

## 8 Triggers - Funções preditivas

### Visão geral

Algumas vezes existem sinais de problemas que estão chegando. Estes sinais podem ser percebidos e possibilitar a execuções de ações preventivas ou minimizadoras dos impactos do incidente.

O Zabbix tem ferramentas para capazes de prever o comportamento futuro do ambiente monitorado a partir de seus dados históricos. Estas ferramentas são chamadas de funções preditivas.

### 1 Funções

Para trabalhar com estas funções nós precisamos definir duas coisas: como detectar o futuro estado de problema e quanto tempo é necessário para as contra-medidas. Uma vez que tenhamos ambas as informações temos duas formas de configurar a sinalização da trigger para uma situação indesejável. \* Primeiro: a trigger pode ser disparada após o tempo limite em que se espera que o próprio sistema se recupere. \* Segundo: a trigger pode ser disparada antes do tempo limite definido. As funções de trigger correspondentes são: **forecast** e **timeleft**. Observe que a análise estatística subjacente é basicamente idêntica para ambas as funções. Você pode configurar a trigger da forma que preferir, com resultados semelhantes

### 2 Parâmetros

Ambas as funções utilizam o mesmo conjunto de parâmetros. Utilize a lista de [funções suportadas](#) como referência.

#### 2.1 Intervalo de tempo

O primeiro passo é definir um período histórico que o Zabbix irá analisar para a previsão. Você pode configurar isso de formas conhecidas como os parâmetros `sec` ou `#num` e com o opcional `time_shift` da mesma forma que faz com as funções **avg**, **count**, **delta**, **max**, **min** e **sum**.

#### 2.2 Previsão de horizonte

(**forecast** apenas)

O parâmetro `time` especifica quão longe no futuro o Zabbix deverá extrapolar as dependências que encontrar nos dados históricos. Não importa se você utilizar o parâmetro `time_shift` ou não, `time` sempre será contabilizado a partir do momento corrente.

#### 2.3 Limiar a alcançar

(**timeleft** apenas)

O parâmetro `threshold` especifica um valor analisado que determinado item tem que chegar, nao

importa se for superior ou inferior. Uma vez que determinemos 'f(t)' (veja a seguir) que deve resolver a equação 'f(t) = threshold' e retornar que a origem está mais proxima de agora ou mais distante de agora até 999999999999.9999 se não for encontrada a origem.

Quando os valores do item se aproximarem do limiar e, em seguida, ultrapassa-lo **timeleft** assume que a intersecção já é passada e muda o estado para a próxima intersecção limite, se houver. É uma boa prática utilizar as previsões de forma complementar à detecção normal de incidentes, não como uma substituição.<sup>1)</sup>

## 2.4 Funções de ajustes

O 'ajuste' (fit) padrão é uma função *linear*. Mas se seu sistema monitorado é mais complicado existem outras opções.

fit	x = f(t)
<i>linear</i>	$x = a + b*t$
<i>polynomial</i> <sup>2)</sup>	$x = a_0 + a_1*t + a_2*t^2 + \dots + a_n*t^n$
<i>exponential</i>	$x = a*\exp(b*t)$
<i>logarithmic</i>	$x = a + b*\log(t)$
<i>power</i>	$x = a*t^b$

## 2.5 Modos

(**forecast** Apenas)

Toda vez que uma função de trigger é avaliada ela recebe dados de determinado período do histórico e utiliza uma função nos dados. Assim, se os dados estiverem ligeiramente diferentes, os resultados também o serão. Se nós simplesmente calcularmos o valor da função de ajuste em um determinado tempo no futuro você não verá nada sobre como o item analisado se comportará até lá. Em alguns modos de ajuste (tal qual o *polynomial*) uma visão simples do futuro pode ser enganosa.

mode	forecast resultado
<i>value</i>	$f(\text{now} + \text{time})$
<i>max</i>	$\max_{\text{now} \leq t \leq \text{now} + \text{time}} f(t)$
<i>min</i>	$\min_{\text{now} \leq t \leq \text{now} + \text{time}} f(t)$
<i>delta</i>	$\text{max} - \text{min}$
<i>avg</i>	média de f(t) ( $\text{now} \leq t \leq \text{now} + \text{time}$ ) de acordo com <a href="#">a definição</a>

## 3 Detalhes

Para evitar cálculos com números imensos nós consideramos que o carimbo de hora do primeiro valor especificado no período somado a 1 ns como um novo momento zero (o 'epoch' atual é da ordem de  $10^9$ , o quadrado do 'epoch'  $10^{18}$ , a precisão dupla é de cerca de  $10^{-16}$ ). 1 ns é adicionado para prover todos os valores positivos de tempo para ajustar *logarithmic* e *power* encaixando no cálculo do log(t). A mudança de tempo não afeta *linear*, *polynomial*, *exponential* (além de cálculos mais fáceis e precisos) mas muda a forma das funções de *logarithmic* e *power*.

## 4 Erros potenciais

As funções retornam -1 nas seguintes situações:

- o período de avaliação definido não contém dados;
- o resultado da operação matemática não está definido ( por exemplo o ajuste de funções *exponential* ou *power* envolvem o cálculo do  $\log()$  de valores de itens. Se o dado contém valores zerados ou negativos você receberá um erro uma vez que a função de  $\log()$  é definida apenas para valores positivos.);
- complicações numéricas (infelizmente, para alguns conjuntos de intervalos de dados de entrada a precisão dupla e ponto flutuante se mostram insuficientes) <sup>3)</sup>.

Não há avisos ou erros sinalizados se o ajuste escolhido descrever mal os dados escolhidos ou existirem poucos dados para uma previsão precisa.

## 5 Exemplos e tratativa de erros

Para receber um aviso quando você está prestes a ficar sem espaço em disco em um host você pode criar uma expressão de trigger similar à esta:

```
{host:vfs.fs.size[/,free].timeleft(1h,,0)}<1h
```

No entanto, o código de erro -1 pode surgir e habilitar a sua trigger em um estado de 'incidente'. Normalmente isso pode ser bom pois você receberá um aviso que suas predições não estão funcionando corretamente e você deverá analisa-la com mais cuidado. Mas em algumas vezes o -1 pode ser ruim pois pode indicar simplesmente que não existiam dados sobre espaço livre em disco coletados na última hora. Caso ocorram falso-positivos pode ser interessante o uso de expressões de disparo mais complexas:

```
{host:vfs.fs.size[/,free].timeleft(1h,,0)}<1h and ({TRIGGER.VALUE}=0 and {host:vfs.fs.size[/,free].timeleft(1h,,0)}<>-1 or {TRIGGER.VALUE}=1)
```

)):

```
{host:vfs.fs.size[/,free].timeleft(1h,,0)}<1h and {host:vfs.fs.size[/,free].timeleft(1h,,0)}<>-1
```

A situação é um pouco mais difícil com **forecast**. Primeiramente, -1 pode ou não colocar a trigger em estado de 'incidente' dependendo de ter ou não uma expressão similar a:

```
{host:item.forecast(...)}<...
```

or like

```
{host:item.forecast(...)}>...
```

Além disso, -1 pode ser uma projeção válida se é normal para o valor do item a ser negativo. Mas a

probabilidade de esta situação ocorrer no mundo real é insignificante (veja [como](#) o operador = funciona). Então adicione

```
... or {host:item.forecast(...)}=-1
```

ou

```
... and {host:item.forecast(...)}<>-1
```

se você desejar ou não tratar o valor -1 como um problema.

## Veja também

1. [Funções preditivas \(pdf\)](#) em [zabbix.org](http://zabbix.org)

1)

Por exemplo, um código simples de trigger como

```
{host:item.timeleft(1h,,X)} < 1h
```

pode entrar em estado de 'incidente' quando o valor do item X subitamente se aproximar de X e subtamente se recuperar quando se afastar de X. Se o problema é o valor do item abaixo de X:

```
{host:item.last()} < X or {host:item.timeleft(1h,,X)} < 1h
```

Se o problema é o valor do item acima de X use:

```
{host:item.last()} > X or {host:item.timeleft(1h,,X)} < 1h
```

2)

O grau polimomial pode ser de 1 a 6, *polynomial1* é equivalente a linear. Mas é possível utilizar polimônios mais elevados [com cautela](#). Se o período de análise contiver menos pontos do que o necessário para determinar os coeficientes polimoniais o grau polimomial será reduzido (ex. é solicitado *polynomial5*, mas só existem 4 pontos, logo será alcançado o *polynomial3*).

3)

Para *linear*, *exponential*, *logarithmic* e *power* o ajuste de todos os cálculos podem ser escritos explicitamente. Para *polynomial* apenas *value* pode ser calculado sem passos adicionais. Calculo de *avg* envolve computação polimomial antiderivada (analítica). Computando *max*, *min* e *delta* envolve computação polimomial derivativa (analiticamente) e encontrar suas raízes (numéricas). A solução de 'f(t) = 0' implica encontrar raízes polimoniais (numéricas).

From:

<https://www.zabbix.com/documentation/4.0/> - **Zabbix Documentation 4.0**

Permanent link:

<https://www.zabbix.com/documentation/4.0/pt/manual/config/triggers/prediction>

Last update: **2017/10/06 14:12**

