On a hybrid MPI-Pthread approach for parallel branch-and-bound

Juan F. R. Herrera, Leocadio G. Casado
Informatics Department
University of Almería, Agrifood Campus of Int. Excell. (ceiA3)
Almería, Spain
Email: juanfrh@ual.es, leo@ual.es

Eligius M. T. Hendrix
Department of Computer Architecture
University of Málaga
Málaga, Spain
Email: Eligius.Hendrix@wur.nl

Branch-and-bound (B&B) algorithms are widely used to solve Global Optimization (GO) problems with a guaranteed accuracy. In this paper, we focus on multidimensional Lipschitz GO problems [1] and the HECToR supercomputer. Results show performance improvements compared to OpenMP and MPI versions used in previous work [2].

PARALLELIZATION

A pure MPI code is not necessarily the best solution to obtain the maximum performance [3]. When the search space is irregular, load balance problems usually appear and they may deteriorate performance as the number of MPI processes increases.

For the code to scale to a larger number of cores, one solution is to combine MPI with a threaded model which has load balancing capabilities. The POSIX threads (Pthreads) model used here permits dynamic load balancing through thread generation facilitating to handle irregular data structures as those presented in B&B algorithms [4].

A mixture of MPI and Pthreads is studied. An MPI model is used for the initial work distribution and gathering the final solution. A threaded model (with intrinsic dynamic load balancing) is used for parallelism within a node. This will generally produce a code with better scaling properties than the MPI approach without dynamic workload balance.

Every MPI process performs a sequential phase in which the feasible region is divided by face-to-face vertex triangulation. If the number of generated subsimplices is less than the number of processes, a sequential B&B process is initiated until the number of unexamined simplices becomes equal to the number of MPI processes. Then, each MPI process selects the corresponding simplices it is in charge of. In this schema there is no communications between MPI processes. Therefore, upper bounds are not interchanged and the load balancing is static. This model is similar to [2].

Within each MPI process, the execution starts using one thread. A thread can create a new thread if a core is idle, there is enough work to share with the new thread and the maximum number of threads is not reached. The new generated thread will receive half of the simplices stored in its parent [4]. The best upper bound is shared between the threads using a shared variable. Each thread handles its own work-pool.

RESULTS

The algorithm has been coded in C and compiled using the Cray Compiler Suite. The experiments have been carried out on several nodes of HECToR (UK National Supercomputing Service). Each node contains two 16-core 2.3 GHz AMD Interlagos Opteron processors and 32 GB of main memory.

The test problems used to evaluate the parallel versions are taken from [2]. Four-dimensional problems have been solved with an accuracy of $\varepsilon = 0.25L_2$, the rest with $\varepsilon = 0.5L_2$.

Figure 1 shows the behaviour of the hybrid version. The achieved speedup is better than the one reported in [2], but is still less than linear. Due to the small number of MPI processes determining the initial static distribution of work, an acceptable imbalance is achieved. In general, practical problems exhibit less symmetry and benefit from dynamic load balancing with respect to intra-node as well as inter-node balancing.

Further exploitation of the HECToR architecture can be reached by implementing approaches that handle inter-node dynamic load balancing strategies. This will be investigated in the future.
ACKNOWLEDGMENT

The work was carried out under the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission - Capacities Area - Research Infrastructures. This work has also been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117) and Junta de Andalucía (P11-TIC-7176), in part financed by the European Regional Development Fund (ERDF). Support was also given by the ICT COST Action IC0805. Juan F. R. Herrera is a fellow of the Spanish FPU programme.

This work made use of the facilities of HECToR, the UK’s national high-performance computing service, which is provided by UoE HPCx Ltd at the University of Edinburgh, Cray Inc and NAG Ltd, and funded by the Office of Science and Technology through EPSRC’s High End Computing Programme.

REFERENCES


