Název: Load Balancing Techniques for Ray-tracer on Shared Memory Machine

Autor: Vlastimil HAVRAN

Školitel: Doc. Ing. Pavel SLAVÍK, CSc.

Pracoviště: Department K336, FEL-ČVUT
Adresa: Karlovo nám. 13, Praha 2
Tel.: 2435 7311
E-mail: havran@cs.felk.cvut.cz

The ray-tracing has become very popular method in computer graphics. The method enables the sufficient grade of fidelity. Its main drawback, which holds liable to the most of the algorithm used, is its time complexity. This is caused by huge number of intersections the rays with the objects in the scene. There are some approaches to speed up the computation. The first one lies in the decrease of calculation by some smarter subdivision of the scene onto the cells. Although this approach speeds up the rendering up to 100 times or more, it is not still enough for practical applications. Therefore the ray-tracing algorithm should be parallelized by some ways. There are more methods how to parallelized ray-tracer. The first one subdivides the scene space onto the cells and to each processor is assigned one of them. The second one consists in the subdivision of the image space onto rectangular areas and each processor computes some of them by some strategies. The problem of the latter, although it is more efficient, lies in the load balancing of the processors. Each of the area of the same size can be completely different computational complexity. Therefore I have implemented five strategies to examine, which of them is the most suitable for ordinary user. The tests were performed on Alpha 8400 cluster system with 8 processors Alpha 21164 with shared memory. The methods for the image subdivision are following:

1) n horizontal bands, where n is the number of processes
2) i x j rectangles, each of them is subdvided into rectangles assigned to the processors
3) i x j rectangles, they are assigned to the processors on demand
4) each processor computes the color of n-th pixel of the row
5) each processor computes the color on n-th row

Properties of these methods influences the speed up of the rendering a lot. The second requirement on the computation scheme is the neighbourness of two pixels evaluated in succession to preserve the data in the processor cache. It can decrease the computation, even if the computational complexity does not change.

Summary:

Measurements performed on real unix machine have shown, that for such a small number of processor and the image size 512x512 the best strategy is number 5). The tests were performed on the granularity of unix processes. For parallelization was used the library developed specially to these purposes.