Aurora and Friends on the Sun (Abstract)

Manuel Eduardo Correia, Fernando Silva and Vítor Santos Costa*
Laboratório de Inteligência Artificial e Ciência de Computadores
University do Porto
R. Do Campo Alegre, 823
Porto, Portugal

1 Abstract

One of the most important problems in designing or-parallel systems is how to represent environment, ie, bindings, in the different branches of the search tree. One solution to this problem was proposed in the SRI model and is known as the *binding array*. This solution was implemented as the Aurora or-parallel system [3]. The binding array was also used in Andorra-I to accomodate a combination of determinate dependent and-parallism and or-parallelism [4]. Finally, the PBA based implementation of C-trees [2] uses an extension of binding arrays, the concept of *paged binding array* to support both independent and-parallelism and or-parallelism.

Both Aurora and Andorra-I have been implemented for the Sequent Symmetry shared-memory parallel computers. As computer technology moves on, newer and faster parallel machines, particularly shared-memory machines, have been developed. The new parallel machines have at least one order of magniguted faster single processor performance, but at a cost. Performance depends much more on cache efficiency, as access to the shared memory bus is relatively more expensive. The question therefore arises of whether systems such as Aurora can perform well in these new architecures.

One very popular example of the new shared memory computer architecture is the SparcCenter family of computing servers. We have ported Aurora to this machine. In this paper we explain what were the main difficulties in this port and present some performance results.

Porting Aurora to the SparcCenter

Our work in porting Aurora to the SparcCenter benefitted from previous work in porting Aurora to the Butterfly machines and to the Sequent IPX, and from Karlsson and Ali's work in porting Muse [1] to the Sparc Center machine. We found three different category of problems:

Processor: The Sequent uses the Intel I386 chip-family, versus the sparc architecture used in the SparcCenter. The major difference is that the sparc V8 that is used in the SparcCenter does not support atomic operations. Other differences concern the exact instructions used for locking, and endianess.

Operating System Related: Here the major issues were signal handling, shared memory allocation, and saved states.

Compiler Related: We used `gcc` to perform this port. This compiler has a good optimiser which much improves performance, To use these optimisations we needed to add the keyword `volatile` to several definitions in the scheduler.

*Correspondence should be addressed to Vítor Santos Costa, Laboratório de Inteligência Artificial e Ciência de Computadores, Universidade do Porto, R. Do Campo Alegre 823, 4100 Porto, Portugal; telephone: +351 2 6001672x111; fax: +351 2 6003654; email: vsc@ncc.up.pt

We ported Aurora including the Bristol scheduler. Other schedulers, such as the Dharma scheduler, would need similar changes.

Performance

We have run the systems with several tests. Our initial observations suggest that:

- Single-processor performance is about one order of magnitude (around 10 times, but sometimes more) than on the Sequent.
- Quite good speedups (over 5fold for 6 processors) have been obtained.
- In general, speedups are worse than for the Sequent, as expected, but they are still reasonable. The results are particularly good considering that granularity is about 10 times smaller.
- For programs with little parallelism, there can be some slowdown, probably due to contention in the bus.
- Muse performs better than Aurora so far. Sequential performance is about 20-40% better. Speedups from Muse are also better, we suspect this is due to problems with scheduler tuning.

Future Work and Conclusions

We have ported Aurora to the Sun SparcCenter. Our initial results suggest that Aurora performs reasonably well in these new architectures. We believe that we can improve performance further by some tuning in the Aurora scheduler.

Porting Aurora is part of our work in the research of parallel systems for logic programming. We plan to continue this work by porting Andorra-I, and by developing a logic programming system based on the PBA data model.

References

[1] K. A. M. Ali and R. Karlsson. The Muse Or-parallel Prolog Model and its Performance. In *Proceedings of the North American Conference on Logic Programming*, pages 757–776. MIT Press, October 1990.

[2] G. Gupta, M. Hermenegildo, and V. S. Costa. And-or parallel prolog: A recomputation based approach. *New Generation Computing*, 11(3,4):297–322, 1993.

[3] E. Lusk, R. Butler, T. Disz, R. Olson, R. Overbeek, R. Stevens, D. H. D. Warren, A. Calderwood, P. Szeredi, S. Haridi, P. Brand, M. Carlsson, , A. Ciepelewski, and B. Hausman. The Aurora or-parallel Prolog system. *New Generation Computing*, 7(2,3):243–271, 1990.

[4] V. Santos Costa, D. H. D. Warren, and R. Yang. The Andorra-I Engine: A parallel implementation of the Basic Andorra model. In *Proceedings of the Eighth International Conference on Logic Programming*, pages 825–839. MIT Press, June 1991.