The Parallel Search Bench ZRAM and its Applications

Adrian Brüngger and Ambros Marzetta,
Institut für Theoretische Informatik,
ETH Zentrum, 8092 Zürich

November 14, 1996

Abstract

The solution of optimization problems can consume huge amounts of computing power. This power is available in the form of parallel computers, but it is of use to non-specialists only if computer scientists supply the necessary software in the form of libraries. We have developed ZRAM, a library of parallel search algorithms including branch and bound and reverse search. The combination of ZRAM with the power of the Cenju-3 has led to new results in three areas: the quadratic assignment problem, vertex enumeration and the 15-puzzle.

1 Combinatorial optimization and parallel computing

Optimization problems occur in many fields—almost everywhere energies or costs have to be minimized. Depending on the nature of the constraints and the objective function, as well as on the desired quality of the result (optimal versus near-optimal), different algorithms are used to solve such problems. Our main interest lies in problems for which an optimum can be found only by searching a large state space exhaustively.

The huge complexity of such search algorithms suggests the use of parallel computers, but how many optimization specialists are also specialists in parallel programming? This gap can be bridged by separating an optimization algorithm into an application-specific and an application-independent part. The former can be developed on a sequential computer, and the latter can be taken from a library.

The libraries of parallel search algorithms currently being constructed at several universities are, in general, limited to branch and bound. With our portable library ZRAM, we aim at a broader range of applications. Although it is still under development, ZRAM has already demonstrated its efficiency and usefulness by giving new results in various areas.

2 The ZRAM library

ZRAM [2] is an extensible library of parallel search algorithms. It has four layers (Figure 1) separated by clearly defined interfaces.

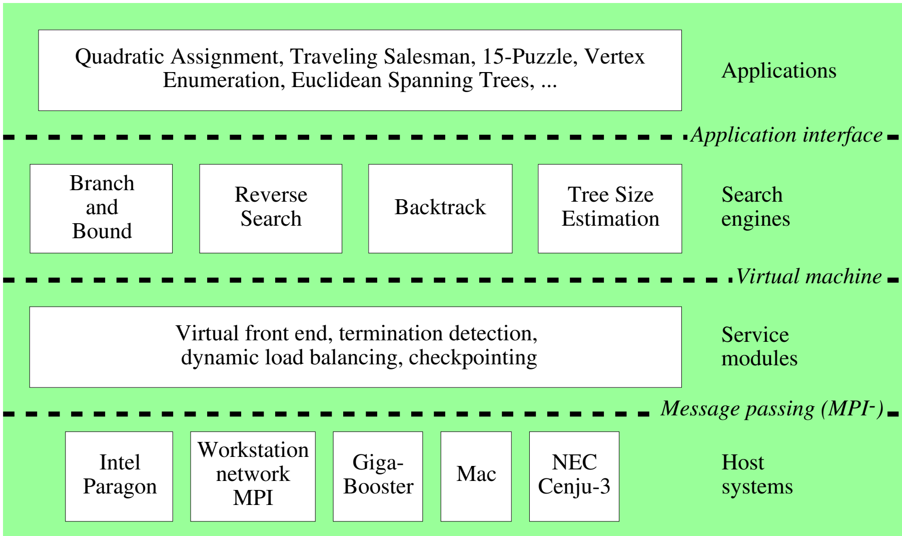

Figure 1: ZRAM architecture

- The *host system* layer hides all machine dependencies. ZRAM is portable to a broad range of message-passing machines.
- The *virtual machine* abstracts from the hardware and contains higher-level functionality convenient for programming parallel search algorithms, such as dynamic load balancing, distributed termination detection, and checkpointing. Computations can be interrupted and continued later with a different number of processors, which makes ZRAM suitable for long-term computations.
- The *search engine layer* contains parallel search engines for branch and bound, reverse search, and backtrack, as well as a general implementation of Knuth's tree size estimation.
- In the top layer, a variety of applications demonstrate the usability of the ZRAM interfaces. This *application layer* generally contains no explicit parallelism.

3 Three Applications

The **quadratic assignment problem** [1] is often formulated as the problem of assigning facilities to locations such that the total cost of the assignment is minimized. The cost is defined as the product of a given flow between every pair of facilities and the distance of their locations.

We ported a branch-and-bound algorithm to ZRAM, and ran it on a set of standard test instances. Amongst them 10 unsolved problems could be solved on the Cenju-3 and on the Paragon. The largest solved instance has about $5 \cdot 10^{10}$ nodes in the search tree, and its solution took about 12 days with a varying number of processors involved.

The **vertex enumeration problem** consists of enumerating all vertices of a d-dimensional convex polyhedron defined by a set of m linear inequalities. Our parallel reverse search code was able to solve three large polytopes on the Cenju-3 with 64 processors (Table 1). These instances could not be solved on any other computer.

Name	d	m	vertices	bases	time
01torus15x	14	240	101445	409794857	3 days
mit71-61	60	71	3149579	57613364	4.5 days
01torus16x	15	340	519275	3971059018	38 days

Table 1: Three previously unsolved polytopes

Sam Loyd's 15-puzzle consists of a permutation of 15 tiles and an empty space over a 4×4 board. Any tile adjacent to the empty space can slide into the empty space in a single move. A branch-and-bound algorithm for finding a minimum sequence of moves leading from any given starting permutation to the goal state runs for seconds to days.

We aimed at a higher-order problem: finding the hardest 15-puzzle positions. The state space of this problem contains about 10^{13} positions and was a target for a new milestone in exhaustive search. Our computations proved that there is no position requiring more than 80 moves. Furthermore we found a few new positions requiring 80 moves.

Acknowledgements. We thank David Avis, Jens Clausen, Komei Fukuda, Anton Gunzinger, Michael Perregaard, Will Sawyer, and the Swiss Center for Scientific Computation for their contribution to this work.

References

[1] A. Brüngger, A. Marzetta, J. Clausen, M. Perregaard, Joining Forces in

Solving Large-Scale Quadratic Assignment Problems in Parallel, submitted to 11th International Parallel Processing Symposium

[2] A. Brüngger, A. Marzetta, K. Fukuda, J. Nievergelt, The Parallel Exhaustive Search Workbench ZRAM and its Applications, Annals of Operations Research, to appear.