

Uma Arquitetura para Compartilhamento de Dados e Recursos Computacionais de Armazenamento em Redes P2P Sociais

Fabício Barros Gonçalves, Carlo Oliveira, Vanessa Braganholo

Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Rio de Janeiro (UFRJ)

fabriciobarros@saocamilo-es.br, carlo@nce.ufrj.br,
braganholo@dcc.ufrj.br

Relatório Técnico NCE 03/09

Resumo. *O gerenciamento e o acesso transparente a dados e aos recursos computacionais de armazenamento em ambientes altamente distribuídos de e-science, é um problema difícil. Uma alternativa para esse problema é o modelo de computação peer-to-peer (P2P) devido à possibilidade de agregar recursos computacionais de armazenamento sob demanda. Assim, redes P2P podem ser utilizadas para formar infraestruturas de ambientes computacionais altamente distribuídos. Como pesquisas normalmente envolvem colaboração, cientistas podem formar redes sociais sobre redes P2P para compartilhar recursos computacionais e dados de interesse de todos. Por isso, este trabalho propõe um modelo arquitetural para o compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados compartilhados em redes P2P sociais. A arquitetura permite que recursos computacionais de armazenamento possam ser agregados em larga escala e, como consequência disso, grandes volumes de dados compartilhados possam ser armazenados sobre esses recursos.*

1. Introdução

O uso de ferramentas avançadas para computação de alto desempenho (por exemplo, grades computacionais), e de sistemas de gerenciamento de workflows têm revolucionado as atividades de pesquisa em diversos domínios como química, física, biologia, medicina, e até mesmo ciências sociais. Nesse cenário, cientistas definem suas hipóteses e criam experimentos com base em um conjunto de algoritmos e programas que operam em ambientes de computação distribuída, permitindo, então, a geração, coleta e armazenamento de um grande montante de dados experimentais. Porém, organizar esses dados em um repositório ao qual pesquisadores possam ter acesso a qualquer hora e de qualquer local requer atividades especializadas, tais como as exercidas por administradores de infraestruturas computacionais, ou a utilização de ferramentas computacionais para lidar com grandes volumes de dados. Nesse sentido,

tais ferramentas devem permitir a distribuição de dados sobre recursos computacionais de armazenamento em um ambiente altamente distribuído.

Segundo Mattoso *et al.* (2009), o gerenciamento e a transparência de acesso aos recursos computacionais de armazenamento em um ambiente altamente distribuído, com uma boa relação custo/desempenho, é um problema difícil. Além disso, é possível observar a falta de técnicas de computação distribuída que escalem para configurações muito grandes, e, ao mesmo tempo, tratem da autonomia, dinâmica e heterogeneidade dos recursos [Mattoso *et al.* 2009]. Assim, uma alternativa para o tratamento desses problemas é o modelo de computação *peer-to-peer* (P2P), devido à possibilidade de poder agregar recursos computacionais (ex., processadores, memória, discos, conteúdo, software, etc.) em larga escala ao ambiente, por meio de computadores localizados na periferia ou conectados diretamente ao *backbone* da Internet.

Com base nesse contexto, infraestruturas como as apresentadas por Semenov (2005) têm sido propostas com o objetivo de tirar proveito dos tipos de recursos mencionados anteriormente. Tais infraestruturas são conhecidas como redes *peer-to-peer* (P2P). Dessa forma, uma vez que uma infraestrutura de rede P2P é disponibilizada, camadas de software podem ser construídas, permitindo o compartilhamento e acesso transparente a dados e recursos computacionais compartilhados [Milojicic et al. 2002]. Nesse sentido, uma camada de software baseada em requisitos de redes sociais pode ser construída a fim de agregar recursos computacionais de *peers* com interesses em comum [Liu, Antonopoulos e Mackin 2007; Lin et al. 2007; Carchiolo et al. 2008]. Essa camada de software, então, permite criar uma abstração de rede social sobre uma infraestrutura de rede P2P, onde *peers* representam pessoas e os conjuntos de conexões entre *peers* representam os relacionamentos pessoais. Esses relacionamentos sociais são formados por interesses comuns (por exemplo, física, química) entre usuários e são utilizados para guiar as conexões entre os *peers* pertencentes ao um mesmo aglomerado de *peers*. Com isso, mensagens são roteadas de maneira inteligente, ou seja, elas são direcionadas para os *peers* que possuem os mesmos interesses, diferente das redes P2P tradicionais, que necessitam de um índice centralizados para manter o registro de grupos ou comunidades e os *peers* associados a esses.

Tendo em vista que redes sociais P2P são formadas sobre uma infraestrutura P2P, essas podem atuar como nuvem computacional, onde cada *peer* compartilha parte de seus recursos computacionais e hospeda de forma colaborativa os dados compartilhados entre os *peers* com os mesmos interesses. Desde então, para que isso aconteça, é necessário responder alguns questionamentos: Como usuários com interesses comuns podem compartilhar recursos e dados em um ambiente distribuído descentralizado de larga escala? Como criar redes sociais a partir dos relacionamentos sociais dos usuários? Como os computadores dos usuários, participantes de uma partição de rede social, podem ser utilizados para armazenar colaborativamente os dados compartilhados? Como dados, mantidos sobre as várias partições de redes sociais no ambiente distribuído, podem ser recuperados de acordo com a necessidade do usuário?

Para responder os questionamentos apresentados acima e visando colaborar na solução para problemas de desempenho apresentados por Mattoso *et al.* (2009), este trabalho propõe um modelo arquitetural para o compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados em redes P2P

sociais. Muito se tem estudado sobre algoritmos para controle de topológico, descoberta de recursos e roteamento de mensagens em redes P2P sociais, no entanto, pouca atenção tem sido dada ao problema de compartilhamento, agregação e utilização de recursos computacionais em redes P2P sociais, que é o foco da arquitetura proposta por esse trabalho.

Para tanto, o restante deste artigo está dividido em 4 seções. A Seção 2 apresenta os problemas e trabalhos relacionados ao compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados em redes P2P sociais. A Seção 3 apresenta um cenário de uso, onde pesquisadores compartilham uns com os outros os recursos computacionais de armazenamento de seus computadores pessoais, a fim de que os dados compartilhados por cada um deles sejam hospedados de maneira colaborativa sobre os recursos computacionais compartilhados. A Seção 4 apresenta a proposta de um modelo arquitetural em camadas para o compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados em redes sociais. Por fim, a Seção 5 apresenta as conclusões deste trabalho, assim como a sua viabilidade de pesquisa e trabalhos futuros.

2. Trabalhos Relacionados

Esta seção discute os problemas associados ao compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados em redes P2P sociais.

2.1. Redes P2P Sociais

Redes P2P são redes de computadores onde todos os computadores têm as mesmas funcionalidades e papéis. Assim, não existe diferença entre clientes e servidores, pois um *peer* atua como cliente, quando ele requisita um recurso ou serviço compartilhado na rede, e como servidor, quando um de seus recursos ou serviços compartilhados é requisitado e utilizado por outros *peers*. Isso implica na distribuição de autoridades, não somente para prevenir a introdução de pontos de falha, mas também a possibilidade de assegurar um nível de autonomia para auto-organização, onde qualquer *peer* ou grupo de *peers* pode espontaneamente manter-se ou recuperar-se de falhas. Devido a essas características, soluções algorítmicas foram propostas a fim de construir infraestruturas de redes P2P. Segundo Semenov (2005), infraestruturas de redes P2P são divididas em duas categorias: redes P2P estruturadas (ex. Chord [Stoica *et al.* 2001], CAN [Ratnasamy *et al.* 2001], Pastry [Druschel e Rwostron 2001] e Tapestry [Zhao *et al.* 2001]) e não estruturadas (ex. Gnutella [Repeanu 2001] e FreeNet [Clarke *et al.* 2001]).

Apesar de existirem diversas soluções para estruturação de redes P2P, elas não levam em consideração as preferências de usuário. Com isso, algoritmos para construção de redes P2P sociais têm sido propostos a fim de criar redes sociais sobre infraestruturas de redes P2P já conhecidas [Liu, Antonopoulos e Mackin 2007; Lin *et al.* 2007; Carchiolo *et al.* 2008]. Tais trabalhos possuem uma característica em comum: eles propõem algoritmos que combinam os interesses dos *peers* a fim de selecionar aqueles com maior grau de similaridade possível, baseando-se em conjuntos de palavras chaves que, por sua vez, especificam os interesses dos *peers*.

2.2. Backup Colaborativo de Dados

Sistemas que armazenam dados de maneira confiável satisfazem um requisito fundamental: novas cópias de dados são criadas rapidamente à medida que as cópias já existentes do sistema são perdidas. Satisfazer esses requisitos é difícil em redes P2P tradicionais, pois *peers* falham frequentemente e criar novas réplicas de dados e transmiti-las sobre um conjunto de conexões com largura de banda limitada demanda tempo. A combinação de falhas frequentes em redes P2P e o longo tempo demandado para replicação de dados, perdidos durante as saídas dos *peers* de uma rede, podem tornar o armazenamento de grandes volumes de dados dispendioso ou até mesmo impossível em termos de desempenho.

Na vida real, pessoas tendem a comportar-se de maneira cooperativa, quando essas possuem interesses comuns e, por isso, elas tendem a criar relacionamentos entre si. Baseados nessa observação, Li e Dabek (2007) apresentaram uma forma de estruturar sistemas P2P para armazenamento de dados. A proposta consiste em conectar *peers* cujos donos possuem relacionamentos de amizade. A formação dessas redes sociais permite o aumento da oferta de recursos computacionais de armazenamento, conforme novos *peers* se relacionam, além de permitir uma maior disponibilidade dos dados, enquanto os relacionamentos são mantidos.

Uma vez que usuários de ambientes distribuídos baseados em redes sociais criam relacionamentos uns com os outros, eles podem compartilhar parte de seus recursos computacionais de armazenamento. A partir disso, os *peers* podem hospedar de maneira colaborativa os conteúdos compartilhados pelos usuários pertencentes à mesma rede social P2P. Isso deve assegurar a confiabilidade e a recuperação de dados, mesmo que alguns *peers* não estejam conectados todo o tempo. Para isso, duas técnicas de redundância de dados, tipicamente utilizadas em sistemas de armazenamento, são utilizadas: replicação e *erasure coding* [Li e Dabek 2007]. Em replicação, cópias exatas dos dados são distribuídas e armazenadas em *peers* distintos. Já em *erasure coding*, os dados são divididos em blocos e codificados, para que, em seguida, sejam distribuídos e armazenados em *peers* distintos. A partir de um conjunto de blocos codificados é possível reconstruir os dados originais. Estudos comparativos têm sido feitos a fim de identificar situações em que essas técnicas escalam melhor. Rodrigues e Liskov (2005) focaram na comparação da replicação e *erasure coding* em *Distributed Hash Tables* (DHT), e concluíram que quando a disponibilidade dos *peers* é alta, replicação escala melhor, enquanto em cenários onde a disponibilidade dos *peers* é baixa, *erasure code* escala melhor. Watherspoon e Kubiawicz (2002) compararam quantitativamente as técnicas de redundância de dados em termos do consumo de banda e *overhead* de armazenamento. Eles concluíram que *erasure coding* consome menos recursos computacionais de armazenamento e banda que a replicação. Por fim, Oliveira et al. (2008) compararam as técnicas de redundância de dados em termos de tempo de recuperação dos dados. Eles concluíram que, para *erasure coding*, o aumento do *overhead* de armazenamento diminui a velocidade de recuperação de dados.

Com base nos desafios de gerenciamento de experimentos científicos em larga escala, levantados por Mattoso *et al.* (2009), o backup colaborativo de dados é uma abordagem promissora, uma vez que pesquisadores cooperam entre si e usam computadores de maneira intensiva para compartilhar dados relacionados a seus experimentos científicos.

2.3. Controle de Réplicas

Um assunto importante em replicação de dados é o mecanismo de controle de réplicas. Gray et al. (1996) classifica os mecanismos de controle de réplicas de acordo com dois parâmetros: *onde* atualizar e *quando* atualizações são propagadas para todas as réplicas. De acordo com o primeiro parâmetro (*onde*), protocolos de replicação podem ser classificados como soluções único-mestre ou múltiplos mestres. Já de acordo com o segundo parâmetro (*quando*), estratégias de propagação de atualizações são divididas em abordagens síncronas e assíncronas. Os mecanismos de controle de réplicas também são afetados pelo modo como as réplicas são distribuídas sobre uma rede.

Para manter a disponibilidade de dados em uma rede P2P social, é necessário que réplicas dos dados compartilhados sejam mantidas localmente nos computadores dos usuários. Em ambientes distribuídos, onde usuários colaboram uns com os outros, compartilhando, acessando e atualizando dados, é necessário o uso de uma abordagem de replicação múltiplos mestres para assegurar a disponibilidade dos dados [Martins et al. 2006]. Dessa forma, para cenários com alta colaboração dos usuários, como em *e-science*, este trabalho adota replicação múltiplos mestres para definir onde os dados serão atualizados. Para definir quando os dados devem ser atualizados, este trabalho adota replicação otimista, pelo fato dessa abordagem permitir a atualização assíncrona das réplicas. Além disso, replicação assíncrona atende a maioria dos requisitos de ambientes distribuídos de larga escala que permitem o trabalho colaborativo entre usuários [Martins et al. 2006].

Várias soluções para controle de réplicas podem ser encontradas na bibliografia, tais como IceCube [Shapiro, Bhargavan e Khrishna 2004] e DSR [Martins et al. 2006]. Essas duas soluções são mais genéricas e flexíveis, o que as tornam adequadas para ambientes altamente distribuídos para apoiar colaboração de usuários. No entanto, IceCube é uma solução centralizada, o que vai contra os problemas levantados por Mattoso et al. (2009), no que diz respeito a soluções distribuídas para tratar o armazenamento de grandes volumes de dados. O DSR é uma solução próxima ao que se pretende alcançar a partir deste trabalho, pois ela se baseia em um subconjunto de *peers*, chamados reconciliadores, que são selecionados para concorrentemente reconciliar atualizações conflitantes.

3. Cenário de Uso

Segundo Mattoso et al. (2009), faltam técnicas de computação distribuída que escalem para configurações muito grandes, e, ao mesmo tempo, tratem da autonomia, dinâmica e heterogeneidade dos recursos em ambientes computacionais *e-science*. Com base nisso, um exemplo de cenário de uso da abordagem proposta nesse trabalho é apresentado na Figura 1. Nesse cenário, três pesquisadores (Carla, Marcos e Rita) necessitam compartilhar informações para realização de um experimento científico, que será realizado no centro de pesquisas onde Rita trabalha. Como o projeto é em conjunto com outros pesquisadores, Rita precisa compartilhar dados com os demais pesquisadores e vice-versa.

Conforme a Figura 1, os computadores pessoais de cada pesquisador possuem conexão com a Internet. Assim, a proposta desse trabalho é que cada computador pessoal se transforme em um *peer*, isto é, cada computador pode atuar como cliente ou

servidor ao mesmo tempo, iniciar uma conexão com outros computadores conectados a uma rede P2P a qualquer instante de tempo e, ainda, compartilhar os recursos computacionais e/ou conteúdos existentes em computadores pessoais. Isso deve permitir que recursos computacionais de armazenamento sejam agregados, para formar uma entidade computacional de armazenamento, que neste trabalho é chamada de Nuvem Computacional, a fim de que os dados compartilhados pelos pesquisadores sejam mantidos nessa entidade computacional.

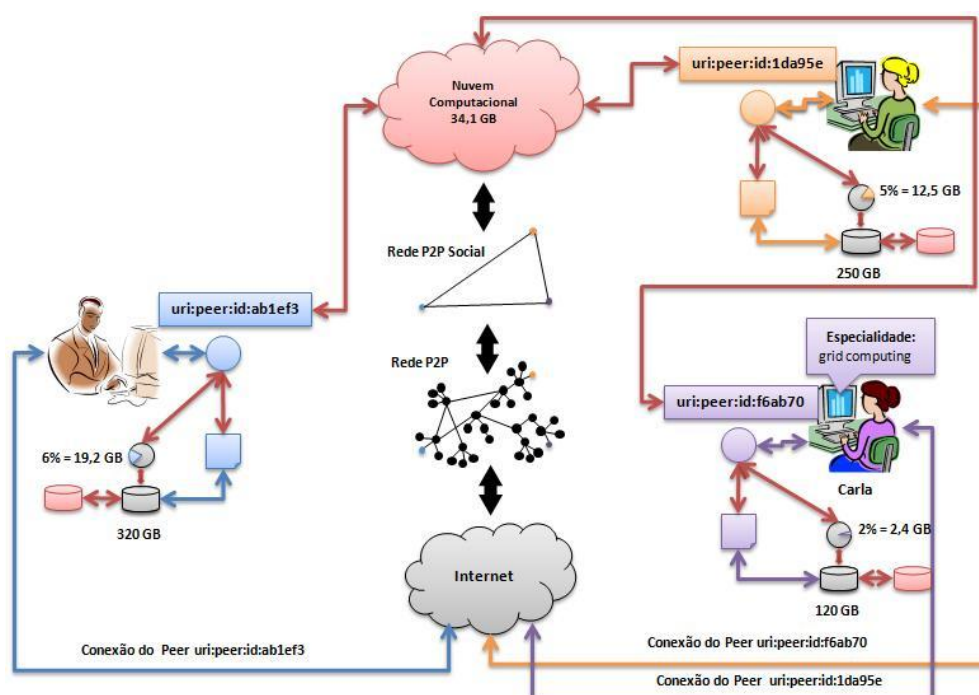


Figura 1. Cenário de uso para o compartilhamento de recursos computacionais e hospedagem de dados compartilhados em redes P2P sociais

Quando os computadores pessoais dos pesquisadores se conectam a uma rede P2P, eles recebem um identificador único, conforme exibido pela Figura 1. Esse identificador é construído com base no modelo URN (*Uniform Resource Name*), que é uma cadeia compacta de caracteres que é usada para denominar um recurso na Internet. Com isso, cada computador pessoal passa a fazer parte de uma infraestrutura descentralizada de sistema distribuído. Nela, cada pesquisador pode compartilhar livremente os conteúdos e recursos computacionais de armazenamento de seus computadores pessoais. Apesar de nesse cenário somente existirem três pesquisadores, uma rede P2P pode ser formada pelos computadores pessoais de outros usuários (note que a rede é representada como um grafo pela Figura 1).

Uma vez que seus computadores pessoais fazem parte da rede P2P, os pesquisadores podem compartilhar conteúdos e recursos computacionais de armazenamento. Assim, quando um pesquisador compartilha um conteúdo sobre a rede P2P, informações sobre o compartilhamento são disseminadas na rede. Nesse momento, as mensagens de publicação de conteúdo procuram *peers* cujo usuário tenha algum interesse sobre o seu conteúdo. Assim, da mesma forma que um usuário compartilha um conteúdo por meio de palavras-chave que classificam o item compartilhado, ele também pode explicitar o seu interesse por algum conteúdo, bastando informar as possíveis

palavras-chave relacionadas ao conteúdo idealizado. Com isso, à medida que mensagens com informações de compartilhamento passam pelos *peers*, elas são recebidas e, em seguida, analisadas para verificar se as informações de conteúdo compartilhado satisfazem os interesses do usuário representado pelo *peer*. Outro ponto importante sobre as mensagens de publicação de conteúdo é que elas, além de carregar informações dos conteúdos compartilhados pelos usuários, carregam informações sobre os interesses dos usuários que compartilham conteúdos. Essas informações são importantes, pois elas ajudam o mecanismo de descoberta e roteamento a tomarem decisões sobre qual o melhor caminho que deve ser seguido por uma mensagem na infraestrutura de rede P2P.

Tabela 1. Relacionamentos e compartilhamentos de recursos computacionais de armazenamento

| Pesquisador | Tamanho do Disco (GB) | Relacionamento entre Pesquisadores | Taxa de Compartilhamento (%) | Espaço Compartilhado (GB) |
|---------------|-----------------------|------------------------------------|------------------------------|---------------------------|
| Carla | 250 | Marcos | 2 | 5 |
| | | Rita | 3 | 7,5 |
| Marcos | 320 | Carla | 4 | 12,8 |
| | | Rita | 2 | 6,4 |
| Rita | 120 | Carla | 1 | 1,2 |
| | | Marcos | 1 | 1,2 |
| Totais | 690 | 6 | 13 | 34,1 |

Após cada pesquisador ter compartilhado conteúdos, especificado seus interesses e recebido as informações de compartilhamento de conteúdo uns dos outros, eles decidem formar uma rede social, como é exibido no cenário apresentado pela Figura 1. As redes sociais, formadas pelos usuários conectados ao sistema distribuído, são criadas a partir da adição de um usuário na lista de relacionamentos de outro usuário, formando pequenos grafos sobre a infraestrutura de rede P2P. A criação do relacionamento depende somente do usuário, ou seja, ele decide se deseja manter algum relacionamento com os demais usuários do sistema. É importante lembrar que essa decisão, por parte do usuário, é tomada com base em alguma afinidade que um usuário tenha com o outro. Assim, no cenário apresentado nesta seção, têm-se três pesquisadores que fazem parte de um projeto de bioinformática e que desejam acompanhar e estudar os dados de um experimento científico, que será realizado por um deles. Quando os relacionamentos são criados entre os pesquisadores, eles disponibilizam os conteúdos que estão compartilhando e, também, um percentual de recurso computacional de armazenamento que desejam compartilhar. Os relacionamentos entre os pesquisadores e os compartilhamentos são exibidos pela Tabela 1.

Após os relacionamentos e compartilhamentos terem sido efetivados pelos pesquisadores, um pequeno aglomerado de recursos computacionais de armazenamento é formado, totalizando um espaço compartilhado de 34,1 GB. Esse aglomerado de recursos denota uma nuvem computacional, que neste trabalho é uma entidade de alto nível, atuando como um agrupamento de recursos computacionais oriundos dos relacionamentos sociais entre usuários do ambiente. Na visão dos pesquisadores, as nuvens são vistas de maneira transparente, ou seja, eles não precisam especificar qualquer tipo de endereço para acessar uma nuvem, pois ela é uma visão nomeada dos relacionamentos entre um pesquisador e os outros e, vice-versa. Na Figura 2, são exibidas as diferentes visões da nuvem por parte dos pesquisadores deste cenário de

uso. As visões da nuvem auxiliam o usuário a visualizar seus relacionamentos sociais, além de possibilitar uma visão única do espaço de armazenamento criado de maneira colaborativa. Assim, cada pesquisador pode ter acesso aos conteúdos compartilhados na nuvem. Além disso, quando um conteúdo é compartilhado com um ou mais usuários, esses se tornam também proprietários do conteúdo e passam a hospedar o mesmo.

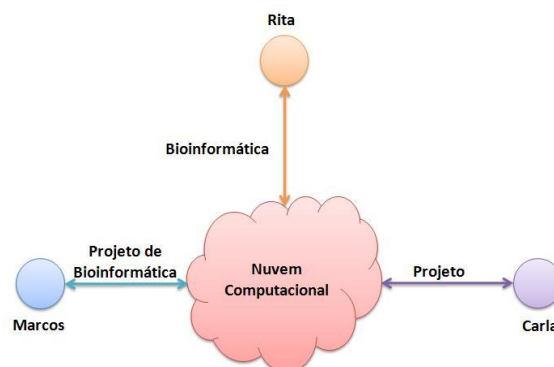


Figura 2. Nuvem computacional e as visões nomeadas dos relacionamentos, segundo a perspectiva de cada pesquisador

Por fim, à medida que novos usuários ingressam em nuvens computacionais, por meio dos relacionamentos sociais com os usuários dessas entidades computacionais, a disponibilidade dos conteúdos compartilhados mais importantes aumenta, pois, como mencionado anteriormente, cada usuário compartilha parte dos seus recursos computacionais para auxiliar a hospedagem de conteúdos. Com isso, a escala da nuvem cresce conforme o número de relacionamentos sociais entre os usuários da nuvem, ou seja, à medida que os relacionamentos são criados, quantidades de recursos computacionais de armazenamento são ofertadas para uso na nuvem. Outra forma de aumentar a escala da nuvem é por meio do aumento do tamanho do compartilhamento dos recursos computacionais de armazenamento.

4. Arquitetura Proposta

Esta seção apresenta a proposta de um modelo arquitetural para o compartilhamento de recursos computacionais e hospedagem colaborativa de dados compartilhados em redes P2P sociais. O modelo arquitetural proposto

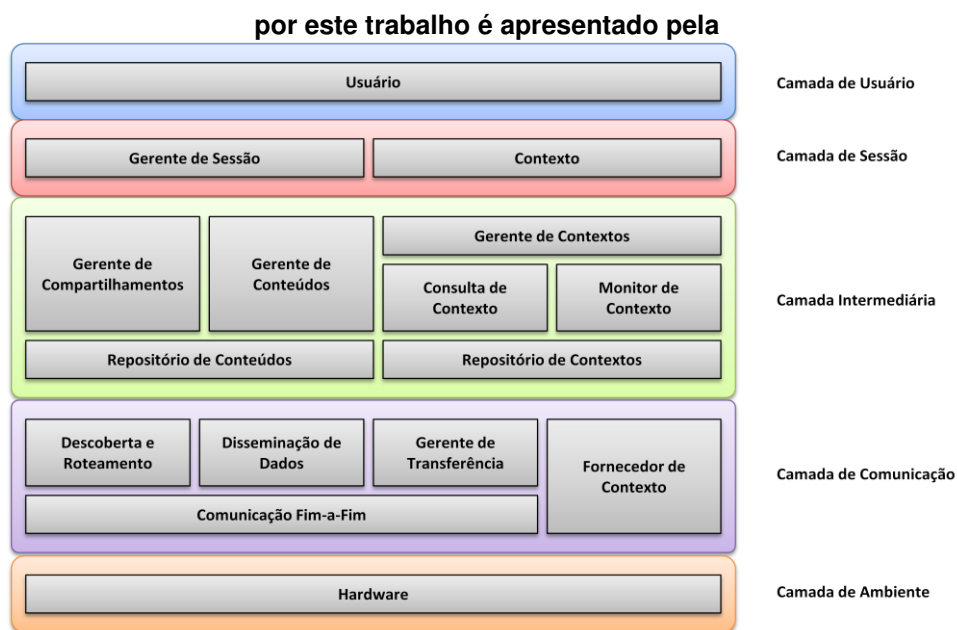


Figura 3.

Para que o modelo arquitetural possa ser compreendido, as seções seguintes estão organizadas com objetivo de explicar cada camada, iniciando pela camada de ambiente, passando pelas camadas de comunicação, intermediária e sessão e, por fim, a camada de usuário.

4.1. Camada de Ambiente

Esta camada contém uma coleção de componentes, representados pelo componente hardware, que devem fornecer interfaces para controlar recursos computacionais, tais como: discos, adaptadores de redes, memórias, processadores entre outros. Além disso, a camada também deve conter componentes que encapsulam APIs de sistema operacional, ou seja, componentes com interfaces bem definidas para esconder detalhes de gerenciamento de arquivos, gerenciamento de memória, protocolos de rede, comunicação de rede e gerenciamento de processos.

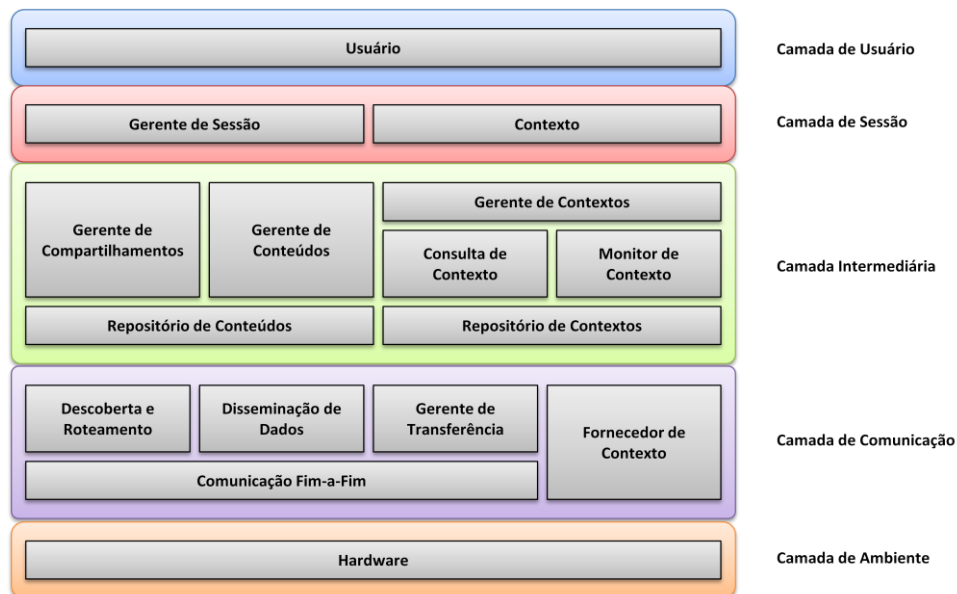


Figura 3. Modelo arquitetural para compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados compartilhados em redes P2P sociais.

Os componentes das camadas acima devem ser associados aos componentes desta camada, a fim de obter informações de hardware como, por exemplo, espaço livre em disco, capacidade de disco, quantidades de discos, endereços IP, velocidade de transmissão e quantidade de adaptadores de redes. Deste modo, esta camada permite que a arquitetura de software seja independente de hardware. Assim, componentes de instrumentação de hardware podem ser desenvolvidos para sistemas operacionais específicos.

4.2. Camada de Comunicação

A camada de comunicação é composta por mecanismos para comunicação fim-a-fim, descoberta de nós, roteamento de mensagens, disseminação de informação, gerenciamento de transferência de dados e fornecimento de contexto. Todos os componentes da camada devem utilizar o componente de Comunicação Fim-a-Fim, pois ele deve ser responsável por enviar e receber mensagens dos *peers* conectados na rede P2P. Portanto, o componente de Comunicação Fim-a-Fim deve encapsular as primitivas de envio e recebimento de mensagens do protocolo P2P utilizado pela arquitetura, por exemplo, Chord [Stoica *et al.* 2001], CAN [Ratnasamy *et al.* 2001], Pastry [Druschel e Rwostron 2001], Tapestry [Zhao *et al.* 2001], Gnutella [Repeanu 2001] entre outros.

Os mecanismos de descoberta e roteamento são representados pelo componente de Descoberta e Roteamento. Assim, quando *peers*, recursos e dados são descobertos na rede P2P, o componente de Descoberta e Roteamento armazena referências para eles, permitindo que uma topologia de rede seja formada à medida que os *peers* se conectam uns aos outros. Dessa forma, uma rede de cobertura é criada e, quando um *peer* envia mensagens para outro *peer*, essas são roteadas até que elas cheguem aos seus destinos.

O mecanismo para disseminação de informações sobre *peers* e compartilhamentos, tanto de recursos computacionais como de dados, é representado pelo componente de Disseminação de Informação. Para isso, técnicas de disseminação de informação, como *Push*, *Pull* e *Push and Pull*, podem ser utilizadas para publicar informações de *peers* e compartilhamentos na rede P2P. O componente de Disseminação de Informação faz uso das referências para os *peers* e para recursos e dados compartilhados, que são mantidos pelo componente de Descoberta e Roteamento.

Outro componente que utiliza as referências mantidas pelo mecanismo de Descoberta e Roteamento é a Gerência de Replicação. Esse componente é responsável pelo envio e recepção de réplicas de dados compartilhados entre usuários. Além disso, o componente de Gerência de Replicação também controla a atualização de réplicas de dados, a fim de que os dados compartilhados entre os usuários mantenham-se atualizados, após alguma alteração realizada por parte de um ou mais usuários.

Por fim, o componente Fornecedor de Contexto é responsável pela captura de informações de contextos e por direcioná-las para a camada acima, que é a camada intermediária. Entende-se como contexto qualquer informação que possa ser usada para caracterizar a situação de uma entidade, que pode ser uma pessoa, lugar ou objeto que é considerado importante para a interação entre os usuários do sistema [El-Kahtib et al. 2004].

4.3. Camada Intermediária

Em um sistema distribuído, a computação acontece a partir da colaboração de nós que compõem esse sistema. Para isso, é necessário que componentes e operações de alto nível sejam bem definidos, a fim de que detalhes de computação distribuída não sejam revelados. Desde então, a camada intermediária responsabiliza-se pela gerência e monitoramento de recursos computacionais, conteúdos e contextos. Portanto, a camada é composta pelo Gerente de Compartilhamentos, Gerente de Conteúdos, Repositório de Conteúdos, Gerente de Contexto, Consulta de Contexto, Monitor de Contexto e Repositório de Contextos.

O Gerente de Contexto é um componente para recuperação e atualização de contextos mantidos pelo Repositório de Contextos. A atualização de contexto pode ser feita pelo Monitor de Contexto e Consulta de Contexto. O Monitor de Contexto atualiza o Repositório de Contexto com informações locais de contexto como, por exemplo, quantidade de unidades de discos, quantidade de espaço livre em unidades de discos, relacionamentos de usuários. Já a Consulta de Contexto atualiza o Repositório de Contexto com informações coletadas pelas consultas remotas, que são feitas sobre a rede P2P. Conteúdos que combinam com os interesses do usuário local e usuários com um determinado grau de semelhança com o usuário local são exemplos de consultas que atualizam o Repositório de Contexto.

O gerenciamento dos conteúdos compartilhados é feito utilizando um conjunto de componentes, que é composto pelo Gerente de Compartilhamentos, Gerente de Conteúdos e Repositório de Conteúdos. A função do Gerente de Compartilhamento é manter os compartilhamentos de conteúdos realizados pelo usuário. Quando um compartilhamento é criado, o conteúdo é mantido pelo Repositório de Conteúdos. As funcionalidades para acesso e manipulação de conteúdos compartilhados são disponibilizadas pelo Gerente de Conteúdos. Esse componente acessa e manipula os

conteúdos mantidos no Repositório de Conteúdos. As manipulações de conteúdos compartilhados podem causar alguma alteração do mesmo. Por isso, qualquer alteração em conteúdo compartilhado reflete na execução de operações da camada abaixo da camada intermediária.

4.4. Camada de Sessão

Esta camada manipula informações sobre as interações entre o usuário e o ambiente distribuído. Para tanto, a camada de sessão envia e recupera tais informações para a camada intermediária, que por sua vez, as armazenam na forma de contextos.

Quando o usuário interage com o ambiente distribuído, uma sessão é aberta, a fim de que informações sobre a interação sejam mantidas, enquanto o usuário interage com algum elemento presente no sistema distribuído. Usuários e conteúdos compartilhados são exemplos de elementos presentes no sistema distribuído.

4.5. Camada de Usuário

A camada de usuário manipula a identificação e perfil do usuário. Assim, essa camada é responsável em capturar contextos do usuário como, por exemplo, informações pessoais e interesses pessoais sobre tipos de conteúdos. Esses dados são importantes para interação do usuário com o ambiente distribuído. Assim, interesses do usuário por determinados tipos de conteúdos são utilizados para buscar conteúdos compartilhados e informações sobre usuários do ambiente distribuído. Por fim, informações pessoais são utilizadas para identificar o usuário e, por isso, elas podem ser utilizadas para autenticação de usuários.

5. Conclusão

As tecnologias em ambientes distribuídas têm impulsionado a ciência em larga escala [Mattoso et al. 2009]. Com o objetivo de colaborar com propostas de soluções para a manipulação de dados em ambientes distribuídos, este trabalho propõe um modelo arquitetural para o compartilhamento de recursos computacionais de armazenamento e hospedagem colaborativa de dados compartilhados em redes P2P sociais. Deste modo, recursos computacionais de armazenamento podem ser agregados em larga escala e, como consequência disso, grandes volumes de dados compartilhados podem ser armazenados sobre esses recursos.

Atualmente, estamos trabalhando em um simulador para realizar experimentos de escalabilidade da arquitetura proposta. Como trabalhos futuros, estão a construção de um *middleware* para gerenciamento de armazenamento de dados em redes P2P sociais. Isso permitirá que os recursos computacionais de armazenamento em ambientes de *e-science* possam ser compartilhados, a fim de armazenar grandes volumes de dados, oriundos de experimentos científicos. Outra direção deste trabalho é a formalização das políticas para compartilhamento, agregação e uso de recursos computacionais de armazenamento e dados em redes P2P sociais, que são formadas a partir dos relacionamentos entre usuários com interesses em comum. Essas políticas diferem de propostas como [Andrade, Santos-Neto e Brasileiro 2008; Zhou e Lo 2004; Figueiredo et al. 2008], pois elas não dependem de infraestruturas específicas de redes P2P, ou seja, elas poderão ser aplicadas sobre redes P2P sociais, que podem ser construídas sobre redes P2P estruturadas ou não estruturadas.

6. Referências

- Carchiolo, V.; Malgeri, M; Mangioni, G.; Nicosia, V. (2008). Emerging structures of the P2P networks induced by social relationships, *Computer Communications*, v 31, pp 620-628.
- Clarke, I.; Sandberg, O.; Wiley, B.; Hong, T. W. (2001). Freenet: a distributed anonymous information storage and retrieval system. LNCS, v. 2009, Heidelberg, Germany, pp 46-66.
- Druschel, P; Rwostron, A. (2001). Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: *IFIP/ACM International Conference on Distributed Systems Platforms*, Heidelberg, Germany.
- El-Khatib, K.; Zhang, Z. E.; Hadibi, N.; e van Bochman, G. (2004). Personal and Mobility in Ubiquitous Computing Environment. *Wireless Communications and Mobile Computing*, v. 4, pp 595-607.
- Gray, J.; Helland, P.; O'Neil, P., Shasha, D. (1996). The dangers of the replication solution. In: *SIGMOD*, Montreal, Canada, pp 173-182.
- Li, J.; Dabek, F. (2006). F2F: reliable storage in open networks. In: *International Workshop on Peer-to-Peer Systems*, NY, USA.
- Lin, C. J.; Chang, Y. T.; Tsai, S. C.; Chou, C. F. D. (2007). Distributed social-based overlay adaptation for unstructured P2P networks. In: *IEEE Global Internet Symposium*. Anchorage, AK, pp 1-6.
- Liu, L.; Antonopoulos, N.; Mackin, S. (2007). Social Peer-to-Peer for resource discovery. In: *EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing*, Napoli, Italy, pp 459-466.
- Martins, V; Akbarinia, R; Pacitti, E; Valduriez, P. (2006). Reconciliation in the APPA P2P system. In: *International Conference on Parallel and Distributed Systems*, Minneapolis, MM.
- Mattoso, M.; Werner, C.; Travassos, G.; Braganholo, V.; Murta, L.; Ogasawara, E.; Oliveira, F.; Martinho, W. (2009). Desafios no apoio à composição de experimentos científicos em larga escala. In: *SEMISH*, Bento Gonçalves, RS, Brazil, 2009.
- Milojicic, D. S.; Kalogeraki, V.; Lukose, R.; Nagaraja, K.; Pruyne, J.; Richard, B.; Rollins, S; e Xu, Z. (2002). Peer-to-Peer Computing. HP, HP Laboratories Palo Alto, Technical Report HPL-2002-57.
- Oliveira, M. Y., Cirne, W., Brasileiro, F. e Guerrero, D. (2008). On the Impact of the Data Redundancy Strategy on the Recoverability of Friend-to-Friend Backup Systems. In: *SBRC*, Rio de Janeiro, Brasil.
- Ratnasamy, S.; Francis, P.; Handley, M.; Karp, R; Shenker, S. (2001). A scalable content addressable network. . In: *Conference on Applications, technologies, architectures, and protocols for computer communications*, San Diego, California, pp 161-172.
- Repeanu, M. (2001). Peer-to-peer architecture case study: gnutella network. In: *International Conference on Peer-to-Peer Computing*, Linkoping, pp 99-100.

- Rodriguez, P.; Liskov, B. (2005). High availability in DHTs: erasure coding vs. replication. In: International Workshop on P2P Systems, Ithaca, NY.
- Semenov, A. (2005). Evolution of Peer-to-peer algorithms: Past, present and future. Seminar on Internetworking, HUT T-110.551, Helsinki University of Technology, 2005. Disponível em: <http://www.tml.tkk.fi/Publications/C/18/semenov.pdf>. Acessado em: 08 de junho de 2009.
- Shapiro, M.; Bhargavan, H.; Krishna, N. (2004). A constraint-based formalism for consistency in replicated systems. In: International Conference on Principles of Distributed Systems, Grenoble, France.
- Stoica, I.; Morris, R.; Karger, D.; Kaashoek, M. F; e Balakrishnan, H. (2001). Chord: a scalable peer-to-peer lookup service for internet applications. In: Conference on Applications, technologies, architectures, and protocols for computer communications, San Diego, California, pp 149-160.
- Weatherspoon, H.; Kubiatowicz, J.D. (2002) Erasure coding vs. replication: A quantitative comparison. In: International Workshop on Peer-to-Peer Systems , Cambridge, Massachusetts, United States.
- Zhao, B Y.; Kubiatowicz, J; Joseph, A. (2001). Tapestry: an infrastructure for fault-tolerant wide-area location and routing, Tech. Rep. UCB/CSD-01-1141, University of California at Berkeley, Computer Science Department.
- Andrade, N.; Santos-Neto, E.; Brasileiro, F. (2008). Scalable resource annotation in peer-to-peer grids. In: International Conference on Peer-to-Peer Computing. Aachen, Germany, pp 231-234.
- Zhou, D.; Lo, V. (2004). Clustering on the fly: resource discovery in a cycle sharing peer-to-peer systems. In: IEEE International Symposium on Cluster Computing and the Grid. Chicago, USA, pp 66-73.
- Figueiredo, R. J.; Boykin, P. O.; Juste, P. S.; Wolinsky, D. (2008). Facilitating the deployment of ad-hoc virtual organizations with integrated social and overlay networks. In: HPDC, Boston, USA, pp 201-204.