An Efficient Static Assignment Parallelization Scheme for Algebraic Fractals

F. Chris MacPhee and Virendrakumar C. Bhavsar
Advanced Computational Research Laboratory, Faculty of Computer Science,
University of New Brunswick, Fredericton, NB, E3B 5A3, Canada
Chris.Macphee@unb.ca, bhavsar@unb.ca; http://www.cs.unb.ca/acrl/

ABSTRACT

Although parallelization of algebraic fractal computations has been done in the past, the issue of efficient parallel computation has not been fully addressed in the literature. The objective of this paper is to examine the computational characteristics of algebraic fractal computations and determine an efficient scheme for parallel computation.

1. Introduction

Generating fractal images from the self-squared function, $z \leftarrow z^2 + c$, is discussed widely in the literature (for example, see [1]). Gujar and Bhavsar [2] have proposed the generation of algebraic fractals with positive and negative, integer as well as real, exponents of z. Many techniques for the vectorization [4] as well as parallelization [chap. 3, 6] of fractal computations are given in the literature.

In this paper, we analyze the computational characteristics of the generation of fractal images for $\alpha > 0$ and present parallelization strategies. The efficiencies of these strategies, when implemented on an IBM SP parallel computer system, are found out.

2. Computational Characteristics

We consider the following parameters for fractal computations: the pixels map is 1024 × 1024 with $|Re(z)| \leq 1.5$ and $|Im(z)| \leq 1.5$, the maximum allowable number of iterations per pixel is 100, and z is said to diverge if $|z| > 10$.

The geometric properties of $z \leftarrow z^{\alpha} + c$ fractals with $\alpha > 0$, as given in [2], allude to the fact that an efficient static work assignment model could be produced. To examine the fractal images, we partition the image into 1024 columns, where the number of iterations of z for each pixel in the column is summed. Various values of α between 1 and 100,000 are analyzed.

Figure 1 gives the distribution of the number of iterations for various columns. It can be seen that the distribution tends toward a parabolic curve as α increases. This curve is approximated using the data given by $\alpha =$ 100000. As seen in the figure, the number of iterations required for generating the fractal image is better approximated by the function as α increases.

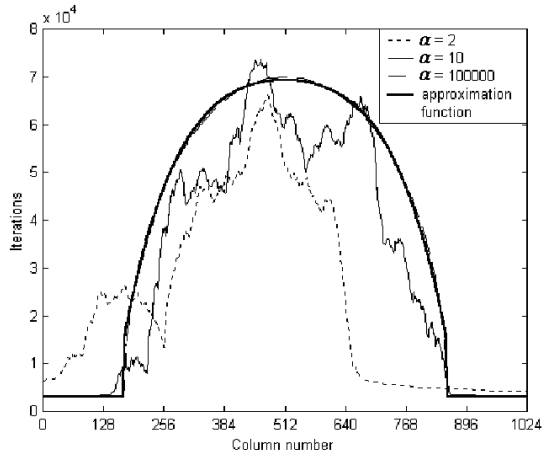

Figure 1: Distribution of the number of iterations for α = 2, 10, 100000 and the approximation function

3. Parallelization

The two parallel fractal programs given in [5], that use static and dynamic work assignments, were modified to conduct the experiments. In the dynamic work assignment program, the master parses work to the slaves in work sizes of 64 columns. The static work assignment program divides column groups evenly between processors. We modified this program by approximating the workload with the calculation of the area under the approximation curve and dividing the workload, rather than columns, evenly over the processors.

Experiments were performed on the IBM SP in the Advanced Computational Research Laboratory of the University of New Brunswick (http://www.cs.unb.ca/acrl/). The IBM SP is configured with 4 compute nodes, with each node containing 2 × 2-way 375 MHz POWER3 64-bit Winterhawk II Processor cards and 1 GB of memory. A total of 9 processors were used during the experiments, configured as 1 master and 8 slaves.

4. Experimental Results

Figures 2-4 show the computing time of the static, dynamic, and the new static work assignment programs for each slave processor for various values of α. The static assignment program results in more work to some processors. As α increases, computing time curve program becomes more bell-shaped. The dynamic assignment program tries to allocate work evenly over all processors. The new static assignment program attempts to allocate work evenly to all processors, but is unable to properly estimate the workload for small values of α. As α increases, the work distributes more evenly. Note that the new static assignment scheme will have a smaller overhead compared to the dynamic scheme. Additional details of the new scheme and results for $\alpha<0$ are included in [7].

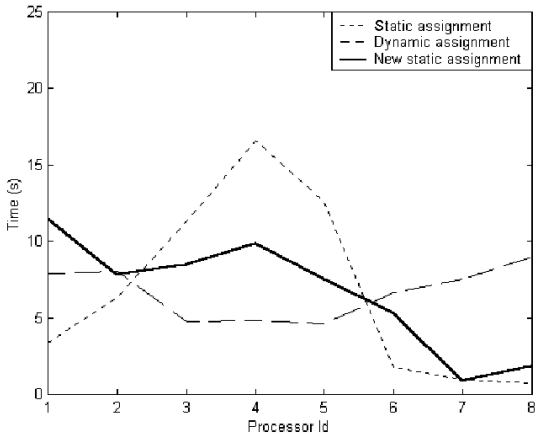

Figure 2: Computing time for each slave processor for $\alpha = 2$

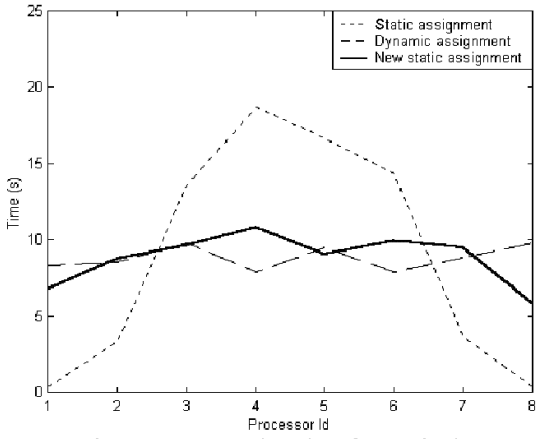

Figure 3: Computing time for each slave processor for $\alpha = 10$

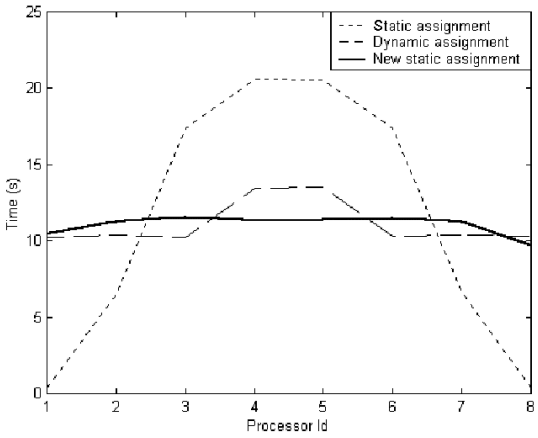

Figure 4: Computing time for each slave processor for $\alpha = 100000$

5. Conclusion

The computational characteristics of fractal images have been analyzed and a static assignment method for efficient parallel processing is developed. Since the approximation function more accurately estimates the workload for larger values of α, the efficiency of the new static assignment program increases as α increases.

References

[1] H. O. Peitgen and P. Richter, *The Beauty of Fractals*, Springer-Verlag, Berlin, 1996.

[2] U. G. Gujar and V. C. Bhavsar, "Fractals from $z \leftarrow z^{\alpha} + c$ in the Complex z-plane", *Comp. and Graph.*, 16(1), pp. 45-49, 1992.

[3] S. V. Dhurandhar, V. C. Bhavsar, and U. G. Gujar, "Analysis of z-plane fractal images from $z \leftarrow z^{\alpha} + c$ for $\alpha < 0$", *Comp. and Graph.*, 17(1), pp. 89-94, 1993.

[4] V. C. Bhavsar, U. G. Gujar, N. Vangala, "Vectorization of generation of fractals from $z \leftarrow z^{\alpha} + c$ on IBM 3090 / 180VF", *Comp. and Graph.*, 17(2), pp. 169-174, 1993.

[5] E. Aubanel, "Parallel Programming with Generalized Fractals," Faculty of Computer Science, University of New Brunswick, , February 2002, http://www.cs.unb.ca/profs/aubanel/aubanel_fractals.html.

[6] B. Wilkinson and M. Allen, *Parallel Programming*, Prentice Hall, Upper Saddle River, 1999.

[7] F. Chris MacPhee and V. C. Bhavsar, Efficient Parallelization of Algebraic Fractals, Technical Report, Faculty of Computer Science, University of New Brunswick, Fredericton, N.B., Canada, in preparation.