

ZABBIX

6.0

BASELINE MONITORING AND ANOMALY DETECTION



- all our microphones are muted
- ask your questions in Q&A, not in the Chat
- use Chat for discussion, networking or applause
- use the hashtag if you post something in your social media channels: #ZabbixMeetingOnline

ZABBIX 6.0 LTS

One of the new features of Zabbix 6.0 LTS is focus on **anomaly detection**

Zabbix 6.0 offers:

- ✓ Possibility to use **trends** to analyze large periods of data
- ✓ Baseline calculation using **baselinewma** and **baselinedev** functions
- ✓ Anomalous metric detection using **trendstl** function

01

ZABBIX
6.0

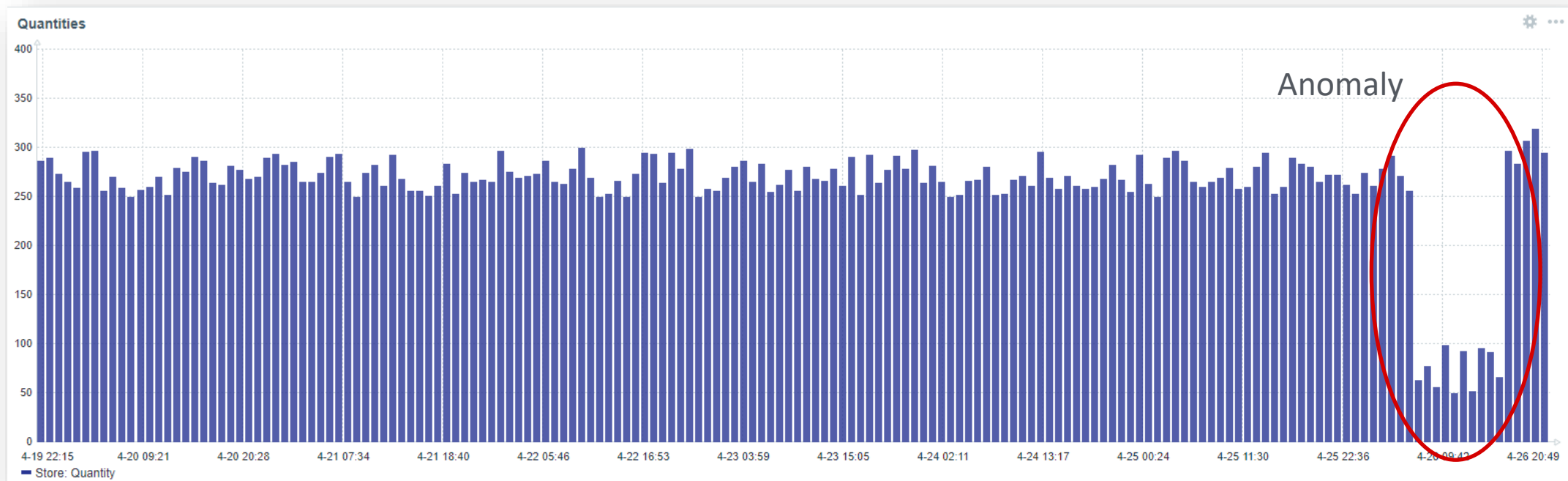
BASELINE MONITORING OVERVIEW



BASELINE AND ANOMALIES

Anomaly detection is a type of data analytics whose goal is detecting unusual patterns in a dataset:

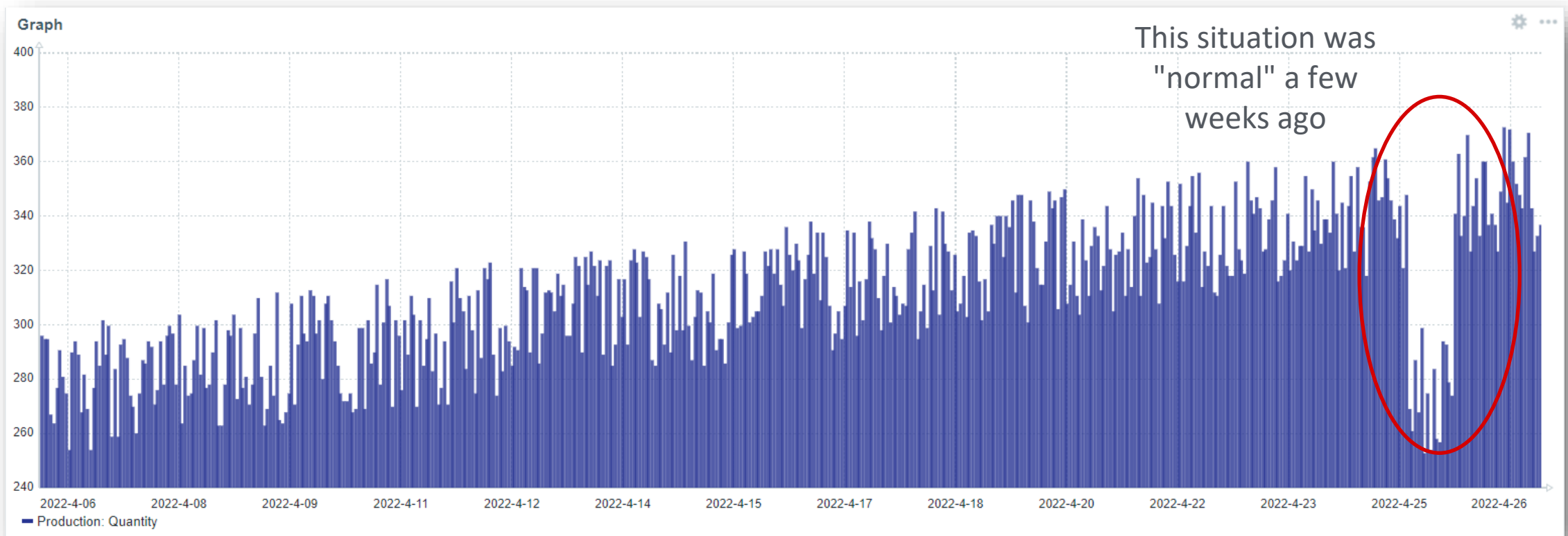
- ✓ Data must be **normally distributed**, following a set of rules
- ✓ Anomaly detection measures how far a data point is away from the mean
- ✓ When the value deviates too much from the mean, it is considered to be anomalous



BASELINE VS FIXED TRIGGERS

BASELINE MONITORING CAN MONITOR VALUES WHICH FOLLOW A **PATTERN**

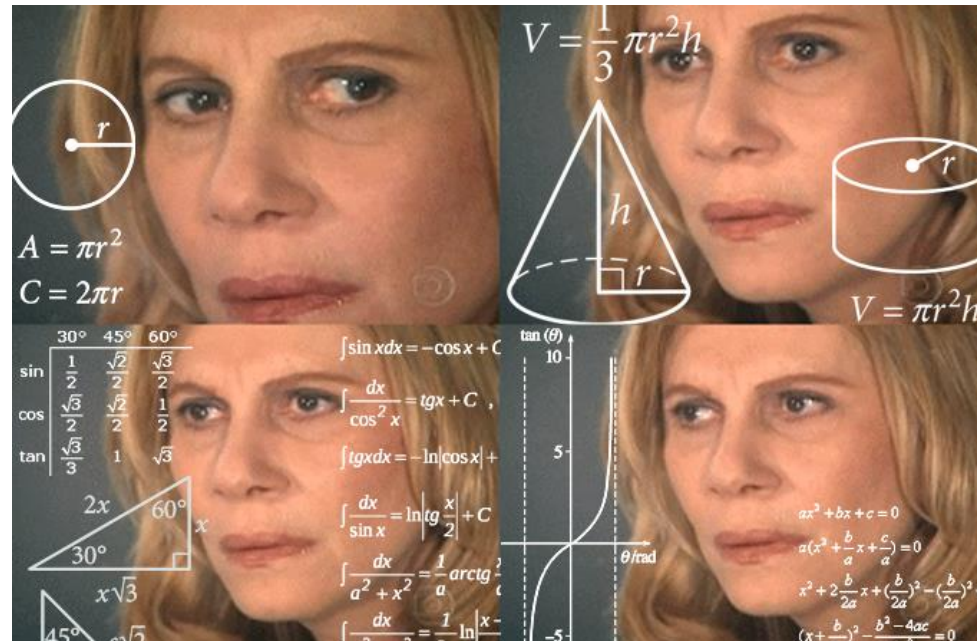
- ✓ Fixed trigger thresholds will either give false alarms or ignore the problem
- ✓ Baseline monitoring can adapt to such situation



ANOMALY DETECTION OVERVIEW

Anomaly detection is based on a set of statistic functions using:

- ✓ Standard deviation (σ)
- ✓ Mean absolute deviation (MAD)
- ✓ Weighted moving average (WMA)
- ✓ Seasonal and Trend decomposition using Loess (STL)

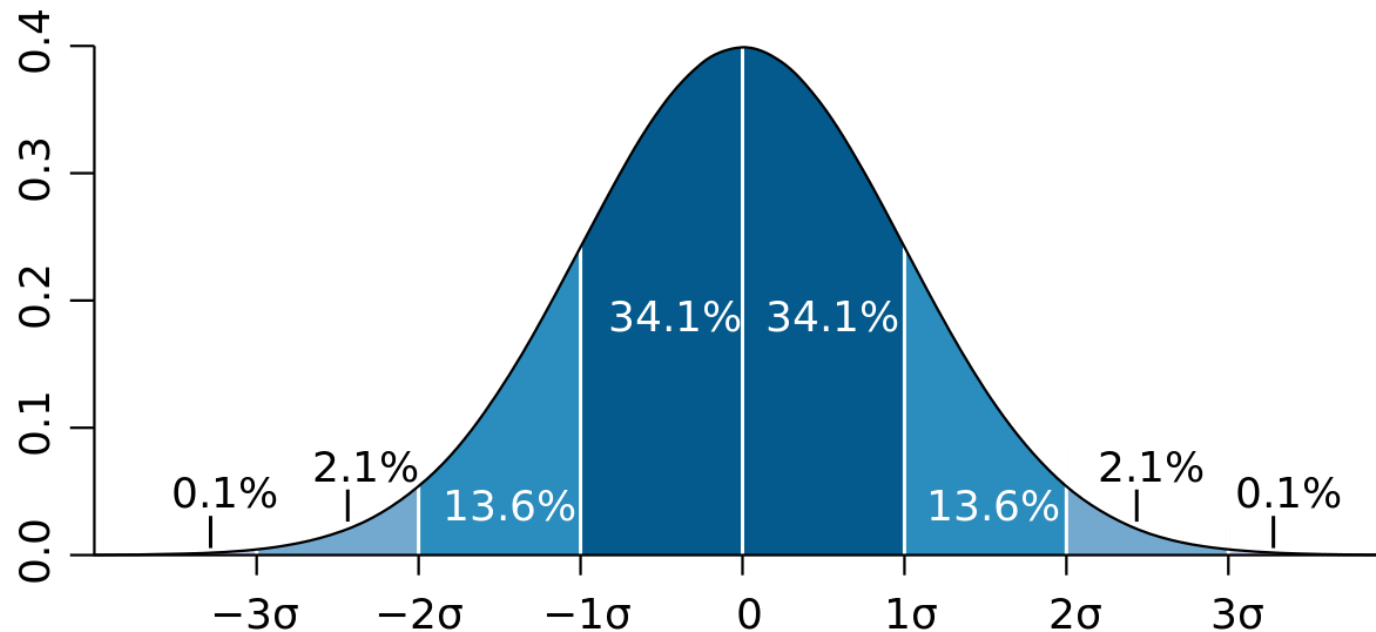


STANDARD DEVIATION

The **standard deviation** σ defines how far the normal distribution is spread around the mean

☑ When a metric is normally distributed it follows some [interesting laws](#):

- 68% of all values fall between $[\text{mean}-\sigma, \text{mean}+\sigma]$
- 95% of all values fall between $[\text{mean}-2*\sigma, \text{mean}+2*\sigma]$
- 99,7% of all values fall between $[\text{mean}-3*\sigma, \text{mean}+3*\sigma]$



MEAN ABSOLUTE DEVIATION (MAD)

The **mean absolute deviation** (MAD) of a dataset is the average distance between each data point and the mean

- ✓ Calculate the mean
- ✓ Calculate how far away each data point is from the mean using positive distances (deviations)
- ✓ Sum those deviations together
- ✓ Divide the sum by the number of data points

$$\text{MAD} = \frac{\sum |x_i - \bar{x}|}{n}$$

WEIGHTED MOVING AVERAGE ALGORITHM

A **weighted moving average** (WMA) puts more weight on recent data and less on past data

- ✓ The most recent data is more heavily weighted, and contributes more to the final WMA value
- ✓ The weighting factor used to calculate the WMA is determined by the period

For example, a 5 period WMA would be calculated as follows:

$$\text{WMA} = (P1 * 5) + (P2 * 4) + (P3 * 3) + (P4 * 2) + (P5 * 1) / (5 + 4 + 3 + 2 + 1)$$

$$\text{WMA} = \frac{\text{Price}_1 \times n + \text{Price}_2 \times (n - 1) + \dots + \text{Price}_n}{\frac{n \times (n + 1)}{2}}$$

02

TIMESHIFTS

ZABBIX

6.0



TIMESHIFT SYNTAX

Zabbix can use absolute or relative timeshift to compare current and past periods of data

- ☑ **Relative timeshift** specifies time period relatively to the current time:

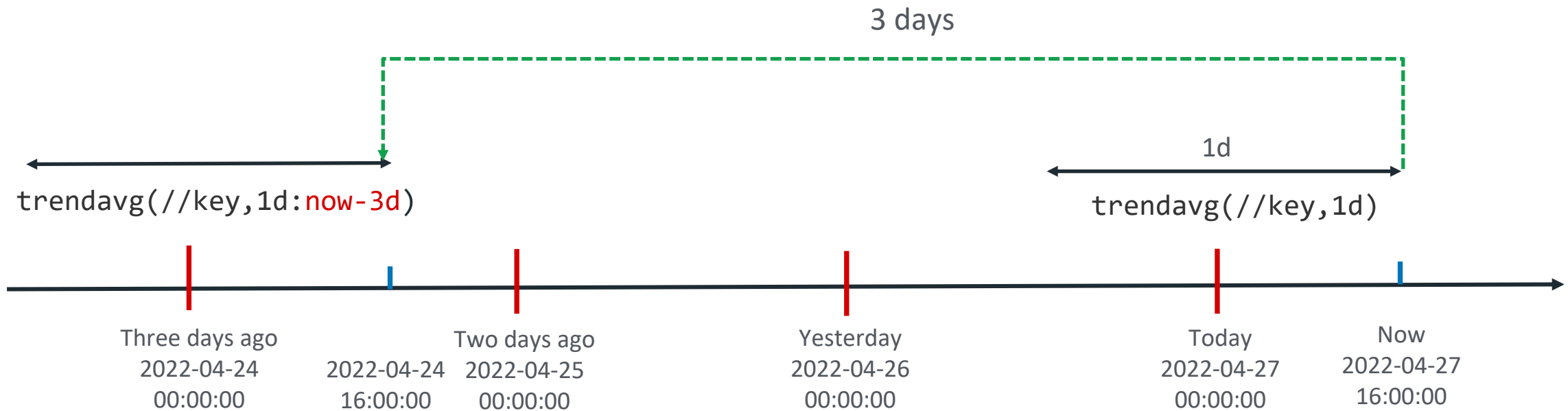
```
trendavg(/host/key,1d:now-3d)
```

- ☑ **Absolute timeshift** specifies the time period for analysis:

```
trendavg(/host/key,1d:now/d-3d)
```

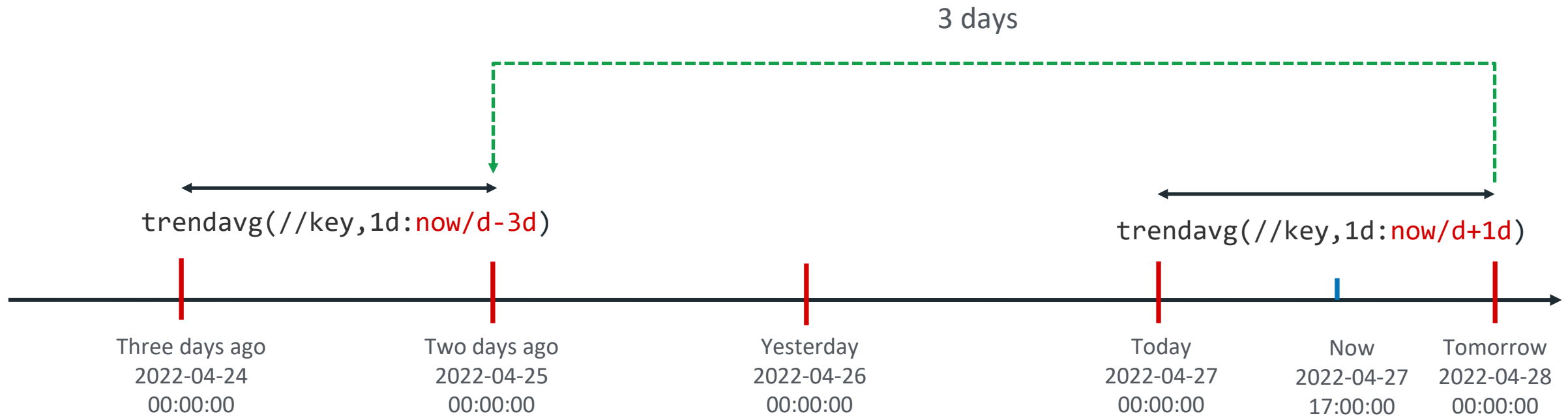
RELATIVE TIMESHIFT

Relative timeshift specifies sliding time period relatively to the current time:



ABSOLUTE TIMESHIFT

Absolute timeshift specifies fixed time period calculated from the end of the period:



03

TREND FUNCTIONS

ZABBIX

6.0



TREND FUNCTIONS

Zabbix 6.0 offers eight different trend functions for long-term data analysis

☑ Trends analysis:

- trend**sum** (/host/key,time period:time shift)
- trend**avg** (/host/key,time period:time shift)
- trend**count** (/host/key,time period:time shift)
- trend**max** (/host/key,time period:time shift)
- trend**min** (/host/key,time period:time shift)
- trend**stl** (/host/key,eval period:time shift,detection period,season,<dev>,<devalg>,<s_window>)

☑ Baseline calculation:

- baseline**dev** (/host/key,data period:time shift,season_unit,num_seasons)
- baseline**wma** (/host/key,data period:time shift,season_unit,num_seasons)

STORING TRENDS

Trends in the trends cache are calculated in real-time independently from historical data

- ✓ TrendCache always has the actual trends value for every item
 - to calculate the new average after then nth number, you multiply the old average by n-1, add the new number, and divide the total by n.

$$\mu_n = \frac{(n - 1)\mu_{n-1} + x_n}{n}$$

- ✓ It is possible to store only trends data

* History storage period

Do not keep history

Storage period

* Trend storage period

Do not keep trends

Storage period

720d

TREND CACHES

Zabbix trends are written to database at the beginning of each hour

- ☑ Trends for the **current hour are unavailable** for trend functions
- ☑ Two different Zabbix server internal caches are used by trends

```
### Option: TrendCacheSize
#      Size of trend write cache, in bytes. Shared memory size for storing trends data.
# Range: 128K-2G
# Default:
TrendCacheSize=512M
```

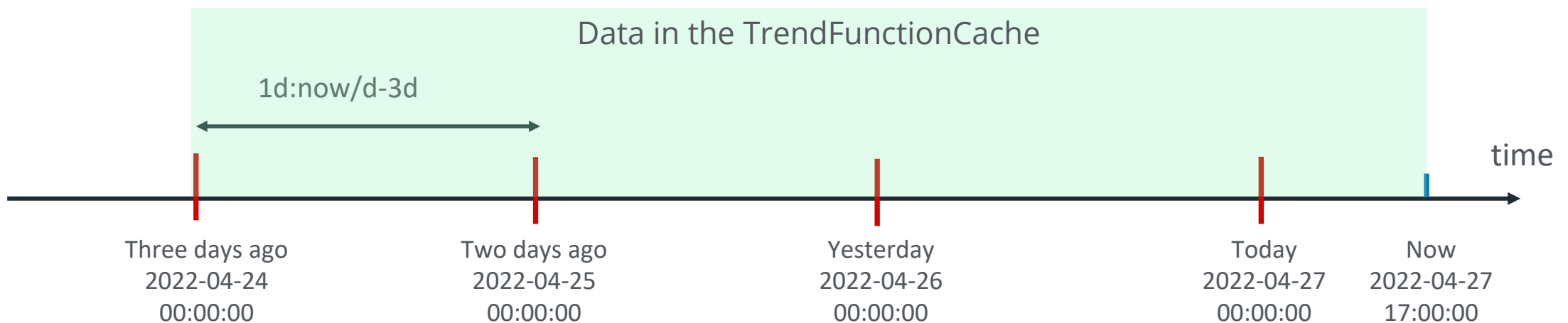
```
### Option: TrendFunctionCacheSize
#      Size of trend function cache, in bytes.
#      Shared memory size for caching calculated trend function data.
# Range: 128K-2G
# Default:
TrendFunctionCacheSize=256M
```

TRENDS VS HISTORY

TRENDS UTILIZES LESS SPACE THAN HISTORY IN BOTH DATABASE AND MEMORY CACHES

- ✓ When functions with time-shift are used for analysis, all data between now and function period are stored in the memory cache
- ✓ History data could utilize gigabytes of data in such scenarios, trends are much more efficient

`trendsum(//key,1d:now/d-3d)`



04

BASELINE MONITORING

ZABBIX

6.0



BASELINE MONITORING OVERVIEW

Baseline monitoring can be used to analyze recent data by:

- ✓ Comparing it to baseline from previous periods using the (baselinewma)
- ✓ Calculating the number of deviations from previous periods (baselinedev)

Previous data periods must be defined as **seasons** using:

- ✓ Season units (h, d, w, M, y) - cannot be smaller than data period
- ✓ Number of seasons

baseline*/(/host/key,1h:now/h,"d",3)

baseline function based on the last full hour within the last 3-day period

MORE BASELINE FUNCTION EXAMPLES

baseline*/(/host/key,1d:now/d,"M",6)

baseline function based on the previous day and the same day of month in the previous 6 months
If the date does not exist in a previous month, last day of month will be used

baseline*/(/host/key,2h:now/h,"d",7)

baseline function based on the last two hours and the same hours within a 7-day period

baseline*/(/host/key,1w:now/w,"m",3)

baseline function based on the previous week and other weeks within a 3-month period

BASELINEWMA() FUNCTION

Calculates baseline data by **averaging data** from the same timeframe in multiple equal time periods

- ✓ Weighted moving average algorithm (WMA) is used
- ✓ Baseline can be compared to recent trends data to detect anomalies

baselnewma (/host/key,data period:time shift,season_unit,num_seasons)

trendavg (/Web/nginx.req,1d:now/d) > baselnewma (/Web/nginx.req,1d,4w) * 2

Web requests yesterday (1d:now/d) is more than **twice** as high than baseline on the same weekdays(1d) over last 4 weeks

BASELINEDEV() FUNCTION

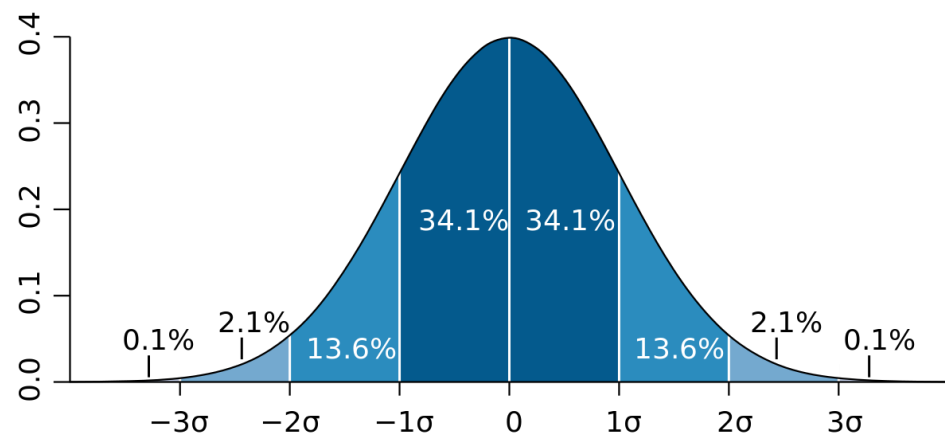
Calculates **number of deviations** (σ) between the last data period and periods in preceding seasons

- ✓ stddevpop algorithm is used (calculates standard deviation based on the entire population)
- ✓ High number of deviations indicates anomalies

baselinedev (/host/key,data period:time shift,season_unit,num_seasons)

baselinedev(/Production server/system.cpu.load,1h,10d,10) > 3

Check if load for last hour is more than **3 deviations** away from mean using **10 one-hour** periods over last **10 days**



05

ZABBIX

6.0

ANOMALY DETECTION USING PATTERNS



TRENDSTL FUNCTION

STL will decompose data in predefined intervals and will find anomalies based on a **repeating pattern**.

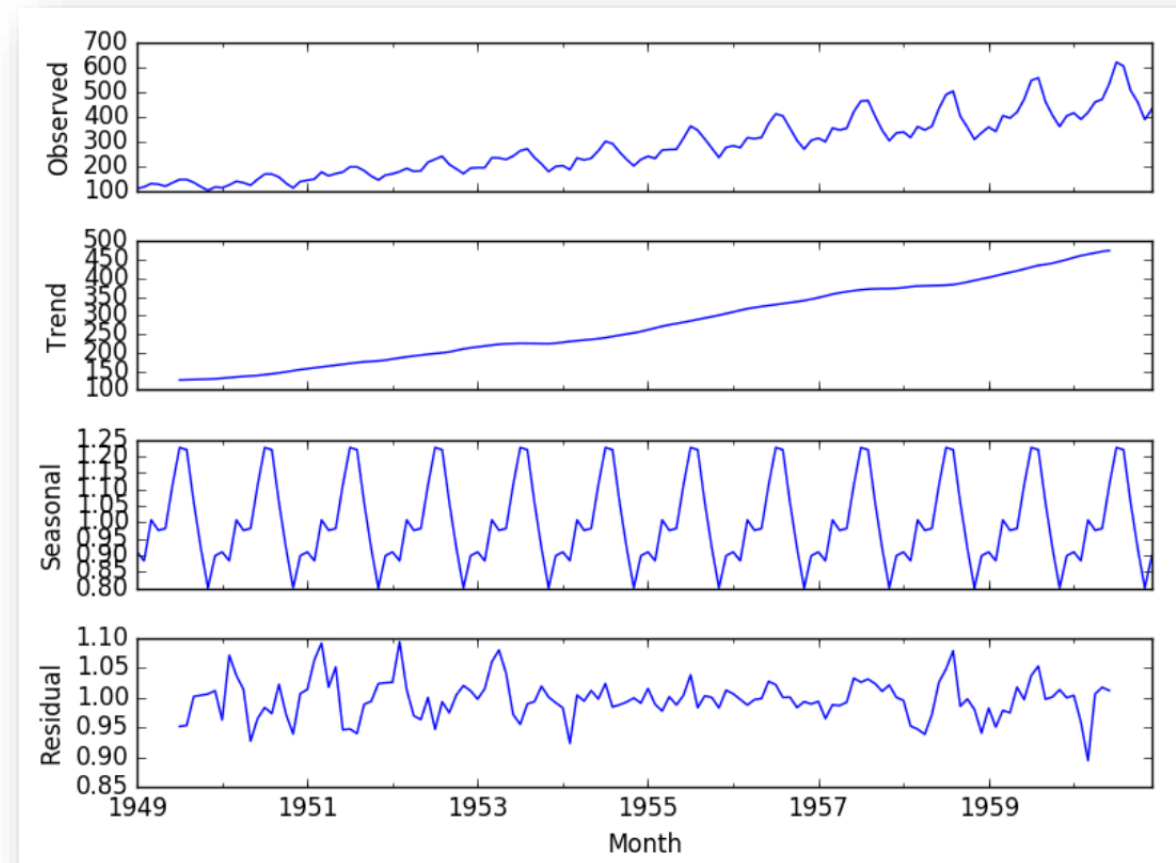
- ☑ trendstl() function use standard deviation to detect anomalies and returns **anomaly rate** (0 - 1):
 - Function compares smaller **detection period** to larger **evaluation period**
 - A standard deviation means how far values are from the average
 - By default, the MAD algorithm is used, can be also stddevpop or stddevsamp
 - The number of deviations can be specified (default is 3)
 - **s_window** is the span (in lags) of the loess window for seasonal extraction

trendstl(/host/key,**eval period**,**detection period**,season,<dev>,<devalg>,<s_window>)

SEASONAL-TREND DECOMPOSITION USING LOESS

Seasonal-Trend decomposition using LOESS (STL) is a robust method of time series decomposition

- ✓ The STL method uses locally fitted regression models to decompose a time series into
 - trend components
 - seasonal components
 - residual components



TRENDSTL FUNCTION EXAMPLE

```
trendstl(/host/net.if.out[eth0],30d:now/d,7d,12h) > 0.1
```

- ✓ Analyzing the last 30 days of trend data
- ✓ Find the anomalies rate for the previous 7days of that period
- ✓ Expecting the periodicity to be 12h (traffic varies between day and night time)
- ✓ The number of deviations to count as anomaly equals 3 (default)
- ✓ MAD algorithm is used (default)
- ✓ Anomaly rate is larger than 0.1 (10% of all values)

ZABBIX

6.0

Thank you

www.zabbix.com

