Backup distribuído: uma implementação funcional

Ciro Esteves Lima Sobral
NCE - UFRJ
cirosobral@ufrj.br

Álvaro Vinícius de Souza Coêlho
DCET - UESC
degas@uesc.br

***Resumo*—O processo de cópia de segurança (backup) responde pela salvaguarda de patrimônios que apresentam valor financeiro e são importantes para a continuidade dos trabalhos das organizações que os possuem: os dados. Enquanto isso, essas mesmas organizações possuem estações de trabalho, que em geral apresentam espaço ocioso em disco. O presente trabalho descreve um sistema de backup de dados (cópia de segurança), que utiliza o espaço em disco disponível de um conjunto de máquinas para armazenar os dados de backup de outras. Este processo permite criar redundância dos dados dos servidores, aumentando a segurança dos mesmos, aproveitando um recurso já existente. Mostramos que o sistema proposto realiza com sucesso as tarefas de backup e restauração dos dados, apresentando um bom desempenho nas duas tarefas.**

I. Introdução

O grande volume de dados e informações produzidas pela sociedade moderna demanda estruturas de reprodução e armazenamento cada vez mais potentes, tornando-as cada vez mais abundantes nas organizações. Com isso, a capacidade de armazenamento dos discos rígidos dos computadores cresce mais do que os usuários comuns necessitam para o seu trabalho diário. Assim, em empresas que possuem estações de trabalho, é comum a existência de espaço de armazenamento de informações ocioso nos discos dessas máquinas.

Por outro lado, a cada dia também cresce a necessidade de se fazer o armazenamento e proteção das informações nas empresas para prevenir os diversos riscos inerentes à perda de dados, muitos dos quais tem grande impacto financeiro. Desse modo, se gasta cada vez mais com estruturas e estratégias de backup (cópia de segurança) para minimizar os riscos, com a contrapartida do aumento de custos. As alternativas de backup existentes, mesmo as mais complexas e onerosas, são passíveis de falha, justificando a necessidade de estudos na busca de formas mais eficazes e de baixo custo para proteger as informações empresariais e mesmo pessoais de grande valor para os usuários.

Neste trabalho apresentamos uma implementação funcional de um um sistema de backup distribuído, que torna útil o espaço ocioso nos discos das máquinas de empresas e organizações, utilizando-os para o armazenamento das cópias de segurança das suas informações relevantes, provendo confiabilidade e segurança sem custos adicionais.

II. Descrição do Problema

No mundo corporativo, a realização de backups é uma prática cada vez mais necessária no cotidiano das empresas. Como a perda de dados tem potencial de causar grandes danos financeiros e operacionais, em algumas organizações a realização de backups cumpre importante papel no gerenciamento da continuidade de negócios [1]. Este é um problema complexo de se equacionar. Por um lado, a ausência de um sistema seguro de backup oferece o risco da perda de informações, com os custos daí decorrentes: custo da recuperação de arquivos, lucros cessantes, custos com possíveis ações judiciais e outros ainda mais difíceis de serem contabilizados, como dano à imagem da corporação [1]; por outro lado, as estratégias de backup, em geral, oneram as empresas com a compra e manutenção de equipamentos, contratação de pessoal e treinamento para proteger seus dados. A escolha do mecanismo de backup utilizado deve buscar minimizar os riscos, mantendo os custos em níveis aceitáveis. Isto é verdade mesmo para usuários domésticos, que também estão sujeitos à necessidade de garantir a persistência dos dados contidos nos computadores, como e-mails, contatos, e arquivos de valor sentimental (vídeos, fotos e músicas). Estudos mostram que 6% dos usuários domésticos perdem alguma informação dos computadores por ano [2].

O processo de backup de dados pode ser efetuado através da simples cópia destes em mídias como CDs, DVDs, pendrives, discos rígidos, fita magnética, etc [2]. De acordo com estudo da Cibecs [1], no mundo corporativo a cópia dos arquivos para um servidor de dados ou a cópia para um disco rígido externo representam 47% dos sistemas de backup utilizados. Este processo gera custos com equipamentos de uso específico para manutenção do backup. Além disso, os equipamentos de backup estão potencialmente à mercê de vários incidentes que causam também a perda dos dados originais, como roubo, incêndios, etc. Isto faz com que as políticas de backup determinem que a mídia em que foi realizada a cópia seja transportada para outro local, a fim de garantir a segurança do equipamento e, consequentemente, dos dados. Por esse motivo, o processo de backup implica em custos cada vez mais significativos para as organizações.

Ao mesmo tempo, enquanto a capacidade instalada em redes de computadores cresce, a utilização da capacidade de armazenamento dos discos rígidos dos computadores é tipicamente baixa. Estudos apontam que se utiliza aproximadamente 50% do espaço disponível nos discos [3]. Considerando que as empresas normalmente trabalham com uma grande quantidade de máquinas, pode-se estimar que há um contingente considerável de recursos de armazenamento ociosos. Estes recursos podem ser aproveitados a fim de suportar o processo de backup, tornando-o mais simples e barato, além de agregar valor aos dispositivos

de armazenamento já disponíveis. Neste sentido, uma estratégia viável e economicamente interessante é fazer o backup usando a transmissão dos dados através da rede de computadores, armazenando-os em máquinas distintas, geograficamente separadas.

III. O Backup Distribuído

No processo de backup que propomos neste trabalho possui três atores principais: o *cliente*, que solicita a operação de backup, o *servidor de backup*, que oferece este serviço e o *servidor de metadados*, que provê um indexador de forma que os dados de cada cliente possam ser localizados satisfatoriamente.

A. Funcionamento

Na operação deste sistema, o cliente se conecta ao servidor de metadados e solicita deste a lista de servidores de backup ativos. De posse desta lista, o cliente prepara seus arquivos para serem copiados, empacotando-os junto com suas estruturas de diretórios em um único arquivo e compactando-o a fim de reduzir o espaço necessário para armazenamento e diminuir o tráfego na rede. Este processo irá gerar um único arquivo contendo todos os dados a serem persistidos. Em seguida este arquivo é criptografado, por questão de segurança já que será alojado em máquinas remotas, e particionado em múltiplos arquivos menores, de forma a permitir que cada servidor de backup aloje parte dos dados de um mesmo cliente. Desta forma, pode-se implementar uma política de redundância, em que cada parte dos dados esteja alojada em mais de um servidor. O cliente então escolhe aqueles servidores com espaço em disco suficiente para receber seus arquivos, e envia os arquivos. Após a confirmação da chegada dos dados nos servidores de backup, o cliente informa ao servidor de metadados quais os arquivos foram armazenados em quais servidores de dados.

Caso seja necessária a restauração dos dados, o cliente solicita o início do processo junto ao servidor de metadados, que o atende enviando uma lista com os nomes dos arquivos que compõem seu backup, além dos respectivos servidores em que estes se encontram. De posse desta informação, o cliente se conecta aos servidores, requisitando destes os arquivos que lhe interessam. Após receber todos os fragmentos do backup, procede-se a um processo inverso daquele realizado antes do envio: os arquivos são concatenados, formando um único arquivo que será descriptografado, descompactado e finalmente desempacotado, restaurando a estrutura de pastas e arquivos original.

B. Detalhes da Arquitetura

O servidor de metadados é um nó central no sistema, muitas vezes referido na literatura como *tracker* [4]. É o responsável por armazenar as "informações dos dados", que permitem definir que parte dos dados de cada cliente está com cada servidor. Estas informações são chamadas de metadados. Neste trabalho, estas informações são o nome do arquivo, o *hostname* do cliente e o *hostname* do servidor, havendo uma entrada desta natureza para cada um dos arquivos persistidos. Cabe também ao servidor de metadados indicar ao cliente quais os *hostname* dos servidores de dados ativos, bem como o espaço em disco livre em cada um deles. Para o pleno funcionamento do sistema é necessário que o servidor de metadados esteja ativo, acessível pela rede tanto pelas máquinas que abrigarão os servidores de dados quanto pelas que utilizaram o sistema como cliente.

O servidor de backup é o agente que irá de fato efetuar a operação de armazenamento dos dados a serem persistidos. Na concepção deste sistema, idealmente deve existir um conjunto de servidores de dados, de forma a se reduzir a carga de trabalho solicitada a cada um deles, bem como para se implementar políticas de redundância de dados. Quando iniciado o serviço, estes servidores se conectam ao servidor de metadados, informando seu endereço de rede e o espaço em disco disponível. A partir deste momento, o servidor de metadados inicia o monitoramento do estado da conexão de cada um dos servidores de dados. Este monitoramento é realizado para que a lista dos servidores disponíveis esteja sempre atualizada, pois é ela que irá indicar ao cliente em que máquinas poderá ser alojado seu backup.

O cliente é a parte do sistema que terá seus dados armazenados. Para operar o sistema, o cliente precisa se conectar ao servidor de metadados, a fim de enviar suas solicitações tanto de armazenamento quanto de restauração de dados.

Caso o cliente envie uma solicitação de backup, o servidor de metadados irá retornar uma lista contendo o *hostname* dos servidores de dados conectados e o espaço em disco disponível em cada um deles. O cliente, então, vai empacotar, compactar, criptografar e particionar seus dados e então se conectar aos servidores disponíveis para proceder ao envio. Após a confirmação do armazenamento, o cliente informa ao servidor de metadados o seu *hostname* e a lista de arquivos armazenados, juntamente com o *hostname* dos servidores onde estes foram alojados.

Se, por outro lado, a solicitação for de restauração, o servidor de metadados irá buscar nos registros dos arquivos enviados através do *hostname* do cliente e enviará uma lista contendo os arquivos e o *hostname* dos servidores de dados. O cliente então tratará de se conectar a cada um dos servidores e solicitar a recuperação de seus arquivos.

C. Detalhes da Implementação

O projeto foi desenvolvido utilizando a linguagem de programação Java. Portanto o *bytecode* pode ser executado em qualquer plataforma que rode uma máquina virtual Java capaz de se comunicar através de *sockets*.

Apenas o cliente utiliza ferramentas do sistema operacional Linux como o `tar`, para empacotamento dos arquivos, `gzip` para compactação e `openssl` para criptografia. Todos esses programas são chamados através de um *bash script*. Dessa forma, o cliente deverá se executado em um plataforma Linux. Ainda assim, caso o usuário deseje utilizar em uma plataforma Windows, basta usar um conjunto de ferramentas GNU como *MinGW* ou *Cygwin*.

IV. RESULTADOS

Para verificar o funcionamento do sistema, foi feita uma instalação-piloto para servir o backup dos dados do Colegiado do Curso de Ciência da Computação (COLCIC) da Universidade Estadual de Santa Cruz, em Ilhéus/Ba. Os testes basicamente executaram tarefas de backup e restauração de dados utilizando a arquitetura descrita na seção III-B. Foi feito o backup do diretório de usuários (`/home`) de uma máquina Linux, contendo uma estrutura com um total de 8.883 arquivos, cujo tamanho total foi de 330,7MB. No total foram utilizadas 5 máquinas conectadas numa rede Ethernet, nomeadas como segue.

- 3118-01 (cliente);
- 3118-02 (servidor de metadados); e
- 3118-03, 3118-05 e 3118-06 (servidores de dados).

O servidor de metadados foi instalado na máquina 3118-02, enquanto os servidores de backup executaram nas máquinas 3118-03, 3118-05, 3118-06. O experimento processou toda a operação desde o início. Assim, o cliente, executado na máquina 3118-01, se conectou ao servidor de metadados e recebeu a lista dos servidores de dados ativos. Neste experimento, após o empacotamento, compactação e criptografia, dados de backup acabaram divididos em partes de no máximo 100MB, resultando em um total de 2 partes com tamanho 100MB e uma com tamanho de $11,9$MB. Cada uma das três partes do backup foi transferida para um servidor de backup diferente.

Nesse teste, foram medidos os tempos de realização do processo de backup e de restauração com o programa `time` do Linux. No total, o tempo gasto na compactação, encriptação e divisão dos arquivos é mostrado na tabela I.

Tabela I
TEMPOS GASTOS NOS DIFERENTES PROCESSOS

	Compactação, encriptação e divisão dos arquivos	Processo completo de Backup	Concatenação, desencriptação e descompactação dos arquivos	Processo completo de Restauração
Real	$0m43.348s$	$1m6.755s$	$0m32.786s$	$0m57.183s$
User	$0m48.827s$	$1m7.656s$	$0m32.710s$	$0m52.055s$
Sys	$0m02.396s$	$0m3.168s$	$0m02.252s$	$0m03.616s$

A. Considerações sobre Escalabilidade

O ambiente em que este sistema foi testado é pequeno, em função de que os recursos necessários para se montar experimentos maiores, que envolvam mais máquinas e mais dados para backup ainda não estão disponíveis. Isto não permitiu uma análise aprofundada a respeito das características de funcionamento do sistema – notadamente sua escalabilidade. Todavia algumas considerações podem ser feitas.

De modo geral os aspectos que impactam na escalabilidade deste sistema, e consequentemente no tempo necessário para se executar as operações de backup e restauração, são: a velocidade da rede, pois quanto mais rápida forem as conexões mais rapidamente os arquivos poderão ser copiados; a quantidade de servidores de backup, já que quanto mais servidores existirem maior será a quantidade de redundância possível de ser implementada, aumentando a confiabilidade e possibilitando maior vazão de dados no processo caso a infraestrutura de rede permita; e o servidor de metadados, que pode não atender com presteza a todas as solicitações de backup e restauração em ambientes onde exista muita demanda por estes serviços. Este problema pode ser minimizado com a prática de se fazer as operações de backup em momentos cujo tráfego na rede seja pequeno. Além disso, pode-se estudar estratégias que implementem o servidor de metadados de maneira distribuída, diluindo a carga de trabalho por diferentes máquinas.

V. TRABALHOS CORRELATOS

Lilliebridge et al. propõem um sistema de backup cooperativo através da Internet [5]. O sistema funciona utilizando uma rede entre pares (P2P) com a qual dois usuários podem formar parceria, disponibilizando iguais espaços em disco, garantindo uma troca justa. Desta forma, um usuário A armazenaria os dados do usuário B e vice-versa. O sistema proposto, no entanto, sofre do inconveniente dos *free-riders* [1] , que podem realizar o backup de seus dados sem garantir, reciprocamente, a integridade dos dados do outro. A cooperação dos pares é assegurada através da realização de testes periódicos, que verificam a presença e a integridade dos arquivos de backup na máquina remota. Este trabalho presume a utilização de servidores de um mesmo domínio administrativo, eliminando o problema dos *free-riders*.

O OurBackup é um sistema de backup que utiliza uma rede social para construção da relação entre os usuários [6]. Nesse sistema, um usuário efetua o backup de uma parcela selecionada de seus dados na máquina de seus amigos. A rede social embutida no sistema tem como finalidade prática diminuir a presença de *free-riders* e aumentar a recuperabilidade do backup. Apresenta um mecanismo de verificação da integridade do backup, da mesma forma que no trabalho de Lillibridge et al.. Entretanto, difere-se daquele sistema pela utilização de um servidor centralizado para armazenar os metadados e localizar os usuários do sistema. Neste trabalho também se utiliza um servidor centralizado para armazenar as informações sobre os arquivos armazenados e os servidores de dados ativos.

O pStore é descrito como um sistema de backup, baseado em uma rede P2P adaptativa, que utiliza espaço em disco de computadores conectados através da Internet para prover proteção de dados aos usuários [7]. Esse sistema é visto como uma junção de um sistema de armazenamento distribuído com um *framework* de versionamento. Alguns dos objetivos desse sistema são garantir especialmente confiabilidade e segurança. A confiabilidade do backup é garantida através da replicação de dados, criando cópias dos mesmos em servidores distintos. A segurança é obtida com a encriptação dos dados pelo cliente antes do envio.

[1]Em sistemas P2P, o termo *Free-rider* é utilizado para denominar usuários que se beneficiam sem colaborar, ou colaborando o mínimo possível.

Essa estratégia também será utilizada neste trabalho, criptografando os dados antes de enviá-los para o servidor.

Foram buscadas implementações dos sistemas citados acima mas, infelizmente, não foi encontrada a implementação ou registros de utilização real de nenhum dos trabalhos relacionados. Seria interessante obter informações sobre esses sistemas para comparação com o sistema proposto neste trabalho. Além disso, os testes realizados em cada trabalho tinham como objetivo verificar variáveis diferentes, não permitindo a comparação entre elas.

Lillibridge et al. simularam uma rede com 10 participantes e mediu a recuperabilidade do backup utilizando o método de correção de Reed-Solomon, com k = 6 e n = 2. Dessa forma, até 2 das 8 partes do backup poderiam se perder. O processo de backup para 100MB de dados foi realizado em 12 minutos e de 1 GB em 2 horas [5]. No OurBackup foi simulada uma rede com banda, tamanho, redundância do backup, framentação dos arquivos e número de pares variáveis. Verificou recuperabilidade e tempo de recuperação tanto no uso da técnica de replicação quanto com erasure code [6]. No pStore os testes verificaram a utilização de rede e o espaço necessários para realização de backup incremental.

VI. Conclusão

Neste trabalho foi proposto um sistema de backup que utiliza o espaço em disco ocioso de máquinas de uma organização, alcançando segurança e confiabilidade do processo a baixo custo, e aumentando a utilização dos recursos de armazenamento já existentes, sem que seja necessário nenhum investimento adicional para a operação de backup de dados.

O sistema encontra-se hospedado no Gitorious e seu endereço é https://gitorious.org/dbackup. Pode-se também fazer um clone deste projeto através do repositório remoto. Todos os códigos fontes, **scripts** e um relatório técnico a respeito se encontram neste endereço.

Com este sistema o processo de backup tem maior confiabilidade e menor custo, sem perda da confidencialidade pois apesar de alojados remotamente, os dados são enviados após um processo de criptografia. A operação do sistema é simples, baseada em linhas de comando, e sua versão atual é completamente funcional.

Os metadados são imprescindíveis ao funcionamento do sistema e, por este motivo, o sistema não pode tolerar uma falha neste componente. Neste sentido, um trabalho que se posa é a implementação de soluções mais robustas do que o servidor centralizado utilizado aqui. Uma alternativa é fazer com que os servidores de backup sejam eles mesmos também servidores de metadados. Neste caso, a busca de um cliente pelos servidores de backup que possuem seus arquivos seria baseada em um processo de inundação, conforme acontece em sistemas P2P similares ao Gnutella [8].

Outra alternativa é a implementação de uma DHT para distribuir os metadados, adicionando redundância. Neste sentido, algumas implementações de DHT já foram propostas na literatura, como a CAN, Chord, Pastry e Tapestry [9], [10].

Assim como o servidor de metadados, o computador que hospeda os servidores de backup também podem sofrer falhas e perde-los ou corrompê-los. Para lidar com este problema potencial é necessário criar um módulo de verificação de integridade dos arquivos, permitindo ao cliente periodicamente verificar o estado de seu backup. Uma estratégia que pode ser utilizada é a geração de um código MD5 pelo cliente e seu envio ao servidor de metadados. Desta forma, quando o cliente for verificar a integridade dos arquivos, o servidor de dados deve calcular o MD5 e enviar o resultado ao cliente que efetuará a comparação com o código armazenado no servidor de metadados. Além disso, pode-se empregar técnicas de Utilização de *erasure codes*, que permite a recuperação da integridade dos dados mesmo com a perda parcial deles.

Referências

[1] Cibecs, "Business data loss survey," 2011, acessado em: 15 de novembro de 2011. [Online]. Available: http://resources.idgenterprise.com/original/AST-0052774_NEW_Survey-2011_Aug.E.pdf

[2] Seagate, "Importance of backup," [entre 1998 e 2011], acessado em: 17 de janeiro de 2012. [Online]. Available: http://www.seagate.com/ww/v/index.jsp?vgnextoid=011ad85104b51210VgnVCM1000001a48090aRCRD

[3] J. R. Douceur and W. J. Bolosky, "A large-scale study of file-system contents," *SIGMETRICS Perform. Eval. Rev.*, vol. 27, pp. 59–70, May 1999. [Online]. Available: http://doi.acm.org/10.1145/301464.301480

[4] L. Toka, M. Dell'Amico, and P. Michiardi, "On scheduling and redundancy for p2p backup," *CoRR*, vol. abs/1009.1344, p. 9, 2010.

[5] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A cooperative internet backup scheme," in *In Proceedings of the 2003 USENIX Annual Technical Conference*, 2003, pp. 29–41.

[6] M. I. d. S. Oliveira, *OurBackup: Uma Solução P2P de Backup Baseada em Redes Sociais*. Campina Grande: UFCG, 2007.

[7] C. Batten, K. Barr, A. Saraf, and S. Trepetin, "pstore: A secure peer-to-peer backup system," Massachusetts Institute of Technology Laboratory for Computer Science, Technical Memo MIT-LCS-TM-632, October 2002.

[8] A. H. Rasti, D. Stutzbach, and R. Rejaie, "On the long-term evolution of the two-tier gnutella overlay." in *INFOCOM*. IEEE, 2006. [Online]. Available: http://dblp.uni-trier.de/db/conf/infocom/infocom2006.html#RastiSR06

[9] K. Aberer and M. Hauswirth, "An overview on peer-to-peer information systems," 2002.

[10] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Comput. Surv.*, vol. 36, pp. 335–371, December 2004. [Online]. Available: http://doi.acm.org/10.1145/1041680.1041681