Check whether an upsize for the host's instances is needed

Contents

Introduction

A frequent question that needs to be answered, is when a merchant's host needs to be resized. The following resizing cases may occur:

- Downsizing: The downsizing of a host may be relatively easy: When the reasons for the upsize do not apply any longer, a downsize may be considered.
- Upsizing: Sometimes the necessity of an upsize is very clear: When the CPU, LOAD, Memory, Network Dashboards show continuous high usages over large time periods, above the empirical threshold of – say - 80%, then an upsize is the recommended option.

There are cases however, where a host's upsize is necessary, even though metrics for classic variables such as CPU, Memory and load do not show the need for one. In this article, we are going to describe cases where the upsize is needed even though the regular dashboards show average values which do not necessarily pass threshold values.

Gathering of the metrics which point to an upsize

1) The necessity to find the merchant's Peak time

The peak time is defined as the time period in a regular day (24 hrs) where the traffic on the host's instances reaches the maximum. Peak time is usually the time period within the day where the customers are able to shop without any interruptions. This can be in theory during any time within a 24 hr period,

however there are certain time periods where a customer does not shop. Examples of such time periods are the nights where the customers sleep and the early mornings where the customers have to work.

It is important to find the host's peak time because it is this time period where the host should perform well. When a host works in idle traffic conditions, the probability of an issue to emerge is minimal. Additionally, there is an abundance of computer resources available during non-peak times, which provide the users responsible for the host's performance a kind of "assurance" that such low traffic will be served well. On the other hand, during peak time the availability of computer resources and their optimal usage becomes of utmost importance. If the host lacks resources during peak time, it may 1) perform sub optimally and 2) may force its applications and extensions to undergo errors.

Therefore, the need for a resize should be considered based upon data and analysis taken during peak times, and not in time periods such as for example an entire 24hr day. Using metrics which contain a large time period, non-relevant to the peak time, may also include values which skewer critical metrics and lead to incorrect decisions.

A clear way to identify the peak time period for a merchant's host, is to observe the host's daily pageviews and identify the peak time as the one where pageviews reach the global maximum. To create a chart in New Relic where the user can measure pageviews, the following query may be used:

```
SELECT count(*) from PageView TIMESERIES 10 minutes SINCE '2020-07-02 00:00:00 UTC' UNTIL '2020-07-03 00:00:00 UTC'
```

For the above query, the request is made to evaluate the sum of the Pageviews occurrences (the "count" function applied to the asterisk "*" highlighted in yellow) on a timestep of 10 minutes (the "timeseries" function) on the timeperiod of one day (the "SINCE" and "UNTIL" commands, highlighted in light gray). The resulted chart may looks like the following:

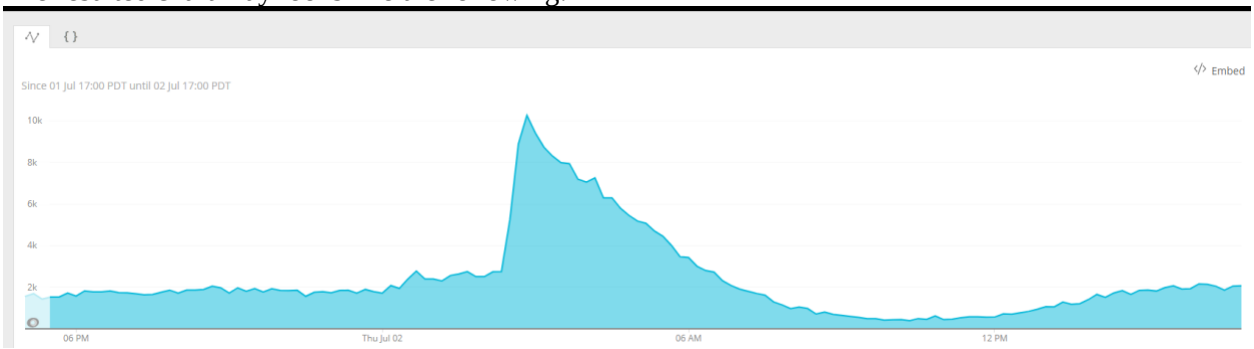

Figure 1

In the above graph, we observe a pageviews peak which occurs roughly in the middle of the day (in UTC). The peak is reached and then the pageviews gradually get reduced, to reach a low level during the merchants non peak time, roughly around 12 pm UTC.

There may be cases where the pageviews peak is significantly higher compared to the overall "norm", like in the following example:

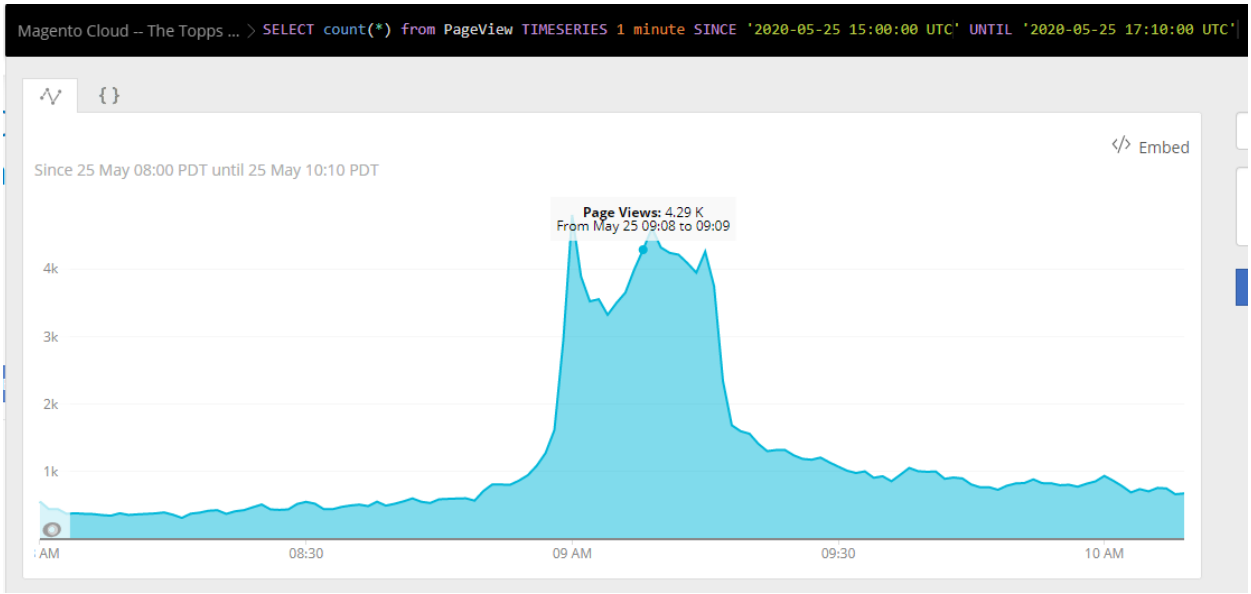

Figure 2

In the above example, pageviews increase roughly 400% within only 1 minute, and stay high for a brief period of 20 minutes.

In all cases, the user can do a “mouse over” or refine the query, to identify with precision when sharp increases of pageviews occur, and deduce the number (if there are more than one) and length of the peak times

2) Gather host’s performance metrics

Once a pageview peak time period (or multiple time periods) has been identified, a performance check can be done during the peak time period. A typical check can be done by using New Relic queries to observe:

- CPU
- Load
- Memory usage.

The example below can be used to outline the queries:

Example

Suppose that a pageviews peak time period has been identified from the pageviews query as approximately three hours: '2020-07-02 09:00:00 UTC' - '2020-07-02 12:00:00 UTC'

Then the New Relic Queries used to collect CPU, LOAD and MEMORY performance are the following:

CPU:

```
SELECT average(cpuPercent) FROM SystemSample TIMESERIES FACET entityName
SINCE '2020-07-02 09:00:00 UTC' UNTIL '2020-07-02 12:00:00 UTC'
```

Load:

```
SELECT average(loadAverageFiveMinute) FROM SystemSample TIMESERIES FACET
entityName SINCE '2020-07-02 09:00:00 UTC' UNTIL '2020-07-02 12:00:00
UTC'
```

Memory:

```
SELECT average(memoryUsedBytes) FROM SystemSample TIMESERIES FACET
entityName SINCE '2020-07-02 09:00:00 UTC' UNTIL '2020-07-02 12:00:00
UTC'
```

Analyze the gathered metrics and assess the necessity of an upsize

CPU case where an upsize is not required

The pageviews for the example's time period are shown in the following chart:

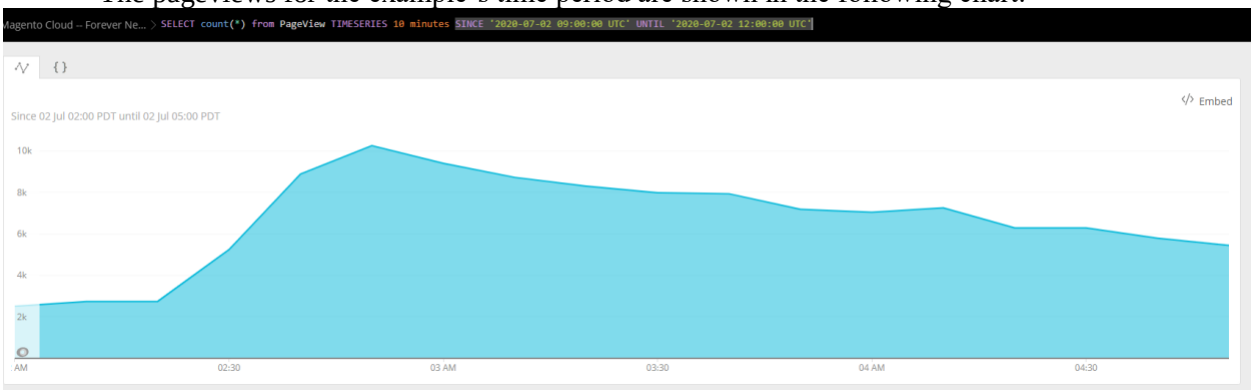

Figure 3

The CPU graph resulted from the New Relic Query from the same example outlined in the previous chapter may look like the following:

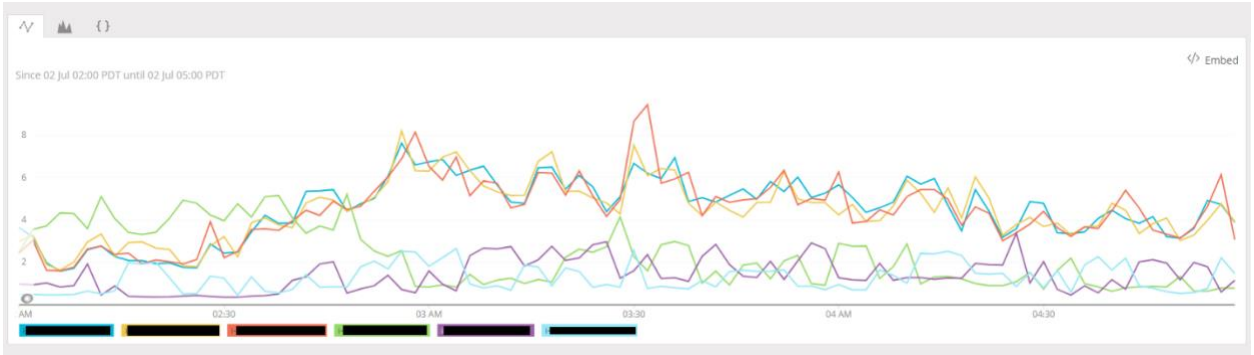

Figure 4

In Figure 4, we view a weak correlation between the CPU usage of three servers - the ones whose lines reach the upper CPU limit - and the corresponding pageviews shown in the Pageviews chart (Figure 3). We observe the starting of the increase in both charts (Figures 3 and 4) to be roughly at 3:00 AM PDT. However,

it is clear that in this case, an increase in pageviews does not lead to an alarming increase of CPU since the CPU % from Figure 4 does not surpass 9% usage. Though the Pageviews Peak is considerable, there is no followed-up CPU usage increase.

If the Pageviews-CPU usage for the host outlined in Figures 3 and 4 is observed on a daily basis, without major changes/fluctuation, then the user reaches the conclusion that an upsize due to daily CPU increase in peak time may not be considered.

CPU case where an upsize is required

There may be cases where CPU usage jumps to very high levels during peak time. This is depicted in the graph below:

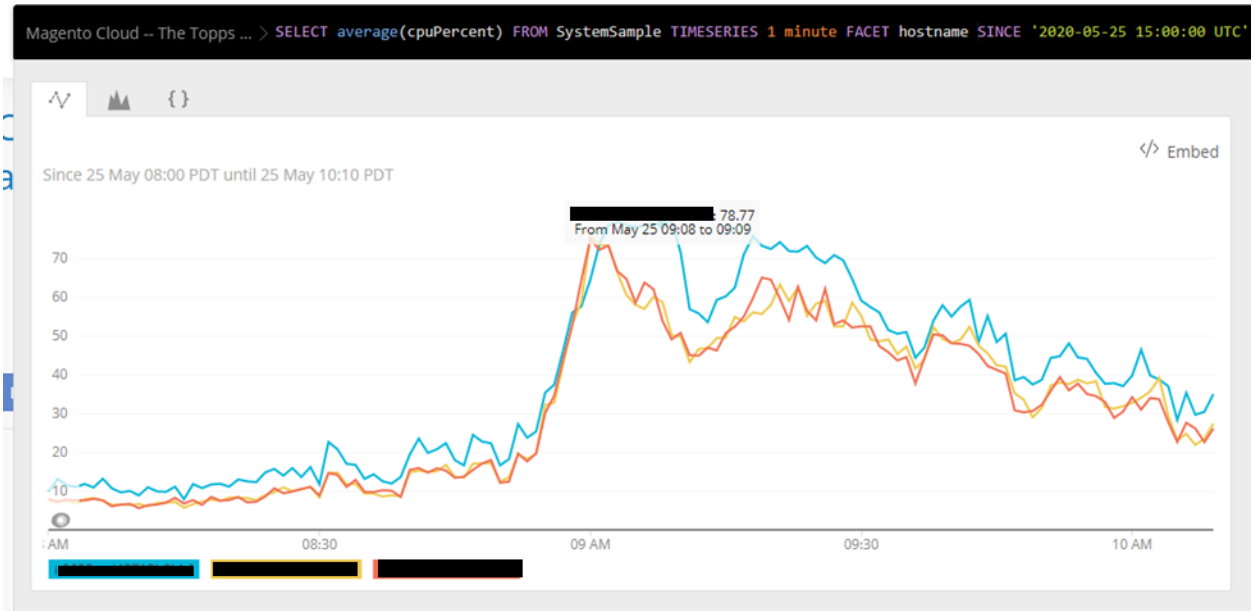

Figure 5

In this graph CPU usage of all three servers jumps to very high levels (70% usage and above) at exactly the same time where the corresponding pageviews chart shows a sharp increase (see second pageviews example of Figure 2).

If this Pageview-CPU behavior described above is regularly repeated during a consistent number of days (for example during the weekend, or every Tuesday), then an upsize based upon the daily observation of increased CPU usage during peak time should be considered.

Load and Memory values where an upsize should be considered

Similar to the CPU cases, the user may observe increased Load and/or increased Memory usage during the host's peak time. Though an increase in the mentioned metrics is not considered to be as urgent as an increase in CPU usage, a significant increase in Load or Memory during peak time may point to an upsize. Some empirical thresholds are the following:

- Normalized Load: Normalized load of an instance is the ratio of its current load with the total number of CPU's. If Normalized load reaches values significantly greater than the value of 1 and for long periods of time, it is implied that this instance does not have the capacity to complete its designated tasks in a timely manner. Therefore, an upsize may be required to provide the needed capacity
- Used Memory: When used memory reaches its limit (100% usage), the instance (or instances) becomes significantly slow. To avoid that, it is recommended that when used Memory reaches the threshold of 90% or above for significant amount of time, an upsize should be considered

Future traffic increase and the assessment for an upsize

All metrics of key computer variables are important for the day-to-day observability of performance. An important question is raised though, regarding future performance and how can "past and current" metrics be useful for the performance assessment of the future. Metrics by themselves cannot predict future traffic. However, if traffic predictions are provided, metrics can point with reasonable accuracy to the correct hardware configuration for optimal performance in the future.

During a time period, the number of pageviews measures traffic and the corresponding metrics (like CPU, Memory) provide a picture of how computer resources are used. The user can obtain pageviews and metrics of interest and create an approximate relation between traffic and resources usage. The task to create an exact model that estimates hardware usage from traffic – measured in this case by pageviews – is very difficult and time consuming. However, good estimation of the required hardware resources for the host's optimal performance can be achieved with the use of reasonable approximations.

One such approximation is for the key metric of CPU usage to be linearly dependent upon traffic, during peak times. That means that, a predicted increase in traffic during a future time period, relative to a past time period of reference, will result in the same relative increase of CPU usage. If then the (estimated) relative CPU increase is higher than the CPU availability at the time period of reference, the host may not be able to perform optimally, and an upsize may be needed.

Examples:

- Increased traffic, upsize not necessary: Lets assume in this example that the host described by Figures 3 and 4, will have a predicted 30% increase in traffic during a day in the near future, due to an event. The event may be for example a promotion, or the availability of a brand-new product. We see that from Figure 3 the peak time is around 3:00 AM and afterwards. From Figure 4, we see that CPU usage for certain instances during the same time period which was found from Figure 3 is around 6%-8%. By using the liner approximation of Traffic-CPU usage, we can estimate the CPU usage during peak time of the event day will have a 30% increase relative to the values of Figure 4. This means that the estimated CPU usage on the event day is on the range of 7.8%-10.5%. Therefore we conclude that during the event day, the hardware configuration can easily handle the estimated traffic increase.
- Increased traffic, upsize is necessary: Lets assume that we have the host described in Figures 2 and 5, and that this host is estimated to have a 30% increased traffic on a future day, again because of an event. From Figure 2 we find the peak time, and from Figure 5 we see that during peak time some instances reach around 70% CPU usage. From the linear approximation of Traffic-CPU usage, we estimate that the CPU usage on the event day's peak time will be around 91%. Even during the day of reference, the CPU usage is high during peak time. The estimated CPU usage for the event day is even higher, reaching the host's limit. In this case, an upsize should be considered, before the event day.

There are plenty of examples and cases where the user should consider an upsize due to future increase in computer resources consumption. In this chapter we outlined the most important, and perhaps the easiest one to estimate: CPU usage. Other key metrics such as load, Memory usage, Disk Usage, Database operation and many more can be of paramount importance for the host's optimal performance. By measuring the key metrics during a time period of reference and by creating reasonable approximations, the user has a powerful arsenal of data which can point to the correct host's hardware configuration.

Consideration of a Split (Horizontal) system resize

A horizontal scaling is the case where the server's configuration consists of instances dedicated exclusively to either Database usage or Web service usage, but never to both. Hence the name "Split configuration", frequently used in Engineering groups.

The choice of a "split" – horizontal configuration is most certainly a complicated matter, and this article does not aim to elucidate all cases where a merchant may need one. However, a simple rule of thumb for the need of such a system is when the merchant faces Web traffic which forces the site to use significantly more resources for Web service, relative to Database service.

A simple way to identify whether a merchant during peak time uses more resources for Web services relative to the Database services is by using those simple New Relic queries described below:

- Web Services (PHP-FPM):

```
SELECT sum( cpuPercent) from ProcessSample WHERE commandName LIKE 'php-fpm%' TIMESERIES 10 minute FACET entityName SINCE '2020-07-01 12:58:00 UTC' UNTIL '2020-07-01 16:12:00 UTC'
```

- Database Services:

```
SELECT sum(cpuPercent) from ProcessSample WHERE commandName = 'mysqld' TIMESERIES 10 minute FACET entityName SINCE '2020-07-01 12:58:00 UTC' UNTIL '2020-07-01 16:12:00 UTC'
```

The results for these two New Relic queries are shown in the below graphs:

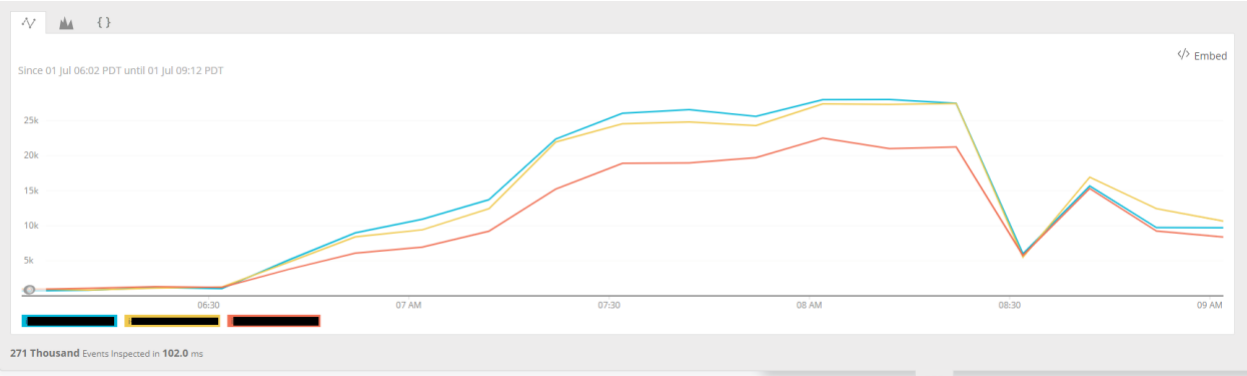

Figure 6: Web Services (PHP-FPM) CPU usage

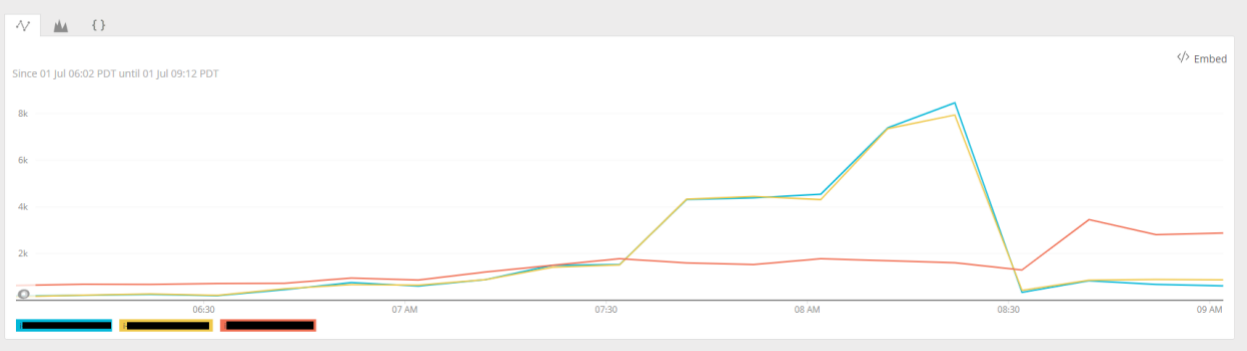

Figure 7: Database CPU Usage

From these two graphs we observe that CPU usage for Web services relative to Database services is roughly 25K/8K, approximately 3/1. That means that this merchant during peak times may observe higher performance if a horizontal configuration is set up to serve the Web service with large, resource “rich” instances and the Database service with smaller instances.