4.3.10 Scalability

Return to Non-Functional Requirements

- ☐ **[char]Please Review**

About

Scalability is the ability of a system to accomplish more work while maintaining the quality (i.e., without degradation) of the products produced or services provided by the system. There are different ways to calculate the work produced or performed by the system, which usually depends on the kind of product produced or services rendered.

For software systems, here are some of the common metrics used to quantify the products produced or the services provided:

- Number of transactions or events per unit of time (e.g., 5,000 credit card transactions a second[1)])
- Number of tweets processed per unit of time (e.g., 6,000 tweets a second[2)])
- Number of hours of videos uploaded per minute (e.g., 500 hours of fresh video per minute[3)])
- Number of searches per day (e.g., 3.5 billion searches per day[4)]).

Scalability is about being able to increase the output of products and services without major disruptions, interruptions or increased costs. Often, because of the Economies of Scale, the estimates for the costs should actually decline.

Two valid approaches to achieve Scalability are:

- Scaling Up: An approach generally applied to centralized or decentralized systems or products. It amounts to just adding more resources with more powerful versions. This method works until you have reach the limits on the availability of more sources power, for example, the speed of the Central Processing Unit (CPU), the amount of available memory or even the network capacity or speed. In decentralized systems(i.e., Cloud Computing) scaling can be either Vertical or Horizontal.
- Scaling Out: An approach usually associated with a Distributed System, which by its nature, allows for more Network Nodes, replicated with prepacked applications (i.e., Distributed Application (ĐApp or DApp)), that can be added with minimal overhead cost. In essence, adding more nodes offering redundant products and services. Alternatively, one can divide up the problem by functionality. For example, if accessing customer data is the bottleneck, then add more nodes with which to access the same data. If access to the actual data is the bottleneck, adding replications to the data store is recommended.

DIDO Specifics

Return to Top

☐ *TBD - to be added/expanded in future revisions of the DIDO RA*

1)

Ryan Vlastelica, Why bitcoin won't displace Visa or Mastercard soon Market Watch, 18 December 2017, Accessed 10 August 2020, https://www.marketwatch.com/story/why-bitcoin-wont-displace-visa-or-mastercard-soon-2017-12-15

2)

Kit Smith, 60 Incredible and Interesting Twitter Stats and Statistics, Bandwidth, 2 January 2020, Accesssed 10 August 2020, https://www.brandwatch.com/blog/twitter-stats-and-statistics/

3)

James Hale, More Than 500 Hours Of Content Are Now Being Uploaded To YouTube Every Minute, Tubefilter, 7 May 2019, Acessed 10 August 2020, https://www.tubefilter.com/2019/05/07/number-hours-video-uploaded-to-youtube-per-minute/

4)

Maryam Mohsin, 10 Google Search Statistics You Need to Know in 2020 [Infographic], Oberlo, 3 April 2020, Accessed 10 August 2020, https://www.oberlo.com/blog/google-search-statistics

From:
https://omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc:16_scalability&rev=1622857084

Last update: **2021/06/04 21:38**