

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
CURSO DE ESPECIALIZAÇÃO EM REDES DE COMPUTADORES

ITAMIR DE MORAIS BARROCA FILHO

**CONFIGURAÇÃO DO ZABBIX PARA O MONITORAMENTO DO SERVIDOR DE
APLICAÇÃO *JBOSS* UTILIZANDO O *JAVA GATEWAY***

NATAL / RN
2012

ITAMIR DE MORAIS BARROCA FILHO

**CONFIGURAÇÃO DO ZABBIX PARA O MONITORAMENTO DO SERVIDOR DE
APLICAÇÃO JBOSS UTILIZANDO O JAVA GATEWAY**

Monografia apresentada ao Curso de Especialização em Redes de Computadores, do Centro Universitário do Rio Grande do Norte – UNI-RN, como requisito para a obtenção do título de Especialista em Redes de Computadores.

Orientador: Prof. M.Sc. Aluizio F. Rocha Neto.

NATAL / RN
2012

ITAMIR DE MORAIS BARROCA FILHO

**CONFIGURAÇÃO DO ZABBIX PARA O MONITORAMENTO DO SERVIDOR DE
APLICAÇÃO JBOSS UTILIZANDO O JAVA GATEWAY**

Monografia apresentada ao Curso de Especialização em Redes de Computadores, do Centro Universitário do Rio Grande do Norte – UNI-RN, como requisito para a obtenção do título de Especialista em Redes de Computadores.

Aprovado em: 26/10/2012

BANCA EXAMINADORA

Prof. M.Sc. Aluizio Ferreira da Rocha Neto

Membro examinador

Membro examinador

*“Viva como se fosse morrer amanhã,
aprenda como se fosse viver para
sempre.”*

(Gandhi)

RESUMO

Com o aumento da utilização da internet para disponibilização de serviços por meio de aplicações corporativas *JEE (Java Enterprise Edition)*, a necessidade de se garantir alta disponibilidade das aplicações torna-se um fator crítico e de extrema importância. Uma das maneiras para se garantir alta disponibilidade é o monitoramento efetivo dos servidores de aplicação, como por exemplo, o *JBoss* e o *Tomcat*. Tais servidores de aplicação são responsáveis pela execução desses serviços e o seu monitoramento pode ser realizado por meio de ferramentas *open source*, tais como o *Zabbix*. Sob esta ótica, este trabalho demonstra a configuração dessa ferramenta para o monitoramento e gerenciamento de um servidor de aplicação *JBoss*, utilizando o *Zabbix Java Gateway*, abordando desde a instalação do servidor e agente *Zabbix* à realização do monitoramento.

Palavras-chave: *Zabbix*. *JBoss*. *Java Gateway*. Monitoramento.

ABSTRACT

With the increase in use of the internet as a mean of providing services through JEE (Java Enterprise Edition) corporate applications, the need to ensure high availability for these applications becomes a critical and extremely important factor. One way to ensure high availability is through the effective monitoring of application servers such as the JBoss and the Tomcat. These application servers are responsible for implementing the services and its monitoring can be done through the use of open source tools such as the Zabbix. From this perspective, this work demonstrates the configuration this tool for the effective monitoring and management of a JBoss application server using the Zabbix Java Gateway, and addresses the process since the installation of the server and Zabbix agent to the completion of the process of monitoring.

Keywords: *Zabbix. JBoss. Java Gateway. Monitoring.*

LISTA DE FIGURAS

Figura 1: Computadores com Internet no Brasil.....	8
Figura 2: Portabilidade da máquina virtual.....	10
Figura 3: Estrutura da máquina virtual <i>Java</i>	11
Figura 4: Conteúdo do servidor de aplicação JBoss.....	11
Figura 5: Conteúdo de uma instância do servidor de aplicação <i>JBoss</i>	12
Figura 6: Exemplo de monitoramento de utilização da rede no servidor do <i>Postgres</i>	13
Figura 7: Servidores monitorados pelo <i>Zabbix</i>	14
Figura 8: Descompactação do Zabbix para instalação.....	15
Figura 9: Criação do grupo e dos usuários para iniciação do Zabbix.....	16
Figura 10: Informações sobre armazenamento da base de dados <i>PostgreSQL</i>	16
Figura 11: Instalação do aplicativo <i>PostgreSQL Client</i> para criação da base de dados e esquemas necessários para o <i>Zabbix</i>	17
Figura 12: Criação da base de dados do <i>Zabbix</i>	17
Figura 13: Interface da ferramenta <i>PgAdmin</i>	17
Figura 14: Instalação dos pacotes libcurl e libsnmp.....	19
Figura 15: Comandos configure e make install.....	19
Figura 16: Inicialização do servidor e agente zabbix.....	19
Figura 17: Cópia dos arquivos da <i>frontend web</i> e instalação do <i>PHP</i>	20
Figura 18: Arquivo <i>zabbix.conf.php</i> com informações da conexão com a base de dados <i>zabbix</i>	20
Figura 19: Instalação do agente <i>zabbix</i>	21
Figura 20: Instalação do <i>Zabbix Java Gateway</i>	22
Figura 21: Estrutura criada que contém o <i>Zabbix Java Gateway</i>	22
Figura 22: Configuração do Zabbix Java Gateway no arquivo <i>zabbix_server.conf</i>	23
Figura 23: Configuração do gerenciamento <i>JMX</i> no arquivo <i>run.conf</i> do servidor de aplicação <i>JBoss</i>	23
Figura 24: Comando para verificar se o <i>Zabbix Java Gateway</i> foi inicializado.....	24
Figura 25: Tela do <i>Zabbix</i> para configuração dos <i>hosts</i> monitorados.....	24
Figura 26: Configuração para o monitoramento do servidor zabbix.....	25
Figura 27: Configuração para o monitoramento do servidor <i>JBoss</i>	25

Figura 28: Trecho da criação do <i>host</i> servidor <i>Zabbix</i> destacando a definição do agente <i>Zabbix</i>	26
Figura 29: Trecho da criação de <i>host</i> servidor <i>JBoss</i> destacando a definição do <i>Zabbix Java Gateway</i>	26
Figura 30: <i>Hosts</i> configurados para serem monitorados pelo <i>Zabbix</i>	27
Figura 31: Mapa com os servidores monitorados pelo <i>Zabbix</i>	27
Figura 32: <i>Dashboard</i> do <i>Zabbix</i> contendo informações do status dos servidores monitorados.....	28
Figura 33: Gráfico de <i>garbage collector collections</i> por segundo no servidor <i>Jboss</i>	28
Figura 34: Gráfico de pool de memória <i>perm gen</i> no servidor <i>Jboss</i>	29
Figura 35: Gráfico de class loaders no servidor <i>Jboss</i>	29
Figura 36: Gráfico de Threads utilizadas no servidor <i>Jboss</i>	30

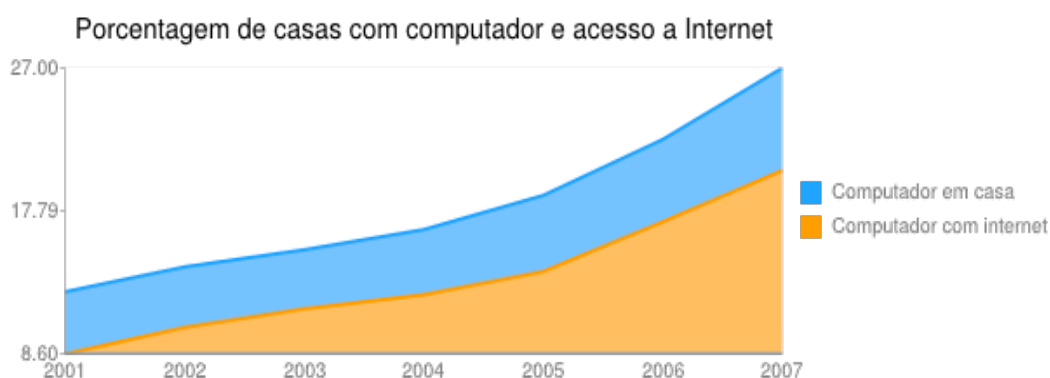
SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVOS	9
2 REFERENCIAL TEÓRICO	10
2.1 JVM (<i>JAVA VIRTUAL MACHINE</i>)	10
2.2 SERVIDOR DE APLICAÇÃO <i>JBoss</i>	11
2.3 SOLUÇÃO DE MONITORAMENTO <i>ZABBIX</i>	12
3 ESTUDO DE CASO – MONITORAMENTO DO <i>JBoss</i> COM O <i>ZABBIX</i>	15
3.1 INSTALAÇÃO DO SERVIDOR E AGENTE <i>ZABBIX</i>	15
3.2 INSTALAÇÃO <i>ZABBIX</i> JAVA GATEWAY	21
3.3 CONFIGURAÇÃO DO AMBIENTE MONITORADO	24
3.4 REALIZAÇÃO DO MONITORAMENTO DO <i>JBoss</i> COM <i>ZABBIX</i>	26
4 CONCLUSÕES	31
REFERÊNCIAS	32

1 INTRODUÇÃO

Com o crescimento da utilização da internet, apresentado na Figura 01, as grandes empresas privadas e governamentais estão cada vez mais disponibilizando, por meio da internet, serviços que até então exigia o deslocamento do usuário para o estabelecimento. Esses serviços, dentre eles portais eletrônicos, aplicações bancárias e comércio eletrônico, são disponibilizados como aplicações em servidores e acessados pelo usuário por meio de navegadores como o *Google Chrome*, *Firefox* e *Internet Explorer*.

Figura 1 - Computadores com Internet no Brasil



Fonte: [3]

Esse novo paradigma de disponibilização de serviços por meio da internet faz-se com que as aplicações sejam criadas com plataformas que possibilitem o desenvolvimento corporativo, como por exemplo, a plataforma *JEE* (*Java Enterprise Edition*). De acordo com [1],

A plataforma *Java, Enterprise Edition (Java EE)* é um padrão industrial para a computação *Java* corporativa. Utiliza o novo e leve *Profile Web Java EE 6* para criar a próxima geração de aplicações *web*, e o poder total da plataforma *Java EE 6* para aplicações corporativas.

Dentre os benefícios do desenvolvimento com a plataforma *Java EE* tem-se maior produtividade com relação à utilização de anotações, mais *POJOs* e menos *XMLs* [1], o que torna interessante o desenvolvimento de aplicações para internet com essa tecnologia. As facilidades de se trabalhar com a plataforma *Java* se tornam um atrativo e disseminam a sua utilização. Em 2007, por exemplo, 46% das empresas brasileiras já a tinham escolhido como plataforma de desenvolvimento [2].

Uma vez desenvolvida uma aplicação com o *Java EE*, é necessário, para que a mesma ganhe vida, sua execução por meio de um servidor de aplicação. Existem diversos tipos de servidores de aplicação como, por exemplo, o *Glassfish*, o *Apache Tomcat* e o *Jboss*, próprios para execução de aplicações corporativas desenvolvidas em *Java*.

Ao publicar uma aplicação corporativa, o monitoramento do servidor de aplicação torna-se uma variável importante para garantir a disponibilidade da mesma. Existem diversas ferramentas *open source* para realizar esse monitoramento de servidores, tais como o *Nagios*, o *Zennos*, o *Cacti* e o *Zabbix*. Uma vantagem da utilização do *Zabbix* para o monitoramento de servidores de aplicação *Java*, em comparação com as demais ferramentas, é o *daemon Zabbix Java Gateway*, introduzido na versão 2.0 [4], que permite, por intermédio de *API JMX (Java Management Extensions)*, o monitoramento de parâmetros do servidor e da máquina virtual *Java*, como por exemplo, *classloaders*, *perm gen space*, *pools* de memória e *threads*.

1.1 Objetivos

Diante da necessidade do monitoramento dos servidores de aplicações corporativas *JEE*, exposta no capítulo 1, como uma forma de garantir disponibilidade desses serviços, este trabalho tem o objetivo de demonstrar as etapas necessárias para configurar o monitoramento de uma aplicação *JEE* disponibilizada através do *JBoss 4.2.2*, por meio do *Zabbix 2.0* e utilizando o *daemon Zabbix Java Gateway*. O mesmo irá abordar desde a configuração do *Zabbix* à configuração do *gateway*. Por fim, serão apresentados dados como: *pool* de memória, *PermGen space*, *Threads*, *Class Loaders* e *Garbage Collection*, coletados por meio dessa ferramenta, servindo, assim, como um guia para profissionais que desejem implantá-la como ferramenta de monitoramento de servidores aplicações *JBoss*.

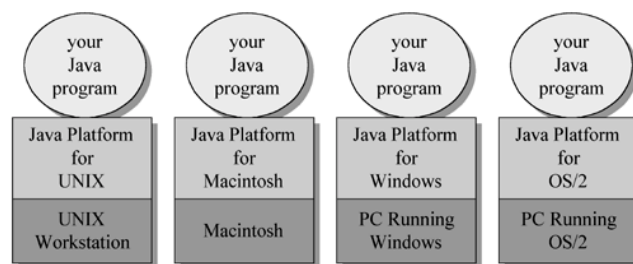
2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados conceitos relacionados à infraestrutura, importantes para o entendimento do processo de implantação do monitoramento. Serão abordados os temas *JVM (Java Virtual Machine)*, servidor de aplicação *JBoss* e a solução de monitoramento *open source Zabbix*.

2.1 JVM (Java Virtual Machine)

A máquina virtual *Java* é uma máquina imaginária que é executada através da emulação em uma máquina real. Essa característica de ser virtual garante um dos principais atrativos da plataforma *Java*, que é a portabilidade, apresentada na Figura 02. Essa característica permite que um mesmo programa escrito em *Java* execute em diferentes plataformas de *hardware* e sistema operacional.

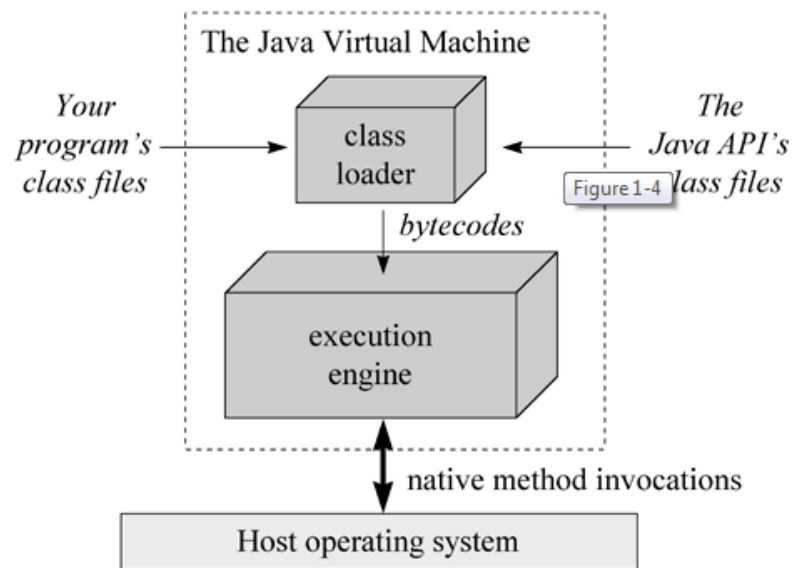
Figura 2 - Portabilidade da máquina virtual.



Fonte: [6]

A mesma é responsável pelo funcionamento do *JBoss* e, em conjunto com ele, garante a execução de aplicações desenvolvidas para plataforma *JEE*, conforme apresentado na Figura 03. Para isso, a execução de uma aplicação *Java* se dá em duas etapas gerais:

1. *Class Loader* e *Bytecode verifier*: onde se dá o carregamento das classes da aplicação e a verificação dos *bytecodes* para que as mesmas sejam executadas;
2. Execução: execução das classes carregadas.

Figura 3: Estrutura da máquina virtual *Java*.

Fonte: [6]

2.2 Servidor de aplicação JBoss

Após entender o funcionamento da máquina virtual *Java* (JVM), torna-se importante o entendimento do servidor de aplicação *JBoss*, cujas aplicações monitoradas estarão em seu *container* de execução. O *JBoss* é um servidor de aplicação de código fonte aberto, o que permite customizações, baseado na plataforma *JEE*. Dessa forma, por ser desenvolvido em *Java*, conforme a característica de portabilidade citada na seção 2.1, o mesmo pode ser instalado em qualquer sistema operacional que suporte a plataforma a JVM (*Java Virtual Machine*).

O conteúdo do *JBoss* é apresentado na Figura 04, composto pelas pastas *bin*, *client*, *docs*, *lib* e *server*. A pasta *bin* contém principalmente os *jars* para iniciar, *run.jar*, e parar, *stop.jar*, as instâncias definidas na pasta *server*.

Figura 4: Conteúdo do servidor de aplicação JBoss.

```
root@tristram:/usr/local/jboss-4.2.2.GA# ls
bin      copyright.txt  jar-versions.xml  lgpl.html  readme.html
client   docs          JBossORG-EULA.txt lib         server
```

Esse servidor de aplicação permite a inicialização de várias instâncias em portas diferentes, que devem estar definidas como subpastas na pasta *server*. Uma instância é definida como uma pasta e contém as subpastas *conf*, *data*, *deploy*, *lib*, *log*, *logs*, *tmp* e *work*, apresentadas na Figura 05.

Figura 5: Conteúdo de uma instância do servidor de aplicação *JBoss*.

```
root@tristram:/usr/local/jboss-4.2.2.GA/server/instancia1# ls
conf  data  deploy  lib  log  logs  tmp  work
```

A pasta *conf* contém os arquivos de configuração da instância como, por exemplo, o *jboss-service.xml*, que define principalmente o caminho dos arquivos das aplicações *web* que serão adicionadas ao *container* de execução do *JBoss*. A pasta *deploy*, por sua vez, possui informações necessárias ao *deploy* da aplicação, como por exemplo, a definição dos *datasources* com informações para conexões da mesma com o banco de dados. As informações de *log*, importantes para auditoria nas aplicações, ficam armazenadas na pasta *log*.

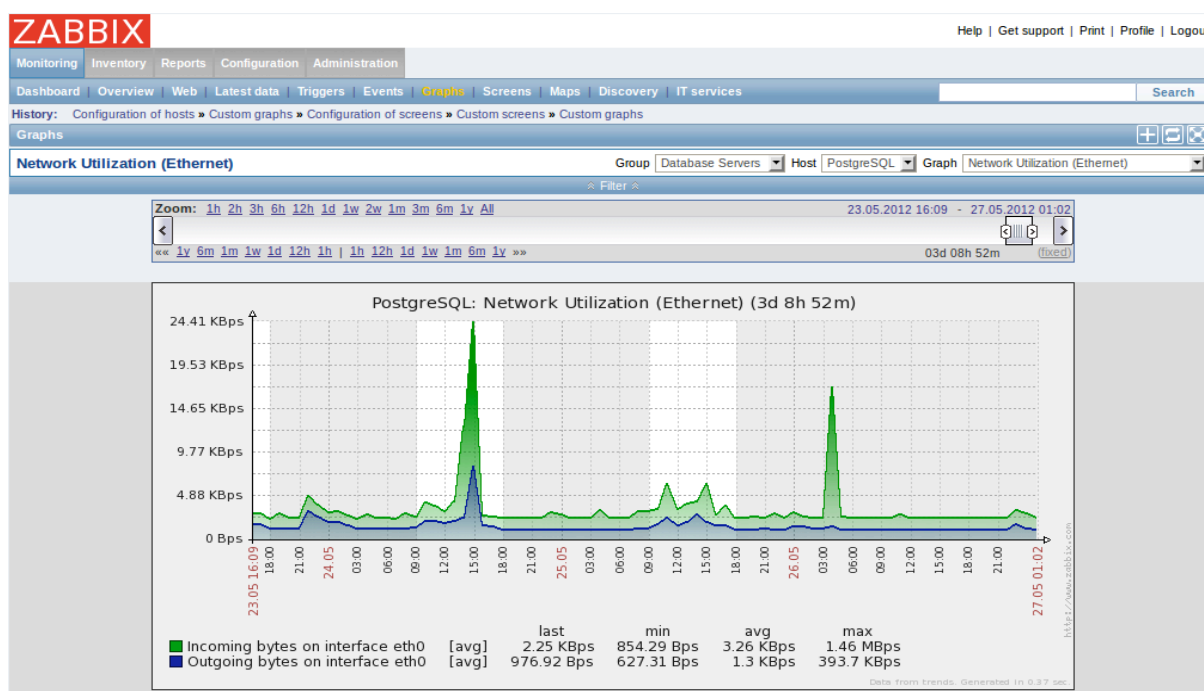
2.3 Solução de monitoramento *Zabbix*

O *Zabbix* é uma ferramenta *open source* para monitoramento de desempenho e disponibilidade. Ele oferece monitoramento avançado, alertas e características de visualização que faltam atualmente em outros sistemas de monitoramento, até em alguns sistemas mais comerciais e considerados os melhores do mercado [7]. Dentre as principais funcionalidades fornecidas por essa ferramenta, estão:

- Descoberta automática de servidores e dispositivos de redes;
- Baixo nível de intrusão dos agentes;
- Monitoramento distribuído centralizado através de uma aplicação *web*;
- *Software* servidor para *Linux*, *Solaris*, *HP-UX*, *AIX*, *FreeBSD*, *OpenBSD* e *OSx*;
- Autenticação e segurança;
- *Interface* de configuração *web*;
- *Logs* para auditoria.

Com o *Zabbix*, é possível realizar o monitoramento de servidores, ativos de rede e aplicações, centralizando na *interface web* estatísticas e dados de desempenho. Informações de CPU, memória, rede e espaço no disco rígido podem ser monitorados com os agentes *Zabbix*, disponíveis para plataformas como *Linux*, *Windows* e *Unix*. Essa ferramenta também suporta o uso de agentes *SNMP*, que utilizam o protocolo *SNMP (Simple Network Management Protocol)* para realização desse monitoramento. Na Figura 06, é apresentado um exemplo de monitoramento da utilização da rede em um servidor com o banco de dados *Postgres* [7].

Figura 6: Exemplo de monitoramento de utilização da rede no servidor do *Postgres*.



Fonte: [7]

Esse monitoramento pode ser realizado utilizando os agentes disponibilizados por essa ferramenta, que são: agente *zabbix*, *SNMP*, *Java Gateway* e *IPMI*, conforme apresentado na Figura 07. O *Zabbix Java Gateway* foi introduzido na versão 2.0 dessa ferramenta, possibilitando o monitoramento de servidores de aplicação como *JBoss* e *Tomcat*.

Figura 7: Servidores monitorados pelo Zabbix.

ZABBIX

Monitoring

Inventory

Reports

Configuration

Administration

Help

Get support

Print

Profile

Logout

Host groups

Templates

Hosts

Maintenance

Web

Actions

Screens

Slide shows

Maps

Discovery

IT services

History:

Configuration of hosts

Configuration of host groups

Configuration of hosts

Configuration of host groups

Configuration of hosts

CONFIGURATION OF HOSTS

Create host

Import

Hosts

Group

Clusters

Displaying 1 to 4 of 4 found

Filter

☐	Name	Applications	Items	Triggers	Graphs	Discovery	Interface	Templates	Status	Availability
☐	Apache Hadoop	Applications (8)	Items (102)	Triggers (5)	Graphs (15)	Discovery (0)	192.168.3.36: 32010	Template Unix (Template CPU, Template Disk, Template File System, Template Memory, Template Network, Template System, Template Zabbix Agent)	Monitored	
☐	Microsoft Cluster Server	Applications (8)	Items (102)	Triggers (5)	Graphs (15)	Discovery (0)	192.168.3.27: 32010	Template Unix (Template CPU, Template Disk, Template File System, Template Memory, Template Network, Template System, Template Zabbix Agent)	Monitored	
☐	Red Hat Cluster	Applications (8)	Items (102)	Triggers (5)	Graphs (14)	Discovery (0)	192.168.3.27: 32010	Template Windows (Template CPU, Template Disk, Template File System, Template Memory, Template Network, Template System, Template Zabbix Agent)	Monitored	
☐	Solaris Cluster	Applications (8)	Items (102)	Triggers (5)	Graphs (15)	Discovery (0)	192.168.3.42: 32010	Template Unix (Template CPU, Template Disk, Template File System, Template Memory, Template Network, Template System, Template Zabbix Agent)	Monitored	

Export selected

Go (0)

Fonte: [7]

3 ESTUDO DE CASO – MONITORAMENTO DO JBOSS COM O ZABBIX

Uma vez apresentadas as tecnologias referentes à infraestrutura no capítulo 2, neste capítulo será abordado o foco deste trabalho, que é o monitoramento do servidor de aplicação *JBoss* com o *Zabbix* por meio do *Zabbix Java Gateway*. Serão detalhadas tanto a instalação do servidor e agente *Zabbix*, o *Zabbix Java Gateway*, quanto a configuração de um ambiente monitorado e, por fim, os dados coletados pela ferramenta.

3.1 Instalação do Servidor e Agente *Zabbix*

Para iniciar o monitoramento, foi necessária a instalação do servidor *Zabbix*, que dependeu de uma base de dados para coleta das informações dos ativos de rede. Para isso, foi utilizado o Sistema de Gerenciamento de Banco de Dados - SGBD *Postgres*.

O servidor foi instalado em uma máquina virtual com *Ubuntu Server*, versão 12.04 com *kernel* 3.2. Para instalação do servidor *Zabbix* versão 2.0, foi necessário o *download* do arquivo *zabbix-2.0.1.tar.gz* em [8]. Caso fosse utilizada a versão 1.8, essa instalação poderia ser feita por meio do comando *apt-get install zabbix-server-pgsql*, porém essa versão não possui a possibilidade de utilização do *zabbix java gateway* necessário para monitoramento do servidor de aplicação *JBoss*. Após a realização do *download* do *Zabbix*, o desempacotamento do arquivo foi realizado por meio do comando apresentado na Figura 08.

Figura 8: Descompactação do *Zabbix* para instalação.

```
root@ubuntu01:/home/itamir# tar -zxvf zabbix-2.0.1.tar.gz
```

Após a descompactação, foi realizada a criação do grupo e dos usuários que seriam responsáveis pela inicialização do servidor e agente *Zabbix*. Essa criação é feita utilizando o comando *groupadd* e *useradd*, que são apresentadas, respectivamente, na Figura 09. A título de exemplificação, foram criados o grupo *zabbix* e os usuários *zabbix* e *zabbixag*. A execução do *Zabbix* como *root*, *bin*, ou qualquer outra conta com direitos especiais é um risco de segurança [9], por isso a necessidade da criação desses usuários.

Figura 9: Criação do grupo e dos usuários para iniciação do Zabbix.

```
root@ubuntu01:/home/itamir# groupadd zabbix
root@ubuntu01:/home/itamir# adduser -m -s zabbix zabbix_
root@ubuntu01:/home/itamir# adduser -m -s zabbix zabbixag _
```

Uma vez criados os usuários, tornou-se necessária a criação da base de dados utilizada pelo *Zabbix* para armazenar informações de configurações dos servidores monitorados, usuários cadastrados, entre outras. Cabe frisar que, como base de dados para o *Zabbix*, podem ser utilizados os sistemas de gerenciamento de bases de dados – SGBDs *PostgreSQL*, *MySQL*, *IBM DB2*, *Oracle* e *SQLite*. O SGBD selecionado para ser usado com o *Zabbix* nesse estudo de caso foi o *PostgreSQL*.

Cumprе frisar que o *PostgreSQL* é um poderoso sistema de base de dados relacional que tem mais de quinze anos de desenvolvimento ativo e possui uma arquitetura que ganhou uma forte reputação com relação a confiabilidade, integridade e correteude [10]. Informações sobre capacidade do *PostgreSQL* são apresentados na Figura 10.

Figura 10: Informações sobre armazenamento da base de dados *PostgreSQL*.

Limit	Value
Maximum Database Size	Unlimited
Maximum Table Size	32 TB
Maximum Row Size	1,6 TB
Maximum Field Size	1 GB
Maximum Rows per Table	Unlimited
Maximum Columns per Table	250 - 1600 depending on column types
Maximum Indexes per Table	Unlimited

Fonte: [10]

Não serão apresentadas informações sobre a instalação do SGBD. Considerando a base de dados já instalada, foi utilizado o comando *psql* para a criação da base de dados e dos esquemas que foram utilizados pela aplicação. Caso o comando *psql* não estivesse instalado, a instalação poderia ser realizada por meio do comando apresentado na Figura 11, a seguir.

Outra maneira para criação da base de dados é conectando-se ao SGBD utilizando o aplicativo *PgAdmin*, uma ferramenta *open source* voltada para a administração do SGBD *PostgreSQL*. A interface do *PgAdmin* é apresentada na Figura 13. Os comandos para criação da base de dados, dos esquemas, das imagens e dados, por sua vez, são apresentados na Figura 12. Convém ressaltar que os scripts *SQL* dos esquemas, imagens e dados estão disponíveis na pasta *database*, no pacote descompactado do *Zabbix*.

Figura 11: Instalação do aplicativo *PostgreSQL Client* para criação da base de dados e esquemas necessários para o *Zabbix*.

```
root@ubuntu01:/home/itamir# apt-get install postgresql-client-common_
```

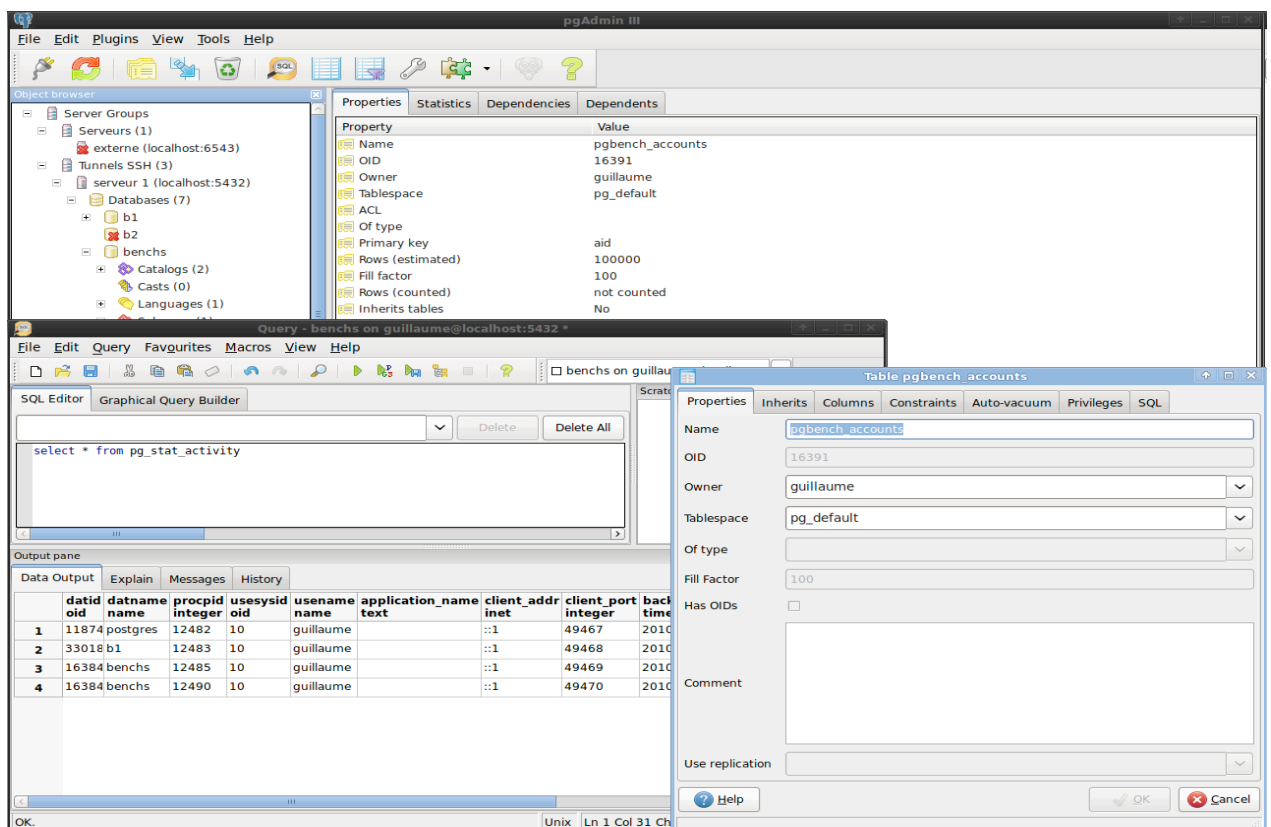
Figura 12: Criação da base de dados do *Zabbix*.

```
root@ubuntu01:/home/itamir/zabbix-2.0.1/database/postgresql# psql -U postgres za
bbix < schema.sql_
```

```
root@ubuntu01:/home/itamir/zabbix-2.0.1/database/postgresql# psql -U postgres za
bbix < data.sql_
```

```
root@ubuntu01:/home/itamir/zabbix-2.0.1/database/postgresql# psql -U postgres za
bbix < images.sql_
```

Figura 13: Interface da ferramenta *PgAdmin*.



Fonte: [11]

Após a criação das tabelas, foi realizada a alteração das permissões dos esquemas e tabelas para o usuário *zabbix* do *PostgreSQL*, utilizando os seguintes comandos:

1. `SELECT 'ALTER TABLE ' || n.nspname || '.' || c.relname || ' OWNER TO zabbix;' FROM pg_catalog.pg_class c LEFT JOIN pg_catalog.pg_namespace n ON n.oid = c.relnamespace WHERE c.relkind IN ('r','S') AND n.nspname NOT IN ('pg_catalog', 'information_schema', 'pg_toast') ORDER BY c.relname;`
2. `SELECT 'ALTER SCHEMA ' || schema_name || ' OWNER TO zabbix;' FROM information_schema.schemata WHERE schema_name NOT LIKE 'pg_%' AND schema_name NOT IN ('information_schema');`

Com a escolha do SGBD e a criação da base dados, passou-se então à instalação do servidor *Zabbix*. Para a realização do *configure*, foi necessária a instalação dos pacotes *libcurl4-openssl-dev* e *libsnmp-dev*, conforme exibido na Figura 14. Em seguida, foi executado o comando *configure* com os seguintes parâmetros:

1. *enable-server*: habilitando a instalação do servidor *zabbix*;
2. *enable-agent*: habilitando o agente *zabbix*;
3. *with-postgresql*: define a utilização do SGBD *PostgreSQL*;
4. *enable-ipv6*: habilita a possibilidade de utilização do IPV6;
5. *with-net-snmp* e *with-libcurl*: especificando a utilização do SNMP e com a biblioteca *libcurl* instalada anteriormente.

Com a execução do comando *configure* realizada com sucesso, passou-se ao comando *make install*, utilizado para efetivar a instalação. Os comandos *configure*, já com os parâmetros utilizados, e *make install* são exibidos na Figura 15.

Figura 14: Instalação dos pacotes libcurl e libsnmp.

```
root@ubuntu01:/home/itamir# apt-get install libcurl4-openssl-dev libsnmp-dev_
```

Figura 15: Comandos configure e make install.

```
root@ubuntu01:/home/itamir/zabbix-2.0.1# ./configure --enable-server --enable-agent --with-postgresql=/usr/local/postgresql-9.0.4/bin/pg_config --enable-ipv6 --with-net-snmp --with-libcurl_
```

```
root@ubuntu01:/home/itamir/zabbix-2.0.1# make install_
```

Com a finalização da instalação, foi necessário, para iniciar o servidor *zabbix*, apenas realizar o *login* com o usuário *zabbix* criado anteriormente e executar o comando *zabbix_server*. Similarmente, para iniciar o agente *zabbix*, foi necessário apenas realizar o *login* com o usuário *zabbixag* e executar o comando *zabbix_agentd*. A execução desses comandos é apresentada na Figura 16.

Figura 16: Inicialização do servidor e agente zabbix.

```
root@ubuntu01:/home/itamir# su zabbix
zabbix@ubuntu01:/home/itamir$ zabbix_server_

root@ubuntu01:/home/itamir# su zabbixag
zabbixag@ubuntu01:/home/itamir$ zabbix_agentd_
```

É importante salientar que existem dois arquivos que definem configurações para o servidor e agente *zabbix*. São eles: *zabbix_server.conf* e *zabbix_agentd.conf*, ambos localizados na pasta */usr/local/etc/*. O arquivo de configurações do servidor *zabbix*, *zabbix_server.conf*, será utilizado para configurar o *zabbix java gateway*, configuração esta que será detalhada na seção 3.3 deste trabalho.

Com o servidor *zabbix* devidamente configurado, foi criada uma pasta denominada “*zabbix*”, dentro de */var/www/* e foi copiado para ela o conteúdo da pasta *frontends/php/*, presente no pacote descompactado. O objetivo do processo é criar o *frontend web* para configuração e monitoramento dos *hosts* por meio do servidor *zabbix*. Em seguida, foi necessária a instalação do *apache* e do *PHP*. Esses comandos são apresentados na Figura 17.

Figura 17: Cópia dos arquivos da *frontend web* e instalação do *PHP*.

```
root@ubuntu01:/home/itamir# mkdir /var/www/zabbix_
root@ubuntu01:/home/itamir# cp -a /home/itamir/zabbix-2.0.1/frontends/php/* /var/
/var/www/zabbix/_
root@ubuntu01:/home/itamir# apt-get install apache2 php5 php5-pgsql php5-gd php5
-curl_
```

Para conclusão da instalação, já integrada com a *interface web*, foi necessário copiar o arquivo *zabbix.conf.php* para o diretório */var/www/zabbix/conf/*. Esse arquivo continha informações da conexão com o servidor que possuía a base de dados *zabbix* criada anteriormente, tais como: tipo do SGBD, endereço do servidor, porta de conexão, nome da base dados, usuário e senha. Um trecho desse arquivo é apresentado na Figura 18, a seguir.

Figura 18: Arquivo *zabbix.conf.php* com informações da conexão com a base de dados *zabbix*.

```
<?php
// Zabbix GUI configuration file
global $DB;

$DB['TYPE']           = 'POSTGRESQL';
$DB['SERVER']         = 'localhost';
$DB['PORT']           = '5432';
$DB['DATABASE']       = 'zabbix';
$DB['USER']            = 'zabbix';
$DB['PASSWORD']       = 'zabbix';

// SCHEMA is relevant only for IBM_DB2 database
$DB['SCHEMA']         = '';

$ZBX_SERVER           = 'localhost';
$ZBX_SERVER_PORT      = '10051';
$ZBX_SERVER_NAME      = '';

$IMAGE_FORMAT_DEFAULT = IMAGE_FORMAT_PNG;
?>
```

Com a realização deste processo, foi finalizada a instalação do servidor e agente *zabbix*. Para realizar o *login* na aplicação *web*, foi necessário apenas acessar o link *http://<ip do servidor zabbix>/zabbix* utilizando o *login* “Admin” e a senha: “zabbix”. Esse *login* e senha são padrões e podem ser alterados pelo usuário na aplicação *web*. Caso fosse necessária a instalação do agente em outra máquina, bastaria realizar o mesmo procedimento de criação do usuário *zabbixag* e a execução comando *configure* com os parâmetros alterados:

1. *--enable-agent*: habilita o agente *zabbix*;
2. *--enable-static*: habilita a utilização IPs estáticos;
3. *--enable-ipv6*: habilita a possibilidade de utilização do IPV6.

Dessa forma, seria habilitado apenas o agente e, em seguida, realizado o *make*. É importante lembrar que é necessária a instalação das bibliotecas *libcurl4-openssl-dev* e *libsnmp-dev*. A execução dos comandos dessa instalação foi apresentada na Figura 19.

Figura 19: Instalação do agente *zabbix*.

```
root@ubuntu01:/home/itamir# apt-get install libcurl4-openssl-dev libsnmp-dev_
root@ubuntu01:/home/itamir/zabbix-2.0.1# ./configure --enable-agent --enable-sta
tic --enable-ipv6_
root@ubuntu01:/home/itamir/zabbix-2.0.1# make install_
```

3.2 Instalação Zabbix Java Gateway

Nesta seção, serão apresentados os passos necessários para a instalação do *Zabbix Java Gateway*, realizados após a instalação do servidor e agente *Zabbix*. Esse *daemon* foi introduzido na versão 2.0 do *Zabbix* e permitiu que as variáveis *JMX* de uma aplicação *Java*, sendo executadas em um servidor de aplicação como, por exemplo, o *JBoss*, sejam monitoradas.

No *Zabbix* 1.8, caso o usuário deseje o monitoramento dessas variáveis, a melhor escolha seria o *Zapcat JMX Zabbix Bridge* [12]. O monitoramento poderia ser realizado em variáveis como: *pool* de memória, *PermGen space*, *Threads*, *Class Loaders* e *Garbage Collection*, cujo monitoramento efetivo facilitaria a garantia de disponibilidade da aplicação *Java web*.

Cabe frisar que as aplicações *Java*, além de utilizarem recursos do sistema operacional, também consomem recursos da *JVM* (*Java Virtual Machine*) para disponibilizar acesso aos seus aplicativos, fazer conexão com o banco de dados, etc [13]. Por isso, é altamente recomendado que se monitore os serviços e aplicativos *Java*. Por exemplo, uma aplicação pode travar ou mesmo deixar de iniciar por falta de memória na *JVM*, apesar de ter memória livre no sistema operacional [13].

Dado um exemplo da importância do monitoramento de uma aplicação *web Java*, é preciso acentuar que, para a instalação do *Zabbix Java Gateway*, não é

necessário nenhum *software* adicional, pois os *softwares* necessários encontram-se disponíveis dentro do pacote do *Zabbix*. No entanto, para que ele seja instalado, é necessário que o comando *configure* seja executado considerando os seguintes parâmetros:

1. *--enable-java*: habilitando a instalação do *Zabbix Java Gateway*;
2. *--prefix=/opt/zabbix*: local onde o *Zabbix Java Gateway* será instalado. No caso, foi escolhido o local */opt/zabbix*.

No processo de instalação, após executado o comando *configure* seguido do comando *make install*, conforme apresentado na Figura 20, foi realizada a instalação desse *gateway* e criada a estrutura apresentada na Figura 21. Porém, para que o *gateway* fosse executado e se tornasse ativo, foi necessário realizar, ainda, algumas configurações.

Figura 20: Instalação do *Zabbix Java Gateway*.

```
root@tristram:/home/itamir/zabbix-2.0.1# ./configure --enable-java --prefix=/opt/zabbix_
root@tristram:/home/itamir/zabbix-2.0.1# make install_
```

Figura 21: Estrutura criada que contém o *Zabbix Java Gateway*.

```
root@tristram:/opt/zabbix/sbin/zabbix_java# ls -la
total 28
drwxr-xr-x 4 root root 4096 Jul 26 11:28 .
drwxr-xr-x 3 root root 4096 Jul 26 11:28 ..
drwxr-xr-x 2 root root 4096 Jul 26 11:28 bin
drwxr-xr-x 2 root root 4096 Jul 26 11:28 lib
-rw-r--r-- 1 root root 665 Jul 26 11:28 settings.sh
-rwxr-xr-x 1 root root 299 Jul 26 11:28 shutdown.sh
-rwxr-xr-x 1 root root 1191 Jul 26 11:28 startup.sh
```

Após a finalização da instalação do *gateway*, duas configurações foram necessárias para inicializá-lo. Primeiramente, foi preciso definir o *Java Gateway* no arquivo de configuração *zabbix_server.conf*, presente nas configurações do servidor *zabbix* (*/usr/local/etc/zabbix_server.conf*), especificando os seguintes parâmetros:

1. *JavaGateway*: IP do servidor que contém o *Zabbix Java Gateway*;

2. *JavaGatewayPort*: Porta utilizada para comunicação do *gateway* com servidor *zabbix*, onde por padrão é utilizada a porta 10052;
3. *StartJavaPollers*: Número de processos relacionados ao monitoramento das variáveis *JMX*, onde, por padrão, é colocado o valor cinco.

Configurado os parâmetros do *Zabbix Java Gateway*, conforme apresentado na Figura 22, foi necessário reiniciar o servidor *zabbix*. Prosseguindo com a configuração para inicialização do *gateway*, foi necessário definir os parâmetros de gerenciamento *JMX* no arquivo *run.conf* presente na pasta *bin* do servidor de aplicação *JBoss*. Os parâmetros configurados, apresentados na Figura 23, foram:

1. *jmxremote.port*: define a porta do serviço *JMX* do servidor de aplicação *JBoss*;
2. *jmxremote.ssl*: define se o serviço *JMX* utiliza SSL;
3. *jmxremote.authenticate*: define se será necessário autenticação para utilização do serviço *JMX* do *JBoss*.

Figura 22: Configuração do Zabbix Java Gateway no arquivo *zabbix_server.conf*.

```
JavaGateway=172.16.66.129
JavaGatewayPort=10052
StartJavaPollers=5
```

Figura 23: Configuração do gerenciamento *JMX* no arquivo *run.conf* do servidor de aplicação *JBoss*.

```
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=9999"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
```

Realizadas as configurações, o *Zabbix Java Gateway* mostrou-se pronto para ser inicializado. Para isso, foi necessário executar o *script startup.sh*, presente na pasta */opt/zabbix/sbin/zabbix_java/*, que contém esse *gateway*. Para parar o *gateway*, foi necessário apenas executar o *script shutdown.sh*. Para verificar se o *gateway* havia sido inicializado corretamente, foi executado o comando *ps -ef | grep logback*, conforme apresentado na Figura 24.

Figura 24: Comando para verificar se o *Zabbix Java Gateway* foi inicializado.

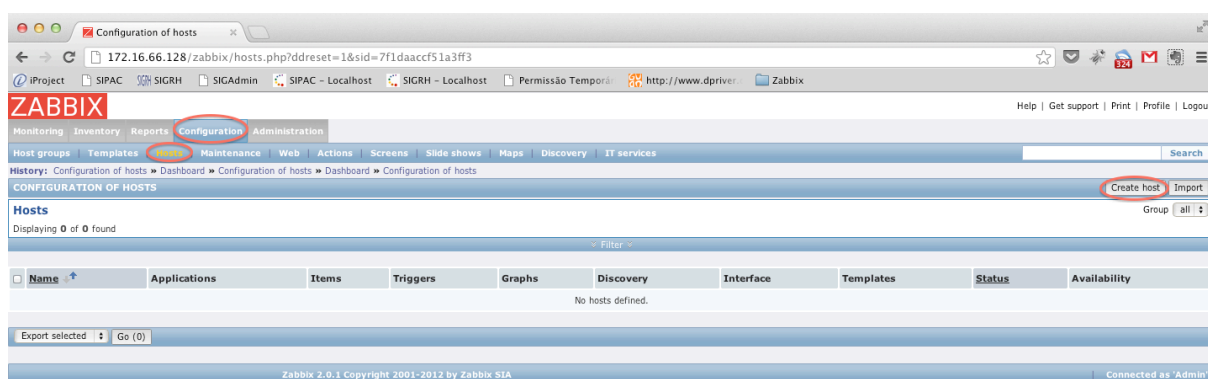
```
root@tristram:/opt/zabbix/sbin/zabbix_java# ps -ef | grep logback
root      1979      1  0 00:41 tty1      00:00:12 java -server -classpath lib:libl
ogback-classic-0.9.27.jar:lib/logback-core-0.9.27.jar:lib/org-json-2010-12-28.jar:lib/slf4j-api-1.6.1.jar:bin/zabbix-java-gateway-2.0.1.jar -Dzabbix.pidFile=/tm
p/zabbix_java.pid com.zabbix.gateway.JavaGateway
root      2674    1395  0 02:12 tty1      00:00:00 grep --color=auto logback
```

3.3 Configuração do Ambiente Monitorado

Com a instalação do servidor e do *Java Gateway* do *Zabbix*, a infraestrutura para configuração dos *hosts* a serem monitorados estava pronta. Então, o próximo passo necessário para iniciar o monitoramento foi a configuração do ambiente monitorado, realizada por meio do *login* da aplicação *Zabbix* acessando o endereço *http://<ip do servidor zabbix>/zabbix* e utilizando o usuário “Admin” e a senha “zabbix”. Em seguida, o caminho *configuration -> Hosts* foi acessado.

Ao clicar nesse caminho, a tela apresentada na Figura 25 foi exibida. Nessa tela clicou-se no botão “*Create Host*”, destacado nessa figura, para que fosse incluídos os *hosts* que deveriam ser monitorados. A título de exemplificação do monitoramento, dois *hosts* foram definidos: o servidor *JBoss* e o servidor *Zabbix*. O primeiro foi monitorado utilizando o *Zabbix Java Gateway* e o segundo, monitorado utilizando o *agente Zabbix*.

Figura 25: Tela do *Zabbix* para configuração dos *hosts* monitorados.



Ao acessar a funcionalidade para a criação de um *host* utilizando o botão “*Create host*”, uma tela para configuração de um novo *host* a ser monitorado foi exibida. Nessa tela, configurações como nome e agentes são definidas. Primeiramente, foi definida a configuração para monitoramento do servidor *Zabbix*, o

qual é apresentado na Figura 26. Em seguida, foi definida a configuração para o servidor *JBoss*, cuja configuração é apresentada na Figura 27.

Figura 26: Configuração para o monitoramento do servidor zabbix.

The screenshot shows the Zabbix configuration interface for a host named 'Zabbix server'. The interface includes a top navigation bar with tabs for Host groups, Templates, Hosts, Maintenance, Web, Actions, Screens, Slide shows, Maps, Discovery, and IT services. Below the navigation bar, there is a breadcrumb trail: History: Configuration of hosts » Dashboard » Configuration of hosts » Configuration import » Configuration of hosts. The main section is titled 'CONFIGURATION OF HOSTS' and shows the host 'Zabbix server' with a status of 'Monitored'. The host is associated with the 'Zabbix servers' group. The configuration includes fields for Host name, Visible name, and Groups. The 'Agent interfaces' section shows a single interface with IP address '127.0.0.1', DNS name, and port '10050'. The 'SNMP interfaces', 'JMX interfaces', and 'IPMI interfaces' sections are empty. The 'Monitored by proxy' field is set to '(no proxy)' and the 'Status' is 'Monitored'. At the bottom, there are buttons for 'Save', 'Clone', 'Full clone', 'Delete', and 'Cancel'.

Figura 27: Configuração para o monitoramento do servidor *JBoss*.

The screenshot shows the Zabbix configuration interface for a host named 'Servidor JBoss'. The interface includes a top navigation bar with tabs for Host groups, Templates, Hosts, Maintenance, Web, Actions, Screens, Slide shows, Maps, Discovery, and IT services. Below the navigation bar, there is a breadcrumb trail: History: Custom screens » Configuration of screens » Dashboard » Configuration of host groups » Configuration of hosts. The main section is titled 'CONFIGURATION OF HOSTS' and shows the host 'Servidor JBoss' with a status of 'Monitored'. The host is associated with the 'Linux servers' group. The configuration includes fields for Host name, Visible name, and Groups. The 'Agent interfaces' section shows a single interface with IP address '172.16.66.129', DNS name, and port '10052'. The 'SNMP interfaces', 'JMX interfaces', and 'IPMI interfaces' sections are empty. The 'Monitored by proxy' field is set to '(no proxy)' and the 'Status' is 'Monitored'. At the bottom, there are buttons for 'Save', 'Clone', 'Full clone', 'Delete', and 'Cancel'.

É importante frisar a diferença de configurações com relação a ambos os *hosts*. Nota-se que no *host* do servidor *zabbix*, um agente na porta 10050 foi definido com endereço *localhost*. Isso só foi possível devido à inicialização do agente *zabbix* no *host* que contém o servidor *zabbix*.

Já no *host* que contém o servidor *JBoss*, uma interface *JMX* e o *Zabbix Java Gateway* foram definidos para o endereço desse servidor respectivamente nas portas 9999 e 10052. A porta 10052 foi definida na configuração do *zabbix_server.conf*, e a porta 9999 foi definida no *run.conf* do *JBoss*, conforme apresentado na seção 3.2. Ambas as definições do agente *Zabbix* e do *Zabbix Java Gateway* são destacadas respectivamente nas Figuras 28 e 29.

Figura 28: Trecho da criação do *host* servidor *Zabbix* destacando a definição do agente *Zabbix*.

Agent interfaces	IP address	DNS name	Connect to	Port	Default
↓ 127.0.0.1 ↑ Add			IP DNS	10050	• Remove

Figura 29: Trecho da criação de *host* servidor *JBoss* destacando a definição do *Zabbix Java Gateway*.

Agent interfaces	IP address	DNS name	Connect to	Port	Default
↓ 172.16.66.129 ↑ Add			IP DNS	10052	• Remove
SNMP interfaces					
Add					
JMX interfaces	↓ 172.16.66.129 ↑ Add		IP DNS	9999	• Remove

3.4 Realização do monitoramento do *JBoss* com *Zabbix*

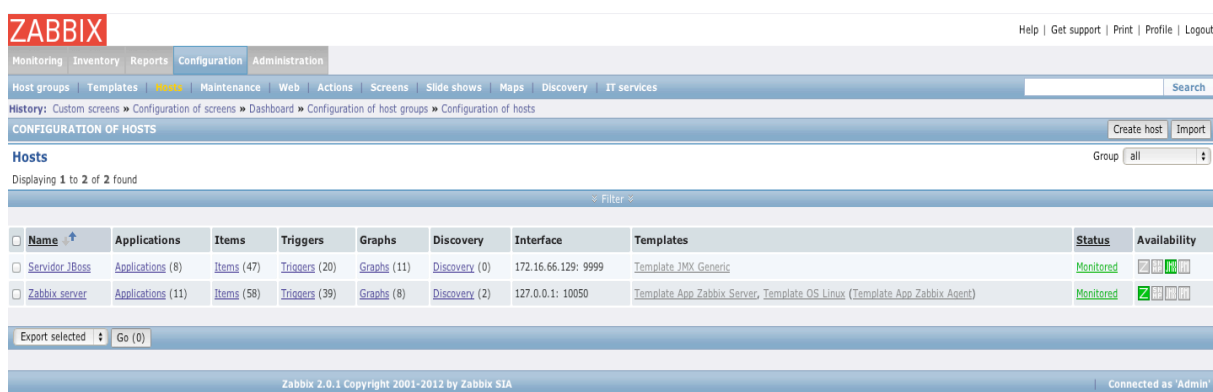
Com a configuração do servidor, agente e *java gateway* do *zabbix* e, posteriormente, com a definição dos *hosts* monitorados, o servidor *Jboss* e servidor *Zabbix*, cujo resultado é apresentado na Figura 30, iniciou-se o monitoramento da aplicação. Para realizar o monitoramento do *Jboss*, os seguintes itens foram criados no *Zabbix*:

- Mapa do ambiente monitorado, apresentado na Figura 31, definido a partir do caminho *Configuration -> Maps*, que contém a representação gráfica dos servidores *Jboss* e *zabbix*. Esse mapa foi denominado de ambiente monitorado;
- *Dashboard*, apresentado na Figura 32, onde foram adicionados os gráficos para monitoramento de *garbage collector*, *pool de perm gen*, *class loaders* e

threads utilizadas no *Jboss* para a aplicação monitorada. Esses gráficos são apresentados respectivamente nas Figuras 33, 34, 35 e 36.

O acompanhamento efetivo do *dashboard*, que centraliza informações importantes da situação dos servidores monitorados, e dos gráficos com os dados do monitoramento, facilita a resposta com relação a possíveis instabilidades nos mesmos. Consequentemente, esse acompanhando influencia na garantia de disponibilidade das aplicações executadas nos servidores de aplicações dos *Jboss* monitorados.

Figura 30: *Hosts* configurados para serem monitorados pelo *Zabbix*.



Name	Applications	Items	Triggers	Graphs	Discovery	Interface	Templates	Status	Availability
Servidor JBoss	Applications (8)	Items (47)	Triggers (20)	Graphs (11)	Discovery (0)	172.16.66.129: 9999	Template JMX Generic	Monitored	
Zabbix server	Applications (11)	Items (58)	Triggers (39)	Graphs (8)	Discovery (2)	127.0.0.1: 10050	Template App Zabbix Server, Template OS Linux (Template App Zabbix Agent)	Monitored	

Figura 31: Mapa com os servidores monitorados pelo *Zabbix*.

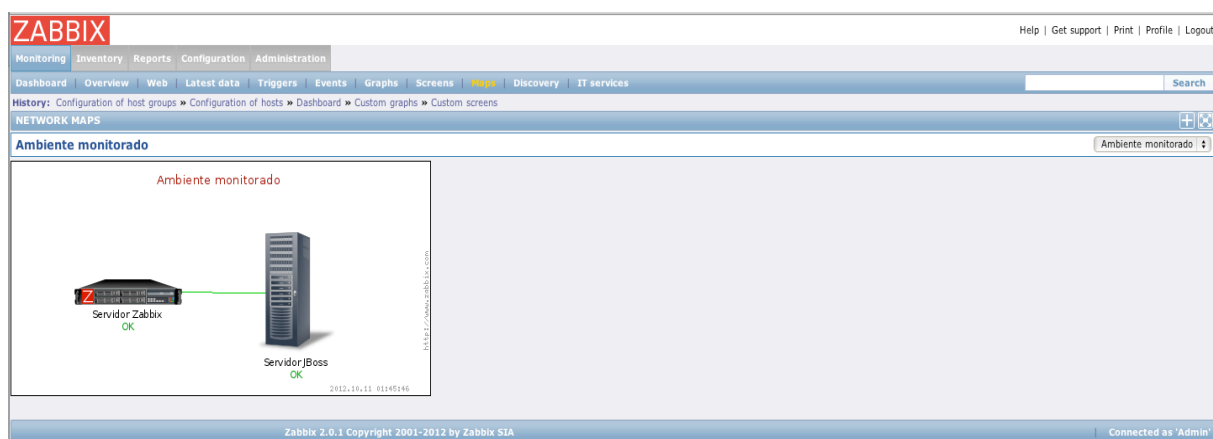


Figura 32: *Dashboard* do Zabbix contendo informações do status dos servidores monitorados.

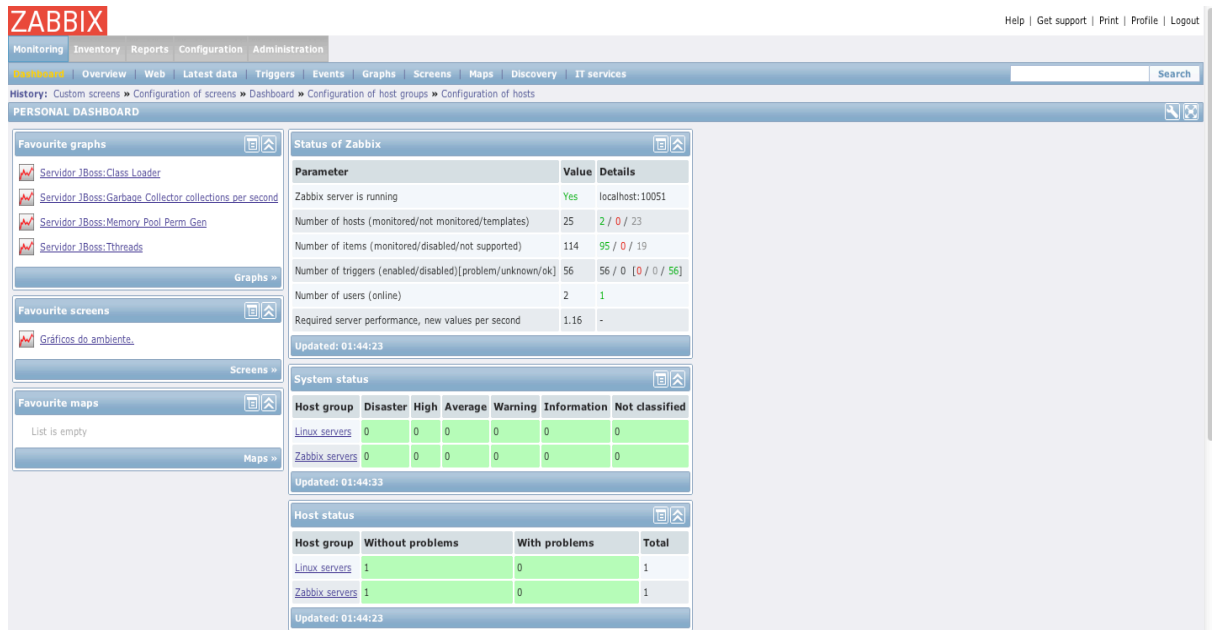


Figura 33: Gráfico de *garbage collector collections* por segundo no servidor Jboss.

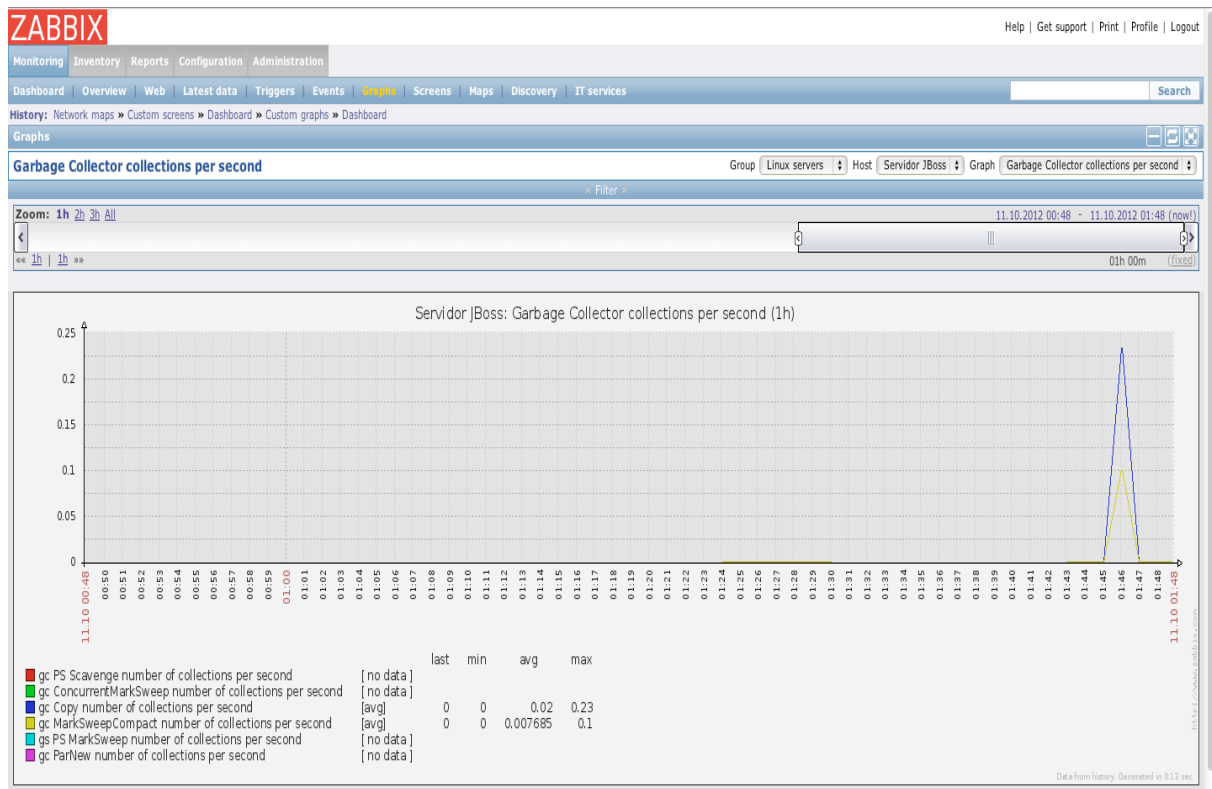


Figura 34: Gráfico de pool de memória *perm gen* no servidor Jboss.

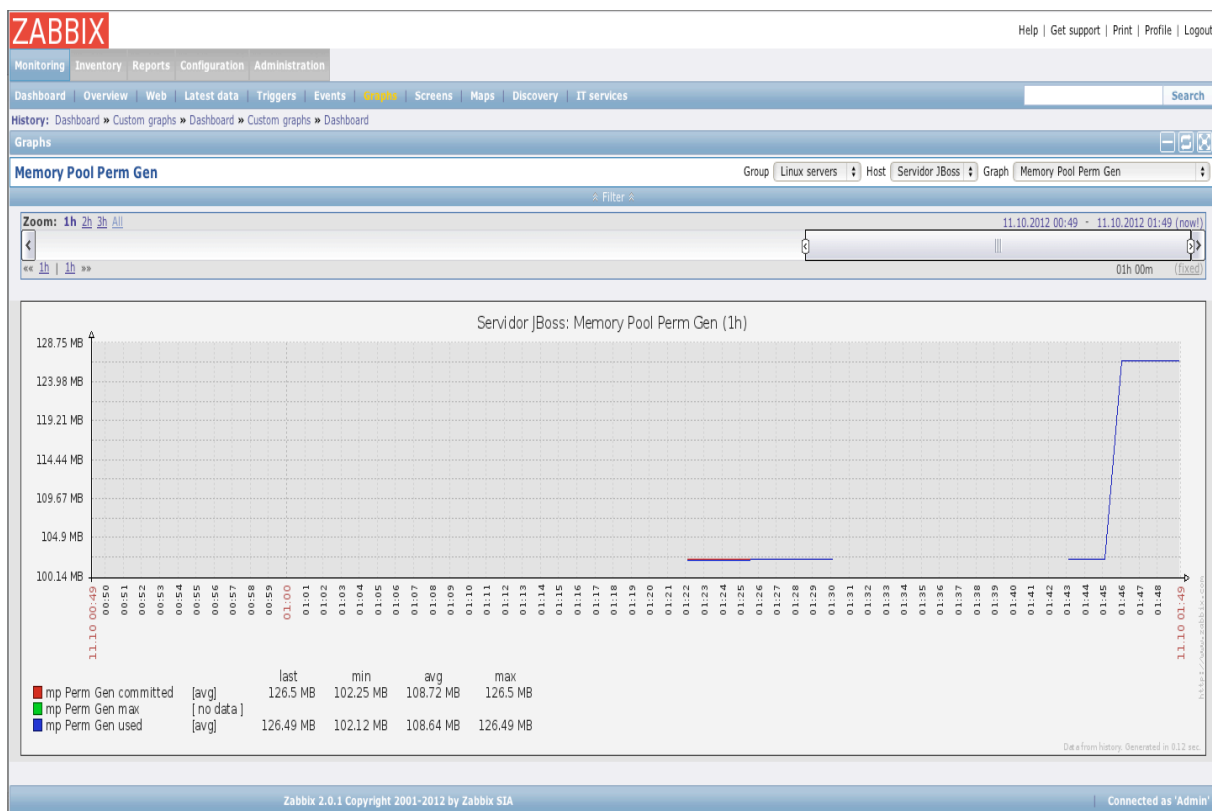


Figura 35: Gráfico de class loaders no servidor Jboss.

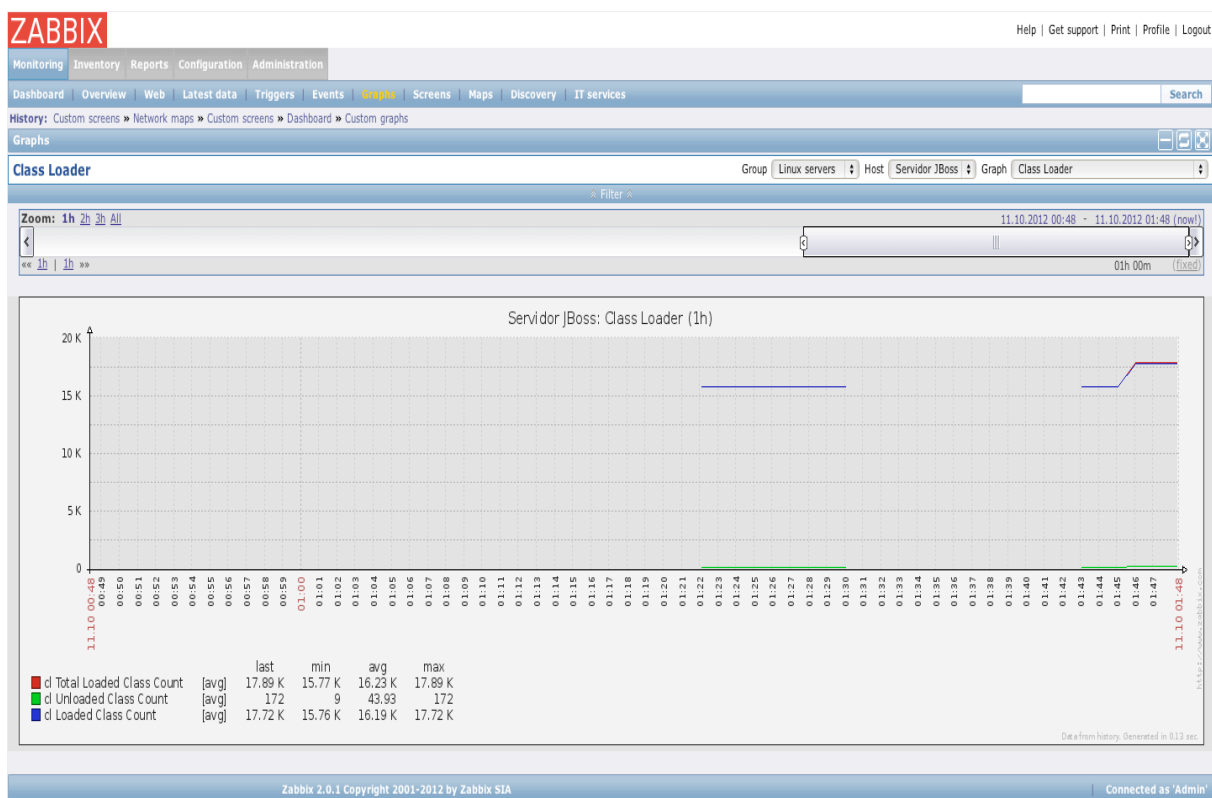
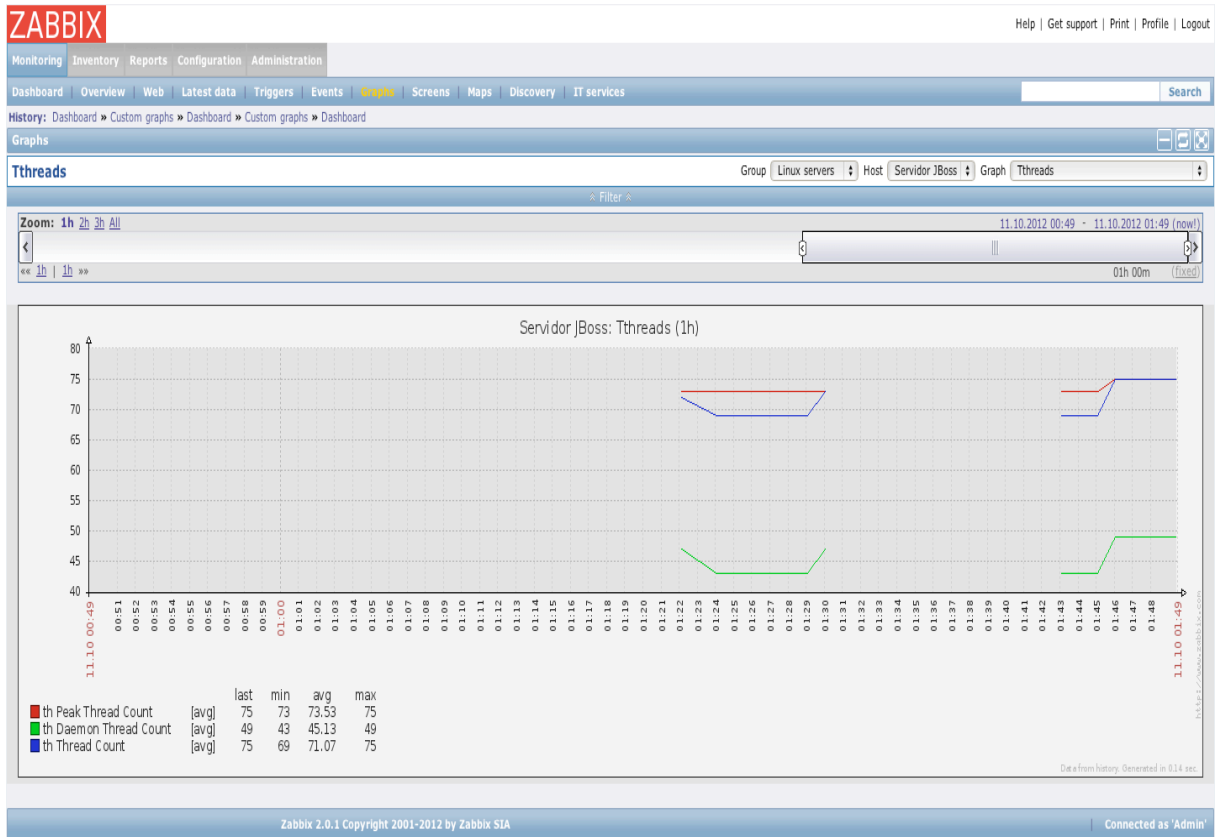


Figura 36: Gráfico de Threads utilizadas no servidor Jboss.

4 CONCLUSÕES

Ao término deste trabalho, o objetivo de demonstrar e relatar os passos necessários à configuração do *zabbix* para monitoramento do *JBoss* por meio do *zabbix java gateway* e formalizá-los em relatório foi atingido. Observou-se a eficiência desse *daemon* no monitoramento do servidor de aplicação *JBoss*, principalmente nos gráficos obtidos apresentados na seção 3.4, onde, por exemplo, ao iniciar a execução desse servidor, o gráfico de *threads* exibia um aumento significativo no contador das mesmas.

Com relação à configuração, notou-se que, seguidos os passos demonstrados nesse trabalho, não existe muita dificuldade em tornar o monitoramento possível. O *zabbix java gateway* apresentou-se estável durante os testes realizados com o monitoramento do *JBoss*.

Dentre as dificuldades obtidas para a realização do trabalho, pode ser citada, principalmente, a falta de documentos, artigos e tutoriais que demonstrem a realização do monitoramento com o *java gateway*, com exceção da documentação própria do *zabbix*. Possivelmente, isso se deve ao fato desse *gateway* ter sido adicionado ao *zabbix* recentemente, na versão 2.0, disponibilizada em 01 de junho de 2012, conforme [14].

Porém, apesar dessas dificuldades, foi possível concluir o trabalho com apoio da documentação oficial do *zabbix*, fóruns da internet, além dos vários conhecimentos adquiridos durante o Curso de Especialização em Redes de Computadores do Centro Universitário do Rio Grande do Norte – UNI-RN.

REFERÊNCIAS

- [1] JAVA EE ORACLE. Disponível em: <<http://www.oracle.com/technetwork/java/javasee/overview/index.html>>. Acesso em: 10 jul. 2012.
- [2] SANCHES, Ligia. Disponível em: <<http://itweb.com.br/29076/46-das-empresas-brasileiras-desenvolvem-em-java/>>. Acesso em: 12 jul.2012.
- [3] ANTONIOLI, Leonardo. Disponível em: <http://tobeguarany.com/internet_no_brasil.php>. Acesso em: 12 jul. 2012.
- [4] DOCUMENTAÇÃO Zabbix Java Gateway. Disponível em <http://www.zabbix.com/documentation/2.0/manual/concepts/java>>. Acesso em: 20 jul. 2012.
- [5] JAVA JMX. Disponível em: <<http://www.oracle.com/technetwork/java/javase/tech/javamanagement-140525.html>>. Acesso em: 20 jul. 2012.
- [6] VENNERS, Bill. **Inside The Java Virtual Machine**. EUA: Mc Graw Hill, 2000.
- [7] ZABBIX. Disponível em: <<http://www.zabbix.com/product.php>>. Acesso em: 12 set. 2012.
- [9] INSTALAÇÃO do Zabbix. Disponível em: <http://www.zabbix.com/documentation/2.0/manual/installation/install#from_the_sources>. Acesso em: 14 set. 2012.
- [8] DOWNLOAD do Zabbix. Disponível em: <<http://www.zabbix.com/download.php>>. Acesso em: 14 set. 2012.
- [10] POSTGRESQL. Disponível em: <<http://www.postgresql.org/about/>>. Acesso em: 15 set. 2012.
- [11] PGADMIN. Disponível em: <<http://www.pgadmin.org/screenshots/>>. Acesso em: 20 set. 2012.
- [12] MONITORAMENTO JMX - Disponível em http://www.zabbix.com/documentation/2.0/manual/config/items/itemtypes/jmx_monitoring. Acesso em 30 de setembro de 2012.
- [13] LIMA, Janssen. Disponível em: <<http://janssenlima.blogspot.com/2012/07/monitorando-servidor-jboss-no-zabbix.html>>. Acesso em: 30 set. 2012.

[14] RELEASES do Zabbix. Disponível em: <<http://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/>>. Acesso em: 10 out. 2012.