

Concurrency - Objects, Algorithms, Formal Methods

Researching concurrent objects properties, (designing) new structures and so on.

Concurrent Bisimulation Algorithm

Abstract:

The coarsest bisimulation-finding problem plays an important role in the formal analysis of concurrent systems. For example, solving this problem allows the behavior of different processes to be compared or specifications to be verified. Hence, in this paper an efficient concurrent bisimulation algorithm is presented. It is based on the sequential Paige and Tarjan algorithm and the concept of the state signatures. The original solution follows Hopcroft's principle „process the smaller half“. The presented algorithm uses its generalized version „process all but the largest one“ better suited for concurrent and parallel applications. The running time achieved is comparable with the best known sequential and concurrent solutions. At the end of the work, the results of tests carried out are presented. The question of the lower bound for the running time of the optimal algorithm is also discussed.

The complete source code distribution together with supplementary libraries and test data can be downloaded [here](#).

file dokuwiki_fancy_my.tpl does not exist

Concurrent Van Emde Boas Array

Abstract:

Increasing demand for computationally efficient algorithms and processors has turned the attention of researchers toward parallel and concurrent solutions. Because the frequency of contemporary processors cannot be tweaked infinitely, the only hopes for squeezing more performance from computers are parallel processing and parallel computation. The important part of every parallel solution is concurrent data structures, which allow multithread programming environments to be taken advantage of. In this article, a new concurrent dynamic set structure is proposed. It is based on the van Emde Boas trees concept, where on every node of a tree, an array of the node's children is stored. The structure is equipped with a simple but effective locking algorithm, which allows it to be used concurrently by any number of threads. The presented algorithm idea is accompanied by an experimental implementation written in JAVA 6. Preliminary tests prove that, especially for moderately larger data sets with a predominance of read operations, the concurrent van Emde Boas array proposed in this article may be a viable alternative for other structures providing a similar functionality

file dokuwiki_fancy_my.tpl does not exist

Dynamic Concurrent Van Emde Boas Array

When the above article was submitted, it turned out that some important aspects of Concurrent van Emde Boas array might be improved. These observations, critical reviews received from the

anonymous reviewers, and above all, the desire to make the structure more practical and handy for regular programmers prompt me to design a new structure. An article entitled „On dynamic concurrent van Emde Boas array” has just been submitted for review.

file dokuwiki_fancy_my.tpl does not exist

Two Concurrent Algorithms of Discrete Potential Field Construction

Abstract:

Increasing demand for computational power in contemporary constructions has created the need to build faster CPUs and construct more efficient algorithms. In this context especially the concurrent algorithms seem to be very promising. Depending on the number of available CPUs they may offer significant reductions in computation time. In this article two concurrent algorithms of potential field generation are proposed. They present two different approaches to problem domain partitioning called by the authors respectively as tearing and nibbling. It is shown that depending on the problem topology either Tear algorithm or Nibble algorithm is faster. Conclusions are summed up in form of experimental results presenting how the algorithms work in practice. However algorithms construct a discrete potential field according to some specific scheme, there should be no major problems with generalization them to other potential field schemes.

file dokuwiki_fancy_my.tpl does not exist

From:

<http://home.agh.edu.pl/~kkulak/> - **Konrad Kułakowski's Home Page**

Permanent link:

<http://home.agh.edu.pl/~kkulak/doku.php?id=user:konrad:research:concurrency>

Last update: **2016/05/12 18:52**

