

Cloud Mista

Douglas Siqueira e Naara Macedo

*Universidade Federal do Piauí

I. PROPOSTA

Este trabalho tem como objetivo o desenvolvimento de uma nuvem de armazenamento em redes P2P (*peer-to-peer*). Para utilização da aplicação pelo usuário, será necessário a realização de um login de cadastro no servidor (como na Figura 1 (a)). A partir daí, o servidor autoriza o cliente a se tornar um nó na rede (como na Figura 1 (b)), o classificando, também, como um servidor de armazenamento. Este terá acesso aos demais nós da rede para armazenar e fazer download do conteúdo.

Para a realização do trabalho foram selecionados alguns requisitos obrigatórios. Nos quais, seis destes serão atendidos:

- **Alta Disponibilidade**, por ter uma quantidade considerável de nós com uma réplica da informação armazenada.
- **Transparência de Replicação**, por conta da informação ser distribuída para vários servidores.
- **Transparência de falha**, porque mesmo que um servidor esteja indisponível, como a informação foi replicada, não será perceptível ao cliente.
- **Transparência de concorrência**, pois dois computadores poderão requisitar no mesmo servidor sem que haja conflito entre as operações.
- **Segurança**, pelo fato de o usuário só poder participar da rede com a permissão do servidor. Além da modificação de dados pessoais, que o acesso será restringido apenas ao computador que o enviou.
- **Escalabilidade**, pois um novo nó poderá participar da rede sem que haja modificações no algoritmo.

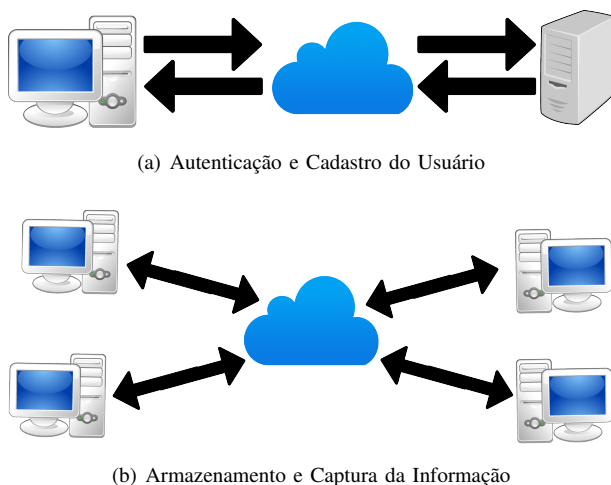


Figura 1. Comparação das Arquiteturas variando a Carga de Trabalho

II. TECNOLOGIAS

Um dos propósitos do Broker de mensagem é manipular mensagens recebidas de aplicativos e executar alguma ação nelas. Essas ações podem, por exemplo, ser roteadas para um ou vários destinos, fornecer roteamento de mensagens dependendo das chaves de roteamento, transformar mensagens em uma representação alternativa ou responder a eventos ou erros. Por esta razão, a tecnologia de Broker de Mensagem será usado por este trabalho.

Dentre os Brokers existentes no mercado, este trabalho fará uso do RabbitMQ para que seja feito a distribuição dos arquivos aos nós conectados. A escolha do RabbitMQ é dada por ser uma plataforma *open-source*, de uso fácil, documentação bem estruturada e melhor para trabalhar com threads.

A plataforma do RabbitMQ necessariamente precisa de um protocolo para comunicação entre nós. O protocolo usado pelo RabbitMQ, por padrão é o AMQP (*Advanced Message Queuing Protocol*) que tem por objetivo a interoperabilidade entre os sistemas, de forma a unificar e centralizar os protocolos antes existentes. Além disso, tem como um dos objetivos, a possibilidade de troca de informação entre produtos (sistemas ou dispositivos) heterogêneos.

Mesmo com estas vantagens o AMQP e o RabbitMQ contém a limitação de apenas funcionarem em rede local. Dessa forma, será usado o CloudAMQP, que são servidores RabbitMQ gerenciados nas filas de mensagens hospedadas em nuvem que permitem a transmissão de mensagens entre processos e outros sistemas. As mensagens são publicadas em uma fila por um produtor, os consumidores podem tirar as mensagens da fila quando o consumidor quiser manipular as mensagens.

Por fim, de uso das tecnologias será feito o uso do RPC (*Remote Procedure Call*) é um modelo de comunicação que opera intermediando, comumente, operações entre cliente-servidor em um serviço de rede. O seu funcionamento é iniciado a partir de uma chamada do cliente ao servidor. O servidor recebe a chamada a executa e responde ao cliente, o que termina a execução do RPC. Sendo este funcionamento o ideal para o que é pretendido alcançar no atual projeto.

III. DESAFIOS

Por conta do uso das tecnologias anteriormente citadas e o modelo de comunicação abordado, há desafios para o desenvolvimento dessa aplicação. Dentre os desafios, podem ser destacados: o envio de arquivos pela internet e segurança dos dados armazenados.

Por se tratar de uma *cloud* de armazenamento, este trabalho tem por obrigatoriedade enviar arquivos pela rede. No entanto,

nem todas as redes suportam o tráfego de uma determinada quantidade de dados. Sendo assim, é recomendável o uso desta aplicação em redes com pelo menos 10 Mbs (*Megabits*) e transmissão dados por no máximo 5 MB (*Mega Bytes*).

Em relação a segurança, são destacados duas dificuldades. A primeira é a modificação da mensagem de maneira a não ser compreendida pelo computador servidor. A segunda é ocultar a mensagem com o intuito de não ser visualizada pelos servidores de armazenamento.

IV. APLICAÇÃO

Este sistema esta dividido em dois o primeira foca em ser um servidor de autenticação de informações. A segunda é a realiza o envio e recepção das informações dos usuários.

No servidor de autenticação contém inicialmente uma conexão com o **CloudAMQP**. Após isso, é aberto o arquivo com nome cadastro.txt, que manterá os cadastros dos usuários.

A segunda aplicação é a utilizada pelos usuários. Nesta contém interfaces gráficas para fácil manipulação do usuários. As interfaces estão divididas em Autenticação, Cadastro, Janela Principal e Envio, como pode ser visto na Figura 2.

Na interface de cadastro o usuário insere informações básicas como, nome, senha e a confirmação da senha para que possa se autenticar no projeto. Após autenticado, o usuário pode utilizar o sistema para envio e recepção das informações.

Nesta parte o usuário terá acesso há pasta *Serverpeer*, *Clientpeer*, *todownload.txt* e *send.txt*. Estas pastas tem funcionalidades de configuração dos sistemas para o seu melhor funcionamento.

Na pasta *Senderpeer* será salvo os arquivos recebidos pelos usuários. Na pasta *Receiverpeer* será salvo os arquivos enviados pelos usuários.

V. IMPLEMENTAÇÕES ACRESCENTADAS AO PROJETO

O presente trabalho tem como objetivo desenvolver uma aplicação para armazenamento de informações em computadores disponíveis em rede mundial. Para este trabalho foi definido 6 requisitos de 15, que são considerados positivos na área de Sistemas. Distribuídos. Os 6 requisitos estipulados foram Alta Disponibilidade, Transparência de Replicação, Transparência de Falha, Transparência de Concorrência, Segurança e Escalabilidade.

Além dos requisitos estipulados satisfazem conceitos da área de Sistemas Distribuídos. Mas, além destes conceitos, outros foram percebidos e implementados no decorrer do desenvolvimento, como a **Transparência de Localização**, **Interoperabilidade** e **Transparência de Acesso**.

A **Transparência de Localização** neste trabalho esta adequada com o fato do cliente não precisar saber onde esta a informação. A **Interoperabilidade** está na localização dos cliente P2P. A **Transparência de Acesso** é implementada de forma ao usuário não saber, como este recurso será acessado em outro computador, ex: ocultação do arquivo.

(a) Autenticação

(b) Cadastro do Usuário

(c) Janela Principal

(d) Envio do Arquivo

Figura 2. Interfaces do Sistema