**Context/Question:**  *Item 3 – Document Search – Provide some thoughts on what you would do on the software or hardware side to make this program scale to handle massive content and/or very large request volume (5000 requests/second or more).*

**Handling large number of requests:**

One of the ways to handle the massive contents and/or very large requests is horizontal scaling.

There could be a Load Balancer to handle the incoming traffic/calls and then several servers, which would actually handle the search request.

Load Balancer server will receive all the incoming traffic/calls, and divert the calls to one of the search server. Calls distribution to these search servers could happen with any of the logic, such as least in use or hash based distribution.

Using this approach, more servers could be added if requests keep increasing drastically and vice-versa, if requests diminish, servers could be taken out from the load balance cluster to save cost.

**Regarding the increasing contents (documents) to search:**

There is couple of scenarios to be considered in order to propose a solution:

1. If the documents get updated with less frequency and number of documents stays under a threshold number, such that these documents can be loaded into the memory without having system running out of the memory:
   * In such case, all the data could be loaded into the memory/Dictionary/HashTable, whenever documents get updated and server(s) could be scaled vertically (more RAM, better(multi) Processor, more storage)
   * All the search servers could update their memory data as and when content is updated. Load Balancer could spare a server from receiving the calls while that server updates its memory data. Once memory data is updated, Load Balancer would include that server and could start diverting the search requests to it
2. If the number documents/content is huge and/or the update frequency of the content/documents is high:
   * A NoSQL database, such as Couchbase (or Cassandra) could be utilized to scale horizontally. Memcached can also be utilized for fast and exact access to the respective search terms
   * Utilizing an easy scalable NoSQL database will make the scalability easier and manageable as traffic increases/decreases
   * Search servers will utilize the database(s) to provide the fast results. Search servers do not need to be heavyset and depending on the workload, more search servers and database servers could be added for horizontal scalability
   * NoSQL database servers would utilize data sharding to keep the load balanced among them