

AMBIENTE PARA EXPERIMENTAÇÃO E AVALIAÇÃO DE REDES DE COMPUTADORES VIRTUALIZADAS

Esdras Barbosa Lima da Silva Júnior¹; Eduardo Antônio Guimarães Tavares²

¹Estudante do Curso de Engenharia da Computação – CIN – UFPE; E-mail: eblsj@cin.ufpe.br,

²Docente/pesquisador do Depto de Informática – CIN – UFPE. E-mail: eagt@cin.ufpe.br.

Sumário: Este artigo apresenta a experimentação de medição do consumo de redes virtualizadas comparada ao uso de recursos por estas redes, na tentativa de encontrar alguma relação entre ambos.

Palavras-chave: consumo de energia; virtualização de redes; utilização de recursos

INTRODUÇÃO

Ao longo dos anos, a *Internet*, rede mundial de computadores, tem provido um ambiente bastante favorável para adoção de diferentes tecnologias de redes de computadores e construção de aplicações distribuídas. Todavia, devido à existência de várias organizações participantes com interesses e políticas conflitantes, modificações na arquitetura atual da *Internet* são extremamente difíceis de serem executadas [1]. Alguns autores denominam tal situação como ossificação [2], e, geralmente, utilizam a baixa adoção do IPv6 na rede mundial de computadores como exemplo representativo [1].

Para lidar com os problemas previamente citados, recentemente, a comunidade científica desenvolveu o conceito de virtualização de redes [2], no sentido que diversas instâncias de redes de computadores possam coexistir em uma mesma infraestrutura física. Provedores de serviço (*Service Providers*), são responsáveis em atender requisições dos usuários, as quais são convertidas em redes virtuais usando um subconjunto da infraestrutura física do provedor (denominada substrato [2]).

Pesquisadores têm concentrado esforços na criação de algoritmos para realizar o mapeamento das requisições do usuário na infraestrutura física de um provedor de serviço [1,3,4,5]. O foco usualmente está associado ao bom uso dos recursos através de técnicas que satisfaçam critérios de balanceamento. Todavia, os algoritmos existentes desconsideram alguns critérios, tais como dependabilidade (e.g., confiabilidade, disponibilidade) [6] e consumo de energia, impactando a maneira que a rede virtual é criada. Ademais, esses algoritmos não foram avaliados em um ambiente de experimentação a fim de validar os resultados obtidos.

MATERIAIS E MÉTODOS

O ambiente que optamos utilizar para virtualização de redes é o VMware Player, por ser uma ferramenta gratuita e eficiente. Para realizarmos as medições utilizamos o EMeter EVM430-F6736 da Texas Instruments, que utiliza o microcontrolador MSP430. Para edições no código de medição usamos o IAR EW com um código em linguagem C. Fizemos o uso também das ferramentas NetBeans IDE com o jdk 1.8.0 para escrever um código em java que receba os dados de medição e colete informações de uso da máquina.

Primeiro deveríamos colocar o medidor (EMeter) para medir valores de corrente e de tensão para podermos utilizar esses valores e tirarmos o consumo de qualquer equipamento. Felizmente já era disponibilizado um código teste de medição que já realizava as medições e cálculos necessários, calculando inclusive o próprio consumo.

A partir daí começamos a trabalhar no envio desses dados de consumo para uma máquina, através da porta serial do medidor. Para isso estudamos a comunicação serial no MSP430 com a finalidade de criarmos nossa própria rotina de envio de mensagens, porém vimos que o código já continha algumas configurações de interrupção que não permitia a simples utilização de nossas rotinas. Contudo descobrimos uma série de rotinas de transmissão serial já implementadas no próprio código, então testamos e realizamos os devidos ajustes para enviarmos os dados pela saída serial.

Para estabelecer a comunicação, faltava apenas criar um código para receber os dados no computador. Para isso, utilizamos a linguagem Java (NetBeans IDE) com uma API para a porta serial (java comm.). Fizemos modificações num código exemplo para receber os dados mandados pelo medidor. Estabelecendo a comunicação, começamos a salvar os dados de medição recebidos em um array para usá-los posteriormente com dados de consumo de recursos (CPU, memória, disco, etc).

Para coletarmos o uso de recursos do sistema, usamos uma API em java (Sigar), com a qual conseguimos obter dados da(s) CPU(s), da memória, acessos ao disco e dados de rede. Integramos esse código ao código de comunicação.

Após a análise da comunicação, foi observado que o tempo entre as mensagens enviadas com os dados de medição estava muito alto (cerca de um segundo) para o medidor, então devemos reduzir esse tempo estudando e modificando o necessário no código da medição.

RESULTADOS

Foi verificado que o código de teste que realiza a medição corretamente e foram identificadas as principais rotinas do código de medição e as estruturas que guardam as informações da medição. Também foram detectadas e devidamente modificadas as rotinas de transmissão de dados via porta serial, para o envio dos dados.

O código java implementado recebe os dados pela porta serial e coleta informações de CPU(s), memória, disco e rede.

DISCUSSÃO

Com os dados fornecidos pelo medidor e os dados coletados do sistema, já foi possível fazer a comparação entre o consumo e o uso de recursos da máquina.

CONCLUSÕES

Após estabelecer a comunicação de dados de consumo, realizamos alguns testes com um notebook com Windows 7 instalado. Verificamos que quanto maior o uso da CPU e de memória física, mais alto era o consumo desse notebook, confirmando que quanto maior o uso de recursos, maior a energia consumida.

AGRADECIMENTOS

Ao PIBIC/UFPE por ter financiado esse projeto em forma de bolsa de IC, ao professor orientador, Eduardo Tavares, que em todo período esteve disponível para quaisquer dificuldades, ao professor do IFPE-Recife, Meuse de Nogueira, por seu auxílio nos testes e esclarecimentos acerca de todo trabalho.

REFERÊNCIAS

[1] Belbekkouche , A. et al. Resource Discovery and Allocation in Network Virtualization. IEEE Communications Surveys & Tutorials, n. 99, 2012.

- [2] Gregor, S., et al. Network virtualization architecture: proposal and initial prototype. In: Workshop on Virtualized Infrastructure Systems and Architectures, Barcelona, 2009. Anais... Barcelona: ACM. p. 63-72.
- [3] Rahman, M. et al. Survivable Virtual Network Embedding. Lecture Notes in Computer Science: Network 2010, v. 6091, p. 40-52, 2010.
- [4] Yu, M. et al. Rethinking virtual network embedding: substrate support for path splitting and migration. ACM SIGCOMM Computer Communication Review, v. 38, n. 2, p. 17-29, 2008.
- [5] Yeow, W. et al. Designing and embedding reliable virtual infrastructures. ACM SIGCOMM Computer Communication Review, v. 41, n. 2, p. 57-64, 2011.
- [6] Meyer, J. F. On Evaluating the Performability of Degradable Computing Systems. IEEE Transactions on Computers. v. 29, n. 8, 1980.
- [7] Lira, V. et al. Virtual Network Resource Allocation Considering Dependability Issues. In: 13o International Conference on Computer and Information Technology (CIT), Sydney, 2013. Anais... Sydney: IEEE. p. 0-0.