**How to measure application performance**

While application design and development, most efforts go to system design and business logic implementation. The entire emphasis is on application development and taking it to the production stage. Once the application reaches production, real business kicks off. Therefore, the most importance goes to keeping applications up and running, providing support to business operations, and adding more features to generate more revenues. Everything looks great until some hidden challenges arise.

Even if the application seems reliable today, that doesn’t mean it will work reliably in the future. Among all these activities, what most of the time gets neglected is application performance. Which becomes a bottleneck once it reaches its threshold limit. As the application matures, more business adds up, which means more load on the application. As the load on the application increases, the entire performance of the system got impacted. As the performance of the system goes down, it ruins entire user experiences, and eventually, businesses got a huge impact due to this.

Once application performance starts degrading, one solution appears that make application Scalable. Scalability means that application should be able to cope with increasing load. Means add more computing resources and memory to handle the additional load on application. System Scalability can be achieve by vertical scaling (Scale up) or horizontal scaling (Scale out).

Vertical scaling mean increasing computing power (CPUs) and memory (RAM) of machine to handler increasing load. Horizontal scaling mean added more machines with computing power (CPUs) and memory (RAM) in cluster to distributing additional load on application. Before reaching any of scalability approach for your application, you must address two main questions:

1. You must describe the load on current system. How much load current system is handling right now?
2. You must describe the performance of current system. What is performance of current system right now?

**Describing Load**

Before solving application scalability problem, we need to understand current load on the system. Load on the current system can be described by getting some fundamental metrices of current system. Here are some important metrices you can collect for your application to describe current load on system. Some of metrices may not belong to your applications. Because these metrices are applicable based on your application architecture.

* Number of requests per second handled by current application server.
* Number of reads requests per second on database.
* Number of writes requests per second on database.
* Ratio of reads and writes request on database.
* Number of concurrent active users on application.
* The hit rate on cache.
* Number of messages pushed per second on messaging queue.

These number can provide you more clear picture of bottlenecks in your system. Perhaps average number of these metrices matter you most, except in some small extreme cases. By analyzing these numbers, you can design some fundament architecture change approaches, which you feel can provide you better results in terms of performance and scalability. But only understanding current load on system alone wouldn’t help you much. You also need to describe and understand current performance of your system.

**Describing Performance**