Video states purpose as being a tutorial on using Dask for situations you may otherwise use Apache Spark as Dask is based in Python and is thus easier to debug if one does not have experience in Scala.

Illustrates the intended use of the delayed function is as a decorator, and calling it directly will generate a task graph instead of running the code. The task graph can can then be visualized with a call to the visualize function, as well as executed with the compute function.

Example made using simple function calls:

x = delayed (inc) (1)

y = delayed (inc) (2)

z = delayed (add) (x, y) generates task graph

z.visualize() shows task graph

z.compute() executes task graph

Video goes on to provide second example with a for loop rather than a sequence of differing instructions.

For loop seems to be limited in speed up by number of cores, further personal testing could be beneficial.

Tutorial on use of Dask with dataframes is also provided, functions similarly to Pandas dataframes but allows for use with greater datasets as distributed memory helps handle dataset size constraints.

Use of Dask similar to with function calls, direct call will generate task graph, must call compute to get result.

Video proceeds to cover application of Dask to machine learning, given lack of applicability to project it can be disregarded, utility of video ends at 12:25 minute mark.

Overall provides an example on the use of Dask Delayed functionality through direct calls, despite mentioning the intended use is as a decorator.