



ZABBIX 2020
Conference
CHINA

演讲主题

主从监控项经验分享

演讲嘉宾

伍昕 宏时数据

01

主从监控项简介

主从监控项

Dependent Items

Zabbix的一个使用场景

场景：

Zabbix的一次数据采集可能获取多个值，（比如与第三方系统对接，获取数据库所有状态）

`show global variables;`

```
| Handler_update | 0 |
| Handler_write | 414 |
| InnoDB_buffer_pool_dump_status | Dumping of buffer pool not started |
| InnoDB_buffer_pool_load_status | Buffer pool(s) load completed at 170531 10:45:37 |
| InnoDB_buffer_pool_resize_status | |
| InnoDB_buffer_pool_pages_data | 513 |
| InnoDB_buffer_pool_bytes_data | 8404992 |
| InnoDB_buffer_pool_pages_dirty | 0 |
| InnoDB_buffer_pool_bytes_dirty | 0 |
| InnoDB_buffer_pool_pages_flushed | 37 |
| InnoDB_buffer_pool_pages_free | 7676 |
| InnoDB_buffer_pool_pages_misc | 2 |
| InnoDB_buffer_pool_pages_total | 8191 |
| InnoDB_buffer_pool_read_ahead_rnd | 0 |
| InnoDB_buffer_pool_read_ahead | 0 |
| InnoDB_buffer_pool_read_ahead_evicted | 0 |
| InnoDB_buffer_pool_read_requests | 2535 |
| InnoDB_buffer_pool_reads | 479 |
| InnoDB_buffer_pool_wait_free | 0 |
| InnoDB_buffer_pool_write_requests | 515 |
| InnoDB_data_fsyncs | 7 |
| InnoDB_data_pending_fsyncs | 0 |
| InnoDB_data_pending_reads | 0 |
| InnoDB_data_pending_writes | 0 |
| InnoDB_data_read | 7918080 |
| InnoDB_data_reads | 505 |
| InnoDB_data_writes | 54 |
| InnoDB_data_written | 641024 |
```



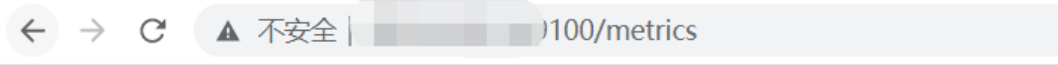
Zabbix server



主从监控项

Dependent Items

http://exporter_ip:9100/metrics



```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.4285e-05
go_gc_duration_seconds{quantile="0.25"} 4.4334e-05
go_gc_duration_seconds{quantile="0.5"} 5.7874e-05
go_gc_duration_seconds{quantile="0.75"} 8.7763e-05
go_gc_duration_seconds{quantile="1"} 0.003743636
go_gc_duration_seconds_sum 28.494698484
go_gc_duration_seconds_count 240076
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.11.2"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.257816e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
```



Zabbix server

主从监控项

Dependent Items

主从监控项

Zabbix的一个使用场景

将Zabbix的一次数据采集获取到的多个值，分别赋值给各个从属监控项

2步操作

2个Zabbix的功能

第一步：

(相关项目) 监控项
DEPENDENT ITEM

第二步：

数据预处理
PREPROCESSING

主监控项准备

Dependent Items

创建一个自定义监控项：

修改配置agent配置文件：`/home/zabbix/zabbix_agent/conf/zabbix_agentd.conf`

`UserParameter=mysql_status,mysqladmin -uzabbix -p"Workshop123#" extended-status`

重启agent

`/home/zabbix/zabbix_agent/start_zabbix.sh restart`

* Name

Type

* Key

* Host interface

Type of information

* Update interval

Timestamp	Value
09/23/2020 02:34:57 PM	+-----+-----+ Variable_name Value +-----+-----+ Aborted_clients 52 Aborted_connects 328 Access_denied_errors 6 Aria_pagecache_blocks_not_flushed 0 Aria_pagecache_blocks_unused 15737 Aria_pagecache_blocks_used 24 Aria_pagecache_read_requests 27941679 Aria_pagecache_reads 100114 Aria_pagecache_write_requests 282531 Aria_pagecache_writes 0

主从监控项

Dependent Items

第一步：配置一个（相关项目）监控项

Item Properties → Type → Dependent item → Master item → Done
创建监控项 → 监控项类型 → 依赖监控项 → 选择“主监控项” → 配置结束

The screenshot shows the Zabbix Item configuration page for a 'Preprocessing' item. The 'Type' dropdown menu is open, displaying a list of item types. The 'Dependent item' option is highlighted at the bottom of the list. Other visible fields include 'Name', 'Key', 'Host interface', 'Type of information', 'Units', 'Update interval', and 'Custom intervals'.

The screenshot shows the Zabbix Item configuration page for a 'Preprocessing' item. The 'Type' dropdown is set to 'Dependent item'. The 'Master item' field is empty, and the 'Select' button next to it is highlighted with a red box. Other visible fields include 'Name', 'Key', 'Type of information', and 'Units'.

主从监控项

Dependent Items

第一步：配置一个依赖监控项

Item Properties → Type → Dependent item → Master item → Done
创建监控项 → 监控项类型 → 依赖监控项 → 选择“主监控项” → 配置结束

Name	Key	Type	Type of information	Status
Aborted_clients	Aborted_clients	Dependent item	Numeric (unsigned)	Enabled
Prometheus_Master_items	Prometheus_metrec_get	HTTP agent	Text	Enabled
数据库状态	mysql_status	Zabbix agent	Text	Enabled

* Name

Type

* Key

* Master item

Type of information

Units

主从监控项

Dependent Items

第一步：配置一个依赖监控项

主监控项：Master Items

Item Properties → Type → Dependent item → Master item → Done

相关项目监控项：DEPENDENT ITEMS

配置完成后，主监控项将会自动将自己的监控数据传给每一个依赖监控项

主从监控项

Dependent Items

第二步：数据预处理

功能：在入库前，对接收到的数据，按照配置的规则做转换处理后入库。



官方文档地址：

[https://www.zabbix.com/documentation/current/manual/config/items/preprocessing/preprocessing_details?s\[\]=preprocessing](https://www.zabbix.com/documentation/current/manual/config/items/preprocessing/preprocessing_details?s[]=preprocessing)

主从监控项

Dependent Items

第二步：配置一个数据预处理

Item Properties → Preprocessing → Preprocessing steps → Done
监控项配置页面 → 数据预处理 → 配置对应的预处理方式 → 配置结束

The screenshot shows the Zabbix configuration interface for a monitoring item. The 'Preprocessing' tab is active. Under 'Preprocessing steps', a table is visible with the following columns: Name, Parameters, Custom on fail, and Actions. The first step is configured with the name 'Regular expression', parameters 'Aborted_clients *|(.*?)|', and a value of '1'. A dropdown menu is open over the 'Name' column, showing a list of preprocessing methods. The 'Regular expression' method is currently selected and highlighted in blue. Other methods include Text, Trim, Right trim, Left trim, Structured data (XML XPath, JSONPath), Arithmetic (Custom multiplier), Change (Simple change, Change per second), Numeral systems (Boolean to decimal, Octal to decimal, Hexadecimal to decimal), Custom scripts (JavaScript), and Validation.

Name	Parameters	Custom on fail	Actions
1: Regular expression	Aborted_clients * (.*?)	<input type="checkbox"/>	Test Remove

Buttons: [Add](#), [Up](#), [Down](#), [History and trends](#), [Delete](#), [Cancel](#), [Test all steps](#)

主从监控项

Dependent Items

第二步：配置一个数据预处理

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	Regular expression	Aborted_clients * (.*?)	<input type="checkbox"/>	Test Remove

[Add](#) [Test all steps](#)

- other - (2 Items)

Aborted_clients	09/20/2020 07:04:57 PM	52
数据库状态	09/20/2020 07:04:57 PM	+-----

Timestamp	Value
09/20/2020 07:04:57 PM	+-----+ Variable_name Value +-----+ Aborted_clients 52 Aborted_connects 290 Access_denied_errors 6 Aria_pagecache_blocks_not_flushed 0 Aria_pagecache_blocks_unused 15737 Aria_pagecache_blocks_used 24

主从监控项

Dependent Items

第二步：配置一个数据预处理

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	Regular expression	Aborted_clients * (.*?)	<input type="checkbox"/>	Test Remove
				Test all steps

[Add](#)

常用的正则表达式预处理规则：

Pattern: 关键字 (.*) 关键字 (.*) 关键字

Output: \N 抓取上述第N个括号里面的内容

输入数据：

```
NAME ZHANG3 AGE 28 SALARY 10000
NAME LI4 AGE 22 SALARY 20000
NAME ZHAO5 AGE 30 SALARY 25000
```

Pattern: ZHANG3 AGE (.*) SALARY (.*)

Output: \2

测试结果: 10000

03

现场练习



现场练习

做一个Prometheus的主从监控项

← → ↻ ⚠ 不安全 | ██████████ 100/metrics

```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.4285e-05
go_gc_duration_seconds{quantile="0.25"} 4.4334e-05
go_gc_duration_seconds{quantile="0.5"} 5.7874e-05
go_gc_duration_seconds{quantile="0.75"} 8.7763e-05
go_gc_duration_seconds{quantile="1"} 0.003743636
go_gc_duration_seconds_sum 28.494698484
go_gc_duration_seconds_count 240076
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.11.2"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.257816e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
```



Zabbix server

现场练习

Node-export 确认

1.登录测试机器

2.启动export 命令： /home/zabbix/node_exporter/node_exporter &

3. Curl <http://127.0.0.1:9100/metrics>

```
[root@iZ8vblu7pa78d410d8yopvZ ~]# curl http://127.0.0.1:9100/metrics
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.4321e-05
go_gc_duration_seconds{quantile="0.25"} 4.192e-05
go_gc_duration_seconds{quantile="0.5"} 5.8568e-05
go_gc_duration_seconds{quantile="0.75"} 8.8028e-05
go_gc_duration_seconds{quantile="1"} 0.003695831
go_gc_duration_seconds_sum 29.074967638
go_gc_duration_seconds_count 245704
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
```


现场练习

监控项类型 : http

url : <http://127.0.0.1:9100/metrics>

数据类型 : 文本/text

* Name

Type

* Key

* URL

Query fields

Name	Value
<input type="text" value="name"/>	<input type="text" value="value"/>

[Add](#) [Remove](#)

Request type

Timeout

现场练习

数据效果

Timestamp	Value
09/24/2020 02:00:04 PM	<pre># HELP go_gc_duration_seconds A summary of the GC invocation durations. # TYPE go_gc_duration_seconds summary go_gc_duration_seconds{quantile="0"} 2.1868e-05 go_gc_duration_seconds{quantile="0.25"} 4.192e-05 go_gc_duration_seconds{quantile="0.5"} 5.8422e-05 go_gc_duration_seconds{quantile="0.75"} 8.5638e-05 go_gc_duration_seconds{quantile="1"} 0.003695831 go_gc_duration_seconds_sum 29.076426714 go_gc_duration_seconds_count 245723 # HELP go_goroutines Number of goroutines that currently exist. # TYPE go_goroutines gauge go_goroutines 7 # HELP go_info Information about the Go environment. # TYPE go_info gauge go_info{version="go1.11.2"} 1 # HELP go_memstats_alloc_bytes Number of bytes allocated and still in use. # TYPE go_memstats_alloc_bytes gauge go_memstats_alloc_bytes 2.862e+06 # HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed. # TYPE go_memstats_alloc_bytes_total counter go_memstats_alloc_bytes_total 6.95714315776e+11 # HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table. # TYPE go_memstats_buck_hash_sys_bytes gauge</pre>

现场练习

预处理类型：Prometheus pattern

预处理语法：

```
node_filesystem_avail_bytes{device="/dev/vda1",fstype="ext4",mountpoint="/"}
```

Preprocessing steps	Name	Parameters
1:	Prometheus pattern	node_filesystem_avail_bytes{device="/dev/vda1",fstype="ext4",mountpoint="/"}

[Add](#)

[Update](#) [Clone](#) [Check now](#) [Clear history and trends](#) [Delete](#) [Cancel](#)

```
# TYPE node_filesystem_avail_bytes gauge
node_filesystem_avail_bytes{device="/dev/vda1",fstype="ext4",mountpoint="/"}
```

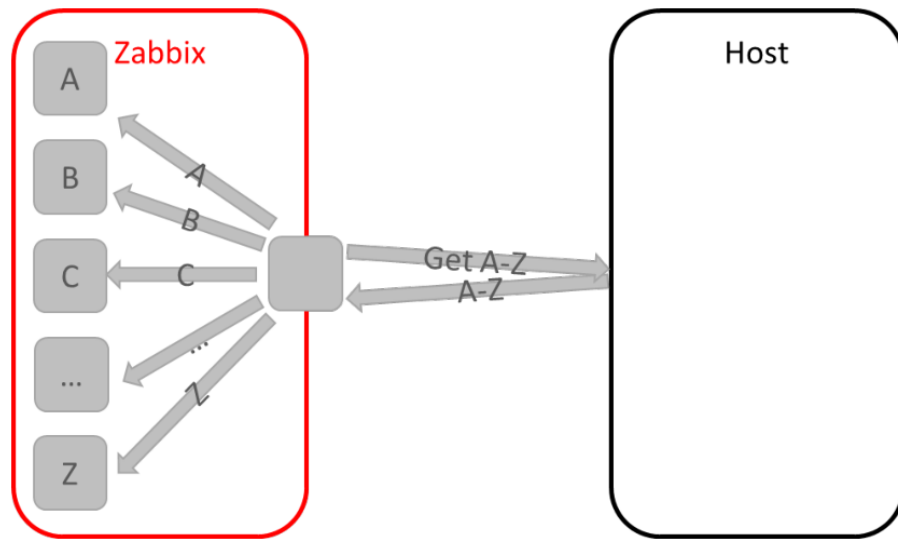
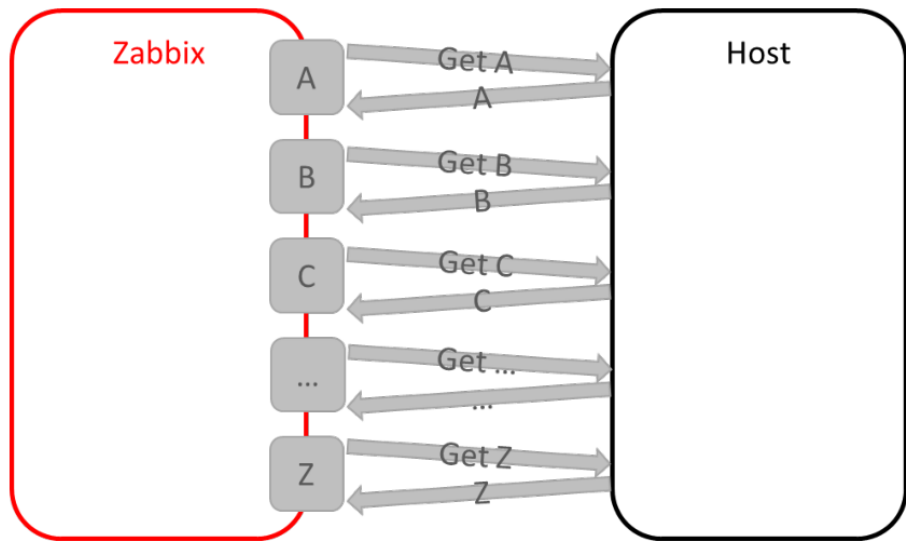
04

一些备注与提问环节

主从监控项

Dependent Items

优点：降低Zabbix对第三方的访问量

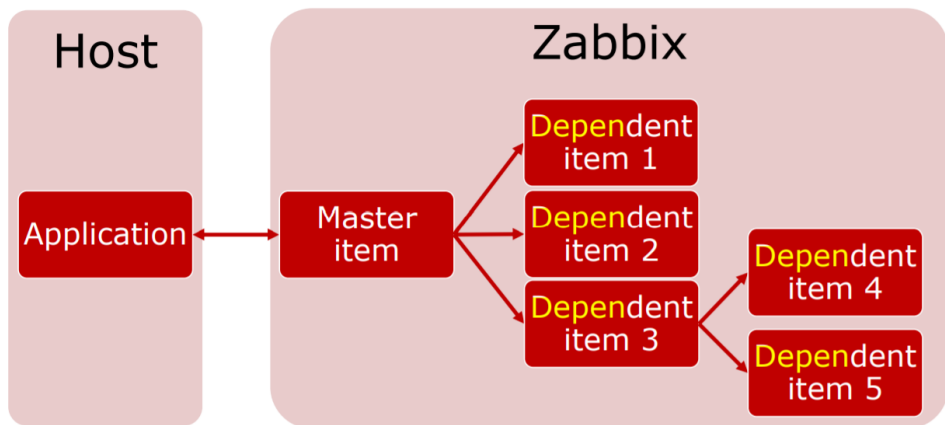


主从监控项

Dependent Items

依赖监控项 DEPENDENT ITEMS

- 依赖监控项的范围限于同一个主机/模板/自动发现规则内
- 一个主监控项最多支持999个依赖监控项
- 依赖监控项最多支持三层
- 如果主监控项在模板中而从属监控项创建在主机上，该从属监控项将无法通过模板导出。
- 主监控项历史数据保存时间可以设置为0



主从监控项

Dependent Items

数据预处理 PREPROCESSING

- 5.0版本前，所有预处理操作在Server端处理，5.0版本及以后预处理操作在Proxy端操作。
- 预处理的相关配置项：StartPreprocessors= 默认为3
- 4.4版本后，当预处理失败后，可以选择输出默认值，4.4版本前需要保证输入值正常，否则处理失败后监控项会变成不支持

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	Regular expression	ZHANG3 AGE (.*) SALARY (.*)	12	<input checked="" type="checkbox"/> Test Remove
Custom on fail	<input type="button" value="Discard value"/>	<input type="button" value="Set value to"/>	<input type="button" value="Set error to"/>	<input type="text" value="value"/>

问答时间

questions



ZABBIX

联系我们

Contact us

Zabbix 中国致力于为国内用户提供培训、咨询、以及其他的专业技术支持。也为国内的用户搭建交流学习的平台。



138-1772-0274



china@zabbix.com



www.grandage.cn
www.zabbix.com/cn



上海市徐汇区虹梅路1905号



Zabbix开源社区



Zabbix中国



Zabbix_China



Zabbix_team



Zabbix 开源社区



加入技术交流群

ZABBIX 2020
Conference
CHINA

THANK YOU 😊

