

## 2 Выражение триггера

### Обзор

Используемые в триггерах выражения являются очень гибкими. Вы можете использовать их для создания сложных логических тестов, учитывая статистику по мониторингу.

Простое полезное выражение может выглядеть примерно так:

```
{<сервер>:<ключ>.<функция>(<параметр>)}<оператор><константа>
```

### 1 Функции

Функции триггеров позволяют ссылаться на собранные значения, текущее время и другие факторы.

Имеется полный список [поддерживаемых функций](#).

### 2 Параметры функций

Большинство числовых функций принимают количество секунд в качестве параметра.

Вы можете использовать префикс **#**, чтобы указать что этот параметр должен иметь другой смысл:

| ВЫЗОВ ФУНКЦИИ   | СМЫСЛ                             |
|-----------------|-----------------------------------|
| <b>sum(600)</b> | Сумма всех значений за 600 секунд |
| <b>sum(#5)</b>  | Сумма последних 5 значений        |

Функция **last** использует другой смысл для значений, когда начинается с решетки - она дает выбрать n-ое предыдущее значение, так что с учетом значений 3, 7, 2, 6, 5 (от наиболее нового до наиболее старого), при **last(#2)** вернется 7 и при **last(#5)** вернется 5.

Несколько функций поддерживают дополнительный, второй параметр сдвиг\_времени. Этот параметр позволят ссылаться на данные из периода времени в прошлом. Например, для **avg(1h,1d)** будет возвращено среднее значение за час днем ранее.

Триггеры лишь оценивают информацию истории. Если история недоступна (особенно актуально для сдвига времени), информация динамики изменений не используется, таким образом история должна храниться по крайней мере на ожидаемый период функций триггеров.

Вы можете использовать поддерживаемые [единицы измерений](#) в выражениях триггеров, например, '5m' (минут) вместо '300' секунд или '1d' (день) вместо '86400' секунд. '1K' будет состоять из '1024' байт.

### 3 Операторы

Следующие операторы поддерживаются для триггеров (**представлены по убыванию приоритета выполнения**):

| ПРИОРИТЕТ | ОПЕРАТОР   | ОПРЕДЕЛЕНИЕ   |
|-----------|------------|---|
| 1         | -          | Унарный минус   |
| 2         | <b>not</b> | Логическое НЕ   |
| 3         | *          | Умножение   |
|           | /          | Деление   |
| 4         | +          | Арифметический плюс   |
|           | -          | Арифметический минус  |
| 5         | <          | Менее чем. Этот оператор может быть представлен в виде:<br>$A < B \Leftrightarrow (A \leq B - 0.000001)$                          |
|           | <=         | Менее чем или равно.  |
|           | >          | Более чем. Этот оператор может быть представлен в виде:<br>$A > B \Leftrightarrow (A \geq B + 0.000001)$                          |
|           | >=         | Более чем или равно.  |
| 6         | =          | Равенство. Этот оператор может быть представлен в виде:<br>$A = B \Leftrightarrow (A > B - 0.000001) \& (A < B + 0.000001)$       |
|           | <>         | Не равно. Этот оператор может быть представлен в виде:<br>$A <> B \Leftrightarrow (A \leq B - 0.000001) \& (A \geq B + 0.000001)$ |
| 7         | <b>and</b> | Логическое И  |
| 8         | <b>or</b>  | Логическое ИЛИ  |

Операторы **not**, **and** and **or** регистрозависимы и должны быть в нижнем регистре. Они также должны быть окружены символами пробелов или круглыми скобками.

Все операторы, кроме унарных - и **not**, имеют ассоциативность слева на право. Унарные операторы - и **not** не ассоциативные (имеется в виду, что необходимо использовать **-(-1)** и **not (not 1)** вместо **--1** и **not not 1**).

Результат вычисления:

- <, <=, >, >=, =, <> операторы должны выдавать '1' в выражении триггера, если указанное отношение правдиво и '0', если оно является ложью;
- **and** должен выдавать '1', если оба его операнда в сравнении не равны '0'; в противном случае, будет выдан '0';
- **or** должен выдавать '1', если какой-либо из его операндов в сравнении не равен '0'; в противном случае, будет выдан '0';
- Результат логического оператора отрицания **not** равен '0', если значение этого операнда в сравнении не равно '0'; '1', если значение этого операнда в сравнении равно '0'.

### 4 Примеры триггеров

#### Пример 1

Высокая загрузка процессора на [www.zabbix.com](http://www.zabbix.com).

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

'www.zabbix.com:system.cpu.load[all,avg1]' представляет короткое имя наблюдаемого параметра. Эта строка указывает, что сервер - 'www.zabbix.com' и наблюдаемый ключ - 'system.cpu.load[all,avg1]'. Используя функцию 'last()', мы ссылаемся на самое последнее значение. И наконец '>5' означает, что триггер перейдет в состояние ПРОБЛЕМА всякий раз, когда самое новое измерение загрузки процессора на сервере [www.zabbix.com](http://www.zabbix.com) будет превышать 5.

#### Пример 2

[www.zabbix.com](http://www.zabbix.com) перегружен

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

Это выражение будет истинным, когда либо текущая загрузка процессора станет более 5, либо загрузка процессора больше значения 2 за последние 10 минут.

#### Пример 3

/etc/passwd был изменен

Используется функция diff:

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

Это выражение будет истинным, когда предыдущее значение контрольной суммы файла /etc/passwd отличается от самого нового значения.

Аналогичные выражения могут быть полезны для мониторинга изменений в важных файлах, таких как /etc/passwd, /etc/inetd.conf, /kernel и других.

#### Пример 4

Кто-то скачивает большой файл из Интернет

Используется функция min:

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

Это выражение будет истинным, когда количество полученных байт на eth0 превышает 100 КБ за последних 5 минут.

### Пример 5

Оба узла кластера SMTP серверов недоступны

Примечание, в выражении используются два разных узла сети:

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

Это выражение будет истинным, когда оба SMTP сервера недоступны на обоих smtp1.zabbix.com и smtp2.zabbix.com.

### Пример 6

Zabbix агент нуждается в обновлении

Используется функция str():

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

Это выражение будет истинным, когда версия Zabbix агента содержит в себе 'beta8' (возможно 1.0beta8).

### Пример 7

Сервер недоступен

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

Это выражение будет истинным, если узел сети "zabbix.zabbix.com" недоступен более 5 раз за последние 30 минут.

### Пример 8

Нет данных за последние 3 минуты

Используется функцию nodata():

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

'tick' должен иметь тип 'Zabbix trapper'. Для того, чтобы этот триггер заработал, элемент данных 'tick' должен существовать. Узел сети должен периодически отправлять данные этому элементу данных используя zabbix\_sender. Если не было получено данных за последние 180 секунд, значением триггера станет ПРОБЛЕМА.

**Пример 9**

Активность CPU в ночное время

Используется функция `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
```

Триггер может изменить свое состояние в истинное только в ночное время (00:00-06:00).

**Пример 10**

Проверка синхронизации времени на клиенте со временем на Zabbix сервере

Используется функция `fuzzytime()`:

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

Триггер изменит состояние на проблему тогда, когда локальное время на сервере MySQL\_DB и Zabbix сервере различаются более чем на 10 секунд.

**Пример 11**

Сравнение средней загрузки сегодня со средним значением загрузки за это же время вчера (использование второго параметра сдвиг\_времени).

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

Триггер изменит свое состояние на проблему, если средняя загрузка за последний час будет в два раза больше чем за аналогичный период времени вчера.

**Пример 12**

Использование значение другого элемента данных в качестве порогового значения триггера:

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

Триггер изменит свое состояние на проблему, если свободное пространство на диски упадет ниже 10 процентов.

**Пример 13**

Использование **результата вычисления** для получения количества триггеров за пределами порога:

```
{server1:system.cpu.load[all,avg1].last()}>5) +  
{server2:system.cpu.load[all,avg1].last()}>5) +  
{server3:system.cpu.load[all,avg1].last()}>5)>=2
```

Триггер изменит свое состояние на проблему, если по крайней мере два триггера в выражении превышают 5.

## 5 Гистерезис

Иногда триггер должен иметь различные условия для разных состояний. Например, мы хотим определить триггер, который перейдет в состояние ПРОБЛЕМА, если температура в серверной комнате поднимется выше 20C. При этом триггер должен оставаться в состоянии ПРОБЛЕМА, пока температура не опустится ниже 15C.

Для того, чтобы сделать это, мы определим следующий триггер:

### Пример 1

Температура в серверной комнате слишком высокая.

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

Обратите внимание, используется макрос {TRIGGER.VALUE}. Он возвращает текущее состояние триггера.

### Пример 2

Очень мало свободного места на диске

Проблема: если меньше 10ГБ за последние 5 минут

Восстановление: если больше 40ГБ за последние 10 минут

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

Обратите внимание, используется макрос {TRIGGER.VALUE}. Он возвращает текущее состояние триггера.

From:

<https://www.zabbix.com/documentation/2.4/> - **Zabbix Documentation 2.4**

Permanent link:

<https://www.zabbix.com/documentation/2.4/ru/manual/config/triggers/expression>

Last update: **2017/08/24 06:11**

