

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/280157327>

Cloud Disk Drive: Uma Abordagem para a Criação de Discos Virtuais de Baixo Custo Utilizando Redes p2p

Conference Paper · May 2014

READS

13

4 authors, including:



Anderson Fonseca e Silva

Federal University of Pernambuco

8 PUBLICATIONS 3 CITATIONS

SEE PROFILE



Vinicius Cardoso Garcia

Federal University of Pernambuco

156 PUBLICATIONS 440 CITATIONS

SEE PROFILE



Rodrigo Elia Assad

Universidade Federal Rural de Pernambuco

39 PUBLICATIONS 33 CITATIONS

SEE PROFILE

Cloud Disk Drive: Uma Abordagem para a Criação de Discos Virtuais de Baixo Custo Utilizando Redes p2p

Anderson Fonseca e Silva¹, Vinicius Cardoso Garcia¹,
Rodrigo Elia Assad², Frederico Durão³

¹Universidade Federal de Pernambuco - Centro de Informática (UFPE)
Recife – PE – Brasil

²Universidade Federal Rural de Pernambuco - Departamento de Informática (UFRPE)
Recife – PE – Brasil

³Universidade Federal da Bahia
Instituto de Matemática, Departamento de Ciências da Computação
Salvador, BA – Brazil

{afs8,vcg}@cin.ufpe.br, assad@deinfo.ufrpe.br, freddurao@dcc.ufba.br

Resumo. *Este trabalho apresenta uma proposta para a disponibilização de unidades virtuais para o armazenamento de dados em nuvem, através da associação de tecnologias e conceitos, tais como: iSCSI e redes p2p. Desta forma, os arquivos podem ser distribuídos em nós conectados à rede, permitindo a oferta de armazenamento em disco de maneira compartilhada e a redução de custos pelo ofertante do serviço.*

1. Introdução

Devido à necessidade das empresas em expandir sua capacidade de armazenamento em rede, motivado pelo alto volume no tráfego de e-mails com anexos, ambientes colaborativos de trabalho, banco de dados e dados multimídia, ou por pressões de negócios em âmbito global, que devem estar alinhado com diferentes modelos regulatórios em vários países, torna-se importante o uso de arquitetura de armazenamento, dos quais se destacam: Direct-attached storage (DAS), Network-attached storage (NAS) e Storage-area networks (SANs). Segundo a Intel¹, a SAN é a arquitetura mais escalável se comparada com as duas anteriores, porém, requer o uso de software e hardware especializado, tornando alto o custo inicial para a aquisição de equipamentos. Como solução, a empresa propôs a utilização da arquitetura SAN associada ao advento do iSCSI [Meth et al. 2004] surgido em 2003.

Com o advento da computação em nuvem, surgiram modelos de serviços com foco no armazenamento de dados, tais como: Storage as a Service (STas) e Data as a Service (DaaS). O primeiro permite que as aplicações em nuvem escalem além da limitação de onde estão hospedados, permitindo que usuários armazenem seus dados em discos remotos, acessando-os em qualquer lugar e a qualquer momento [Jiyi et al. 2010], e o segundo, além de fornecer as mesmas funcionalidades do primeiro, permite informações

¹<http://www.broadberry.co.uk/pdf/intel10gbe/ISCSISANNetworks.pdf>

sobre análise de crédito e contabilidade de empresas para clientes. O modelo DaaS oferece funcionalidades que permitem seus clientes adquirirem ou fornecerem dados sob um modelo de serviços, desconsiderando se a informação ofertada é gratuita ou comercial [Truong and Dustdar 2009].

Neste contexto, este trabalho apresenta uma proposta para a oferta de espaço para armazenamento de dados de forma compartilhada, provendo uma cota pré-definida para cada usuário, porém, alocada de forma virtual, proporcionando uma economia na disponibilização de recursos pelo ofertante. Além disso, a proposta dispensa o uso de instaladores ou *graphical user interface* (GUI) proprietárias, facilitando assim a administração na implantação da solução.

No restante deste artigo, as seções estão estruturadas da seguinte forma: Seção 2, tecnologias relacionadas; Seção 3, a solução proposta, arquitetura e projeto, bem como, sua validação; e Seção 4, considerações finais e trabalhos futuros.

2. Tecnologias relacionadas

O propósito desta seção é apresentar as tecnologias e conceitos que embasam a construção desta proposta. As seguintes subseções discorrem sobre o iSCSI, importante para a comunicação entre os clientes e servidores; o JXTA [Gong 2001], selecionado como tecnologia para a implementação da solução, utilizando tecnologias p2p; e por fim, o USTO.RE uma solução de armazenamento em nuvem privada, utilizada na validação da proposta.

2.1. iSCSI

Segundo a SearchStorage², o iSCSI resulta da junção do SCSI (*Small Computer System Interface*) e, o IP *Internet Protocol* desenvolvido pela IETF. Trafegando comandos SCSI sobre redes IP, o iSCSI facilita a transferência de dados sobre intranets, facilitando o gerenciamento no armazenamento em longas distâncias. Deste modo, este modelo de conectividade tem se tornado uma das tecnologias-chave para a implantação e transmissão de dados no mercado de SAN (*Storage Area Network*), aumentando sua capacidade e performance. Devido a ubiquidade das redes IP, o iSCSI pode ser utilizado para transmitir dados em LAN (*Local Area Networks*), ou Internet permitindo um modelo de armazenamento de dados independente de localização.

2.2. JXTA (Juxtapose)

No contexto de tecnologias para a construção de redes p2p, o JXTA se apresenta como uma alternativa para a implementação utilizando a plataforma Java. As vantagens relativas à utilização de redes p2p comparadas com o modelo cliente/servidor tradicional estão na escalabilidade e tolerância a falhas [Das et al. 2010]. O JXTA é uma especificação para a plataforma p2p desenvolvida pela Sun Microsystems sob a direção de Bill Joy e Mark Clary.

Uma das principais funcionalidades da plataforma é fornecer um padrão, permitindo que desenvolvedores comerciais e de código-aberto criem serviços interoperáveis e aplicações. O JXTA foi modelado utilizando um pequeno número de protocolos para o

²<http://searchstorage.techtarget.com/definition/iSCSI>

tratamento de serviços. Estes protocolos podem ser implementados utilizando qualquer linguagem, permitindo dispositivos heterogêneos se comunicarem um com o outro em uma rede p2p.

2.3. USTO.RE

O projeto USTO.RE [Durao et al. 2013] consiste em uma solução de baixo custo para o armazenamento de arquivos de forma distribuída utilizando redes p2p. Nesta solução, os arquivos são separados em *chunks* (pedaços com tamanho pré-definido) e gravados em outros nós conectados à rede, de acordo com o algoritmo de replicação em execução.

3. Cloud Disk Drive

Nesta seção serão apresentadas as definições relativas a proposta, bem como sua análise, projeto e implementação utilizando como infra-estrutura a associação entre o iSCSI e a solução USTO.RE.

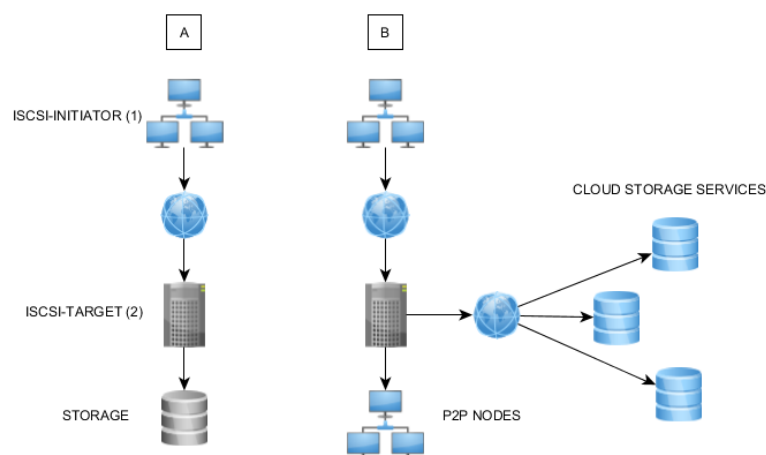


Figura 1. Visão geral.

Na Figura 1.A é possível verificar a solução tradicional para armazenamento de dados utilizando iSCSI, onde os equipamentos acessam o iSCSI-Target através do iSCSI-Initiator, que se encontra disponível na maioria dos sistemas operacionais. Neste modelo, quando o usuário ou aplicação solicita alguma operação, um pacote com um cabeçalho é adicionado antes dos pacotes IP serem enviados (1), e, quando da chegada dos pacotes na outra ponta, estes são separados entre comandos iSCSI e o conteúdo da requisição (2), por fim, os comandos são enviados para o controlador iSCSI, e em seguida, repassados para o dispositivo de armazenamento. As vantagens dessa abordagem estão na facilidade de execução sobre redes Ethernet.

Ainda na Figura 1.B, o modelo proposto segue os mesmos moldes de comunicação da proposta da Figura 1.A, porém, o diferencial está no armazenamento dos dados enviados pelos usuários. Neste, o ofertante do iSCSI-Target não necessita de unidade de armazenamento, a proposta se volta para a utilização dos nós conectados à rede p2p, disponibilizando uma cota de espaço em seus discos rígidos, o que torna possível um melhor aproveitamento dos recursos já adquiridos pela empresa.

3.1. Design da solução

Para a implementação deste trabalho foi tomado como requisito principal, prover ao usuário por meio de uma comunicação iSCSI, uma unidade de disco pré-formatada, utilizando um sistema de arquivos, com um tamanho de armazenamento pré-definido, podendo variar entre 500GB ou 1TB. Contudo, a forma de alocação da área de dados da partição escolhida, no lado do servidor, não deveria acontecer, simulando deste modo, uma quantidade virtualmente disponível de armazenamento.

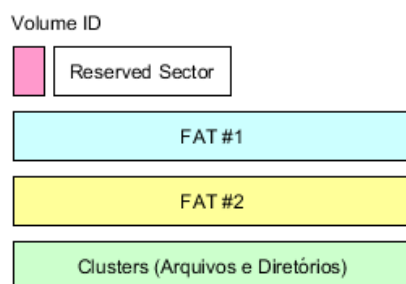


Figura 2. Layout do sistema de arquivos FAT32.

Na implementação inicial foi selecionado o sistema de arquivos FAT32 [Corporation 2004], devido à sua disponibilidade de documentação, sobre como obter o retorno dos arquivos e metadados. Conforme a Figura 2 o modelo FAT32 é simples, onde o primeiro setor é sempre o Volume ID, seguido por espaços não utilizados (Reserved Sector), consequentemente, seguem duas cópias do sistema de alocação de arquivos (FAT). O restante do sistema de arquivos é organizando em *clusters*, sendo esta, a área que consiste na maior parte do tamanho do disco, utilizado para armazenar todos os arquivos e diretórios.

Sendo assim, esta trabalho propôs a separação dos setores iniciais do disco (Volume ID, Reserved Sector, FAT1 e FAT2), da área de *clusters*. Deste modo, como as informações necessárias para inicialização da partição já estão contidas nas primeiras seções do disco, é possível oferecer um disco com capacidade de 1 TB, alocando tão somente 1 GB de dados, o que representa uma economia inicial de 99 por cento na alocação de espaço.

3.2. Implementação e Funcionamento

Para a implementação da proposta foi utilizada a tecnologia Java e um conjunto de ferramentas, tais como: jSCSI³ para a adaptação do iSCSI-Target; FAT32 Format, para permitir a formatação de volumes maiores que o permitido pelo Windows, neste caso, 500GB ou 1TB, e por fim, o SGBD MySQL⁴, para armazenar os endereços dos *clusters*.

Na Figura 3 é apresentada a visão de implementação, onde o usuário solicita a inicialização da partição através de um aplicativo web (1) e, em seguida, inicia a conexão com o iSCSI-Target através de um endereço pré-definido (2). A partir deste momento, o Target recupera o arquivo contendo as informações da partição (particao.dat), onde estão

³<http://jscsi.org/>

⁴<http://mysql.org/>

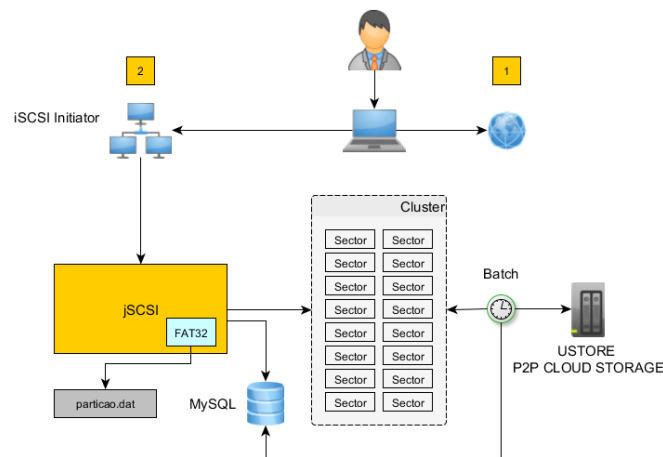


Figura 3. Visão de implementação.

contidas as informações de inicialização. Por fim, a unidade é exibida no gerenciador de arquivos do usuário.

Neste contexto, no momento em que os usuários enviam arquivos ou criam pastas, as informações são enviadas em segmentos para o Target e gravados inicialmente no arquivo de partição. A justificativa para esta abordagem é permitir uma redução da latência no envio dos dados. Em outro momento, havia sido cogitado a gravação dos segmentos em uma estrutura fora da partição, o que gerou uma alta taxa de I/O, degradando a performance da aplicação. Dentro deste processo, os segmentos recebidos são registrados em um banco de dados, bem como, o identificador de sua partição.

Para o envio dos dados para a solução de armazenamento em nuvem, foi criada uma abordagem utilizando uma execução em Batch. Porém, para a leitura das informações na partição, foi criado um extrator que entende como são gravados os dados na FAT32, recompondo-os em forma de arquivos, ao invés de segmentos, para que o Batch, consiga transferir para a solução em nuvem desejada, neste caso, o USTO.RE. Além disso, o Batch tem a responsabilidade de controlar os arquivos extraídos e já enviados, evitando o reenvio ou a recuperação e escrita desnecessária.

3.3. Validação inicial

Neste primeiro momento a validação se deu, com o objetivo de verificar a viabilidade da proposta. O ambiente de execução foi montado utilizando uma rede de 100Mbps, para o iSCSI Initiator, um notebook com CPU core i5, 4GB RAM e 500GB de disco rígido. O iSCSI-Target foi montado em um equipamento com um processador core i-5, 4GB de RAM, sistema operacional Windows 7-64 Bits e 2TB de disco rígido.

Neste contexto, foi simulado um disco de 1TB, neste os arquivos de inicialização da FAT ocuparam 1GB de espaço do arquivo. A vazão média aferida para o envio de 1000 arquivos foi de 10.11 segundos, e, 9.96 para 10.000 arquivos. Considerando o envio para o iSCSI, desprezando a execução do batch para o envio dos dados para o USTO.RE inicialmente.

4. Considerações finais

Esta proposta forneceu as ideias iniciais para a elaboração de um sistema de armazenamento colaborativo de baixo custo, emulando um disco virtual com capacidade superior ao que é possível ofertar de forma física, através do envio dos dados para uma solução em nuvem. Neste, é possível reduzir a curva no aprendizado na utilização da solução, através do uso dos aplicativos de gerenciamento de arquivos já fornecidos pelos sistemas operacionais, possível através do uso do iSCSI.

Este trabalho se encontra em andamento, e questões como performance, bufferização, latência na recuperação dos arquivos e outras *issues*, serão consideradas à medida em que o trabalho evolui.

5. Agradecimentos

Este trabalho foi apoiado parcialmente pelo Instituto Nacional de Ciência e Tecnologia para Engenharia de Software (INES)⁵, financiado pelo CNPq e FACEPE, processos 573964/2008-4 e APQ-1037-1.03/08.

Referências

- Corporation, M. (2004). Microsoft fat specification. pages 1–37.
- Das, S., Agrawal, D., and Abbadi, A. E. (2010). Elastras: An elastic transactional data store in the cloud. *CoRR*, abs/1008.3751.
- Durao, F. A., Assad, R. E., Fonseca, A., Carvalho, J. F. S., Garcia, V. C., and Trinta, F. (2013). Usto.re: A private cloud storage software system. In Daniel, F., Dolog, P., and Li, Q., editors, *ICWE*, volume 7977 of *Lecture Notes in Computer Science*, pages 452–466. Springer.
- Gong, L. (2001). JXTA: A Network Programming Environment. *IEEE Internet Computing*, 5(3):88–95.
- Jiyi, W. J. W., Ping, L. P. L., Ge, X. G. X., Wang, Y. W. Y., and Fu., J. F. J. (2010). Cloud storage as the infrastructure of cloud computing.
- Meth, S. J. K., Sapuntzakis, C., Chadalapaka, M., and Zeidner, E. (2004). Internet small computer systems interface (iscsi). In IETF, editor, *Internet Small Computer Systems Interface (iSCSI)*. IETF-RFC-3720.
- Truong, H.-L. T. H.-L. and Dustdar, S. (2009). On analyzing and specifying concerns for data as a service. In IEEE, editor, *IEEE Asia-Pacific Services Computing Conference (APSCC)*, pages 555–566. IEEE.

⁵<http://www.ines.org.br/>