

Computação distribuída: Desafios do uso do Dropbox como suporte ao espaço de tuplas*

Lucas Eduardo Bretana, Alana Schwendler, Gerson Geraldo H. Cavalheiro

¹Laboratory of Ubiquitous and Parallel Systems – UFPEL
Pelotas, RS - Brasil

{lebretana, aschwendler, gerson.cavalheiro}
@inf.ufpel.edu.br

***Resumo.** A biblioteca ILUCTUS é uma alternativa para o processamento largamente distribuído. Sua solução explora um sistema de arquivos provido em uma nuvem computacional e oferece uma interface de acesso baseada no sistema de Espaço de Tuplas. Neste artigo, questões envolvendo a implementação de ILUCTUS sobre o Dropbox são apresentadas, bem como os tratamentos realizados na biblioteca.*

1. Introdução

As tecnologias de comunicação promovidas pela Internet fomentaram a criação de novos mecanismos de colaboração entre as pessoas. Podemos destacar a grande disponibilidade de serviços de armazenamento em nuvem como um estímulo ao desenvolvimento de projetos para colaboração distribuídas. Com base nisso se deu o desenvolvimento da biblioteca ILUCTUS [Bretana et al. 2017]. A biblioteca apresenta a implementação de uma abstração de comunicação para aplicações distribuídas bem como um modelo de negócios para o desenvolvimento de **Projetos Colaborativos**.

A biblioteca desenvolvida se propõe a servir de meio de compartilhamento do processo de criação do conhecimento. Sendo assim, ILUCTUS permite o compartilhamento de bases de dados entre os participantes do projeto, denominados **Colaboradores**. Além das ferramentas de compartilhamento são também apresentadas primitivas para a manipulação destes dados em um ambiente de endereçamento compartilhado. Para isso a ILUCTUS utiliza do apoio de nuvens computacionais e uma API de comunicação com esta nuvem. Dentre as nuvens que atendiam aos requisitos para o desenvolvimento da biblioteca se optou pela nuvem do Dropbox [Drago et al. 2012] pois, na época da implementação, apresentava melhor documentação da API e um processo robusto de autenticação.

A adoção do Dropbox como suporte de infraestrutura de nuvem para o Espaço de Tuplas implica em decisões de implementações específicas para a ferramenta proposta. Cita-se ainda que existem diferenças entre a versão gratuita, que foi utilizada na presente implementação, da paga. Um destes limites está relacionado ao limite de armazenamento, sendo de 16 GiB na versão não paga e variável nas versões pagas, conforme o plano contratado. O uso de uma API ainda em desenvolvimento também traz alguns desafios

*O presente trabalho foi realizado com apoio do Programa Nacional de Cooperação Acadêmica da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES/Brasil.

que devem ser tratados pelas políticas de funcionamento da biblioteca. Neste trabalho são apresentadas as limitações encontradas no desenvolvimento da ILUCTUS e como estas foram tratadas.

2. Arquitetura da Biblioteca

O desenvolvimento da biblioteca ILUCTUS considera um modelo de negócio onde são desenvolvidos **Projetos Colaborativos**. A proposta deste modelo é ilustrada na Figura 1. A instanciação de um novo projeto é realizada por um **Administrador**, com a criação uma base de dados inicial, com os dados brutos a serem evoluídos. Em seguida, um **Colaborador** interessado no projeto requisita acesso ao projeto para o Administrador, que concede o acesso, fazendo o compartilhamento da base de dados com o Colaborador. Após o acesso ser concedido, a base de dados é copiada para o ambiente de nuvem do Colaborador. A evolução dos dados agora se dá pela execução, por parte do Colaborador, de uma aplicação que é dada pelo Administrador. Esta aplicação faz uso da ILUCTUS para realizar a evolução dos dados na nuvem deste Colaborador, segundo as heurísticas implementadas nesta aplicação. As soluções obtidas pela aplicação são salvas na nuvem em uma nova base de dados até que o processamento chegue ao fim. Os resultados obtidos são então compartilhados com o Administrador. Por fim as cópias dos dados presentes na nuvem do Colaborador são apagadas, de forma a não onerar custos de armazenamento.

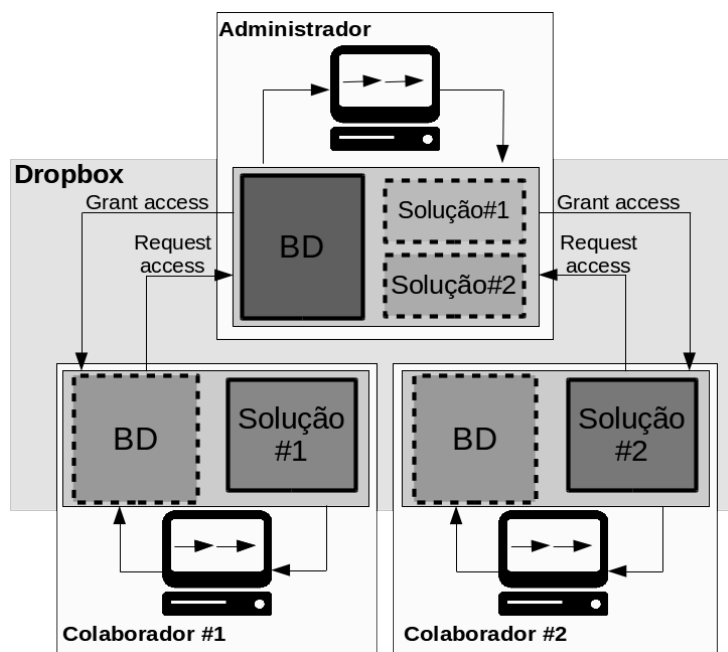


Figura 1. Ilustração da arquitetura da biblioteca.

O modelo de negócio proposto pela biblioteca ILUCTUS abstrai as dificuldades do uso da nuvem, independente da plataforma utilizada para implementação da biblioteca. As soluções desenvolvidas para execução dos processos citados na seção anterior foram tratadas caso-a-caso. Duas situações ocorreram nestes casos, uma onde os recursos da API permitiram realizar a implementação da solução, outra em que, por inexistência de suporte, as soluções foram propostas de forma externa à biblioteca. Limitações provenientes da arquitetura oferecida são tratadas conforme as indicações do suporte ou então modificando o desenho da biblioteca de forma a não inviabilizar o projeto.

3. Aspectos da Implementação

O desenvolvimento da ILUCTUS se deu utilizando a linguagem de coordenação Linda [Ahuja et al. 1986] e a nuvem do Dropbox como substrato para armazenamento de dados. Sendo assim, é importante destacar que, para o Dropbox, todo arquivo e diretório compartilhado irá onerar custos de armazenamento para cada um dos usuários que tem acesso. Além disso, internamente um arquivo compartilhado é tratado como pertencente a um mesmo *namespace*. Todos os demais arquivos não compartilhados de um usuário pertencem ao seu *namespace root*. A API do Dropbox se encontra ainda em desenvolvimento e por isso ainda não apresenta todas as funcionalidades esperadas pela comunidade de interesse. Tem-se ainda algumas restrições por parte da API que são reflexos das políticas de funcionamento e arquitetura do Dropbox. A seguir temos os principais desafios encontrados no desenvolvimento da biblioteca ILUCTUS.

Autenticação no Projeto Colaborativo: No Dropbox, o compartilhamento de um diretório requer o envio explícito da concessão de compartilhamento. Junto à esta concessão, segue as permissões de acesso concedidas. A API SDK Java do Dropbox não oferece primitivas de requisição/concessão de acesso a diretórios, embora seja de desejo da comunidade de desenvolvedores sua existência. Na atual implementação da ILUCTUS, estas operações são realizadas de forma externa à aplicação. Para fazer autenticação em um **Projeto Colaborativo** é necessária a troca de *tokens* de identificação entre **Administrador** e **Colaborador**. Da parte do Administrador, esses *tokens*, ou chaves, identificam o projeto e autorizam uma aplicação a ter acesso aos dados compartilhados na nuvem. *Tokens* são únicos para cada Colaborador em um Projeto Colaborativo. Pelo lado do Colaborador, o *token* identifica-o dentro do projeto, oferecendo, desta forma, acesso aos seus resultados na nuvem. Nesta troca inicial de informações entre o Administrador e Colaborador, além dos *tokens*, também é feita a troca de quaisquer outras informações necessárias para dar início ao processamento dos dados, tais como identificação da base de dados primária, lista de bases já processadas por Colaboradores pré-existentes, permissões sobre estas bases etc.

Compartilhamento de um mesmo arquivo/diretório: as políticas de compartilhamento não preveem um limite de usuários com acesso ao mesmo arquivo. Porém, como forma de segurança contra abuso do serviço oferecido, na versão não paga do Dropbox, é limitado o número de compartilhamentos que podem ser realizados em um período de 24 horas. Neste caso a API responde com uma exceção única e, então, o usuário é informado que deve fazer nova tentativa a posteriori. A documentação disponível não apresenta o número de compartilhamentos permitidos na versão gratuita.

Número excessivo de requisições: as requisições são feitas em relação a um *namespace*. Essas requisições são tratadas internamente utilizando *locks* por *namespace*, i.e., cada operação deve obter o *lock* correspondente, fazer sua tarefa e devolver o *lock*. Sendo assim, é de se esperar que requisições provenientes de diferentes processamentos colidam e gerem o erro de *too many requests* ou *too many write operations*. A solução dada pela API é manter a natureza síncrona da operação, resubmetendo a requisição após um tempo de espera (*back off time*). Este tempo de espera é determinado em função da carga à qual a nuvem (Dropbox) está submetida e é informado também pela própria API. Em caso de nova colisão, o processo é repetido

Limite de colaboradores: o desenvolvimento de **Projetos Colaborativos** en-

tende que um maior número de **Colaboradores** representa uma maior evolução dos dados, seja de forma mais rápida ou de maneiras mais diversas. No entanto, no Dropbox, o número de usuários compartilhando uma aplicação é limitado em 10. É viável, mesmo na versão não paga, estender o número de colaboradores até 500, considerando a aprovação pela Dropbox, Inc. de uma solicitação expressa pelo proprietário dos dados.

Abordagem da primitiva Eval: A nuvem do Dropbox funciona em um modelo de PaaS oferecendo o serviço de armazenamento de dados. Para a implementação canônica da primitiva **Eval** é necessário que a nuvem ofereça algum recurso de processamento de dados. Portanto esta primitiva teve sua semântica modificada para possibilitar sua existência. Na biblioteca ILUCTUS o funcionamento da **Eval** funciona como um agendamento de tarefa, onde é colocada uma função com parâmetros de entrada que será computada futuramente e seu resultado será escrito novamente na nuvem. Esta computação será realizada, oportunamente, por algum dos processos colaboradores envolvidos.

4. Conclusões e discussão

O emprego de serviços gratuitos de nuvem na aplicação é um método positivo para a biblioteca pois facilita a participação de vários **Colaboradores** ao se conectarem a um mesmo **Projeto Colaborativo**. Contudo, o uso das ferramentas gratuitas faz com que existam também algumas limitações quanto a plataforma e sua API. Estas limitações devem ser resolvidas, o quanto for possível, na implementação da biblioteca ou então se estender como uma limitação do funcionamento da própria ferramenta. Ainda assim, é possível usufruir das funcionalidades que a ILUCTUS oferece, utilizando-a como apoio para o desenvolvimento de estudos e pesquisas. ILUCTUS já foi atestada em quesitos de desempenho e funcionamento [Schwendler et al. 2017].

Em trabalhos futuros pretende-se dar manutenção ao nosso software, tornando-o compatível com novas versões de API do Dropbox e possibilitando o gerenciamento do compartilhamento de arquivos de dentro da biblioteca. O processo de requisição e concessão de compartilhamento de dados no Dropbox atualmente é manual, contudo é um desejo da comunidade usuária da API Dropbox que este processo esteja disponível por meio de requisições à API. Para a ILUCTUS isso possibilitaria que a autenticação em Projetos Colaborativos possa ser feito internamente à biblioteca, tornando a autenticação mais robusta e segura. Também serão projetadas novas implementações das primitivas de manipulação da biblioteca, bem como, realização de novos testes para aferir a ILUCTUS.

Referências

- Ahuja, S., Curriero, N., and Gelernter, D. (1986). Linda and friends. *Computer;(United States)*, 19(8).
- Bretana, L. E., Schwendler, A., and Cavalheiro, G. G. H. (2017). ILUCTUS: Uma biblioteca para o apoio ao processamento colaborativo de dados. *XVIII Simpósio em Sistemas Computacionais de Alto Desempenho-WSCAD*.
- Drago, I., Mellia, M., M Munafo, M., Sperotto, A., Sadre, R., and Pras, A. (2012). Inside Dropbox: Understanding personal cloud storage services. pages 481–494.
- Schwendler, A., Bretana, L. E., Pernas, A. M. S., and Cavalheiro, G. G. H. (2017). Um estudo de caso da ferramenta ILUCTUS utilizando o cálculo de Fibonacci. *ERAD Escola Regional de Alto Desempenho*.