Abstract

The need for scheduling algorithms arise from the requirement to perform multitasking, essentially for modern computing systems. Scheduling is the greatest cause that optimizes the objective function that involved with the selection of resources. In scheduling, every aspect of execution is based on decision(s). The general objective of scheduling algorithms is to effectively use the available processors to execute parallel programs, possibly in the least utilization of time. A group of interdependent jobs/tasks forms the workflow application. Scheduling is to map the jobs/tasks on to the collection of heterogeneous resources available in a massive geographic spread. Most complicated applications consist of interdependent jobs that coordinate to solve a problem. The completion of a particular job is the criterion function that has to be essentially met in order to start the execution of those jobs that depend upon it [1]. This kind of workflow application may be represented in the form of a Directed Acyclic Graph (DAG). Grid Workflow is such an application and is modeled by DAG. DAGitizer could generate DAG in a random fashion [2]. This paper proposes a tool (developed using Java) which is an enhanced form of DAGitizer that could find the least cost path between source and destination nodes.
Enhanced DAGitizer for Grid Computing through the Discovery of Least Cost Path


**Index Terms**
- Computer Science
- Algorithms

**Keywords**
- Grid Workflow
- Scheduling
- Directed Acyclic Graph
- Randomizer
- Communication
- Cost
- Computation Cost
- Cost Matrix
- Least Cost Path