Formal analysis of existing checkpointing systems and introduction of a novel approach

József Kovács

One of the important goals of the Grid is to provide a dynamic collection of resources from which applications requiring very large computational power can select. In that sense Grid is a natural extension of the concepts of supercomputers and clusters towards a distributed and dynamic parallel program execution platform and infrastructure.

Due to the huge number of collaborating execution components the Grid is an inherently dynamic and un-reliable execution environment and hence this environment provides an error-prone system. Any component can fail at any time which may lead to abnormal termination or erroneous result. A parallel application containing communicating processes running on a cluster containing nodes is endangered on a bigger degree.

There are several methods and systems providing rollback-recovery support for parallel applications to survive any failure related to infrastructure. These systems are designed in different ways to support different goals. Most of them are lying on a checkpoint mechanism. Checkpoint means to save and restore the overall state of a distributed application where each surrounding component holding any state must be accessed and retrieved.

Checkpoint can be implemented at different levels (kernel, user, application), different ways (coordinated, uncoordinated, etc.) and in different components on a cluster (application, message-passing environment, scheduler, os, etc.). Each system providing „checkpoint" support uses different combination of these options.

The aim of this research is to introduce a formal framework in which any checkpoint system can be modelled. The model is focusing on the architectural components of the software and hardware environment of a cluster. The basic components are application, process, cluster and node. Relations and functions are defined among them according to the principles of Abstract State Machines that emphasize the main important features of the key components related to the checkpoint systems.

In this work the basic checkpoint methods are formalised. It is shown that existing checkpoint system can be defined in this environment and the different systems can be compared to each other while architectural classification of these systems are defined.

Based on the comparison a novel checkpoint approach is defined in this model. The approach aims at defining rules in a way that the resulting checkpoint system does not require any cluster specific checkpoint environment to serve the application in creating and restoring checkpoint information. Checkpoint and resumption of a distributed application can be realised among different clusters without any checkpoint-related restrictions regarding the software environment. This method is able to provide an environment independent checkpoint system combined with unmodified user code. No existing system can give this degree of freedom for cluster developers and programmers at the same time.