**Performance Metrics**

**Throughput**

Some amount of work done in a particular unit of time. (Work/time)

It’s an important concept when we need to know number of API calls done by a system in unit of time.

For example: A food delivery app can serve 1000 orders per 30 mins.

**Bandwidth**

Important in data being transferred over networks.

When you have many resources or **a lot of data** to transfer but **bandwidth of network is limited**, then your **throughput will also be low**, as it is limited due to bandwidth.

When you **have lot of data** to transfer and **bandwidth is also high**, you can transfer high amount of data in a time, hence **throughput will also be high**.

When **data to be transferred is low** but you network provides **high bandwidth**, still **throughput will be low** since we did not have that much amount of data.

Resources (data) with Bandwidth is directly proportional to Throughput.

**Response Time**

Amount of requests served in a unit of time.

Suppose a server is capable of serving 1000 requests in second. (1000/1)  
It takes 1 second to process and respond to a request.

Bandwidth of network is 1Gbps (very high).

But due to some reason, it started taking 2 secs to process and give response.

In that case, overall throughput will decrease to 500 requests in a second. (1000/2)

Hence, to **increase** the overall **Throughput** (serving max. requests in a time), you must have **high** **bandwidth** network, your code on server must be optimized to have **minimum** **response** **time**.

Objective is to achieve **minimum cost** and **maximum performance** with available resources.