

Abstract

An approach to efficiently schedule heterogeneous tasks in a distributed environment is presented. Given a set of tasks each with varied CPU and main memory requirements, and a cluster of compute nodes (which are significantly less than the number of tasks), our goal is to find an assignment of tasks to compute nodes such that the total time taken to execute all the tasks is minimized. The task assignment problem in general is NP-Hard and it is further complicated by the changing dynamics (changes to CPU and main memory availability) of the compute nodes. Our solution methodology involves the following: a) develop an analytical model that will determine the upper and lower bounds on the efficiency of a compute node, given the number of processor cores in each compute node, the number of threads in execution, aggregate CPU load, and main memory and CPU availability and b) using these bounds and some properties of the tasks (CPU and main memory utilization values) two tasks assignment models are proposed and extensively evaluated empirically. The key challenge is to determine the CPU availability for a new task. We have proposed a new model (composite) to derive these bounds and experiments have shown that our derived bounds are consistently tight. For all our empirical evaluations on UNIX systems, we have used tasks that are both synthetic (allowing us to control the CPU and memory requirements) for validating composite model and real-world tasks such as prime number generator, merge sort, image rendering, and others for validating task assignment models.