Session 4: Distributed Systems II  
Session Chair: G. Danese

In the Distributed Algorithm II session there are three papers. The work *Programming Cooperative Systems in Drago* has been partially supported by Spanish Research Council. The paper describes Drago, an experimental language developed as an extension of Ada for the construction of fault-tolerant and cooperative distributed applications. Drago imposes discipline and gives linguistic support to the main concepts of the group communication paradigm. In the paper the authors focus their attention on the Drago linguistic support for the implementation of distributed cooperative applications and discuss three examples of its use; a useful comparing analysis with related work is also depicted by the authors. Drago explicitly supports two process group paradigms to build distributed applications: replicated process groups and cooperative process groups. Replicated process groups allow for the programming of fault-tolerance applications according to the active replication model, while cooperative process groups permit programmers to express parallelism and so increase throughput. The described implementation runs on Unix Systems and uses a hierarchical Ada library that provides reliable atomic broadcast using an original consensus protocol. The current implementation makes use of the underlying TCP/IP protocols and no measures of performance have been made. Actually, the implementation has been conceived more as a proof.

The paper *Integrating HPF in a Skeleton Based Parallel Language* discusses the integration of HPF with SkIE (Skeleton based Integrated Environment) a skeleton base coordination language implemented on top of MPI (Message Passing Interface), which permits to describe complex computational parallel structures. The authors show how HPF can be used inside common forms of parallelism, e.g. pipeline and processor farms, and present experimental results regarding a sample application.

In SkIE, when parallelising a sequential application, the user can reuse large chunks of sequential code written in the most common sequential languages encapsulating them in modules which can then be composed to develop a larger application. In the paper the authors enhance the capabilities of SkIE offering a tool for integrating data parallel parts developed using HPF. A case of study is also presented that demonstrates how the proposed tool can be applied when integrating a data-parallel program written in HPF in a SkIE program. The case of study is a classical FFT 3-D transform which is an application widely used to demonstrate the usefulness of exploiting a mixture of both task and data parallelism. The authors claim that the application achieved performance improvements up to 200 per cent.

The paper *Java objects communication on a high performance network* faces a well known problem for workstation clusters equipped with interfaces to local high performance networking. A high level and effective programming language for such cluster architecture is still missing. Recent work shows the interest in Java for cluster programming. One of the main issues is to handle efficiently the communication of objects to really take advantage of the network speed. The authors present an alternative to the standard serialisation process through the proposal of a Java object communication library. In the paper an experimental library called Expresso is presented that provides efficient communications of Java objects. The proposed transfer scheme does not need memory copy: objects are managed in clusters in order to allow the straight communication of the whole graph of objects. In the paper the authors describe two kinds of object clustering, detail their implementation explaining the memory and the communication management, and report some performance results. The limit of the proposed communication library is that it is at low level requiring the user to handle the clustering of objects. The author are investigating a higher level programming paradigm that can take benefit of Expresso cluster exchanges.