Avaliação Resiliente de Autorização $UCON_{ABC}$ para Computação em Nuvem

Arlindo L. Marcon Jr., Altair O. Santin (*orientador*)
Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR) – Curitiba, PR – Brasil
{almjr, santin}@ppgia.pucpr.br

Resumo. *O consumidor de nuvem necessita controlar o consumo individual de seus usuários. O modelo $UCON_{ABC}$ permite avaliações periódicas, executando a reavaliação contínua dos atributos de autorização. Porém, o $UCON_{ABC}$ não foi projetado para o contexto da nuvem. Este trabalho mostra que é possível prover resiliência ao processo de reavaliação de autorização do $UCON_{ABC}$. O protótipo como prova de conceito mostra que é possível prover elasticidade às entidades responsáveis por avaliar e contabilizar os atributos de uso.*

Abstract. *The cloud consumer needs to control the individual consumption of their users. The $UCON_{ABC}$ model allows periodic evaluations, performing continuous reevaluations of authorization attributes. However, the $UCON_{ABC}$ was not designed for the context of the cloud computing. This work shows that it is possible to provide resilience to the $UCON_{ABC}$ authorization revaluation. The proof-of-concept prototype shows that providing elasticity is feasible for evaluating entities and accounting attributes.*

1. Introdução

A computação em nuvem fornece serviços que podem ser contratados conforme a demanda do consumidor [Hayes, 2008]. As principais entidades envolvidas nesse contexto são: *i) consumidor*, contrata os serviços para atender seus usuários; *ii) usuário*, sujeito que utiliza os serviços - *i.e.* o usuário final; *iii) provedores*, fornecem os serviços para a nuvem computacional - *e.g.*, infraestrutura, plataforma [Marcon Jr. *et al.*, 2010].

Os provedores de serviço devem honrar os contratos (*Service Level Agreements - SLAs*) firmados com o consumidor, controlando o acesso em nível de usuário e administrativo [Emeakaroha *et al.*, 2011]. Adicionalmente, os provedores deveriam monitorar os recursos alocados e oferecer um esquema de gerenciamento eficiente para seus consumidores [CSA, 2011]. Cada usuário do consumidor pode originar diferentes cenários de carga para a nuvem (*e.g.*, processamento, armazenamento), utilizando recursos de vários provedores. Neste caso, a abordagem tradicional (*i.e.*, configuração estática de políticas nos serviços) pode caracterizar a subutilização de recursos em um provedor e a sobrecarga destes em outro (*i.e.*, não há como prever as demandas). Em um contexto ideal, as políticas de uso deveriam ser frequentemente avaliadas visando detectar oscilações no consumo. A monitoração periódica dos recursos asseguraria a utilização uniforme dos serviços, sem prejudicar o acesso do usuário.

Neste trabalho, agentes de monitoramento coletam informações de consumo enquanto o controle de uso (*i.e.*, $UCON_{ABC}$) avalia continuamente as autorizações [Park *et al.*, 2004]. A continuidade do uso é concedida conforme a reavaliação das políticas. O uso pode ser entendido como operações de escrita em um objeto (*e.g.*, um arquivo) ou o consumo de recursos (*e.g.*, ciclos de *CPU*). A frequência da coleta de atributos reflete diretamente no período de tempo em que o usuário pode estar violando uma política, situação esta que caracteriza uma exceção (*i.e.*, disparidade entre a autorização concedida e a política vigente). Evidentemente, é desejável aplicar o menor intervalo de

Este trabalho é parcialmente financiado pelo Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC/MCTI), contrato 1313 e CNPq, processos: 478285/2011-6 e 310671/2012-4.

tempo possível para a obtenção dos atributos de uso e a reavaliação das políticas. Esta proposta mostra um cenário em que os períodos de inconsistência de autorização (*i.e.*, situação na qual o usuário está em condição de exceção) são mitigados.

O trabalho está organizado em: seção 2, aborda a nuvem computacional; seção 3, trata do controle de uso; seção 4, apresenta alguns trabalhos relacionados; seção 5, contém a proposta; seção 6, protótipo e testes; por fim, a seção 7 elenca as conclusões.

2. Computação em Nuvem

A nuvem pode ser descrita pelos seguintes itens [Mell *et al.*, 2009]: *i)* auto-atendimento sob-demanda; *ii)* serviços amplamente disponíveis na rede; *iii) pool* de recursos computacionais; *iv)* rápida elasticidade; *v)* contabilização de consumo. Esse modelo é formado por um conjunto de serviços classificados em: *i) Software como um Serviço - SaaS*: representa o *software* que o consumidor utiliza; *ii) Plataforma como um Serviço - PaaS*: permite ao consumidor instalar aplicações e gerenciar as configurações subjacentes; *iii) Infraestrutura como um Serviço - IaaS*: provê recursos computacionais (*e.g.*, processador, memória) para a execução do sistema operacional (*e.g.*, *Linux*).

3. Controle de Uso

A abordagem mais adequada para ambientes dinâmicos é o controle de uso $UCON_{ABC}$ [Park *et al.*, 2004]. Este modelo reavalia as autorizações periodicamente, levando em consideração a mutabilidade dos atributos. As autorizações seguem o modelo tradicional de avaliação e concessão de direitos (*e.g.*, leitura, escrita). Porém, o $UCON_{ABC}$ faz a avaliação continua, tendo em vista que os atributos do usuário ou objeto (*e.g.*, serviço) podem ser alterados à medida que o acesso é executado. Assim, os atributos de consumo podem ser atualizados antes ou durante a utilização do objeto, necessitando de uma avaliação de autorização antes (*pre*) e durante (*ongoing*) o uso. A avaliação *pre* é clássica, enquanto a avaliação *ongoing* é necessária devido à mutabilidade dos atributos.

4. Trabalhos Relacionados

A proposta de Zhang [Zhang *et al.*, 2009] auxilia os provedores a aumentarem a flexibilidade dos serviços oferecidos na nuvem (*i.e. SaaS*). O trabalho adota o esquema de segurança tradicional, desconsiderando a flexibilidade dos demais níveis de serviço (*e.g. PaaS*, *IaaS*). Para Lim [Lim *et al.*, 2009], políticas munidas de atributos fornecidos pelo provedor auxiliam o consumidor a administrar a utilização dos recursos alocados. O artigo expõe a necessidade do consumidor em gerenciar a cota contratada. O trabalho de Bertram [Bertram *et al.*, 2010] faz o mapeamento de atributos do ambiente em políticas de controle. Porém, a proposta não deixa claro como o processo é executado.

A proposta de Tavizi [Tavizi *et al.*, 2012] aplica o tratamento de obrigações $UCON_{ABC}$ em ambientes de nuvem. O trabalho de Danwei [Danwei *et al.*, 2009] apresenta um módulo de negociação para aumentar a flexibilidade do sistema de controle de acesso. Ambas as pesquisas desconsideram a periodicidade do sistema de monitoramento de atributos, item que reflete diretamente na reavaliação de autorização.

5. Avaliação Resiliente de Autorização $UCON_{ABC}$

A proposta estende o modelo de autorização $UCON_{ABC}$, provendo resiliência a reavaliação das políticas de uso [Marcon Jr. *et al.*, 2013]. Resiliência, nesta abordagem, significa prover ao modelo a habilidade de tratar algumas situações de exceção que ocorrem com os atributos de autorização do usuário. Porém, mantendo as cotas de consumo do serviço dentro dos parâmetros definidos no *SLA* (Figura 1; *evento* SLA_{CO}). O domínio consumidor escreve as políticas de uso para seus usuários (*evento* U_{AQ}),

definindo os atributos de autorização individuais (*i.e.*, as cotas de uso de serviço). A cota é utilizada durante o processo de reavaliação de políticas em substituição aos atributos de autorização. O objetivo deste esquema é flexibilizar as políticas de uso, quando possível, lidando com situações de exceção de autorização.

Atributos que contabilizam o consumo do usuário são obtidos do provedor através de agentes de monitoramento (*Agm*; *evento* A_{UU}; Figura 1). A resiliência para o processo de reavaliação de autorização contínua está definida somente se o SLA_{CO} menos a soma que contabiliza todos os atributos de uso dos usuários é maior que *t*. A constante *t* é uma cota reserva definida pelo consumidor para um serviço. Isto significa que, quando a soma de consumo total dos usuários (*sctu*) estiver próxima do limiar definido por "SLA_{CO} - *t*", um novo *SLA* deverá ser negociado para evitar uma violação de contrato.

Um cenário envolvendo um sistema de arquivos é utilizado para explicar como a cota é utilizada para prover resiliência ao processo de autorização $UCON_{ABC}$. Neste contexto (Figura 1) a resiliência está representada por linhas pontilhadas para os atributos de uso e de autorização. Considerando um consumidor que negocia um contrato (*evento* SLA_{CO}) para um serviço de armazenamento com *600GB*, sendo *t* = *100GB*. O consumidor escreve políticas que definem *userA: 200GB*, *userB: 200GB* e *userC: 100GB*. Quando o usuário solicita acesso ao serviço (*eventos* AC_A e AC_B), o guardião (*GS*) envia uma solicitação de avaliação de autorização para o monitor de referência $UCON_{ABC}$ (*MRU*; *evento PER*). O *MRU* configura a cota de consumo para o usuário (*i.e.*, o valor inicial da cota é igual ao atributo de autorização definido pelo consumidor: *userA: 200GB*, *userB: 200GB* e *userC: 100GB*), fornecendo a permissão de consumo para o mecanismo que executa o controle de acesso (*evento PDE*).

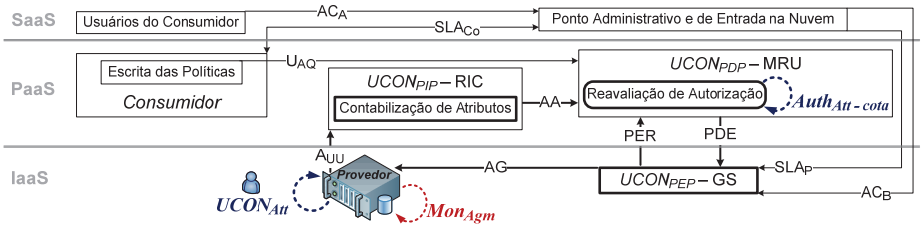

Figura 1. Modelo de autorização resiliente $UCON_{ABC}$

Após ter o acesso liberado pelo guardião (*evento AG*), o usuário começa a utilizar o serviço. Em seguida, o agente de monitoramento (*Agm*) envia os atributos de uso de cada usuário (*e.g.*, *userA: 190GB, userB: 10GB* e *userC: 0GB*) para o repositório de informações de contexto (*RIC*). Passado algum tempo, o guardião (*GS*) solicita uma reavaliação de autorização. Durante a reavaliação (autorização *ongoing*), o monitor de referência (*MRU*) compara os atributos de cada usuário (*evento AA*) com a cota (*i.e.*, mecanismo de resiliência). Neste caso, o armazenamento utilizado por cada usuário está abaixo das cotas predefinidas. Novamente, após um período de tempo, o agente envia outra vez os atributos de uso (*e.g.*, *userA: 250GB, userB: 20GB* e *userC: 10GB*) para o repositório (*RIC*) e uma reavaliação de autorização é necessária. Durante a reavaliação, o *MRU* percebe que os atributos do *userA* estão excedendo a cota predefinida (*i.e. userA: 200GB*). Neste momento, a condição de soma de consumo para os usuários (*sctu*) é de *280GB*. Como o "SLA_{CO} - *t*" admite *500GB* de armazenamento, a cota para o usuário em situação irregular é automaticamente expandida para *userA: 250GB*.

Periodicamente, o agente (*Agm*) envia os atributos de cada usuário (*e.g.*, *userA: 240GB, userB: 160GB* e *userC: 100GB*) para o repositório (*RIC*). Enquanto isto, o guardião (*GS*) continua solicitando as reavaliações de autorização para o monitor de referência (*MRU*). Neste processo, o *MRU* percebe que os atributos do *userA* estão

abaixo da cota expandida, porém, a condição de soma de consumo (*sctu*) é de *500GB* de armazenamento. Neste momento, o processo de reavaliação detecta que o espaço de armazenamento, subtraído da cota reserva (SLA_{CO} - *t*) foi totalmente utilizado. Adicionalmente, a cota do *userA* está excedendo a política, sendo que a parcela de armazenamento extra deveria ser equiparada com o atributo de autorização original - *i.e. userA: 200Gb*. Se, na próxima reavaliação, os atributos do *userA* se mantiverem além da cota, e a soma dos atributos (*sctu*) estiver próxima do SAL_{CO}, o *userA* vai estar em uma condição de exceção (*i.e.*, o usuário utilizou mais de *200GB*, não sendo possível manter a resiliência). Caso contrário, a cota do *userA* poderia ser alterada novamente.

A exceção ocorre porque entre os períodos de reavaliação, o usuário continua consumindo o serviço. Essa situação também pode ser desencadeada pela resiliência do modelo, o qual redefine a cota para o valor original do atributo de autorização. O modelo proposto provê um equilíbrio automático entre o limite estabelecido na política e a quantidade definida no *SLA*. O esquema de reavaliação é resiliente para o processo de autorização, explorando a ociosidade na utilização dos serviços - desde que o *sctu* esteja abaixo do limiar "$SLA_{CO} - t$". Nas abordagens tradicionais, a autorização é avaliada somente no inicio da utilização (*pre*), porém isto pode gerar inconsistências entre a autorização concedida e a política de uso vigente. Mesmo a reavaliação de autorização durante o acesso (*ongoing*) não garante que um consumo autorizado não vai violar a política. Isto ocorre porque as condições de exceção acontecem entre os períodos de reavaliação. Nesta proposta, estes períodos são menores do que nas abordagens tradicionais, dependendo apenas do intervalo entre as reavaliações de autorização.

5.1. Arquitetura do Modelo

A proposta utiliza um ambiente intermediário (Ambiente Federado - *AF*; Figura 2) para facilitar a interação entre as entidades e para prover um *PaaS* seguro para a execução do modelo de autorização resiliente. Os Provedores de Serviço (*PS*) e consumidores filiam-se ao ambiente de nuvem através do *Broker*. O *Broker* faz a intermediação da oferta de serviços (*IaaS*), negocia os *SLAs* com os consumidores, e executa o redirecionamento dos usuários do consumidor (*DC*) para o endereço do provedor definido no *SLA*.

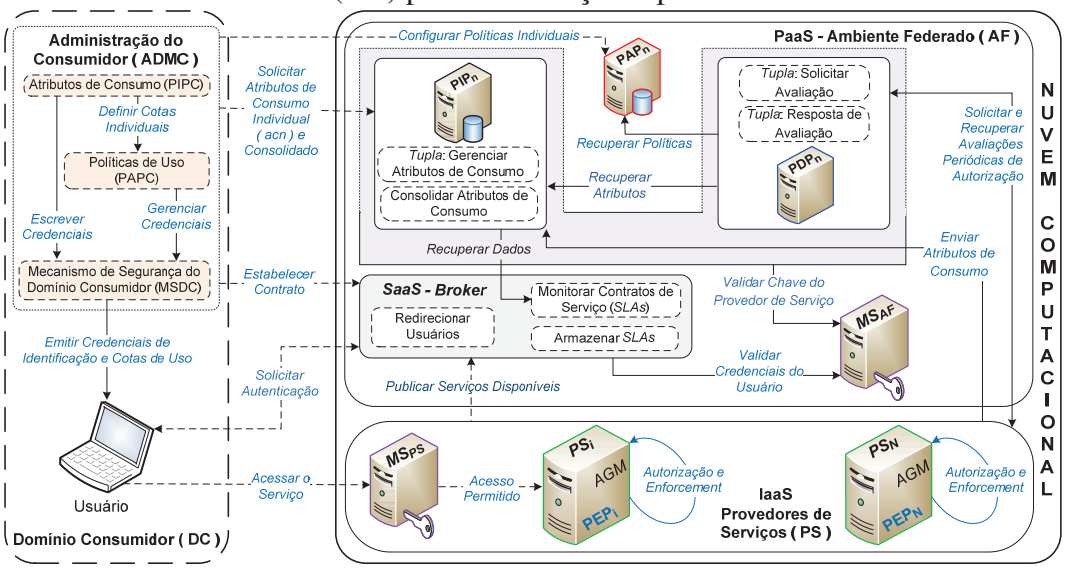

Figura 2. Visão geral da arquitetura proposta

Para cada serviço o provedor envia ao *Broker*: *a) SLA*: define os montantes que podem ser repassados aos consumidores; *b) descrição do serviço*: documento que o

usuário do consumidor utiliza para acessar o serviço. O consumidor estabelece o *SLA* com o *Broker* e define as políticas para seus usuários. As regras de uso são configuradas no ponto de administração de políticas (*PAP*; Figura 2) da federação. Cada *SLA* fornece ao consumidor um conjunto de serviços para gerenciar o ambiente (*e.g.*, armazenamento de atributos (*PIP*) e políticas (*PAP*)). A administração do consumidor (*ADMC*) é responsável por escrever as credenciais de autenticação para seus usuários no Mecanismo de Segurança do Domínio Consumidor (*MSDC*). Estas credenciais permitem aos usuários interagirem com o *AF* e consumirem os serviços instanciados no provedor.

Antes do usuário utilizar sua cota de uso, esse deve enviar ao *Broker* a credencial de autenticação assinada pelo consumidor. O *Broker* é quem redireciona o usuário para o serviço desejado. A solicitação de acesso recebida pelo provedor é interceptada pelo guardião (*PEP*; Figura 2) e avaliada pelo monitor de referência (*PDP*) instanciado no Ambiente Federado (*AF*). Esta avaliação utiliza as informações de contabilização de consumo do usuário armazenadas no repositório de informações de contexto (*PIP*). O repositório é atualizado por agentes de monitoramento (*AGM*) instanciados nos provedores de serviço. Após o consumidor configurar as políticas de uso, e o usuário iniciar a utilização do serviço, o *PIP* passa a armazenar atributos de consumo referentes a cada usuário acessando o serviço. Periodicamente os dados são consolidados no *PIP* para avaliar se o SLA_{CO} está sendo respeitado. No Domínio Consumidor, estes dados são analisados para decidir se as políticas de uso individuais precisam ser readequadas.

Considerando avaliações anteriores, o modelo *outsourcing* se mostrou mais adequado a nuvem devido à dinamicidade deste ambiente [Marcon Jr *et al.*, 2009]. Com esse modelo evitam-se as inconsistências causadas pelo *cache* de políticas no provedor. A proposta emprega serviços de espaço de *tuplas*, sendo desacoplada no tempo e espaço. Estes serviços são utilizados para armazenar as solicitações de reavaliação de autorização (enviada pelos *PEPs*) e as respostas das avaliações (envidas pelos *PDPs*).

5.2. Gerenciamento de Atributos

Os atributos da camada *IaaS* (Figura 3; *evento rsv*) refletem o consumo da máquina virtual como um todo (*e.g. CPU-v*). A camada *PaaS* fornece os atributos do usuário que está consumindo o serviço (*e.g. Atr-Usuário*; *evento ua*). Os dados armazenados no *PIP* são enviados por agentes (*AGM*) instanciados nas máquinas virtuais dos provedores (*evento tp*). O *PIP* da federação fornece as informações de consumo individual para o consumidor (*PIPC*; *evento st*) e para o *Broker* (*evento ar*) monitorar os *SLAs*.

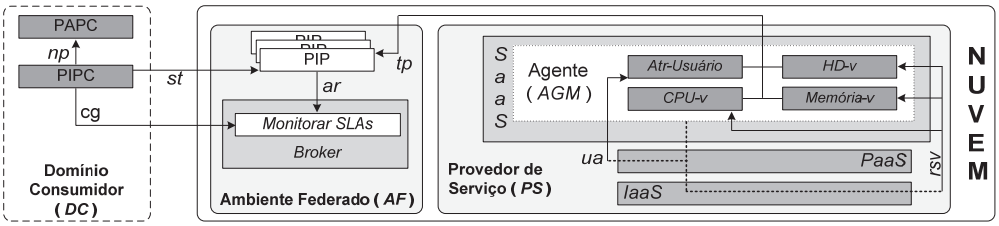

Figura 3. Gerenciamento de atributos de consumo

Com os atributos individuais (*evento st*) e o consumo consolidado (*evento cg*) é possível identificar quem está utilizando os serviços (*evento ua*) e a existência de recursos ociosos neste ambiente (*evento rsv*). Estes atributos são fornecidos para a administração de políticas do consumidor (*PAPC*; *evento np*) com intuito de otimizar a taxa de utilização dos recursos ou justificar a contratação de mais serviços.

5.3. Controle de Uso e Gerenciamento de Políticas

O administrador de políticas (*PAPC*) e o mecanismo de segurança (*MSDC*; Figura 4) transformam o *SLA* em regras de uso e credenciais de acesso (*evento re*). Utilizando as credenciais (*evento au*), os usuários acessam o Repositório de Interfaces (*RI*; *evento is*) do *Broker* e os serviços no provedor (*evento ac*). As políticas de uso são transferidas para a federação e armazenadas no *PAP* (*evento en*). Estas regras serão utilizadas para configurar a cota do usuário e para que um dos *PDP´s* pertencente ao *pool* de servidores da federação possa avaliar as solicitações de autorização (*eventos av, rp*). A avaliação utiliza os dados de consumo disponíveis no repositório de atributos (*PIP*; Seção 5.2).

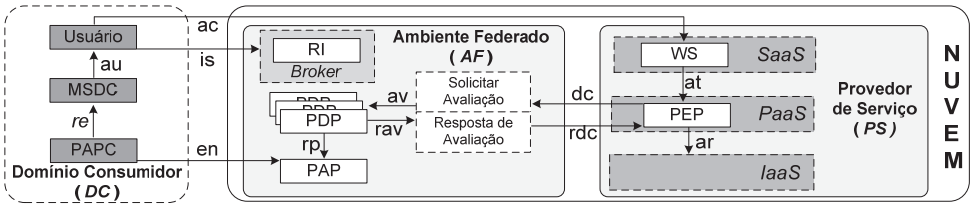

Figura 4. Gerenciamento de políticas de uso

Com a utilização do modelo *outsourcing* [Yavatkar *et al.*, 2010], o usuário precisa ser autorizado pelo *PDP* e ter o acesso liberado pelo *PEP* (camada *PaaS*; *evento at*) para poder consumir o recurso (camada *IaaS*; *evento ar*). Esta proposta libera o provedor e o consumidor da tarefa de implementar o monitor de referência $UCON_{ABC}$ e o gerenciamento de atributos de consumo (*i.e.* serviços fornecidos pela federação).

6. Protótipo e Testes de Avaliação

O protótipo utiliza os seguintes projetos e *APIs*: *Java* (*java.lang.management*); *SIGAR* (*hyperic.com/support/docs/sigar*); *JavaSysMon* (*jezhumble.github.com/javasysmon*); *Sun XACML* (*sunxacml.sourceforge.net*); *Apache TomCat* (*tomcat.apache.org*); módulo *Rampart* (*axis.apache.org/axis2/java/rampart*) integrado ao *Axis2* (*axis.apache.org*) para proteger as mensagens *SOAP* [W3C, 2007] (*i.e.* especificações *WS-Security* [OASIS, 2004] e *WS-Trust* [OASIS, 2007]); espaço de *tuplas River* (*river.apache.org).*

6.1. Testes e Avaliação

O serviço *web* [W3C, 2004] oferecido ao usuário simula um *e-commerce*. Nos testes, cada agente enviou uma *tupla* com *17* dados diferentes para o *PIP* (em torno de *2KB*). Os conteúdos enviados são: dados da máquina virtual que hospeda o serviço *web* (*e.g.* utilização do processador, memória); taxas referentes à *Java heap*; identificação da *thread* responsável por executar o serviço; tempo gasto pelo agente para coletar os atributos; tempo gasto pelo serviço para atender a solicitação do usuário (*overhead*).

As medidas foram obtidas executando-se *1000* interações e calculando a média entre estas. O coeficiente de variabilidade foi abaixo de *5%* em todos os casos. Os testes executados no espaço de *tuplas* (Tabela 1) visam identificar o número de entradas que este espaço consegue armazenar. Pode-se perceber que, na média, enquanto o tamanho da *tupla* dobra de valor, o número de entradas armazenadas reduz pela metade. Adicionalmente, o tempo gasto para armazenar cada *tupla* é alterado significativamente, resultando no melhor rendimento para entradas com tamanhos entre *4KB* e *16KB*.

Um teste similar foi executado no *PDP* para o processo de reavaliação de políticas (Tabela 2). Neste caso o *PDP* recupera a solicitação de avaliação do espaço de *tuplas*, (em torno de *2KB*), e a política do repositório (*PAP*). Na sequência, o *PDP* avalia a solicitação e envia o resultado para o espaço de *tuplas* correspondente. Este experimento mostrou a capacidade do *PDP* em atender as solicitações de reavaliação e o tempo gasto para executar a avaliação. De maneira semelhante, o tempo gasto pelo

PEP para escrever uma solicitação de avaliação no espaço de *tuplas* é similar ao tempo gasto pelos agentes (*AGM*) para escrever um atributo de consumo (*tupla* com *2KB*; Tabela 1).

As medidas apresentadas nas Tabelas 1 e 2 fornecem uma noção quanto ao "gatilho de elasticidade" da proposta, indicando que o número de servidores e serviços no *pool* deveria ser incrementado quando a demanda estiver próxima de causar a parada dos serviços. A abordagem de espaço de *tuplas*, juntamente com o esquema de avaliação distribuído (*i.e.* vários *PDPs* instanciados sob demanda) fornece elasticidade ao ambiente de avaliação $UCON_{ABC}$. O gerenciamento de atributos segue a mesma abordagem, fornecendo elasticidade ao sistema de contabilização.

Tabela 1: Armazenamento de *Tuplas*

Tamanho da tupla (incluindo 1KB de cabeçalho)	*Número de tuplas armazenadas antes do serviço recusar conexões*	*Tempo gasto para armazenar uma tupla (ms)*	*Throughput (KB/ms)*
2 KB	69864	2,54	55011
4 KB	45333	2,57	70557
8 KB	26651	3,01	70833
16 KB	14605	3,4	68729
32 KB	7670	4,94	49684
64 KB	3935	7,9	31878
128 KB	1492	13,44	14209
256 KB	1002	25,39	10102
512 KB	502	48,84	5262
1024 KB	250	98,88	2589
2048 KB	124	200,83	1264

Tabela 2: Reavaliação de Políticas

Tamanho da política recuperada do PAP	*Número de políticas reavaliadas paralelamente antes do serviço parar de responder*	*Tempo gasto para avaliar uma política (ms)*
4 KB	1690	0,129916
8 KB	1360	0,129123
16 KB	1080	0,128985
32 KB	640	0,130264
64 KB	470	0,131786
128 KB	310	0,133511
256 KB	220	0,14456
512 KB	130	0,144513
1024 KB	90	0,161233
2048 KB	70	0,186473

7. Conclusões

Esta seção será usada também para apresentar as principais *contribuições* da tese. O trabalho apresentou uma abordagem inovadora para a reavaliação de autorização contínua em controle de uso. O monitoramento constante de serviços e a reavaliação dos atributos de autorização permitem a identificação de disparidades entre autorizações e políticas. O esquema provê resiliência (*i.e.*, relaxamento das regras da política) para os atributos de autorização em algumas circunstâncias, sem perda para o consumidor (*e.g.*, violação de *SLA*). Quando o esquema de resiliência não é possível, o usuário estará em condição de exceção. Para este caso, o consumidor possui algumas alternativas para reparar a situação, o que não acontece nas abordagens tradicionais.

O serviço de contabilização proposto e a reavaliação contínua provê fina granularidade ao esquema de monitoramento e controle de acesso. A resiliência dos atributos de autorização (cotas) tornou o controle de acesso mais flexível. As violações em políticas de controle são monitoradas e tratadas no ambiente de gerenciamento da federação (*SLAs*) e do consumidor (condições de exceção). Este esquema permite o uso dos recursos sem ociosidade ou abuso de consumo dos serviços contratados.

A abordagem proposta mostrou que é possível executar o gerenciamento e a consolidação de atributos utilizando padrões abertos (*e.g.*, Serviços *Web*, Espaço de *Tuplas*). O esquema é adequado para o nível de acesso fornecido pela camada *IaaS*, não necessitando de mudanças no contexto atual. O gerenciamento é executado por serviços que trabalham de acordo com a demanda do consumidor. Sem estes serviços, não seria possível executar o controle fino de consumo, considerando que nenhum provedor de *IaaS* oferece serviços similares.

Resultados (publicações, patente e minicurso)

Marcon Jr, A. L., Santin, A. O., Stihler, M. e Bachtold, J. (2013). *A $UCON_{abc}$ Resilient Authorization Evaluation for Cloud Computing*. IEEE Transactions on Parallel and Distributed Systems, 11 April 2013. IEEE Computer Society Digital Library.

Marcon Jr, A. L., Santin, A. O., e Stihler, M. (2013). *Avaliação Resiliente de Autorização UCON$_{abc}$ para Computação em Nuvem*. XIII *SBSeg 2013*, pg. 16-29.

Marcon Jr, A. L., Santin, A. O., Stihler, M. *Método de Gerenciamento Elástico do Controle de Uso na Computação em Nuvem*. 2013. Patente: BR1020130267562. Data de depósito: 17/10/2013, INPI - Instituto Nacional da Propriedade Industrial.

Marcon Jr, A. L., Laureano, M., Santin, A. O. e Maziero, C. A. (2010). *Aspectos de Segurança e Privacidade em Ambientes de Computação em Nuvem*. X *SBSeg 2010*. *Anais de MiniCursos*, pg. 53-102.

Tese disponível em: *http://www.ppgia.pucpr.br/lib/exe/fetch.php?media=tese_-_arlindo_luis_marcon_junior.pdf*

Referências

Bertram S., Boniface M., Surridge M., Briscombe N. e Hall-May M. (2010). *On-Demand Dynamic Security for Risk-Based Secure Collaboration in Clouds*. 3rd CLOUD, pg. 518-525.

CSA (2011). *Security Guidance for Critical Areas of Focus in Cloud Computing v3.0*. Disponível: *cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf*. Aces. Mar. 2014.

Danwei C., Xiuli H. e Xunyi R., (2009). *Access Control of Cloud Service Based on UCON*. *1st CloudCom* 2009. LNCS. pg. 559-564.

Emeakaroha V. C., Netto M. A. S., Calheiros R. N., Brandic I., Buyya R. e Rose C. A. F. De. (2011). *Towards Autonomic Detection of SLA Violations in Cloud Infrastructures*. *Elsevier* FGCS, pg. 1-13.

Hayes B. (2008). *Cloud computing*. *Communications* ACM, vol. 51, no. 7, pg. 9-11.

Lim H. C., Babu S., Chase J. S. e Parekh S. S. (2009). *Automated Control in Cloud Computing: Challenges and Opportunities. 1st* ACDC, pg. 13-18.

Marcon Jr. A. L., Santin A. O., Lima Jr. L. A. de P., e Stihler M. (2009). *Policy Management Architecture Based on Provisioning Model and Authorization Certificates*. *ACM* SAC, pg. 1594-1598.

Marcon Jr, A. L., Laureano, M., Santin, A. O. e Maziero, C. A. (2010). *Aspectos de Segurança e Privacidade em Ambientes de Computação em Nuvem*. *Anais de MiniCursos do SBSeg 2010, SBC,* pg. 53-102.

Marcon Jr, A. L., Santin, A. O., Stihler, M. e Bachtold, J. (2013). *A UCON$_{abc}$ Resilient Authorization Evaluation for Cloud Computing*, *IEEE TPDS.*

Mell P. e Grance T. (2009). *The NIST Definition of Cloud Computing. Special Publication* 800-145. Acesso: Jan. 2014.

OASIS (2004). *Web Services Security SOAP Message Security 1.1*. Disponível em: *docs.oasis-open.org/wss/v1.1*. Acesso: Jan. 2014.

OASIS (2007). *WS-Trust 1.3*. Disponível em: *www.oasis-open.org/standards#wstrustv1.3*. Acesso: Jan. 2014.

Park J. e Sandhu R. (2004). *The UCON$_{ABC}$ Usage Control Model*. *ACM TISSEC*, vol. 7, no. 1, pg. 128-174.

Yavatkar R., Pendarakis D. e Guerin R. (2000). *A Framework for Policy-based Admission Control*, RFC 2753.

Tavizi T., Shajari M. e Dodangeh P., (2012). *A Usage Control Based Architecture for Cloud Environments*. *IEEE IPDPSW 2012*, *pg. 1534-1539*.

W3C (2004). *Web Services Architecture*. Available at: *www.w3.org/TR/ws-arch*. W3C (2007) *SOAP Version 1.2*. Disponível em: *www.w3.org/TR/soap*.

Zhang L. J. e Zhang J. (2009). *An Integrated Service Model Approach for Enabling SOA*. *IEEE* IT Pro. pg. 28-33.

