2 Expressões de Trigger

Visão geral

As expressões de triggers são muito flexíveis. Você pode utiliza-las para criar testes com lógicas complexas em relação a dados e estatísticas da monitoração.

A sintaxe básica de uma expressão pode ser definida assim:

```
{<server>:<key>.<function>(<parameter>)}<operator><constant>
```

1 Funções

As funções de Trigger permitem referenciar dados coletados, horário atual e outros fatores.

Uma lista completa das funções suportadas está disponível.

2 Parâmetros das funções

A maioria das funções numéricas aceita uma quantidade de segundos como parâmetro (uma unidade de tempo).

Você pode utilizar o prefixo **#** para especificar um parâmetro com significado diferente:

Chamada da Função	Significa
sum(600)	Sumarização de todos os valores nos últimos 600 segundos (10 minutos)
sum(#5)	Sumarização dos últimos 5 valores

A função **last** utiliza uma forma diferente para valores prefixados com **#**, para ela isso indicará que ela deverá recuperar o enézimo valor anterior (do mais recente para o mais antigo). Exemplo, suponhamos que os últimos 10 valores coletados são: (2, 3, 4, 9, 10, 11, 11, 16, 7, 0)

- **last(#2)** - irá retornar o penúltimo valor *7*
- **last(#5)** - irá retornar *10*.

As funções **avg**, **count**, **last**, **min** e **max** possuem um parâmetro adicional: o `time_shift`. Este parâmetro permite referenciar o dado em determinado período de tempo no passado. Por exemplo, **avg(1h,1d)** irá retornar o valor médio analisando 1 hora de valores de 1 dia antes do momento da coleta.

As triggers só analisam os dados que estão no histórico. Se o dado desejado não estiver no histórico, mas estiver nas médias, a informação das médias **não será utilizada**. Logo, é necessário que se mantenha o histórico por tempo compatível com as triggers que se deseja criar.

Você pode utilizar os símbolos de unidades nas expressões, por exemplo '5m' (minutos) ao invés de '300' segundos ou '1d' (dia) ao invés de '86400' segundos, '1K' ao invés de '1024' bytes.

3 Operadores

Os seguintes operadores são suportados nas expressões de triggers **(em ordem descedente de prioridade de execução)**:

Prioridade	Operador	Definição
1	**-**	Símbolo de negativo
2	**not**	Não Lógico
3	*	Multiplicação
	/	Divisão
4	**+**	Soma aritmética
	-	Redução aritmética
5	**<**	Menor que. O operador é definido como: A<B ⇔ (A<=B-0.000001)
	<=	Menor ou igual a.
	>	Maior que. O operador é definido como: A>B ⇔ (A>=B+0.000001)
	>=	Maior ou igual a.
6	**=**	É igual. O operador é definido como: A=B ⇔ (A>B-0.000001) e (A<B+0.000001)
	<>	Diferente. O operador é definido comoas: A<>B ⇔ (A<=B-0.000001) ou (A>=B+0.000001)
7	**and**	Operador lógico E
8	**or**	Operador lógico OU

Os operadores **not**, **and** e **or** são sensíveis ao caso e deverão ser escritos em letra minúscula. Eles devem estar envoltos em espaços ou parênteses.

Todos os operadores, exceto o “Símbolo de negativo” e “Não lógico”, tem associação da esquerda para a direita.

4 Exemplos de triggers

Exemplo 1

Análise de carga de processamento na CPU: “Processor load is too high on www.zabbix.com”

```
{www.zabbix.com:system.cpu.load[all,avg1].last()}>5
```

'www.zabbix.com:system.cpu.load[all,avg1]' provê um nome curto para o parâmetro monitorado. Neste caso se refere ao host 'www.zabbix.com' com a chave 'system.cpu.load[all,avg1]'. Utilizando a função 'last()', nós estaremos referindo ao valor mais recente. Finalmente, '>5' define que a trigger deverá ir para o estado de “INCIDENTE” quando o valor mais recente desta chave, neste host for superior a 5.

Exemplo 2

www.zabbix.com is overloaded

```
{www.zabbix.com:system.cpu.load[all,avg1].last()}>5 ou
{www.zabbix.com:system.cpu.load[all,avg1].min(10m)}>2
```

A expressão será verdadeira quando o último valor coletado para a carga da CPU for superior a 5 ou superior a 2 nos últimos 10 coletados.

Exemplo 3

/etc/passwd has been changed

Utilize a função 'diff()':

```
{www.zabbix.com:vfs.file.cksum[/etc/passwd].diff()}=1
```

A expressão será verdadeira quando o último valor da verificação 'checksum' do arquivo '/etc/passwd' for diferente da penúltima verificação.

De forma similar esta técnica pode ser utilizada para monitorar vários outros arquivos, tais quais: /etc/inetd.conf, /kernel, etc.

Exemplo 4

Alguém está baixando um arquivo muito grande da internet (ou um tráfego intenso por um longo período)

Utilize a função 'min()':

```
{www.zabbix.com:net.if.in[eth0,bytes].min(5m)}>100K
```

A expressão será verdadeira quando a quantidade de bytes recebidos nos últimos 5 minutos na interface 'eth0' for superior a 100 KB.

Exemplo 5

Ambos os nós do cluster de SMTP estão indisponíveis

Observe que a expressão utiliza dados de dois hosts diferentes:

```
{smtp1.zabbix.com:net.tcp.service[smtp].last()}=0 and
{smtp2.zabbix.com:net.tcp.service[smtp].last()}=0
```

A expressão será verdadeira quando ambos os servidores (smtp1.zabbix.com e smtp2.zabbix.com)

SMTP estiverem fora do ar.

Exemplo 6

A versão do Zabbix Agent precisa ser atualizada

Use a função 'str()':

```
{zabbix.zabbix.com:agent.version.str("beta8")}=1
```

A expressão será verdadeira se a versão do Zabbix Agent possuir o texto “beta8” (por exemplo 1.0beta8).

Exemplo 7

Servidor indisponível

```
{zabbix.zabbix.com:icmpping.count(30m,0)}>5
```

A expressão será verdadeira se o host “zabbix.zabbix.com” estiver inacessível por mais de 5 vezes nos últimos 30 minutos.

Exemplo 8

Sem dados nos últimos 3 minutos

Use a função 'nodata()':

```
{zabbix.zabbix.com:tick.nodata(3m)}=1
```

Neste exemplo 'tick' é um item do tipo 'Zabbix trapper'. Para que esta trigger funcione o item 'tick' precisará ter sido definido. O host precisará enviar periodicamente o dado para este item através do comando 'zabbix_sender' ou similar.

A expressão será verdadeira se nenhum dado for recebido nos últimos 180 segundos.

Exemplo 9

Alta carga de CPU no período noturno

Utilize a função 'time()':

```
{zabbix:system.cpu.load[all,avg1].min(5m)}>2 and
{zabbix:system.cpu.load[all,avg1].time()}>000000 and
{zabbix:system.cpu.load[all,avg1].time()}<060000
```

A expressão será verdadeira se a carga de CPU for superior a 2, entre a meia noite e as seis da manhã.

Exemplo 10

Verifica se o horário local do host monitorado e do servidor do Zabbix estão sincronizados

Use a função 'fuzzytime()':

```
{MySQL_DB:system.localtime.fuzzytime(10)}=0
```

A expressão será verdadeira se o horário do servidor 'MySQL_DB' tiver uma diferença maior que 10 segundos em relação ao horário do Zabbix Server.

Exemplo 11

Comparando a carga atual de CPU com a carga no mesmo horário do dia anterior (usando o parâmetro de `time_shift`).

```
{server:system.cpu.load.avg(1h)}/{server:system.cpu.load.avg(1h,1d)}>2
```

A expressão será verdadeira se a carga da última hora for duas vezes superior a carga deste mesmo período um dia antes (24 horas).

Exemplo 12

Usando o valor de outro item como limite para a trigger:

```
{Template PfSense:hrStorageFree[{#SNMPVALUE}].last()}<{Template
PfSense:hrStorageSize[{#SNMPVALUE}].last()}*0.1
```

A expressão será verdadeira se o espaço livre for inferior a 10%.

5 Técnicas 'anti-flapping'

Algumas vezes você precisa ter condições diferentes para estados diferentes (INCIDENTE/OK). Por exemplo, nós podemos ter que definir uma trigger para avisar quando a temperatura de uma sala for superior a 20C (vinte graus) que é o máximo suportável para os servidores funcionarem com segurança, mas a temperatura ideal de funcionamento deveria ser de até 15C (quinze graus). Temos como definir uma trigger desta forma no Zabbix, ela será ativada (mudar para o estado de INCIDENTE) se a temperatura ultrapassar o máximo aceitável, mas não será inativada (retornar ao estado OK) enquanto a temperatura não for inferior à temperatura ideal.

Para fazer isso podemos definir uma trigger como a do “Exemplo 1”. A trigger do “Exemplo 2” apresenta a mesma técnica de “anti-flapping” para espaço em disco.

Exemplo 1

A temperatura na sala dos servidores está muito alta

```
({TRIGGER.VALUE}=0 and {server:temp.last()}>20) or
({TRIGGER.VALUE}=1 and {server:temp.last()}>15)
```

Exemplo 2

Pouco espaço livre no disco

Incidente: se for menor que 10GB nos últimos 5 minutos

Recuperação (OK): se for maior que 40GB nos últimos 10 minutos

```
({TRIGGER.VALUE}=0 and {server:vfs.fs.size[/,free].max(5m)}<10G) or
({TRIGGER.VALUE}=1 and {server:vfs.fs.size[/,free].min(10m)}<40G)
```

Observe que a macro *{TRIGGER.VALUE}* retorna o estado corrente da trigger (0 - OK, 1 - Incidente).