initMAX

Webinar

Advanced problem detection

all our microphones are muted

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

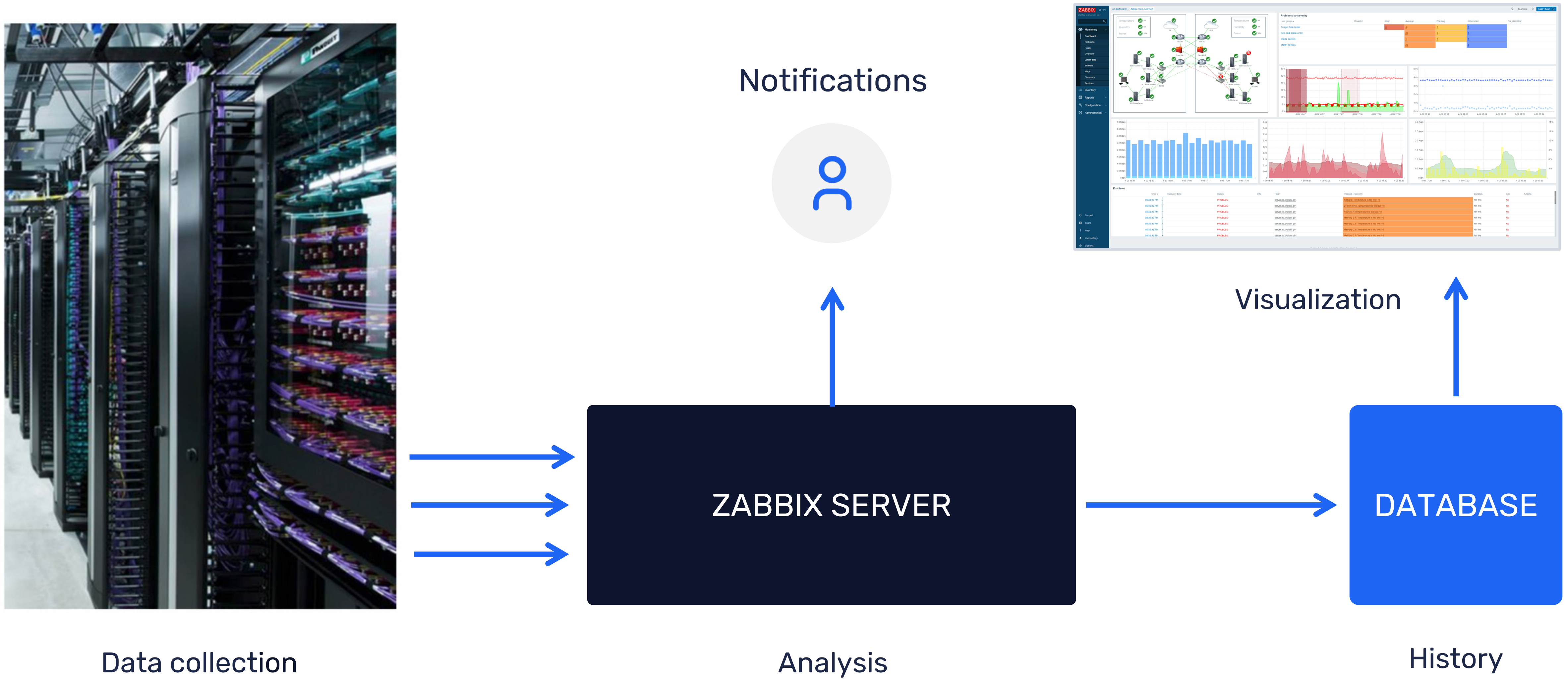
use Chat for discussion, networking or applause

1

Zabbix data flow

ADVANCED PROBLEM DETECTION

Zabbix data flow



How often to execute checks?

Every N seconds

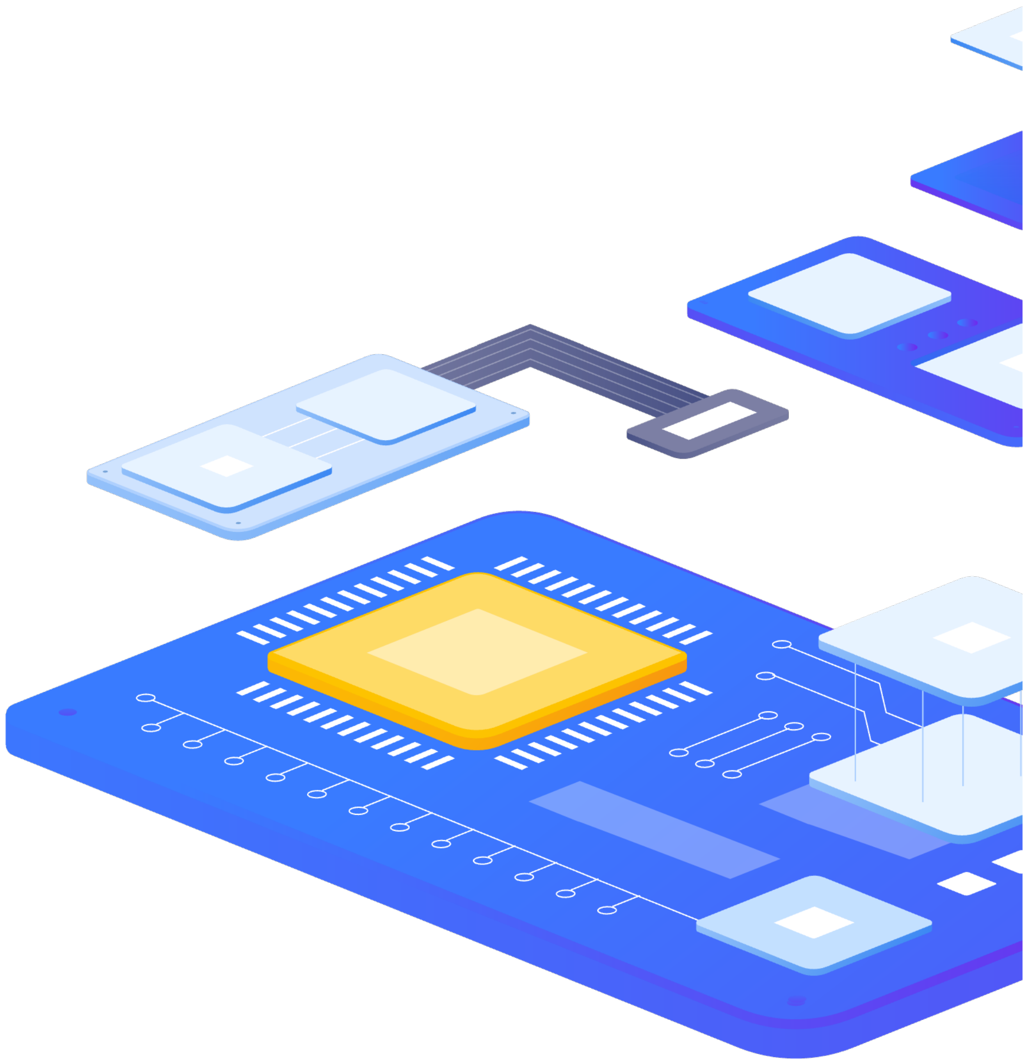
- Zabbix will evenly distribute checks

Different frequency in different time periods

- Every X seconds in working time
- Every Y second in weekend

At a specific time (Zabbix 3.0)

- Ready for business checks
- Every hour starting from 9:00 at working hours (9:00, 10:00,…, 18:00)



2

Triggers

Trigger – problem definition

function(/host/key,parameter)<operator><constant>

- last(/server/system.cpu.load) > 5

Operators

- - + / * < > = <> >= <= not or and

Functions

- min max avg last count date time diff regexp and much more!

Analyze everything: any metric and any host

- last(/node1/system.cpu.load) > 5 and last(/node2/system.cpu.load) > 5 and last(/nodes/tps) < 5000

Scope of Usage

- Triggers, calculated items, expression macros

Macros – variable

Zabbix supports a number of built-in macros which may be used in various situations. These macros are variables, identified by a specific syntax:

- {MACRO}

Zabbix supports the following macros:

{MACRO}	- built-in macro
{<macro>.<func>(<params>)}	- macro functions
{$MACRO}	- user-defined macro, optionally with context
{#MACRO}	- macro for low-level discovery
{?EXPRESSION}	- expression macro

User Macros

Macro resolution precedence:

1. host level macros (checked first)
2. macros defined for first level templates of the host (i.e., templates linked directly to the host), sorted by template ID
3. macros defined for second level templates of the host, sorted by template ID
4. macros defined for third level templates of the host, sorted by template ID, etc.
5. global macros (checked last)

- If a macro does not exist for a host, Zabbix will try to find it in the host templates of increasing depth. If still not found, a global macro will be used, if exists.

Macros in trigger expressions

User macros can be used in:

- trigger name and description
- trigger expression parameters and constants

Examples:

- net.tcp.service[ssh,,{$SSH_PORT}]
- last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
- min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}

User Macros with context

An optional context can be used in user macros, allowing to override the default value with a context-specific one.

- {$MACRO:"static text"}
- {$MACRO:regex:"regular expression"}

Examples:

- {$LOW_SPACE_LIMIT:/tmp}
- {$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}

Trigger Example:

- last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}

Expression Macro

{?EXPRESSION_MACROS}

- If defined, this name will be used to create the problem event name, instead of the trigger name.
- The event name may be used to build meaningful alerts containing problem data
- The same set of macros is supported as in the trigger name, plus {TIME} and {?EXPRESSION} expression macros.
- Supported since Zabbix 5.2.0
- Can be used in different locations – **Event Name,** Maps, name of Graphs

Expression Macro

Junior

- Problem: Load of **Exchange** server increased by more than 10% last month

Expert

- Problem: Load of **Exchange** server increased by **24%** in **July** (**0.69**) comparing to **June** (**0.56**)

- Load of {HOST.HOST} server increased by
  - {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.load,1M:now/M-1M)}.fmtnum(0)}% in
  - {{TIME}.fmttime(%B,-1M)}
  - ({{?trendavg(//system.cpu.load,1M:now/M)}.fmtnum(2)}) comparing to
  - {{TIME}.fmttime(%B,-2M)}
  - ({{?trendavg(//system.cpu.load,1M:now/M-1M)}.fmtnum(2)})

https://www.zabbix.com/documentation/6.0/en/manual/config/triggers/expression?hl=expression#examples-of-triggers

3

Trigger Functions

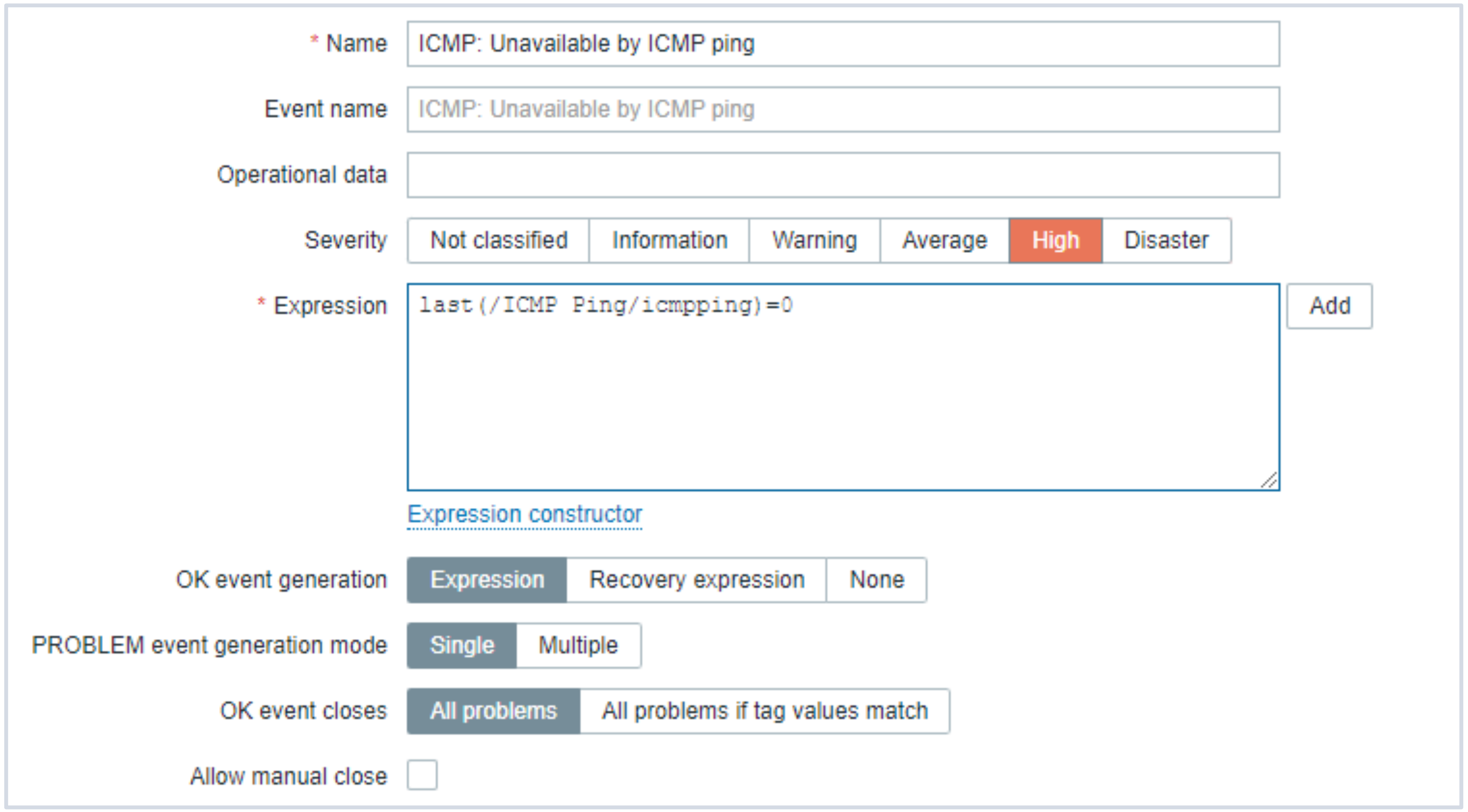
Basic functions - last

last(/host/key,parameter)

- The most recent value.
- Supported value types: Float, Integer, String, Text, Log.

- Parameters:
  - See common parameters;
  - #num (optional) - the Nth most recent value.

Configuration



* Name	ICMP: Unavailable by ICMP ping
Event name	ICMP: Unavailable by ICMP ping
Operational data	
Severity	Not classified / Information / Warning / Average / [✓ High] / Disaster
* Expression	`last(/ICMP Ping/icmpping)=0`
	Expression constructor
OK event generation	[✓ Expression] / Recovery expression / None
PROBLEM event generation mode	[✓ Single] / Multiple
OK event closes	[✓ All problems] / All problems if tag values match
Allow manual close	☐

initMAX

ADVANCED PROBLEM DETECTION

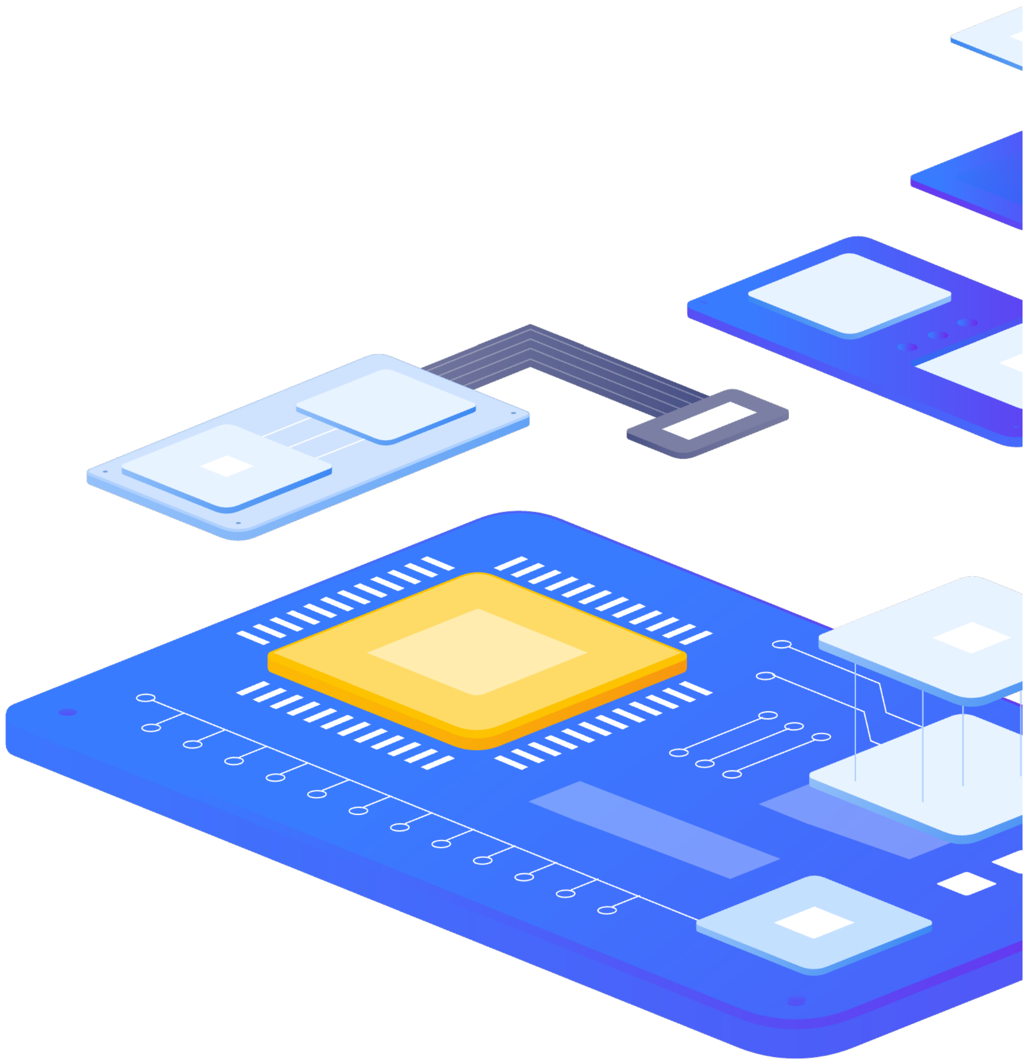
Junior level

Performance

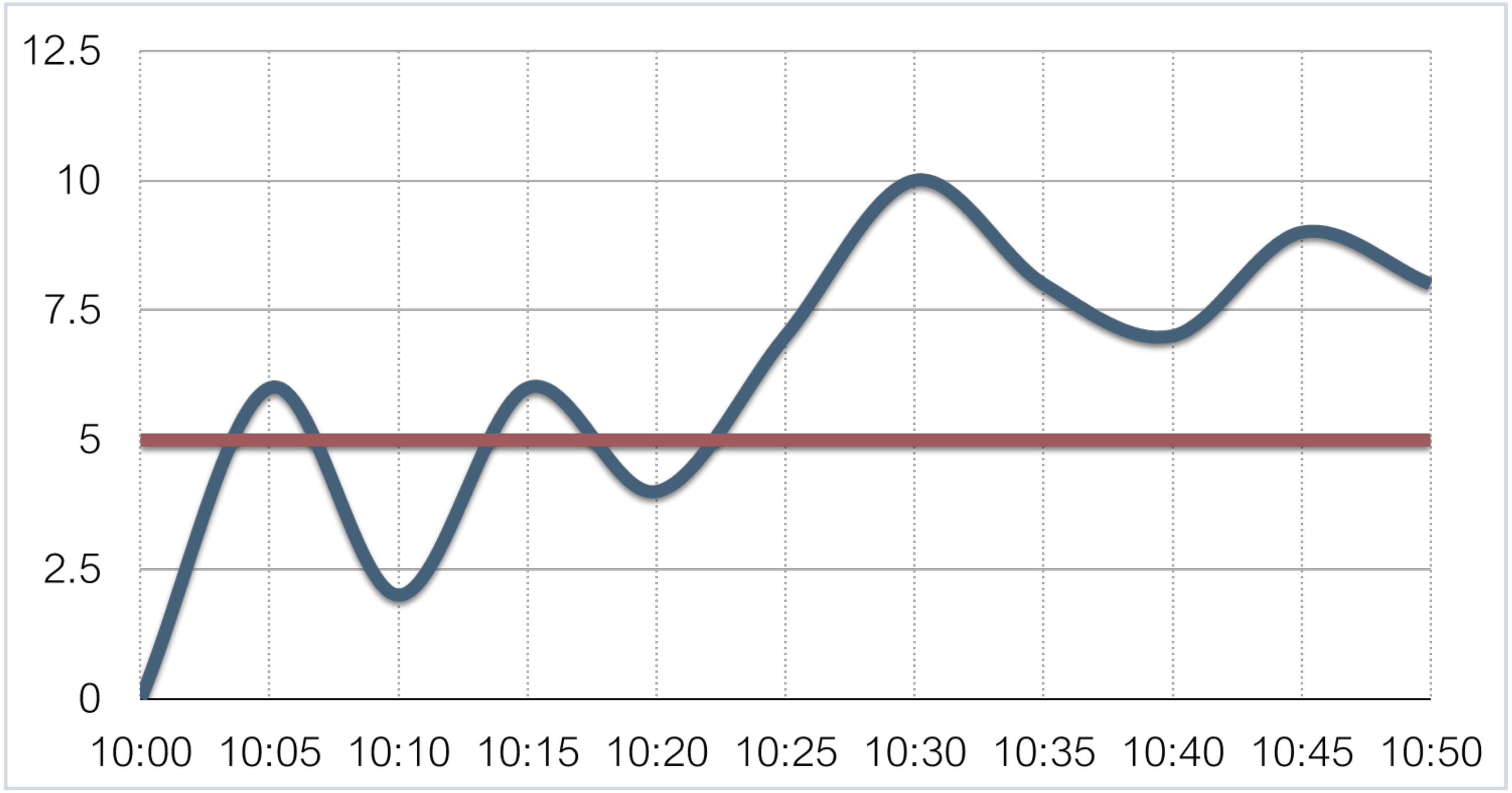
- last(/server/system.cpu.load) > 5

Availability

- last(/server/net.tcp.service[http]) = 0



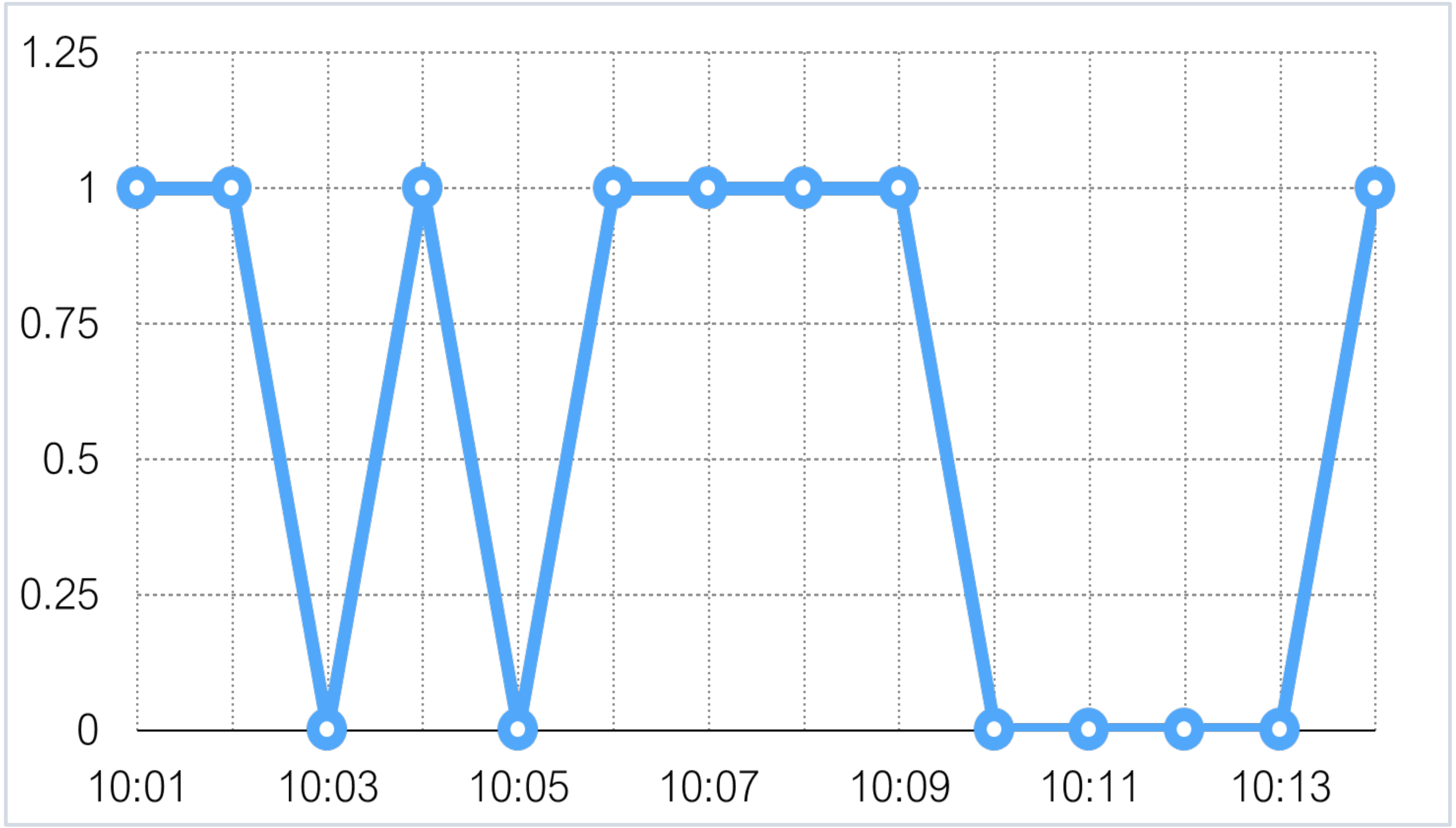
False positives



last(/server/system.cpu.load) > 5

ADVANCED PROBLEM DETECTION

Too sensitive

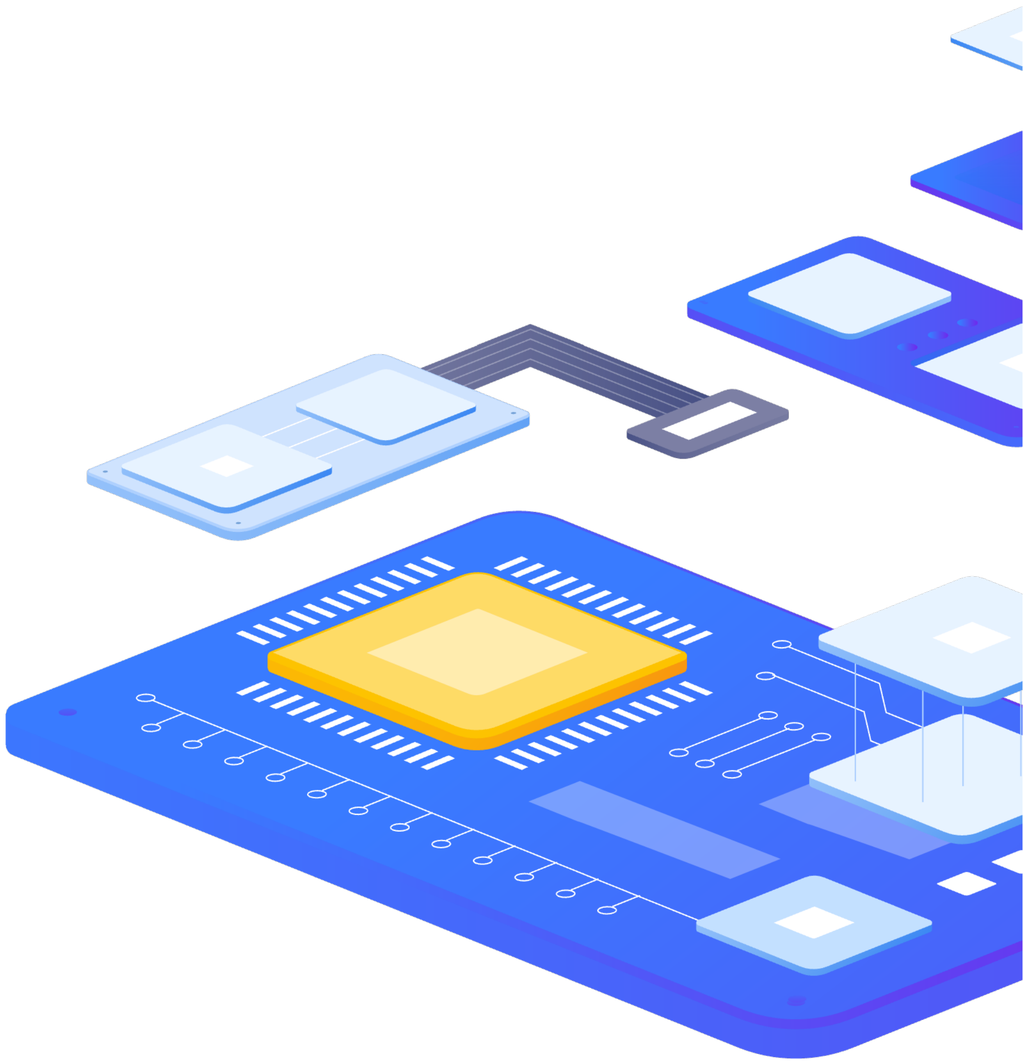


last(/server/net.tcp.service[http]) = 0

Junior level

Too sensitive leads to

- False positives



4

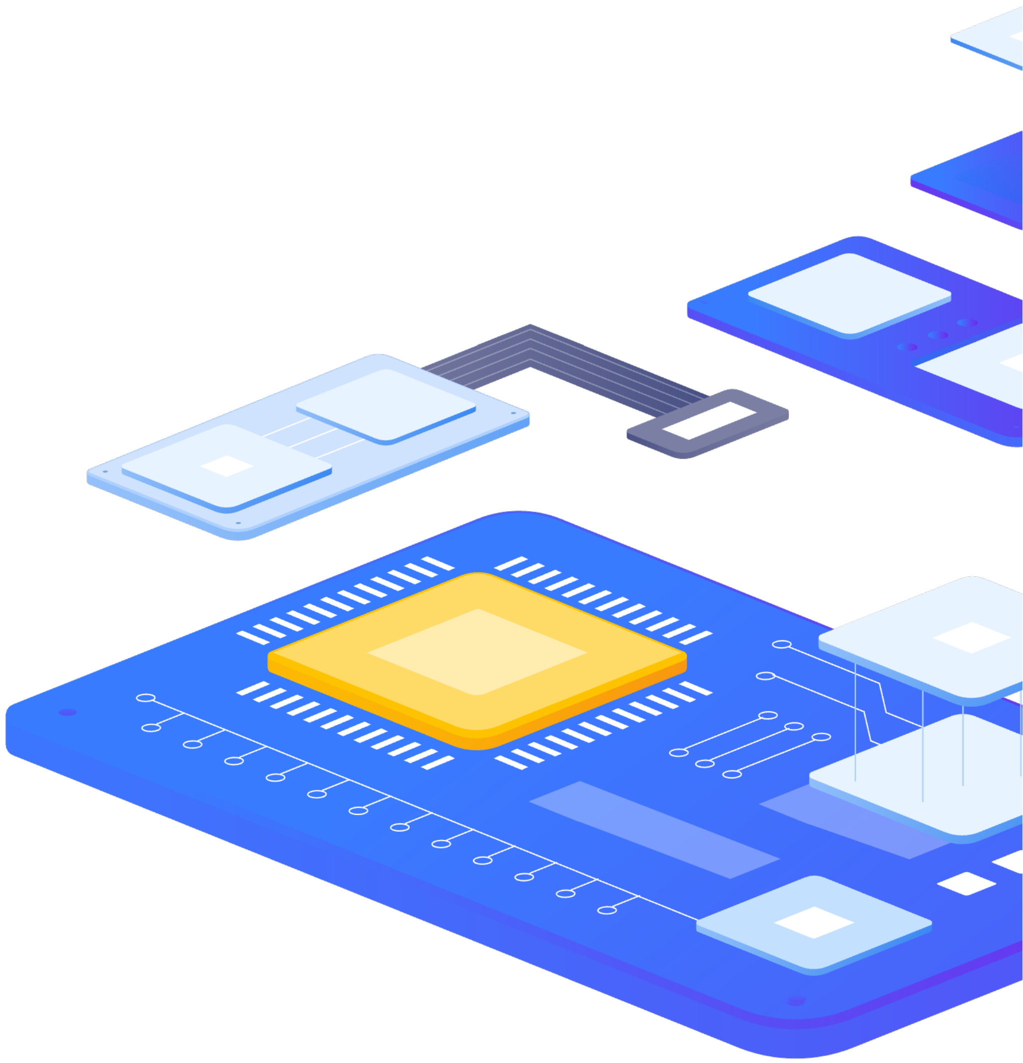
False positives

How to avoid false positives?

Be careful and define problems wisely!

What does it really mean?

- system is overloaded
- application does not work
- service is not available



Examples

Problem:

- CPU load > 5

No problem:

- CPU load = 4.99 $\longrightarrow$ **Resolved?**

Problem:

- free disk space < 10%

No problem:

- free disk space = 10.001% $\longrightarrow$ **Resolved?**

Problem:

- SSH check failed

No problem:

SSH is up $\longrightarrow$ **Resolved?**

Analyze history

Performance

- min(/server/system.cpu.load,10m) > 5

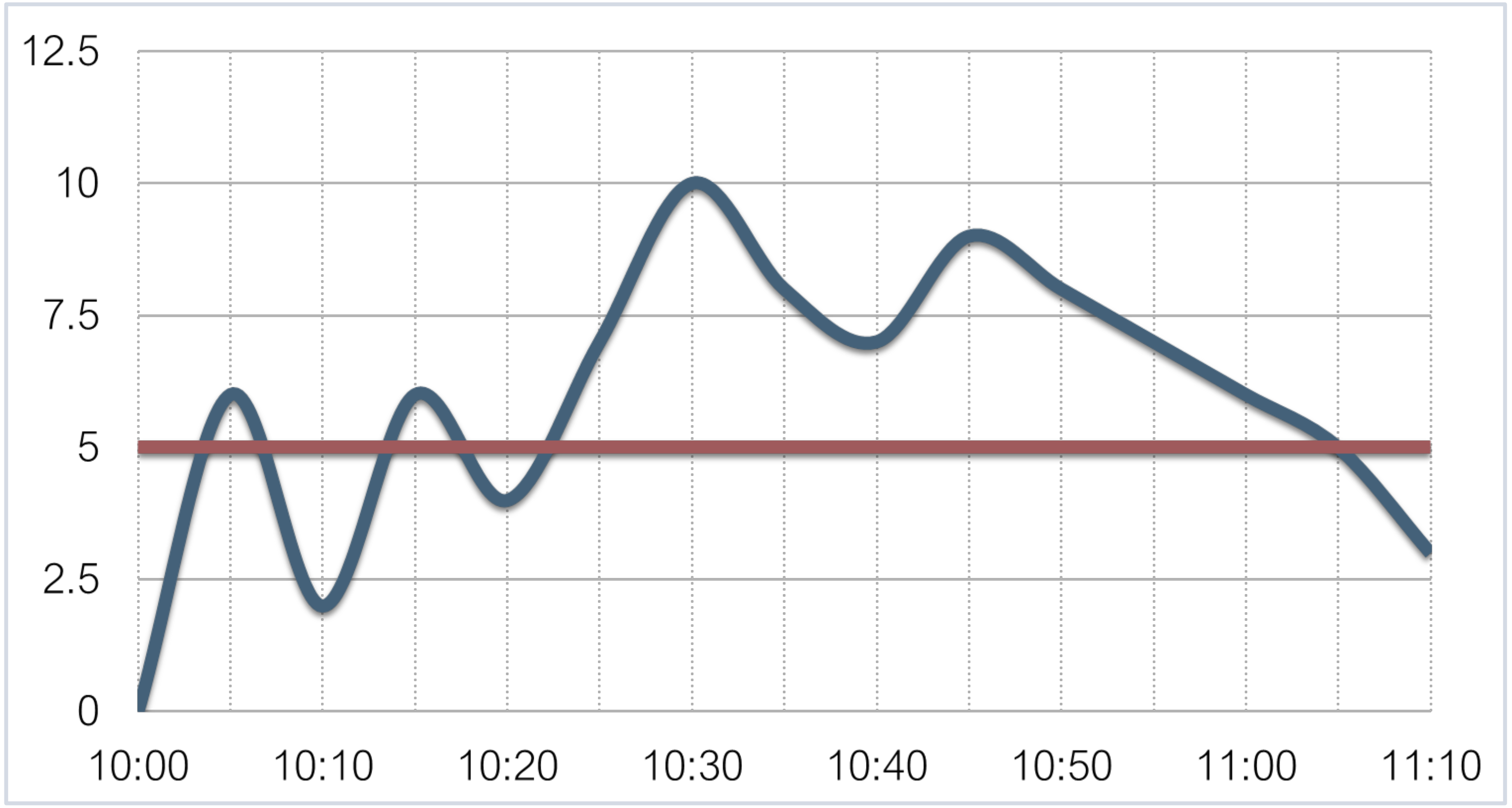
Availability

- max(/server/net.tcp.service[http],5m) = 0
- max(/server/net.tcp.service[http],#3) = 0

initMAX

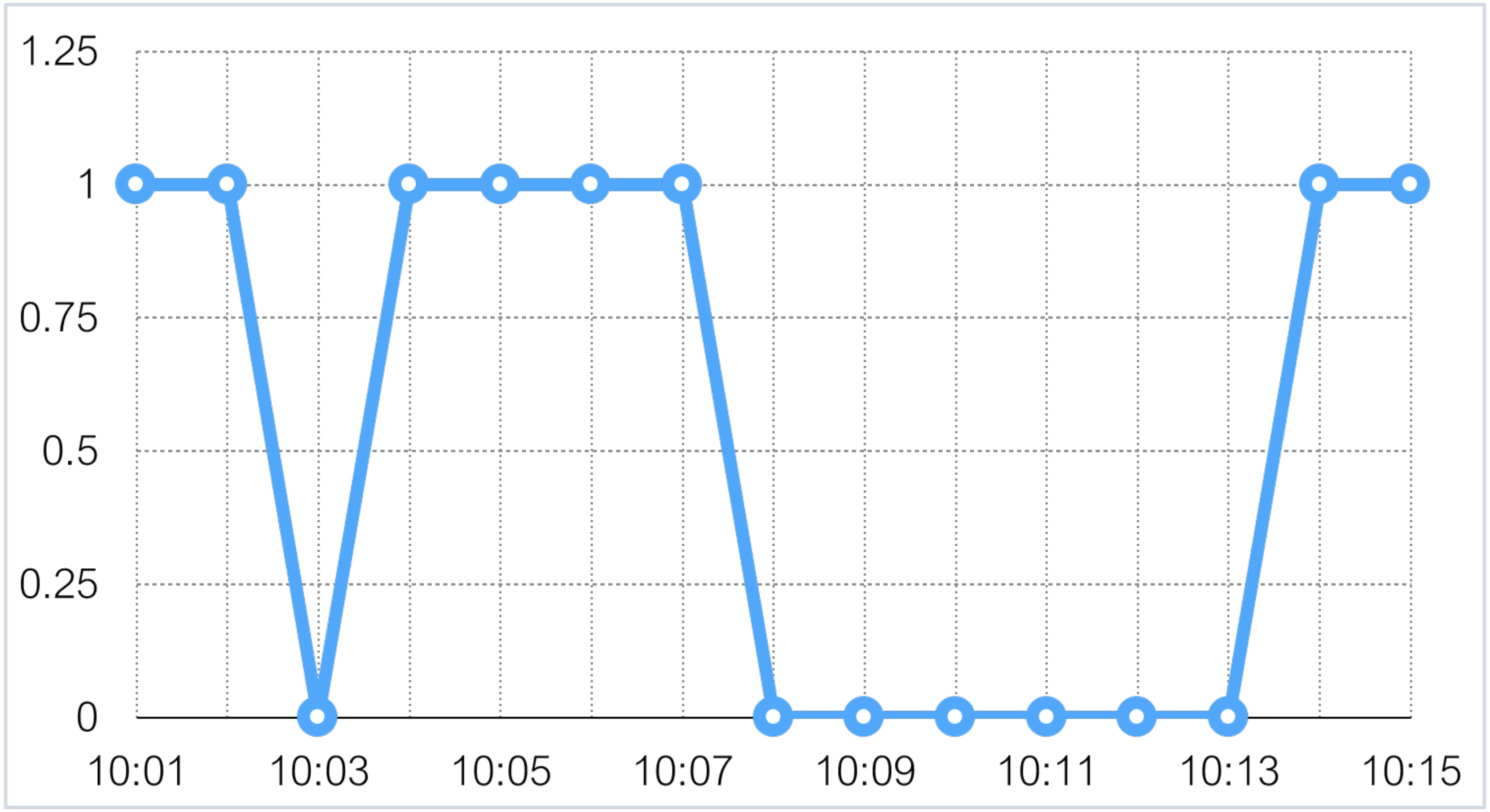
ADVANCED PROBLEM DETECTION

Analyze history



min(/server/system.cpu.load,10m) > 5

Analyze history



max(/server/net.tcp.service[http],#3) = 0

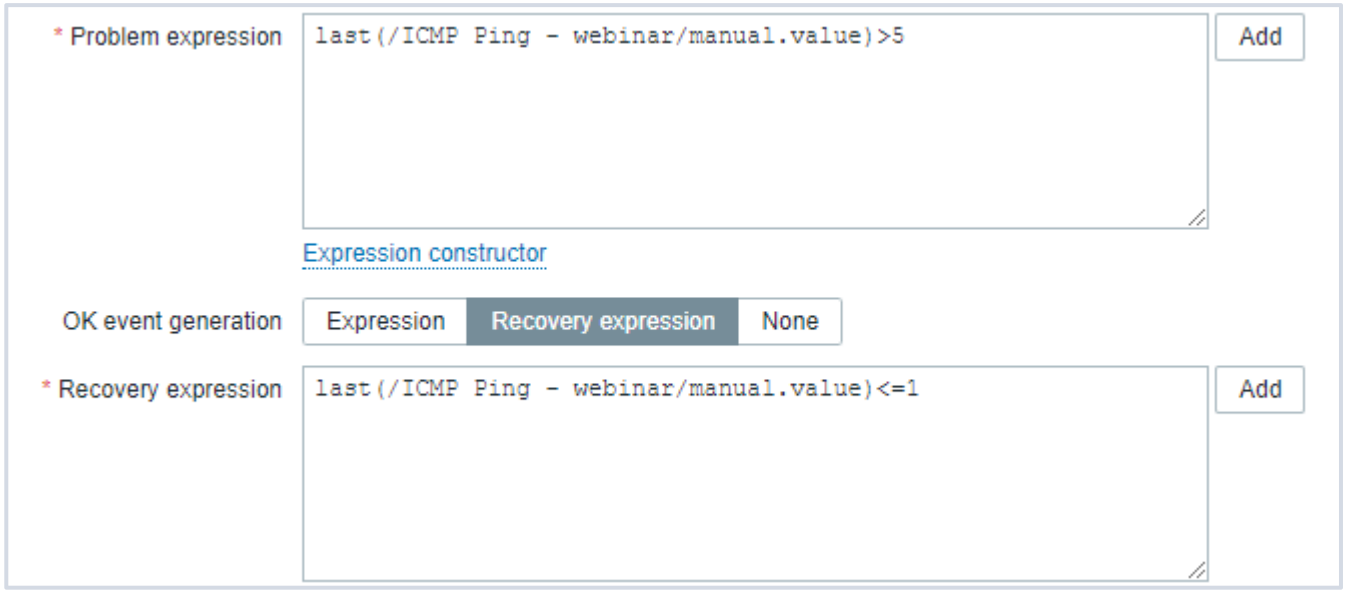
Different conditions for problem and recovery

Before

- last(/server/system.cpu.load) > 5

Now

- Problem definition: last(/server/system.cpu.load)>5
- Recovery expression: last(/server/system.cpu.load)}<=1



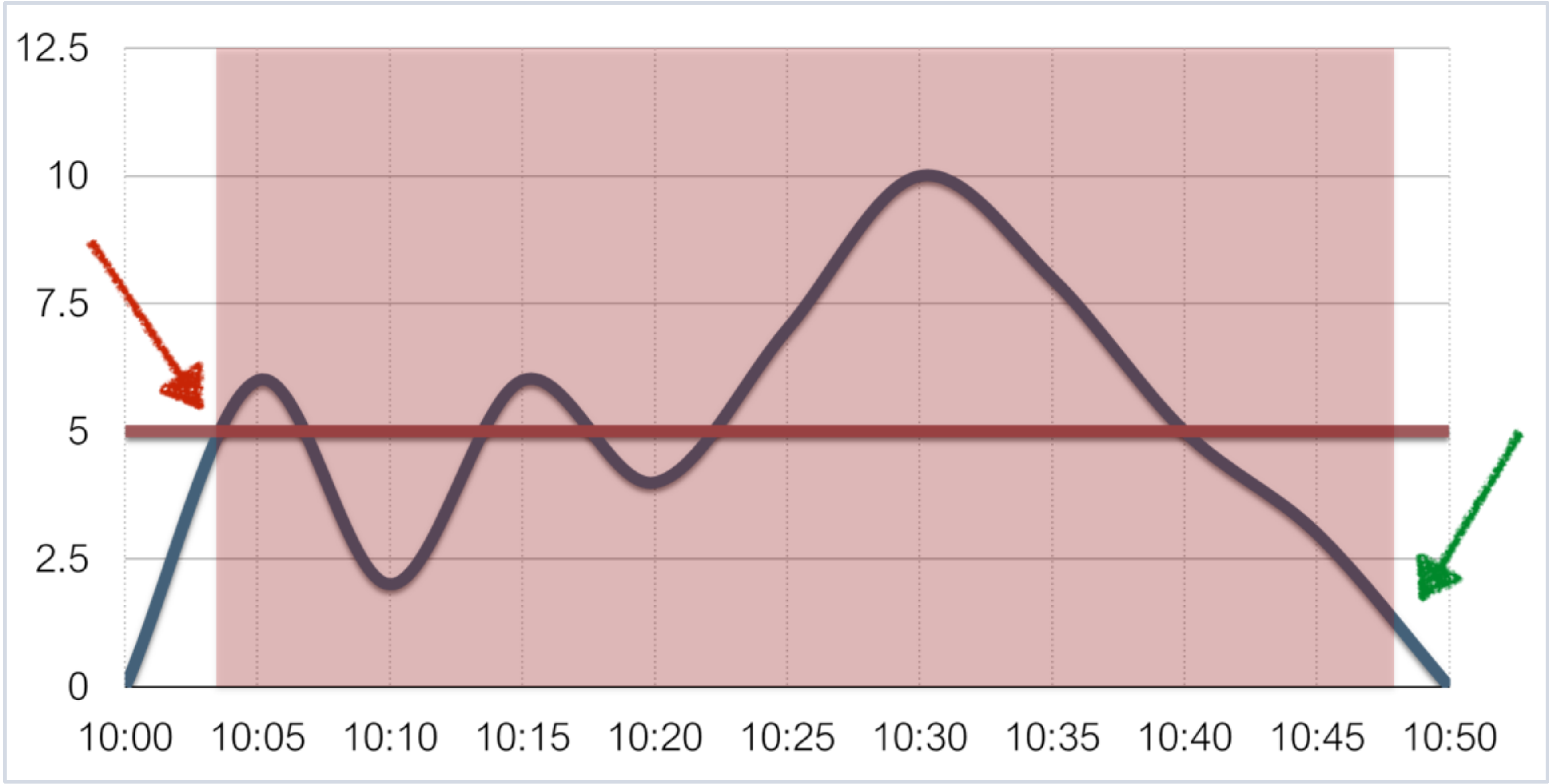
Problem expression: `last(/ICMP Ping - webinar/manual.value)>5`

OK event generation: Expression [✓ Recovery expression] None

Recovery expression: `last(/ICMP Ping - webinar/manual.value)<=1`

ADVANCED PROBLEM DETECTION

Different conditions for problem and recovery



Problem definition: last(/server/system.cpu.load)>5 …Recovery expression: last(/server/system.cpu.load)}<=1

Examples

System is overloaded

Problem definition:

- min(/server/system.cpu.load,5m)>3

Recovery expression:

- max(/server/system.cpu.load,2m)<=1

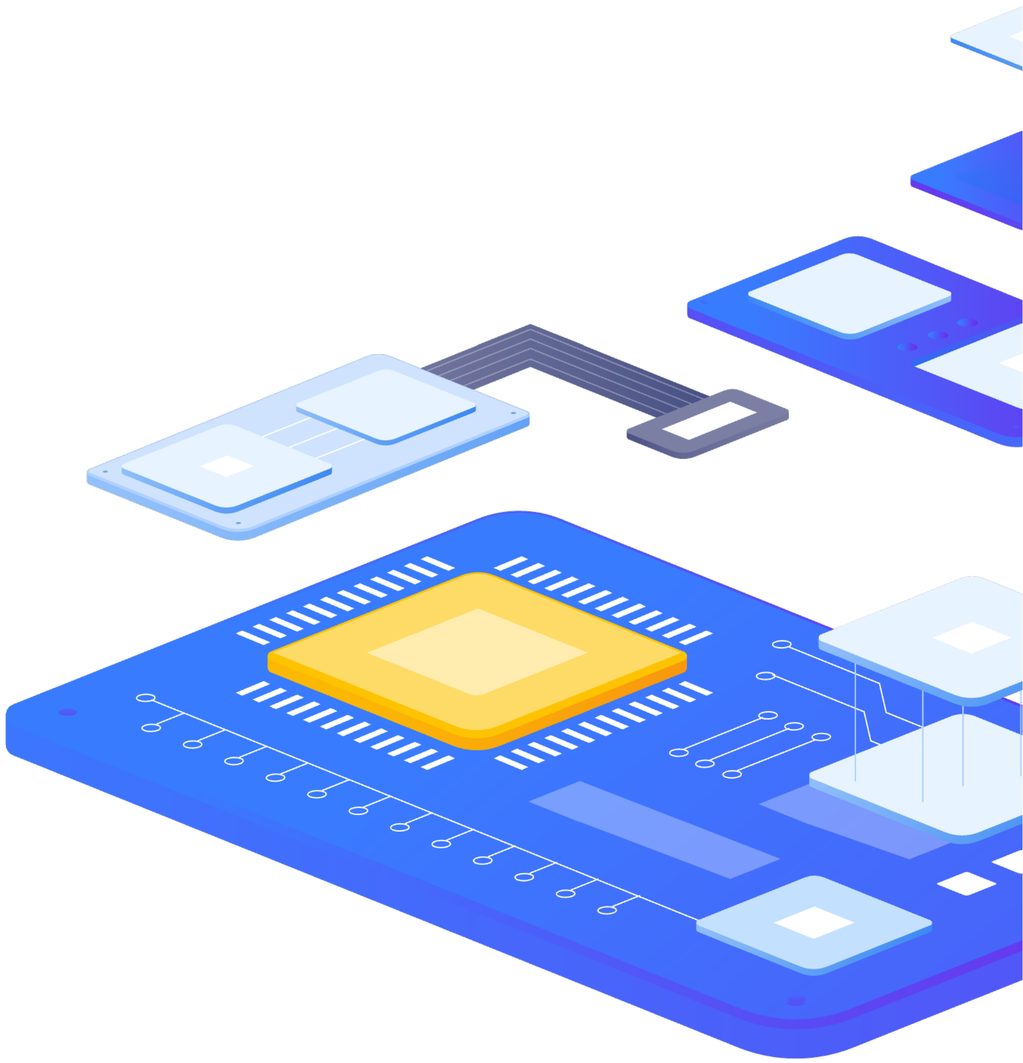
No free disk space /

Problem definition:

- last(/server/vfs.fs.size[/,pfree])<10

Recovery expression:

- min(/server/vfs.fs.size[/,pfree],15m)>30



ADVANCED PROBLEM DETECTION

Examples

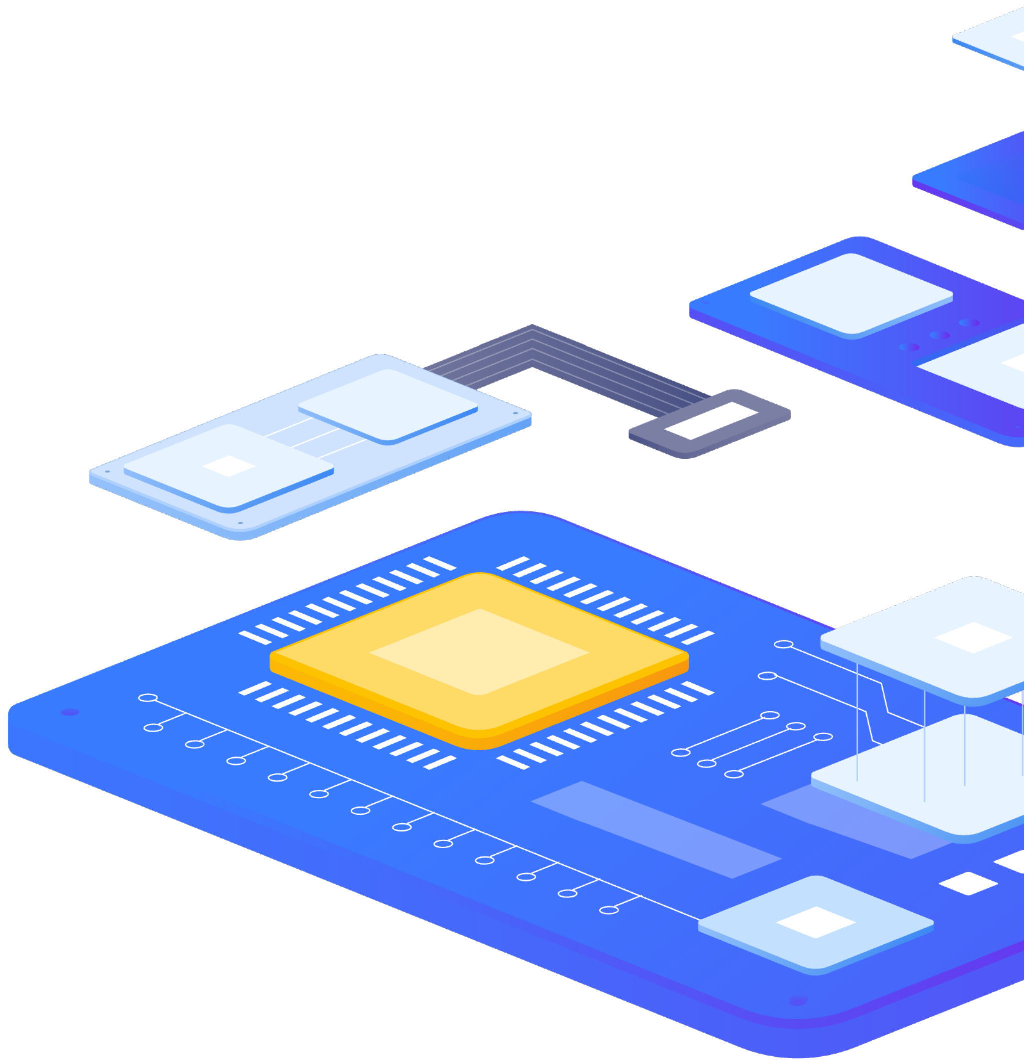
SSH is not available

Problem definition:

- max(/server/net.tcp.service[ssh],#3)=0

Recovery expression:

- min(/server/net.tcp.service[ssh],#10)=1



Anomalies

How to detect?

By comparing with the data from the same period, the period is taken from the past.

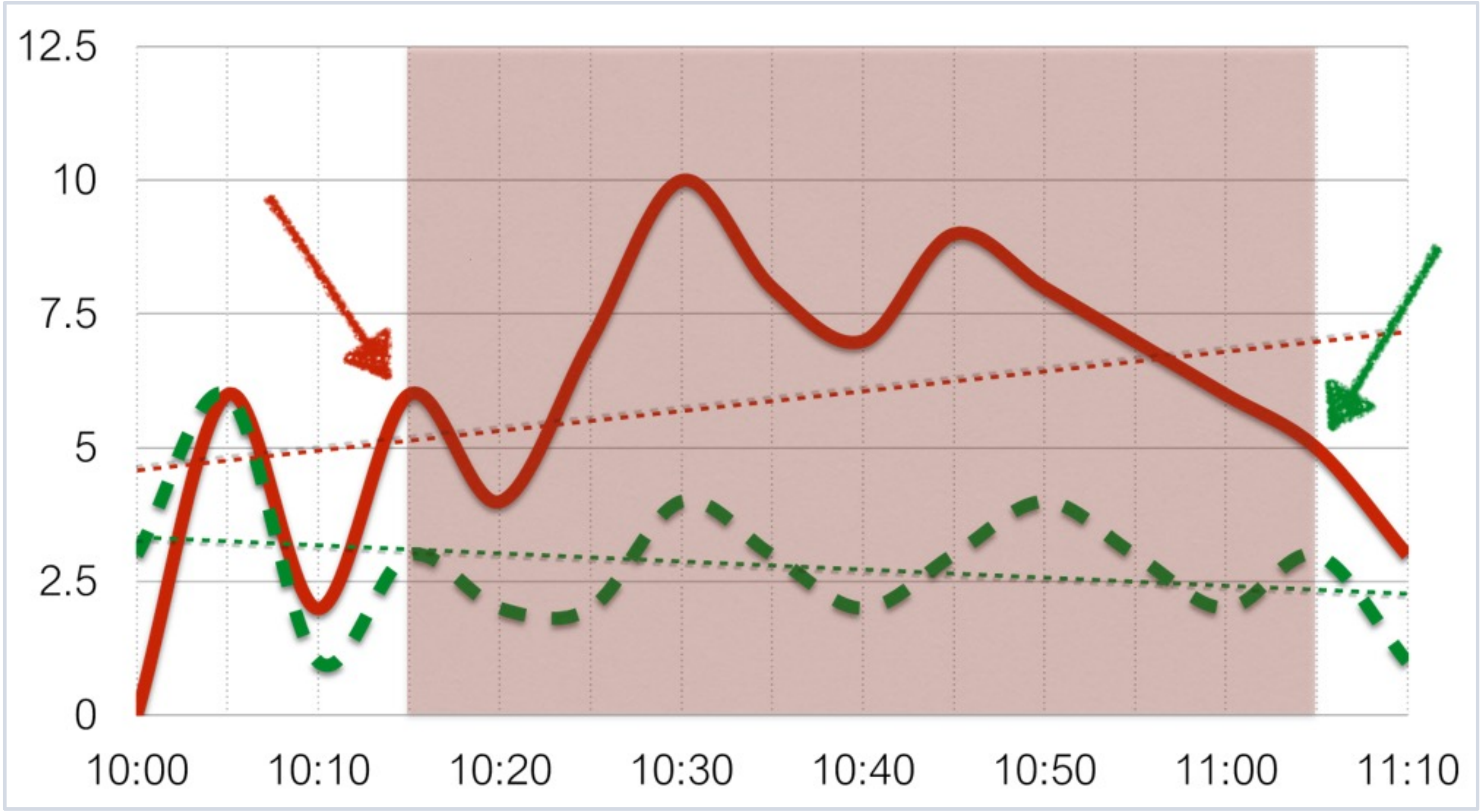
Average CPU load for the last hour is 2x higher than

CPU load for the same period week ago

- avg(/server/system.cpu.load,1h) > 2* avg(/server/system.cpu.load,1h:now-7d)

ADVANCED PROBLEM DETECTION

Anomalies



Comparison with the data 7 days ago

ADVANCED PROBLEM DETECTION

Flapping

How to detect?

By comparing changecount of the data from the time period.

Operational status changes of interface

- changecount(/SNMP v2/net.if.status[ifOperStatus.{#SNMPINDEX}],{$FLAP.PERIOD})>{$FLAP.NUMBER}

Trigger dependency

Link down -> Flapping Detected

5

Agregate functions

Basic functions – min,max,avg

min(/host/key,parameter,#3)

max(/host/key,parameter,#3)

avg(/host/key,parameter,#3)

- The lowest value of an item within the defined evaluation period.
- The highest value of an item within the defined evaluation period.
- The average value of an item within the defined evaluation period.
- Supported value types: Float, Integer.

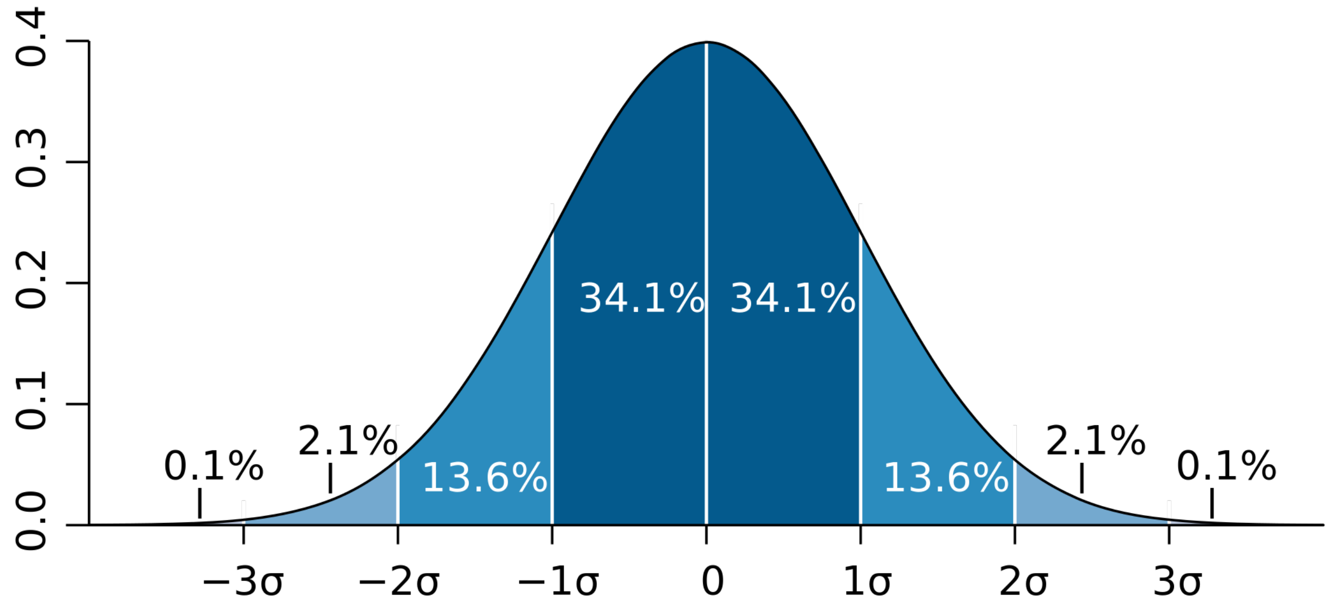
Basic functions – stddevsamp, stddevpop

stddevpop(/host/key,1h)

stddevsamp(/host/key,1h)

- The population standard deviation in collected values within the defined evaluation period.
- The sample standard deviation in collected values within the defined evaluation period.

$$s_N = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2}.$$



Aggregate functions

avg	The average value of an item within the defined evaluation period.
bucket_percentile	Calculates the percentile from the buckets of a histogram.
count	The count of values in an array returned by a foreach function.
histogram_quantile	Calculates the φ-quantile from the buckets of a histogram.
item_count	The count of existing items in configuration that match the filter criteria.
kurtosis	The "tailedness" of the probability distribution in collected values within the defined evaluation period.
mad	The median absolute deviation in collected values within the defined evaluation period.
max	The highest value of an item within the defined evaluation period.
min	The lowest value of an item within the defined evaluation period.
skewness	The asymmetry of the probability distribution in collected values within the defined evaluation period.
stddevpop	The population standard deviation in collected values within the defined evaluation period.
stddevsamp	The sample standard deviation in collected values within the defined evaluation period.
sum	The sum of collected values within the defined evaluation period.
sumofsquares	The sum of squares in collected values within the defined evaluation period.
varpop	The population variance of collected values within the defined evaluation period.
varsamp	The sample variance of collected values within the defined evaluation period.

6

Mathematical functions

initMAX

Mathematical functions

abs	The absolute value of a value.
acos	The arccosine of a value as an angle, expressed in radians.
asin	The arcsine of a value as an angle, expressed in radians.
atan	The arctangent of a value as an angle, expressed in radians.
atan2	The arctangent of the ordinate (value) and abscissa coordinates specified as an angle, expressed in radians.
avg	The average value of the referenced item values.
cbrt	The cube root of a value.
ceil	Round the value up to the nearest greater or equal integer.
cos	The cosine of a value, where the value is an angle expressed in radians.
cosh	The hyperbolic cosine of a value.
cot	The cotangent of a value, where the value is an angle expressed in radians.
degrees	Converts a value from radians to degrees.
e	The Euler's number (2.718281828459045).

initMAX

ADVANCED PROBLEM DETECTION

Mathematical functions

exp	The Euler's number at a power of a value.
expm1	The Euler's number at a power of a value minus 1.
floor	Round the value down to the nearest smaller or equal integer.
log	The natural logarithm.
log10	The decimal logarithm.
max	The highest value of the referenced item values.
min	The lowest value of the referenced item values.
mod	The division remainder.
pi	The Pi constant (3.14159265358979).
power	The power of a value.
radians	Converts a value from degrees to radians.
rand	Return a random integer value.
round	Round the value to decimal places.
signum	Returns '-1' if a value is negative, '0' if a value is zero, '1' if a value is positive.

Mathematical functions

sin	The sine of a value, where the value is an angle expressed in radians.
sinh	The hyperbolical sine of a value, where the value is an angle expressed in radians.
sqrt	The square root of a value.
sum	The sum of the referenced item values.
tan	The tangent of a value.
truncate	Truncate the value to decimal p

Mathematical min x aggregate min:

- min(<value1>,<value2>,...)
- min(avg(/host/key),avg(/host2/key2))

x

- min(/host/key,parameter,#3)

7

History functions

ADVANCED PROBLEM DETECTION

fuzzytime

fuzzytime(/host/key,60s)

- Check how much the passive agent time differs from the Zabbix server/proxy time.
- fuzzytime(/host/key,60s)=0 #detect a problem if the time difference is over 60 seconds

change

change(/host/key)

- The amount of difference between the previous and latest value.
- Supported value types: Float, Integer, String, Text, Log.
- For strings returns: 0 - values are equal; 1 - values differ.

- change(/host/key)>10

changecount

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

- The number of changes between adjacent values within the defined evaluation period.
- Supported value types: Float, Integer, String, Text, Log.
- mode (must be double-quoted) - possible values:
  - all - count all changes (default);
  - dec - count decreases;
  - inc - count increases

- changecount(/host/key,#10,"inc")

count

count(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

- The number of values within the defined evaluation period.
- Supported value types: Float, Integer, String, Text, Log.

- operator (must be double-quoted). Supported operators:
  - eq - equal (default for integer, float)
  - ne - not equal
  - gt - greater
  - ge - greater or equal
  - lt - less
  - le - less or equal
  - like (default for string, text, log) - matches if contains pattern (case-sensitive)
  - bitand - bitwise AND
  - regexp - case-sensitive match of the regular expression given in pattern
  - iregexp - case-insensitive match of the regular expression given in pattern
- pattern - the required pattern (string arguments must be double-quoted).

countunique

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

- The number of unique values within the defined evaluation period.
- Supported value types: Float, Integer, String, Text, Log.

- operator (must be double-quoted). Supported operators:
  - eq - equal (default for integer, float)
  - ne - not equal
  - gt - greater
  - ge - greater or equal
  - lt - less
  - le - less or equal
  - like (default for string, text, log) - matches if contains pattern (case-sensitive)
  - bitand - bitwise AND
  - regexp - case-sensitive match of the regular expression given in pattern
  - iregexp - case-insensitive match of the regular expression given in pattern
- pattern - the required pattern (string arguments must be double-quoted).

History functions

change	The amount of difference between the previous and latest value.
changecount	The number of changes between adjacent values within the defined evaluation period.
count	The number of values within the defined evaluation period.
countunique	The number of unique values within the defined evaluation period.
find	Find a value match within the defined evaluation period.
first	The first (the oldest) value within the defined evaluation period.
fuzzytime	Check how much the passive agent time differs from the Zabbix server/proxy time.
last	The most recent value.
logeventid	Check if the event ID of the last log entry matches a regular expression.
logseverity	The log severity of the last log entry.
logsource	Check if log source of the last log entry matches a regular expression.
monodec	Check if there has been a monotonous decrease in values.
monoinc	Check if there has been a monotonous increase in values.
nodata	Check for no data received.
percentile	The P-th percentile of a period, where P (percentage) is specified by the third parameter.
rate	The per-second average rate of the increase in a monotonically increasing counter within the defined time period.

8

Foreach functions

Foreach functions

avg_foreach	Returns the average value for each item.
bucket_rate_foreach	Returns pairs (bucket upper bound, rate value) suitable for use in the histogram_quantile() function, where "bucket upper bound" is the value of item key parameter defined by the <parameter number> parameter.
count_foreach	Returns the number of values for each item.
exists_foreach	Returns '1' for each enabled item.
last_foreach	Returns the last value for each item.
max_foreach	Returns the maximum value for each item.
min_foreach	Returns the minimum value for each item.
sum_foreach	Returns the sum of values for each item.

ADVANCED PROBLEM DETECTION

Foreach Functions - tip

Calculated Items on:

Host level

- sum(last_foreach(/host/net.if.in[*]))

Hostgroup level

- avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)

TAG level

- avg_foreach(/*/key[a,*,c]?[(tag=„ENV:production")],10m)

Complex level

- avg_foreach(/*/key[a,*,c]?[(group=„Servers" and tag=„EU") or (group=„Linux„) and (tag=„CZ" or tag=„ENV:production"))],5m)

9

Bitwise functions

Bitwise functions

bitand	The value of "bitwise AND" of an item value and mask.
bitlshift	The bitwise shift left of an item value.
bitnot	The value of "bitwise NOT" of an item value.
bitor	The value of "bitwise OR" of an item value and mask.
bitrshift	The bitwise shift right of an item value.
bitxor	The value of "bitwise exclusive OR" of an item value and mask.

10

Date and time functions

now

Example – certificate expiration:

- (last(/Website certificate by Zabbix agent 2/cert.not_after) - now()) / 86400 < {$CERT.EXPIRY.WARN}

now	The number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).

Date and time functions

date	The current date in YYYYMMDD format.
dayofmonth	The day of month in range of 1 to 31.
dayofweek	The day of week in range of 1 to 7.
now	The number of seconds since the Epoch (00:00:00 UTC, January 1, 1970).
time	The current time in HHMMSS format.

11

Trend functions

Trend functions

Trend functions, in contrast to history functions, use trend data for calculations.

- Trends store hourly aggregate values. Trend functions use these hourly averages, and thus are useful for long-term analysis.
- Trend function results are cached so multiple calls to the same function with the same parameters fetch info from the database only once. The trend function cache is controlled by the **TrendFunctionCacheSize** server parameter.

- Triggers that reference trend functions only are evaluated once per the smallest time period in the expression. For instance, a trigger like:

trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2

- will be evaluated once per day. If the trigger contains both trend and history (or time-based) functions, it is calculated in accordance with the usual principles.

12

Baselines

ADVANCED PROBLEM DETECTION

baselinewma

baselinewma (/host/key,data period,season_unit,num_seasons)

- Returns baseline by averaging data periods in seasons
- Uses Weighted Moving Average algorithm (WMA)

- baselinewma(/host/key,1h:now/h,"d",3)
  - #calculating the baseline based on the last full hour within a 3-day period that ended yesterday. If "now" is Monday 13:30, the data for 12:00-12:59 on Friday, Saturday, and Sunday will be analyzed
- baselinewma(/host/key,2h:now/h,"d",3)
  - #calculating the baseline based on the last two hours within a 3-day period that ended yesterday. If "now" is Monday 13:30, the data for 11:00-12:59 on Friday, Saturday, and Sunday will be analyzed
- baselinewma(/host/key,1d:now/d,"M",4)
  - #calculating the baseline based on the same day of month as 'yesterday' in the 4 months preceding the last full month. If the required date doesn't exist, the last day of month is taken. If today is September 1st, the data for July 31st, June 30th, May 31st, April 30th will be analyzed.

baselinedev

baselinedev(/host/key,data period:time shift,season unit,num seasons)

- Returns the number of deviations (by stddevpop algorithm) between the last data period and the same data periods in preceding seasons.

- baselinedev(/host/key,1d:now/d,"M",6)
  - #calculating the number of standard deviations (population) between the previous day and the same day in the previous 6 months. If the date doesn't exist in a previous month, the last day of the month will be used (Jul,31 will be analysed against Jan,31, Feb, 28,... June, 30)
- baselinedev(/host/key,1h:now/h,"d",10)
  - #calculating the number of standard deviations (population) between the previous hour and the same hours over the period of ten days before yesterday

Trend functions

baselinedev	Returns the number of deviations (by stddevpop algorithm) between the last data period and the same data periods in preceding seasons.
baselinewma	Calculates the baseline by averaging data from the same timeframe in multiple equal time periods ('seasons') using the weighted moving average algorithm.
trendavg	The average of trend values within the defined time period.
trendcount	The number of successfully retrieved trend values within the defined time period.
trendmax	The maximum in trend values within the defined time period.
trendmin	The minimum in trend values within the defined time period.
trendstl	Returns the rate of anomalies during the detection period - a decimal value between 0 and 1 that is ((the number of anomaly values)/(total number of values)).
trendsum	The sum of trend values within the defined time period.

13

Operator functions

Operator functions

› between	Check if the value belongs to the given range.
› in	Check if the value is equal to at least one of the listed values.

between(value,min,max)

- Check if the value belongs to the given range.
- Supported value types: Integer, Float.
- Returns: 1 - in range; 0 - otherwise.

in(value,value1,value2,...valueN)

- Check if the value is equal to at least one of the listed values.
- Supported value types: Integer, Float, Character, Text, Log.
- Returns: 1 - if equal; 0 - otherwise.

14

Prediction functions

Prediction functions

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

- The future value, max, min, delta or avg of the item.
- Supported value types: Float, Integer.
- forecast(/host/key,#10,1h)   #forecast the item value in one hour based on the last 10 values

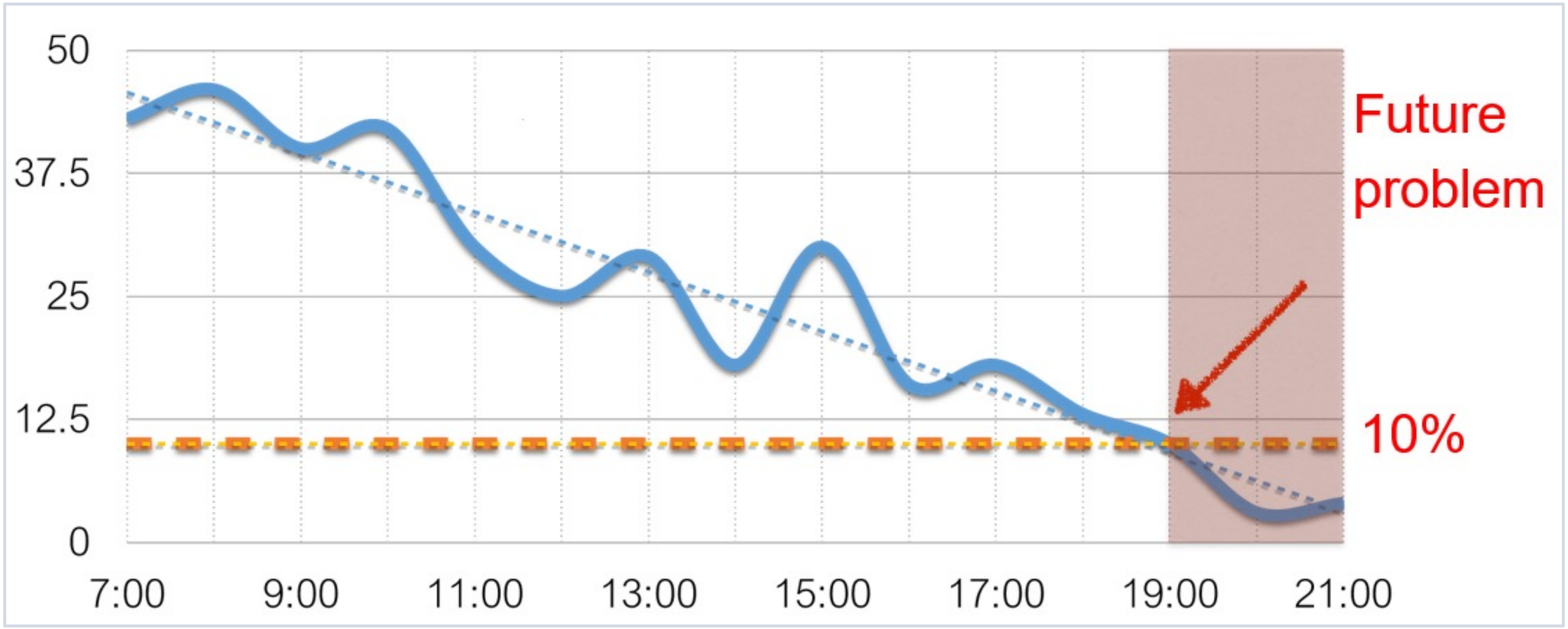
timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

- The time in seconds needed for an item to reach the specified threshold.
- Supported value types: Float, Integer.
- timeleft(/host/key,#10,0)   #the time until the item value reaches zero based on the last 10 values

Prediction functions

- fit (optional; must be double-quoted) - the function used to fit historical data. Supported fits:
  - linear - linear function (default)
  - polynomialN - polynomial of degree N (1 <= N <= 6)
  - exponential - exponential function
  - logarithmic - logarithmic function
  - power - power function
  - Note that polynomial1 is equivalent to linear;
- mode (optional; must be double-quoted) - the demanded output. Supported modes:
  - value - value (default)
  - max - maximum
  - min - minimum
  - delta - max-min
  - avg - average

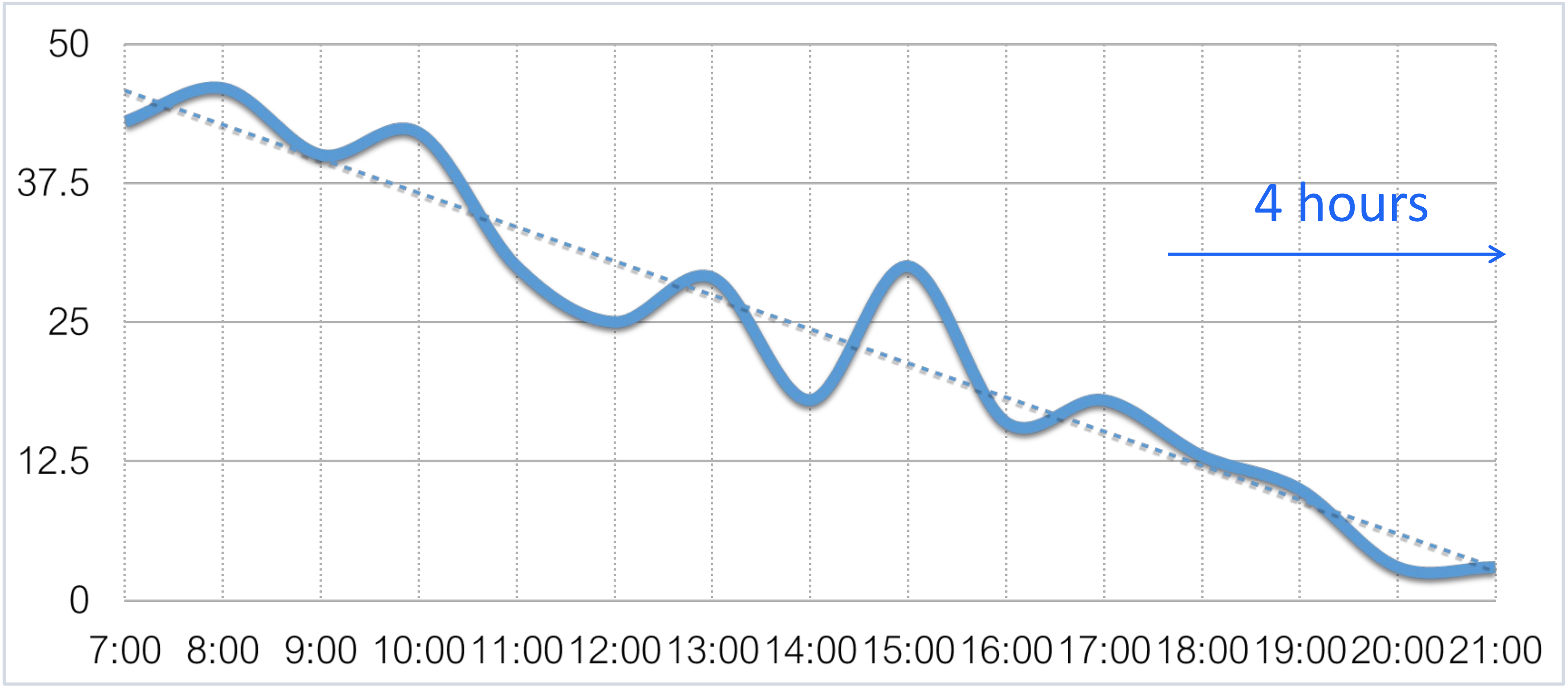
Forecast



Trigger function timeleft

ADVANCED PROBLEM DETECTION

Forecast

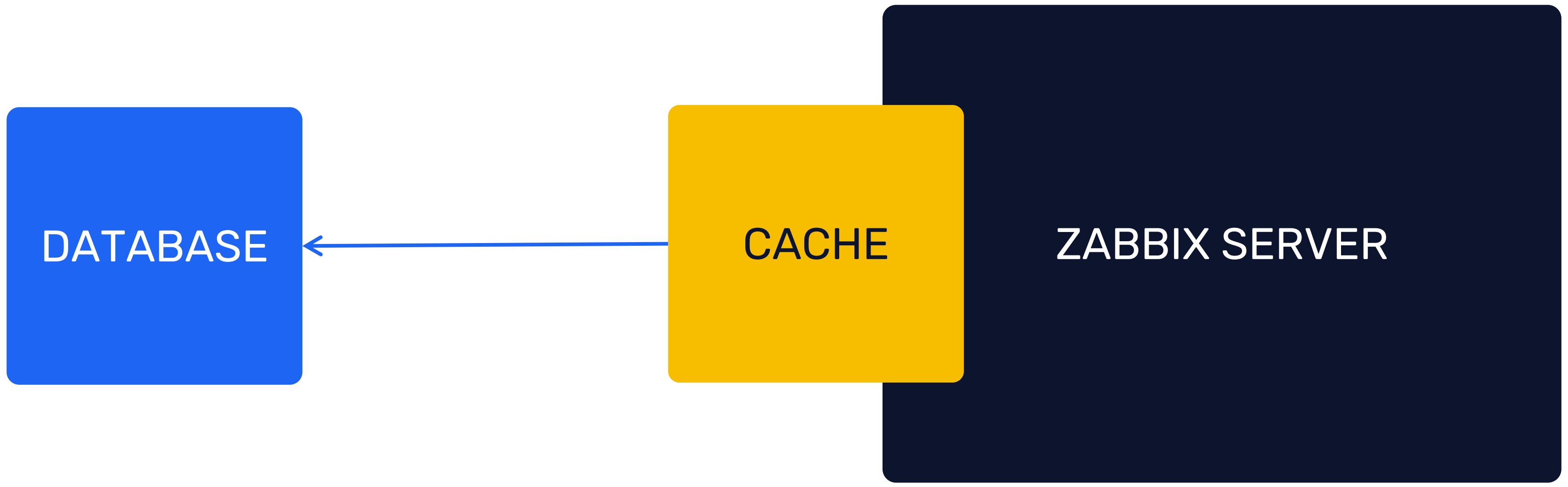


Trigger function forecast

Does history analysis affect performance of Zabbix?

Yes, but not significantly.

Especially as of Zabbix 2.2.0.



15

String functions

String functions

ascii	The ASCII code of the leftmost character of the value.
bitlength	The length of value in bits.
bytelength	The length of value in bytes.
char	Return the character by interpreting the value as ASCII code.
concat	The string resulting from concatenating the referenced item values or constant values.
insert	Insert specified characters or spaces into the character string beginning at the specified position in the string.
left	Return the leftmost characters of the value.
length	The length of value in characters.
ltrim	Remove specified characters from the beginning of string.
mid	Return a substring of N characters beginning at the character position specified by 'start'.
repeat	Repeat a string.
replace	Find the pattern in the value and replace with replacement.
right	Return the rightmost characters of the value.
rtrim	Remove specified characters from the end of string.
trim	Remove specified characters from the beginning and end of string.

16

Zabbix 7.0

ADVANCED PROBLEM DETECTION

Zabbix 7.0

jsonpath(value,path,<default>)

Return the JSONPath result.

Supported value types: String, Text, Log.

- jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")

xmlxpath(value,path,<default>)

Return the XML XPath result.

Supported value types: String, Text, Log.

- xmlxpath(last(/host/xml_result),"/response/error/status")

Zabbix 7.0

Updated functions

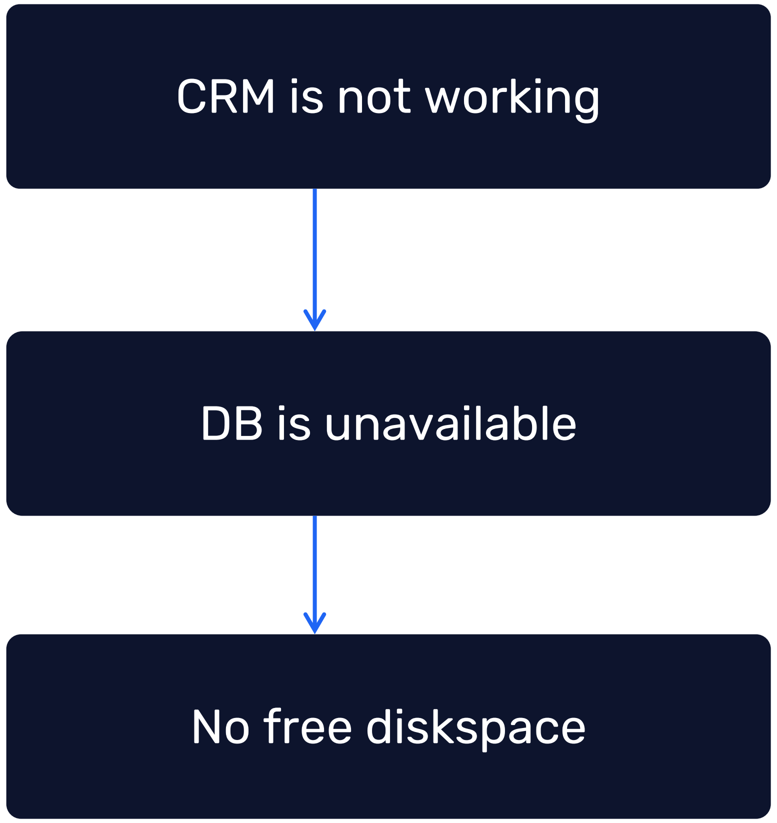
- Aggregate functions now also support non-numeric types for calculation. This may be useful, for example, with the count and count_foreach functions.
- The count and count_foreach aggregate functions support optional parameters operator and pattern, which can be used to fine-tune item filtering and only count values that match given criteria.
- All foreach functions no longer include unsupported items in the count.
- The function last_foreach, previously configured to ignore the time period argument, accepts it as an optional parameter.
- Supported range for values returned by prediction functions has been expanded to match the range of double data type. Now timeleft() function can accept values up to 1.7976931348623158E+308 and forecast() function can accept values ranging from -1.7976931348623158E+308 to 1.7976931348623158E+308.

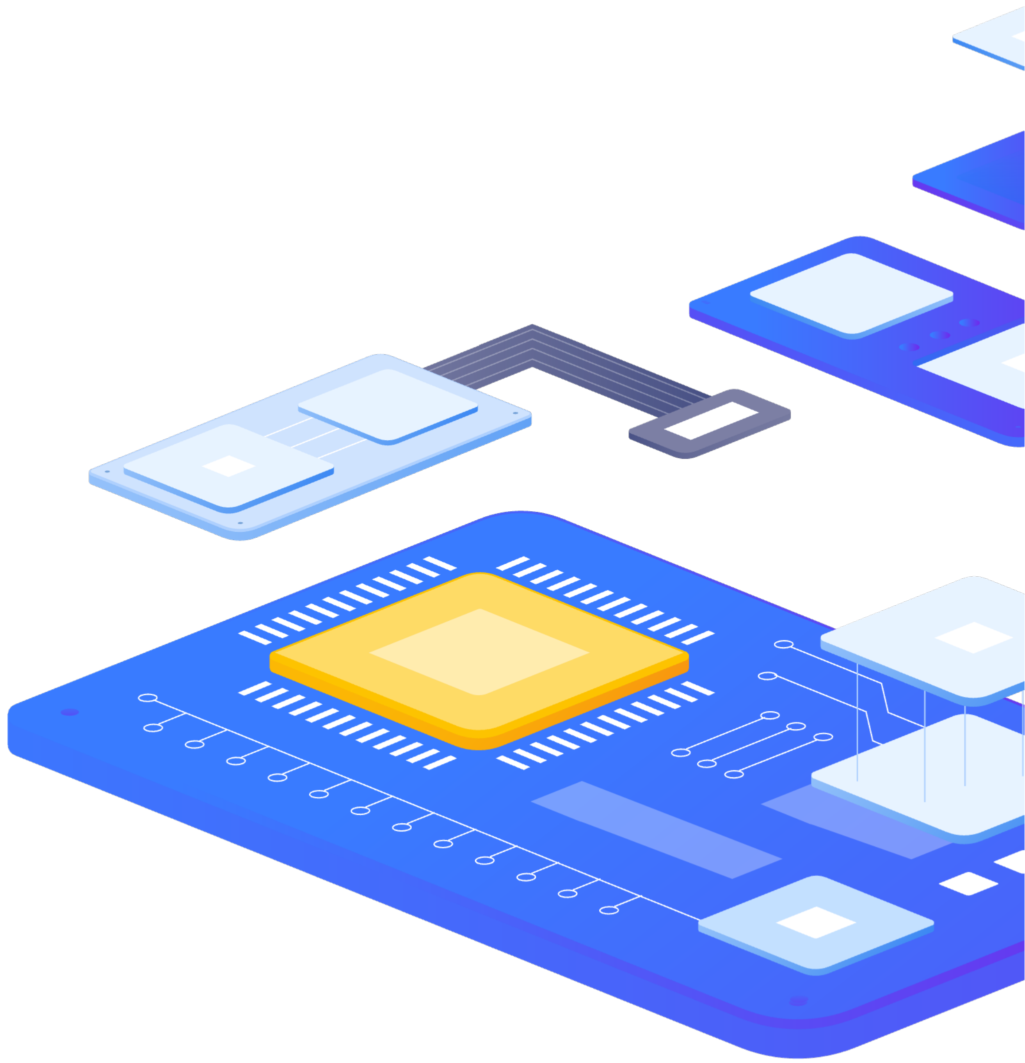
17

Dependencies

ADVANCED PROBLEM DETECTION

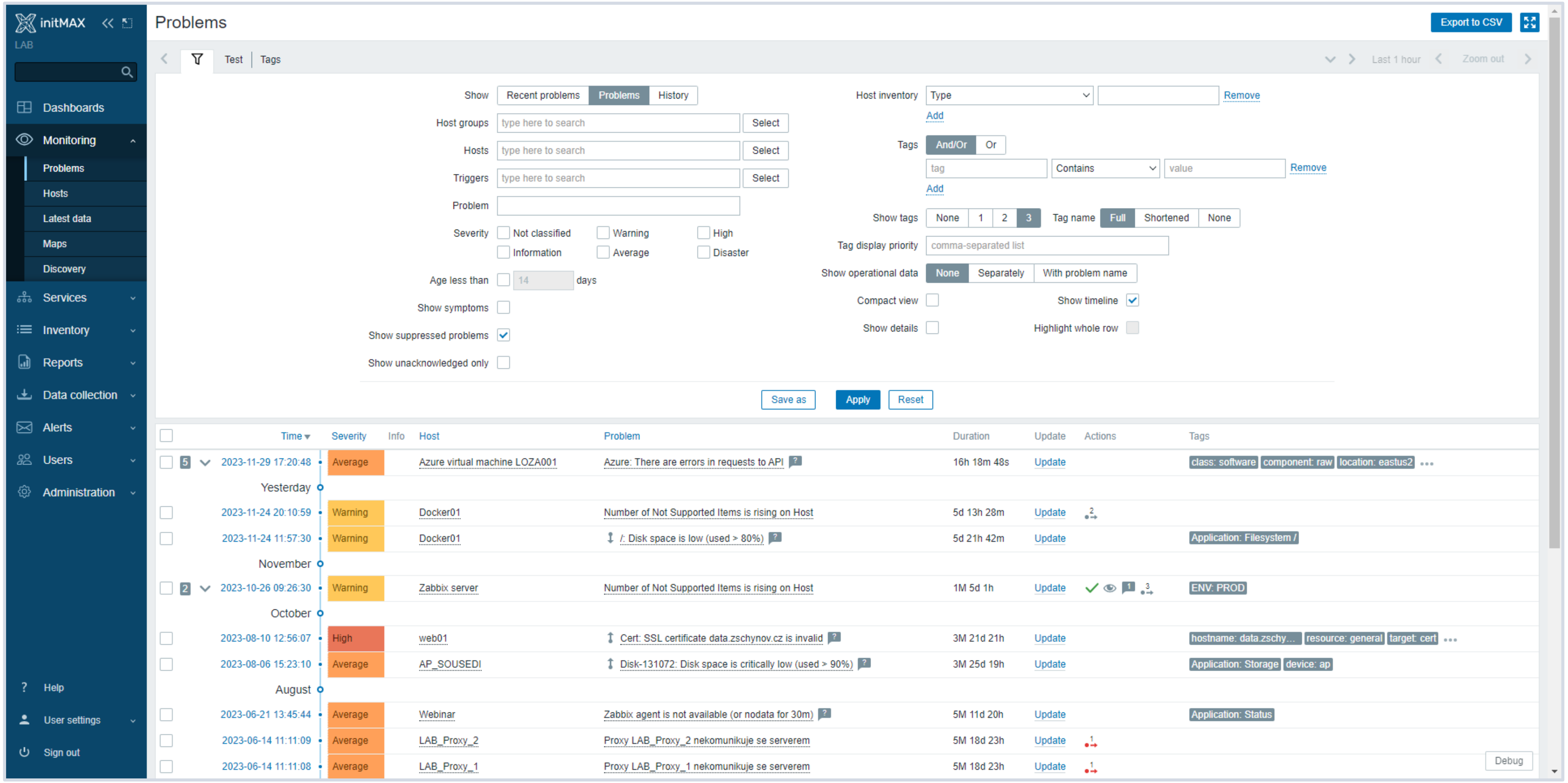
Dependencies





ADVANCED PROBLEM DETECTION

Section „Problems“



Problems

Show: Recent problems [✓ Problems] History

Host groups: type here to search — Select

Hosts: type here to search — Select

Triggers: type here to search — Select

Problem:

Severity: Not classified, Warning, High, Information, Average, Disaster

Age less than: 14 days

Show symptoms: ☐

Show suppressed problems: ✓

Show unacknowledged only: ☐

Host inventory: Type — Remove — Add

Tags: [✓ And/Or] Or — tag — Contains — value — Remove — Add

Show tags: None 1 2 [✓ 3]   Tag name: [✓ Full] Shortened None

Tag display priority: comma-separated list

Show operational data: [✓ None] Separately With problem name

Compact view: ☐   Show timeline: ✓

Show details: ☐   Highlight whole row: ☐

Time ▾	Severity	Info	Host	Problem	Duration	Update	Actions	Tags
5 2023-11-29 17:20:48	Average		Azure virtual machine LOZA001	Azure: There are errors in requests to API	16h 18m 48s	Update		class: software, component: raw, location: eastus2 ...
Yesterday								
2023-11-24 20:10:59	Warning		Docker01	Number of Not Supported Items is rising on Host	5d 13h 28m	Update	2	
2023-11-24 11:57:30	Warning		Docker01	/: Disk space is low (used > 80%)	5d 21h 42m	Update		Application: Filesystem /
November								
2 2023-10-26 09:26:30	Warning		Zabbix server	Number of Not Supported Items is rising on Host	1M 5d 1h	Update	✓ 1 3	ENV: PROD
October								
2023-08-10 12:56:07	High		web01	Cert: SSL certificate data.zschynov.cz is invalid	3M 21d 21h	Update		hostname: data.zschy..., resource: general, target: cert ...
2023-08-06 15:23:10	Average		AP_SOUSEDI	Disk-131072: Disk space is critically low (used > 90%)	3M 25d 19h	Update		Application: Storage, device: ap
August								
2023-06-21 13:45:44	Average		Webinar	Zabbix agent is not available (or nodata for 30m)	5M 11d 20h	Update		Application: Status
2023-06-14 11:11:09	Average		LAB_Proxy_2	Proxy LAB_Proxy_2 nekomunikuje se serverem	5M 18d 23h	Update	1	
2023-06-14 11:11:08	Average		LAB_Proxy_1	Proxy LAB_Proxy_1 nekomunikuje se serverem	5M 18d 23h	Update	1	

In summary

- Analyze history
- No problem!= Solution
- Use different conditions for problem definition and recovery
- Pay attention to anomaly detection
- Use correlation
- Resolve common problems automatically

18

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