Calculated items

CALCULATED ITEMS

With calculated items, it is possible to create calculations based on the values of other items:

- Calculated items have their own update intervals
- Latest values of items will be used for calculation
- Calculated items have the same syntax as trigger expressions

Calculations may use both:

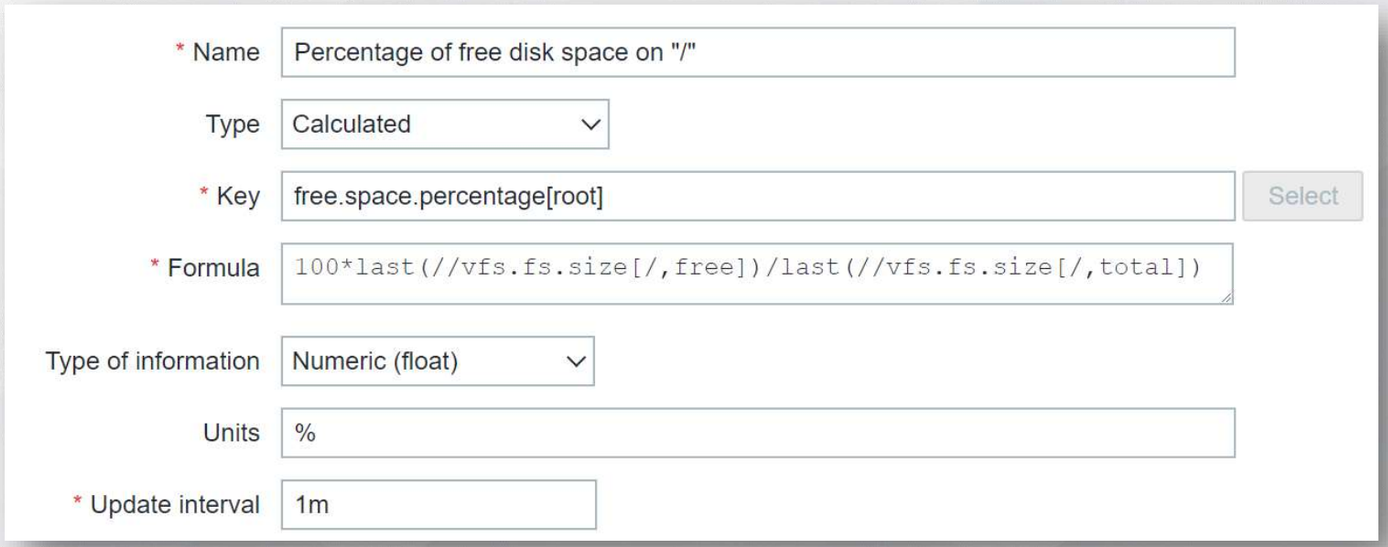
- Single values of individual items
- Complex filters to select multiple items for aggregations

Calculated checks are performed by History pollers:

```
### Option: StartHistoryPollers
#       Number of pre-forked instances of history pollers.
#       Only required for calculated and aggregated checks.
StartHistoryPollers=5
```

Calculated item configuration is different from other items

- The key is just a unique item identifier
- Calculation definition should be entered in the Formula field
- There is no connection between the formula and the key
- Numeric or character data types are supported as a result of calculation
- Calculated items can be tested and forcibly executed



CALCULATED ITEM FORMULA

The syntax of a simple calculated item formula is:

function(/host/key,<parameter>,<parameter>,...)

A host can be omitted if the current host is used

function(//key,<parameter>,<parameter>,...)

function	one of the supported functions (last, min, max, avg, count etc.)
host	host of the item used for calculation
key	any of the existing item keys (must match exactly!)
parameters	additional parameters for the function, if required

Example: avg(//net.if.in[eth0,errors],10m)

the item key must exist on the host !

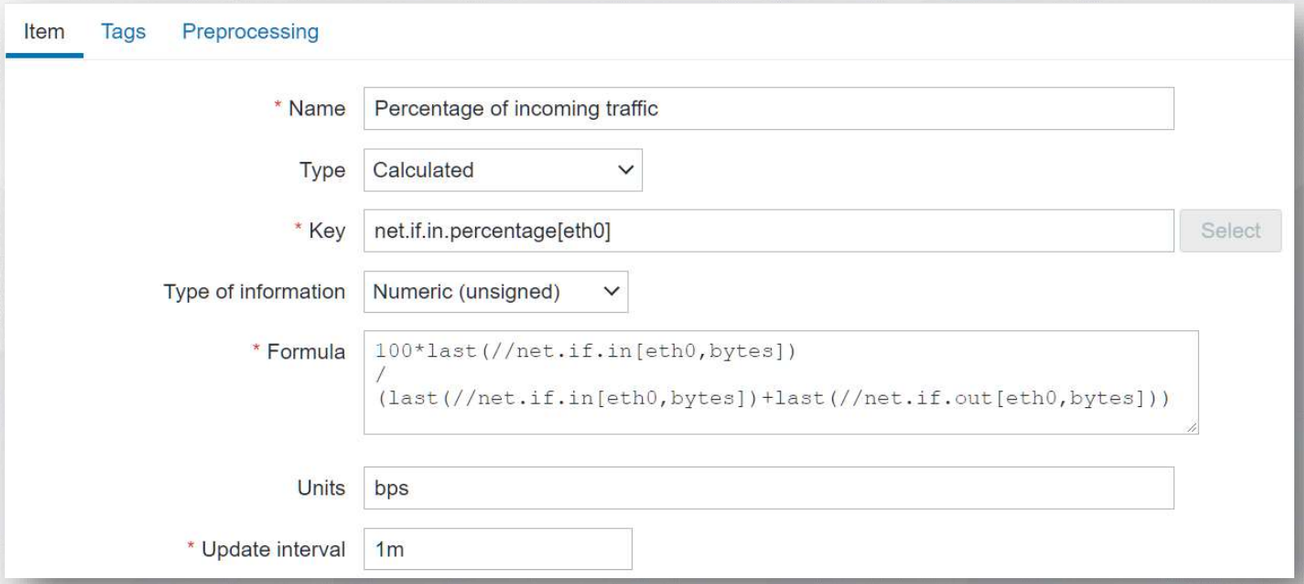
More complicated calculated items can be created:

- Result of multiple functions used for calculation

  last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])

- Complex expressions using mathematical operations

  100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))



Foreach function returns array of values from the history of multiple items:

- From multiple hosts from one or multiple host groups
- From multiple items with the same pattern in the item key

foreach_function(/host pattern/item pattern?[host group or tag filter])

The following foreach functions are supported

avg_foreach	returns average values
count_foreach	returns the number of values
exists_foreach	returns the number of currently enabled items
last_foreach	returns last values
max_foreach	returns maximum values
min_foreach	returns minimum values
sum_foreach	returns the sum of values

https://www.zabbix.com/documentation/6.0/en/manual/config/items/itemtypes/calculated/aggregate

AGGREGATING HOSTS AND ITEMS

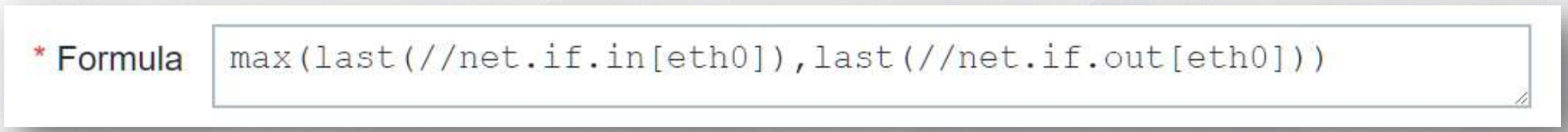
In aggregate calculations, information from several items may be collected:

- One of the aggregate functions must be used (avg, max, min, sum, etc.)

Aggregates can be retrieved by working with either:

- Aggregate function using historical data from items as parameters

  aggregate_function(function(/host1/item,parameter),function(/host2/item,parameter),...)



- Aggregate function using foreach function as the only parameter

  aggregate_function(foreach_function(/*/key?[group="host group"],timeperiod))

AGGREGATION EXAMPLES

Some aggregate examples:

- Total disk space of host group 'MySQL Servers'

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

- 5-minute average of the number of queries per second for host group 'MySQL Servers'

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

- Sum of incoming traffic matching net.if.in[*] on the current host

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

- Sum of incoming traffic matching net.if.in[*] from two host groups' matching tags

```
sum(last_foreach(/*/net.if.in[*])?[(group="Servers A" or group="Servers B") and tag="Service:MySQL"])
```

Only host groups and item tags can be used in aggregation filter