Adjust Exercise 2:

Imagine a server with the following specs:
- 4 times Intel(R) Xeon(R) CPU E7-4830 v4 @ 2.00GHz
- 64GB of ram
- 2 tb HDD disk space
- 2 x 10Gbit/s nics

The server is used for SSL offloading and proxies around 25000 requests per second.
Please let us know which metrics are interesting to monitor in that specific case and how would you do that? What are the challenges of monitoring this?

[Michael:]

Metrics:
I'd consider monitoring basic hardware metrics such as:

- CPU utilization, for each of the available CPUs (gauge)
- System/fans temperature (gauge)
- Available RAM (gauge)
- Available Diskspace (gauge)
- Bandwidth utilization for each of the nics (gauge)

Additionally I'd consider some SSL Server related metrics:

- Number of received/handled SSL requests per second, as percentage of a maximum allowed value, such as 35K (gauge)
- Number of received/handled SSL requests in the past hour (counter)
- Number of erroneous requests per second (gauge)
- Number of erroneous requests in the past hour (counter)
- Histogram of number of requests for each hour in the past 24H/week

These metrics may generate alerts for values that are too high, as well as for exceptionally low, which may indicate a problem that is blocking traffic.
The histogram may be used to indicate peak or low hours, as well as any unusual usage patterns.

Another metric that may be interesting for SSL offloaders is some kind of indication of the number of requests arriving from the same source IP. In some cases this may indicate a buggy client or a malicious attack on our server. We may consider an histogram indicating the top 10 source IPs for SSL requests to our server. This may be accompanied by a counter indicating the amount of time the top IP has remained the same. Alerts should be generated in case the same IP is sending requests for too long.

Implementation:
Collection of the metric values may be collected by a standard monitoring tool, such as Prometheus or InfluxDB, and by running some sort of collection agent, such as Node Exporter on our target server (the SSL offloader). The Collector agent will periodically collect the metric values from the server and either push them to the monitoring tool or wait for the monitoring tool to pull them.
The monitoring tool may then produce alerts, graphs, reports or any other indications required by IT, as well as queries.

Challenges:
There are 2 main challenges that occur to me regarding this use case:
The first involves the server being too busy to promptly run the collection process when data is required. This is especially problematic during hours of peak usage, in which the metrics are required the most.
Another problem may be that the SSL offloader server and the monitoring app are separated by a NAT/Firewall, preventing the monitoring tool from pulling the metrics.
In both cases we may consider having the collector agent push the metrics to the monitoring tool once metrics are available, instead of having the monitoring tool attempt to pull them. This will both allow the data to reach the monitoring tool, as well as allow the server to produce them when it can.
In case it is mandatory to produce the metrics in fixed intervals regardless of any pressure on the server, I'd consider dedicating one of the CPUs to running the collector agent.