

8 预测触发功能

概述

有时候有即将到来问题的迹象。可以发现这些迹象，以便提前采取行动，以防止或至少最小化问题的影响。

Zabbix具有基于历史数据预测受监视系统的未来行为的工具。这些工具通过预测触发功能实现。

1 功能

需要知道的两件事是如何定义问题状态以及需要多少时间来采取行动。有两种方法可以设置一个关于潜在的不必要的情况的触发信号。第一：触发器必须在系统发生“时间作用”之后才会发生故障状态。第二：当系统在不到“时间行为”的时候达到问题状态时，触发器必须触发。使用相应的触发器功能是**forecast**和 **timeleft**。请注意，两个功能的基本统计分析基本相同。您可以设置触发器，以您喜欢的方式，以类似的结果。

2 参数

这两个功能使用几乎相同的参数集。列表请参见[supported functions](#) 支持的功能。

2.1 时间间隔

首先，你应该指定Zabbix应该分析的历史时期来进行预测。你可以通过“秒”或“#num”参数和可选的“time_shift”以熟悉的方式进行操作，就像使用**avg**、**count**、**delta**、**max**、**min**和**sum**功能。

2.2 预测范围

(forecast only)

参数time指定了将来Zabbix应该在多大程度上推断其在历史数据中找到的依赖关系。 无论是否使用“time_shift”“时间”始终从当前时刻算起。

2.3 阈值

(timeleft only)

参数“阈值”指定分析的项目必须达到的值，如果从上或下都没有差异。一旦我们确定了 $f(t)$ （见下文），我们就要解方程 $f(t) = \text{“阈值”}$ ，如果没有这样的根，返回更靠近现在和向右的根或9999999999.9999。

当监控项值接近阈值并超过它时，**timeleft**假定交叉点已经过去，因此切换到下一个“阈值”级别的交叉点（如果有的话）。最佳实践应该是使用预测作为普通问题诊断的补充，而不是替代。¹⁾

2.4 Fit函数

默认fit是线性函数。但是如果你的监控系统更复杂，你有可以有更多的选择。

fit	x = f(t)
线性	$x = a + b*t$
多项式 ²⁾	$x = a_0 + a_1*t + a_2*t^2 + \dots + a_n*t^n$
指数	$x = a*\exp(b*t)$
对数	$x = a + b*\log(t)$
幂	$x = a*t^b$

2.5 模式

(forecast only)

每次触发功能被评估时，它都会从指定的历史时段获得数据并将指定的函数拟合到数据。因此，如果数据略有不同，拟合函数将略有不同。对于某些“fit”选项（如多项式），未来的简单值可能会产生误导。

模式	预测结果
值	$f(now + time)$
最大	$\max_{now \leq t \leq now + time} f(t)$
最小	$\min_{now \leq t \leq now + time} f(t)$
增量	$max - min$
平均	$f(t)$ 的平均值 ($now \leq t \leq now + time$) 参考 定义

3 细节

为了避免大量的计算，我们考虑在指定周期内的第一个值的时间戳加上1ns作为一个新的零时间。（当前时间时期是 10^9 ，时期平方是 10^{18} ，双精度约为 10^{-16} ）。添加1 ns以提供对数和幂拟合的所有正时间值，其涉及计算 $\log[t]$ 时间偏移不影响线性、多项式、指数（除了更容易和更精确的计算），但改变对数和幂函数的状态。

4 潜在错误

函数如下情况下返回-1：

- 指定的评估期不包含数据；
- 数学运算结果未定义³⁾；
- 数值问题（不幸的是，对于一些输入数据范围和双精度浮点格式的精度变得不足⁴⁾。

如果选择合适不好描述提供的数据或只有太少的数据用于精确预测，就不会有警告或错误被标记。

5 示例和错误处理

要在主机上的可用磁盘空间用完时收到警告，可以使用如下触发器表达式：

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

然而，错误代码-1可能会发挥作用，并将您的触发器置于异常状态。一般来说，这是很好的，因为你收到一个警告，你的预测不能正常工作，你应该更深入地了解它们，找出原因。但有时它是坏的，因为-1可以

简单地意味着没有关于最后一小时内获得的主机可用磁盘空间的数据。如果您收到太多错误警报，则应考虑使用更复杂的触发器表达式⁵⁾：

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

形势比**预测**有点困难。首先，-1可能会也可能不会将触发器置于问题状态，具体取决于您是否具有表达式：或者像

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

此外，如果项目值为负值，-1可能是有效的预测。但这种情况实际发生的可能性很小，（参见运算符=**如何**工作）。因此添加

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

如果你想或不想把1作为一个问题来对待。

参阅

1. [Predictive trigger functions \(pdf\) on zabbix.org](#)

1)

例如，一个简单的触发器

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

当监控项值接近X时可能进入异常状态，然后一旦达到值X就突然恢复。如果异常是监控项值低于X[]请使用：

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

如果异常是项目值高于X[]请使用：

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

2)

多项式度可以从1到6，**多项式1**等于**线性**。然而，谨慎使用更高阶多项式**with caution**。如果评估周期包含比确定多项式系数所需的更少的点数，则多项式度将降低。（例如请求**多项式5**，但只有4点，因此**多项式3**更合适）。

3)

比如将**指数**或者**幂函数**计入log()监控项值. 如果数据包含零或负数，您将收到错误，因为log()仅限于正值。

4)

对于**线性**[]**指数**[]**对数**和**幂**适合所有必要的计算都可以明确地写出来。对于**多项式**，只有在没有任何附加步骤的情况下才能计算出值。计算**avg**涉及计算多项式反导数（解析）。计算**最大**[]**最小**和**增量**涉及计算多项式导数（解析），并找到其根源（数字）。求解f(t)[]0涉及求多项式根（数值）。

5)

但是在这种情况下，1可能导致触发器从问题状态恢复。充分保护使用：

```
{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
```

From:

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

Permanent link:

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

Last update: **2018/09/26 06:23**

