Abstract

Computational Grids aim to make high performance computing available to users that do not have enough computational resources locally, by moulding geographically distributed resources into a single computing environment. However, these distributed resources are typically heterogeneous, non-dedicated, and are provided without any performance or availability guarantees. All of this makes the development of efficient Grid applications, capable of harnessing the benefits of such an ambient, a difficult and complex job. In this work, we propose a new approach that reduces the difficulty of Grid-enabling existing applications and which also provides a means for their efficient execution.

Nowadays, most developers of Grid middleware adopt a model based around a Resource Management System (RMS) whose objective is to maximize the utilization of the available Grid resources. One of the principal functions of the RMS is to provide services like scheduling and fault tolerance. Unfortunately, the decisions taken by the RMS are based solely on the state of the system. In the methodology proposed, each user application is fitted with an Application Management System (AMS) that provides management services specifically tuned to the needs of that individual application. The EasyGrid AMS is embedded into MPI applications at compile time, resulting in *system-aware* applications (Smart G-Apps). Results show that the new *system-aware* MPI applications are indeed faster than their conventional (cluster-based) counterparts. Also, the EasyGrid AMS has been developed in accordance with a standard implementation of MPI and, since it the AMS does not rely other system installed middleware, thus affords Smart G-Apps a greater degree of portability. Furthermore, no modifications to the user's source code are required.