PARALLEL RAY-TRACER ON SHARED MEMORY MULTIPROCESSOR MACHINES CONNECTED VIA INTERNET

V. Havran

CTU, Fac. of Electrical Eng., Dept. of Computer Science and Engineering
Karlovo nám. 13, 121 35 Praha 2

Key words: computer graphics, parallel ray-tracing, load balancing, Internet computing

Computer graphics applications are very computationaly demanding, especially if we want a high fidelity pictures. The example of such an algorithm is ray-tracing. Therefore are number of solutions, which accelerates the rendering process. When the algorithmical solutions are all utilized and the rendering is still slow for user's application, then only one way of acceleration is parallel implementation of ray-tracer. We have been interested on this topic for several years using various approaches[1]. The last solution which we have developed is joining of computational power of multiprocessors machines using Internet. This situation is depicted in the Fig. 1:

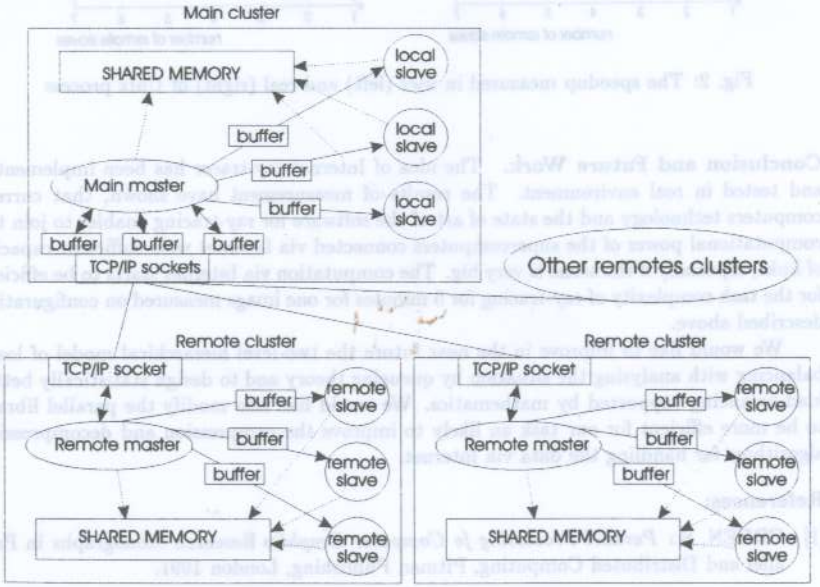

Fig. 1: Architecture of Internet Ray-tracing

Implementation. Two machines with shared memory are computers located in supercomputing centers in Prague and in Brno. Both of them are equipped with 12 processors R10000. To improve the efficiency of this solution we have designed and implemented the buffer technique and two-level control hierarchy for load-balancing via Internet. To decrease the latency of transmission of data the input scene file and the output image are compressed by LZW and RLE. We have designed several scheduling algorithms, which minimize the idleness of remote slaves and therefore increases the speedup. This is done by controlling of the buffer sizes for all local slaves and for remote masters during the ray-tracing process, but further details are out of the scope of this text. For parallelization we have used modified library p4. Two graphs in the Fig. 2 show the speedup of the ray-tracer, when the both machines and network connecting them were heavily loaded. The computation time of ray-tracing for one processor is 350 seconds for given scene. We can see the growth of the speedup, if the number of remote slaves is increased. The speedup should be almost linear, if we will use high speed network.

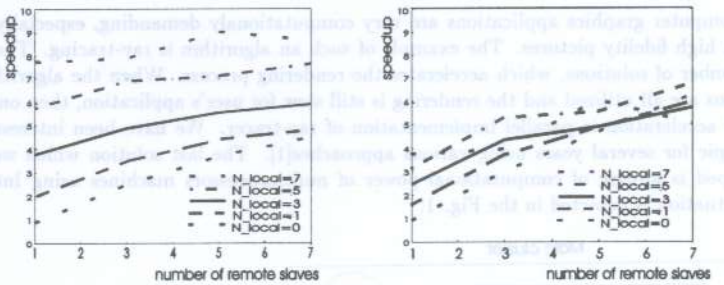

Fig. 2: The speedup measured in user (left) and real (right) of Unix process

Conclusion and Future Work. The idea of Internet ray-tracer has been implemented and tested in real environment. The results of measurement have shown, that current computers technology and the state of art of the software for ray-tracing enables to join the computational power of the supercomputers connected via Internet with sufficient capacity of links, especially if the scene is very big. The computation via Internet starts to be efficient for the task complexity of ray-tracing for 5 minutes for one image measured on configuration described above.

We would like to improve in the near future the two-level hierarchical model of load-balancing with analysing the situation by queueing theory and to design statistically better load-balancing supported by mathematics. We would like also modify the parallel library to be more efficient for our task an likely to improve the compression and decompression algorithms for handling the data via Internet.

References:

[1] GREEN, S.: *Parallel Processing fo Computer Graphics* Research Monographs in Parallel and Distributed Computing, Pitman Publishing, London 1991.