

7 Fonctions prédictives de déclenchement

Aperçu

Parfois, il y a des signes du problème à venir. Ces signes peuvent être repérés afin que des actions puissent être prises à l'avance pour prévenir ou au moins minimiser l'impact du problème.

Zabbix dispose d'outils permettant de prédire le comportement futur du système supervisé sur la base de données historiques. Ces outils sont exécutées via des fonctions prédictives de déclenchement.

1 Fonctions

Il faut savoir comment définir un problème et combien de temps il faut pour agir. Ensuite, il existe deux manières de configurer un déclencheur signalant une situation potentiellement indésirable. Premièrement : le déclencheur doit se déclencher lorsque le système après "temps pour agir" est censé être en état de problème. Deuxièmement : le déclencheur doit être déclenché lorsque le système va atteindre l'état problématique en moins de "temps pour agir". Les fonctions de déclenchement correspondantes à utiliser sont **forecast** et **timeleft**. Notez que l'analyse statistique sous-jacente est fondamentalement identique pour les deux fonctions. Vous pouvez configurer un déclencheur de la manière que vous préférez avec des résultats similaires.

2 Paramètres

Les deux fonctions utilisent presque le même jeu de paramètres. Utilisez la liste des [fonctions supportées](#) comme référence.

2.1 Intervalle de temps

Tout d'abord, vous devez spécifier la période historique que Zabbix doit analyser pour établir une prédiction. Vous le faites de manière familière au moyen du paramètre `sec` ou `#num` et de l'option facultative `time_shift` comme vous le faites avec les fonctions **avg**, **count**, **delta**, **max**, **min** et **sum**.

2.2 Horizon de prévision

(**forecast** seulement)

Le paramètre `time` spécifie combien de temps dans le futur Zabbix doit extrapoler les dépendances trouvées dans les données historiques. Peu importe si vous utilisez `time_shift` ou non, `time` est toujours compté à partir du moment présent.

2.3 Seuil à atteindre

(timeleft seulement)

Le paramètre `threshold` spécifie une valeur que l'élément analysé doit atteindre, aucune différence si elle est au dessus ou en dessous. Après avoir déterminé $f(t)$ (voir ci-dessous), nous devons résoudre l'équation $f(t) = \text{threshold}$ et renvoyer la racine la plus proche du moment actuel, situé à droite du moment actuel, ou renvoyer 9999999999.9999 s'il n'existe pas de telles racines.

Lorsque les valeurs d'éléments s'approchent du seuil puis le dépassent, **timeleft** suppose que l'intersection est déjà passée et passe donc à la prochaine intersection avec le niveau de `threshold`, le cas échéant. La meilleure pratique consiste à utiliser les prévisions en complément des diagnostics de problèmes ordinaires, et non en tant que substitution.¹⁾

2.4 Fonctions d'ajustement

Par défaut, `fit` est une fonction *linéaire*. Mais si votre système supervisé est plus compliqué, vous aurez plus d'options à choisir.

fit	x = f(t)
<i>linéaire</i>	$x = a + b*t$
<i>polynôme</i> ²⁾	$x = a_0 + a_1*t + a_2*t^2 + \dots + a_n*t^n$
<i>exponentiel</i>	$x = a*\exp(b*t)$
<i>logarithmique</i>	$x = a + b*\log(t)$
<i>puissance</i>	$x = a*t^b$

2.5 Modes

(forecast seulement)

Chaque fois qu'une fonction de déclenchement est évaluée, elle obtient des données de la période d'historique spécifiée et adapte une fonction spécifiée aux données. Donc, si les données sont légèrement différentes, la fonction ajustée sera légèrement différente. Si nous calculons simplement la valeur de la fonction ajustée à une heure future spécifiée, vous ne saurez rien sur le comportement attendu de l'élément analysé d'ici à ce moment-là.

Pour certaines options de `fit` (comme *polynôme*), une simple valeur future peut être trompeuse.

mode	résultat prévisionnel
<i>value</i>	$f(\text{maintenant} + \text{time})$
<i>max</i>	$\max_{\text{maintenant} \leq t \leq \text{maintenant} + \text{time}} f(t)$
<i>min</i>	$\min_{\text{maintenant} \leq t \leq \text{maintenant} + \text{time}} f(t)$
<i>delta</i>	$\text{max} - \text{min}$
<i>avg</i>	moyenne de $f(t)$ ($\text{maintenant} \leq t \leq \text{maintenant} + \text{time}$) selon la définition

3 Détails

Pour éviter les calculs avec de grands nombres, considérons l'horodatage de la première valeur de la période spécifiée plus 1 ns comme une nouvelle heure nulle (l'heure actuelle est de l'ordre de 10^9 , l'époque au carré est de 10^{18} , la double précision est d'environ 10^{-16}). 1 ns est ajoutée pour fournir toutes les valeurs de temps positives pour les ajustements *logarithmiques* et les ajustements de *puissance* impliquant le calcul de $\log(t)$. Le décalage temporel n'affecte pas les calculs *linéaires*,

polynomiaux, exponentiels (à l'exception des calculs plus faciles et plus précis), mais modifie la forme des fonctions *logarithmiques* et de *puissance*.

4 Erreurs potentielles

Les fonctions retournent -1 dans de telles situations :

- la période d'évaluation spécifiée ne contient aucune donnée ;
- le résultat de l'opération mathématique n'est pas défini ³⁾ ;
- complications numériques (malheureusement, pour certains ensembles de données en entrée, la double précision et la virgule flottante sont insuffisantes) ⁴⁾.

Aucun avertissement ni aucune erreur ne sont signalés si l'ajustement choisi décrit mal les données fournies ou s'il y a trop peu de données pour une prédiction précise.

5 Exemples et traitement des erreurs

Pour recevoir un avertissement lorsque vous êtes sur le point de manquer d'espace disque disponible sur votre hôte, vous pouvez utiliser une expression de déclencheur comme celle-ci :

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

Cependant, le code d'erreur -1 peut survenir et placer votre déclencheur dans un état problème. Généralement, c'est bien parce que vous recevez un avertissement que vos prédictions ne fonctionnent pas correctement et que vous devriez les examiner de plus près pour savoir pourquoi. Mais parfois, c'est mauvais parce que -1 peut simplement signifier qu'il n'y avait aucune donnée obtenue concernant l'espace disque disponible de l'hôte au cours de la dernière heure. Si trop de faux positifs se produisent, envisagez d'utiliser une expression de déclencheur plus complexe ⁵⁾ :

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

La situation est un peu plus difficile avec **forecast**. Tout d'abord, -1 peut ou non mettre le déclencheur dans un état problème en fonction de votre expression, telle que :

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

ou

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

De plus, -1 peut être une prévision valide s'il est normal que la valeur de l'élément soit négative. Mais la probabilité de cette situation dans la situation réelle est négligeable (voir [comment fonctionne l'opérateur =](#)). Ajoutez donc

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

ou

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

si vous voulez ou ne voulez pas traiter -1 comme un problème, respectivement.

Voir aussi

1. [Fonctions prédictives de déclenchement \(pdf\)](#) sur zabbix.org

1)

Par exemple, un déclencheur simple comme

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

peut passer à l'état problème lorsque la valeur de l'élément approche de X, puis se rétablir soudainement une fois que la valeur X est atteinte. Si le problème est que la valeur de l'élément est inférieure à X, utilisez :

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

Si le problème est que la valeur de l'élément est supérieure à X, utilisez :

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

2)

Le degré de polynôme peut aller de 1 à 6, *polynome1* est équivalent à *linéaire*. Cependant, utilisez les polynômes de degré supérieur [avec prudence](#). Si la période d'évaluation contient moins de points que nécessaire pour déterminer les coefficients polynomiaux, le degré polynomial est abaissé (par exemple, le *polynôme5* est demandé, mais il n'y a que 4 points, donc le *polynôme 3* sera ajusté).

3)

par exemple, l'ajustement des *fonctions exponentielles* ou des *puissances* implique le calcul de $\log()$ des valeurs d'élément. Si les données contiennent des zéros ou des nombres négatifs, vous obtiendrez une erreur, car $\log()$ est défini uniquement pour les valeurs positives.

4)

Pour *linéaire*, *exponentiel*, *logarithmique* et *puissance* l'ajustement de tous les calculs nécessaires peuvent être écrits explicitement. Pour *polynôme* seule la *valeur* peut être calculée sans étape supplémentaire. Le calcul *avg* implique le calcul d'un antiderivatif polynomial (analytique). Le calcul de *max*, *min* et *delta* implique le calcul analytique de la dérivée polynomiale et la recherche de ses racines (numérique). La résolution de $f(t) = 0$ implique la recherche de racines polynomiales (numérique).

5)

dans ce cas, -1 peut entraîner la récupération de votre déclencheur à partir de l'état problème. Pour être entièrement protégé, utilisez :

```
{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/fr/manual/config/triggers/prediction>

Last update: **2019/03/24 16:02**

