Nagios em Ambientes de Alta Disponibilidade

Introdução

O Nagios é uma ferramenta de gestão de redes funcional e versátil, com um *GUI* (interface de utilizador gráfico) comparável a outras ferramentas comerciais. Por isto e por se tratar de *software* livre, é um candidato perfeito para empresas.

Ao ser utilizado em ambientes de produção, a sua componente de alta disponibilidade é muito desejada. No entanto, as soluções apresentadas na documentação oficial, não tornam esta ferramenta muito robusta às necessidades presentes neste tipo de ambientes. Aqui vamos propor uma solução mais eficiente. Primeiro vamos abordar as soluções já existentes e depois uma solução nova.

Este documento subentende que o leitor tem conhecimento prévio das soluções existentes para ambientes de alta disponibilidade (`http://nagios.sourceforge.net/docs/1_0/redundancy.html`).

Cenário 1 – Monitorização em Redundância

Ao aplicar esta solução deparamo-nos com diversas falhas. A mais evidente de todas é o aumento de informação redundante na rede que, por conseguinte, desperdiça a largura de banda disponível e aumenta a carga nas máquinas que estão a ser monitorizadas. Outra situação indesejável é ser possível que, após uma recuperação do processo *master*, duas máquinas enviem notificações. Isto é porque o processo *master* não tem qualquer conhecimento do processo *slave*, tornando necessário que o *slave* desactive prontamente as notificações. Por último, quando um dos processos nagios falha, não vai recuperar a informação obtida durante o período em que esteve inoperacional. Deste modo o histórico não é salvaguardado.

Cenário 2 - Monitorização por Falha

A diferença desta solução para o cenário 1 é que, não existem duas máquinas a fazer verificações. Deste modo, as máquinas monitorizadas não são submetidas a uma carga extra e a utilização da largura de banda é mais eficiente. No entanto, os problemas relacionados com o sincronismo do histórico e das notificações mantêm-se.

Cenário 3 – Monitorização Sincronizada por Falha

Introdução

Esta solução garante, na generalidade, o sincronismo consistente do histórico e que, o único processo nagios activo, é o que se encontra mais actualizado.

Propõe-se utilizar ferramentas de *software livre*, já existentes, como ponto de partida para solucionar o problema. Depois abordamos o método de controlo que elimina a redundância e garante o sincronismo dos processos.

Sincronismo do Histórico

Uma parte da solução é usar replicação em MySQL, o que resolve, em parte, o sincronismo do histórico. O problema é que o Nagios não guarda o histórico em bases de dados, mesmo quando compilado com MySQL. A solução é utilizar uma ferramenta que já faz isto – o PerfParse (`http://perfparse.sourceforge.net`), um *add on* do Nagios. O leitor terá de consultar a documentação do PerfParse e do MySQL (secção 4.11 da documentação oficial - replicação) para poder efectuar os procedimentos correctos.

A replicação em MySQL é cruzada, ou seja, cada servidor mysql é, ao mesmo tempo, *master* e *slave*. Esta garante a recuperação da informação em caso de fecho inapropriado de alguma máquina. Para isto acontecer deve colocar os seguintes parâmetros no ficheiro 'my.cnf':

```
log-bin
log-slave-updates
report-host=<nome_da_máquina>
replicate-do-db=<nome_da_base_de_dados_do_Nagios_e_do_PerfParse>
server-id=<1_ou_2__os_servidores_mysql_teem_de_ter_números_diferentes>
```

A sequência de eventos modelo para preparar esta parte da solução é:

1. instalar o Mysql e configurar o ficheiro 'my.cnf';
2. instalar o Nagios;
3. instalar os *plugins* do Nagios;
4. instalar o PerfParse (provavelmente vai obrigar a reconfigurar e reinstalar o Nagios);
5. reconfigurar o MySQL para sincronizar as bases de dados – configurar *masters* e *slaves* na consola mysql e inicializá-los.

A solução, que garante o sincronismo consistente do histórico, fica completa se forem utilizadas, convenientemente, estas duas ferramentas. Com isto é possível fazer sincronismo cruzado das bases de dados MySQL, que o Nagios+PerfParse utilizam. A configuração e versões escolhidas para trabalhar com estas ferramentas, influenciam muito o desempenho da solução em ambientes de alta disponibilidade. É aconselhável uma leitura cuidada sobre todas as ferramentas.

Sincronismo do Processo Nagios

Apesar de estar garantido o sincronismo do histórico, ainda existem algumas questões a resolver:

- garantir que apenas um processo nagios esteja a correr e a enviar informação para a base de dados, a qual é replicada;
- garantir que o processo nagios activo é o mais actualizado.

A forma encontrada foi compor dois *scripts* de controlo (`controller_master` e `controller_slave`), a correr separadamente em duas máquinas. Estes *scripts* têm a função de gerir a execução do processo nagios, e de tratar alguns erros. Para poderem ser úteis, é necessário que o `check_nrpe` e o *daemon* NRPE (a correr no arranque da máquina) estejam instalados nas máquinas. O *daemon* NRPE tem de cumprir as seguintes verificações:

- `check_nagios`
- `check_mysql`

Os *scripts* podem correr no cron:

```
* * * * * /usr/local/nagios/bin/controller_de_uma_das_maquinas
```

Os *scripts* de controlo devem ser optimizados no que se refere à sua execução. Com o propósito de mostrar esta optimização, apresentam-se os seus diagramas de actividades:

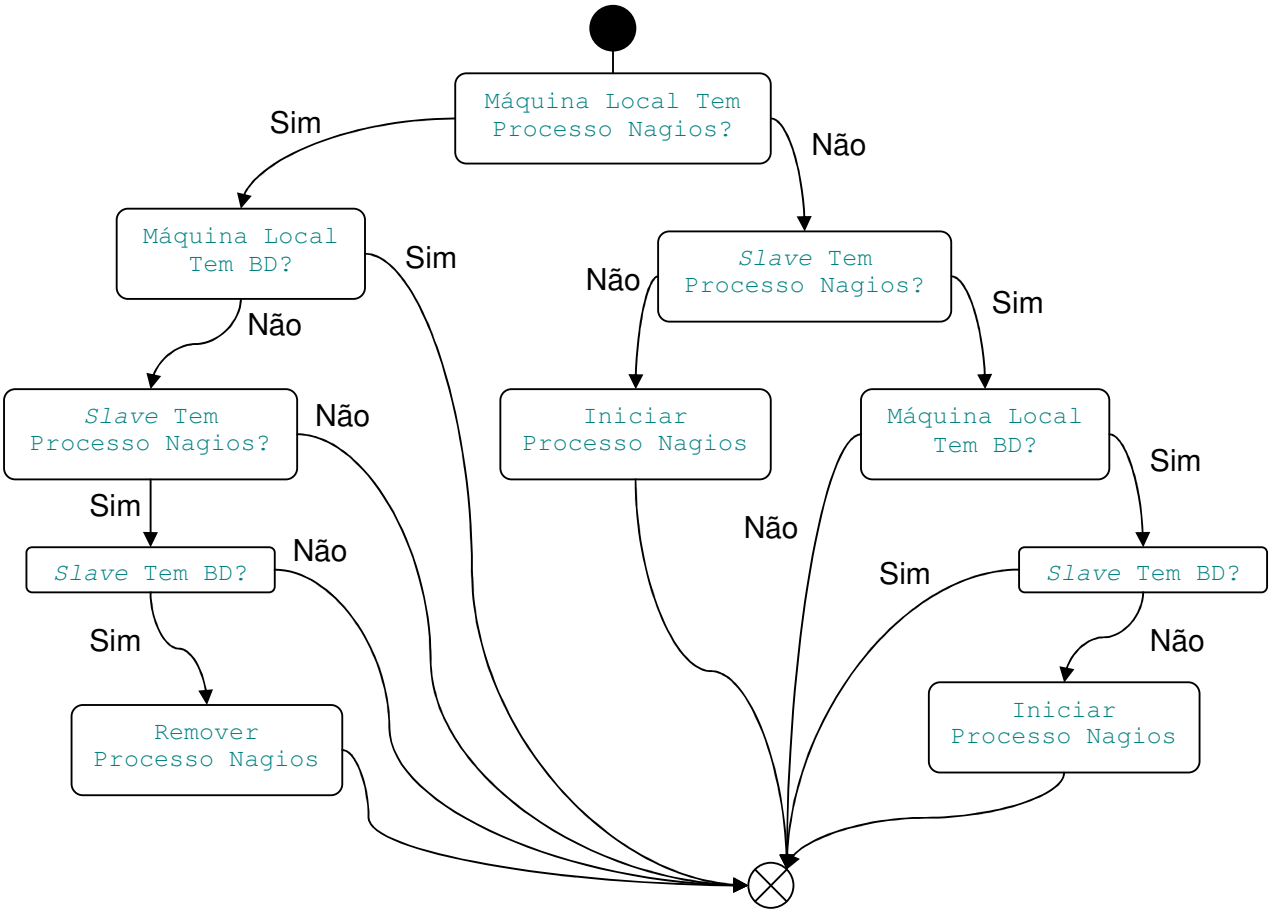

Diagrama de Actividades 1 – *Script* controller_master

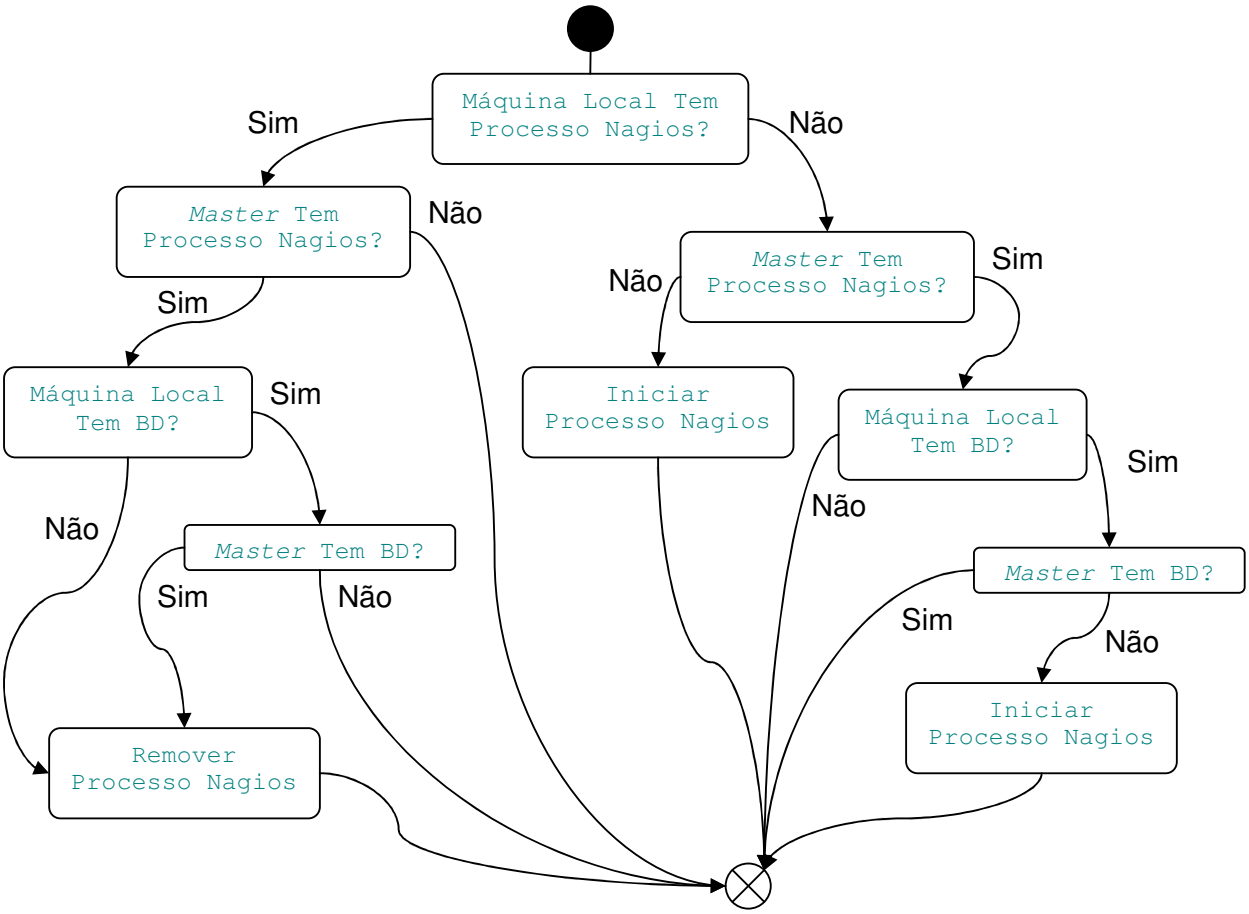

Diagrama de Actividades 2 – *Script* controller_slave

Concorrência dos Scripts Controladores

Os dois *scripts* não têm exactamente uma relação *master*/*slave* no seu comportamento. Na realidade, o *script master* só assume o domínio caso já se encontre a correr sem problemas, caso contrário tem o mesmo comportamento do *script slave*. Isto significa que os dois processos nagios podem ser dominantes alternadamente, em caso de falhas. Esta solução foi equacionada porque é necessário garantir que, o processo nagios a correr, é o que tem a base de dados mais actualizada.

Complexidade

O contra evidente desta solução é a sua complexidade. É exigida a manipulação de várias soluções aplicacionais. No entanto, esta pode ser uma mais valia na versatilidade de toda a solução.

Listagem de controller_master

```
#!/bin/sh

#       NAGIOS MASTER CONTROLLING SCRIPT

#       This script has the job of controlling Nagios Process in a High Availability Environment.
#       It must run integrated with other controlling systems; namely Nagios, NRPE, PerfParse, MySQL
#       with crossed replication and the other completing control script for the Slave Replica.
#       This solution is presented by Ricardo David Martins.

#       In order to run correctly, you must give the proper values to the next list of variables.

DEBUG=3
DEBUG_FILE=/var/log/messages
NAGIOS_WAITING_KILL_TIME=1
CHECK_NAGIOS=/usr/local/nagios/libexec/check_nagios
CHECK_NRPE=/usr/local/nagios/libexec/check_nrpe
CHECK_MYSQL=/usr/local/nagios/libexec/check_mysql
NAGIOS_LOG=/usr/local/nagios/var/nagios.log
NAGIOS_LOG_AGE_LIMIT=99999999
NAGIOS_COMMAND=/usr/local/nagios/bin/nagios
NAGIOS_CONFIG=/usr/local/nagios/etc/nagios.cfg
NAGIOS_EXT_FILE=/usr/local/nagios/var/rw/nagios.cmd
REMOTE_SLAVE_HOST=???.???.???.???
NAGIOS_DB=nagios
NAGIOS_DB_USER=nagios
NAGIOS_DB_PASSWORD=*****

# DO NOT change anything below this point, unless you know what you are doing.

function time_stamp()
{
        date '+%b %e %T'
}

#DEBUG MUST BE 3 OR GREATER
PREFIX_INFO="$(time_stamp) $HOSTNAME nagios: Master Control: INFORMATION:"

#DEBUG MUST BE 2 OR GREATER
PREFIX_WARNING="$(time_stamp) $HOSTNAME nagios Master Control: WARNING:"

#DEBUG MUST BE 1 OR GREATER
PREFIX_ERROR="$(time_stamp) $HOSTNAME nagios Master Control: ERROR:"

function check_local_nagios()
{
        $CHECK_NAGIOS -F $NAGIOS_LOG -C $NAGIOS_COMMAND -e $NAGIOS_LOG_AGE_LIMIT > /dev/null 2>&1
}

function check_remote_nagios()
{
        $CHECK_NRPE -H $REMOTE_SLAVE_HOST -c check_nagios > /dev/null 2>&1
}

function check_local_mysql()
{
        $CHECK_MYSQL -d $NAGIOS_DB -u $NAGIOS_DB_USER -p $NAGIOS_DB_PASSWORD > /dev/null 2>&1
}

function check_remote_mysql()
{
        $CHECK_NRPE -H $REMOTE_SLAVE_HOST -c check_mysql > /dev/null 2>&1
}

function kill_nagios()
{
        PID=`pidof -o %PPID $NAGIOS_COMMAND`
        kill -15 $PID > /dev/null 2>&1
}

function kill_forced_nagios()
{
         PID=`pidof -o %PPID $NAGIOS_COMMAND`
         kill -9 $PID > /dev/null 2>&1
}

function launch_nagios()
{
        $NAGIOS_COMMAND -d $NAGIOS_CONFIG > /dev/null 2>&1
}
```

```
function seek_nagios_external_file()
{
	[ -e $NAGIOS_EXT_FILE ]
}

function remove_nagios_external_file()
{
	rm -f $NAGIOS_EXT_FILE > /dev/null 2>&1
}

function solution_kill_nagios()
{
	kill_nagios
	sleep $NAGIOS_WAITING_KILL_TIME
	check_local_nagios
	error1=$?
	if [ $error1 -eq 0 ]; then
		if [ $DEBUG -gt 0 ]; then
			echo "$PREFIX_ERROR Não foi possível remover, normalmente, o processo nagios \
da máquina local." >> $DEBUG_FILE
		fi
		kill_forced_nagios
		sleep $NAGIOS_WAITING_KILL_TIME
		check_local_nagios
		error2=$?
		if [ $error2 -ne 0 -a $DEBUG -gt 1 ]; then
			echo "$PREFIX_WARNING O processo nagios foi removido forçadamente da máquina \
local." >> $DEBUG_FILE
		elif [ $error2 -eq 0 -a $DEBUG -gt 0 ]; then
			echo "$PREFIX_ERROR Não foi possível remover, forçadamente, o processo nagios \
da máquina local." >> $DEBUG_FILE
		fi
	elif [ $DEBUG -gt 1 ]; then
		echo "$PREFIX_WARNING O processo nagios foi removido da máquina local." >> $DEBUG_FILE
	fi
}

function solution_launch_nagios()
{
	launch_nagios
	check_local_nagios
	error3=$?
	if [ $error3 -ne 0 ]; then
		if [ $DEBUG -gt 0 ]; then
			echo "$PREFIX_ERROR Não foi possível lançar o processo nagios na máquina local." \
>> $DEBUG_FILE
		fi
		seek_nagios_external_file
		error4=$?
		if [ $error4 -eq 0 ]; then
			if [ $DEBUG -gt 1 ]; then
				echo "$PREFIX_WARNING É necessário remover o ficheiro de comandos \
externos do nagios." >> $DEBUG_FILE
			fi
			remove_nagios_external_file
			seek_nagios_external_file
			error5=$?
			if [ $error5 -ne 0 ]; then
				if [ $DEBUG -gt 1 ]; then
					echo "$PREFIX_WARNING O ficheiro de comandos externos do nagios \
foi removido" >> $DEBUG_FILE
				fi
				launch_nagios
				check_local_nagios
				error6=$?
				if [ $error6 -eq 0 -a $DEBUG -gt 1 ]; then
					echo "$PREFIX_WARNING O processo nagios foi lançado na máquina \
local." >> $DEBUG_FILE
				elif [ $error6 -ne 0 -a $DEBUG -gt 0 ]; then

					echo "$PREFIX_ERROR Não foi possível lançar o processo nagios na \
máquina local." >> $DEBUG_FILE
				fi
			elif [ $DEBUG -gt 0 ]; then
				echo "$PREFIX_ERROR Não foi possível remover o ficheiro de comandos \
externos do nagios." >> $DEBUG_FILE
			fi
		elif [ $DEBUG -gt 0 ]; then
			echo "$PREFIX_ERROR Não é possível lançar o processo nagios por motivo \
desconhecido." >> $DEBUG_FILE
		fi
	elif [ $DEBUG -gt 1 ]; then
		echo "$PREFIX_WARNING O processo nagios foi lançado na máquina local." >> $DEBUG_FILE
	fi
```

```
}


#*********************************************************
#Test if nagios process is running OK on the local machine*
#*********************************************************
check_local_nagios
error7=$?
if [ $error7 -eq 0 ]; then
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO A máquina local está a correr o processo nagios." >> $DEBUG_FILE
        fi

        #*****************************************************************
        #Test if nagios mysql data base is running OK on the local machine*
        #*****************************************************************
        check_local_mysql
        error8=$?
        if [ $error8 -ne 0 ]; then
                if [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR A base de dados local não está a correr." >> $DEBUG_FILE
                fi

                #*******************************************************
                #Test if nagios process is running OK on the remote host*
                #*******************************************************
                check_remote_nagios
                error9=$?
                if [ $error9 -eq 0 ]; then
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO A máquina remota está a correr o processo nagios." >> \
$DEBUG_FILE
                        fi

                        #*****************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*****************************************************************
                        check_remote_mysql
                        error10=$?
                        if [ $error10 -eq 0 ]; then
                                if [ $DEBUG -gt 2 ]; then
                                        echo "$PREFIX_INFO A base de dados remota está a correr." >> \
$DEBUG_FILE
                                fi
                                solution_kill_nagios
                        elif [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR A base de dados remota não está a correr." >> \
$DEBUG_FILE
                        fi

                elif [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO A máquina remota não está a correr o processo nagios." >> \
$DEBUG_FILE
                fi
        elif [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO A base de dados local está a correr." >> $DEBUG_FILE
        fi
else
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO A máquina local não está a correr o processo nagios." >> $DEBUG_FILE
        fi

        #*******************************************************
        #Test if nagios process is running OK on the remote host*
        #*******************************************************
        check_remote_nagios
        error11=$?
        if [ $error11 -eq 0 ]; then
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO A máquina remota está a correr o processo nagios." >> \
$DEBUG_FILE
                fi

                #*****************************************************************
                #Test if nagios mysql data base is running OK on the local machine*
                #*****************************************************************
                check_local_mysql
                error12=$?
                if [ $error12 -eq 0 ]; then
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO A base de dados local está a correr." >> $DEBUG_FILE
                        fi

                        #*************************************************************
```

```
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*************************************************************
                        check_remote_mysql
                        error13=$?
                        if [ $error13 -ne 0 ]; then
                                if [ $DEBUG -gt 0 ]; then
                                        echo "$PREFIX_ERROR A base de dados remota não está a correr." >> \
$DEBUG_FILE
                                fi
                                solution_launch_nagios
                        elif [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO A base de dados remota está a correr." >> $DEBUG_FILE
                        fi
                elif [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR A base de dados local não está a correr." >> $DEBUG_FILE
                fi
        else
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO A máquina remota não está a correr o processo nagios." >> \
$DEBUG_FILE
                fi
                solution_launch_nagios
        fi
fi

#*****************************************************
#Test if nrpe is running on the local and remote hosts*
#*****************************************************
if [ $DEBUG -gt 0 ]; then
        $CHECK_NRPE -H localhost
        error14=$?
        if [ $error14 -ne 0 ]; then
                echo "$PREFIX_ERROR A máquina local não tem o servidor de NRPE activo." >> $DEBUG_FILE
        fi
        $CHECK_NRPE -H $REMOTE_SLAVE_HOST
        error15=$?
        if [ $error15 -ne 0 ]; then
                echo "$PREFIX_ERROR A máquina remota não tem o servidor de NRPE activo." >> $DEBUG_FILE
        fi
fi
```

Listagem do controller_slave

```
#!/bin/sh

#       NAGIOS SLAVE CONTROLLING SCRIPT

#       This script has the job of controlling Nagios Process in a High Availability Environment.
#       It must run integrated with other controlling systems; namely Nagios, NRPE, PerfParse, MySQL
#       with crossed replication and the other completing control script for the Master Replica.
#       This solution is presented by Ricardo David Martins.

#       In order to run correctly, you must give the proper values to the next list of variables.

DEBUG=3
DEBUG_FILE=/var/log/messages
NAGIOS_WAITING_KILL_TIME=1
CHECK_NAGIOS=/usr/local/nagios/libexec/check_nagios
CHECK_NRPE=/usr/local/nagios/libexec/check_nrpe
CHECK_MYSQL=/usr/local/nagios/libexec/check_mysql
NAGIOS_LOG=/usr/local/nagios/var/nagios.log
NAGIOS_LOG_AGE_LIMIT=99999999
NAGIOS_COMMAND=/usr/local/nagios/bin/nagios
NAGIOS_CONFIG=/usr/local/nagios/etc/nagios.cfg
NAGIOS_EXT_FILE=/usr/local/nagios/var/rw/nagios.cmd
REMOTE_MASTER_HOST=???.???.???.???
NAGIOS_DB=nagios
NAGIOS_DB_USER=nagios
NAGIOS_DB_PASSWORD=*****

# DO NOT change anything below this point, unless you know what you are doing.

function time_stamp()
{
        date '+%b %e %T'
}

#DEBUG MUST BE 3 OR GREATER
PREFIX_INFO="$(time_stamp) $HOSTNAME nagios: Slave Control: INFORMATION:"

#DEBUG MUST BE 2 OR GREATER
PREFIX_WARNING="$(time_stamp) $HOSTNAME nagios: Slave Control: WARNING:"

#DEBUG MUST BE 1 OR GREATER
PREFIX_ERROR="$(time_stamp) $HOSTNAME nagios: Slave Control: ERROR:"

function check_local_nagios()
{
        $CHECK_NAGIOS -F $NAGIOS_LOG -C $NAGIOS_COMMAND -e $NAGIOS_LOG_AGE_LIMIT > /dev/null 2>&1
}

function check_remote_nagios()
{
        $CHECK_NRPE -H $REMOTE_MASTER_HOST -c check_nagios > /dev/null 2>&1
}

function check_local_mysql()
{
        $CHECK_MYSQL -d $NAGIOS_DB -u $NAGIOS_DB_USER -p $NAGIOS_DB_PASSWORD > /dev/null 2>&1
}

function check_remote_mysql()
{
        $CHECK_NRPE -H $REMOTE_MASTER_HOST -c check_mysql > /dev/null 2>&1
}

function kill_nagios()
{
        PID=`pidof -o %PPID $NAGIOS_COMMAND`
        kill -15 $PID > /dev/null 2>&1
}

function kill_forced_nagios()
{
         PID=`pidof -o %PPID $NAGIOS_COMMAND`
         kill -9 $PID > /dev/null 2>&1
}

function launch_nagios()
{
        $NAGIOS_COMMAND -d $NAGIOS_CONFIG > /dev/null 2>&1
}
```

```
function seek_nagios_external_file()
{
       [ -e $NAGIOS_EXT_FILE ]
}

function remove_nagios_external_file()
{
       rm -f $NAGIOS_EXT_FILE > /dev/null 2>&1
}

function solution_kill_nagios()
{
       kill_nagios
       sleep $NAGIOS_WAITING_KILL_TIME
       check_local_nagios
       error1=$?
       if [ $error1 -eq 0 ]; then
              if [ $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR Não foi possível remover, normalmente, o processo nagios da \
máquina local." >> $DEBUG_FILE
              fi
              kill_forced_nagios
              sleep $NAGIOS_WAITING_KILL_TIME
              check_local_nagios
              error2=$?
              if [ $error2 -ne 0 -a $DEBUG -gt 1 ]; then
                     echo "$PREFIX_WARNING O processo nagios foi removido forçadamente da máquina \
local." >> $DEBUG_FILE
              elif [ $error2 -eq 0 -a $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR Não foi possível remover, forçadamente, o processo nagios da \
máquina local." >> $DEBUG_FILE
              fi
       elif [ $DEBUG -gt 1 ]; then
              echo "$PREFIX_WARNING O processo nagios foi removido da máquina local." >> $DEBUG_FILE
       fi
}

function solution_launch_nagios()
{
       launch_nagios
       check_local_nagios
       error3=$?
       if [ $error3 -ne 0 ]; then
              if [ $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR Não foi possível lançar o processo nagios na máquina local." \
>> $DEBUG_FILE
              fi
              seek_nagios_external_file
              error4=$?
              if [ $error4 -eq 0 ]; then
                     if [ $DEBUG -gt 1 ]; then
                            echo "$PREFIX_WARNING É necessário remover o ficheiro de comandos \
externos do nagios." >> $DEBUG_FILE
                     fi
                     remove_nagios_external_file
                     seek_nagios_external_file
                     error5=$?
                     if [ $error5 -ne 0 ]; then
                            if [ $DEBUG -gt 1 ]; then
                                   echo "$PREFIX_WARNING O ficheiro de comandos externos do nagios \
foi removido" >> $DEBUG_FILE
                            fi
                            launch_nagios
                            check_local_nagios
                            error6=$?
                            if [ $error6 -eq 0 -a $DEBUG -gt 1 ]; then
                                   echo "$PREFIX_WARNING O processo nagios foi lançado na máquina \
local." >> $DEBUG_FILE
                            elif [ $error6 -ne 0 -a $DEBUG -gt 0 ]; then

                                   echo "$PREFIX_ERROR Não foi possível lançar o processo nagios na \
máquina local." >> $DEBUG_FILE
                            fi
                     elif [ $DEBUG -gt 0 ]; then
                            echo "$PREFIX_ERROR Não foi possível remover o ficheiro de comandos \
externos do nagios." >> $DEBUG_FILE
                     fi
              elif [ $DEBUG -gt 0 ]; then
                     echo "$PREFIX_ERROR Não é possível lançar o processo nagios por motivo \
desconhecido." >> $DEBUG_FILE
              fi
       elif [ $DEBUG -gt 1 ]; then
              echo "$PREFIX_WARNING O processo nagios foi lançado na máquina local." >> $DEBUG_FILE
       fi
```

```
}


#*********************************************************
#Test if nagios process is running OK on the local machine*
#*********************************************************
check_local_nagios
error7=$?
if [ $error7 -eq 0 ]; then
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO A máquina local está a correr o processo nagios." >> $DEBUG_FILE
        fi

        #*********************************************************
        #Test if nagios process is running OK on the remote host*
        #*********************************************************
        check_remote_nagios
        error9=$?
        if [ $error9 -eq 0 ]; then
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO A máquina remota está a correr o processo nagios." >> \
$DEBUG_FILE
                fi

                #*****************************************************************
                #Test if nagios mysql data base is running OK on the local machine*
                #*****************************************************************
                check_local_mysql
                error8=$?
                if [ $error8 -ne 0 ]; then
                        if [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR A base de dados local não está a correr." >> \
$DEBUG_FILE
                        fi
                        solution_kill_nagios
                else
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO A base de dados local está a correr." >> $DEBUG_FILE
                        fi

                        #*****************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #*****************************************************************
                        check_remote_mysql
                        error10=$?
                        if [ $error10 -eq 0 ]; then
                                if [ $DEBUG -gt 2 ]; then
                                        echo "$PREFIX_INFO A base de dados remota está a correr." >> \
$DEBUG_FILE
                                fi
                                solution_kill_nagios
                        elif [ $DEBUG -gt 0 ]; then
                                echo "$PREFIX_ERROR A base de dados remota não está a correr." >> \
$DEBUG_FILE
                        fi
                fi
        elif [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO A máquina remota não está a correr o processo nagios." >> $DEBUG_FILE
        fi
else
        if [ $DEBUG -gt 2 ]; then
                echo "$PREFIX_INFO A máquina local não está a correr o processo nagios." >> $DEBUG_FILE
        fi

        #*********************************************************
        #Test if nagios process is running OK on the remote host*
        #*********************************************************
        check_remote_nagios
        error11=$?
         if [ $error11 -eq 0 ]; then
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO A máquina remota está a correr o processo nagios." >> \
$DEBUG_FILE
                fi

                #*****************************************************************
                #Test if nagios mysql data base is running OK on the local machine*
                #*****************************************************************
                check_local_mysql
                error12=$?
                if [ $error12 -eq 0 ]; then
                        if [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO A base de dados local está a correr." >> $DEBUG_FILE
                        fi
```

```
                        #************************************************************
                        #Test if nagios mysql data base is running OK on the remote host*
                        #************************************************************
                        check_remote_mysql
                        error13=$?
                        if [ $error13 -ne 0 ]; then
                                if [ $DEBUG -gt 0 ]; then
                                        echo "$PREFIX_ERROR A base de dados remota não está a correr." >> \
$DEBUG_FILE
                                fi
                                solution_launch_nagios
                        elif [ $DEBUG -gt 2 ]; then
                                echo "$PREFIX_INFO A base de dados remota está a correr." >> $DEBUG_FILE
                        fi
                elif [ $DEBUG -gt 0 ]; then
                        echo "$PREFIX_ERROR A base de dados local não está a correr." >> $DEBUG_FILE
                fi
        else
                if [ $DEBUG -gt 2 ]; then
                        echo "$PREFIX_INFO A máquina remota não está a correr o processo nagios." >> \
$DEBUG_FILE
                fi
                solution_launch_nagios
        fi
fi

#*****************************************************
#Test if nrpe is running on the local and remote hosts*
#*****************************************************
if [ $DEBUG -gt 0 ]; then
        $CHECK_NRPE -H localhost
        error14=$?
        if [ $error14 -ne 0 ]; then
                echo "$PREFIX_ERROR A máquina local não tem o servidor de NRPE activo." >> $DEBUG_FILE
        fi
        $CHECK_NRPE -H $REMOTE_MASTER_HOST
        error15=$?
        if [ $error15 -ne 0 ]; then
                echo "$PREFIX_ERROR A máquina remota não tem o servidor de NRPE activo." >> $DEBUG_FILE
        fi
fi
```