**FDT**

The Fast Data Transfer (FDT) [fdt] is high performance data transfer program for Wide Area Network. It is based on concurrent multi-thread IO operations and in this way can effectively use high bandwidth networks with standard TCP together with high performance storage systems.

FDT is being developed to support efficient large scale data transfer services and also to help in the active monitoring of the available bandwidth between sites.

FDT is written in Java, runs an all major platforms and it is easy to use. FDT can be used as an independent application but it can also be controlled and managed by the MonALISA [monalisa] system to provide effective data transfer services.

FDT is based on an asynchronous, flexible multithreaded system and is using the capabilities of the Java NIO libraries. Its main features are:

• Streams a dataset (list of files) continuously, using a managed pool of buffers through one or more TCP sockets.

• Uses independent threads to read and write on each physical device.

• Transfers data in parallel on multiple TCP streams, when necessary.

• Uses appropriate-sized buffers for disk I/O and for the network.

• Restores the files from buffers asynchronously.

• Resumes a file transfer session without loss, when needed.

FDT can be used to stream a large set of files across the network, so that a large dataset composed of thousands of files can be sent or received at full speed, without the network transfer restarting between files. The FDT architecture allows to “plug-in” external security APIs and to use them for client authentication and authorization. FDT supports several security schemes: IP filtering, SSH, GSI-SSH, Globus-GSI, and SSL.

FDT is well integrated in the MonALISA framework. FDT can be dynamically controlled by the MonALISA system and he bandwidth used to transfer datasets can be dynamically adjusted for large scale data transfer services that support priorities and have real-time information on network topology.