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.

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 respon 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 me
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

Figure 2

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

sharp increases of pageviews occur, and deduce the number (if there are more than one) and length of the peak times

2) erformance 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

- Normalized Load: Normalized load of an instance is the ratio of its current load with the total . 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.

Metrics by themselves cannot predict future traffic. HowG[pr)-73z3(r)-7(m)5(a)-10(nc)12(e)-10(.)]TJETQq0.00000912 0 612 792 re5[illegible]

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 o
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

Consideration of a Split (Horizontal) system resize

Figure 7: Database CPU Usage