Implementação de um Detector de Anomalias de Tráfego de Rede Baseado na Entropia de Métricas para Sistemas de Computação em Nuvem

Ana Cristina Oliveira[1], Aleciano Lobo Júnior[2]

[1] Professora do Instituto Federal da Paraíba – IFPB Campus Campina Grande. e-mail: ana.oliveira@ifpb.edu.br
[2] Graduando do Instituto Federal da Paraíba – IFPB Campus Campina Grande. Bolsista: CNPQ. e-mail: aleciano@gmail.com

Resumo: Detectar anomalias de tráfego em redes de computadores é um problema conhecido e bastante estudado para evitar ataques e utilização não desejada da infraestrutura de comunicação. Para realizar essa tarefa, um conjunto de técnicas pode ser utilizado, como por exemplo, a criação de sistemas de detecção de intrusão e o monitoramento da utilização dos recursos. As técnicas existentes normalmente são baseadas na construção de perfis de tráfego normal e na descoberta de padrões para assinaturas de comportamentos anômalos, como vírus e ataques. Algumas técnicas procuram ser dinâmicas e se baseiam em estudos de como seria o comportamento normal em uma janela de tempo para, em seguida, predizer o grau de desorganização do sistema (entropia), supondo em casos de discrepância do comportamento normal, a existência de anomalias. Neste trabalho foi desenvolvido um detector de anomalias de tráfego de rede para sistemas de computação em nuvem baseado na entropia de métricas de tráfego. Os resultados obtidos apontam para a necessidade de melhor ajuste nos parâmetros de detecção para melhorar sua acurácia e a completude. Esses ajustes devem levar em consideração a natureza do tráfego e seu comportamento padrão.

Palavras–chave: Detecção de anomalias de tráfego, computação em nuvem, entropia.

1. INTRODUÇÃO

A comunicação para fins militares para a qual eram usadas as redes de computadores em sua concepção, deram espaço a inúmeros tipos de aplicações e serviços. Com o passar dos anos, a infraestrutura para equipamentos de rede de comunicações passou a prover serviços de comunicação global em altíssima velocidade, aumentando a escala e o escopo de utilização.

Esse aumento da demanda de serviços de comunicação trouxe também a necessidade da redução dos custos necessários para montar e manter o sistema de comunicação necessário para os usuários. Tentando solucionar essa questão, um novo paradigma tem tomado posição de destaque na prestação de serviços em redes de computadores: a virtualização de recursos como, roteadores, linhas de transmissão, servidores, discos, sistemas operacionais, entre outros. Esta tecnologia ganhou a atenção dos provedores de serviços como forma de redução de custos, realização de testes de software e otimização da utilização de recursos. Um contexto importante para sua aplicação é na computação em nuvem ou *cloud computing* [1][2][3][4][5][6][7][8]. *Cloud computing* define uma infraestrutura virtual para prestação de serviços em rede sob demanda. Os clientes contratam serviços em que a infraestrutura primária de hardware e software encontra-se em centros de dados (*data centers*) remotos e não localmente e sobre seu próprio domínio.

Exemplos de serviços na nuvem são o armazenamento de dados em discos virtuais, como o DropBox, em que os clientes têm a impressão de estarem editando arquivos localmente, as atualizações são salvas nos servidores que compõem a nuvem e podem ser acessadas via Internet em qualquer lugar do mundo. Há ainda a possibilidade de nem ser necessária a instalação local de utilitários como editores de texto, planilha eletrônica, agenda eletrônica e clientes de e-mail, como ocorre com o Google Docs, GMail e Google Calendar. Consequentemente, os arquivos dos usuários podem ser criados e editados remotamente, podendo ser acessados via Internet a qualquer momento.

No entanto, isso implica em algumas preocupações relacionadas à confidencialidade das informações, à segurança oferecida, à detecção de problemas no provimento de serviços, à separação de responsabilidades para garantias de QoS entre a empresa prestadora de computação em nuvem e da concessionária de telecomunicações (ISP - *Internet Service Provider*), e como o cliente pode acompanhar a utilização dos serviços e gargalos na distribuição de conteúdo.

ISBN 978-85-62830-10-5

A computação em nuvem gerou novas demandas. Há demanda por sistemas de contabilização eficientes, que sejam leves o suficiente para garantir bom desempenho em tempo real e que ao mesmo tempo extraiam as informações necessárias, seguindo padrões e com agilidade na busca e recuperação de informações. Os sistemas de contabilização fornecem subsídios para os sistemas de tarifação que também necessitam estar alinhados ao tipo de serviço acordado entre as partes e o que é realmente oferecido.

O tráfego de rede produzido por sistemas de computação em nuvem revela o comportamento dos usuários com relação à utilização dos serviços. Analisar o tráfego e reconhecer os fluxos de cada aplicação do sistema permite que sejam modelados os comportamentos de uso de cada serviço e que sejam criados padrões para o funcionamento normal do sistema.

Com base no estudo do comportamento correto do sistema, podem-se identificar desvios na operação normal do sistema. Estes podem indicar que aplicações estão gerando tráfego além do esperado, por exemplo, um serviço com erro está enviando pacotes de *broadcast* na rede; ou que não há presença de tráfego às 10 horas para um serviço que sempre está operando em horário comercial. Ao longo deste trabalho, esses desvios serão chamados de anomalias.

Gerenciar e analisar tráfego em sistemas de computação em nuvem de larga escala é um desafio. As técnicas empregadas para monitorar e analisar tráfego em sistemas distribuídos convencionais apresentam diferenças com relação aos centros de dados de sistemas de computação em nuvem. Nos métodos convencionais são feitas suposições de que existem padrões para os fluxos de rede, até aceitáveis para aplicações em redes corporativas, mas as aplicações que rodam em centros de dados na nuvem podem sofrer mudanças significativas nos padrões de tráfego[9].

Wang [10] propôs um novo método para detecção online de anomalias baseado no cálculo da entropia sobre a distribuição dos valores uma métrica do sistema, ou uma composição de métricas. O método proposto não é intrusivo, utiliza mecanismos leves de caixa-preta ou caixa-cinza, é escalável e não requer modelos previamente determinados de anomalias. O nome dado a essa técnica é EbAT (*Entropy-based Anomaly Testing*). O fundamento da mesma é estudar o comportamento das aplicações, extraindo o grau de dispersão ou concentração das distribuições de métricas. Isso é obtido a partir do estabelecimento de séries temporais de entropia, resultantes da análise de ondulação e na detecção visual de picos nas distribuições de métricas, ao invés da observância de limiares individuais para as métricas.

Wang, Talwar, Schwan e Ranganathan [11] realizaram um experimento para demonstrar a viabilidade e a acurácia do método EbAT, comparando-o a métodos baseados em detecção por limiar. As anomalias consideradas nesse trabalho foram erros em operações, falhas de hardware ou software e super/subprovisionamento de recursos. Os autores fazem um levantamento de técnicas tradicionais de detecção e criticam o fato de serem fundamentadas em complexas análises estatísticas ou a falta de escalabilidade dos métodos que realizam mineração em grandes quantidades de dados com métricas desagregadas. Uma das vantagens desse método é não requerer intervenção humana nem o uso de regras ou modelos de anomalias pré-definidos.

O objetivo deste estudo é apresentar uma metodologia de detecção de anomalias baseada em entropia [10]. A metodologia em estudo foi inicialmente e avaliada com métricas de utilização de processamento e de memória por Wang *et al.* [11] e será adaptada para a análise de tráfego de rede.

2. COMPUTAÇÃO EM NUVEM

Na computação em nuvem existe o estabelecimento e o cumprimento de acordos de nível de serviço (SLA – *Service Level Agreement*) firmados entre o cliente e o provedor de serviços. Essa não é uma característica nativa de outros sistemas de computação sob demanda, como grades computacionais, pois estas apresentam imprevisibilidade na alocação dos recursos e, portanto, também na garantia de metas técnicas para os serviços prestados.

Weinhardt, Anandasivam, Blau e Stößer [12] definiram uma ontologia para modelos de negócios em computação em nuvem, contemplando as três camadas da arquitetura de nuvem: infraestrutura, plataforma e software-como-serviço, como pode ser visto na Figura 1. Esses autores ressaltam que linhas de pesquisa em computação em nuvem irão se focar na padronização de uma

ISBN 978-85-62830-10-5

API, em questões de segurança, novos modelos de negócios e sistemas de precificação dinâmicos para atender a serviços complexos.

Armbrust *et al.* [13] discutiram o modelo econômico de computação em nuvem, discriminando as diferenças entre computação em nuvem e centros de dados privados, ressaltando que nos sistemas de computação em nuvem a escala econômica pode ser expandida rapidamente para sistemas extremamente grandes ou reduzida rapidamente. Para manter as características de disponibilidade infinita de recursos, o monopólio sobre serviços não deve pertencer a um determinado provedor, haja vista que a existência de múltiplos provedores dá suporte a negociações e acordos para balanceamento de carga federado e que sejam assumidos acordos que possam ir além do provisionamento de pico.

Os 10 principais obstáculos e oportunidades com foco na computação em nuvem foram apontados por Armbrust *et al.* [13]: i) continuidade de negócio e disponibilidade de serviço; ii) bloqueio de dados; iii) confidencialidade e auditoria dos dados; iv) gargalos na transferência dos dados; v) imprevisibilidade de desempenho; vi) armazenamento escalável; vii) falhas em sistemas distribuídos de larga escala; viii) rapidez para escalar o sistema; ix) compartilhamento de reputação; x) licenciamento de software.

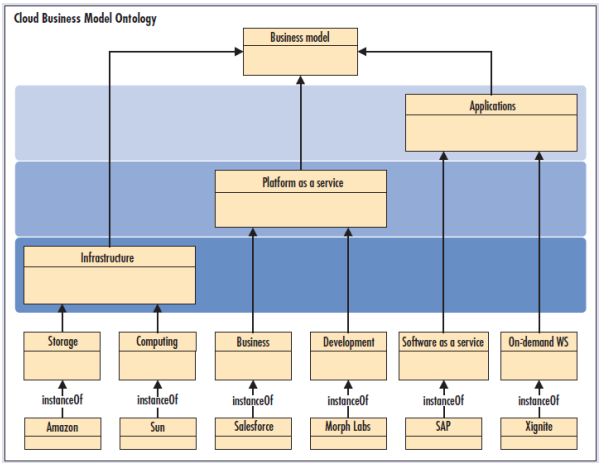

Figura 1 - Arquitetura de computação em nuvem e exemplos de serviços oferecidos por cada camada[12].

Armbrust *et al.* deram enfoque ao fato dos sistemas de computação em nuvem poderem tanto crescer quanto reduzir muito rapidamente, que essa característica é bem endereçada pelos esquemas de pagamento pelo que for utilizado (*pay-for-use*), que a infraestrutura deve ter ciência do que está rodando ou não nas máquinas virtuais (VM) e que o monitoramento e a tarifação devem acompanhar a execução do sistema desde o princípio do funcionamento; chamam atenção também ao fato de que a cobrança dos serviços prestados deve considerar os custos de desempenho oferecidos pelo sistema, em termos da especificação do hardware, e custos proporcionais ao trabalho do sistema, como a utilização de memória ou energia elétrica

3. MONITORAMENTO E ANÁLISE DE TRÁFEGO

O monitoramento de sistemas de larga escala envolve diversos desafios. Um exemplo disso é como monitorar o tráfego de um sistema como a Internet. É preciso definir as políticas de monitoramento, como a opção por agregamento de tráfego, fazendo análise de fluxo, ou o monitoramento da carga útil individual de cada pacote que transita na rede.

Callado et al. [14] discutem a completude e a acurácia de técnicas para identificação de tráfego da Internet. Esse trabalho faz um levantamento do estado-da-arte de técnicas em coleta de tráfego usando medições ativas e passivas e técnicas para identificar tráfego, classificando as aplicações de rede. Ao se optar por uma análise *offline* de tráfego, ou seja, armazenando o tráfego de rede para

ISBN 978-85-62830-10-5

posterior análise, além do tempo de processamento de grandes volumes de dados, há o problema de armazenamento desses dados. Por exemplo, para uma rede que opera a 1 Gbps, é requerido 3,6 TB de espaço em disco para apenas 1 h de coleta de tráfego.

Uma solução alternativa é a análise *online* de tráfego, no entanto, há uma demanda alta por poder de processamento instantâneo, o que está associada a uma ocupação de memória RAM intensiva também. Por exemplo, para o monitoramento de uma rede durante 8 s, considerando que ela opera a 1 Gbps, é necessário que sejam armazenados 1 GB de dados em memória RAM. O processamento desses dados deve ser realizado, ainda, de modo que não interfira no processo de produção e transmissão dos mesmos. Portanto, mecanismos leves e não intrusivos de medição e análise devem utilizados para sistemas com alta vazão de dados.

Callado et al. [15] propõem melhorias ao mecanismo de identificação de aplicações, propondo uma composição de técnicas encontradas na literatura para melhorar a acurácia e a completude da classificação de tráfego. Os autores ressaltam que as características do tráfego afetam o resultado obtido, portanto estabelecer uma técnica genérica com um desempenho esperado conhecido, para qualquer cenário de tráfego, continua um problema em aberto. A análise de tráfego não é uma tarefa trivial, especialmente em sistemas de alto desempenho. Há necessidade de mecanismos de monitoramento leves, de alto desempenho e não intrusivos.

4. DETECÇÃO DE ANOMALIAS BASEADA EM ENTROPIA

Neste trabalho foi implementada a técnica de detecção de anomalias de tráfego proposta por Wang [10] no contexto de tráfego de rede de sistemas de computação em nuvem. O trabalho de avaliação da técnica foi iniciado com o trabalho de Wang *et al.* [11] sobre a detecção de anomalias em tráfego de computação em nuvem utilizando distribuições de métricas e realizando o cálculo de entropia sobre elas. O artigo apresenta os principais problemas que ocorrem no ambiente de nuvem e cita algumas das técnicas abordadas atualmente, mostrando em quais pontos são uteis à nuvem e onde falham. De acordo com os autores, a maioria dos problemas provém de erros de operação, falhas de hardware e software, recursos mal provisionados e outros semelhantes. Posteriormente no trabalho, é detalhado o processo do cálculo da entropia, a forma de testes utilizada para aferir o desempenho do analisador, comparação com outros métodos de detecção baseado em limiares e os resultados em comparação com estes outros métodos.

O método chamado de EbAT (*Entropy-based Anomaly Testing*) apresenta de acordo com o artigo uma precisão de 57,4% na detecção de anomalias e em média é 59,3% melhor na taxa de alarmes falsos. Além disso, tem a vantagem de ser bastante leve, não requerer intervenção humana, funcionar *online* e é totalmente escalável no ambiente de nuvem, podendo apresentar a entropia de uma máquina virtual, host, rack, data center, até a nuvem completa. Isto é possível graças à forma que os dados de métricas são armazenados, permitindo que a entropia seja calculada em diversos níveis no ambiente de *data center* de forma hierárquica.

Após estudar e compreender o que era proposto no artigo, foi iniciada a fase de implementação do analisador baseado em entropia. Foi decidido utilizar os mesmo valores de variáveis e métricas citados no artigo para as primeiras versões. O percentual de uso de processador e a quantidade de memória principal utilizada em *bytes* foram escolhidos para serem as métricas locais do analisador. Foi necessário algum software para captar estas informações no ambiente e as passar ao analisador.

Para garantir a escalabilidade do analisador, as métricas em estudo foram armazenadas em filas de forma que uma ou várias métricas sejam, com tamanho facilmente configurável sem necessidade de alteração no código fonte, sendo informado no momento de executar o analisador. Para garantir portabilidade do código, também foram incluídos mecanismos para verificar se há um ou mais núcleos nos processadores, permitindo que assim o analisador funcione corretamente em vários tipos de máquinas. O processo de captura de métricas, como apresentado na Figura 1, é a primeira operação que deve ser feita pelo analisador.

Após serem armazenadas em filas, as métricas devem passar por um processo chamado normalização, que ocorre a cada vez que novas métricas são enfileiradas. A Equação 1 mostra como deve ser feito o cálculo para obter os valores normalizados. Foram necessárias duas variáveis, *r* e *m*,

ISBN 978-85-62830-10-5

que são respectivamente: um valor que corresponde a um intervalo iniciado em 0 e um valor que irá dividir este intervalo *r* igualmente em *m* partes. Valores dentro da mesma porção de *r* recebem um número, que crescem de 0 até *m*-1 para os valores da última porção. Valores maiores que *r* recebem *m* como número. No outro parêntese temos outro valor, que consiste na divisão de um dado valor de métrica pela média dos valores lidos até o momento ou de toda a janela de observação, caso a mesma já tenha sido preenchida pelo menos uma vez. O processo de normalização pode ser aplicado a qualquer **valor amostrado** (*sample value*) que possa ser inserido como entrada no analisador.

$$normalized_{value} = (sample_{value} / mean_{values}) / (r / m) \quad (1)$$

Após esta operação, o analisador armazena o valor resultante da Equação 1, para uma ou mais métricas, em outra fila de mesmo tamanho, que posteriormente é utilizada para o cálculo de entropia. Tais dados, assim como as métricas puras, representam valores de eventos de observação de certo momento, neste caso, normalizados.

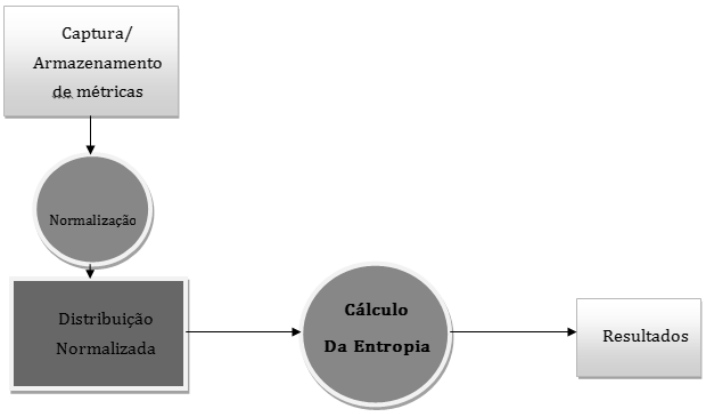

Figura 1 – Sequência de funcionamento do analisador baseado em entropia

Cálculo de Entropia

A terceira e mais importante operação é feita sobre os valores que foram normalizados através da ação feita anteriormente, como nos passos descritos na Figura 1. O cálculo da entropia é realizado sobre esta distribuição de valores, no caso, uma fila de tamanho *n*, correspondente à capacidade da janela de observação. A cada valor inserido, inclusive para o primeiro, o cálculo é (re)feito.

Consideremos um evento e_i como o último a integrar a fila de valores normalizados. É feita uma verificação de quantos eventos iguais ao evento e_i existem na distribuição, chamaremos este número de eventos iguais de n_i. Este número é importante para caracterizar se os valores estão concentrados ou dispersos. Quanto maior o valor de n_i, menos dispersos, apresentando assim, menor entropia.

O tamanho da janela de observação é subtraído em n_i vezes para não realizar o somatório para valores de eventos repetidos, chamaremos este valor resultante de *v*. O *log* na base 10 foi adotado para que as unidades resultantes sejam decimais.

A entropia *H(E)* é calculada através da Equação 2.

$$H(E) = -\sum_{i=1}^{v} \frac{n_i}{n} \log_{10} \frac{n_i}{n} \quad (2)$$

Por exemplo, tendo a seguinte distribuição de métricas normalizadas:

Valores Normalizados	e_1	e_2	e_3	e_4	e_5
Métrica 1	2	1	2	2	2
Métrica 2	3	2	2	2	3

Sendo o evento e_5 o último a ser inserido na lista, os valores das variáveis seriam:

ISBN 978-85-62830-10-5

n_i	1	Existe um evento igual ao evento e_5
n	5	Tamanho da janela
v	4	Valor resultante para o cálculo

A entropia seria calculada da seguinte forma:

$$H(E) = -\left\{\left[1/5 * log_{10}\left(1/5\right)\right] + \left[1/5 * log_{10}\left(1/5\right)\right] + \left[1/5 * log_{10}\left(1/5\right)\right] + \left[1/5 * log_{10}\left(1/5\right)\right]\right\} \quad (3)$$

5. VALIDAÇÃO DA METODOLOGIA DE DETECÇÃO

Caracterização do Experimento

Os experimentos consistiram em avaliações de métricas obtidas através de uma rede local de computadores. Foi utilizado um software para gerar um fluxo unidirecional entre um servidor e um cliente conhecido. O fluxo foi configurado para apresentar variações de largura de banda, induzindo anomalias no tráfego.

A característica do fluxo foi determinada para simular limiares de uma aplicação de *streaming* de vídeo, usando valores de largura de banda e latência. O servidor enviou pacotes para o cliente em um período de 12 minutos, intercalando anomalias a cada 2 minutos. Elas eram alternadas entre de alta e baixa taxa de transmissão.

As métricas pacote-a-pacote foram coletadas em arquivos e utilizadas como entrada posteriormente no analisador baseado em entropia. Como dito, o analisador é bastante flexível e permite que os dados sejam analisados também de forma *off-line*.

Resultados Obtidos

Os resultados obtidos demonstraram que são necessários mais experimentos para ter-se uma melhor escolha dos parâmetros de detecção e com isso poder ajustar o analisador para melhores resultados. A captura de métricas pacote-a-pacote também faz com que seja necessária uma janela de observação muito grande e valores de *m* e *r* compatíveis para que as variações sejam percebidas. Uma grande janela de observação causa um uso de memória principal não esperado para as propostas iniciais do analisador. Os valores sugeridos por Wang, Talwar, Schwan e Ranganathan [11] foram utilizados para métricas de naturezas diferentes, fazendo com que diferentes ajustes sejam necessários.

6. CONCLUSÕES

Monitorar um enlace entre dois pontos de uma rede de computadores e identificar as aplicações presentes no tráfego em tempo real, quer seja por técnicas baseadas em padrões de assinaturas de tráfego ou por meio de métodos probabilísticos de análise de comportamento, são atividades custosas. Portanto, para que o monitoramento seja realizado na prática, pode ser necessário optar por amostragem de pacotes ou agregação do tráfego em fluxos, causando perda de informações relevantes, o que pode levar a um impacto significativo na completude e precisão dos resultados.

Sistemas de computação em nuvem apresentam características particulares que tornam o monitoramento dos enlaces e a classificação do tráfego ainda mais desafiadora, como larga escala, variabilidade de carga, inúmeros serviços diferentes e, especialmente quando são monitorados enlaces de provedores de computação em nuvem, há alta vazão de tráfego.

O termo anomalia é bastante genérico e abrange ataques, tráfego indesejado na rede devido a aplicações com algum erro de configuração, perda de pacotes, injeção de tráfego de aplicações não permitidas, entre outros. Portanto, é necessário restringir o escopo relacionado à detecção de

ISBN 978-85-62830-10-5

anomalias para que esta área seja estudada no contexto de computação em nuvem e que novas soluções possam ser propostas, ou melhorias a soluções existentes.

Uma forma de tornar factível o estudo de anomalias em sistemas de computação em nuvem é analisar as metas do negócio e verificar se o tráfego de rede está adequado ao que foi negociado entre as partes. Uma possível abordagem para essa detecção de anomalias é analisar o tráfego de rede observando se os limites acordados para as métricas dos SLAs negociadas estão sendo garantidos e alertar possíveis desvios a esses acordos.

Como trabalho futuro, espera-se avaliar melhor como calibrar o mecanismo de detecção, propondo otimizações para as estimativas de parâmetros de modo a obter melhores resultados para a completude e acurácia do mecanismo de detecção.

REFERÊNCIAS

[1] Mikkilineni, Rao; Sarathy, Vijay, Cloud Computing and the Lessons from the Past. In Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops, July, 2009.

[2] Voas, J. and Zhang, J. 2009. Cloud Computing: New Wine or Just a New Bottle?, IT Professional 11, 2, March, 2009.

[3] Buyya, R., et al, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems. Volume 25, Number 6, Pages: 599-616, June 2009.

[4] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Matei Zaharia, Above the Clouds: A Berkeley View of Cloud Computing, Technical Report No. UCB/EECS-2009-28, Univ. of California, Berkeley, Feb 10, 2009.

[5] Grossman, Robert L., The Case for Cloud Computing, IT Professional, vol. 11, no. 2, pp. 23-27, Mar./Apr. 2009

[6] Hutchinson, C., Ward, J., Castilon, K., Navigating the Next-Generation Application Architecture, IT Professional, vol. 11, no. 2, pp. 18-22, Mar./Apr. 2009.

[7] Phillip A. Laplante, Jia Zhang, Jeffrey Voas, What's in a Name? Distinguishing between SaaS and SOA, IT Professional, vol. 10, no. 3, pp. 46-50

[8] Goyal, P. The Virtual Business Services Fabric: an integrated abstraction of Services and Computing Infrastructure - First International IEEE workshop on Cloud Computing IEEE, 2009.

[9] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18, May 2010.

[10] Chengwei Wang. 2009. EbAT: online methods for detecting utility cloud anomalies. In *Proceedings of the 6th Middleware Doctoral Symposium* (MDS '09). ACM, New York, NY, USA, Article 4 , 6 pages. DOI=10.1145/1659753.1659757 http://doi.acm.org/10.1145/1659753.1659757

[11] Chengwei Wang; Talwar, V.; Schwan, K.; Ranganathan, P.; , "Online detection of utility cloud anomalies using metric distributions," *Network Operations and Management Symposium (NOMS), 2010 IEEE* , vol., no., pp.96-103, 19-23 April 2010. DOI: 10.1109/NOMS.2010.5488443. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5488443&isnumber=5488304.

[12] Christof Weinhardt, Arun Anandasivam, Benjamin Blau, and Jochen Stößer. 2009. Business Models in the Service World. *IT Professional* 11, 2 (March 2009), 28-33. DOI=10.1109/MITP.2009.21 http://dx.doi.org/10.1109/MITP.2009.21

[13] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia. A View of Cloud Computing. *Magazine Communications of the ACM*. Volume 53 Issue 4, April 2010. New York, NY, USA. Doi:10.1145/1721654.1721672

[14] Callado, A.; Kamienski, C.; Szabo, G.; Gero, B.; Kelner, J.; Fernandes, S.; Sadok, D. "A Survey on Internet Traffic Identification," *Communications Surveys & Tutorials*, IEEE , vol.11, no.3, pp.37-52, 3rd Quarter 2009. Doi: 10.1109/SURV.2009.090304 - URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5208732&isnumber=5208728

ISBN 978-85-62830-10-5

[15] Arthur Callado, Judith Kelner, Djamel Sadok, Carlos Alberto Kamienski, Stenio Fernandes.Better network traffic identification through the independent combination of techniques, *Journal of Network and Computer Applications*, Volume 33, Issue 4, July 2010, Pages 433-446, ISSN 1084-8045, DOI: 10.1016/j.jnca.2010.02.002.

ISBN 978-85-62830-10-5