

Analizando o uso da Ferramenta de Monitoramento Zabbix para Ambientes Paralelos

Bruno S. Marques, Guilherme Feyh, Mateus R. Aubin, Rodrigo da Rosa Righi

¹Universidade Vale do Rio dos Sinos (UNISINOS)

São Leopoldo – RS – Brasil

contato@brunomarques.com.br, {guilherme.feyh,mateus.aubin}@gmail.com

rrrighi@unisinis.br

Resumo. *Monitoramento de sistemas paralelos é uma ação chave para obter desempenho em aplicações, visto que é pertinente para a escolha de nós para o lançamento dos processos e posterior análise de gargalos durante a execução deles. Nesse sentido, esse artigo discute o uso da ferramenta Zabbix, bem conhecida para administração de redes locais, no âmbito de ambientes paralelos. Os testes mostram que a ferramenta possui baixa intrusividade quanto ao volume de dados na rede e pode ser configurada para atender ambientes de grid.*

1. Introdução

Redes de computadores têm se tornado cada vez mais importantes para o funcionamento de sistemas de larga escala [Kuwabara et al. 2012]. Em especial, no ramo da computação paralela é possível agregar o poder computacional de supercomputadores, redes locais e clusters para formar uma arquitetura de grid com maior poder de processamento. No momento que o sistema cresce em escala, também cresce a complexidade de administração e monitoramento das máquinas que compõem o ambiente paralelo. Uma vez que grids comumente possuem ligações de rede mais onerosas e computadores heterogêneos, um sistema de monitoramento pode ser útil para indicar quais os nós mais propensos a receber processos. Em adição aos benefícios no momento do escalonamento, o monitoramento também é pertinente para a detecção de anomalias, de modo que o administrador do ambiente paralelo possa tomar decisões para a reobtenção do estado normal.

Nesse contexto, esse artigo apresenta uma avaliação da ferramenta de monitoramento Zabbix num ambiente distribuído. Uma das configurações suportadas pelo Zabbix é aquela baseada em procuradores (*proxies*), que é importante para ambientes de grid. Cada componente do grid tem um procurador que reúne informações sobre os recursos sob sua jurisdição e as repassa para um nó centralizador. Assim, o artigo descreve brevemente a ferramenta Zabbix e a compara com trabalhos relacionados e apresenta os seus dois modos de operação, de forma totalmente centralizada e com procuradores. Os testes se concentram em analisar as variáveis monitoradas pelo Zabbix e a análise de sua intrusividade quanto ao funcionamento da rede de comunicação.

2. Sistemas de Monitoramento para Cluster

Os sistemas de monitoramento mais tradicionais para redes locais são o Nagios e o Zabbix [White et al. 2012]. Já o Ganglia é frequentemente usado para o âmbito de clusters. Essa seção discute a atuação de Zabbix frente a seus concorrentes.

2.1. Zabbix

O Zabbix se destaca pelo monitoramento de recursos dos nós da rede, como suas interfaces de rede, o uso de memória e a carga de CPU. Em adição, é possível capturar informações sobre o tempo que a máquina permanece ligada, espaço livre e velocidade média de gravação de dados em disco e programas em execução. Para que se tenha acesso a tais informações é necessária a instalação de um programa em cada nó monitorado, chamado de Agente Zabbix. Ele coleta as informações de tempos em tempos (conforme configurado) e as prepara para envio ao servidor central. O agente também pode instalar/desinstalar programas e executar scripts remotamente.

2.2. Ganglia

O Ganglia é uma ferramenta similar ao Zabbix, muito usada na área acadêmica para monitoramento de sistemas distribuídos de alto desempenho. Este software está disponível para diversos sistemas operacionais e atualmente é usado em clusters ao redor do mundo. Diferentemente do Zabbix, o Ganglia carece de um sistema nativo de notificações, sendo esse um dos seus pontos negativos. Na versão atual do Ganglia (3.5.2), não é possível a inclusão de contadores personalizados, nem a identificação de hosts por outras formas que não o hostname. Por fim, o Ganglia usa multicast para a tarefa de monitoramento, o que facilita sua instalação em redes pequenas, mas pode aumentar substancialmente a sua intrusividade em redes de grande porte. Essa característica limita o seu uso em grids compostos de diferentes sub-redes.

2.3. Nagios

Enquanto o Zabbix é considerado uma ferramenta mais completa e abrangente, o Nagios é tido como mais customizável e capaz de cobrir as mesmas funcionalidades do seu concorrente através de plug-ins [Issariyapat et al. 2012][Enterprises 2011]. A dependência desta ferramenta por tais plug-ins é tamanha que ela não possui mecanismos que a permitam executar, por si própria, as tarefas de monitoramento de rede [Katsaros et al. 2011].

3. Arquitetura de avaliação

O Zabbix permite diferentes configurações de arquitetura. Esta seção aborda as arquiteturas utilizadas na execução dos testes, ressaltando as diferenças entre elas.

3.1. Configuração típica

O Zabbix normalmente é configurado utilizando somente um servidor central. Ou seja, os agentes comunicam-se com um único equipamento que se encarrega de manipular e gravar os dados recebidos. Esta configuração tem como principal vantagem a facilidade de configuração, já que o ambiente é centralizado, mas pode apresentar problemas de escalabilidade possuindo um único ponto de falha e um limite de máquinas que podem ser monitoradas. A figura abaixo exemplifica a arquitetura com um único servidor.

3.2. Zabbix Proxy

Para evitar o congestionamento da rede ou diminuir o overhead existe o Zabbix Proxy. Nesta arquitetura o servidor central possui o auxílio de no mínimo um agente modificado, chamado de procurador (*proxy*). Este agente não necessita de tanto poder de processamento quanto o servidor central, apenas de um pequeno banco de dados local. A sua

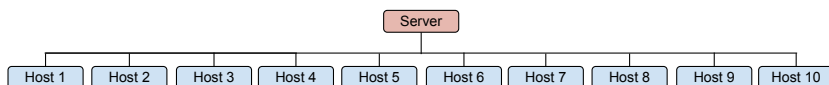


Figura 1. Arquitetura básica utilizando apenas um servidor

função é reter as informações enviadas pelos demais agentes por um determinado período e entregá-las em intervalos de tempo maiores ao servidor central [Olups 2010]. Assim o servidor central não fica sobrecarregado e a qualidade do monitoramento não é afetada, visto que não há perda de dados. Neste modo não há comunicação direta entre o servidor central e os hosts.

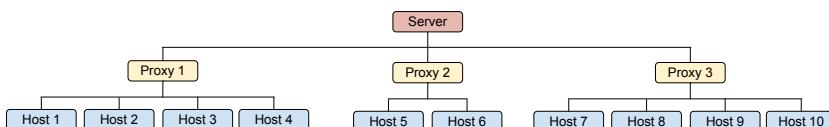


Figura 2. Arquitetura básica com uso de três proxies e um servidor central

4. Testes Preliminares

Os testes foram executados a partir de uma rede local da Unisinos composta por 14 computadores. Ela foi mapeada para duas configurações: (i) monitoramento com servidor centralizado; (ii) uso da arquitetura de três procuradores. A figura 2 ilustra a configuração com procuradores. Ambas as configurações permitem a análise de memória, CPU e disco nas 14 máquinas (inclusive do servidor). Outro teste realizado contempla a análise da intrusividade do Zabbix no uso normal da rede. Para tal, todas as máquinas da rede local foram ligadas e nenhuma carga de processamento ou comunicação foi aplicada. Os resultados podem ser vistos na figura 3(b).

Na configuração com servidor centralizado, tem-se uma média de 13.3 *kbits/s* tanto para as atividades de entrada quanto de saída da placa de rede do servidor. Já para aquela que usa *proxies*, obtém-se uma média de 11.3 *kbits/s* para entrada e 13.8

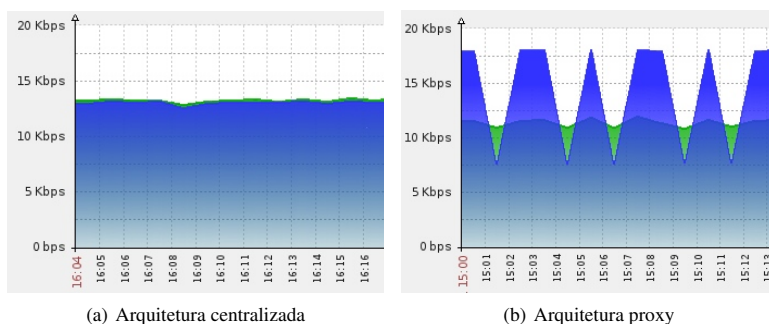


Figura 3. Tráfego de rede no Zabbix Server.

kbits/s para saída da placa de rede. Uma vez que os *proxies* são passivos, o servidor faz consultas periódicas aos *proxies* para verificar se estão ativos e obter dados atualizados. Na configuração usando procuradores percebe-se um menor uso de banda para a entrada de dados. Isso é justificável pelo fato de que, nessa arquitetura, tem-se a recepção de dados de somente três estações (*proxies*) ao invés de 13, como na arquitetura centralizada. Por fim, outro teste realizado foi o desligamento de um dos *proxies*. A partir desse momento, o servidor não consegue coletar informações dos hosts subordinados.

5. Conclusão

Esse artigo apresentou testes iniciais da ferramenta Zabbix em uma rede local com diferentes configurações. Em especial, a configuração com *proxy* é pertinente para emular o funcionamento de um grid, onde se tem diferentes redes cada qual com o seu *proxy*. Essa característica faz com que Zabbix seja apropriada para grids. Além disso, essa constatação é embasada no fato de Zabbix apresentar uma baixa intrusão na carga da rede.

Referências

- Enterprises, N. (2011). How nagios compares to zabbix. http://assets.nagios.com/datasheets/compare/How_Nagios_Compares_To_Zabbix.pdf.
- Issariyapat, C., Pongpaibool, P., Mongkolluksame, S., and Meesublak, K. (2012). Using nagios as a groundwork for developing a better network monitoring system. In *Technology Management for Emerging Technologies (PICMET), 2012 Proceedings of PICMET '12:*, pages 2771–2777.
- Katsaros, G., Ku andbert, R., and Gallizo, G. (2011). Building a service-oriented monitoring framework with rest and nagios. In *Services Computing (SCC), 2011 IEEE International Conference on*, pages 426–431.
- Kuwabara, S., Shimizu, K., and Maruyama, M. (2012). Adaptive network monitoring system for large-volume streaming services in multi-domain networks. In *World Telecommunications Congress (WTC), 2012*, pages 1–6.
- Olups, R. (2010). *Zabbix 1.8 Network Monitoring*. Packt Publishing Ltd., 1th edition.
- White, K. J. S., Pezaros, D. P., and Johnson, C. W. (2012). Increasing resilience of atm networks using traffic monitoring and automated anomaly analysis. In *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems, ATACCS '12*, pages 82–92, Toulouse, France, France. IRIT Press. <http://dl.acm.org/citation.cfm?id=2325676.2325687>.