



Webinar

Preprocessing

all our microphones are muted

ask your questions in Q&A, not in the Chat

use Chat for discussion, networking or applause

1

Preprocessing

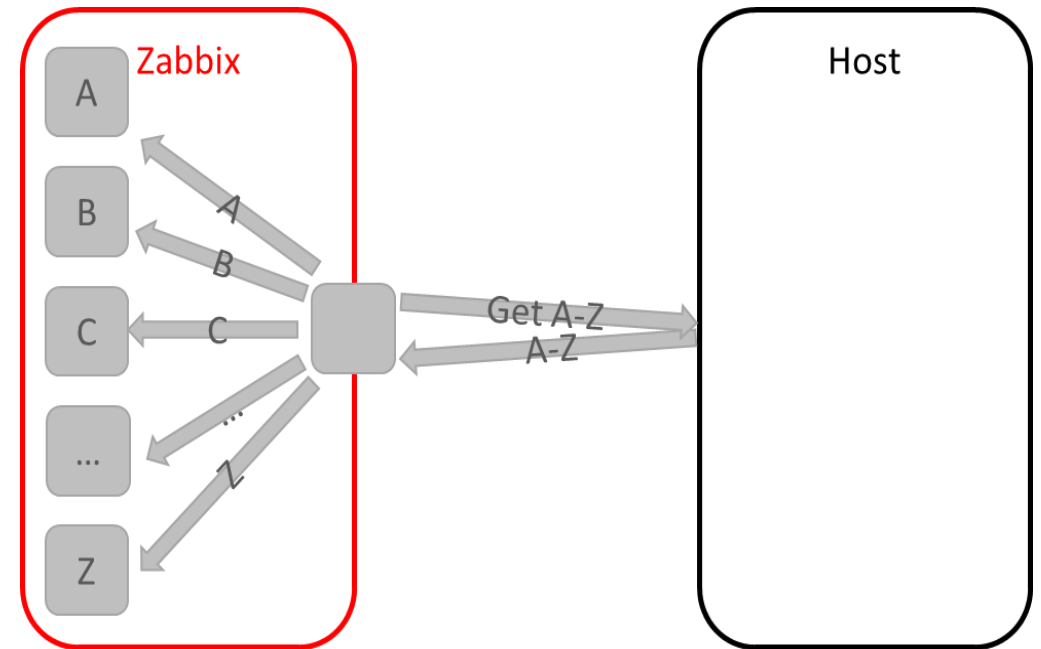
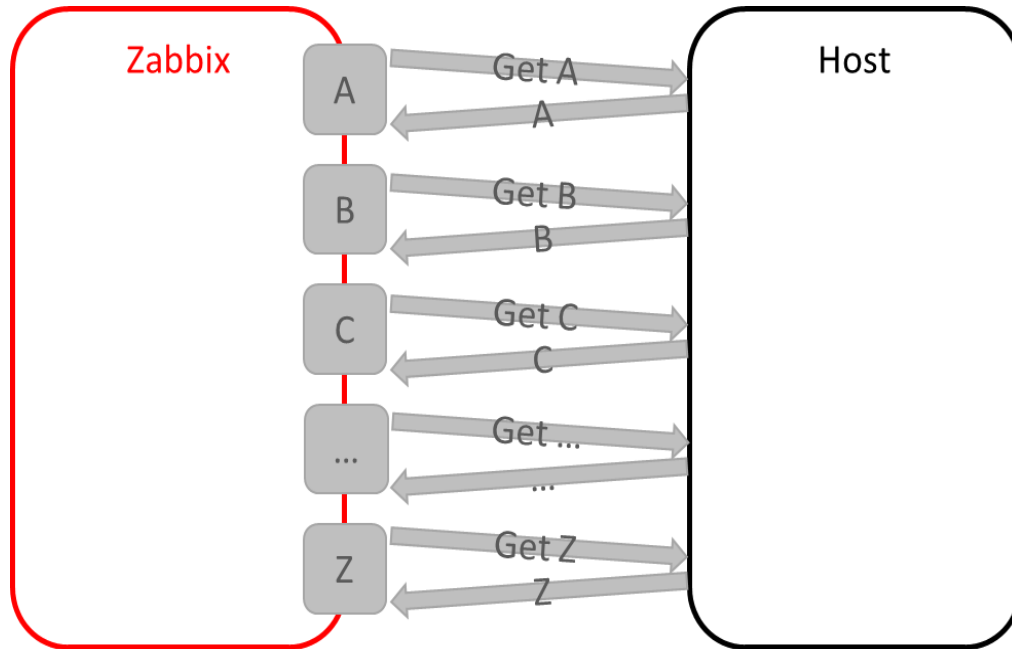


Preprocessing

VALUE PREPROCESSING

- ▶ Preprocessing of item values allows to define and execute data transformation rules for received item values.
- ▶ This can be used for simple as well as dependent items.

/usr/sbin/zabbix_server: preprocessing manager #1 [queued 41, processed 100 values, idle 5.001769 sec during 5.005980 sec]



VALUE PREPROCESSING

12 C



Trim



Temperature: 12

OFF (down)



Replace



Service: 0 (down)

{ "users":10004 }



JSON



Number of users: 10004

"GET /index.html HTTP/1.0"
200 28083



Regexp



Response code: 200
Size: 28083

Unstructured text



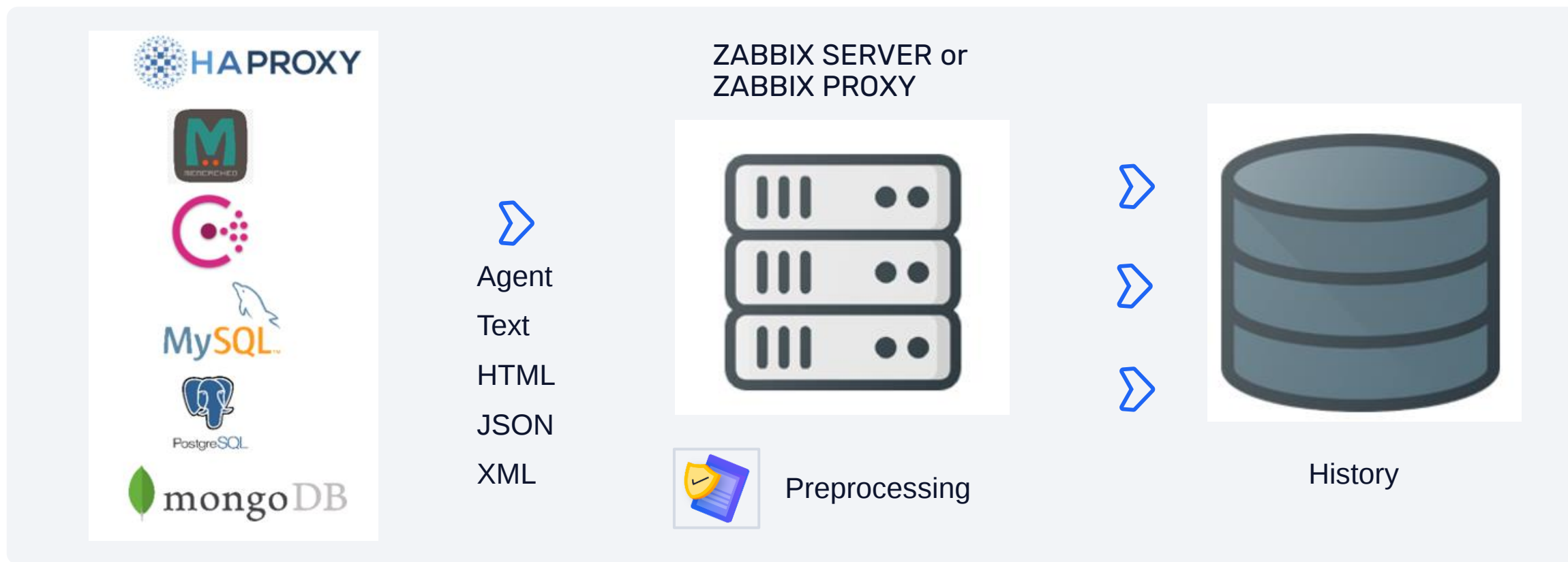
Regexp



Version: Apache 2.2.15 (Unix)
DNS lookup threads: 10

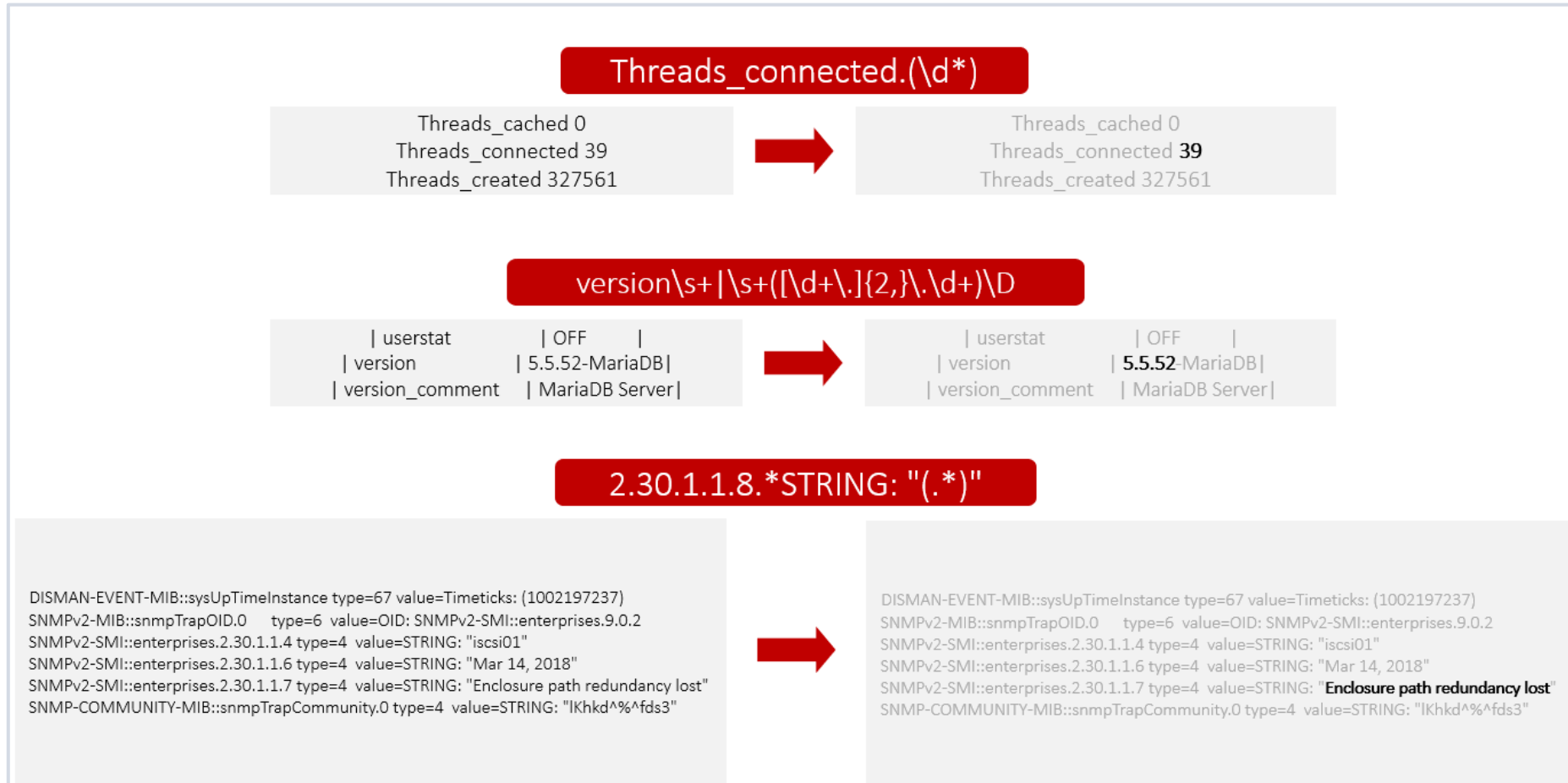
Preprocessing

DATA FLOW



Preprocessing

PCRE Examples



Preprocessing

JSON PATH

`$.store.book[0].price`

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```



```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```

Preprocessing

JSON PATH

`$.store.bicycle`

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```



```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```

Preprocessing

XML PATH

```
<?xml version="1.0" encoding="UTF-8"?>

<bookstore>

  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year> <price>30.00</price>
  </book>

  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>

  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>

</bookstore>
```



sum(/bookstore/book/price)

99.94



count(/bookstore/book)

3



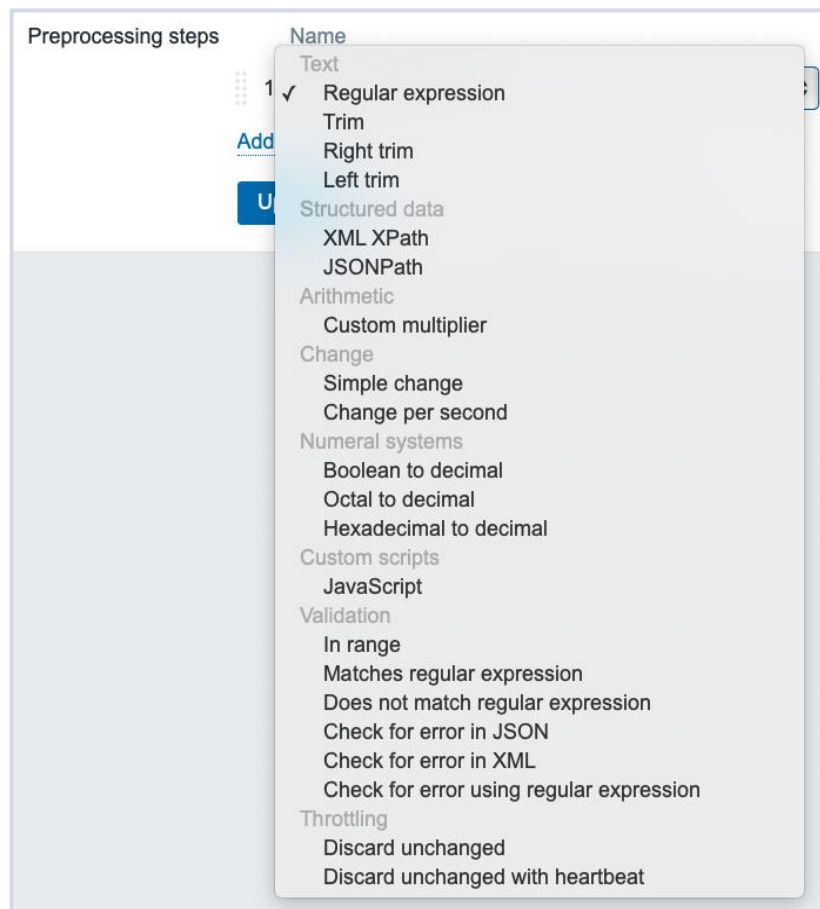
number(/bookstore/book[price<30]/price)

29.99

Preprocessing

THROTTLING

The throttling functionality allows to gather values from devices and applications at a very frequent interval, but the data is written to the database only when the incoming values are changed.



Zabbix simply ignores repeating values.

- › The repeating values are not saved as historical data and don't have an impact on triggers.

Options:

- › Discard a value if it has not changed.
- › (empty graphs and nodata() can cause false alarms)
- › Discard a value with a heartbeat.

Preprocessing

THROTTLING

Use cases

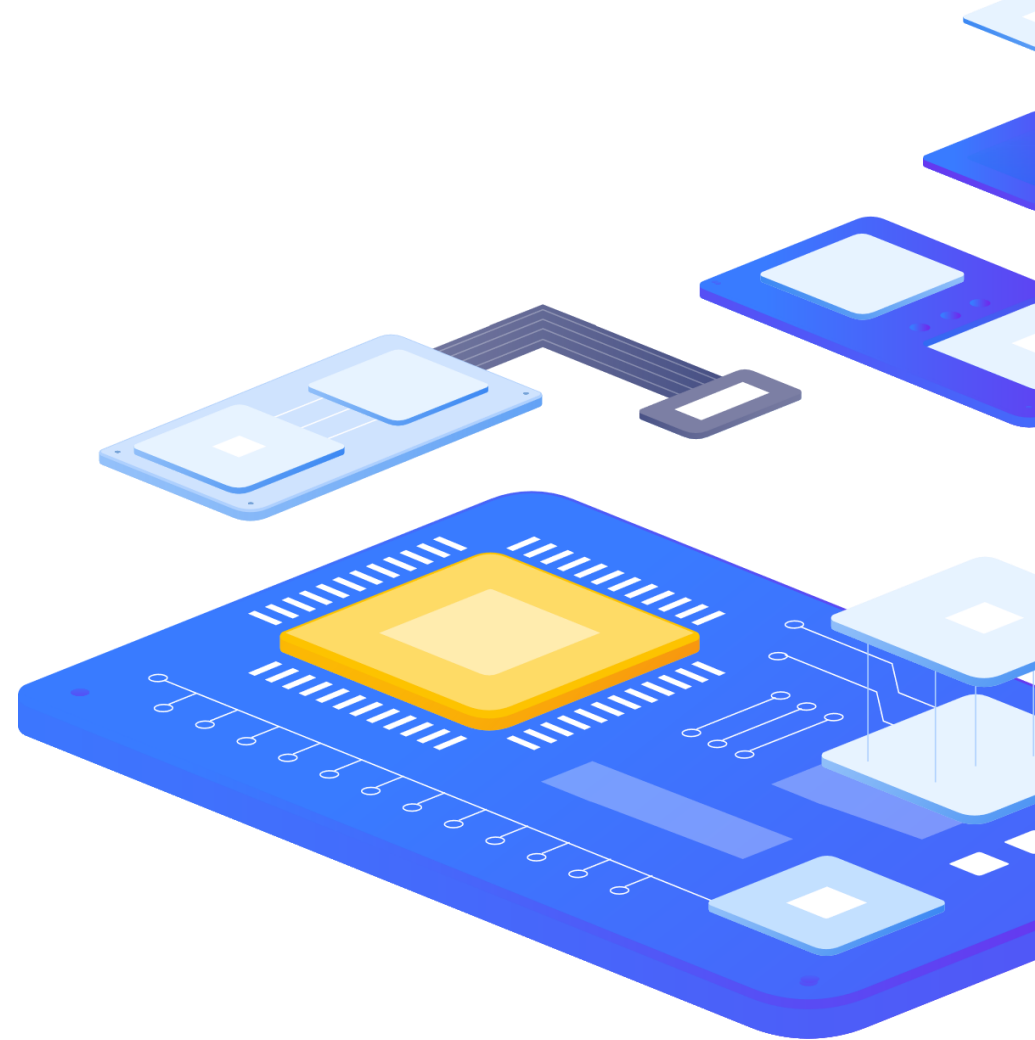
High frequency monitoring:

- › Saves space on the disk
- › Usable graphs

Monitoring of discrete data: statuses, states

Dependent metrics:

- › Master: 10s
- › Dependent metrics: 60s Throttling with a heartbeat



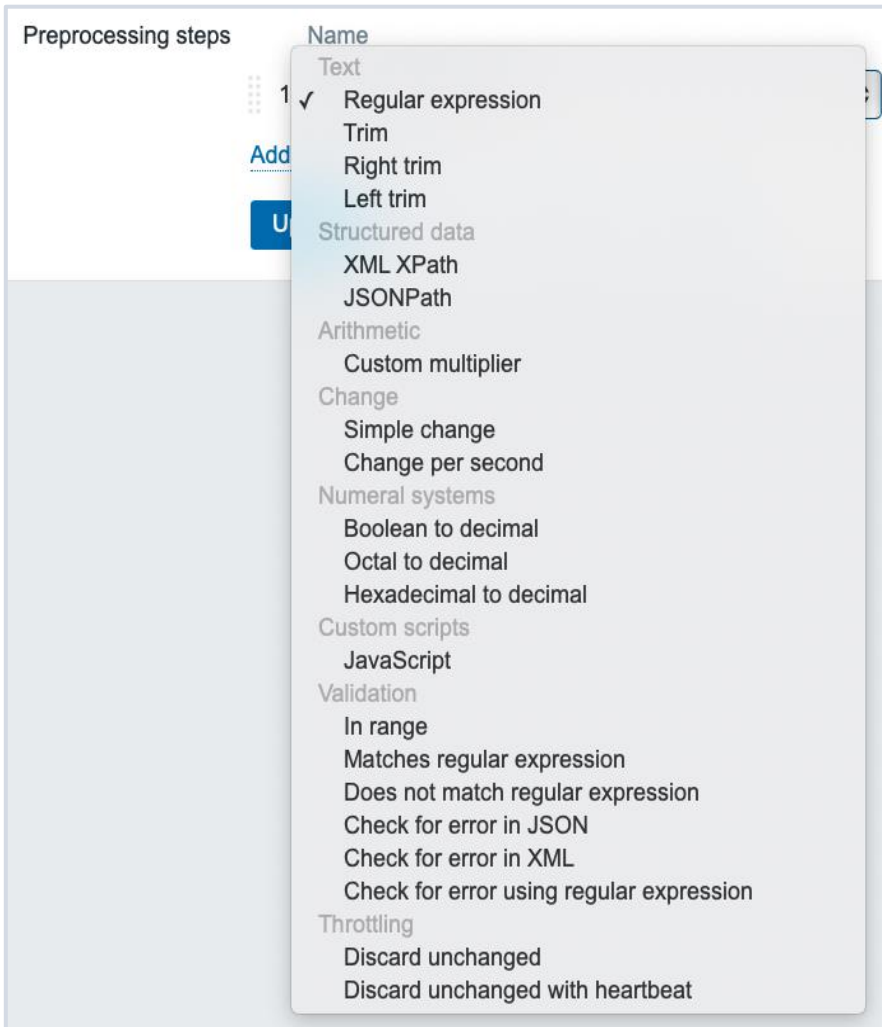
Preprocessing

THROTTLING Example

mm:ss	Values	Discard unchanged	Discard unchanged with heartbeat 30s	
00:00	0	0	0	
00:05	0			No value because same as previous
00:10	0			
00:15	1	1	1	Received different value
00:20	1			
00:25	1			
00:30	1			
00:35	1			
00:40	0	0	0	Received different value
00:45	0			
00:50	0			
00:55	0			
01:00	0			
01:05	0			
01:10	0		0	Value written because of heartbeat 30s
01:15	0			
01:20	1	1	1	Received different value
01:25	1			
01:30	0	0	0	Received different value
01:35	0			
01:40	0			
01:45	0			
01:50	0			
01:55	0			
02:00	0		0	Value written because of heartbeat 30s
02:05	0			

Preprocessing

DATA VALIDATION



Options:

- › In Range: A - B
- › Matches regular expression
- › Does not match regular expression
- › Check for error in JSON/XML
- › Check for error using regex

Preprocessing

ERROR PROCESSING

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Action
1:	In range	0 100	☐	Remove

- › Default error message

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Action
1:	In range	0 100	☒	Remove
Custom on fail Discard value Set value to Set error to				

- › Discard values outside of the defined range

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Action
1:	In range	0 100	☒	Remove
Custom on fail Discard value Set value to Set error to -1				

- › Set a custom value in case of an error

Item Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Action
1:	In range	0 100	☒	Remove
Custom on fail Discard value Set value to Set error to Temperature must be in range of 0-100				

- › Custom error message on fail

Preprocessing

JAVASCRIPT

- ▶ Creation of custom Javascript processing scripts
- ▶ A simple example:

Item

Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	JavaScript	<code>return (value - 32) * 5 / 9</code>	☐	Test Remove

Add

Add

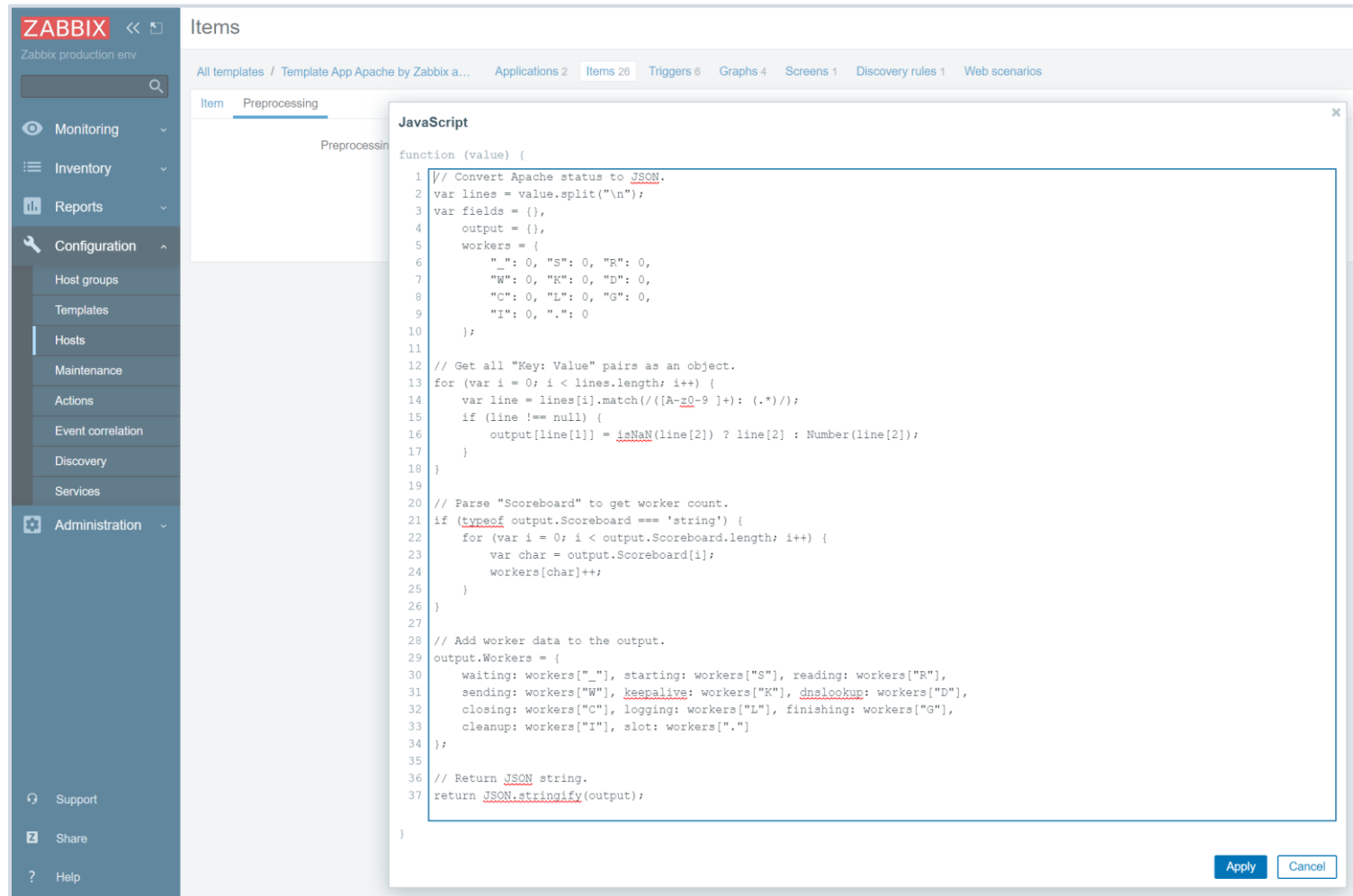
Cancel

- Name
- Text
- ✓ Regular expression
- 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
- Validation
- In range
- Matches regular expression
- Does not match regular expression
- Check for error in JSON
- Check for error in XML
- Check for error using regular expression
- Throttling
- Discard unchanged
- Discard unchanged with heartbeat

Preprocessing

JAVASCRIPT

- Transformation, aggregation, filters, arithmetic and logic operations and many more.



The screenshot shows the Zabbix web interface. On the left is a sidebar with navigation links: Monitoring, Inventory, Reports, Configuration (Host groups, Templates, Hosts, Maintenance, Actions, Event correlation, Discovery, Services), and Administration (Support, Share, Help). The main panel is titled 'Items' and shows a list of items with a 'Preprocessing' tab selected. A modal window titled 'JavaScript' is open, displaying a JavaScript function for preprocessing. The function takes a 'value' parameter and returns a JSON string. It converts Apache status to JSON, processes the scoreboard to get worker counts, and adds worker data to the output.

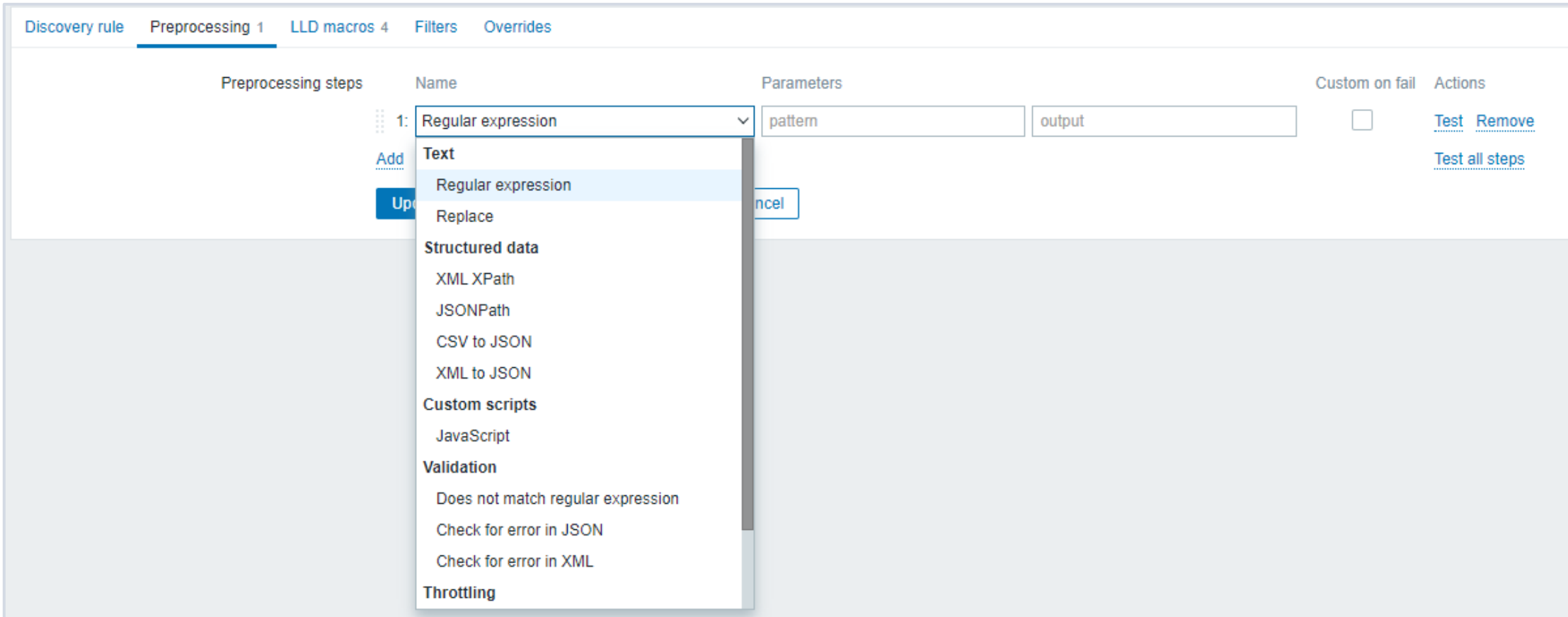
```
function (value) {  
  1 // Convert Apache status to JSON.  
  2 var lines = value.split("\n");  
  3 var fields = {},  
  4   output = {},  
  5   workers = {  
  6     "_": 0, "S": 0, "P": 0,  
  7     "W": 0, "K": 0, "D": 0,  
  8     "C": 0, "L": 0, "G": 0,  
  9     "I": 0, ".": 0  
 10  };  
 11  
 12 // Get all "Key: Value" pairs as an object.  
 13 for (var i = 0; i < lines.length; i++) {  
 14   var line = lines[i].match(/([A-Z0-9 ]+): (.*)/);  
 15   if (line !== null) {  
 16     output[line[1]] = isNaN(line[2]) ? line[2] : Number(line[2]);  
 17   }  
 18 }  
 19  
 20 // Parse "Scoreboard" to get worker count.  
 21 if (typeof output.Scoreboard === 'string') {  
 22   for (var i = 0; i < output.Scoreboard.length; i++) {  
 23     var char = output.Scoreboard[i];  
 24     workers[char]++;  
 25   }  
 26 }  
 27  
 28 // Add worker data to the output.  
 29 output.Workers = {  
 30   waiting: workers["_"], starting: workers["S"], reading: workers["R"],  
 31   sending: workers["W"], keepalive: workers["K"], dnslookup: workers["D"],  
 32   closing: workers["C"], logging: workers["L"], finishing: workers["G"],  
 33   cleanup: workers["I"], slot: workers["."] }  
 34 };  
 35  
 36 // Return JSON string.  
 37 return JSON.stringify(output);  
}
```

At the bottom right of the modal, there are 'Apply' and 'Cancel' buttons.

Preprocessing

LLD PREPROCESSING

- ▶ LLD supports preprocessing
- ▶ HTTP agent and JavaScript support enables the ability to perform the discovery without the need to use external scripts



The screenshot displays the 'Preprocessing 1' configuration tab within a software interface. The top navigation bar includes 'Discovery rule', 'Preprocessing 1', 'LLD macros 4', 'Filters', and 'Overrides'. The main area is titled 'Preprocessing steps' and contains a table with columns: 'Name', 'Parameters', 'Custom on fail', and 'Actions'.

Preprocessing steps	Name	Parameters	Custom on fail	Actions	
1:	Regular expression	pattern	output	☐	Test Remove

Below the table, there is an 'Add' button and a dropdown menu. The dropdown menu is open, showing a list of preprocessing options categorized into 'Text', 'Structured data', 'Custom scripts', 'Validation', and 'Throttling'. The 'Regular expression' option is currently selected.

- Text**
 - Regular expression
 - Replace
- Structured data**
 - XML XPath
 - JSONPath
 - CSV to JSON
 - XML to JSON
- Custom scripts**
 - JavaScript
- Validation**
 - Does not match regular expression
 - Check for error in JSON
 - Check for error in XML
- Throttling**

Preprocessing

PROMETHEUS

Storage

- Ceph exporter
- Ceph RADOSGW exporter
- Gluster exporter
- Hadoop HDFS FSImage exporter
- Lustre exporter
- ScaleIO exporter

HTTP

- Apache exporter
- HAProxy exporter (**official**)
- Nginx metric library
- Nginx VTS exporter
- Passenger exporter
- Squid exporter
- Tinyproxy exporter
- Varnish exporter
- WebDriver exporter

APIs

- AWS ECS exporter
- AWS Health exporter
- AWS SQS exporter
- Cloudflare exporter
- DigitalOcean exporter
- Docker Cloud exporter
- Docker Hub exporter
- GitHub exporter
- InstaClustr exporter
- Mozilla Observatory exporter
- OpenWeatherMap exporter
- Pagespeed exporter
- Rancher exporter
- Speedtest exporter

Hardware related

- apcupsd exporter
- Collins exporter
- IBM Z HMC exporter
- IoT Edison exporter
- IPMI exporter
- knxd exporter
- Netgear Cable Modem Exporter
- Netgear Router exporter
- Node/system metrics exporter (**official**)
- NVIDIA GPU exporter
- ProSAFE exporter
- Ubiquiti UniFi exporter

Messaging systems

- Beanstalkd exporter
- EMQ exporter
- Gearman exporter
- Kafka exporter
- NATS exporter
- NSQ exporter
- Mirth Connect exporter
- MQTT blackbox exporter
- RabbitMQ exporter
- RabbitMQ Management Plugin exporter

Other monitoring systems

- Akamai Cloudmonitor exporter
- Alibaba Cloudmonitor exporter
- AWS CloudWatch exporter (**official**)
- Cloud Foundry Firehose exporter
- Collectd exporter (**official**)
- Google Stackdriver exporter
- Graphite exporter (**official**)
- Heka dashboard exporter
- Heka exporter
- InfluxDB exporter (**official**)
- JavaMelody exporter
- JMX exporter (**official**)
- Munin exporter
- Nagios / Naemon exporter
- New Relic exporter
- NRPE exporter
- Osquery exporter
- OTC CloudEye exporter
- Pingdom exporter
- scollector exporter
- Sensu exporter
- SNMP exporter (**official**)
- StatsD exporter (**official**)

Databases

- Aerospike exporter
- ClickHouse exporter
- Consul exporter (**official**)
- Couchbase exporter
- CouchDB exporter
- Elasticsearch exporter
- EventStore exporter
- Memcached exporter (**official**)
- MongoDB exporter
- MSSQL server exporter
- MySQL server exporter (**official**)
- OpenTSDB Exporter
- Oracle DB Exporter
- PgBouncer exporter
- PostgreSQL exporter
- ProxySQL exporter
- RavenDB exporter
- Redis exporter
- RethinkDB exporter
- SQL exporter
- Tarantool metric library
- Twemproxy

Preprocessing

PROMETHEUS

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
```

```
# TYPE wmi_logical_disk_free_bytes gauge
```

```
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
```

```
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
```

```
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

```
wmi_service_state{name="devicesflowusersvc_7b100",state="running"} 1
```

```
wmi_service_state{name="devicesflowusersvc_7b100",state="start pending"} 0
```

```
wmi_service_state{name="devicesflowusersvc_7b100",state="stop pending"} 0
```

```
wmi_service_state{name="devicesflowusersvc_7b100",state="stopped"} 0
```

```
wmi_service_state{name="devicesflowusersvc_7b100",state="unknown"} 0
```

```
wmi_service_state{name="dhcp",state="continue pending"} 0
```

```
wmi_service_state{name="dhcp",state="pause pending"} 0
```

```
wmi_service_state{name="dhcp",state="paused"} 0
```

```
wmi_service_state{name="dhcp",state="running"} 1
```

```
wmi_service_state{name="dhcp",state="start pending"} 0
```

```
wmi_service_state{name="dhcp",state="stop pending"} 0
```

```
wmi_service_state{name="dhcp",state="stopped"} 0
```

```
wmi_service_state{name="dhcp",state="unknown"} 0
```

Preprocessing

PROMETHEUS

```
# HELP cpu_usage_system Telegraf collected metric
# TYPE cpu_usage_system gauge
cpu_usage_system{cpu="cpu-total",host="host1"} 1.1940298507220641
cpu_usage_system{cpu="cpu0",host="host1"} 1.1940298507220641
cpu_usage_system{cpu="cpu1",host="host1"} 1.1340298507220641
```

Item

Preprocessing

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	Prometheus pattern	cpu_usage_system{cpu="cpu-total"}	<label name>	☒ Test Remove
Custom on fail		Discard value	Set value to	Set error to
Add		Test all steps		
Add		Cancel		

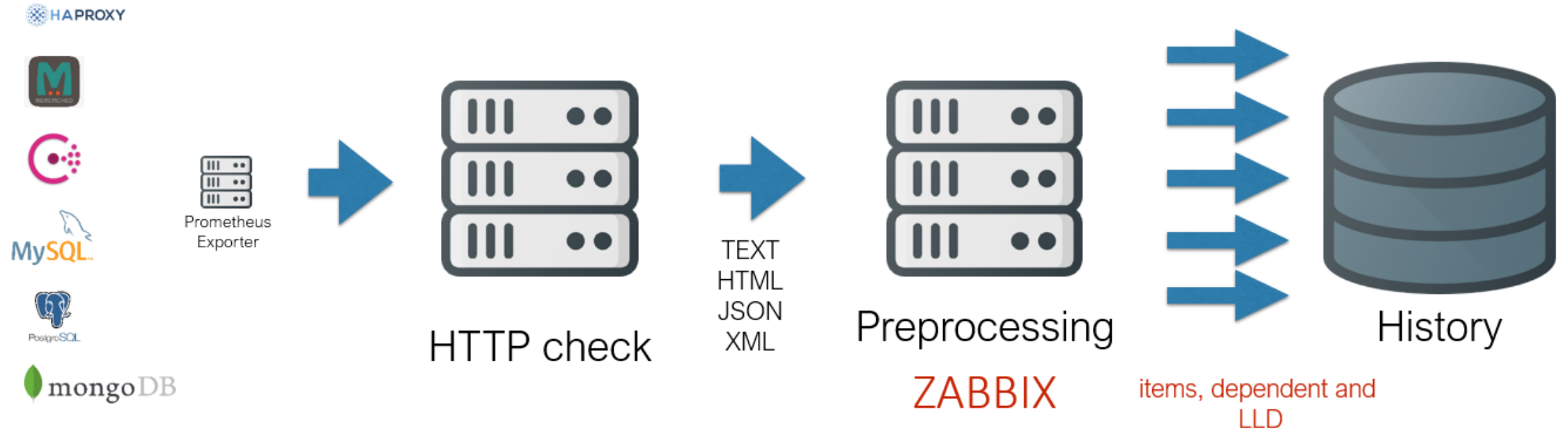
cpu_usage_system{cpu="cpu-total"}



1.1940298507220641

Preprocessing

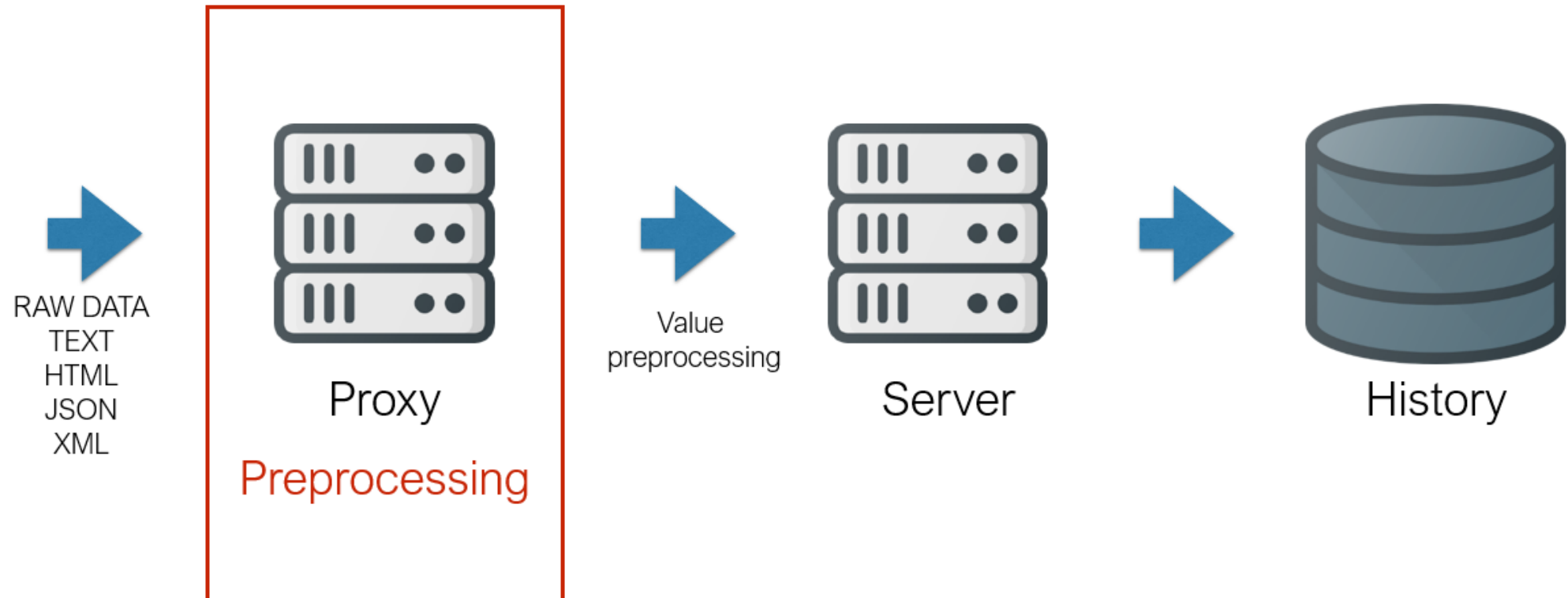
DATA FLOW



Prometheus data processing

Preprocessing

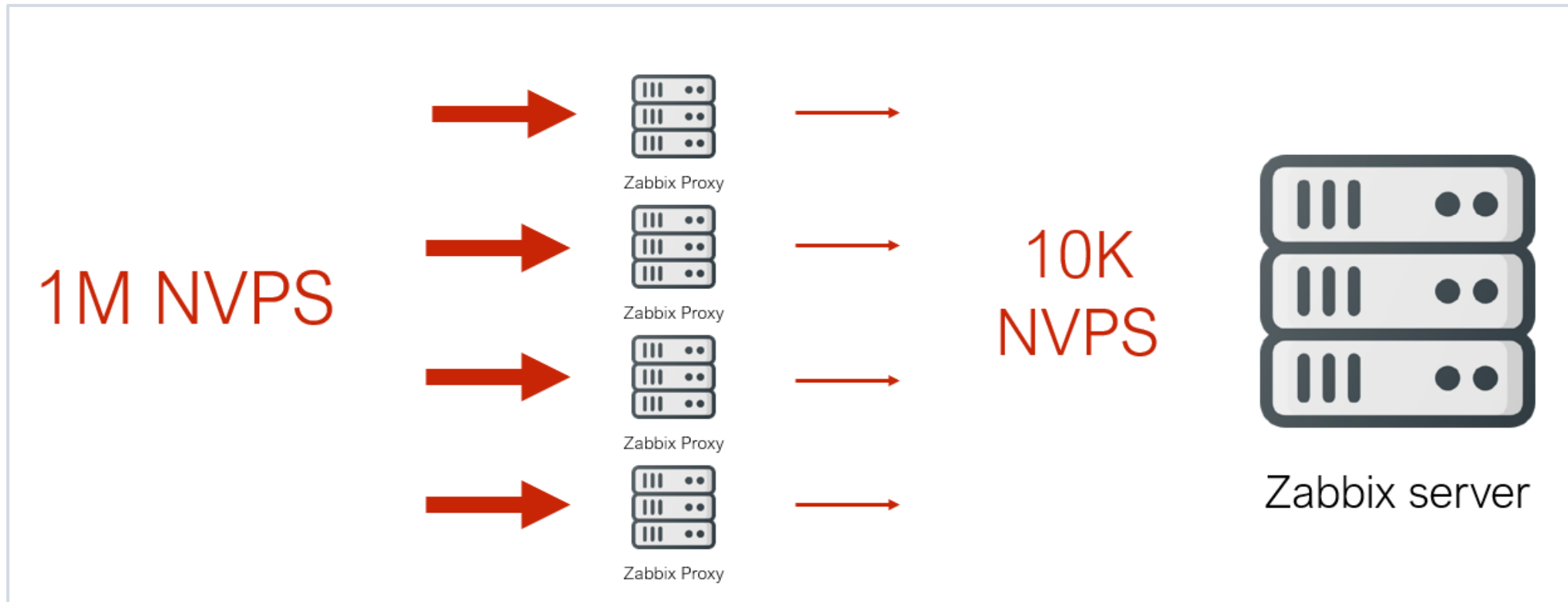
DATA FLOW



Preprocessing

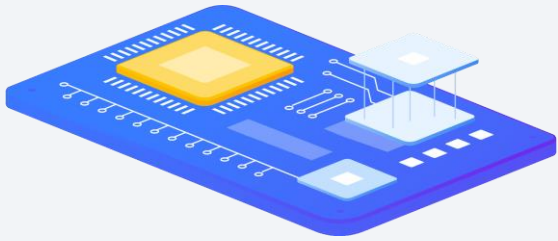
DATA FLOW

- Effective load scaling on the account of preprocessing being performed on the proxy side



Preprocessing

DATA FLOW



Preprocessing on the proxy
side



Advantages



Improved Zabbix server performance by offloading preprocessing



Proxy + throttling enables high frequency monitoring

Preprocessing

TESTING THE PREPROCESSING STEPS

- ▶ Testing of the defined preprocessing steps can be done on the web frontend for rapid debugging

Preprocessing steps	Name	Parameters	Custom on fail	Actions
1:	Left trim	-	☐	Test Remove
2:	Custom multiplier	8	☐	Test Remove
3:	JavaScript	<code>return Math.round(value)</code>	☐	Test Remove
Add				Test all steps

Test item

Value

Time

☐ Not supported

Previous value

Prev. time

End of line sequence

☒ LF ☐ CRLF

Preprocessing steps

Name	Result
1: Left trim	1.234
2: Custom multiplier	9.872
3: JavaScript	10

Result

10

Preprocessing

USEFUL LINKS

- › XPath: "xmllint" or <https://codebeautify.org/Xpath-Tester>
- › PCRE: "grep -P" or <https://regex101.com/>
- › JsonPath: <https://jsonpath.com>

- › Official Zabbix documentation

2

News!



Preprocessing

ZABBIX 6.0 News - preprocessing

- › Bulk processing for Prometheus metrics
- › Result processing for Prometheus pattern
- › Field size limit - Item preprocessing parameters
- › Handling of NaN values in Prometheus preprocessing

Preprocessing

ZABBIX 6.2 News – preprocessing

- ▶ Limits for JavaScript objects in preprocessing (backported in **6.0.14**)

The total size of all messages that can be logged with the `log()` method has been limited to **8 MB** per script execution.

The initialization of multiple `HttpRequest` objects has been limited to **10 per script execution**.

The total length of header fields that can be added to a single `HttpRequest` object with the `addHeader()` method has been limited to **128 Kbytes** (special characters and header names included).

Preprocessing

ZABBIX 6.4 News - preprocessing

› **Two new preprocessing steps have been added**

SNMP walk to JSON - convert SNMP walk values to JSON

SNMP walk value - extract and format SNMP walk value

› **Thread-based preprocessing workers**

The item value preprocessing has been rewritten to use thread-based preprocessing workers for enhanced parallelism and reduced overhead. This development should help avoid situations when the prolonged preprocessing of one item holds up others.

This change adds a new required library for Zabbix server/proxy - libevent_pthreads.

2

Demonstration





Questions?



CONTACT US:

Phone:



+420 800 244 442

Web:



<https://www.initmax.cz>

Email:



tomas.hermanek@initmax.cz

LinkedIn:



<https://www.linkedin.com/company/initmax>

Twitter:



<https://twitter.com/initmax>

Tomáš Heřmánek:



+420 732 447 184