Um Modelo de Conjunto de Trabalho de Arquivos Aplicado à Recuperação de Backup P2P

Eduardo M. Colaço, Marcelo Iury S. Oliveira, Alexandro S. Soares, Francisco Brasileiro, Dalton S. Guerrero

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Laboratório de Sistemas Distribuídos
Av. Aprígio Veloso, s/n, Bodocongó
58.109-970 – Campina Grande – PB – Brasil

{eduardo,iury,alexandro,fubica,dalton}@lsd.ufcg.edu.br

Abstract. *The high churn and low bandwidth characteristics of peer-to-peer (P2P) backup systems make recovery a time consuming activity that increases system's outage. This is especially disturbing from the user perspective, because during outage the user is prevented from carrying out useful work. Nevertheless, at any given time, a user typically requires only a small fraction of her data to continue working. If the backup system is able to quickly recover such files, then the system's outage can be greatly reduced, even if a large portion of the data lost is still being recovered. In this paper, we evaluate the use of a file system working set model to support efficient recovery of a P2P backup system. By exploiting a simple LRU-like working set model, we have designed a recovery mechanism that substantially reduces outage and allows the user to return faster to work. The simulations we have performed show that even this simple model is able to reduce the outage by as much as 80%, when compared to the state-of-practice in P2P backup recovery.*

Resumo. *A alta intermitência e as limitações de banda passante dos nós características de sistemas de backup entre-pares (P2P, do inglês peer-to-peer) aumentam o tempo necessário para recuperar o backup, o que por sua vez aumenta a indisponibilidade do sistema (outage). Contudo, a qualquer instante, apenas uma fração dos dados é necessária para que o usuário prossiga com o seu trabalho. Se o sistema de backup for capaz de recuperar prioritariamente essa fração, o outage pode ser reduzido, mesmo que uma parcela significativa dos dados ainda esteja sendo recuperada. Neste artigo nós avaliamos o uso de um modelo de conjunto de trabalho de arquivos para aumentar a eficiência da recuperação de um sistema de backup P2P. Nós exploramos um modelo muito simples que prioriza os arquivos mais recentemente usados (LRU). A avaliação por simulação do mecanismo proposto mostra que ele é bastante eficiente, podendo atingir reduções de até 80% do tempo de outage, quando comparado ao mecanismo usado atualmente pelos sistemas de backup P2P.*

1. Introdução

Os computadores estão cada vez mais presentes nas nossas vidas, facilitando a execução de nossas atividades e permitindo armazenar informações digitalmente. O armazenamento digital de informação permitiu ao indivíduo médio armazenar uma grande massa de dados regularmente. Contudo, a integridade desses dados pode ser comprometida por falhas de *hardware*, vírus, entre outras causas comuns, sem aviso prévio. Em muitos casos, a perda desses dados não é um fato tolerável e pode causar enormes prejuízos. A realização de cópias dos dados (*backup*) pode anular ou, ao menos amenizar, tais prejuízos. Atualmente, existem várias técnicas de *backup,* as mais comuns são: armazenamento de cópias em mídias removíveis, serviços web de *backup* (ex. http://mozy.com/ e http://DataDeposit Box.com/) e sistemas de *backup* colaborativo entre-pares (P2P, do inglês *peer-to-peer*) [1, 2, 3, 4].

Com base no fato de que, comumente, usuários não utilizam por completo a capacidade de armazenamento de suas máquinas [5], sistemas de *backup* P2P compartilham a capacidade de armazenamento ociosa dos nós[1] e os conecta de maneira colaborativa, promovendo um serviço mútuo de *backup*. Cada nó possui uma coleção de arquivos (*backup set*) que é replicada no espaço ocioso dos outros nós. Dado que os nós são componentes não-confiáveis, isto é, podem abandonar o sistema ou sofrer falhas a qualquer momento, torna-se um desafio a construção de um sistema de armazenamento confiável [4]. Dessa forma, confiabilidade e recuperabilidade surgem como métricas essenciais para avaliar o seu funcionamento.

A confiabilidade diz respeito à persistência de um *backup* ao longo de um intervalo de tempo. Um sistema é dito confiável se provê garantias de integridade dos dados e maximiza a sua probabilidade de recuperação no futuro. A natureza distribuída dos sistemas P2P beneficia a confiabilidade, uma vez que a distribuição geográfica dos nós previne a perda de dados por catástrofes locais [4].

A recuperabilidade, por sua vez, avalia a eficiência do mecanismo de recuperação, comparando o tempo necessário para recuperar o *backup* em um determinado sistema ao tempo de recuperação do mesmo *backup* em um sistema ideal, no qual toda a banda do nó consegue ser efetivamente alocada durante todo o processo de restauração de *backup*. Na prática, a recuperabilidade é muito menor que o caso ótimo. Isso ocorre porque a ocupação da banda depende da disponibilidade e da largura de banda provida pelos outros nós do sistema P2P. Além disso, mesmo quando a recuperabilidade é alta, o tempo total de recuperação pode não ser satisfatório. Considerando, por exemplo, um *backup set* de 10GB e uma conexão com a Internet de 300Kbps, são necessárias pelo menos 78 horas para recuperar os dados. Durante esse período, o usuário permanecerá, potencialmente, impedido de utilizar o sistema, trazendo um inconveniente que pode acarretar prejuízos.

O período de tempo durante o qual o sistema fica indisponível para o usuário é chamado de *outage*. Esse período inicia-se com a falha que comprometeu os dados do usuário e tem seu término quando o sistema reúne condições para que o usuário retorne às suas atividades. Na maioria dos sistemas de *backup*, o usuário só pode voltar a utilizar o

[1] Usaremos o termo nó para nos referirmos à máquina de um usuário do sistema P2P, ou seja, um nó é um *peer* no sistema P2P.

sistema após a recuperação completa do *backup set*, caso em que o *outage* coincide com o tempo de recuperação. Isso ocorre porque o mecanismo de recuperação trata o *backup set* como uma porção homogênea de dados, desconsiderando características que tornam os arquivos distintos entre si e que podem ser exploradas para a elaboração de mecanismos mais eficientes de recuperação.

É sabido que acessos ao sistema de arquivos não são realizados aleatoriamente [6, 7]; a correlação de arquivos quanto ao acesso cria distinções entre si. Além disso, o usuário não precisa de todo o conjunto de dados para prosseguir com o seu trabalho. A qualquer instante, apenas uma fração dos dados, denominada *conjunto de trabalho*, é efetivamente utilizada. Um sistema de *backup* pode explorar as relações de acesso entre arquivos para identificar essa fração e priorizar a sua recuperação. Dessa forma, arquivos prioritários à demanda de trabalho do usuário são recuperados primeiro, fazendo com que o usuário possa retomar suas atividades antes do tempo total de recuperação. O processo de recuperação segue com a restauração do restante do *backup set*, sem prejuízo ao trabalho do usuário.

Neste trabalho, nós avaliamos o impacto que a utilização de um modelo de conjunto de trabalho de arquivos tem na recuperabilidade de sistemas de *backup* P2P. Nós apresentamos um modelo de conjunto de trabalho simples que é usado para priorizar os arquivos que devem ser recuperados prioritariamente após a ocorrência de uma falha catastrófica[2]. Utilizamos simulações alimentadas por dados de rastros de execução (*traces*) reais de utilização de sistemas de arquivos para avaliar esta nova abordagem de mecanismo de recuperação. Nossos resultados indicam que mesmo uma abordagem simples que prioriza os arquivos mais recentemente usados (LRU, do inglês *Least Recently Used*), pode diminuir em até 80% o *outage*. Em alguns casos, a espera do usuário é reduzida em dias e em nenhum caso o modelo de conjunto de trabalho incrementou o *outage*.

Na seção seguinte, discutimos os trabalhos relacionados à priorização de arquivos em sistemas de armazenamento. Na Seção 3, apresentamos a solução proposta, detalhando a relação entre o *outage* e o tempo de recuperação total de um *backup*. A avaliação do mecanismo proposto é apresentada na Seção 4. A Seção 5 conclui o trabalho, com um breve sumário das contribuições e apontando possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

Modelos que priorizam arquivos em sistemas de armazenamento têm sido alvo de vários estudos reportados na literatura. Entretanto, nenhum dos modelos apresentados tem como foco o processo de recuperação de *backup*.

Santhosh [8] apresentou um algoritmo de substituição para *cache* semântico, alimentado por padrões de acessos ao sistema de arquivos. A solução considera que o padrão de acesso de arquivos em sistemas de arquivos distribuídos não é aleatório e que é possível inferir relações entre arquivos através deste padrão. Tais relações podem ser utilizadas em mecanismos de *pre-fetching* de arquivos.

Tait e Duchamp [6] propuseram uma solução de *file hoarding*, chamada *transparent analytical spying*, que monta um conjunto de árvores de processos, formadas de a-

[2] Uma falha catastrófica causa a perda de todos os dados do usuário; um exemplo típico é o *crash* do disco.

cordo com a ordem em que os processos são executados e os arquivos acessados. Montadas as árvores de acesso, heurísticas são utilizadas para distinguir arquivos de aplicações dos arquivos do usuário. A implementação desta solução requer um acoplamento maior ao sistema operacional, devido à ausência de uma API comum, que forneça informações sobre acesso ao sistema de arquivos agregadas por processo.

Kuenning propôs o Seer [7], que utiliza o conceito de distância semântica para agrupar arquivos relacionados. A distância semântica relaciona os arquivos segundo seus padrões de acesso. No mesmo trabalho, são apresentadas heurísticas para remover acessos ao sistema de arquivos que não possuem valor semântico, realizados por aplicações de busca e *scanners*, que prejudicam o algoritmo de inferência de relações entre os arquivos. Posteriormente, em um estudo de otimização dos parâmetros do Seer, Kuenning et al [9] descobriram que, nos cenários que apresentavam os melhores resultados, o Seer se comportava como uma versão levemente modificada do algoritmo de substituição LRU. A simplicidade dessa abordagem, aliada aos bons resultados apresentados no estudo, nos levou a adotá-la como guia para implementação do nosso modelo de conjunto de trabalho.

Ainda como trabalhos relacionados, existem vários sistemas de *backup* P2P [1, 2, 3, 4], dotados de mecanismos próprios de garantia de requisitos de recuperabilidade e confiabilidade. Contudo, nenhum destes sistemas apresenta propostas de mecanismos com objetivo de reduzir o *outage* do sistema.

3. O Papel do Modelo de Conjunto de Trabalho

O problema que estamos tentando resolver é a diminuição do *outage*. A Figura 1 ilustra a relação entre o *outage* e o tempo total de recuperação (*TR*) em um processo de recuperação de um *backup* após uma falha catastrófica. No estágio 1, o sistema está totalmente funcional. O estágio 2 representa a ocorrência de uma falha na qual uma parcela significativa dos dados do usuário são perdidos, impossibilitando que ele continue executando o seu trabalho. A partir deste momento é iniciado o processo de restauração do sistema. O estágio 3 representa o momento em que dados essenciais às atividades do usuário foram recuperados. Neste estágio, o usuário já pode retomar suas atividades, muito embora ainda existam arquivos do *backup set* a serem recuperados. O estágio 4 aponta o fim do processo de recuperação, quando necessariamente o sistema está novamente completamente funcional.

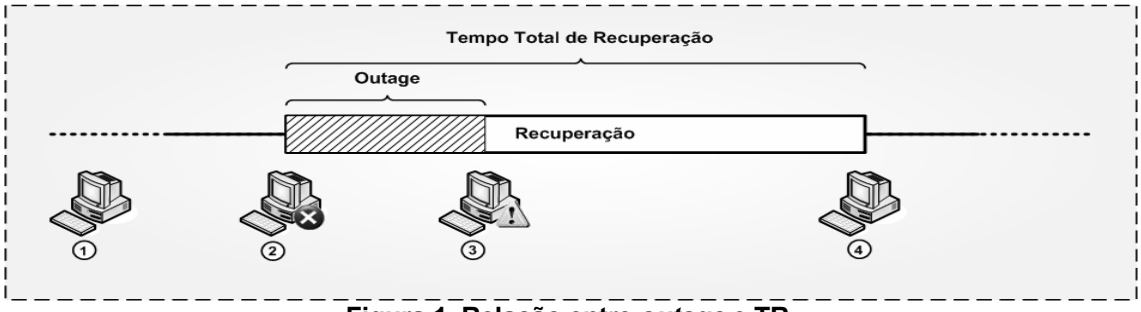


Figura 1. Relação entre *outage* e TR.

É importante perceber que o desenvolvimento das tecnologias da informação e comunicação está fazendo com que a quantidade de informação digital gerada e armazenada por usuários aumente de forma acentuada. Não é incomum que um usuário doméstico possua alguns *giga bytes* (GB) de informação armazenados em seus computadores pes-

soais. Entretanto, como discutido anteriormente, em um determinado instante de tempo, apenas uma pequena fração desses dados é efetivamente utilizada. Denominamos de *conjunto de trabalho* esse subconjunto de arquivos, necessários para que o usuário possa continuar suas atividades após uma falha que exija a recuperação de todo o seu *backup*, ou de pelo menos uma parte significativa do mesmo.

Quando a recuperação do *backup* não usa qualquer mecanismo de priorização dos arquivos, o *outage* tende a ser próximo do *TR*, pois a probabilidade de um arquivo escolhido de forma aleatória do *backup set* pertencer ao conjunto de trabalho é, normalmente, pequena. O papel do modelo de conjunto de trabalho é inferir, com grande probabilidade de acerto, o conjunto de trabalho do usuário, e, assim, priorizar a recuperação desses arquivos. Dessa forma, o modelo de conjunto de trabalho nada mais é que uma relação de ordem total dos arquivos do *backup set* do usuário, baseada na probabilidade inferida de um arquivo ser utilizado em um futuro próximo. Nossa conjectura é que a utilização de um modelo de conjunto de trabalho de arquivos para priorizar a recuperação dos arquivos em um *backup* possa diminuir o *outage*.

Um modelo de conjunto de trabalho ótimo prioriza os arquivos de acordo com a ordem em que serão utilizados no futuro. Dado que é difícil, senão impossível, determinar quando um arquivo será utilizado no futuro, optamos por um modelo de conjunto de trabalho comprovadamente não-ótimo, mas de simples implementação e que apresenta bons resultados. Baseado no trabalho de Kuenning et al [9], o conjunto de trabalho estabelece uma ordenação de relevância temporal, de maneira semelhante ao algoritmo clássico LRU. A ordenação temporal, no entanto, não é utilizada para retirar elementos do conjunto de trabalho, mas para ordená-los de acordo com a sua relevância. Acreditamos que considerar arquivos, ao invés de diretórios, como a unidade a ser tratada pelo LRU nos deixa muito mais próximo de acertar o conjunto de trabalho do usuário, assumindo que a probabilidade de um diretório ter todos os seus arquivos sendo utilizados pelo usuário em um futuro próximo é baixa.

4. Avaliação

O *outage* e o tempo total de recuperação são influenciados por uma combinação de vários fatores. Entretanto, é difícil modelar cada possível evento capaz de prolongar o processo de recuperação de um *backup*. A modelagem analítica se torna ainda mais difícil, considerando-se que estes eventos divergem enormemente quanto a seus valores. Um estudo realizado por Saroiu et al [10] mostrou que os nós participantes das redes Napster e Gnutella são heterogêneos quanto a muitas características: largura de banda, latência, tempo de vida e dados compartilhados. Em alguns casos, esses valores apresentam variações de três a cinco ordens de grandeza entre os nós.

Em geral, simulações permitem a modelagem de sistemas mais próximos da realidade que na abordagem analítica. Cenários em que são variados o tamanho do *backup set*, o momento em que a falha ocorre no sistema, entre outras variáveis, auxiliam na verificação do impacto dessas variáveis na recuperação do *backup*. Portanto, considerando o ambiente dinâmico que envolve o comportamento dos nós em sistemas P2P reais e a difícil modelagem analítica do sistema, o problema foi abordado a partir de simulações.

4.1. Modelo de Simulação

Desenvolvemos um simulador de eventos discretos, no qual as entidades típicas do processo de recuperação de *backup* e do sistema do usuário são modeladas. Um nó exerce o papel de consumidor ou de provedor. Um consumidor submete requisições de recuperação para os provedores. Os provedores, por sua vez, armazenam o *backup* dos consumidores e atendem as requisições de recuperação que disparam a transferência de arquivos dos provedores para um consumidor. Quando a soma das taxas de *upload* dos provedores supera a taxa de *download* de um consumidor, um mecanismo de particionamento do canal de comunicação é executado, dividindo uniformemente o canal, simulando o caso médio de *sockets* TCP.

Assumimos que os arquivos do *backup set* são distribuídos aleatoriamente entre os provedores e que o *backup* efetuado nos nós é sempre perfeitamente sincronizado com o *backup set*. Note que estas simplificações não afetam os resultados, já que estamos interessados exclusivamente no processo de recuperação do *backup*. Mecanismos que mantenham o *backup* sincronizado estão fora do escopo desse trabalho.

Quanto à disponibilidade dos nós, consideramos que o consumidor está todo tempo on-line e que os provedores seguem os padrões de disponibilidade apresentados no estudo de Stutzbach e Rejaie [11], que sugere que a distribuição Weibull (*shape* β, *scale* α) fornece um modelo apropriado para tempo de sessão e tempo de desconexão dos nós em um sistema P2P.

A simulação da recuperação usando um modelo de conjunto de trabalho consiste em duas fases: uma fase de treinamento, que serve para gerar o modelo de conjunto de trabalho e uma fase de recuperação do *backup set*, propriamente dita, utilizando o conjunto de trabalho obtido na fase de treinamento. A simulação da recuperação sem utilizar o modelo de conjunto de trabalho executa apenas a segunda fase descrita acima, fazendo uma escolha aleatória da ordem em que os arquivos do *backup set* são recuperados.

O simulador é alimentado por eventos de sistemas de arquivos, produzidos a partir de rastros de execução reais, que indicam em que instantes de tempo arquivos do sistema de arquivos da máquina de um consumidor são acessados. O parâmetro tempo de falha (*FT*) indica em que instante de tempo a falha catastrófica ocorreu. Ele serve para indicar os pedaços do rastro que serão usados em cada uma das fases de execução da simulação, com a informação anterior a *FT* sendo usada para a fase de treinamento (quando necessária) e a informação posterior a *FT* sendo usada na fase de recuperação do *backup*. Durante a fase treinamento, as entradas do rastro são convertidas em eventos de sistema de arquivos e utilizadas para alimentar o algoritmo de geração do conjunto de trabalho. A recuperação do *backup* se inicia logo após a falha, sendo que decorrido o intervalo de tempo relativo ao *outage* as entradas do rastro posteriores ao *FT* passam a ser utilizadas para simular a seqüência de acesso a arquivos que o usuário faria para continuar suas atividades normais. Definimos que um *file miss* ocorre quando o rastro indica a ocorrência de um acesso a um arquivo que ainda não foi recuperado do *backup*.

O resultado obtido na simulação é composto por dois valores: o tempo total de recuperação e o valor estimado do *outage*. O tempo total de recuperação é medido desde o momento em que a falha ocorreu (*FT*), até o momento em que todos os arquivos foram recuperados. O *outage*, por outro lado, depende da seqüência futura de acessos ao arquivo,sendo definido com o menor intervalo de interrupção na utilização da máquina, tal que

o re-início das atividades do usuário não gere nenhum *file miss* até que o *backup* seja totalmente recuperado. Mais precisamente, seja *TO* o tempo no qual o usuário retoma suas atividades, o *outage* é dado pela diferença *TO-FT*, onde *TO* é o menor tempo tal que a simulação é concluída sem que *file misses* sejam gerados.

Calcular o valor exato do *outage* não é simples. O nosso simulador calcula um valor aproximado da seguinte forma. Considerando que *TO* é um valor dentro do intervalo [*FT*, *FT+TR*], é realizada uma pesquisa binária até que o tamanho do intervalo de busca seja menor que 5 minutos, sendo este, portanto, o valor do erro máximo na estimativa do *outage*.

4.2. Cenários de Simulação

Nas simulações, modelamos apenas um consumidor e 10 provedores, já que nosso objetivo é apenas observar o impacto do conjunto de trabalho na recuperação do *backup* de um consumidor. Além disso, a motivação para a nossa investigação é o desenvolvimento de um sistema de *backup* P2P baseado em redes sociais [4], nas quais o número de provedores é pequeno, pois reflete o tamanho da rede de amigos próximos de um consumidor.

Os dados empíricos observados no estudo de Stutzbach e Rejaie [11] indicam que os eventos de chegada e partida de nós podem ser bem modelados com uma distribuição Weibull de parâmetros β=0,59 e α=40. Já os tamanhos dos arquivos presentes no *backup set* são modelados a partir dos resultados apresentados em Crovella et al [12], que estimam que a distribuição de tamanho de arquivos em um sistema de arquivos UNIX segue uma distribuição Paretto com parâmetros α=1,05 β=3.800.

Outro parâmetro do simulador é a largura de banda de *upload* dos nós provedores e de *download* do nó consumidor. As capacidades de banda passante de *upload* e *download* de cada nó consideradas para os experimentos correspondem a valores fornecidos no estudo de Horrigan [13]. Este estudo aponta que 50% dos nós conectados à Internet utilizam conexão por linha discada e o restante dos nós utiliza conexão de banda larga. Os nós que utilizam banda larga ainda se dividem em dois segmentos: 25% utilizam conexões de 128/384 Kbps para *upload*/*download* e o restante utiliza 384/1.500 Kbps. Baseado nestes dados, os cenários simulados consideram 5 provedores com capacidade de *upload* de 33,6 Kbps, 3 provedores com capacidade de *upload* de 128 Kbps e dois provedores com capacidade de *upload* de 384 Kbps. O nó consumidor possui 1.500 Kbps de capacidade de *download*.

Os rastros de execução utilizados foram coletados por Kuenning et al [9]. Estes rastros foram utilizados para avaliar o Seer e estão disponíveis no diretório público de rastros do Seer [14]. Optamos pelos rastros da máquina Norgay, com padrões de acesso coletados em um intervalo de 61 dias, já que o restante das máquinas apresentava problemas de ordenação temporal em diversos pontos dos rastros. Um outro motivo para a escolha desses rastros foi a longa duração dos mesmos, que permitem a realização de um estudo mais adequado sobre o período de treinamento do modelo de conjunto de trabalho.

Além dos parâmetros já citados, que definem uma configuração base para a nossa avaliação, definimos diferentes cenários através da variação de três outros parâmetros do modelo de simulação: o tamanho do *backup set*, a sobrecarga de armazenamento, e o instante da falha.

O *backup set* usado nas simulações corresponde ao conjunto de arquivos pertencentes à pasta pessoal do usuário. Os tamanhos dos *backup sets* (*S*) utilizados são: 0.5GB, 1.5GB e 3.0GB. Nem todos os rastros utilizados continham referências a arquivos com o volume de dados desejado, então o *backup set* foi preenchido com arquivos fictícios, gerados aleatoriamente, até que este tamanho fosse alcançado. Os arquivos inseridos correspondem aos arquivos que, apesar de pertencerem ao *backup set*, não são acessados durante o período de coleta do rastro de execução. Alguns exemplos desses arquivos são fotos, vídeos e documentos antigos do usuário, que muito embora sejam raramente acessados, compõem, muito provavelmente, uma parte significativa do *backup set*.

A sobrecarga de armazenamento (*k*) é o número de réplicas para cada arquivo do *backup set*, que será armazenada nos provedores. É esperado que maiores valores de *k* produzam menores valores de *outage* e tempo total de recuperação. Por outro lado, maiores valores de *k* implicam na necessidade de uma maior contribuição para o sistema de *backup* P2P. Idealmente, para fazer *backup* de *n* bytes um nó deve ofertar ao sistema $k \times n$ bytes de seu disco. Foram avaliados cenários onde *k* assume os valores 1, 2, 3, 4 e 5.

O tempo de falha (*FT*) é o parâmetro que determina o ponto no rastro de execução em que a falha ocorreu. Para os rastros utilizados, os valores possíveis variam entre 0 e 61 dias. Foram considerados tempos de falha de 26, 29, 32, 35, 38 dias após o início do rastro.

4.3. Resultados dos testes

Para garantir representatividade na análise das amostras, todos os cenários discutidos na seção anterior foram executados um número de vezes suficiente para garantir um nível de confiança de 95% com um erro inferior a 5% para mais ou para menos. No cenário que demandou um maior número de simulações foram necessárias 150 execuções.

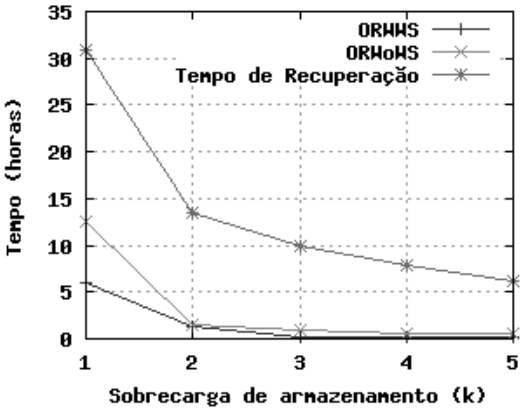
O objetivo do nosso trabalho é avaliar o impacto do uso do modelo de conjunto de trabalho no processo de recuperação do *backup*, mais precisamente na sua influência em relação ao *outage* (*O*). O valor absoluto de *O* é uma indicação de quanto se pode reduzir a indisponibilidade do sistema através do uso de conjuntos de trabalho. Muito embora esta seja uma métrica importante, ela não permite a comparação do desempenho do sistema em cenários diferentes, dado que essa não é uma métrica normalizada. Desse modo, além do valor de *O*, analisamos também uma outra métrica chamada de redução da indisponibilidade do sistema ($O_{decrease}$). A redução de indisponibilidade é uma métrica normalizada definida por:

$$O_{decrease} = (O_{RWoWS} - O_{RWWS})/O_{RWoWS},$$

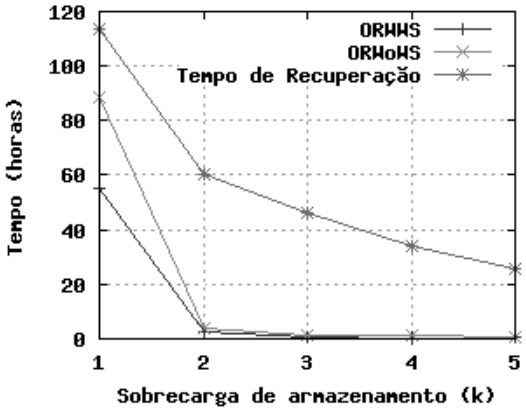
onde O_{RWoWS} é o *outage* obtido sem o uso do conjunto de trabalho na recuperação, em que a ordem de recuperação dos arquivos é aleatória, e O_{RWWS} é o *outage* obtido quando se utiliza o conjunto de trabalho na recuperação.

A Figura 2 mostra os valores do *outage* agrupados pelo tamanho do *backup set* e pela sobrecarga de armazenamento. O impacto do uso do conjunto de trabalho, considerando-se apenas o valor absoluto em horas, é maior para menores valores de *k*. No cenário com *S*=3GB e *k*=1 a diferença entre O_{RWoWS} e O_{RWWS} foi maior que 40 horas, enquanto que no cenário com mesmo valor de *S* e *k*=5 a diferença foi de apenas pouco mais de 4 minutos. Contudo, é importante lembrar que para prover uma sobrecarga de armazenamento maior que 1, os nós de sistema colaborativo de *backup* têm que doar uma quantidade de

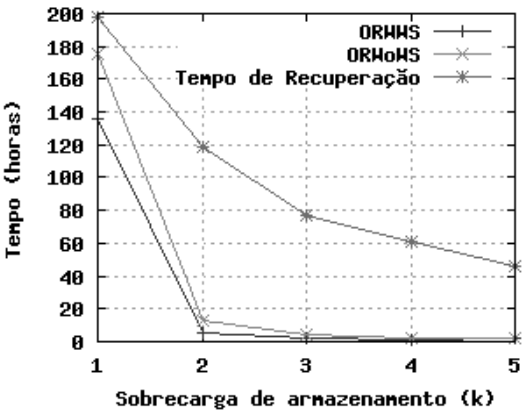
recursos que é diretamente proporcional a *k*. Portanto, o modelo de conjunto de trabalho se apresenta como uma alternativa à sobrecarga de armazenamento, diminuindo o *outage* sem aumentar o consumo de recursos.



(A) *S*=0.5GB



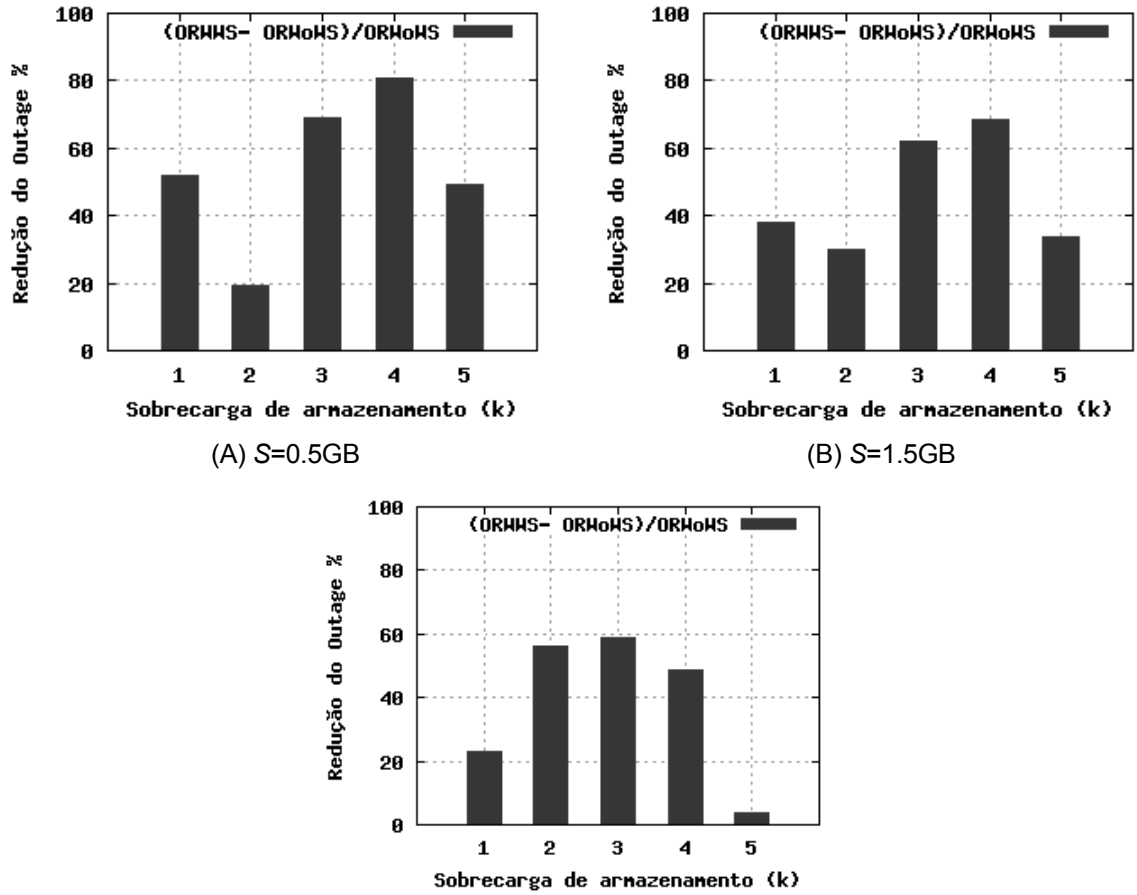
(B) *S*=1.5GB



(C) *S*=3.0 GB

Figura 2. *Outage* em horas X sobrecarga de armazenamento. Resultados para um *backup set* de tamanho (A) 0.5GB, (B) 1.5GB e (C) 3.0GB.

Um resultado esperado que pode ser observado na Figura 2 é que a indisponibilidade aumenta à medida que o tamanho do *backup set* cresce, e é reduzida com o incremento da sobrecarga de armazenamento. O aumento do *backup set* implica em mais dados para serem recuperados, e por conseqüência, maior tempo de recuperação do *backup*. Já o aumento da replicação possibilita melhor uso do canal de comunicação, permitindo uma recuperação mais rápida do *backup*. Finalmente, os resultados apresentados na Figura 2 mostram que o *outage* é bem menor que o tempo de recuperação total. Este resultado confirma nossa hipótese inicial de que o usuário pode retornar às suas atividades muito antes da recuperação total do *backup*, podendo economizar dias de trabalhos que de outro modo seriam perdidos, e reforça a importância de um modelo de conjunto de trabalho para recuperação de *backups* em sistemas P2P.



(A) *S*=0.5GB (B) *S*=1.5GB

(C) *S*=3.0 GB

Figura 3. Redução do *outage* X sobrecarga de armazenamento. Resultados para um *backup set* de tamanho (A) 0.5GB, (B) 1.5GB e (C) 3.0GB.

A Figura 3 mostra os resultados da redução da indisponibilidade, também agrupados pelo tamanho do *backup set* e pela sobrecarga de armazenamento. Estes resultados complementam os da Figura 2, pois demonstram o impacto do uso do conjunto de trabalho de maneira proporcional. A melhor redução (80,9%) foi obtida no cenário com *S*=0.5GB e *k*=4 e o pior caso (3,7%) no cenário com *S*=3GB e *k*=5. Na maioria dos cenários o valor de $O_{decrease}$ foi maior que 35%. Finalmente, observa-se que a indisponibilidade não foi incrementada em nenhum dos cenários analisados.

5. Conclusão e Trabalhos Futuros

Neste trabalho apresentamos um modelo de conjunto de trabalho para recuperação de arquivos em um sistema de *backup* P2P, bem como uma metodologia de avaliação do impacto deste modelo na redução da indisponibilidade do sistema para o usuário. Por fim, apresentamos resultados de experimentos obtidos através da execução desta metodologia, variando parâmetros de tamanho de *backup set*, sobrecarga de armazenamento e tempo de falha. De acordo com os nossos resultados, esta indisponibilidade pode ser reduzida em até 80%, quando comparada com aquela obtida com a utilização da estratégia de recuperação dos atuais sistemas de *backup* P2P. O impacto do conjunto de trabalho foi mais significativo nos cenários com menor sobrecarga de armazenamento, chegando a diminuir o

outage em dias. Além disso, não houve um cenário sequer em que o *outage* fosse incrementado pelo uso do modelo de conjunto de trabalho. Nossos resultados também indicam que o *outage* é muito menor que o tempo total de recuperação, o que implica que o usuário pode retornar ao trabalho muito antes da recuperação total do *backup*.

Apesar dos bons resultados apresentados, ainda há necessidade de complementações do estudo. Acreditamos que uma alta variabilidade no padrão de acesso ao sistema de arquivos pode afetar em muito o mecanismo proposto, por isso faz-se necessária a simulação de mais cenários, com rastros de execução de múltiplas classes de usuários com graus distintos de variação no acesso ao sistema de arquivos. Outro fato que deve ser levado em consideração é a utilização de uma janela de histórico de tamanho fixo. Essa janela limitaria a quantidade de informação apresentada ao mecanismo de detecção de conjunto de trabalho, e poderia fornecer indícios da importância de históricos de curto, médio e longo prazo na recuperação de *backup*.

Há ainda outros aspectos relevantes que foram levantados ao longo da pesquisa deste trabalho e que podem levar a melhoramentos na eficiência de sistemas de *backup* P2P como um todo. Alguns desses aspectos são: o refinamento de heurísticas para detecção de arquivos de maior importância; a implementação de outros modelos de conjunto de trabalho; e a alocação de réplicas de arquivos prioritários em nós que apresentam maior disponibilidade e banda. Além disso, o modelo atual não aproveita as informações de padrões de acesso ao sistema de arquivos durante o processo de recuperação. Essa informação poderia ser utilizada em associação com algoritmos de *pre-fetching*, antecipando a transferência de arquivos necessários que não estavam sendo priorizados pelo conjunto de trabalho no momento anterior à falha.

Todos os aspectos mencionados acima estão sendo considerados no contexto de um sistema de *backup* P2P chamado OurBackup, que está sendo desenvolvido por nosso grupo de pesquisa [4]. Para alimentar nosso algoritmo de conjunto de trabalho é necessário um mecanismo que forneça informação dos tempos de acesso dos vários arquivos do *backup set*, podendo ser implementado através de um mecanismo de notificações de eventos no sistema de arquivos. No Linux, este mecanismo pode ser implementado através do INotify, que é mecanismo assíncrono de notificação de eventos no sistema de arquivos. O Microsoft Windows também fornece, através da API Win32, funcionalidades semelhantes ao INotify, tendo apenas a limitação de não informar eventos de abertura de arquivos. Como o OurBackup segue um modelo P2P, todo o cálculo do LRU será efetuado localmente, implicando apenas na manutenção de uma lista ordenada, não trazendo, assim, problemas de escalabilidade para o sistema.

Agradecimentos

Este trabalho foi desenvolvido em colaboração com a HP Brasil P&D. Agradecemos a Milena Oliveira e Paolo Victor pelas discussões durante a concepção deste trabalho. Francisco Brasileiro é Bolsista do CNPq – Brasil.

Referências

[1] Landers, M. and Zang, H. and Tan, K. L. (2004) "PeerStore: Better Performance by Relaxing in Peer-to-Peer Backup", In Proceedings of the 4th International Conference on Peer-to-Peer Computing, p. 72-79.

[2] Lillibridge, M. and Elnikety, S. and Birrel, A. and Burrows, M. and Isard, M. (2003) "A cooperative internet backup scheme", In Proceedings of the 2003 Usenix Annual Technical Conference, p. 29–41.

[3] Batten, C. and Barr, K. and Saraf, A. and Treptin, S. (2001) "pStore: A secure peer-to-peer backup system", Technical Memo MIT-LCS-TM-632, MIT Laboratory for Computer Science.

[4] Oliveira, M. (2007) "OurBackup: Uma solução P2P de backup baseada em redes sociais", Master of Science Thesis, Universidade Federal de Campina Grande, Campina Grande, PB.

[5] Doucer, J. R. and Bolosky, W.J. (1999) "A Large-Scale Study of File-System Contents", In Proceedings of the 1999 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, p. 59-69.

[6] Tait. C. D. and Duchamp, D. (1991) "Detection and Exploitation of File Working Sets", In Proceedings of the 11th International Conference on Distributed Computing Systems.

[7] Kuenning, G. H. (1997) "Seer: Predictive File Hoarding for Disconnected Mobile Operation", PhD thesis, University of California, Los Angeles, CA.

[8] Santhosh, S. (2004) "Factoring file access patterns and user behavior into caching design for distributed file system". Tech. Rep. MIST-TR-2004-013, Master of Science Thesis, Wayne State University.

[9] Kuenning, G.H., Ma, W., Reiher, P.L. e Popek, G.J. (2002) "Simplifying Automated Hoarding 57 Methods". In Proceedings. of the 5th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'02), Atlanta, GA.

[10] Saroiu, S., Gummadi, P.K. and Gribble, S. G. (2002) "A measurement study of peer-to-peer file sharing systems". In Proceedings of the SPIE Multimedia Computing and Networking (MMCN2002)

[11] Stutzbach, D. and Rejaie, R. (2006) "Understanding churn in peer-to-peer networks". In Proceedings of the 6th ACM SIGCOMM on Internet measurement, pp. 189–202, New York, NY.

[12] Crovella, M.E., Taqqu, M.S. and Bestavros, A. (1998) "Heavy-tailed probability distributions in the World Wide Web". In Applications of Heavy-Tailed Probability Distributions, Adler, Feldman, and Taqqu Ed., Birkhauser, Boston, MA. pp. 3–25.

[13] Horrigan, J. (2005) "Broadband Adoption at home in the United States: Growing but Slowing," In Proceedings of the Telecommunications Policy Research Conference, also available at http://www.pewinternet.org/pdfs/PIP_Broadband.TPRC_Sept05.pdf, accessed Oct. 2007.

[14] Seer Public Traces. Web Page found at www.lasr.cs.ucla.edu/geoff/seer_traces.htm, Accessed in: October 2007.