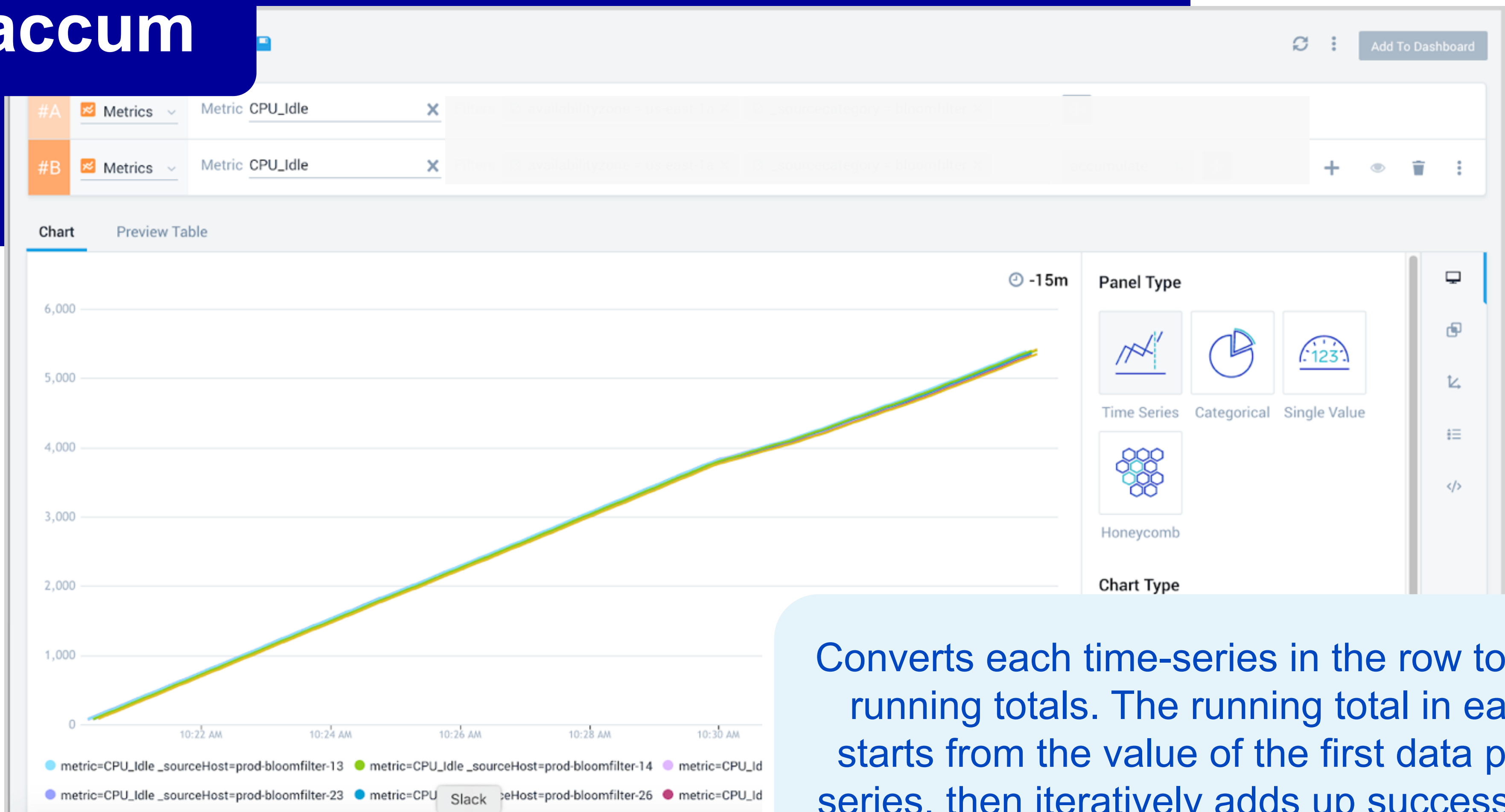


AGGREGATION

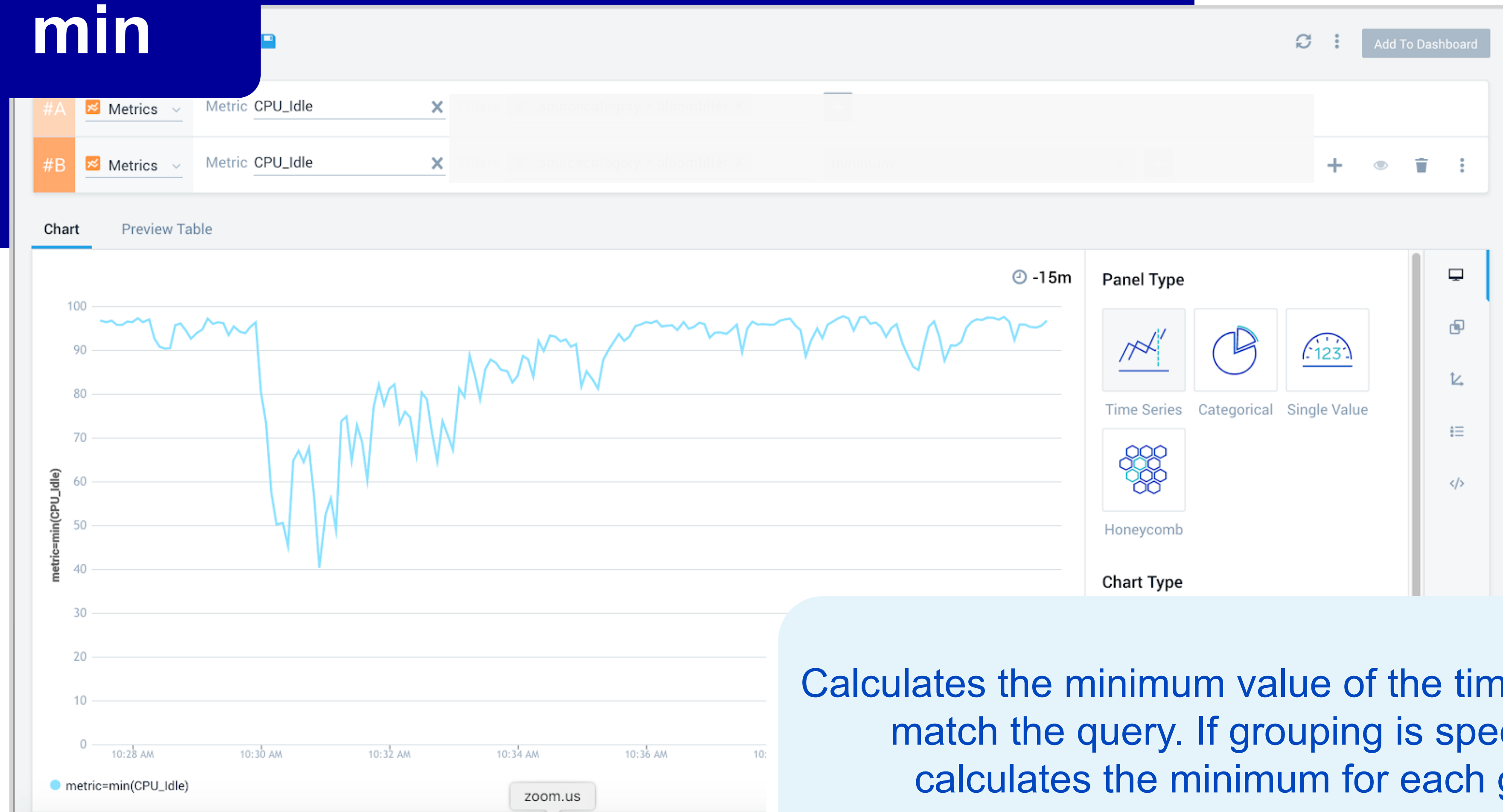
accum



Converts each time-series in the row to a series of running totals. The running total in each series starts from the value of the first data point in the series, then iteratively adds up successive values.

CPU\_Idle\_sourcecategory=bloomfilter | accum

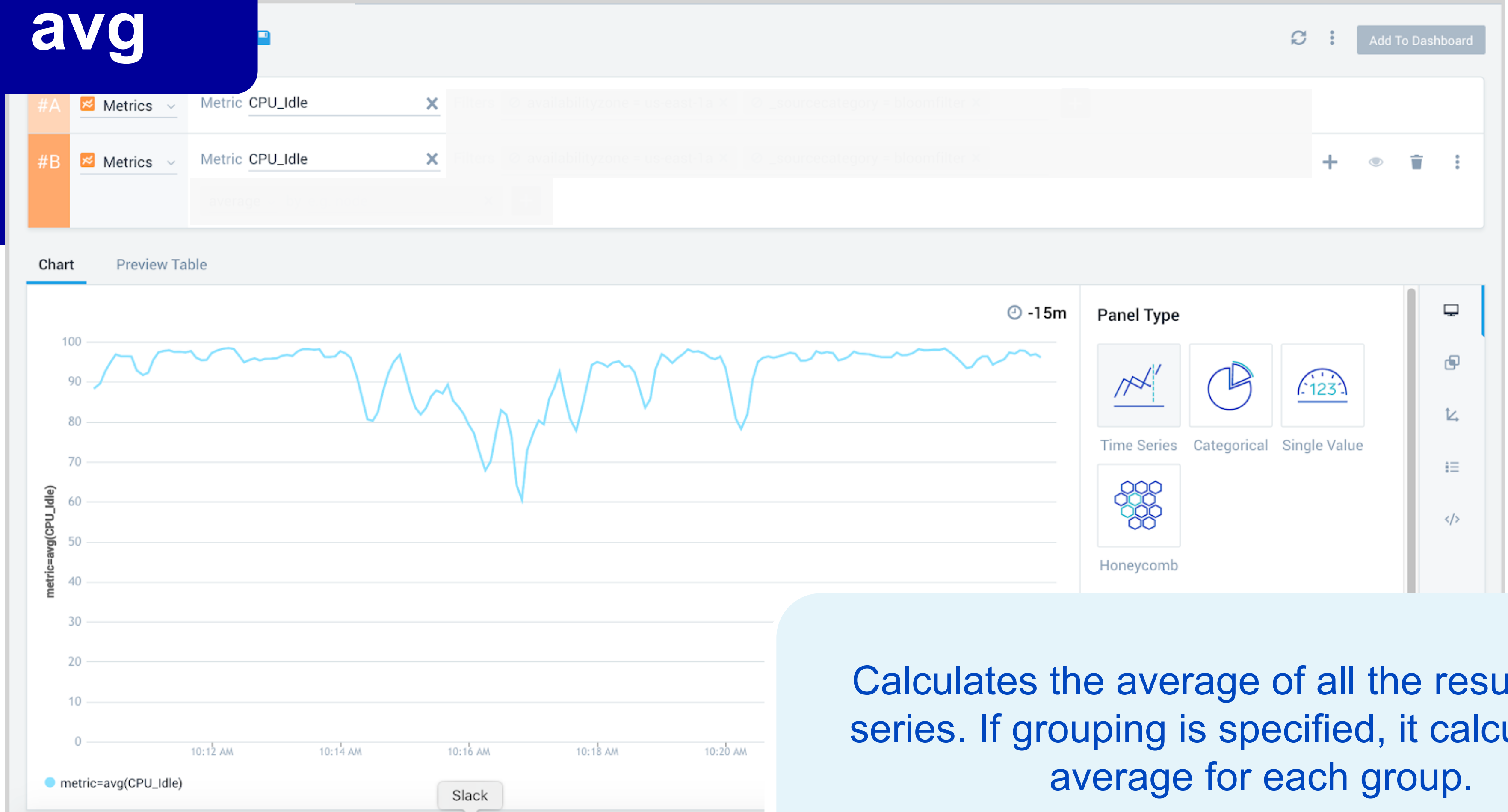
min



Calculates the minimum value of the time series that match the query. If grouping is specified, it calculates the minimum for each group.

CPU\_Idle\_sourcecategory=bloomfilter | min

avg



Calculates the average of all the resulting time series. If grouping is specified, it calculates the average for each group.

CPU\_Idle\_sourcecategory=bloomfilter | avg

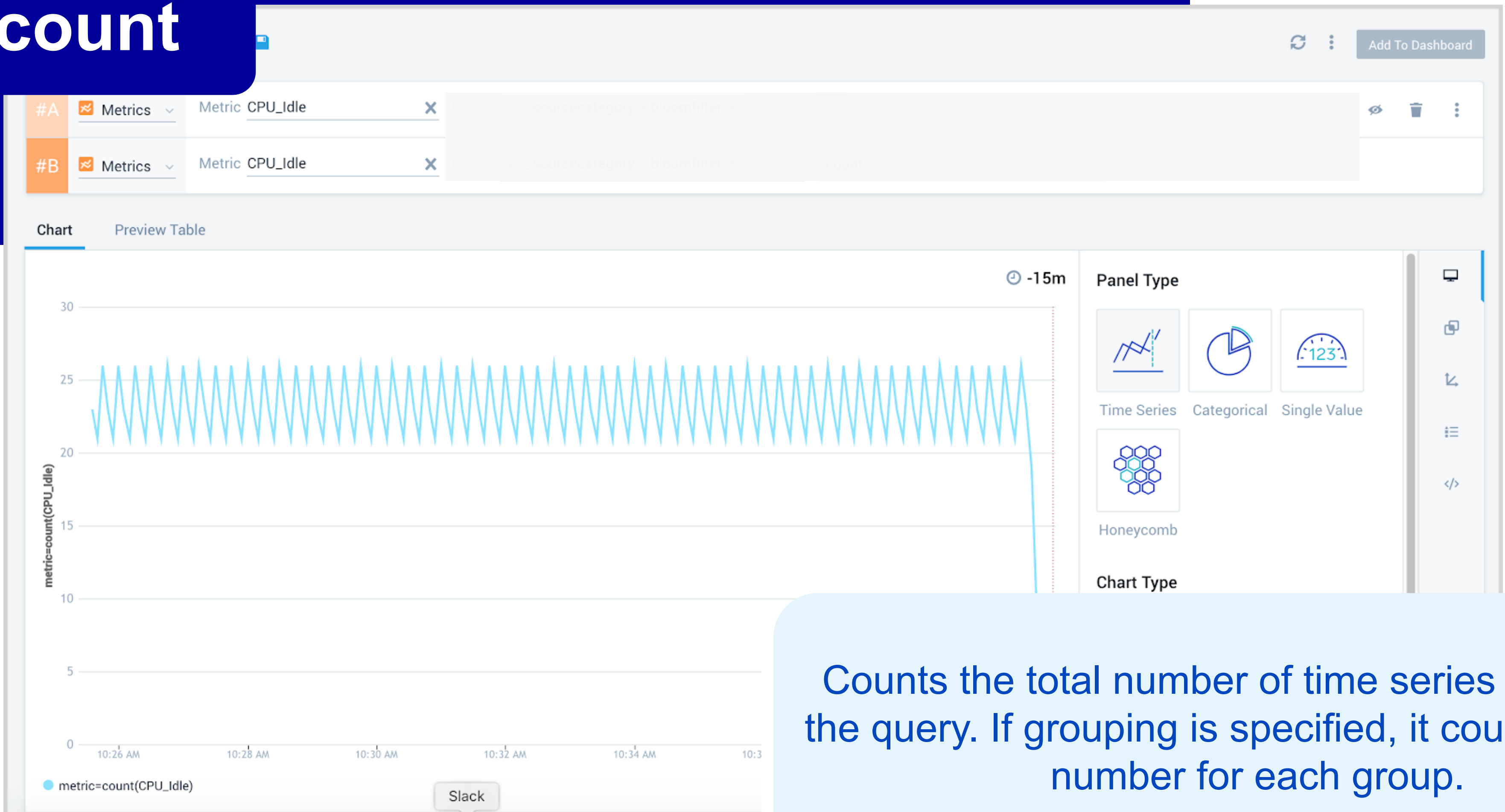
pct



Calculates the specified percentile of the metrics that match the query. If grouping is specified, it calculates the specified percentile for each group.

CPU\_IOWait\_sourcecategory=analytics | pct(50)

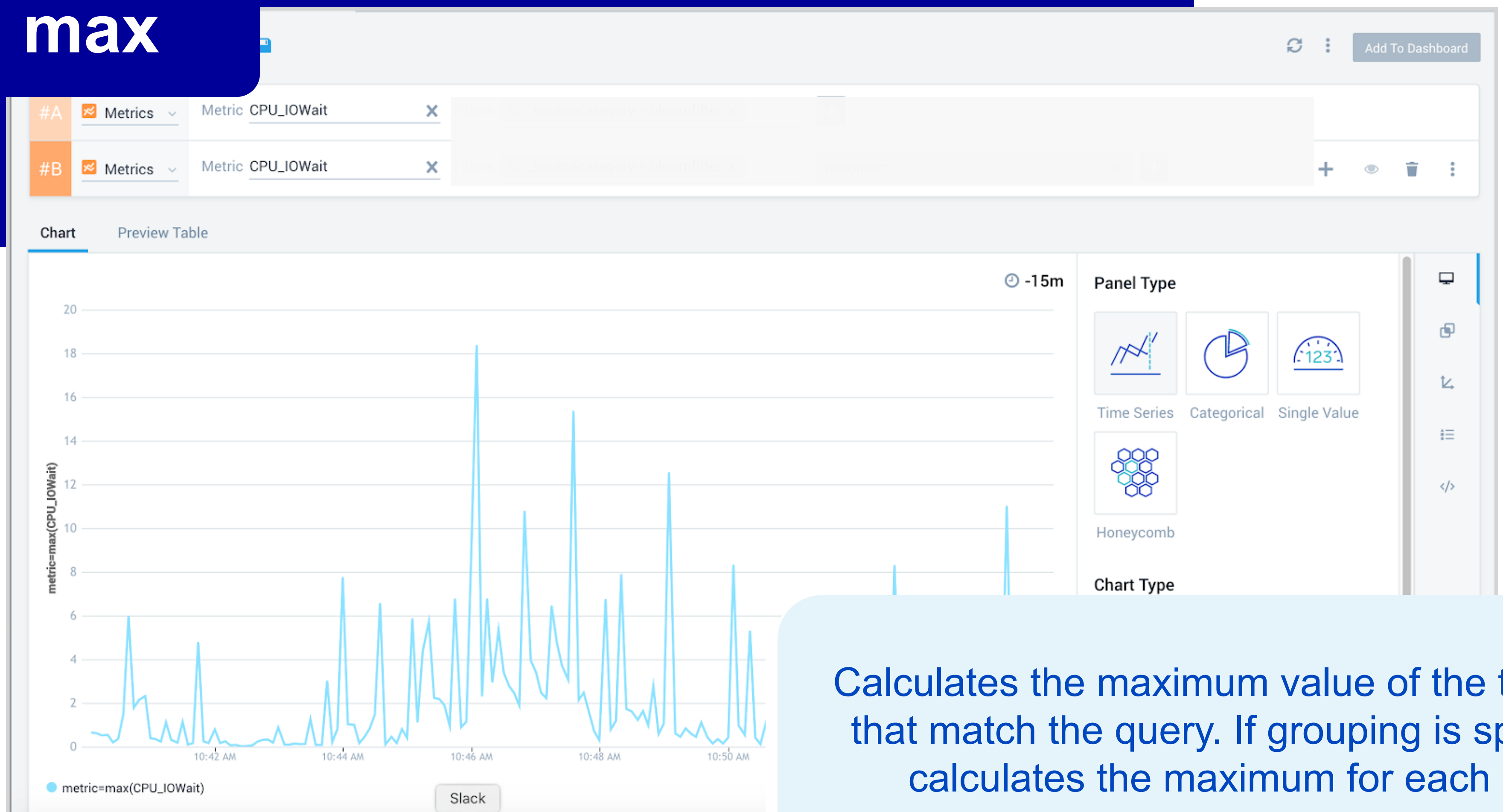
count



Counts the total number of time series that match the query. If grouping is specified, it counts the total number for each group.

CPU\_Idle\_sourcecategory=bloomfilter | count

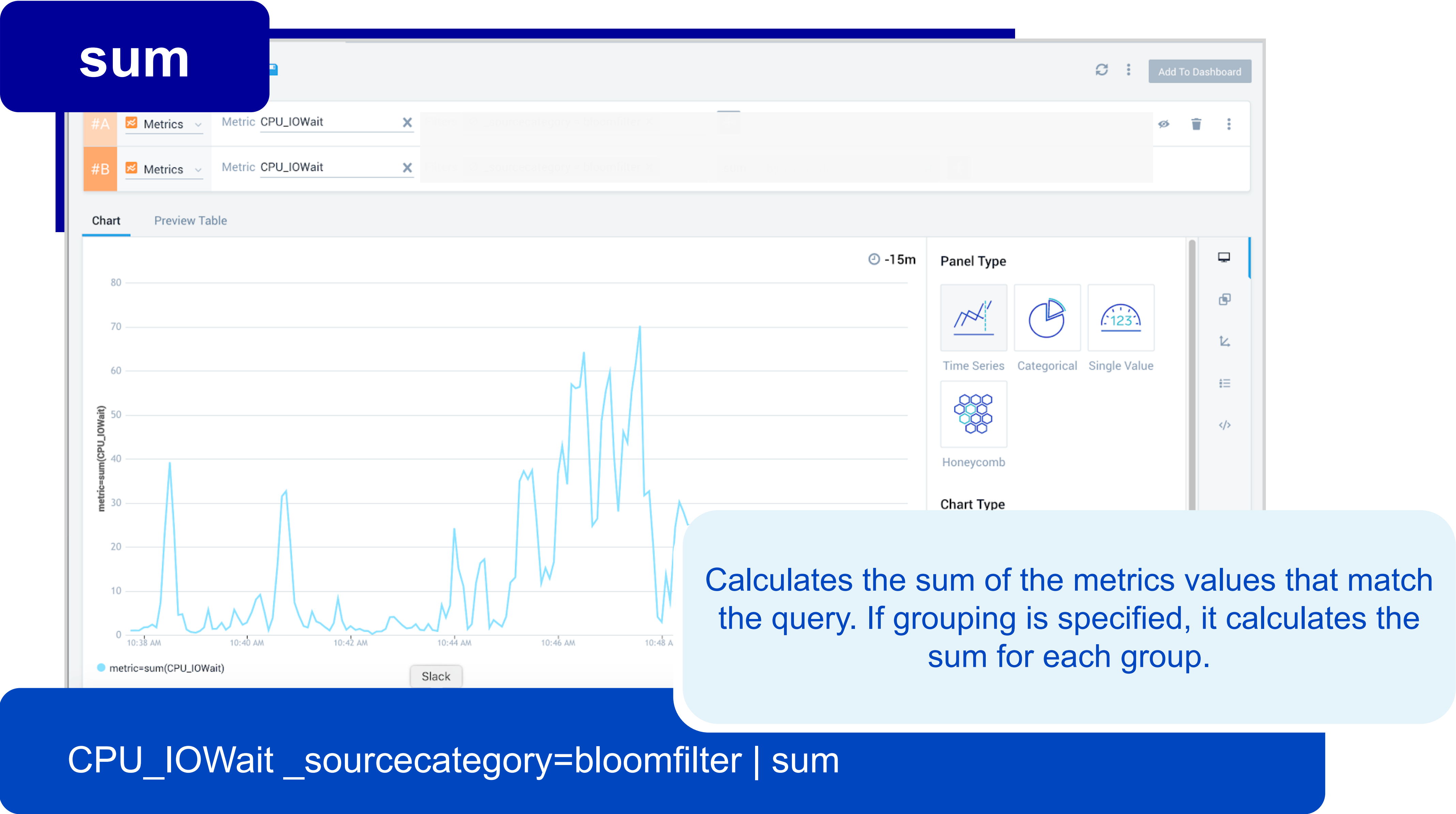
max



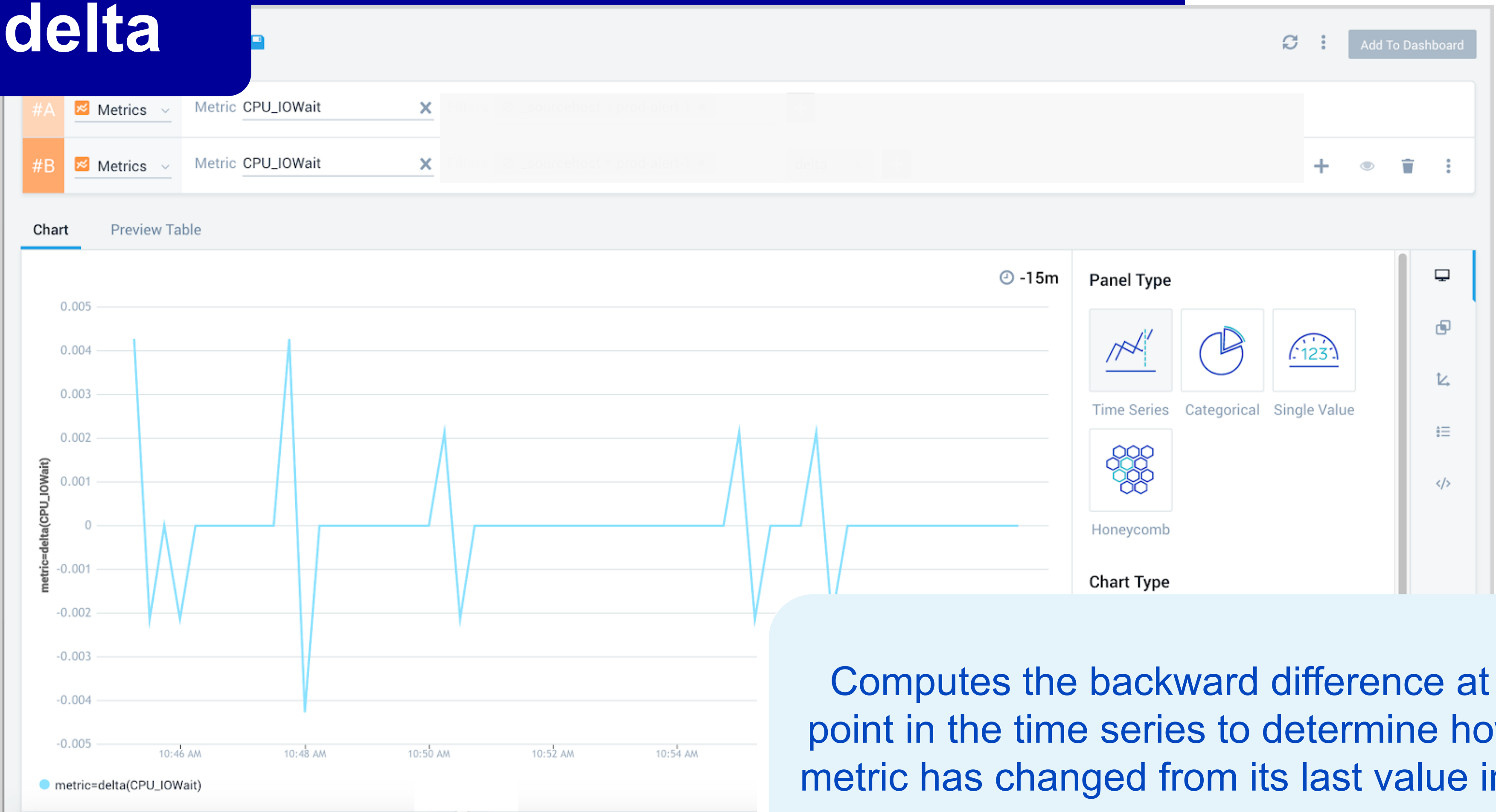
Calculates the maximum value of the time series that match the query. If grouping is specified, it calculates the maximum for each group.

CPU\_IOWait\_sourcecategory=bloomfilter | max





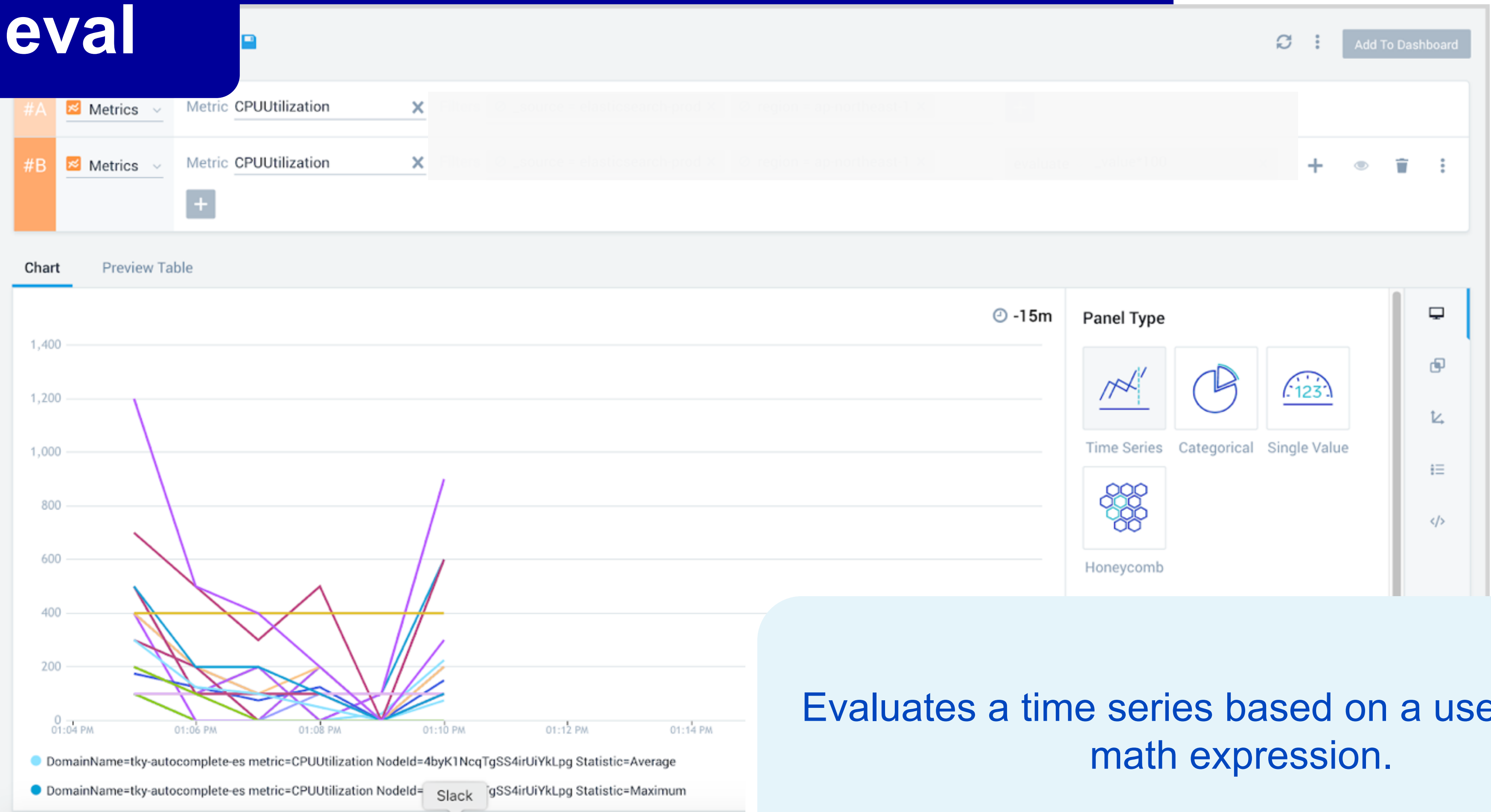
delta



Computes the backward difference at each data point in the time series to determine how much the metric has changed from its last value in the series.

CPU\_IOWait\_sourcecategory=bloomfilter | delta

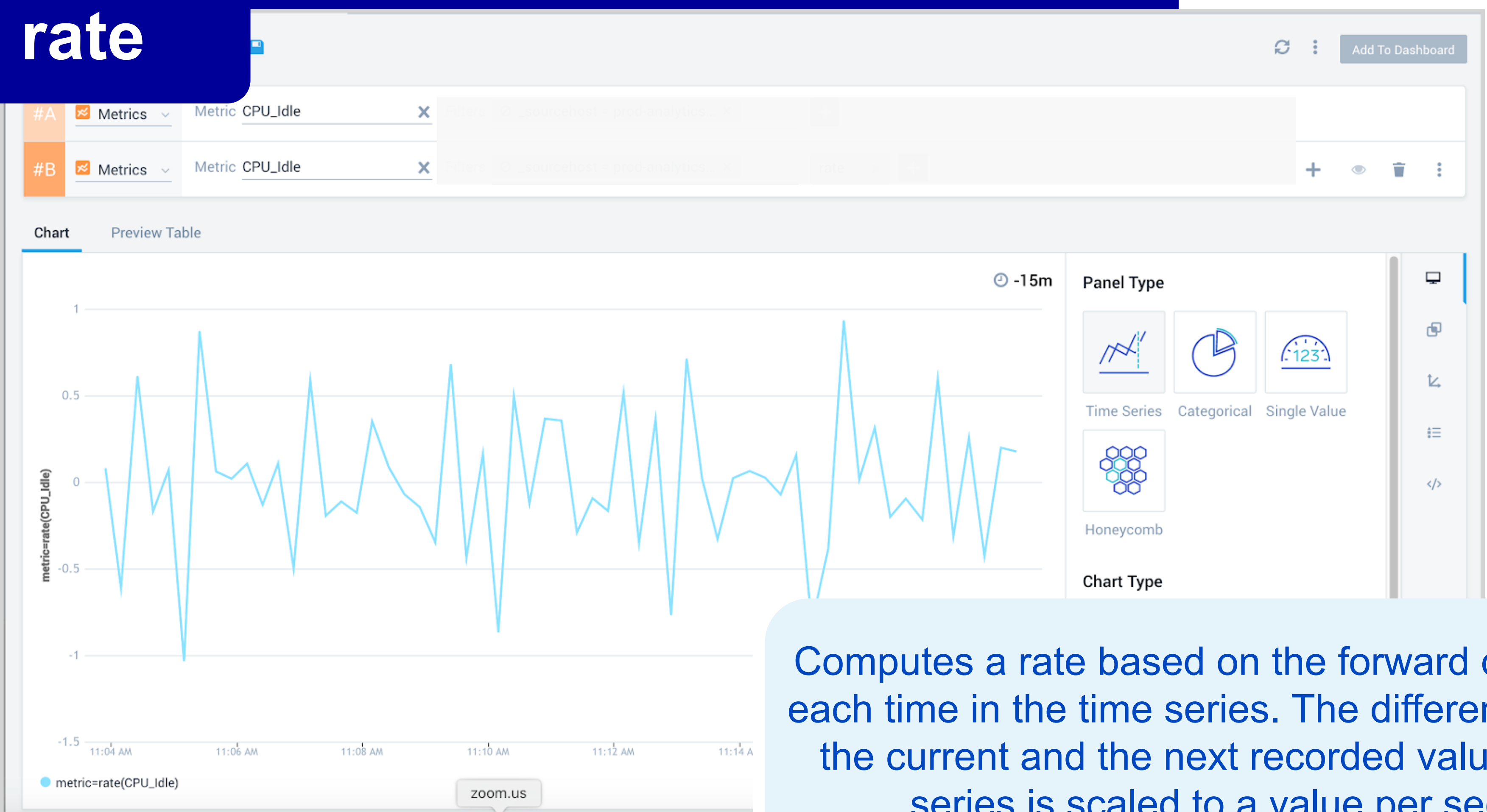
eval



Evaluates a time series based on a user-specified math expression.

CPUUtilization\_sourcecategory=bloomfilter | eval \_value\*100

rate

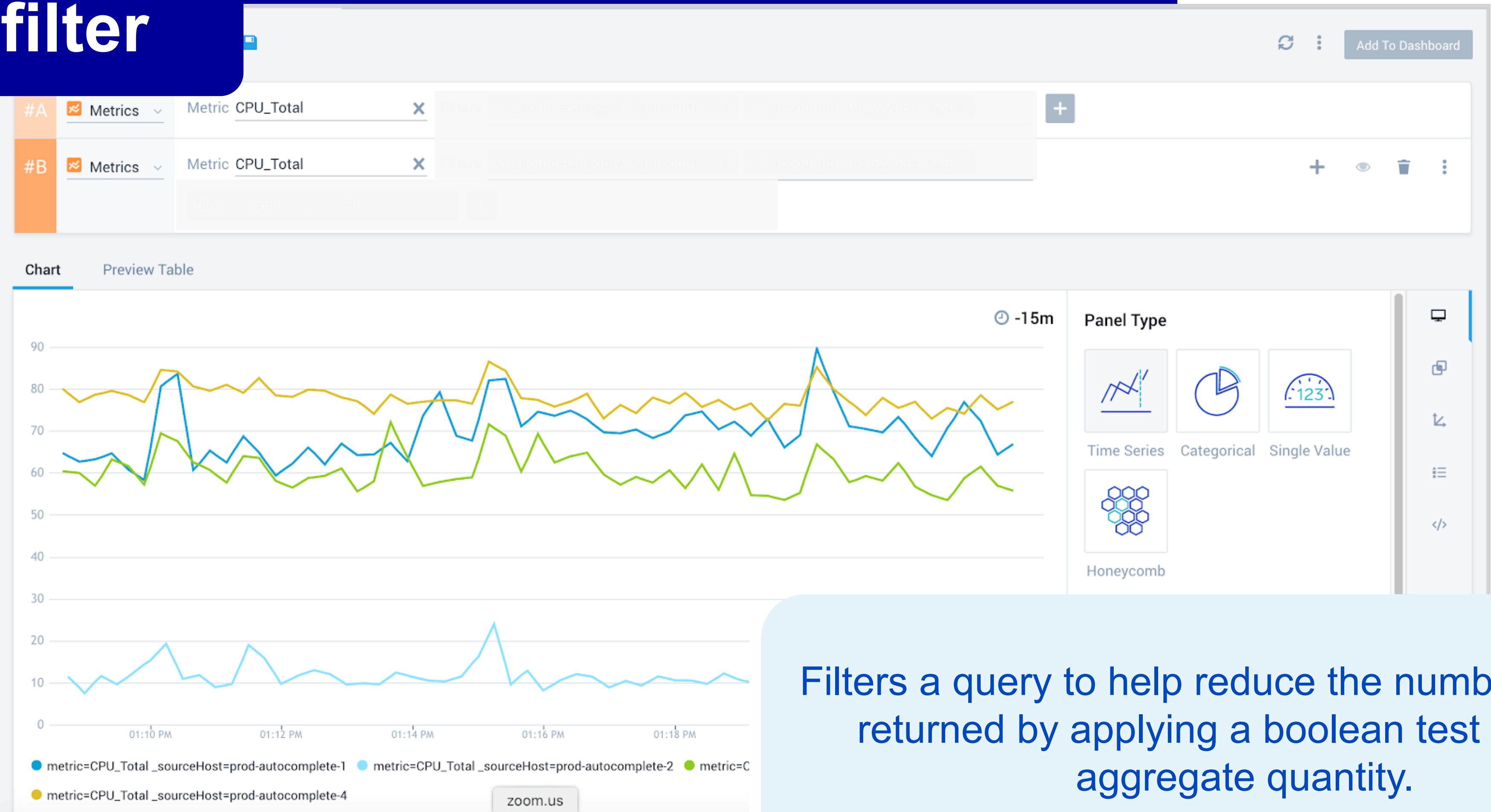


Computes a rate based on the forward difference at each time in the time series. The difference between the current and the next recorded value in a time series is scaled to a value per second.

CPU\_Idle\_sourcecategory=bloomfilter | rate



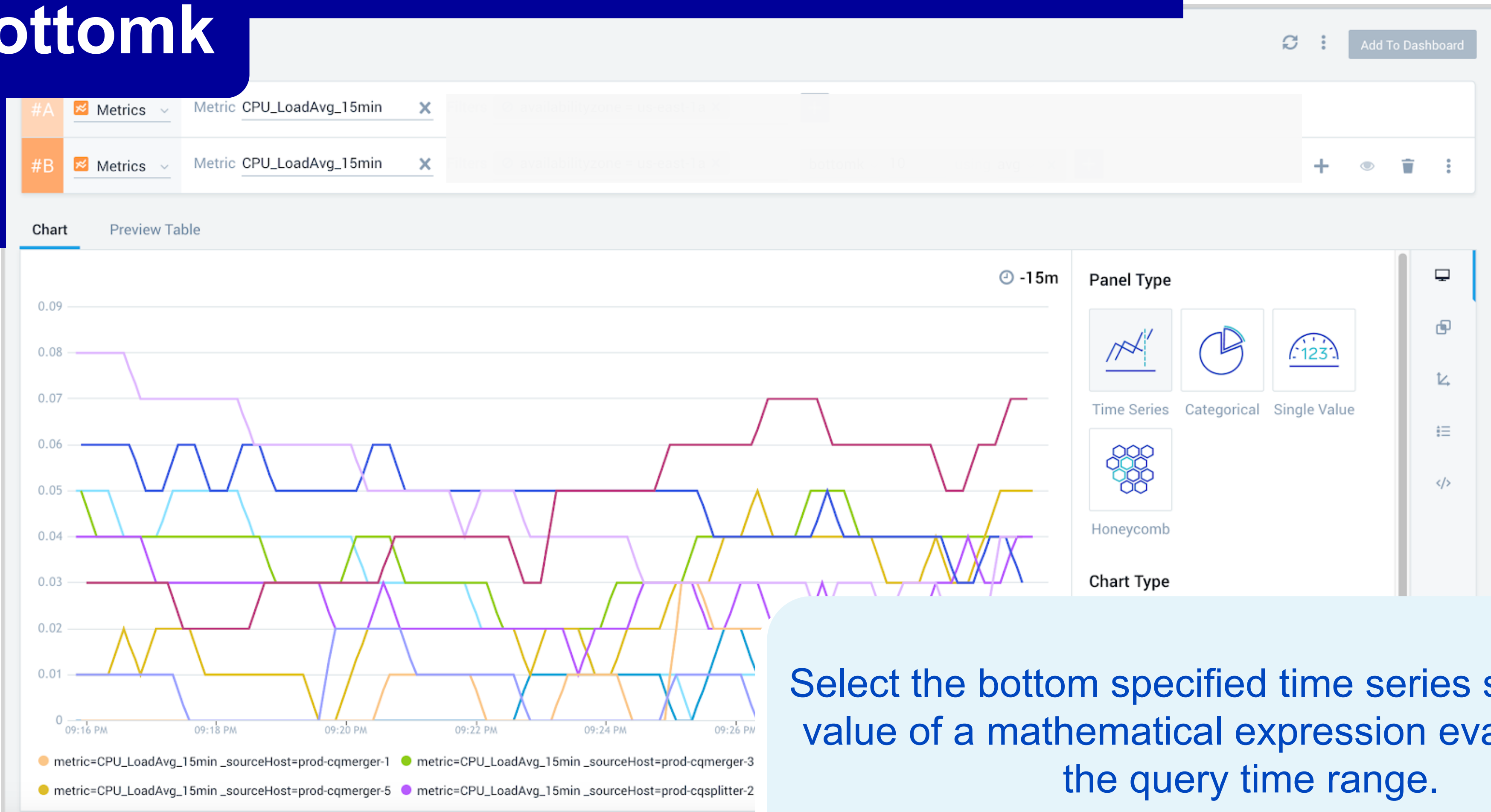
filter



Filters a query to help reduce the number of series returned by applying a boolean test to some aggregate quantity.

CPU\_Total \_sourcecategory=bloomfilter | filter sum>40

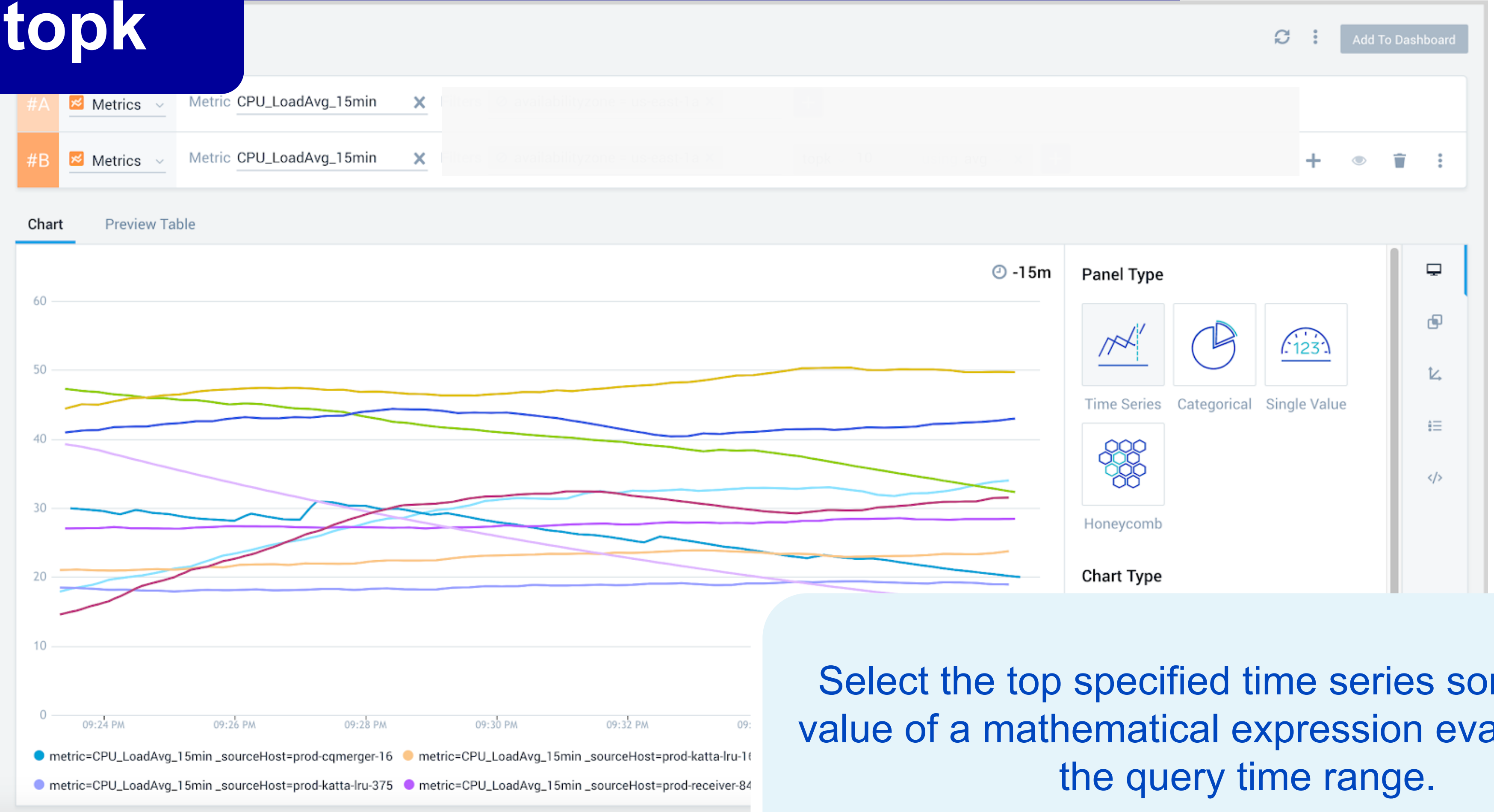
bottomk



Select the bottom specified time series sorted by the value of a mathematical expression evaluated over the query time range.

CPU\_LoadAvg\_15min \_sourcecategory=bloomfilter | bottomk 10 using avg

topk



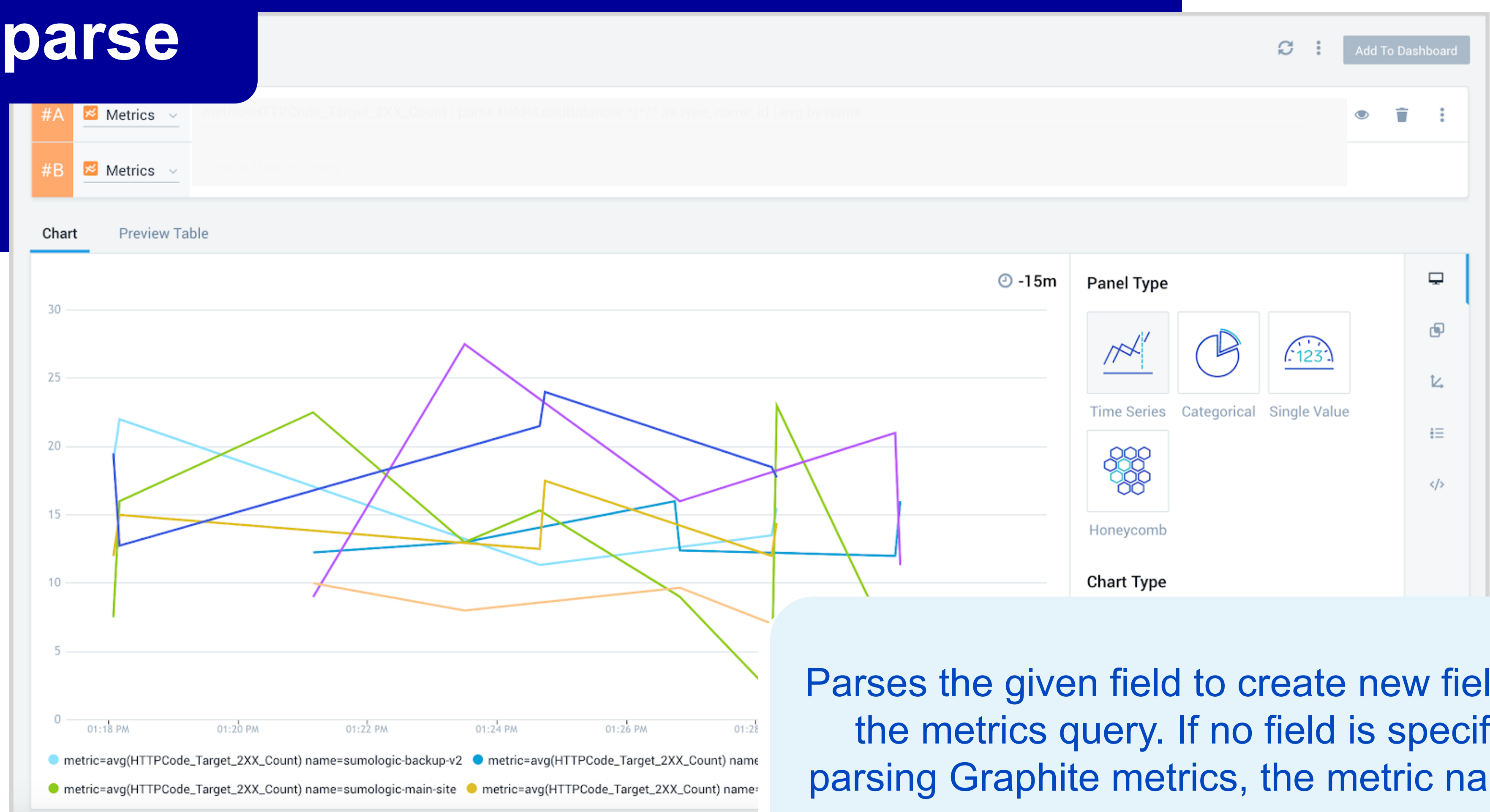
Select the top specified time series sorted by the value of a mathematical expression evaluated over the query time range.

CPU\_LoadAvg\_15min \_sourcecategory=bloomfilter | topk 10 using avg



QUERY OPERATORS

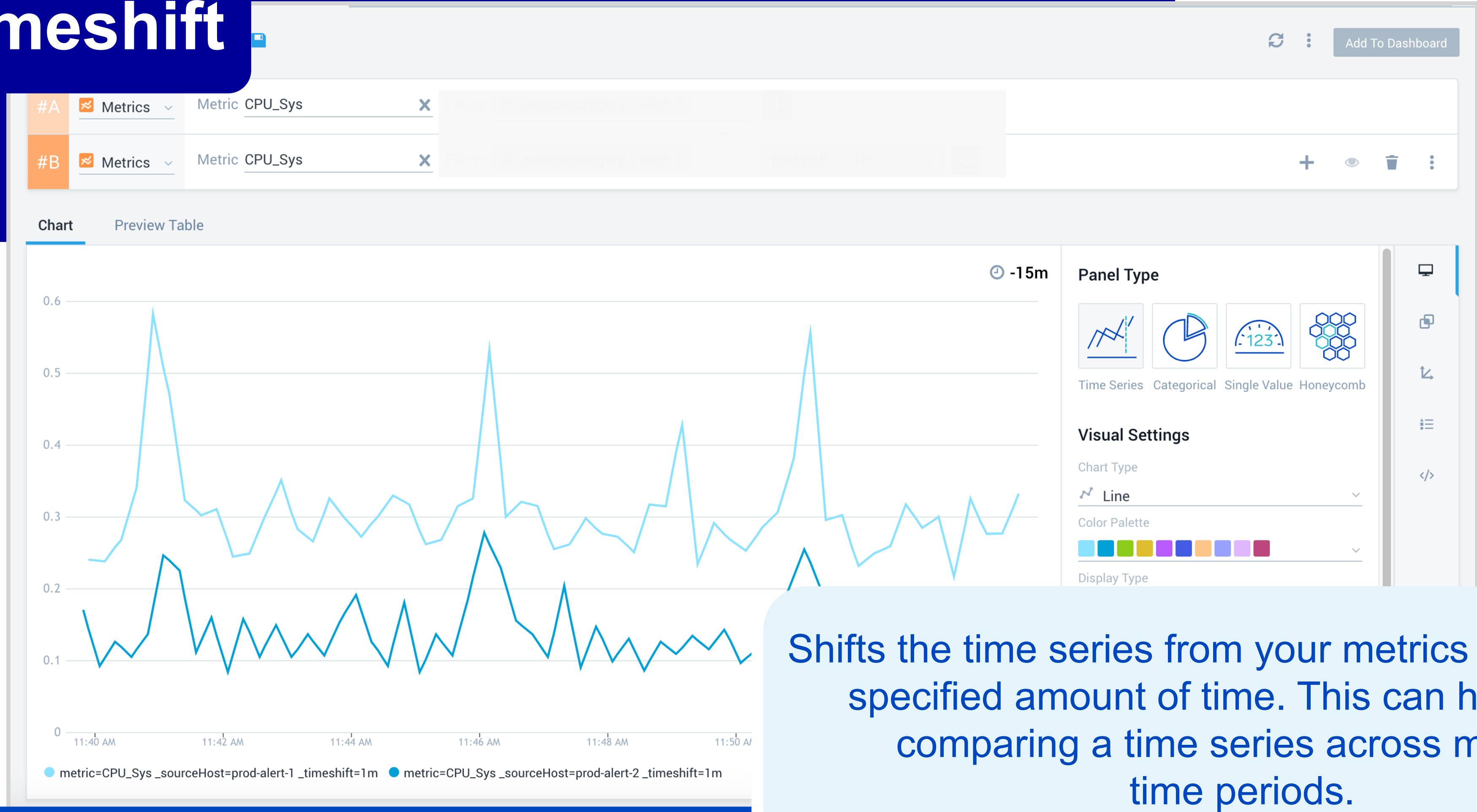
parse



Parses the given field to create new fields to use in the metrics query. If no field is specified while parsing Graphite metrics, the metric name is used.

metric=HTTPCode\_Target\_2XX\_Count | parse field=LoadBalancer \*/\*/\* as type, name, id | avg by name

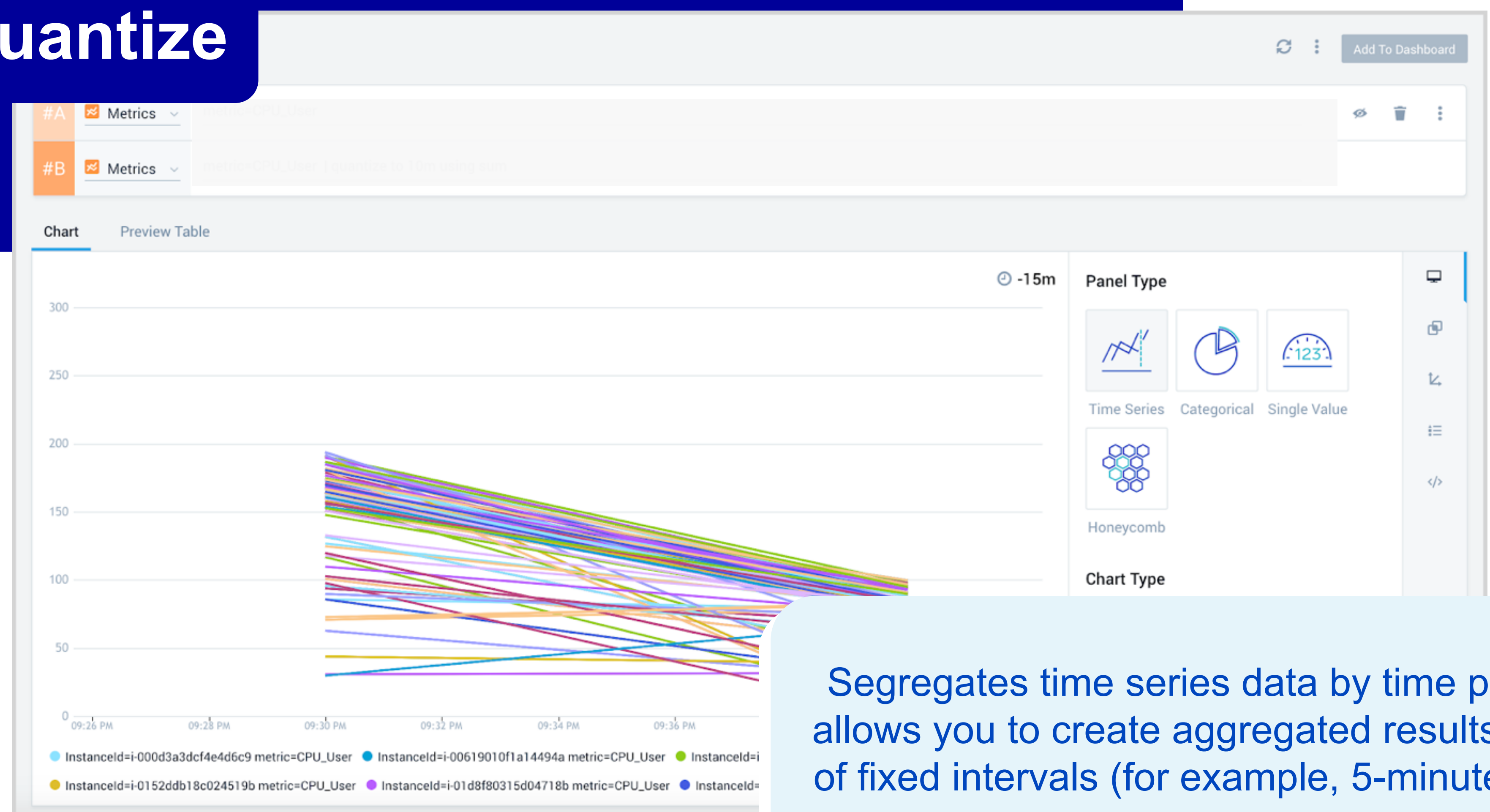
timeshift



Shifts the time series from your metrics query by the specified amount of time. This can help when comparing a time series across multiple time periods.

metric = CPU\_Sys\_sourcecategory=alert | timeshift 1m

quantize

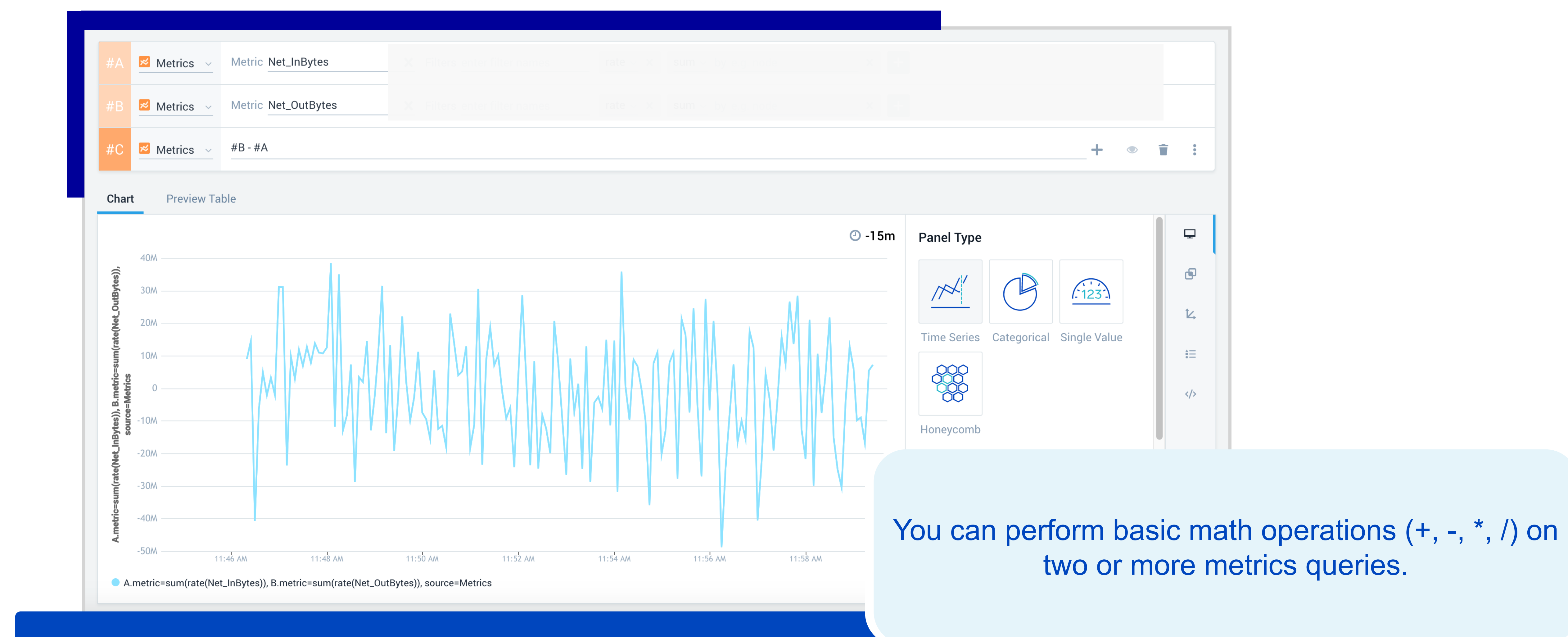


Segregates time series data by time period. This allows you to create aggregated results in buckets of fixed intervals (for example, 5-minute intervals).

metric=CPU\_User| quantize to 10m using sum



# JOIN METRICS QUERIES



## Net incoming and Outgoing Traffic

If you are ingesting network information, you could use metrics queries and math operations to chart the net difference of incoming and outgoing network traffic.

1. Define the incoming traffic as the result of rate and sum (to simplify the data). This will become our variable #A

Metric Net\_InBytes rate sum

2. Define the outgoing traffic as the result of rate and sum (to simplify the data). This will become our variable #B

Metric Net\_OutBytes rate sum

3. We need to subtract row #A from #B

#B - #A