Topic 07
Applications on High-Performance Computers

Yoichi Muraoka, Randall Bramley, David F. Snelling, and Harry Wijshoff

Topic Chairpersons

Applications of High-Performance Computers spans a large intellectual area and now includes all the traditional application science and engineering fields. The end goal of all high performance computing research is eventually to support applications, but those applications have traditionally had a strong feedback effect on computer architecture, hardware design, and systems.

This year nine papers were submitted, in areas ranging from weather simulation and prediction to new areas such as bioinformatics. Given the potentially broad spread of topics, what distinguishes the papers accepted for this year's conference is the careful bridging between the application's algorithmic needs and the underlying high performance computing system. Increasingly the efforts of computer architects, compiler writers, software engineers, and software library developers are helping make high performance computing a staple for scientists and engineers. But the papers in this topic session passed the stringent requirement that they not just use high performance techniques, but specifically address how to map the application to the underlying hardware and software systems.

In addition to the traditional areas of high-speed computation and the development of algorithms amenable to it, one theme that has become crucial in recent years is data-intensive computing: how to handle the large volumes of data which scientific and engineering instruments produce, and how to store it. This year's selection of papers also provides valuable insight into the data management issues.

Scanning Biosequence Databases on a Hybrid Parallel Architecture targets an area of increasing importance in high-performance computing, bioinformatics. Scanning sequence databases is possibly the prototypical computational problem in that area, made even more important by the growth of both breadth and depth of genetic database coverage. This paper is an outstanding example of bridging the application area and the underlying hardware and architecture, introducing a unique cluster system that uses a standard PC cluster running Linux, each augmented with a systolic array board. The authors then show how to map the sequence comparison algorithm to the resulting hybrid architecture.

A Parallel Computation of Power System Equations is a good example of the ability to bridge the needs of an application and the underlying hardware and software. Clusters of PCs such as Beowulf systems are increasingly the choice of universities and research laboratories because of their price/performance ratio. Many of these systems use Pentium processors which have SSE instructions — which are essentially vector instructions. This paper shows how to use those effectively in the context of linear system solvers.

R. Sakellariou et al. (Eds.): Euro-Par 2001, LNCS 2150, pp. 358–359, 2001.

Among the most data-intensive computing tasks currently faced by high speed computing is the handling of the flood of data that will result when the Large Hadron Collider at CERN is scheduled to come on-line in 2005. As *Level-3 Trigger for a Heavy Ion Experiment at LHC* describes, a single experiment will produce 15 GBytes/second of data - more than can or should be transferred to permanent storage since only a small number of "events" from the detectors are of interest to physicists. The paper defines the software and hardware architecture being built to recognize interesting patterns in this data flood, and the integration of real-time compression methods on the saved data. Referees agreed that the techniques and ideas used on this problem may well be useful for other applications in other areas with real-time, irreproducible data coming from valuable scientific instruments.

In addition to the real-time acquisition of large volumes of data from instruments, high performance computing simulations produce data which needs to be stored. Even when scientists are only interested in the final results, the increasing use of large scale and distributed platforms makes checkpointing and fault tolerance crucial for reliable computations. *Experiences in Using MPI-IO on Top of GPFS for the IFS Weather Forecast Code* details how to use the MPI-IO standard with IBM's General Parallel File system to store intermediate steps. This is done in the context of the European Centre' for Medium-Range Weather Forecasts' simulation codes which share features common with many emerging parallel applications: hundreds of parallel tasks writing out intermediate time steps every few seconds. This paper provides an understanding of the underlying mechanisms, and timings to help users determine how many nodes should be dedicated to I/O for optimal performance.

In summary, unlike many submitters of papers those in applications areas typically have to satisfy two widely different sets of goals, in furthering both the science in the application area and developing methodologies that constitute high-performance computing research. We thank all of the authors who submitted contributions and succeeded in both goals, and the many referees who helped the committee in spanning the many applications fields represented.