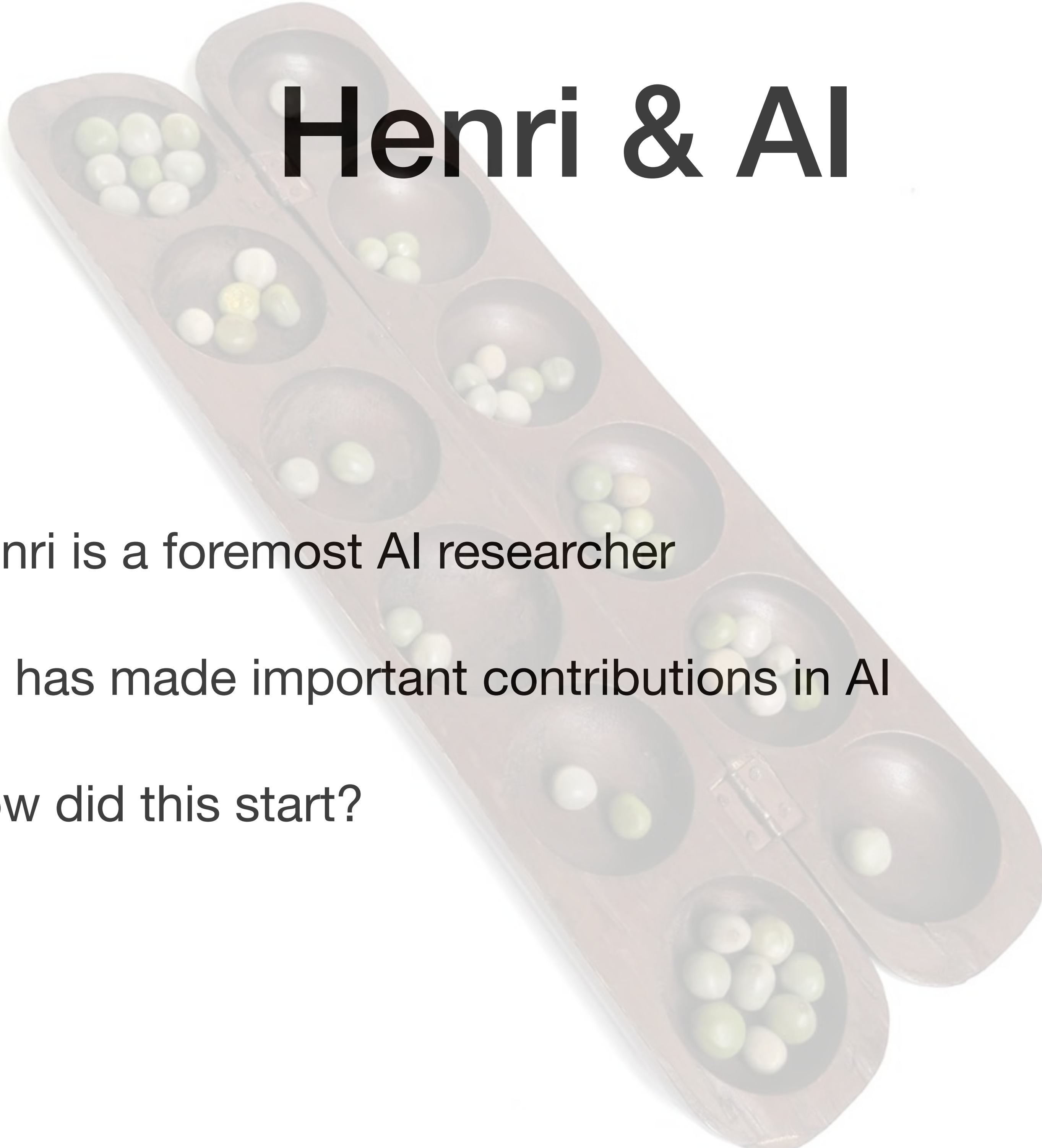
HPC & AI

Aske Plaat

1 mei 2025

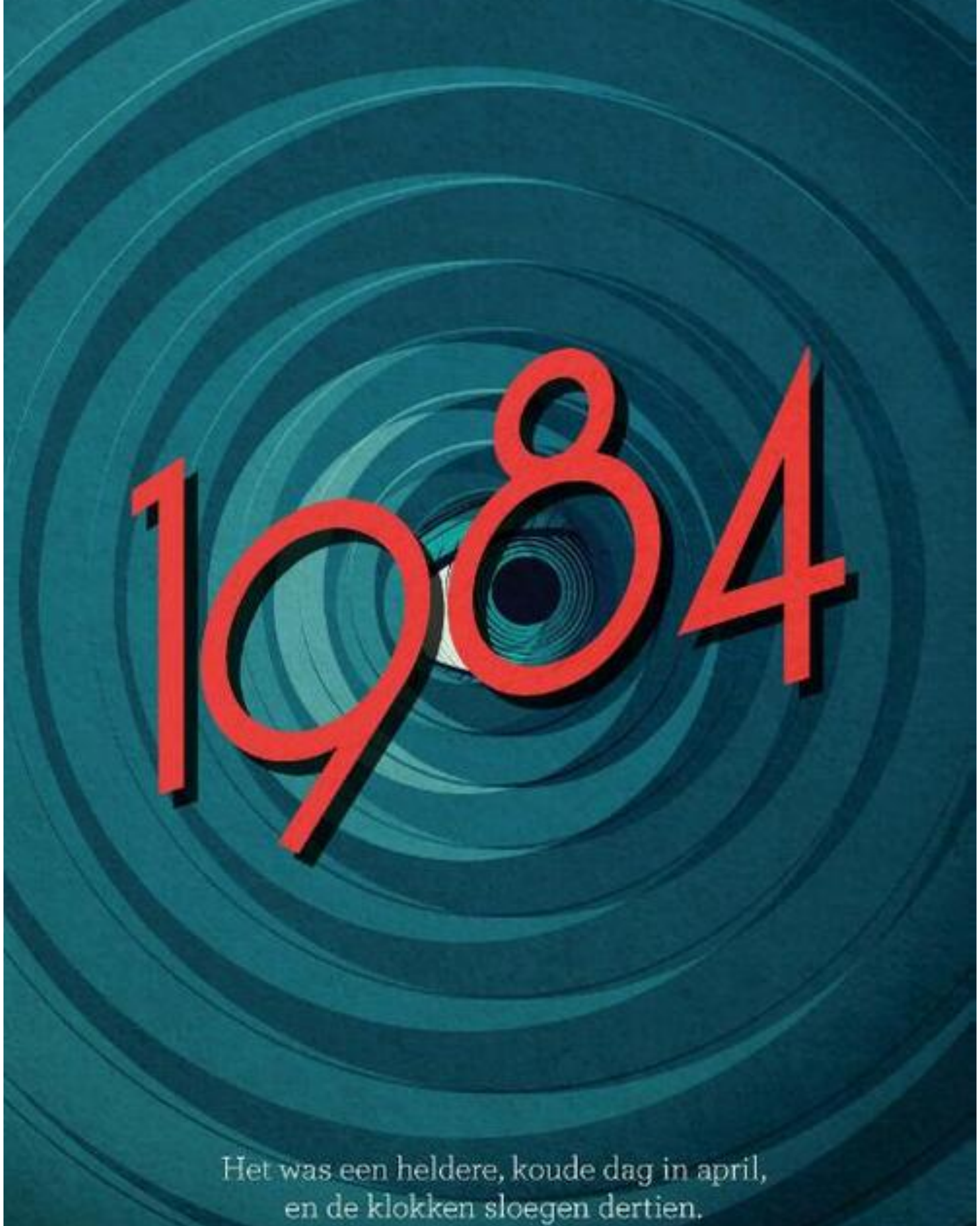
Henri & AI

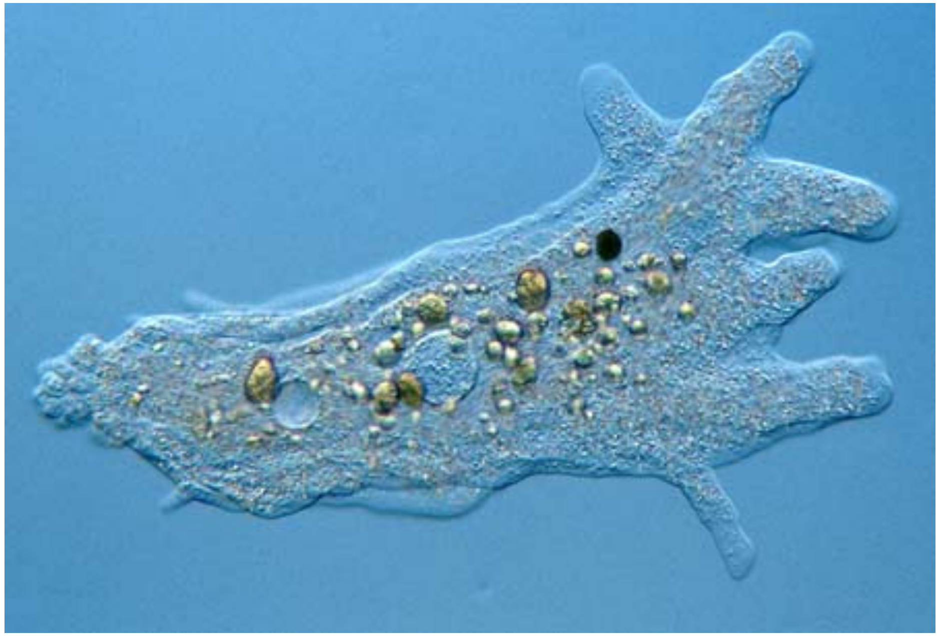
Henri & AI

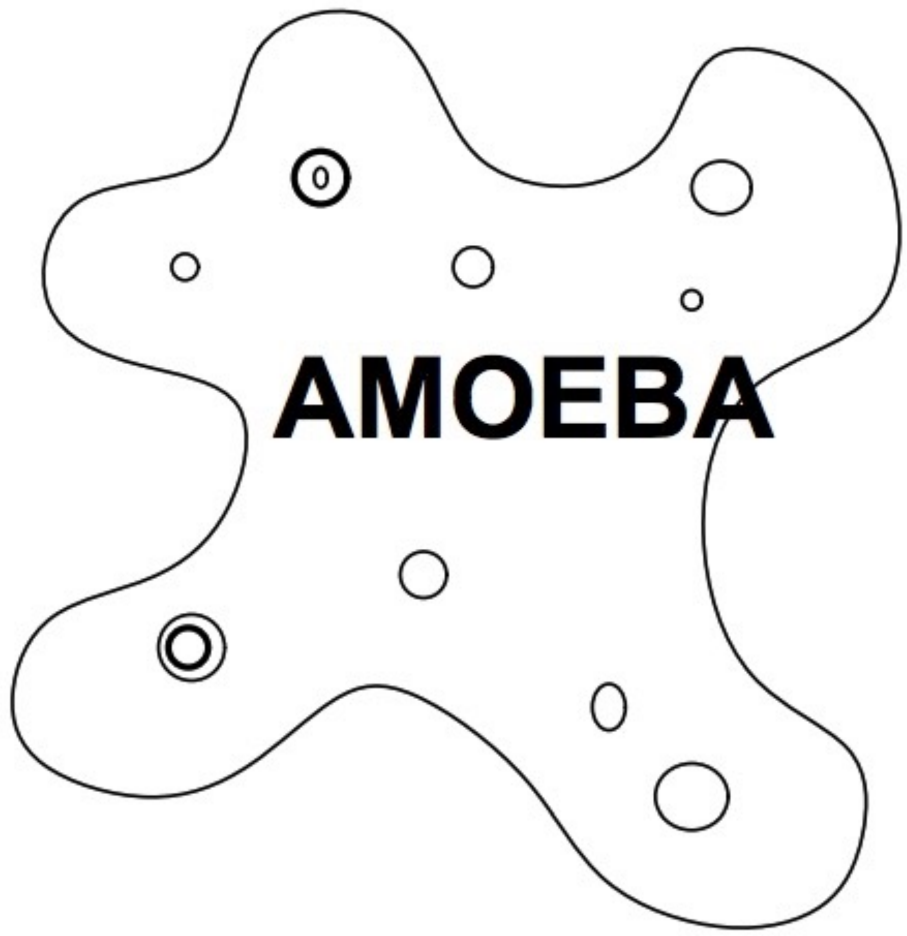


Henri & AI

- Henri is a foremost AI researcher
- He has made important contributions in AI
- How did this start?


1984
Het was een heldere, koude dag in april,
en de klokken sloegen dertien.




AMOEBA

THE SHARED DATA-OBJECT MODEL AS A PARADIGM FOR PROGRAMMING DISTRIBUTED SYSTEMS

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit te Amsterdam,
op gezag van de rector magnificus
dr. C. Datema,
hoogleraar aan de faculteit der letteren,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der wiskunde en informatica
op dinsdag 17 oktober 1989 te 13.30 uur
in het hoofdgebouw van de universiteit, De Boelelaan 1105

door

HENRI ELLE BAL

geboren te 's Gravenhage

Centrale Huisdrukkerij Vrije Universiteit

Amsterdam 1989

THE SHARED DATA-OBJECT MODEL
AS A PARADIGM FOR
PROGRAMMING DISTRIBUTED SYSTEMS

High Performance
High Abstraction

HENRI ELLE BAL

Centrale Huisdrukkerij Vrije Universiteit

Amsterdam 1989

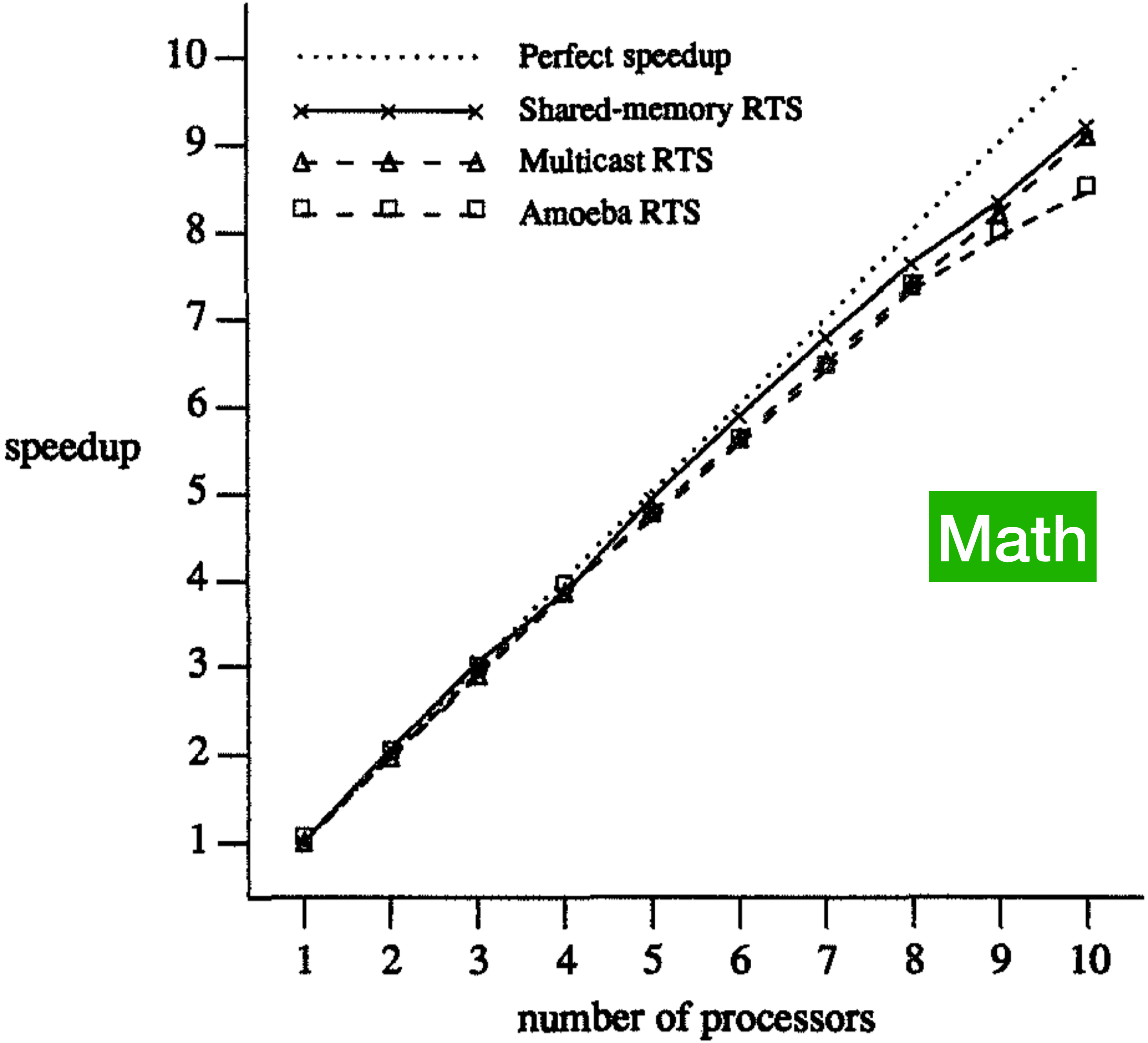


Fig. 7.1. Measured speedups for matrix multiplication, using 250 × 250 integer matrices.

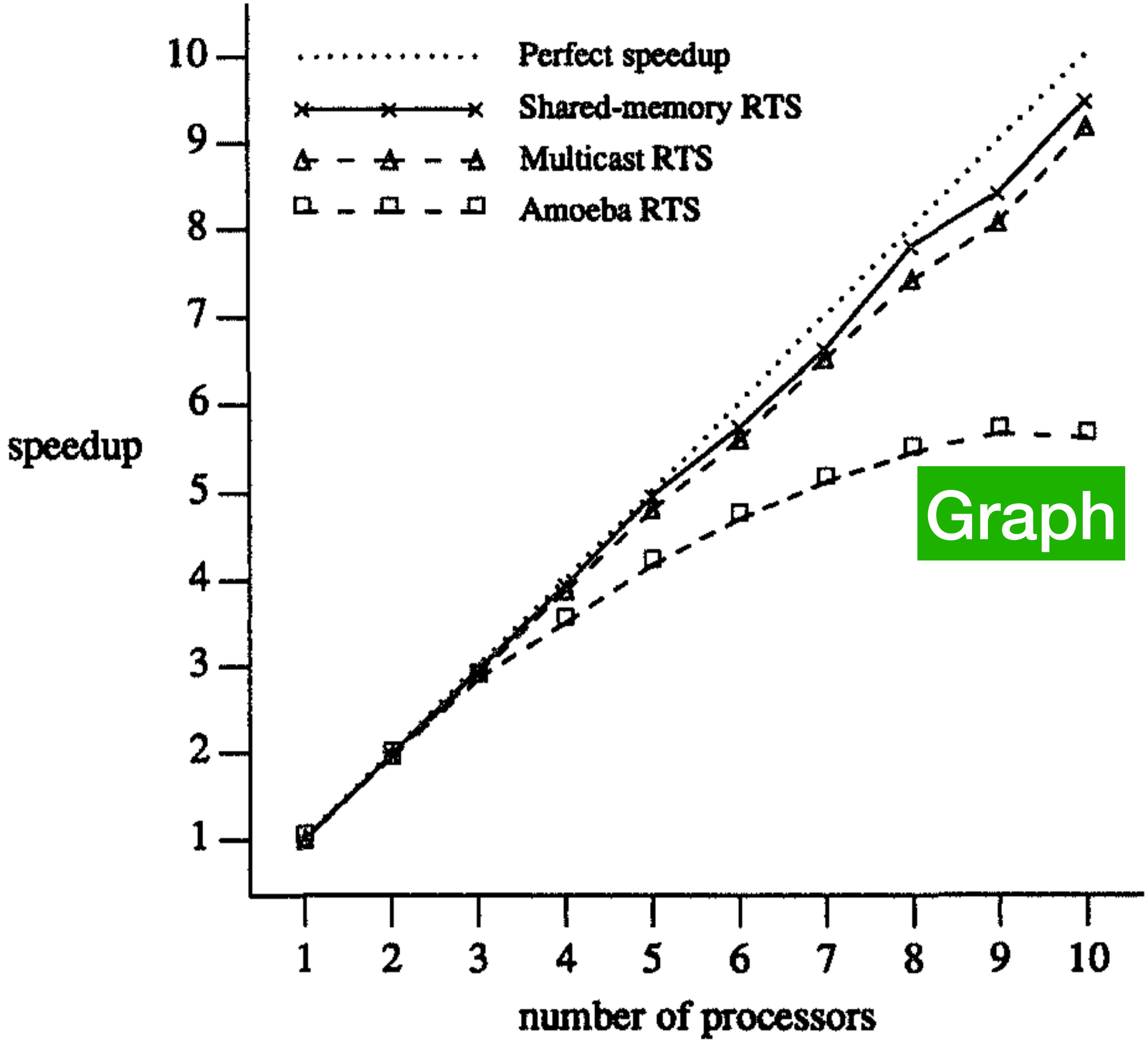


Fig. 7.2. Measured speedups for the All-pairs Shortest Paths problem, using an input graph with 200 nodes.

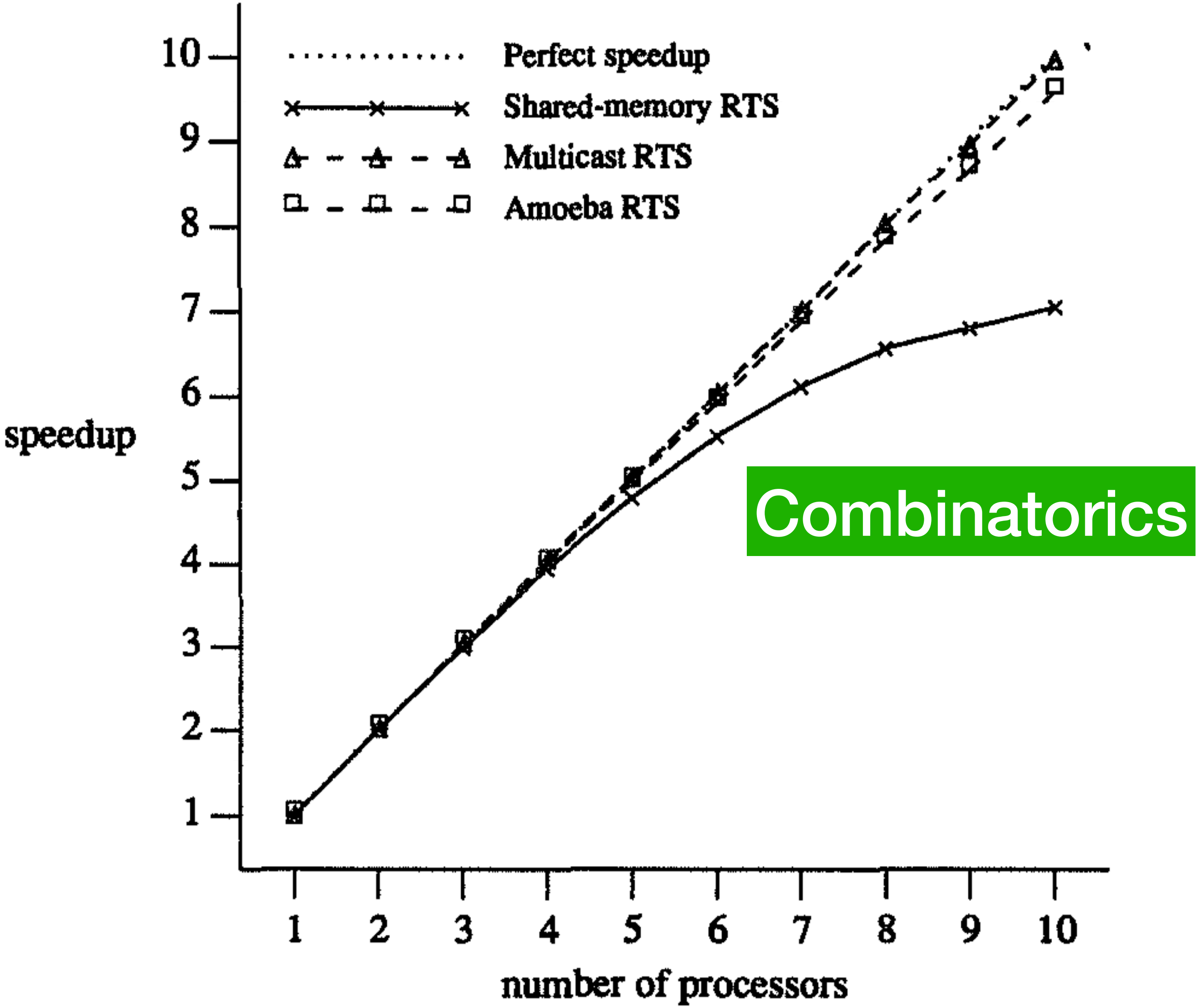


Fig. 7.4. Measured speedups for the Traveling Salesman Problem, averaged over three randomly generated graphs with 12 cities each.

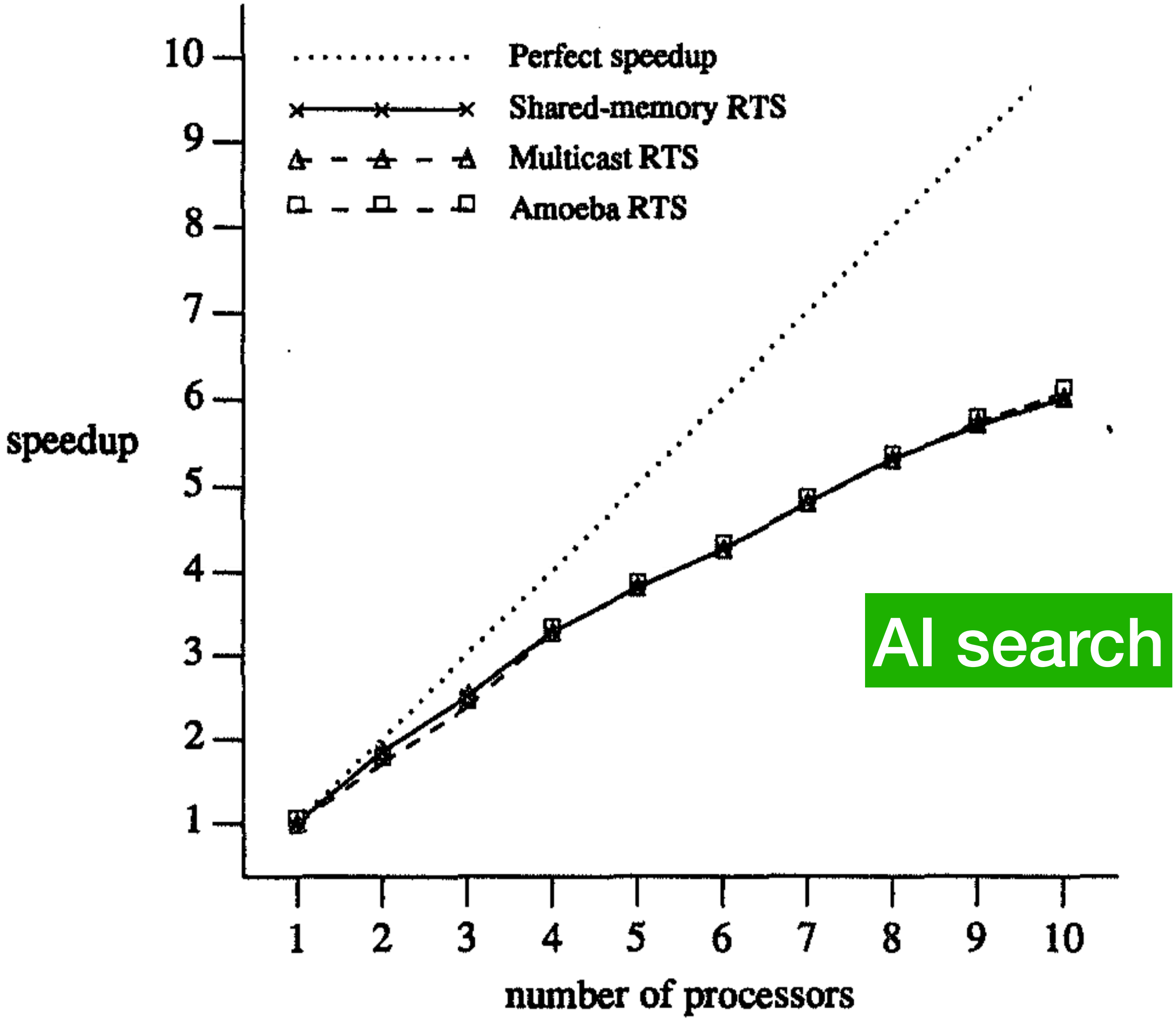


Fig. 7.6. Measured speedups for Alpha-Beta search, averaged over three randomly generated game trees with fanout 38 and depth 6.

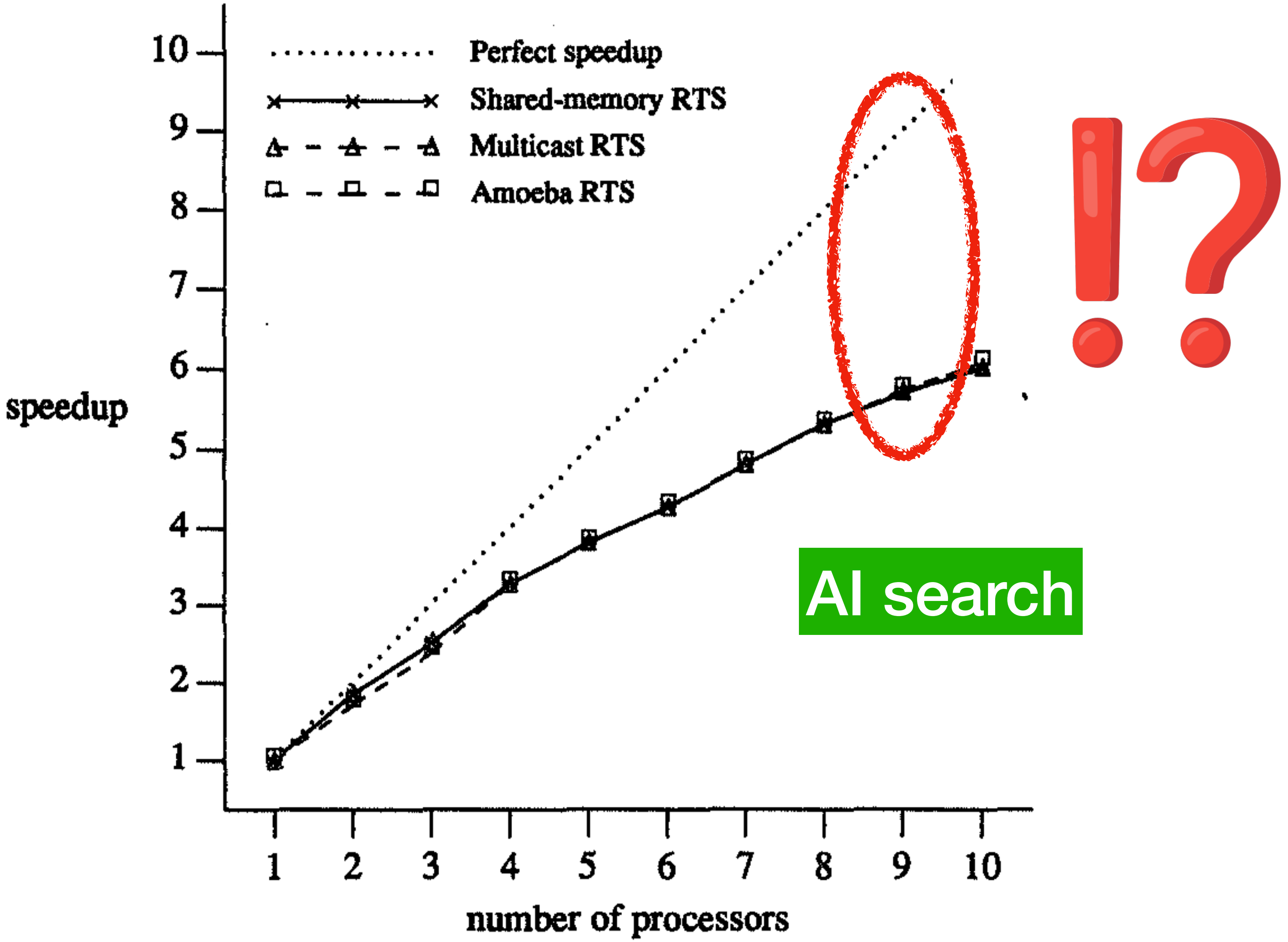


Fig. 7.6. Measured speedups for Alpha-Beta search, averaged over three randomly generated game trees with fanout 38 and depth 6.

Exponential Time
Complex Code
Irregular Parallelism
Fine Grained Communication

1997

1997

Optimizing Parallel Applications for Wide-Area Clusters

Henri E. Bal Aske Plaat Mirjam G. Bakker Peter Dozy Rutger F.H. Hofman
Dept. of Mathematics and Computer Science
Vrije Universiteit, Amsterdam, The Netherlands
http://www.cs.vu.nl/albatross/

Abstract

Recent developments in networking technology cause a growing interest in connecting local-area clusters of workstations over wide-area links, creating multilevel clusters, or meta computers. Often, latency and bandwidth of local-area and wide-area networks differ by two orders of magnitude or more. One would expect only very coarse grain applications to achieve good performance. To test this intuition, we analyze the behavior of several existing medium-grain applications on a wide-area multicluster. We find that high performance can be obtained if the programs are optimized to take the multilevel network structure into account. The optimizations reduce intercluster traffic and hide intercluster latency, and substantially improve performance on wide-area multiclusters. As a result, the range of metacomputing applications is larger than previously assumed.

Keywords*: Meta computing, Wide-area networks, Communication patterns, Parallel algorithms*

1 Introduction

ingly parallel applications) will benefit from multiple WAN-connected clusters. The research question we address here is *how parallel applications perform on a multilevel network structure*, in particular, on systems built out of both LANs and WANs. Existing meta computing projects often use applications with very coarse-grained (job-level) parallelism, which will perform well on any parallel system [8]. We investigate applications with a finer granularity, which were designed originally to run on a local cluster of workstations. In addition, we study optimizations that can be used to improve the performance on multilevel clusters.

The paper presents the following contributions. First, we present performance measurements for eight parallel programs on a wide-area multilevel cluster, and we identify performance problems. Second, we describe optimization techniques that substantially improve performance on a wide area system. Third, we conclude that, with the optimizations in place, many programs obtain good performance, showing that it is beneficial to run parallel programs on multiple WAN-connected clusters. This conclusion is surprising,

- Classic AI

- TSP, Travelling Salesperson
  branch and bound, shared job queue, shared global minimum, dynamic load balancing. exponential in cities. 17 city

- IDA*, search, 15 puzzle
  distributed job queue work stealing, shared lower bound/search depth, dynamic load balancing

- ASP, All-pairs shortest path
  distributed distance matrix, using broadcast, qubic in n, high communication

- RA, Retrograde Analysis, Awari
  enumerate end-game database, irregular, much communication, message combining

- ACP, Arc Consistency Problem, Constraint Satisfaction
  shared replicated object, Asynchronous broadcast, **not implemented**

- Also Water, Successive Overrelaxation, Automatic Test Pattern Generation

program	type	communication	# RPC/s	kbytes/s	# bcast/s	kbytes/s	speedup
Water	*n*-body	exchange	9,061	18,958	48	1	56.5
TSP	search	work queue	5,692	285	134	11	62.9
ASP	data-parallel	broadcast	3	49	125	721	59.3
ATPG	data-parallel	accumulator	4,508	18	64	0	50.3
RA	data-parallel	irregular	240,297	8,493	296	0	25.9
IDA*	search	work stealing	8,156	202	477	1	62.1
ACP	iterative	broadcast	77	826	1,649	557	37.0
SOR	data-parallel	neighbor	18,811	67,540	326	2	46.3

Application characteristics on 64 processors on one local cluster.

2002

A Performance Analysis of Transposition-Table-Driven Work Scheduling in Distributed Search

John W. Romein, Henri E. Bal, *Member, IEEE Computer Society*, Jonathan Schaeffer, and Aske Plaat

AI search

Abstract—This paper discusses a new work-scheduling algorithm for parallel search of single-agent state spaces, called *Transposition-Table-Driven Work Scheduling*, that places the transposition table at the heart of the parallel work scheduling. The scheme results in less synchronization overhead, less processor idle time, and less redundant search effort. Measurements on a 128-processor parallel machine show that the scheme achieves close-to-linear speedups; for large problems the speedups are even superlinear due to better memory usage. On the same machine, the algorithm is 1.6 to 12.9 times faster than traditional work-stealing-based schemes.

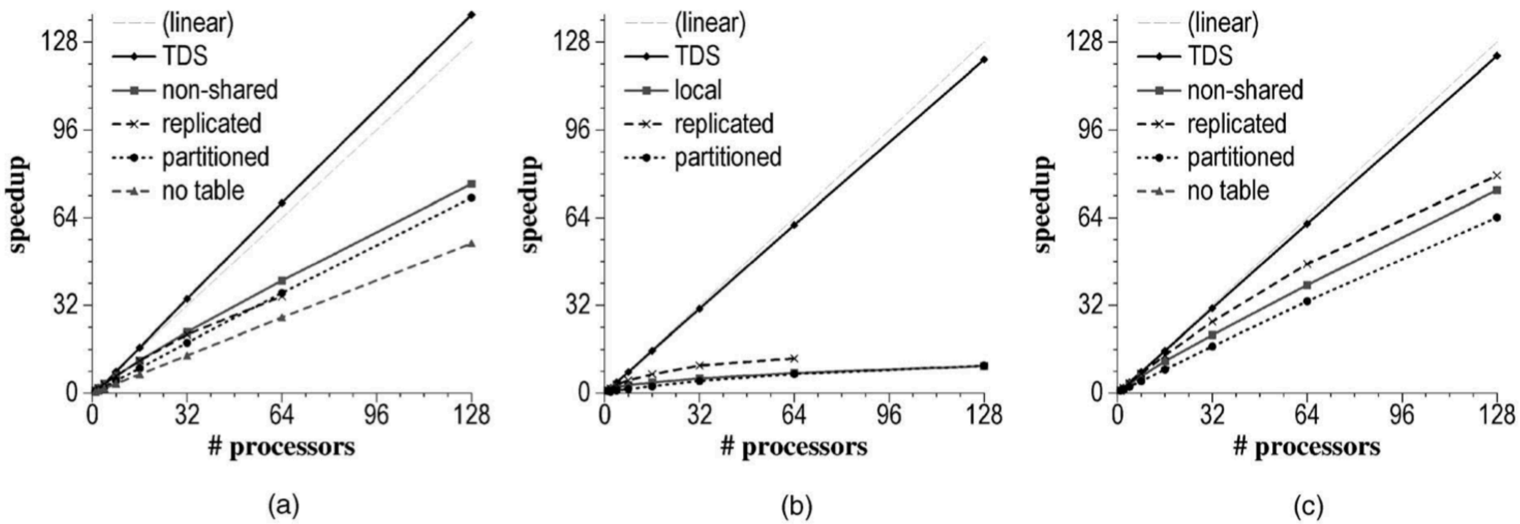


Fig. 5. Average application speedups. (a) 15-puzzle. (b) Double-blank puzzle. (c) Rubik's Cube.

A Performance Analysis of Transposition-Table-Driven Work Scheduling in Distributed Search

John W. Romein, Henri E. Bal, *Member, IEEE Computer Society*,
Jonathan Schaeffer, and Aske Plaat

AI search

Abstract—This paper discusses a new work-scheduling algorithm for parallel search of single-agent state spaces, called *Transposition-Table-Driven Work Scheduling*, that places the transposition table at the heart of the parallel work scheduling. The scheme results in less synchronization overhead, less processor idle time, and less redundant search effort. Measurements on a 128-processor parallel machine show that the scheme achieves close-to-linear speedups; for large problems the speedups are even superlinear due to better memory usage. On the same machine, the algorithm is 1.6 to 12.9 times faster than traditional work-stealing-based schemes.

454 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 13, NO. 5, MAY 2002

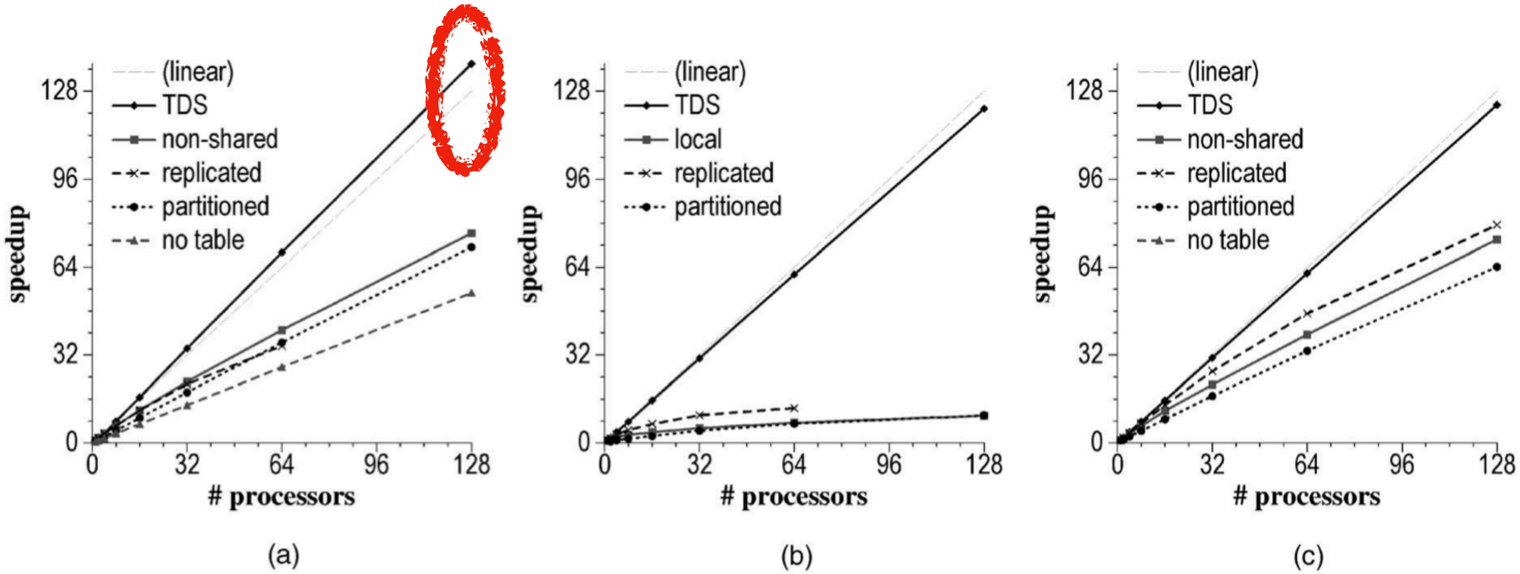


Fig. 5. Average application speedups. (a) 15-puzzle. (b) Double-blank puzzle. (c) Rubik's Cube.

Solving Awari with Parallel Retrograde Analysis

AI search

A new parallel search algorithm running on a large computer cluster solves a popular board game by efficiently computing the best moves from all reachable positions. The resulting databases contain scores for 889 billion positions.

John W. Romein

Henri E. Bal

Vrije Universiteit, Amsterdam

Awari, a 3,500-year-old game that originated in Africa, is played today all over the world. It is the best-known variant of the *mancala* games, in which players "sow" stones into pits on a board. The

we have verified. The largest database contains scores for every reachable position with all 48 stones. Its 204 billion entries occupy 178 gigabytes—exceeding the largest database for any game computed so far in both number of positions and storage size.

Table 1. Entries for some of the *n*-stone databases.

n	Number of entries	*n*	Number of entries	*n*	Number of entries
0	1	5	4,368	43	73,127,652,312
1	12	6	12,375	44	90,517,649,586
2	78	...	...	45	111,548,476,480
3	364	41	47,062,945,644	46	136,883,249,790
4	1,365	42	58,806,619,443	48	203,648,015,936

THE RESULTS: AWARI IS A DRAW

The databases revealed that awari is a draw, so being able to move first in the game provides neither an advantage nor a disadvantage. Remarkably, there is only one good opening move—the move that originates from the rightmost pit (F). All other opening moves will lose.

Solving Awari with Parallel Retrograde Analysis

AI search

A new parallel search algorithm running on a large computer cluster solves a popular board game by efficiently computing the best moves from all reachable positions. The resulting databases contain scores for 889 billion positions.

John W. Romein

Henri E. Bal
Vrije Universiteit, Amsterdam

Awari, a 3,500-year-old game that originated in Africa, is played today all over the world. It is the best-known variant of the *mancala* games, in which players "sow" stones into pits on a board. The

we have verified. The largest database contains scores for every reachable position with all 48 stones. Its 204 billion entries occupy 178 gigabytes—exceeding the largest database for any game computed so far in both number of positions and storage size.

Table 1. Entries for some of the *n*-stone databases.

n	Number of entries	*n*	Number of entries	*n*	Number of entries
0	1	5	4,368	43	73,127,652,312
1	12	6	12,375	44	90,517,649,586
2	78	...	...	45	111,548,476,480
3	364	41	47,062,945,644	46	136,883,249,790
4	1,365	42	58,806,619,443	48	203,648,015,936

THE RESULTS: AWARI IS A DRAW

The databases revealed that awari is a draw, so being able to move first in the game provides neither an advantage nor a disadvantage. Remarkably, there is only one good opening move—the move that originates from the rightmost pit (F). All other opening moves will lose.

Solving Awari with Parallel Retrograde Analysis

AI search

A new parallel search algorithm running on a large computer cluster solves a popular board game by efficiently computing the best moves from all reachable positions. The resulting databases contain scores for 889 billion positions.

John W. Romein

Henri E. Bal

Vrije Universiteit, Amsterdam

Awari, a 3,500-year-old game that originated in Africa, is played today all over the world. It is the best-known variant of the *mancala* games, in which players "sow" stones into pits on a board. The

we have verified. The largest database contains scores for every reachable position with all 48 stones. Its 204 billion entries occupy 178 gigabytes—exceeding the largest database for any game computed so far in both number of positions and storage size.

Table 1. Entries for some of the *n*-stone databases.

n	Number of entries	*n*	Number of entries	*n*	Number of entries
0	1	5	4,368	43	73,127,652,312
1	12	6	12,375	44	90,517,649,586
2	78	...	...	45	111,548,476,480
3	364	41	47,062,945,644	46	136,883,249,790
4	1,365	42	58,806,619,443	48	203,648,015,936

THE RESULTS: AWARI IS A DRAW

The databases revealed that awari is a draw, so being able to move first in the game provides neither an advantage nor a disadvantage. Remarkably, there is only one good opening move—the move that originates from the rightmost pit (F). All other opening moves will lose.

AI 4 HPC

AI 4 HPC

ELSEVIER

Astronomy and Computing
Volume 14, January 2016, Pages 1-7

Full length article

Real-time dedispersion for fast radio transient surveys, using auto tuning on many-core accelerators

A. Sclocco [a b], J. van Leeuwen [b c], H.E. Bal [a], R.V. van Nieuwpoort [d]

AI 4 HPC

2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Reinforcement Learning-Guided Channel Selection Across Time for Multivariate Time Series Classification

Leonardos Pantiskas
Vrije Universiteit Amsterdam
0000-0002-4898-5334

Kees Verstoep
Vrije Universiteit Amsterdam
0000-0001-6402-2928

Mark Hoogendoorn
Vrije Universiteit Amsterdam
0000-0003-3356-3574

Henri Bal
Vrije Universiteit Amsterdam
0000-0001-9827-4461

Conclusion

High Performance?
High Abstraction?


High Performance!
High Abstraction!

Lessons from Henri

- Be curious
- Aim high
- Write clearly
- Be nice

Lessons from Henri

- Be curious
- Aim high
- Write clearly
- Be nice


Thank
you