

GroupUsto.re: Federação de Dados Usando a Plataforma Peer-to-peer

Wilton Oliveira Ferreira^{1,2,3}, Rodrigo Elia Assad^{1,2}, Anderson Fonseca e Silva^{1,2},
Silvio Romero de Lemos Meira¹, Josino Rodrigues Neto^{1,2}

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Jornalista Anibal Fernandes, s/n – Cidade Universitária
50.740-560 – Recife, PE – Brazil

²Usto.re
Rua do Apolo, 161 – Cais do Apolo - Bairro do Recife
50.030-210 – Recife, PE – Brazil

³Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R)
Rua Bione, 220 – Cais do Apolo - Bairro do Recife
50.030-390 – Recife, PE – Brazil

{wof, rea, afs8, srlm, jrn4}@cin.ufpe.br

Abstract. *Studies and experiment performed pointing the overhead of messages in P2P backup systems as a critical factor of this technology. This factor increases drastically the network usage, in addition, the peers involved in these systems do not always have the same power of message processing and in many cases present limitations in speed of data transfer over the network (latency). This paper proposes a mechanism for creating of federations (groups) of peers in P2P systems backups taking as criterion the equivalence of capacity between peers.*

Resumo. *Estudos e experimentos realizados apontam o overhead de mensagens em sistemas de backup P2P como sendo um fator crítico desta tecnologia. Este fator eleva drasticamente o uso da rede, além disso, os peers envolvidos nestes sistemas nem sempre tem o mesmo poder de processamento das mensagens e em muitos casos apresenta limitações na velocidade de transferência de dados pela rede (latência). Este artigo propõe um mecanismo para criação de federações (grupos) de peers em sistemas de backups P2P tomando como critério a equivalência de capacidade entre os peers.*

1. Introdução

Desde o início da década de 1980, o modelo de comunicação P2P (*Peer-to-Peer*) vem sendo tratado como uma arquitetura, e grandes ferramentas como *Napster* [Kan 2001] e *Gnutella* [Kan 2001], foram os propulsores desta nova arquitetura. O crescente número de sistemas P2P estimulou a realização de pesquisas e experimentos, entre os quais foi observada a necessidade de um mecanismo para a auto-organização dos *peers*. Este mecanismo permite que os *peers* possam apresentar mais eficiência. Alguns estudos propõem soluções que permitem organizar computadores em forma de federações.

O conceito de federação apresentado neste trabalho, trata-se de um paradigma criado para agrupar *peers*. Este paradigma possibilita o estabelecimento de interligações de grandes escalas em infraestruturas de tecnologias da informação. Desta forma, serviços e aplicações podem ter ligações entre dois ou mais domínios [Tranoris and Denazis 2010]. O *PanLab* (*Pan-European Laboratory*) apresenta federação como sendo a capacidade de unir sistemas heterogêneos [Tranoris and Denazis 2010][Wahle et al. 2009], sendo definido da mesma forma por [Wang et al. 2011] e também por [Mancini et al. 2009]. Para a comunidade de *Cloud Computing*, federação é a capacidade de organizar e gerenciar um ecossistema fundamentado em armazenamento de dados, entidades e/ou serviços [Buyya et al. 2010].

O Usto.re é um sistema construído sobre a arquitetura P2P. Sistemas desta natureza geralmente apresenta grande *overhead* de tráfego na rede. Com o objetivo de resolver problemas desta natureza, algumas ferramentas utilizadas para a criação de redes P2P apresenta o conceito de grupos de *peer* [Brookshier et al. 2002]. Porém abordagens desta natureza não levam em consideração uma série de fatores fundamentais para sistemas distribuídos. Este artigo tem como objetivo apresentar um mecanismo para auxiliar na criação de federações de *peers*. Além desta seção introdutória, este artigo está estruturado da seguinte forma: Seção 2 Trabalhos Relacionados ; Seção 3 Usto.re; Seção 4 GroupUsto.re; Seção 5 Avaliação dos Resultados; Seção 6 Conclusão.

2. Trabalhos Relacionados

PerfCloud é um *framework* completo que implementa mecanismos que oferecem serviços para a criação e gestão de recursos de computação virtual (clusters virtuais) sob uma *Grid* computacional [Mancini et al. 2009]. Esta ferramenta atua de forma autônoma, dispensando iteração humana. Sendo assim, os clusters podem ser parametrizados exatamente com as características necessária para atender uma requisição. Da mesma forma o CHASE apresenta características semelhantes ao *PerfCloud*. Trata-se de um *framework* que possui um motor de automatização da execução de rotinas de software e serviços em *cloud* ou *grid* [Mancini et al. 2009]. A proposta desta ferramenta consiste em prover auto-organização da infraestruturas e reduzir custos com personalização de paradigmas destinados a comunicação entre *clouds*.

O *FedMi* é um *framework* que integra *Data Grids* heterogêneas[Wang et al. 2011]. A arquitetura *FedMi* é constituída por três componentes principais, sendo estes: *Client Program*, *Meta Data Server* e *Proxy Server*. Estes três componentes trabalham em conjunto para realizar a interoperabilidade entre *Data Grids* heterogêneas, que formam uma federação de *Data Grids*. Já o *gMeasure* é uma ferramenta criada para compartilhar informações em alta velocidade entre computadores em uma rede P2P. Esta ferramenta apresentam abordagens que têm como principal fundamento a análise das características individuais de cada *peer* na rede[Zhang et al. 2002]. O *gMeasure* possui como princípio básico a criação de agrupamento de *peers* com base na latência que os *peers* trafegam dados pela rede. A latência é um dos principais fatores explorados na proposta apresentada neste artigo.

3. Usto.re

O Usto.re é uma solução de data *cloud* desenvolvida desde o início focada na segurança dos arquivos dos seus usuários. Como adicional de segurança cada arquivo antes de ser

salvo e dividido em pedaços e cada pedaço espalhado nos computadores que compõem o a data *cloud*, e a lógica de reconstrução de cada pedaço é privada da solução Usto.re. O projeto Usto.re, trata-se de uma ferramenta, que toma por base a evolução da proposta inicial em [Duarte 2010], onde foi definido um algoritmo estatístico para garantir a disponibilidade de dados armazenados em ambientes P2P. Esta garantia é dada através do cálculo da probabilidade de falha de cada nó que compõe a rede P2P [Assad et al. 2012]. O Usto.re utiliza plataforma P2P para sua implementação, com isso sua arquitetura ilustrada na Figura 1, apresenta um ambiente distribuído para armazenamento de dados.

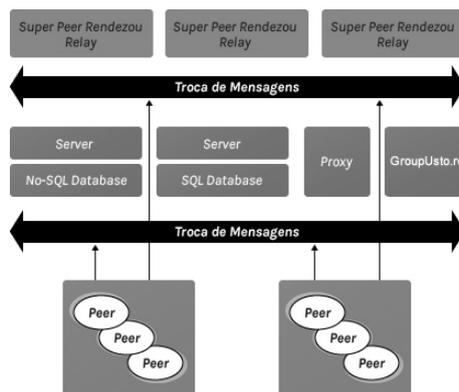


Figure 1. Arquitetura Usto.re

4. GroupUsto.re

O GroupUsto.re é uma ferramenta para recomendação na criação de federações de *peers* em sistemas P2P. A arquitetura GroupUsto.re, ilustrada na Figura 2, apresenta uma série de vantagens com relação aos trabalhos apresentado na Seção 2. O GroupUsto.re além de recomendar os *peers* para formação de federação, também monitora as federações já criadas. O GroupUsto.re recomenda modelos de federações de *peer* seguindo um perfil de capacidade de cada *peer*. Para determinar o perfil do *peer* esta ferramenta conta com os valores obtidos pelos agentes presente na arquitetura ilustrada na Figura 2.

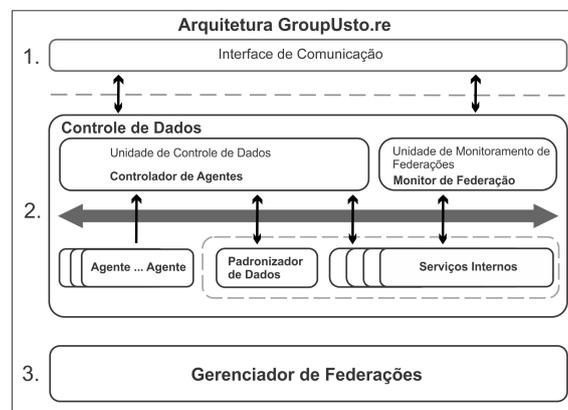


Figure 2. Arquitetura GroupUsto.re

4.1. Controle de Dados

Este módulo é responsável pelo processamento das informações coletadas pelos agentes. As informações obtidas são necessárias para que possa ser definido o perfil de cada *peer*. As informações coletadas pelos agentes apresentam diferentes formatos. Desta forma, faz-se necessário converter tais informações para uma notação comum (processo conhecido como padronização de dados) [Neto and Moita 1998]. Para resolver este problema, a arquitetura do GroupUsto.re apresenta um serviço denominada padronização de dados, que tem por objetivo organizar os dados de forma a ser interpretado igualmente e com a mesma unidade de representação de valores utilizando a transformação “z” [Neto and Moita 1998]. A seguir são apresentados os agentes que compõe esta ferramenta.

4.2. Agentes

LatenceAgent – tem como objetivo verificar a velocidade que o *peer* transfere dados pela rede utilizando método RTT (*Round Trip Time*) [Jan and Noblet 2004]; **MemoryAgent** – tem como objetivo verificar a capacidade de memória RAM (*Random Access Memory*) disponibilizada para uso do sistema P2P; **ProcessorAgent** – tem como objetivo verificar as informações referentes ao número de núcleo de processador disponibilizado para o uso do sistema P2P; **StorageAgent** – tem por objetivo coletar informações referente a capacidade de armazenamento de dados do *peer* e também a quantidade de espaço já utilizado. **HopAgent** – tem como objetivo medir a quantidade de saltos por roteadores que uma mensagem deve executar para sair da origem e alcançar o destino; **AvailabilityAgent** – tem como objetivo mensurar quanto tempo o *peer* tem disponível para o sistema; **ReliabilityAgent** – tem como objetivo medir o quanto o *peer* é confiável com relação a todas as características e serviços por ele disponibilizados.

5. Execução dos Experimentos e Avaliação dos Resultados

A verificação de eficiência na criação das federações sugeridas pelo GroupUsto.re foi realizada a partir de um ambiente parcialmente simulado. Inicialmente foram coletados valores de 20 (vinte) *peers* reais, estes valores foram combinados entre se dando origem a dois novos conjuntos de *peers* simulados, sendo um conjunto com 1162 *peers* e o outro conjunto com 3489 *peers*. O primeiro conjunto com 1162 elementos foi assim escolhido de forma aleatória entre o intervalo de 1000 a 1500 *peers*. Já o segundo conjunto com 3489 *peer* foi obtido a partir da multiplicação de aproximadamente 3 vezes o tamanho do primeiro conjunto. O objetivo de um conjunto amostral ser aproximadamente 1/3 menor que o outro é exatamente observar o comportamento do algoritmo quanto a escalabilidade.

Para execução do experimento foram criados três perfis de federações contendo os seguintes números de *peer*: perfil de federação 1 com 10 *peers*; perfil de federação 2 com 50 *peers* e perfil de federação 3 com 100 *peers*. A quantidade de *peers* em cada federação foi escolhido de forma empírica. Na execução dos experimentos foi primeiramente selecionado o conjunto amostral com 1162 *peers* e em seguida o GroupUsto.re foi executado para cada perfil de federação. Logo em seguida foi selecionado o conjunto com 3489 amostras e da mesma forma o GroupUsto.re foi executado novamente para cada perfil de federação. Os experimentos obteve como resultado 6 (seis) conjuntos de federações. O objetivo do experimento é apresentar a redução de dissimilaridade entre os *peers* que

compõe uma federação. Esta dissimilaridade é dada pelo desvio padrão dos valores obtidos pelos agentes presente nos *peers*, neste caso quanto menor for o desvio padrão maior será a dissimilaridade entre os *peers*.

A verificação de eficiência na sugestão de federação é realizada a partir da comparação de uma federação criada conforme sugestão do GroupUsto.re e um conjunto com o mesmo número de *peer* da federação escolhidos aleatoriamente. Para verificar a eficiência das federações sugeridas pelo GroupUsto.re foi proposta a seguinte equação:

$$a) C = \frac{1}{k} \sum_{z=1}^k S_z \quad b) \bar{X}_k = \frac{1}{n} \sum_{j=1}^n P_j \quad c) S_k = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_i)^2}$$

\bar{X}_k representando a média de valores obtidos por cada agente do *peer* P_j onde k é o número de agente e n é o número de *peers* observados. A equação b) é executada tanto para a federação quanto para o conjunto selecionado para a verificação de eficiência. Após obter o valor de \bar{X}_k tanto para a federação quanto para o conjunto, é realizado o calculado de desvio padrão amostral representado pela equação c) resultando em S_k , onde k é o k -entésimo agente observado para análise e n é o número de *peers* observados. Por fim a equação c) apresenta o coeficiente de redução de dissimilaridade obtido a partir do somatório dos desvios padrões apresentado pela equação c).

Os resultados obtidos nestes experimentos são ilustrados no gráfico da Figura 3. Como pode ser observado, houve redução da dissimilaridade em todos os casos do experimento. Em alguns casos esta redução de dissimilaridade foi maior, já em outros casos a ela foi pouco notável.

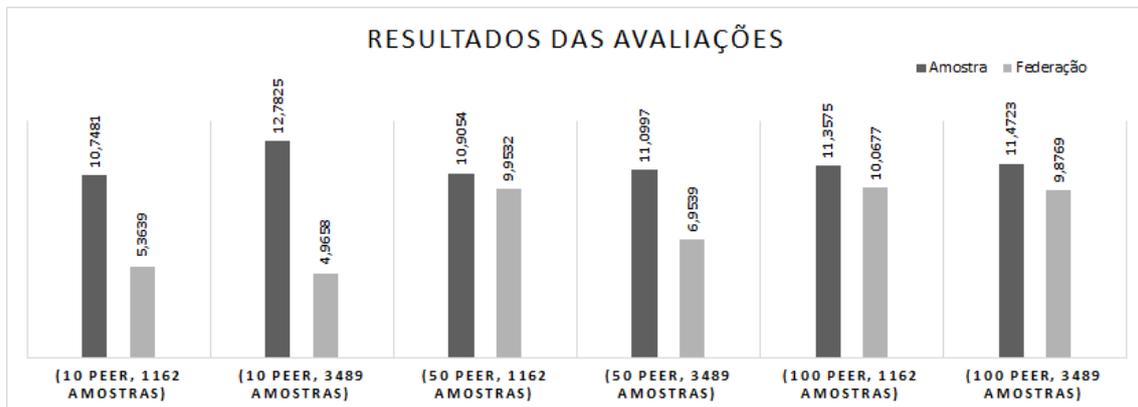


Figure 3. Resultados das Avaliações

6. Conclusão

Como foi observado no gráfico ilustrado na Figura 3, os experimentos apresentaram-se satisfatórios. Com isso conclui-se que o GroupUsto.re sugere organização de *peers* em federações de forma eficiente. E ainda observando o gráfico é possível notar que quanto maior o número de *peer* conectados, mais eficiente torna-se o GroupUsto.re. Desta forma conclui-se que esta ferramenta atende satisfatoriamente demandas de escalabilidades do sistema.

References

- Assad, R. E., Garcia, V., Fernando, P., Machado, M., Soares, F., Vieira, T., Trinta, F., and Meira, S. R. L. (2012). Usto.re: Sistema confiável de armazenamento de dados em nuvem utilizando protocolo p2p. Centro de Estudos e Sistemas Avançado de Recife.
- Brookshier, D., Govoni, D., Krishnan, N., and Soto, J. C. (2002). *JXTA: Java P2P Programming*. Sams Publishing, United States of America, San Francisco, California.
- Buyya, R., Ranjan, R., and Calheiros, R. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and Architectures for Parallel Processing*, volume 6081 of *Lecture Notes in Computer Science*, pages 13–31. Springer Berlin / Heidelberg.
- Duarte, M. P. (2010). Um algoritmo de disponibilidade em sistemas de backup distribuído seguro usando a plataforma peer-to-peer. Master's thesis, Universidade Federal de Pernambuco - Centro de Informática.
- Jan, M. and Noblet, A., D. (2004). Performance evaluation of jxta communication layers. Rapport de recherche RR-5350, INRIA.
- Kan, G. (2001). *Peer to Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly, first edition edition.
- Mancini, E., Rak, M., and Villano, U. (2009). Perfcloud: Grid services for performance-oriented development of cloud computing applications. In *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on*, pages 201–206.
- Neto, J. M. M. and Moita, G. C. (1998). Uma introdução à análise exploratória de dados multivariados. *Química Nova*, 21:467–469.
- Tranoris, C. and Denazis, S. (2010). Federation computing: A pragmatic approach for the future internet. In *Network and Service Management (CNSM), 2010 International Conference on*, pages 190–197.
- Wahle, S., Magedanz, T., Gavras, A., Hrasnica, H., and Denazis, S. (2009). Technical infrastructure for a pan-european federation of testbeds. In *Testbeds and Research Infrastructures for the Development of Networks Communities and Workshops, 2009. TridentCom 2009. 5th International Conference on*, pages 1–8.
- Wang, C.-M., Chen, H.-M., Hsu, C.-C., and Huang, C.-C. (2011). Fedmi: A federation middleware for integrating heterogeneous data grids. In *Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on*, pages 127–134.
- Zhang, X., Liu, J., Zhang, Q., and Zhu, W. (2002). gmeasure: a group-based network performance measurement service for peer-to-peer applications. In *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, volume 3, pages 2528–2532 vol.3.