Distributed computing is considered to be one of the most exciting technologies since the invention of the computers. Distributed systems are present during the last couple of decades, but it is expected that eventually this decade will witness a real proliferation in the real life use of distributed computer systems. Keeping in step, system architectures were evolving from a single system to host centric and now distributed systems. Today’s distributed systems are complex structures composed of many types of hardware and software components resulting a heterogeneous system. A new line of distributed systems is characterized by direct coupling of application programs running on multiple platforms in a networked environment. New paradigms as cluster of computers and geographically distributed computing grids appeared. For the proper application environment a new notation of Single System Image became an important requirement i.e. the illusion created by software and hardware that a collection of computing elements is a single computing resource.

Parallel processing is one of the most effective mechanisms for increasing performance of large systems. Parallel processing is evolving from uniprocessors to symmetric multiprocessing, from massively parallel processing to distributed multiprocessing. Components’ granularity of distributed systems includes processing elements, CPUs + cash, complete computers, multi- or supercomputers. Communication abstractions are shared memory, virtual shared memory in a distributed-memory environment, or different forms and sophistication of message passing. Recently object-oriented parallel programming model and the Java language play a very important role in the developments and applications of distributed systems.

The trend of distributed computing moves to cheap and general purpose systems consisting of loosely coupled components build up from single or multiprocessor PCs and workstations. Even specialized traditional supercomputing platforms are substituted by distributed metacomputing systems.

There are many key issues for distributed computing technologies as reliability, security, scalability, (name and user) transparency, ease of use, performance quality, etc.) A couple of interesting problems of the field are covered by the papers of our session. We have five papers in the session dealing with various aspects of distributed systems.

The first paper by M. Bernaschi and G. Rischelli deals with the collective communication performance of MPI standard-base communication i.e. broadcast for shared memory multiprocessors. It compares the collective communication primitives of the Sun implementation of the MPI standard with routines based on quasi-optimal spanning trees.

The second paper is titled “DVSA and SHOB: Support to Shared Data Structures on Distributed Memory Architectures” by F. Baiardi et al. The paper describes DVSA (Distributed Virtual Shared Areas) which provides abstraction for distributed memory architectures and SHOB (SHared OBjects), a library implementing shared data structures. The paper provides some experimental results to evaluate the performance of DVSA and SHOB libraries.

The third paper by V. Getov et al. deals with message-passing computing with Java. It gives experimental results for performance evaluation and compares Sun E4000, IBM SP-2 and Linux cluster platforms using the EP (Embarrassingly Parallel) and IS (Integer Sort) parallel benchmarks with codes written in Fortran, Java and C. The results are plotted as execution time vs. number of nodes.

The fourth paper is titled “Elimination of Redundant Messages with a Two-pass Static Analysis Algorithm” by A. Girault. It presents optimization for parallel programs with automatic distribution. It eliminates the redundant messages using global data-flow analysis followed by a local elimination. The associated time and memory costs are also calculated.

The last paper of the session is by E. Mehofer and B. Scholz. It provides an optimization technique and parallelization for distributed-memory systems. The optimization is based on a probabilistic flow analysis framework implemented in the C++ language.

Distributed systems are moving from specialized to general-purpose systems and having a more and more important role in the computing systems generally. I hope that this session will provide a meaningful contribution in this process.