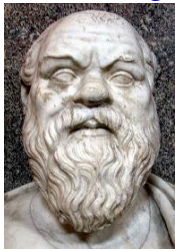


[Home](#) \* [Engines](#) \* [Star Socrates](#)



[Socrates](#) <sup>[2]</sup>

**\*Socrates,**

a parallel chess program, developed by a team from [MIT](#) and [Sandia National Laboratories](#) <sup>[1]</sup> headed by [Charles Leiserson](#). Primary programmers were [Don Dailey](#), co-author of [Socrates](#) and [Titan](#) aka [Socrates II](#), on which \*Socrates was based on, and [Chris Joerg](#). Further contributors were [Robert Blumofe](#), [Matteo Frigo](#), [Larry Kaufman](#), [Bradley Kuszmaul](#), [Keith H. Randall](#), [Eric Brewer](#), [Michael Halbherr](#), [Rolf Riesen](#) (Sandia) and [Yuli Zhou](#).

## Table of Contents

[Massively Parallel Chess](#)

[History of \\*Socrates](#)

[Acknowledgments](#)

[Description](#)

[Awards](#)

[See also](#)

[Publications](#)

[External Links](#)

[References](#)

[What links here?](#)

## Massively Parallel Chess

Quote by [Chris Joerg](#) and [Bradley Kuszmaul](#) <sup>[3]</sup>

Computer chess provides a good testbed for understanding dynamic [MIMD](#)-style computations. To investigate the programming issues, we engineered a parallel chess program called \*Socrates, which running on [NCSA's](#) 512 processor [CM-5](#), tied for third in the [1994 ACM International Computer Chess Championship](#). \*Socrates uses the [Jamboree](#) algorithm to search game trees in parallel and uses the [Cilk 1.0](#) language and run-time system to express and to schedule the computation. In order to obtain good performance for chess, we use several mechanisms not directly provided by Cilk, such as aborting computations and directly accessing the active message layer to implement a global transposition table distributed across the processors. We found that we can use the critical path  $C$  and the total work  $W$  to predict the performance of our chess programs. Empirically \*Socrates runs in time

•

on  $P$  processors. For best-ordered uniform trees of height  $h$  and degree  $d$  the average available parallelism in Jamboree search is

•

\*Socrates searching real chess trees under tournament time controls yields average available parallelism of over 1000.

## History of \*Socrates

History of \*Socrates by [Chris Joerg](#) from his Ph. D. thesis <sup>[4]</sup>

We began work on this program in May of 1994. [Don Dailey](#) and [Larry Kaufman](#) of [Heuristic Software](#) provided us with a version of [Socrates](#), their serial chess program. During May and June we parallelized the program using [Cilk](#), focusing mainly on the [search algorithm](#) and the [transposition table](#). During June Dailey visited [MIT](#) to help tune the program, but we spent most of June simply getting the parallel version of the program to work correctly. In late June, we entered \*Socrates in the [1994 ACM International Computer Chess Championship](#) in Cape May, New Jersey. We ran the program on a 512-node [CM-5](#) at the [National Center for Supercomputing Applications](#) (NCSA) at the [University of](#)

[Illinois](#). Despite the fact that we had begun working on the program less than two months earlier, the program ran reliable and finished in third place.

## Acknowledgments

by [Chris Joerg](#) from his Ph. D. thesis:

I am especially grateful to my thesis supervisor, Professor [Charles Leiserson](#), who has led the Cilk project. I still remember the day he came to my office and recruited me. He explained how he realized I had other work to do but he wanted to know if I would like to help out part time" on implementing a chess program using PCM. It sounded like a interesting project, so I agreed, but only after making it clear that I could only work part time because I had my thesis project to work on. Well, part time" became full time", and at times full time" became much more than that. Eventually, the chess program was completed, and the chess tournament came and went, yet I still kept working on the PCM system (which was now turning into Cilk). Ultimately, I realized that I should give up on my other project and make Cilk my thesis instead. Charles is a wonderful supervisor and under his leadership, the Cilk project has achieved more than I ever expected. Charles' influence can also be seen in this write-up itself. He has helped me turn this thesis into a relatively coherent document, and he has also pointed out some of my more malodorous grammatical constructions.

The Cilk project has been a team effort and I am indebted to all the people who have contributed in some way to the Cilk system: Bobby Blumofe, Feng Ming Dong, Matteo Frigo, Shail Aditya Gupta, Michael Halbherr, Charles Leiserson, Bradley Kuszmaul, Rob Miller, Keith Randall, Rolf Riesen, Andy Shaw, Richard Tauriello, and Yuli Zhou. Their contributions are noted throughout this document.

I thank, along with the members of the Cilk team, the past and present members of the Computation Structures Group. These friends have made MIT both a challenging and a fun place to be. In particular I should thank Michael Halbherr. He not only began the work that lead to the PCM system, but he tried many times to convince me to switch my thesis to this system. It took a while, but I finally realized he was right.

I am also indebted to Don Dailey and Larry Kaufman, both formerly of Heuristic Software. They wrote the serial Socrates program on which \*Socrates is based. In addition, Don and I spent many long nights

debugging, testing, and improving (or at least trying to improve) \*Socrates. Most of this time we even had fun.

Professor Arvind, Dr. Andy Boughton, and Dr. Greg Papadopoulos also deserve many thanks. They provided me the freedom, encouragement, and support to work on a wide range of exciting projects throughout my years at MIT.

## Description

Description given in 1995 from the [ICGA](#)-site <sup>[5]</sup>

The Star Socrates 2.0 chess program developed at the MIT Laboratory for Computer Science, will be running on the 1824 node [Intel Paragon](#) parallel supercomputer located at Sandia National Laboratories. The lead programmers are Don Bailey and Christopher F.Joerg and the project team is lead by Prof. Leiserson. Heuristic Software provided the original Socrates program on which StarSocrates was originally based. The Paragon is about 50 feet long and weighs about 30,000 pounds. Each node consists of two 50MHz [i860](#) processors with either 16 or 32MB of memory. The program currently runs on both the [Connection Machine CM-5](#) and the Intel Paragon.

## Awards

Awards for Cilk Technology & Research <sup>[6]</sup>:

1995, Second Place in the 1995 International Computer Chess Association's Eighth Computer Chess World Championship for \*Socrates 2.0, a chess-playing program written in Cilk.

## See also

- [Astronomy](#)
- [Cilkchess](#)
- [Legend](#)
- [Socrates](#)
- [StarTech](#)

## Publications

- [Chris Joerg](#), [Bradley Kuszmaul](#) (1994). *Massively Parallel Chess*. Proceedings of the Third DIMACS Parallel Implementation Challenge, 1994, [pdf](#)
- [Robert Blumofe](#), [Chris Joerg](#), [Bradley Kuszmaul](#), [Charles Leiserson](#), [Keith H. Randall](#), [Yuli Zhou](#) (1995). *Cilk: An Efficient Multithreaded Runtime System* Proceedings of the Fifth [ACM SIGPLAN](#) Symposium on Principles and Practice of Parallel Programming (PPoPP), [pdf](#)

## External Links

- [Star Socrates' ICGA Tournaments](#)
- [Socrates from Wikipedia](#)

## References

1. <sup>^</sup> [8th World Computer Chess Championship](#), [pdf](#), Courtesy of [Monroe Newborn](#), [The Computer History Museum](#)
2. <sup>^</sup> [Socrates from Wikipedia](#)
3. <sup>^</sup> [Chris Joerg](#), [Bradley Kuszmaul](#) (1994). *Massively Parallel Chess*. Proceedings of the Third DIMACS Parallel Implementation Challenge, [pdf](#)
4. <sup>^</sup> [Chris Joerg](#) (1996). *The Cilk System for Parallel Multithreaded Computing* Ph. D. thesis, Department of Electrical Engineering and Computer Science, [MIT](#), [pdf](#)
5. <sup>^</sup> [Star Socrates' ICGA Tournaments](#)
6. <sup>^</sup> [Awards for Cilk Technology & Research](#)

## What links here?

Page	Date Edited
<a href="#">ACM 1994</a>	May 5, 2017
<a href="#">Bradley Kuszmaul</a>	Oct 12, 2016
<a href="#">Charles Leiserson</a>	Oct 12, 2016
<a href="#">Chris Joerg</a>	Oct 11, 2016
<a href="#">Cilk</a>	Oct 28, 2013
<a href="#">Cilkchess</a>	Jan 21, 2016
<a href="#">Connection Machine</a>	Oct 12, 2016
<a href="#">Don Dailey</a>	Dec 7, 2016
<a href="#">Engines</a>	Mar 10, 2018
<a href="#">Eric Brewer</a>	Oct 12, 2016
<a href="#">Fritz</a>	Feb 26, 2018
<a href="#">i860</a>	Aug 21, 2015
<a href="#">Jamboree</a>	Jun 15, 2015
<a href="#">Keith H. Randall</a>	Jun 16, 2015
<a href="#">Larry Kaufman</a>	Jan 28, 2018
<a href="#">Massachusetts Institute of Technology</a>	Jan 24, 2017
<a href="#">Mathematician</a>	Apr 9, 2018
<a href="#">Matteo Frigo</a>	Jan 7, 2016

Page	Date Edited
<a href="#">Michael Halbherr</a>	Jun 15, 2015
<a href="#">Robert Blumofe</a>	Jun 15, 2015
<a href="#">Rolf Riesen</a>	Dec 5, 2017
<a href="#">Socrates</a>	Jul 3, 2018
<a href="#">SPARC</a>	Jan 17, 2017
<a href="#">Spector</a>	Nov 11, 2016
<a href="#">Star Socrates</a>	Oct 12, 2016
<a href="#">StarTech</a>	Oct 20, 2016
<a href="#">Titan</a>	Jul 3, 2018
<a href="#">University of Illinois at Urbana-Champaign</a>	Oct 12, 2016
<a href="#">Walter Faxon</a>	Mar 6, 2015
<a href="#">WCCC 1995</a>	Jul 12, 2017
<a href="#">Who's Who</a>	Sep 6, 2017
<a href="#">Yuli Zhou</a>	Jun 15, 2015

[Up one Level](#)