# 4.2.7 Performance

Return to Non-Functional Requirements

- ☐ **[char]Please Review**

Performance is the ability of a system to accomplish the required functionality at or under the required specification limits. The limits are generally provided relative to time (i.e., so-many transactions per second, so-many updates per millisecond, so-many recorded entries per second, etc.) The specifications can also include accuracy, precision, precision or even efficiency of other dependent systems as requirements. The following are some examples of performance specifications:

- Time for a data-entry window to appear
- Number of units produced in a given amount of time
- Time to react to a given event
- Amount of energy required to perform an activity
- The amount of computing resources required (i.e., Central Processing Unit (CPU), Random Access Memory (RAM), Read-Only Memory (ROM), Storage Device, Bandwidth, etc)
- Time to process a computational task such as compression-decompression, encryption-decryption, generate a Unique ID (UID), serialize-deserialize, calculate an area, calculate a new trajectory, etc.
- Time to process a storage task such as store records, index the records, retrieve records, etc.
- Time to access memory or cache such as direct memory access(DMA), the hit ratio
- Time to transfer data between components of a computer such as from memory to Central Processing Unit (CPU), from CPU to graphics card, etc

- 01_platform
- 02_application
- 04_nework

## DDS Specifics

Return to Top

One of the more common requirements for communications middleware is performance, usually specified as either Latency or Throughput or a combination of both.

Latency is a measure of the total time it takes a data packet to travel from one Network Node to another network node. Commonly, it is the amount of time it takes for a data packet to travel round trip from one node to another node and back again. In essence, Latency refers to time interval or delay in a system to wait for inter-component communications. The duration of waiting time is called Latency.

Throughput is a measure of the quantity of data transferred from one location to another in a given amount of time, and within the Data Distribution Service (DDS) context is the data transferred over the network.

Network performance is one characteristic that will vary between Data Distribution Service (DDS) implementations. However in general, and for most DDS products, Latency is measured in microseconds. For some of the high-performance DDS implementations, the expected latencies are under 50 microseconds for network communication between network nodes and under 30 microseconds for “on-box” communications. For embedded environments, such as rtos, networkstack, or even the network hardware, a Latency is occurs because of various restrictions and limitations which can result in slightly higher Latencies, but still measuring Latencies in the microseconds.

Looking at Throughput, in general DDS will have Throughputs approaching the theoretical maximum limits of the network type. This holds true for networks rated at 1 megabit per second all the way up to 10 Gigabit per second. DDS takes advantage of aggregation and batching of smaller data packets to further reduce overhead.

These Latency and Throughput numbers make DDS a good choice for all kinds of data and all kinds of data streams.

The DDS specifications were written specifically for near real-time systems. Some examples of specification items result in high-performance communications:

- Specifies minimal data copies in memory - Buffer loaning
- Specifies compact binary encoding on the wire - Reducing bytes on the network. There are no large eXtensible Markup Language (XML) or JavaScript Object Notation (JSON) strings being sent over the network.
- Specifies light-weight notification mechanisms - Asynchronous and Synchronous options available
- Data types are specified at compile time - Allowing optimized: marshalling, filtering, searching
- Specifies User Datagram Protocol (UDP) multicast allowing for 1-to-many communications. Additionally, besteffort and reliable protocols can be used. This reduces network overhead and high performance scalability.
- Intermediate brokers are not required - providing direct Peer-to-Peer (P2P) communications

Return to top

Figure 1:

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

Permanent link:
**https://omgwiki.org/dido/doku.php?id=dido:public:ra:1.4_req:2_nonfunc:40_performance&rev=1605936175**

Last update: **2020/11/21 00:22**