

VM-MON: uma ferramenta multiplataforma para monitoramento de sistemas virtualizados

Fillipi de P. Suszek¹, Ricardo M. Czekster¹, Thais Webber¹, César Marcon², Rodrigo Nedel¹

¹Departamento de Informática – Universidade de Santa Cruz do Sul (UNISC)
Avenida Independência, 2293 – CEP 96815-900 – Santa Cruz do Sul – RS – Brasil

ricardoc@unisc.br

²Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Avenida Ipiranga, 6681 – CEP 90619-900 – Porto Alegre – RS – Brasil

Abstract. *The choice of virtualized solutions for systems execution under different contexts has been a commonly used approach for several domain applications. The utilization of virtualized environments is relevant and crucial, however, in several cases, those technologies are chosen with disregard as to performance or other nonfunctional attributes such as Quality-of-Service (QoS), resilience, reliability and scalability. The objective of the present work consists on the conception of a monitoring tool to assess virtualized systems. The idea is to estimate better software and hardware configurations to virtualized platforms without performance decay.*

Resumo. *A escolha de soluções virtualizadas para executar diferentes sistemas em diferentes contextos tem sido uma abordagem comumente adotada para uma gama de aplicações. O uso de ambientes virtualizados é relevante e crucial, entretanto, em muitos casos estas tecnologias são escolhidas sem verificação prévia de desempenho ou outros atributos não funcionais, tais como a garantia de qualidade de serviço (QoS), resiliência, confiabilidade e escalabilidade. O objetivo deste trabalho consiste na concepção de uma ferramenta para monitorar e inspecionar sistemas virtualizados. A ideia é poder estimar as melhores configurações de software e hardware para plataformas virtualizadas sem decaimento de desempenho.*

1. Introdução

O objetivo deste trabalho é conceber uma nova ferramenta chamada VM-MON para monitorar e inspecionar sistemas virtualizados que possibilite o estudo de configurações para detectar, estimar e prever degradações de desempenho de sistemas virtualizados e explorar aspectos de escalabilidade. Especificamente, foca-se na problemática de virtualização de servidores (*Server Virtualization*), trabalhando com máquinas virtuais executadas sobre um SO (Sistema Operacional), tais como plataformas MS-Windows, GNU/Linux ou outra, e.g. Unix. Os principais exemplos de software que permitem a instalação e configuração de máquinas virtuais (hipervisores) são: *vSphere*TM (VMware), *XenServer* (Citrix), *Hyper-V*TM (Microsoft), *VirtualBox*TM (Oracle) e outras técnicas baseadas em *Kernel-based Virtual Machine* (KVM). A principal contribuição é a possibilidade de integração de ferramentas gratuitas já existentes nos SOs, provendo uma interface gráfica para usuários administradores que desejam utilizar virtualização e Computação na Nuvem nas suas organizações (ACETO 2013, BARHAM 2003, BUYYA 2010, YOUNGE 2011, WARD 2014).

2. Descrição da ferramenta VM-MON

A VM-MON foi desenvolvida em Java™, uma linguagem multiplataforma e amplamente utilizada para o desenvolvimento de sistemas. Isso possibilita que a VM-MON seja executada em qualquer SO. Nesta versão, o foco é dado para máquinas GNU/Linux e MS-Windows. O hipervisor escolhido para executar a VM-MON foi o *Virtual Box*, da *Oracle*, devido à sua gratuidade e facilidade de instalação e operação.

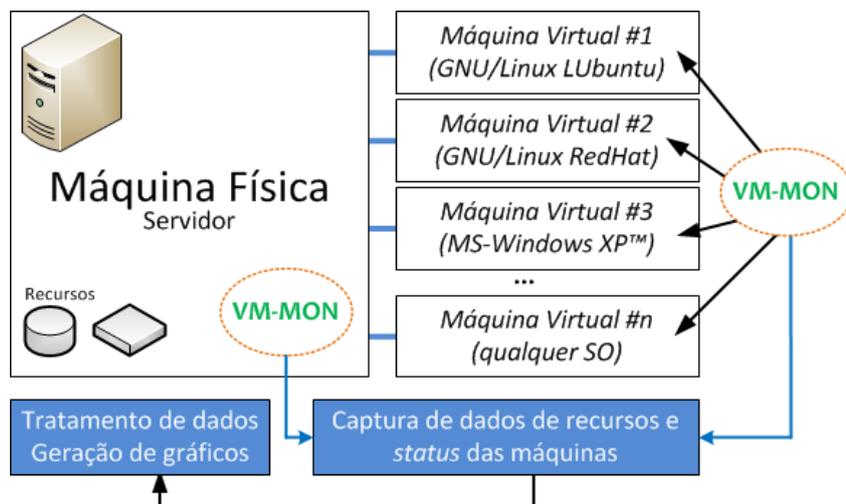


Figura 1: Execução da ferramenta VM-MON em contextos virtualizados.

A Figura 1 mostra os elementos monitorados pela VM-MON em contextos físicos e virtualizados. A ferramenta executa múltiplas instâncias (uma na máquina física e uma em cada máquina virtual). Apresenta uma GUI (*Graphical User Interface*) para comandos não triviais de monitoramento para diferentes plataformas. A VM-MON possui opções para escolher os dispositivos que serão monitorados e o tempo de monitoramento. A partir disso, a ferramenta gera scripts para os SOs específicos com os comandos para as ferramentas auxiliares serem executadas. Quando esta operação termina, o sistema permite que o usuário inicie o processo de análise, executando *scripts* para transformar dados brutos em dados legíveis pelas ferramentas gráficas. Por fim, a ferramenta permite que uma nova sessão de análise seja feita pelo usuário, pois organiza os arquivos em pastas específicas, salvando os dados para análises históricas.

2.1. Dispositivos de monitoramento e ferramentas auxiliares

Os dispositivos escolhidos para efeitos de monitoramento pela VM-MON são: a memória principal (*memory*), a CPU (*Central Processing Unit*), o HD (*Hard Disk*) e as interfaces de comunicação em rede (*Ethernet Interfaces*). No GNU/Linux, foram escolhidas as ferramentas auxiliares de monitoramento: **ps** (*process snapshot*), **vmstat** (*virtual memory statistics*), **iostat** (*input/output statistics*) e **netstat** (*network statistics*). Parâmetros de cada ferramenta auxiliar são utilizados para extrair os dados para análise. No MS-Windows, adotam-se os contadores de desempenho (*performance counters*) através de chamadas específicas de linha de comando. Para processar os dados gerados por cada ferramenta auxiliar foi utilizada a Linguagem de *Scripting Perl* (*Practical Extraction and Report Language*), e para construir os gráficos para análise utilizou-se a ferramenta da biblioteca GNU **gnuplot**. Tanto *Perl* quanto *gnuplot* são ferramentas compiladas para GNU/Linux e MS-Windows, não necessitando

de quaisquer conversões e recompilações. A Tabela 1 indica os principais contadores de desempenho utilizados para monitorar as máquinas MS-Windows.

Tabela 1: Listagem de alguns contadores de desempenho do MS-Windows.

Contador de Desempenho do MS-Windows – perfconfig.txt
"\Memory\Available Bytes"
"\Server\Bytes Total/Sec"
"\Paging File(\??\C:\pagefile.sys)\% Usage"
"\PhysicalDisk(_Total)\% Disk Time"
"\Process(java)\Working set"
"\Processor(_Total)\% Processor Time"
"\Thread(_Total/_Total)\Context Switches/sec"
"\System\Processor Queue Length"

Os contadores criados observam o tempo do disco (*Disk Time*), o Conjunto de Trabalho (*Working Set*) de processos (e.g java), o tempo do processador (*Processor Time*), as trocas de contexto por segundo (*Context Switches/sec*) e o tamanho da fila do processador (*Processor Queue Length*). Outros contadores podem ser definidos.

2.2. Detalhes de implementação

A VM-MON executa comandos independentes de plataforma sempre que possível (e.g. criar um diretório que é um processo depende do SO e Java possui implementações multiplataforma destinadas a este propósito), em conjunção com chamadas de sistema (que executam os *scripts* particulares de cada SO).

A ferramenta emprega diversas linguagens e ferramentas auxiliares, por exemplo, utiliza no GNU/Linux *scripts bash* (.sh), e no MS-Windows, utiliza arquivos de lote (.bat). A ferramenta processa os dados em Perl, gerando arquivos com os resultados tabulados em colunas que representam os dados requisitados pelos usuários e então dispara comandos para gerar gráficos e permitir a análise. VM-MON foi desenvolvida utilizando-se Java *Swing* que fornece uma API (*Application Programming Interface*) rápida e fácil de programar para a adição de funcionalidades futuras.

2.3. Principais comandos e opções da ferramenta

VM-MON realiza chamadas de sistema para outras ferramentas auxiliares provendo uma interface gráfica. Tabela 2 mostra os SOs e as chamadas realizadas internamente pela VM-MON para capturar resultados de monitoramento nas duas plataformas.

Tabela 2: Comandos disparados internamente na VM-MON com chamadas de sistema.

Sistema Op.	Ferram.	Comando de disparo da ferramenta auxiliar
MS-Windows	logman	logman create counter dataCollection -s SERVER1 -cf perfconfig.txt -o \\path\perfmon\%COMPUTERNAME%_dataCollection -v ddMMhhmm -f csv -si 10 -u admin admin
		logman start dataCollection -s SERVER1 -u admin admin
		logman stop dataCollection -s SERVER1 -u admin admin
		logman delete dataCollection -s SERVER1 -u admin admin
GNU/Linux*	iostat	iostat -x \$sleeptime \$iter >> io.dat
	vmstat	vmstat \$sleeptime \$iter >> vm.dat
	ps	ps -eo"pcpu,pmem,time,comm" sort -r head -11 tail -n 10 >> ps.dat
	netstat	netstat -i grep eth >> neth.dat

(*) Estes são *exemplos* de comandos possíveis. As variáveis \$sleeptime, \$iter vem das opções dos usuários.

A ferramenta **logman** usualmente vem instalada com o MS-Windows (com a **perfmon.exe**). No GNU/Linux necessita da instalação do pacote **sysstat** (*System Statistics*), já para o MS-Windows criam-se os contadores de desempenho via *scripts* de inicialização (.bat), que criam contadores (*create counter*) usando a lista de contadores de **perfconfig.txt** (vide Tabela 1), com formato de arquivo específico (DiaMêsHoraMin, ou **ddMMhhmm**), tipo (*comma separated values*, ou CSV), para um servidor específico (SERVER1), com intervalo de monitoramento e um usuário e senha (-u). Também é possível começar um monitoramento (*start*), parar (*stop*) e excluir uma coleção de dados monitorados (*delete*). No GNU/Linux, existe um laço que só termina quando o tempo de monitoramento especificado pelo usuário acaba. Cabe ressaltar que um arquivo por ferramenta auxiliar é sempre criado (arquivo .dat) em uma pasta específica e a cada iteração do laço novos dados são adicionados (*appended*).

2.4. Execução da VM-MON

A ideia da VM-MON é ser leve (*lightweight*), rápida e fácil de configurar os monitoramentos necessários. Para que a ferramenta seja utilizada por uma quantidade considerável de usuários, sua interface foi implementada em inglês. A Figura 3 mostra algumas telas principais da VM-MON e sua execução em um contexto virtualizado que contém uma máquina executando o GNU/Linux com a distribuição LUbuntu. O usuário pode configurar monitoramentos (aba *Entity*), opções (aba *Options*), gerar gráficos (aba *Plot*), ver resumo das escolhas (aba *Overview*) e ajuda (aba *Help*). A Figura 3(a) mostra opções de monitoramento de disco tais como estatísticas gerais de Entrada e Saída (*I/O Statistics*), trocas de contexto (*Context Switches*), número total de threads (*Number of Threads*) e informações de processos (*Process Information*). A Figura 3(b) mostra algumas opções para monitorar os discos existentes, como monitoramento dos discos (*Monitor All Disk Devices*), monitoramento total dos discos (*Full monitoring*), realizando a captura de dados de escrita/leitura. Também permite que sejam capturadas apenas informações de leitura (*Monitor Only Reads*) ou escrita (*Monitor Only Writes*). A Figura 3(c) mostra as opções para monitoramento da memória.

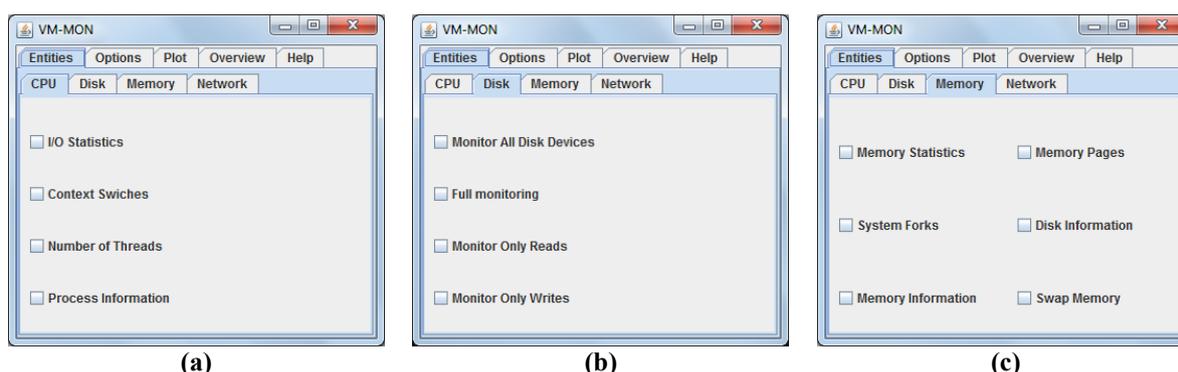


Figura 3: Tela principal da ferramenta VM-MON, mostrando a aba 'Entities' e os monitoramentos da CPU em (a), monitoramentos no disco rígido em (b) e memória em (c).

A Figura 4 mostra a aba *Entities* e em (a) as opções possíveis para monitorar a Rede (*Network*) tais como informações gerais das interfaces (*General Network Information*), captura de dados de passagem pelas interfaces (*Dump all network interfaces*), monitoramento de todas as interfaces (*Monitor all network interfaces*) e monitoramento de pacotes (*Monitor Network Packets Information*).

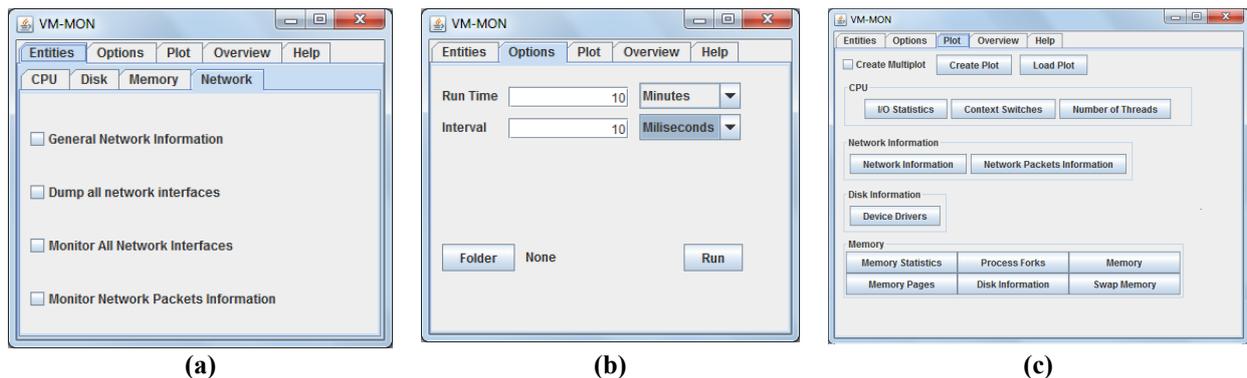


Figura 4: Opções de Redes em (a), opções gerais em (b) e de gráficos em (c).

Esta figura ainda mostra as abas *Options* e *Plot* (as abas *Overview* e *Help* são utilizadas depois que o usuário escolheu o que deseja monitorar e correspondem a resumos específicos dos comandos bem como um manual de ajuda). A Figura 4(b) mostra as opções de execução do *script* de monitoramento, onde se configura o tempo de monitoramento (*Run Time*), o intervalo de monitoramento (*Interval*) e a pasta que se deseja salvar os arquivos. É nesta aba que o usuário inicia a execução dos monitoramentos escolhidos no botão executar (*Run*). A Figura 4(c) mostra as opções de gráficos conforme as coletas efetuadas. É nesta aba que a VM-MON executa o *script* Perl que extrai os dados e constrói arquivos que serão usados nos gráficos *gnuplot*.

3. Considerações finais

Ferramentas de monitoramento são vitais para entender a execução de sistemas e prever gargalos e problemas de desempenho. A VM-MON tenta oferecer uma alternativa aos problemas de monitoramento oferecendo uma ferramenta multiplataforma que integra conceitos de diferentes SOs. VM-MON quando utilizada em modo *stand-alone* em uma determinada arquitetura funciona como uma GUI de monitoramento para a parte física bem como a parte virtualizada da solução. A VM-MON permite a adição de novas ferramentas e parâmetros de forma fácil e intuitiva. A VM-MON está disponível em <http://www.unisc.br/portal/pt/cursos/mestrado/mestrado-em-sistemas-e-processos-industriais/software-para-download.html>

Referências

- ACETO, G., BOTTA, A., de DONATO, W., PESCAPÈ, A.: *Cloud monitoring: A survey*. In: Computer Networks, vol. 57(9), pp. 2093-2115, 2013.
- BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T., HO, A., NEUGEBAUER, R., PRATT, I., WARFIELD, A.: *Xen and the art of virtualization*. In: ACM SIGOPS Operating Systems Review, vol. 37(5), pp. 164-177, 2003.
- BUYYA, R., BROBERG, J., GOSCINSKI, A. M.: *Cloud computing: Principles and paradigms*. vol. 87. Wiley, 2010.
- YOUNGE, A. J., HENSCHER, R., BROWN, J.T., von LASZEWKSI, G., QIU, J., FOX, G. C.: *Analysis of Virtualization Technologies for High Performance Computing Environments*, In: 2011 IEEE CLOUD, pp. 9-16, 2011.
- WARD, J. S., BARKER, A.: *Observing the clouds: a survey and taxonomy of cloud monitoring*. In: Journal of Cloud Computing 3(1), pp. 1-30, 2014.