguir tolerar o acesso em massa por parte dos clientes. Sempre que um cliente não consegue ligar-se ao servidor, vai aguardar um determinado tempo antes de tentar ligar novamente. Todas as rotinas de processamento utilizadas pelos clientes no processamento dos dados estão assinadas digitalmente, permitindo assim que seja possível verificar a autenticidade das mesmas.

Para um grupo de investigação poder utilizar este *software* terá de criar um projecto BOINC e fornecer um conjunto de rotinas de processamento, que serão utilizadas no processamento dos dados. O grupo terá de instalar um servidor BOINC. Para um utilizador da Internet contribuir para um projecto BOINC terá de visitar o *site* do projecto, efectuar um registo e instalar no seu computador uma aplicação cliente. Para motivar os utilizadores a participarem nos projectos, cada projecto BOINC tem um mecanismo de pontuação que mede quantitativamente as contribuições de cada utilizador para o projecto. A pontuação de cada utilizador está relacionada com o tempo de processamento, espaço de armazenamento e largura de banda que este disponibilizou.

O BOINC é um sistema que pretende utilizar os recursos dos computadores ligados à Internet para efectuar processamento de dados científicos. Para processar os dados são utilizadas rotinas de processamento, que são desenvolvidas pelos

investigadores e têm que ser criadas especificamente para cada sistema operativo. Por sua vez, o rArc pretende utilizar os recursos dos computadores para replicar a informação armazenada num arquivo da Web.

5.2 LOCKSS

O LOCKSS[5] (Lots of Copies Keeps Stuff Safe) tem como objectivo a preservação de publicações digitais [17]. Os destinatários deste projecto são as bibliotecas, pois estas têm assumido a responsabilidade de preservação das publicações em formato digital. O LOCKSS é uma iniciativa da Universidade de Stanford e tem a sua implementação disponível em código aberto.

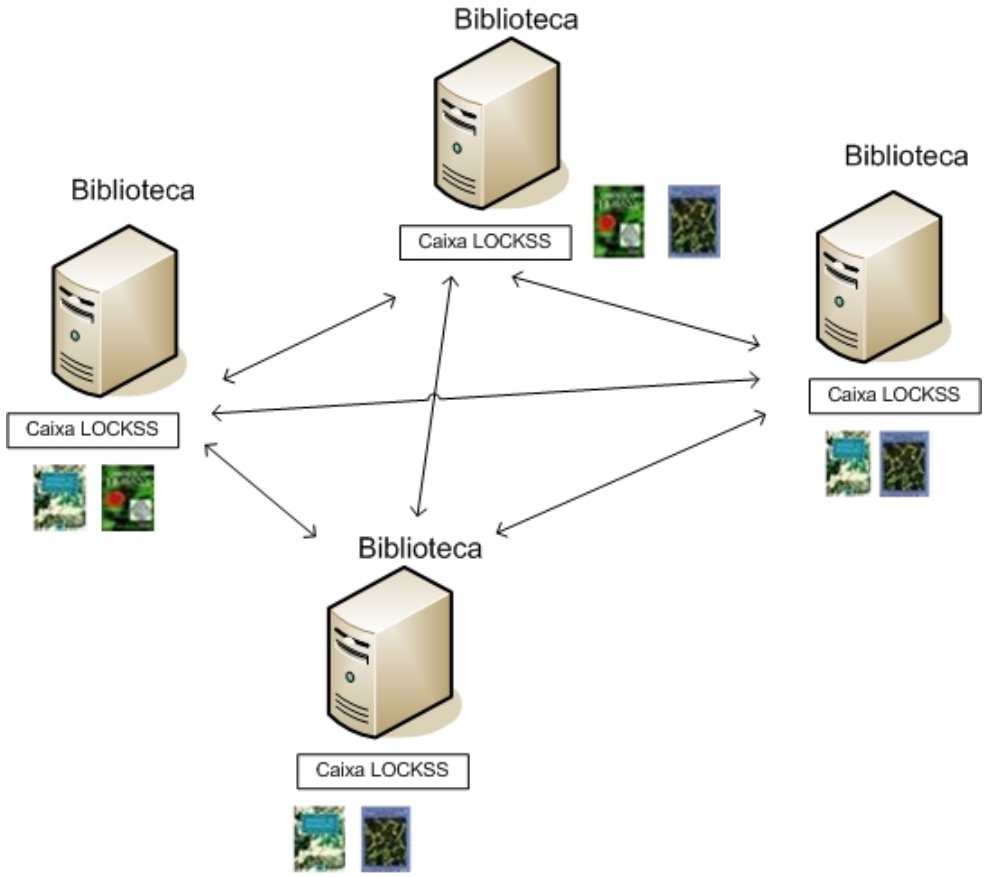


Figura 5.3: Arquitectura do LOCKSS.

Cada biblioteca que participa na iniciativa tem uma caixa LOCKSS. Esta caixa é responsável por recolher, preservar e disponibilizar as publicação digitais da biblioteca. A caixa LOCKSS recolhe as publicações recorrendo a um batedor, semelhante aos utilizados pelos sistemas de arquivo da Web e motores de busca que descarrega as publicações disponibilizadas nos *sites* dos editores. Periodicamente,

[5]http://www.lockss.org/

as caixas que armazenam as mesmas publicações executam processos de votação. Estes processos têm como objectivo comparar os conteúdos das publicações, utilizando resumos criptográficos (do inglês, *hash*). Se uma caixa verificar que alguma publicação está incoerente, vai copiar novamente a publicação a partir das outras caixas LOCKSS. A caixa LOCKSS permite ainda disponibilizar as publicações que armazena quando estas não se encontram disponíveis nos *sites* dos editores, com total transparência para os utilizadores das bibliotecas através da sua integração com os servidores *web proxy*, habitualmente utilizados pelas bibliotecas para controlarem os acessos aos sites dos editores.

O LOCKSS utiliza uma arquitectura *ponto-a-ponto* baseado num modelo descentralizado, como ilustra a Figura 5.3. Neste modelo os processos distribuídos não possuem um papel fixo de cliente ou servidor podendo assumir o papel de cliente ou de servidor dependendo de como a comunicação é iniciada [21].

Para uma biblioteca participar na iniciativa precisa de contactar a equipa do LOCKSS e disponibilizar um computador, no qual será instalada a caixa LOCKSS. A biblioteca tem também que obter autorização por parte dos editores para recolher, preservar e disponibilizar as publicações.

O LOCKSS é um sistema que foi desenhado para funcionar em bibliotecas, e não nos computadores dos utilizadores da Internet. O LOCKSS assume que os computadores das bibliotecas são confiáveis, por isso as publicações encontram-se armazenadas no seu estado original, pois o LOCKSS assume que não existem problemas de confidencialidade. No caso do rArc, os conteúdos replicados pelos computadores do utilizadores não se encontram no seu estado original, encontram-se cifrados, para garantir a confidencialidade da informação.

5.3 OceanStore

O OceanStore[6] é um sistema de armazenamento global de dados, desenhado para permitir o armazenamento persistente à escala mundial [15]. Este sistema foi desenhado para suportar 10^{14} ficheiros, assumindo que existem cerca de 10^{10} utilizadores, tendo cada um cerca de pelo menos 10 000 ficheiros. O OceanStore foi construído de modo a garantir a consistência, disponibilidade, e durabilidade da informação armazenada.

O OceanStore apresenta uma arquitectura *ponto-a-ponto*, como ilustra a Figura 5.4. Este é constituído por servidores individuais e por clientes. Os clientes são as aplicações para inserir e aceder à informação gerida pelo OceanStore. Os servidores são responsáveis por armazenar e preservar a informação inserida pelos clientes. A um conjunto de servidores é dado o nome de domínio de armazenamento. Cada

[6]http://oceanstore.cs.berkeley.edu/

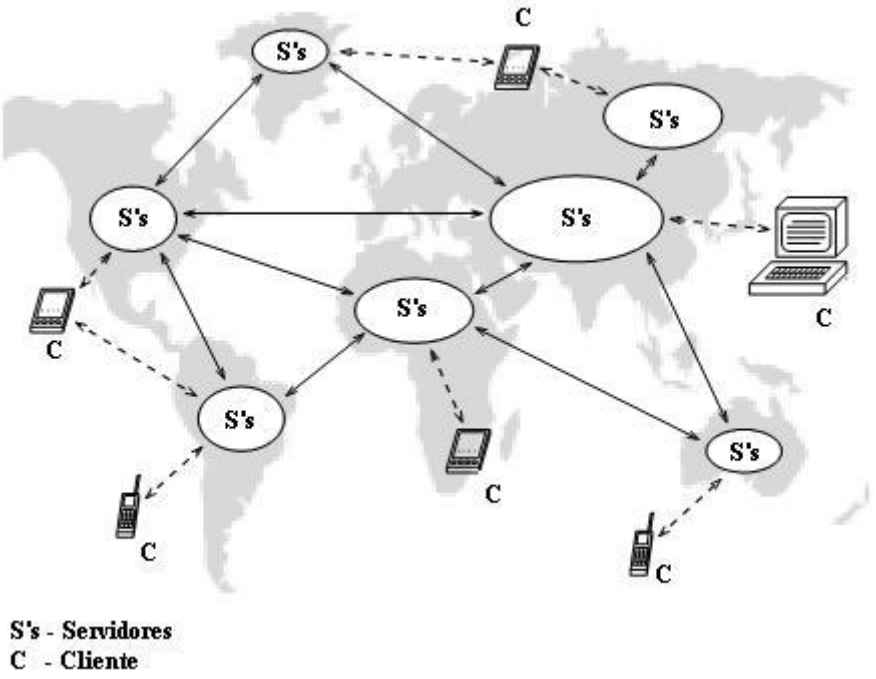


Figura 5.4: Arquitectura do Oceanstore.

domínio é gerido por um prestador de serviços.

O OceanStore assume que a informação armazenada nos servidores pode ficar comprometida quanto à sua confidencialidade. Assim, toda a informação que se encontra armazenada nos servidores encontra-se cifrada. Para garantir a autenticidade da informação armazenada são utilizadas assinaturas digitais da informação. O OceanStore assume também que a informação é nómada, isto é, a informação migra livremente pelos domínios de armazenamento para servidores geograficamente próximos de onde os utilizadores acedem mais frequentemente. A este mecanismo foi dado o nome de *promiscuous caching*. Para garantir a preservação da informação armazenada foi desenvolvido um mecanismo designado por *deep archival storage*, que replica cada arquivo em fragmentos pelo sistema. Este mecanismo assegura que um arquivo possa ser reconstituído mesmo que ocorram catástrofes localizadas. Para garantir a consistência da informação replicada, os servidores comparam entre si a informação que cada um armazena, para detectar problemas de inconsistência. O OceanStore permite que sejam realizadas actualizações sobre a informação armazenada. Sempre que é realizada uma actualização é criada uma nova versão do arquivo. Todas as versões de um arquivo são preservadas.

Qualquer utilizador pode participar no OceanStore, disponibilizando espaço de armazenamento do seu computador pessoal, em troca de uma compensação económica. Os utilizadores precisam de se inscrever num prestador de serviços do OceanStore. Os prestadores de serviços compram e vendem espaço de armazenamento entre si,

com transparência para os utilizadores.

O OceanStore pode ser considerado um sistema de armazenamento distribuído à escala mundial. Este permite que os seus utilizadores possam aceder e alterar a informação armazenada. No caso dos arquivos da Web o principal objectivo é a replicação da informação. Contudo, um arquivo da Web poderia utilizar o OceanStore para replicar os seus conteúdos. Actualmente existe um protótipo do OceanStore, o Pond [22]. Este protótipo ainda não foi testado à escala global e ainda não existe uma versão estável [7].

5.4 Napster

O Napster é um sistema de partilha de ficheiros que permite que os utilizadores da Internet possam partilhar entre si os seus ficheiros de música [21]. Este sistema foi desenvolvido por Shawn Fanning. O Napster foi encerrado em 2002 por problemas legais, devido aos direitos de autores das músicas. Actualmente, o Napster[7] encontra-se novamente activo mas não nos mesmos termos. Agora o Napster vende música de forma legal. Todavia, apenas existe informação sobre o antigo Napster.

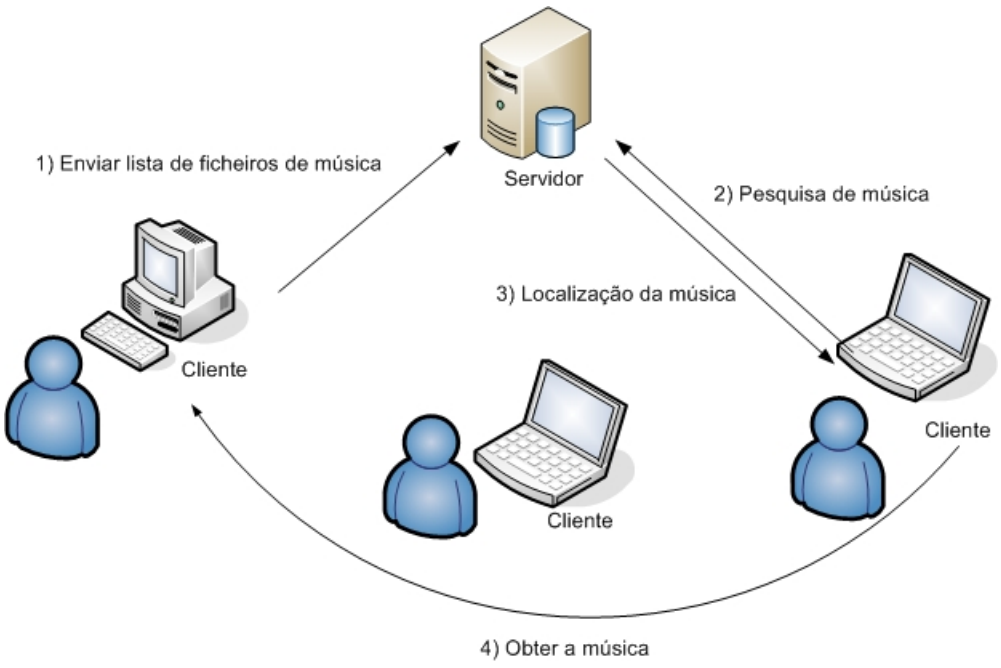


Figura 5.5: Obtenção de um ficheiro de música.

O Napster apresenta uma arquitectura ponto-a-ponto baseada num modelo centralizado. Este tipo de modelo apresenta um nó central (servidor) e vários nós (clientes). O cliente é responsável por enviar para o servidor, sempre que acede ao

[7]http://www.napster.com/

sistema, a lista de ficheiros que armazena. O servidor é responsável por deter todo o estado do sistema, os ficheiros existentes e os clientes que os armazenam.

A Figura 5.5 apresenta o processo completo de como um utilizador obtém uma música. O cliente envia o nome da música para o servidor. Este devolve a lista de clientes que a armazenam. Após receber a lista, o cliente vai obter a música pretendida ligando-se directamente a um cliente que a armazena.

Devido à sua arquitectura o Napster tem como principal desvantagem o facto de se o servidor estiver indisponível, todo o sistema fica inacessível.

O Napster foi criado para permitir que os utilizadores da Internet possam partilhar os seus ficheiros de música. Para um utilizador poder partilhar as suas músicas tinha de instalar no seu computador uma aplicação cliente. Este sistema foi criado para permitir a partilha de ficheiros e não a sua replicação, ao contrário do rArc. A informação que o rArc replica pelos computadores não é do interesse dos utilizadores, ao contrário do que acontece com o Napster.

5.5 SRB

Os sistemas *data grid* têm como objectivo permitir o acesso à informação armazenada em repositórios distribuídos em localizações geograficamente distantes. Estes sistemas são utilizados em projectos científicos, para permitir que os investigadores possam ter acesso à informação para ser utilizada para a computação *grid*. Para permitir um acesso uniforme à informação foi criado o SRB[8] (Storage Resource Broker), um sistema de *middleware* que permite o acesso uniforme e transparente à informação armazenada em sistemas *data grid* [5]. Este sistema foi desenvolvido pelo San Diego Supercomputer Center[9].

O SRB apresenta uma arquitectura cliente-servidor, como ilustra a Figura 5.6. Para um utilizador aceder a uma dada informação terá de utilizar uma aplicação cliente. Este envia o pedido do utilizador para o servidor SRB, que está instalado na instituição do utilizador e é responsável pelo repositório da instituição. Após receber o pedido, o servidor vai reenviar o pedido para o servidor de metadados. Este servidor contém toda a informação sobre os dados armazenados nos repositórios, como por exemplo: a localização física dos dados ou as permissões de acesso. Toda esta informação encontra-se catalogada em metadados, o que permite que os utilizadores possam efectuar pesquisas sobre a informação. O servidor de metadados vai devolver ao servidor SRB a localização física da informação. O servidor SRB vai efectuar o pedido ao servidor SRB que armazena a informação. Este vai devolver a informação ao cliente.

[8] http://www.sdsc.edu/srb/
[9] http://www.sdsc.edu/

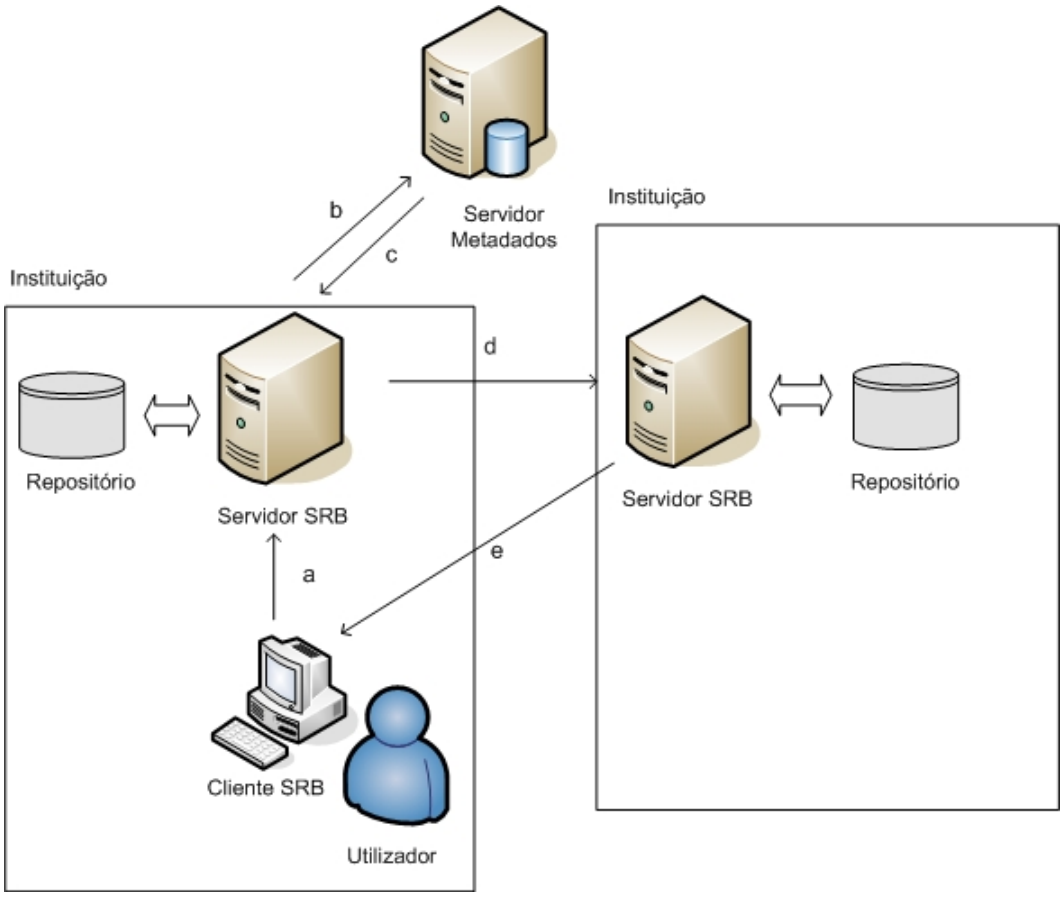


Figura 5.6: Arquitectura do SRB.

O SRB permite que os utilizadores possam aceder à informação sem saberem onde esta se encontra armazenada. O SRB permite que os repositórios armazenem a informação de forma heterogénea. Esta abstracção é importante para os utilizadores, pois estes não necessitam de saber como funciona cada sistema de armazenamento porque cada instituição poderá ter o seu próprio sistema de armazenamento. O acesso à informação armazenada nos repositórios é controlada, apenas os utilizadores com permissões podem aceder à informação.

O SRB tem como principal objectivo permitir o acesso à informação armazenada de uma forma uniforme. Este sistema não tem como objectivo a replicação da informação armazenada nos repositórios, ao contrário do rArc. O SRB tem a sua implementação disponível para instituições académicas e agências governamentais.

5.6 Conclusões

Neste Capítulo foram analisados sistemas pertencentes a várias classes de sistemas, por forma a avaliar a sua aplicabilidade aos objectivos de um projecto de arquivo da Web. Os sistemas de partilha de ficheiros são utilizados para a distribuição de

informação. As bibliotecas digitais permitem a replicação da informação em ambientes confiáveis. Os sistemas data *grid* permitem o acesso uniforme à informação armazenada em repositórios distribuídos. O estudo permitiu verificar que não existe nenhuma solução pronta a ser utilizada pelos arquivos da Web para a replicação da informação que estes armazenam, porque os sistemas não correspondiam totalmente aos requisitos impostos pelos arquivos. Contudo, o estudo destes sistema permitiu reutilizar alguns mecanismos utilizados por estes, como por exemplo, o mecanismo de verificação da integridade das publicações do LOCKSS. Assim, o rArc apresenta-se como uma solução pronta a usar pelos arquivos da Web, para a replicação dos conteúdos armazenados nos seus sistemas de armazenamento.

Capítulo 6

Conclusão e Trabalho Futuro

O trabalho apresentado nesta tese teve como objectivo a criação de um sistema de replicação distribuído que permita contribuir para a preservação dos conteúdos armazenados por arquivos da Web. Este sistema foi denominado rArc. Este permite replicar os conteúdos armazenados no sistema de armazenamento de um arquivo da Web, por computadores ligados à Internet. Caso ocorra uma falha no sistema de armazenamento e ocorra a perda dos conteúdos nele armazenados, o sistema vai recuperar os conteúdos replicados por computadores, para que seja possível restaurar o sistema de armazenamento com os conteúdos perdidos. O rArc poderá não recuperar todos os conteúdos replicados, contudo é importante recuperar parte da informação para que esta não se perca toda.

Este trabalho iniciou-se com o estudo de vários sistemas que realizam replicação de informação. Este estudo permitiu verificar que nenhum sistema poderia ser utilizado no âmbito de um arquivo da Web. Terminado o estudo, iniciou-se o desenvolvimento do rArc, no qual foi realizada a análise, o desenho e a codificação do sistema. De seguida foram realizados testes de desempenho do rArc. Estes testes permitiram identificar alguns problemas com a arquitectura e segurança do rArc. Todavia, corrigindo os problemas identificados e tornando o sistema mais robusto, este poderá contribuir para a preservação dos conteúdos dos arquivos da Web.

O trabalho apresentado nesta tese integra-se na cadeira de Projecto em Engenharia Informática do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa e foi realizado no projecto Arquivo da Web Portuguesa, em curso na Fundação para a Computação Cientifica Nacional.

6.1 Trabalho Futuro

O rArc apresenta uma arquitectura do modelo cliente-servidor. Neste tipo de arquitectura o desempenho do sistema poderá ser afectado se o número de clientes ultrapassar o limite de processamento do servidor. O servidor poderá não ter ca-

pacidade de atender a todos os pedidos, ficando o desempenho do sistema em risco. Actualmente, o servidor rArc é multitarefa pois permite atender a vários clientes em simultâneo. Para garantir a sincronização das operações realizadas o sincronismo é realizado ao nível do servidor e não da base de dados. Este modo de sincronização tem como desvantagem o facto de se existir outro processo servidor, poderá haver problemas de sincronização da operações. Para permitir que possam existir vários processos servidores instalados em máquinas diferentes, a sincronização poderia ser realizada ao nível da base de dados. Isto permitiria que se consegui-se atender a mais clientes em simultâneo, efectuando balanceamento de carga entre as várias máquinas, aumentado o desempenho do sistema. Contudo, a base de dados poderia ser um ponto de congestão. Seria necessário verificar qual o desempenho da base de dados com o aumento de carga.

O rArc foi desenhado e desenvolvido assumindo que a perda dos conteúdos armazenados no sistema de armazenamento do arquivo é total. Este tipo de cenário ocorre quando o sistema de armazenamento sofre um dano irreparável. Normalmente existe apenas a perda parcial dos dados. Actualmente, o sistema não suporta a recuperação parcial dos conteúdos. A implementação desta funcionalidade permitiria que o rArc estivesse sempre em replicação, e só em caso de perda de alguns conteúdos, o sistema iria recuperá-los, mas ao mesmo tempo continuaria a replicar os outros conteúdos armazenados.

Durante o processo de recuperação, o servidor recebe dos clientes as cápsulas que estes armazenam. Sempre que um cliente termina de enviar uma cápsula, o servidor vai verificar a autenticidade do ficheiro Arc. Se este for autentico, o servidor assume que o ficheiro é válido. Contudo, se utilizador malicioso descobrir a chave que permite efectuar a assinatura simétrica do ficheiro, este poderia efectuar ataques de segurança ao servidor durante a recuperação, alterando o ficheiro para conter, por exemplo, um vírus. Para aumentar a garantia da autenticidade de uma cápsula recuperada, o rArc poderia definir um número mínimo de clientes que enviassem a mesma cápsula para o servidor e verificar se existem consenso nas várias assinaturas simétricas para o mesmo ficheiro.

A comunicação entre os clientes e o servidor é realizada de uma forma segura. Esta forma de comunicação permite assegurar a privacidade na comunicação. Contudo, verificou-se que é apenas necessário garantir a privacidade da comunicação quando o cliente se autentica no servidor. Se um utilizador malicioso capturar-se um identificador, este poderia utiliza-lo para participar no rArc, mesmo sem ter uma credencial. Depois da autenticação, a informação trocada entre o cliente e o servidor não é considerada sensível, não sendo assim necessário que haja privacidade na comunicação.

O rArc assume que a informação guarda na base de dados do servidor sobrevive,

mesmo que ocorra uma falha no sistema de armazenamento do arquivo da Web. Quando o servidor inicia o processo de recuperação, este tem acesso à informação da base de dados. O servidor utiliza a informação para verificar a integridade da cápsulas armazenadas nos clientes, antes de estes enviar as cápsulas para o servidor. Contudo, mesmo que o servidor não tivesse acesso à informação da base de dados, era possível efectuar a recuperação das cápsulas. Todavia, seria necessário alterar alguns mecanismos para permitir este novo modo de funcionamento.

Bibliografia

[1] Serge Abiteboul, Gregory Cobena, Julien Masanes, and Gerald Sedrati. A first experience in archiving the French web. In *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 1–15, London, UK, 2002. Springer-Verlag.

[2] David P. Anderson. Boinc: A system for public-resource computing and storage. In *GRID '04: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.

[3] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.

[4] Mark Baker, Rajkumar Buyya, and Domenico Laforenza. Grids and grid technologies for wide-area distributed computing. *Softw. Pract. Exper.*, 32(15):1437–1466, 2002.

[5] Chaitanya Baru, Reagan Moore, Arcot Rajasekar, and Michael Wan. The sdsc storage resource broker. In *CASCON '98: Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research*, page 5. IBM Press, 1998.

[6] Mike Burner and Brewster Kahle. *Arc File Format*, September 1996.

[7] Simon Chong, Paul A. Watters, and Michael Hitchens. Automated physical storage provision using a peer-to-peer distributed file system. In *ICDEW '05: Proceedings of the 21st International Conference on Data Engineering Workshops*, page 1214, Washington, DC, USA, 2005. IEEE Computer Society.

[8] Michael Day. Collecting and preserving the World Wide Web. `http://www.jisc.ac.uk/uploaded_documents/archiving_feasibility.pdf`, 2003.

[9] Richard Entlich. Blog today, gone tomorrow ? Preservation of Weblogs. *RLG Diginews*, 8(4), August 2004.

[10] Hurricanes katrina and rita web archive. `http://websearch.archive.org/katrina/`.

[11] J. S. Glider, C. F. Fuente, and W. J. Scales. The software architecture of a san storage control system. *IBM Syst. J.*, 42(2):232–249, 2003.

[12] Hurricanes katrina and rita web archive. `http://websearch.archive.org/katrina/`.

[13] Brewster Kahle. The Internet Archive. *RLG Diginews*, 6(3), June 2002.

[14] Mads Alhof Kristiansen. Digital preservation using the warc file format. Technical report, Department of Computer Science, University of Copenhagen (DIKU), Universitetsparken 1, DK-2100 Copenhagen OE, 2006.

[15] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Chris Wells, and Ben Zhao. Oceanstore: an architecture for global-scale persistent storage. *SIGOPS Oper. Syst. Rev.*, 34(5):190–201, 2000.

[16] Kevin Leigh, Parthasarathy Ranganathan, and Jaspal Subhlok. General-purpose blade infrastructure for configurable system architectures. *Distrib. Parallel Databases*, 21(2-3):115–144, 2007.

[17] Petros Maniatis, Mema Roussopoulos, T. J. Giuli, David S. H. Rosenthal, and Mary Baker. The lockss peer-to-peer digital preservation system. *ACM Trans. Comput. Syst.*, 23(1):2–50, 2005.

[18] Gordon Mohr, Michele Kimpton, Micheal Stack, and Igor Ranitovic. Introduction to heritrix, an archival quality web crawler. In *4th International Web Archiving Workshop (IWAW04)*, Bath, UK, September 2004.

[19] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.

[20] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*, volume 6ª edição. McGraw-Hill, 2005.

[21] Brian D. Goodman Ramesh Subramanian. *Peer to Peer Computing: The Evolution of a Disruptive Technology*. IGI Global, 2005.

[22] Sean Rhea, Patrick Eaton, Dennis Geels, Hakim Weatherspoon, Ben Zhao, and John Kubiatowicz. Awarded best student paper! - pond: The oceanstore prototype. In *FAST '03: Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, pages 1–14, Berkeley, CA, USA, 2003. USENIX Association.

[23] John Risson and Tim Moors. Survey of research towards robust peer-to-peer networks: search methods. *Comput. Netw.*, 50(17):3485–3521, 2006.

[24] Kristinn Sigurosson. Incremental crawling with heritrix. In *5th International Web Archiving Workshop (IWAW05)*, Viena, Austria, September 2005.

[25] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems: Principles and Paradigms, 2 Edition.* Prentice Hall, 2007.

[26] Srikumar Venugopal, Rajkumar Buyya, and Kotagiri Ramamohanarao. A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Comput. Surv.*, 38(1):3, 2006.

[27] Paulo Verissimo and Luis Rodrigues. *Distributed Systems for System Architects.* Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[28] Andrew Waugh, Ross Wilkinson, Brendan Hills, and Jon Dell'oro. Preserving digital information forever. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 175–184, New York, NY, USA, 2000. ACM.

[29] Hong Iris Xie. Users' evaluation of digital libraries (dls): Their uses, their criteria, and their assessment. *Inf. Process. Manage.*, 44(3):1346–1373, 2008.