

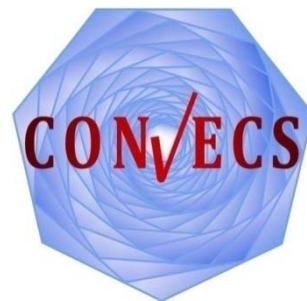
---

# Large-Scale Distributed Verification using CADP Beyond Clusters to Grids

Hubert Garavel

Radu Mateescu

**Wendelin Serwe**



# Introduction

## ● Context

- Explicit-state verification of concurrent systems
- Action-based setting (process-calculi)
- Limitation: *state space explosion*

## ● Distributed verification

- Large memory available in clusters and grid
- Additional benefit: speedup

# CADP (<http://cadp.inria.fr>)



- Construction and Analysis of Distributed Processes
- *Modular* toolbox with *several*
  - **Formal specification languages:**  
LOTOS, LNT, FSP,  $\pi$ -calculus
  - **Verification paradigms:**  
model checking, equivalence checking, visual checking
  - **Analysis techniques:**  
reachability, on-the-fly, compositional, distributed, static analysis, code/test generation, performance evaluation
- Continuous development for more than 20 years
- 150 case-studies and 60 3<sup>rd</sup> party tools

# Communication Support in CADP

- Typical application:  $N$  **workers** and **1 master**
- GCF (*Grid Configuration File*):
  - number of workers
  - for each worker: machine, working directory, user, ...
- communication library: `caesar_network_1`
  - dedicated to distributed verification
  - based on
    - sockets (TCP)
    - remote access (ssh)

```
rsh = ssh -q
rcp = scp -q
connect_timeout = 30
adonis-1.grenoble.grid5000.fr
    directory=/home/wserwe/Demo_41/1
genepi-2.grenoble.grid5000.fr
    directory=/home/wserwe/Demo_41/2
granduc-3.luxembourg.grid5000.fr
    directory=/tmp/3
suno-4.sophia.grid5000.fr
    directory=/home/wserwe/tmp/4
```

# Distributed LTS

- Ordinary LTS: BCG (*Binary Coded Graph*) format
- Distributed LTS: one fragment LTS per worker
- **PBG (*Partitioned BCG Graph*) format**
  - joint work of SENVA (associated CWI-INRIA team)
  - fragment information (LTS, log file, #states, ...)

```
PBG 1.0
# PBG format by the SENVA team -- http://vasy.inria.fr/senva
# created by Distributor (C) INRIA/VASY -- http://vasy.inria.fr/cadp
# (do not modify this file unless you know exactly what you are doing)
grid: "grid_4.gcf"[0]
states: partitioned
edges: incoming
initiator: 1
fragments: 4
1: states: 496285 fragment: "anderson.4-1.bcg"[0] log: "distributor-1.log"[0]
2: states: 497109 fragment: "anderson.4-2.bcg"[0] log: "distributor-2.log"[0]
3: states: 495159 fragment: "anderson.4-3.bcg"[0] log: "distributor-3.log"[0]
4: states: 497507 fragment: "anderson.4-4.bcg"[0] log: "distributor-4.log"[0]
```

# Tools for Distributed LTSs

## ● Creation

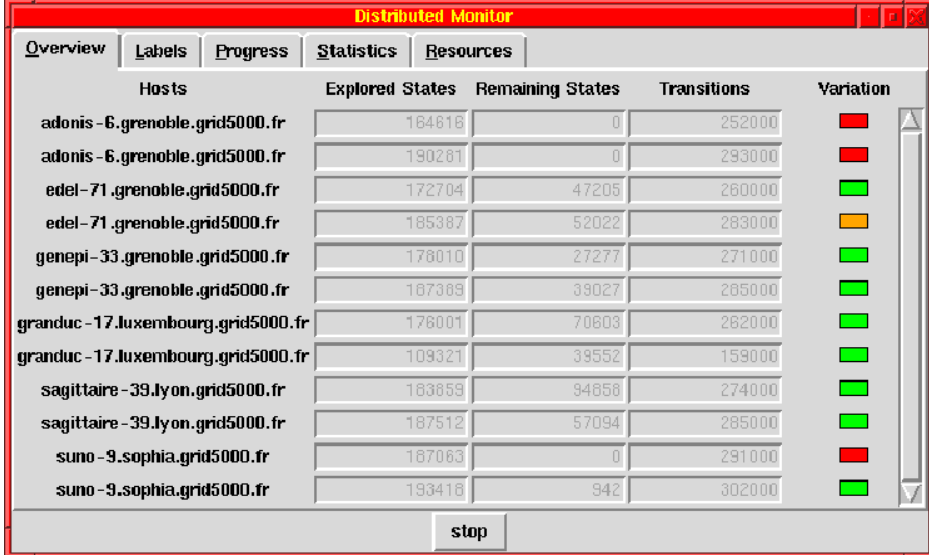
- DISTRIBUTOR
- Monitor window

## ● Inspection and Manipulation

- PBG\_INFO
- PBG\_CP, PBG\_MV, PBG\_RM

## ● Combination into an ordinary LTS: PBG\_MERGE

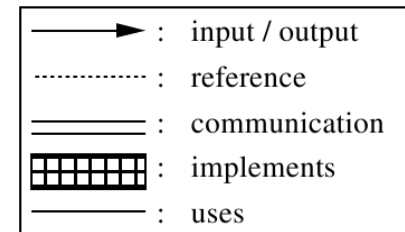
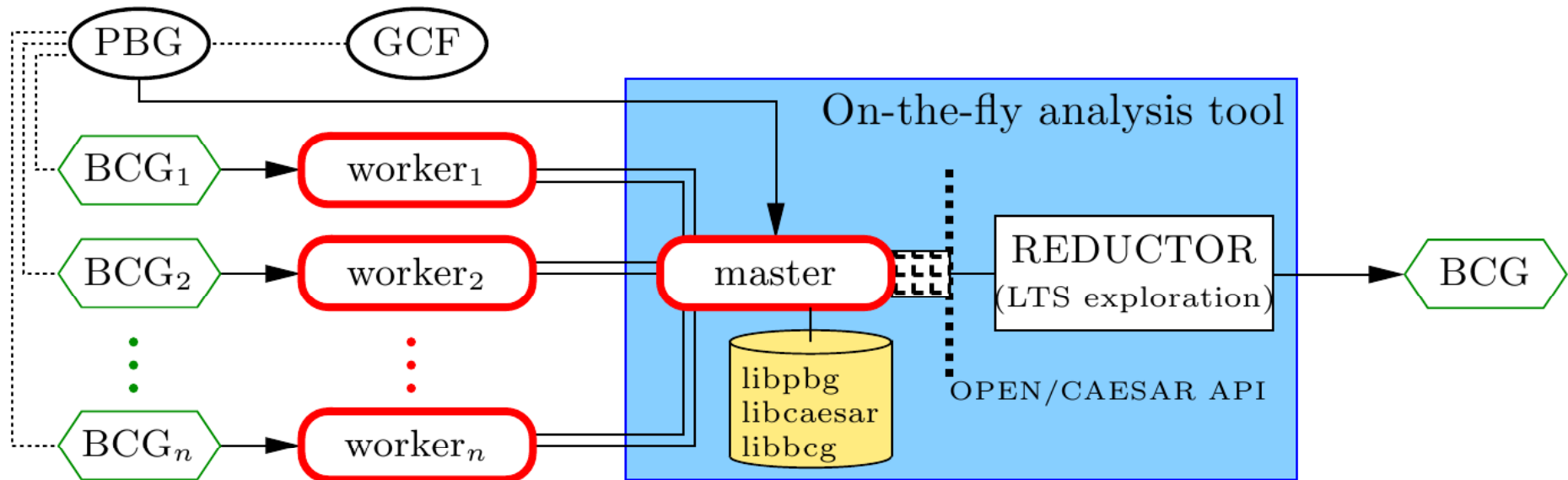
## ● On-the-fly exploration: PBG\_OPEN with *all* on-the-fly tools of CADP



The screenshot shows a window titled "Distributed Monitor" with a red border. It has a menu bar with "Overview", "Labels", "Progress", "Statistics", and "Resources". Below the menu bar is a table with the following columns: "Hosts", "Explored States", "Remaining States", "Transitions", and "Variation". The table contains 12 rows of data for various hosts. A "stop" button is located at the bottom right of the table area.

Hosts	Explored States	Remaining States	Transitions	Variation
adonis-6.grenoble.grid5000.fr	164616	0	252000	Red
adonis-6.grenoble.grid5000.fr	190261	0	293000	Red
edel-71.grenoble.grid5000.fr	172704	47205	260000	Green
edel-71.grenoble.grid5000.fr	185387	52022	283000	Yellow
genepi-33.grenoble.grid5000.fr	178010	27277	271000	Green
genepi-33.grenoble.grid5000.fr	187389	39027	265000	Green
granduc-17.luxembourg.grid5000.fr	178001	70603	262000	Green
granduc-17.luxembourg.grid5000.fr	109321	39552	159000	Green
sagittaire-39.lyon.grid5000.fr	183659	94658	274000	Green
sagittaire-39.lyon.grid5000.fr	187512	57094	285000	Green
suno-9.sophia.grid5000.fr	187063	0	291000	Red
suno-9.sophia.grid5000.fr	193416	942	302000	Green

# PBG\_OPEN Application Architecture



# EXPERIMENTS



# Experimental Platform: Grid'5000

- Several sites all over France and Luxembourg, possibly several clusters per site
- Communication via 1 Gbit/s links
- Access via resource managers (one per site)
- Measurements
  - Execution time including set-up (copy etc.)
  - Memory consumption
    - maximal memory consumption of a process (master and workers)
    - *total memory*: sum of maxima
    - *peak memory*: greatest value of maxima
- Examples: Mutual exclusion protocols  
[\[Mateescu-Serwe-12\]](#)



<http://www.grid5000.fr>

# Experiments: Examples

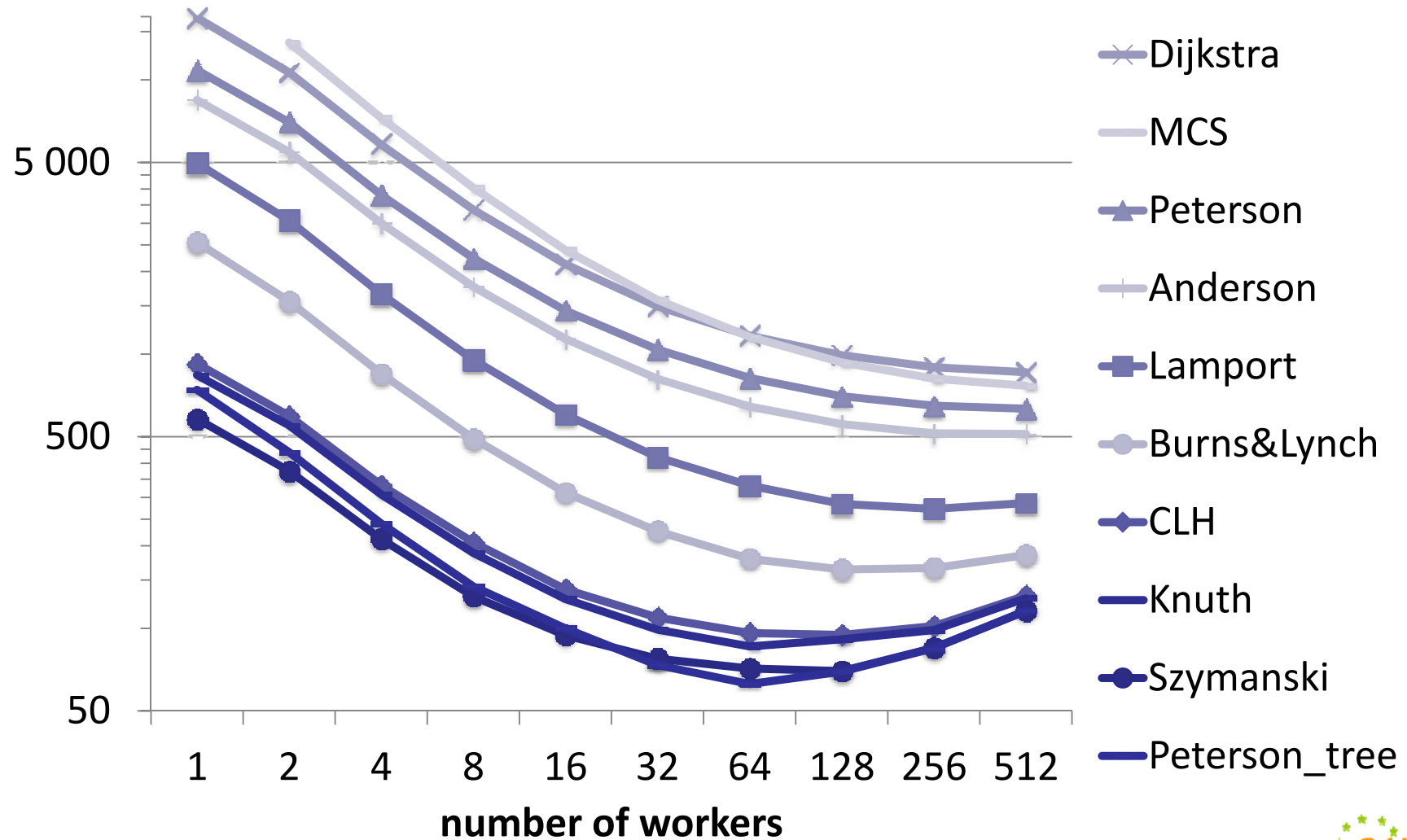
[Mateescu-Serwe-12]

protocol	#proc.	unreduced		minimized	
TTAS	4	18,721	39,736	80	224
Burns&Lynch-4	4	769,244	1,367,318	3,023	11,244
Anderson	4	1,986,060	3,738,240	320	848
MCS	4	4,293,881	7,719,512	320	848
Peterson_tree	4	7,205,545	12,692,584	2,361	8,352
Szymanski	4	9,243,653	18,859,330	3,090	10,356
Knuth	4	16,642,361	32,614,282	6,721	27,281
CLH	4	18,317,849	31,849,616	320	848
Burns&Lynch	5	39,796,190	75,024,550	35,734	167,747
Lamport	4	78,535,973	154,003,176	29,719	99,850
Anderson	5	166,488,027	345,843,975	1,712	4,880
Peterson	4	214,175,671	389,640,061	6,460	21,347
MCS	5	261,064,933	500,744,765	1,712	4,880
Dijkstra	4	289,120,985	542,886,005	41,513	163,538

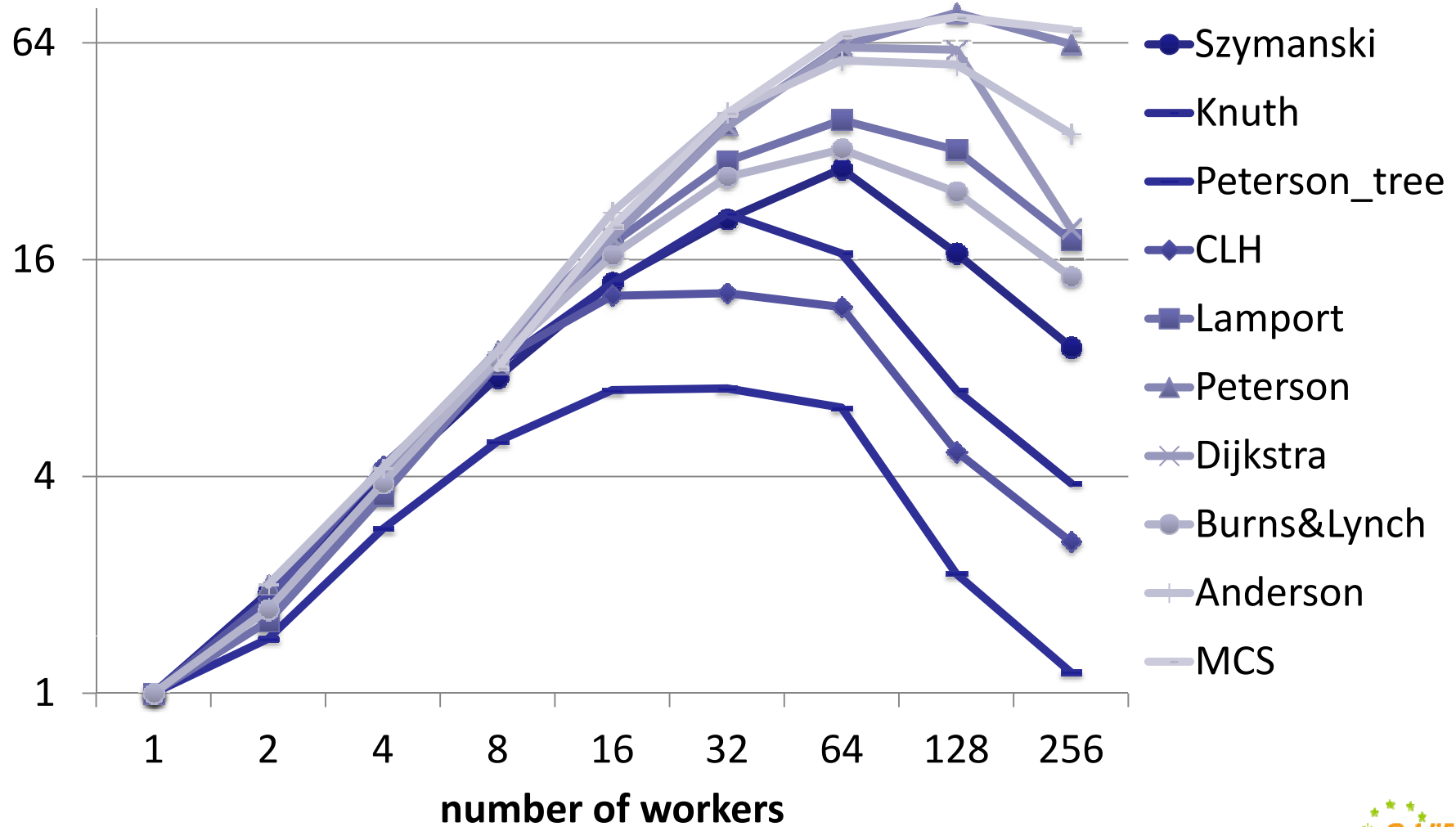
---

# DISTRIBUTED STATE SPACE GENERATION USING DISTRIBUTOR

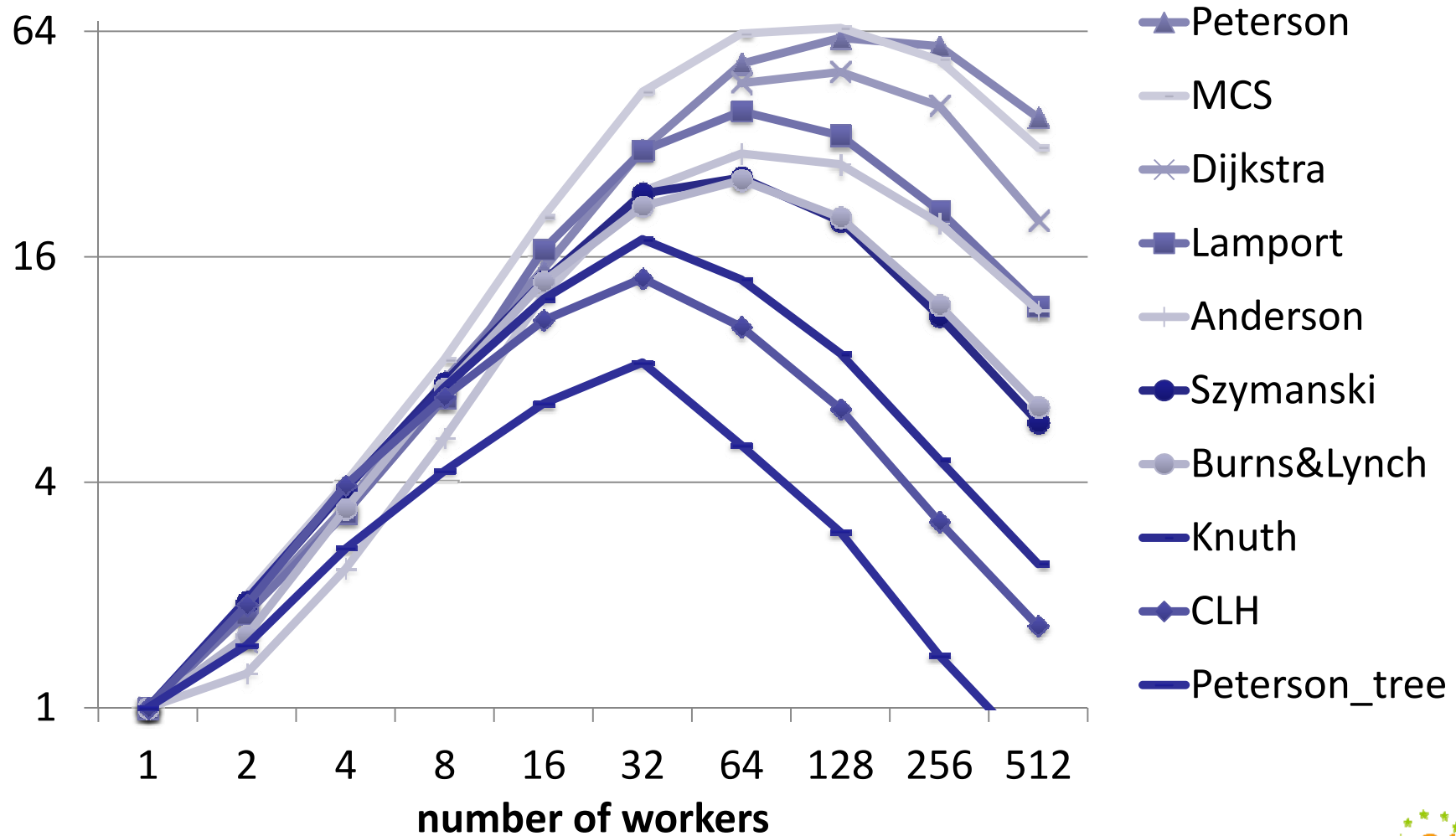
# Generation: Peak Memory (MB)



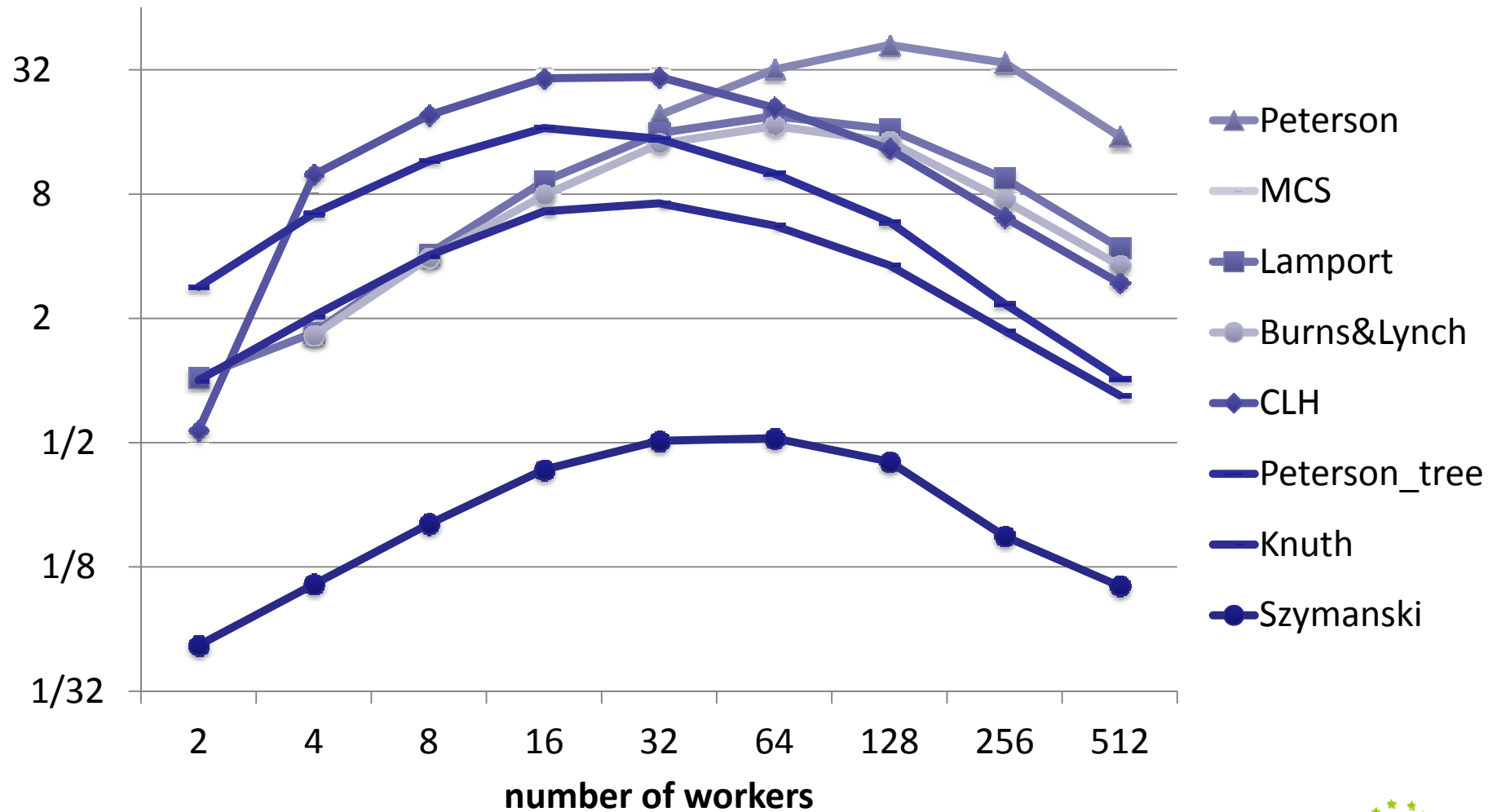
# Generation: Speedup Single-Cluster



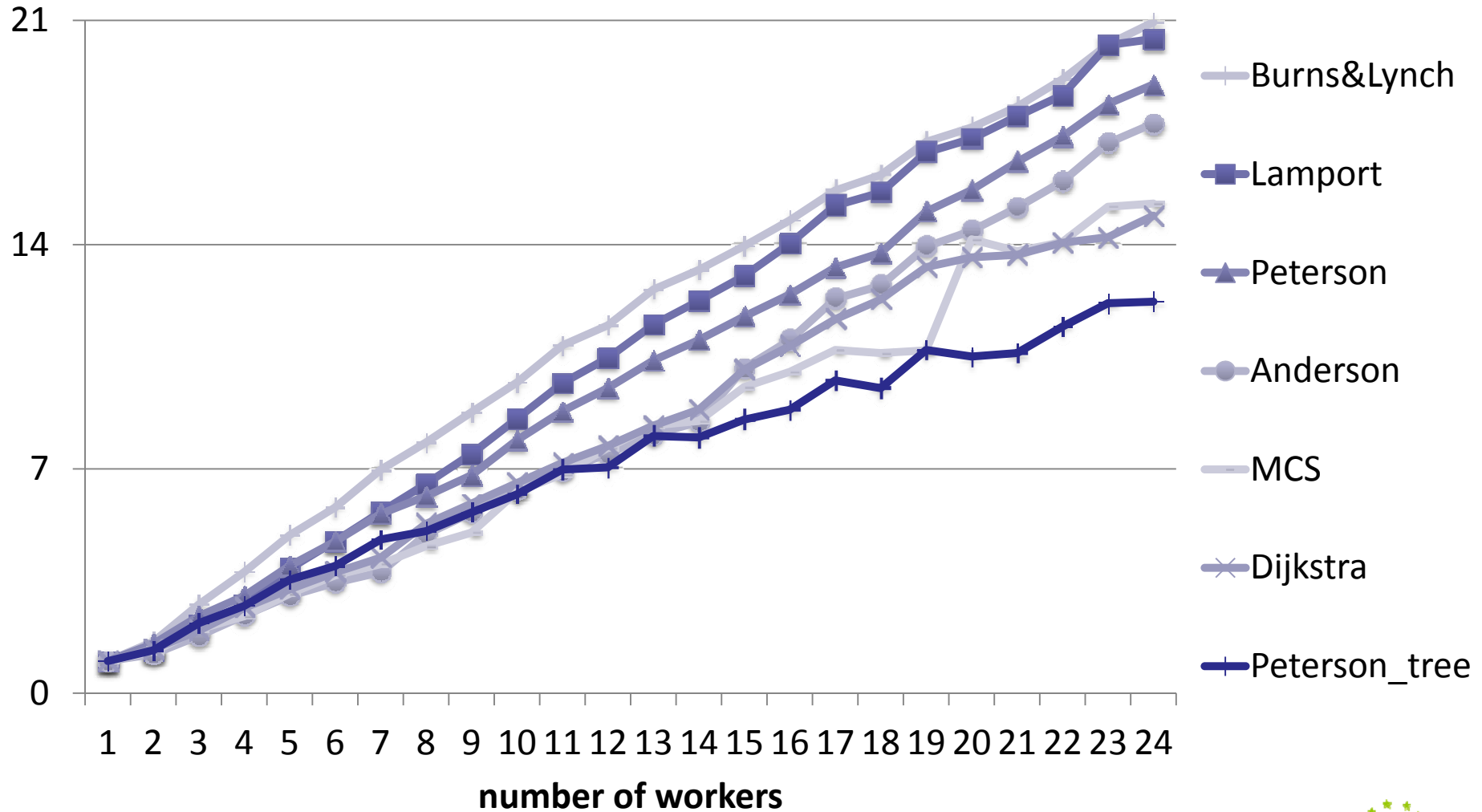
# Generation: Speedup Multi-Cluster



# Generation: Speedup Grid

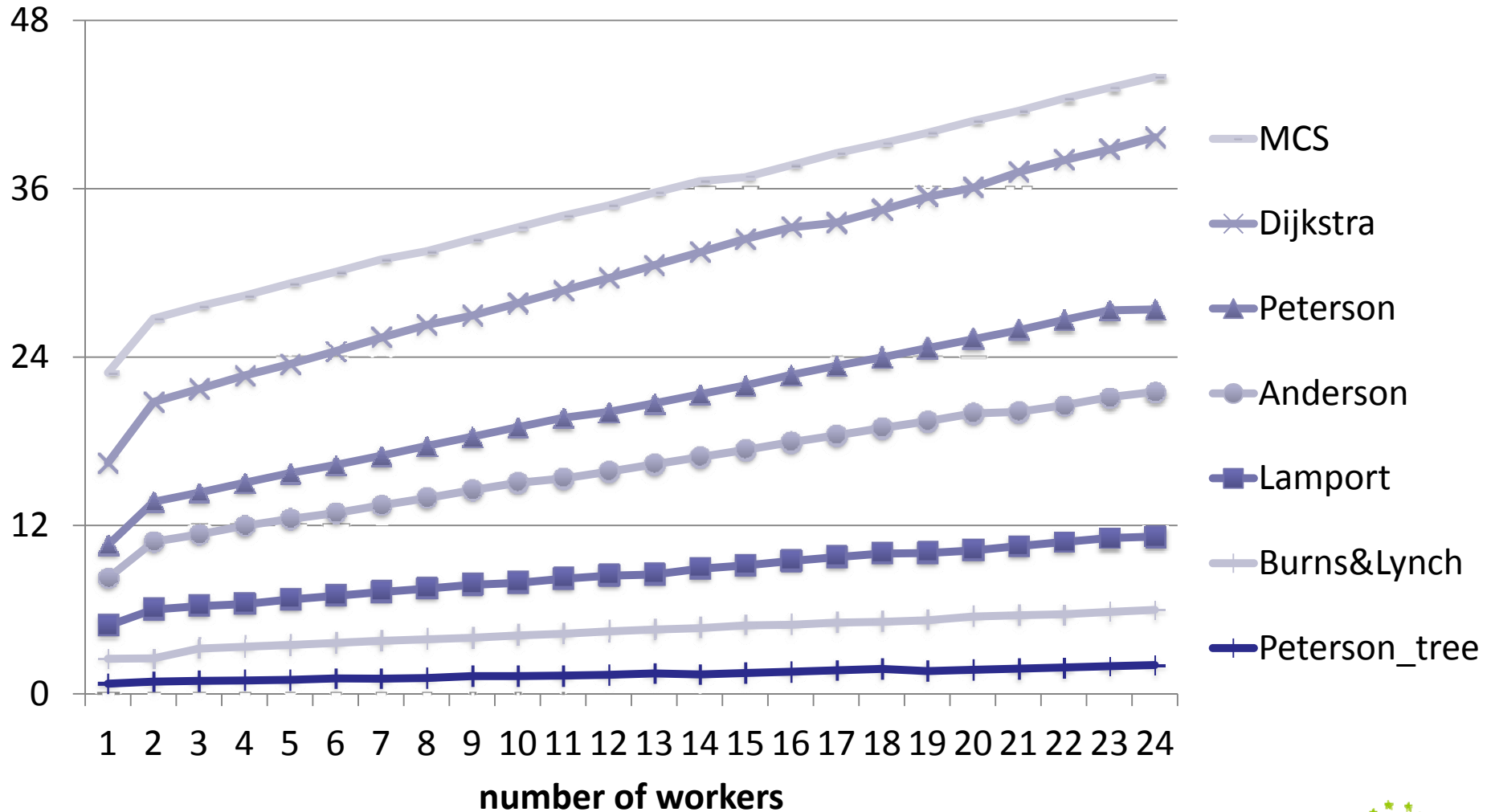


# Generation: Speedup Multicore





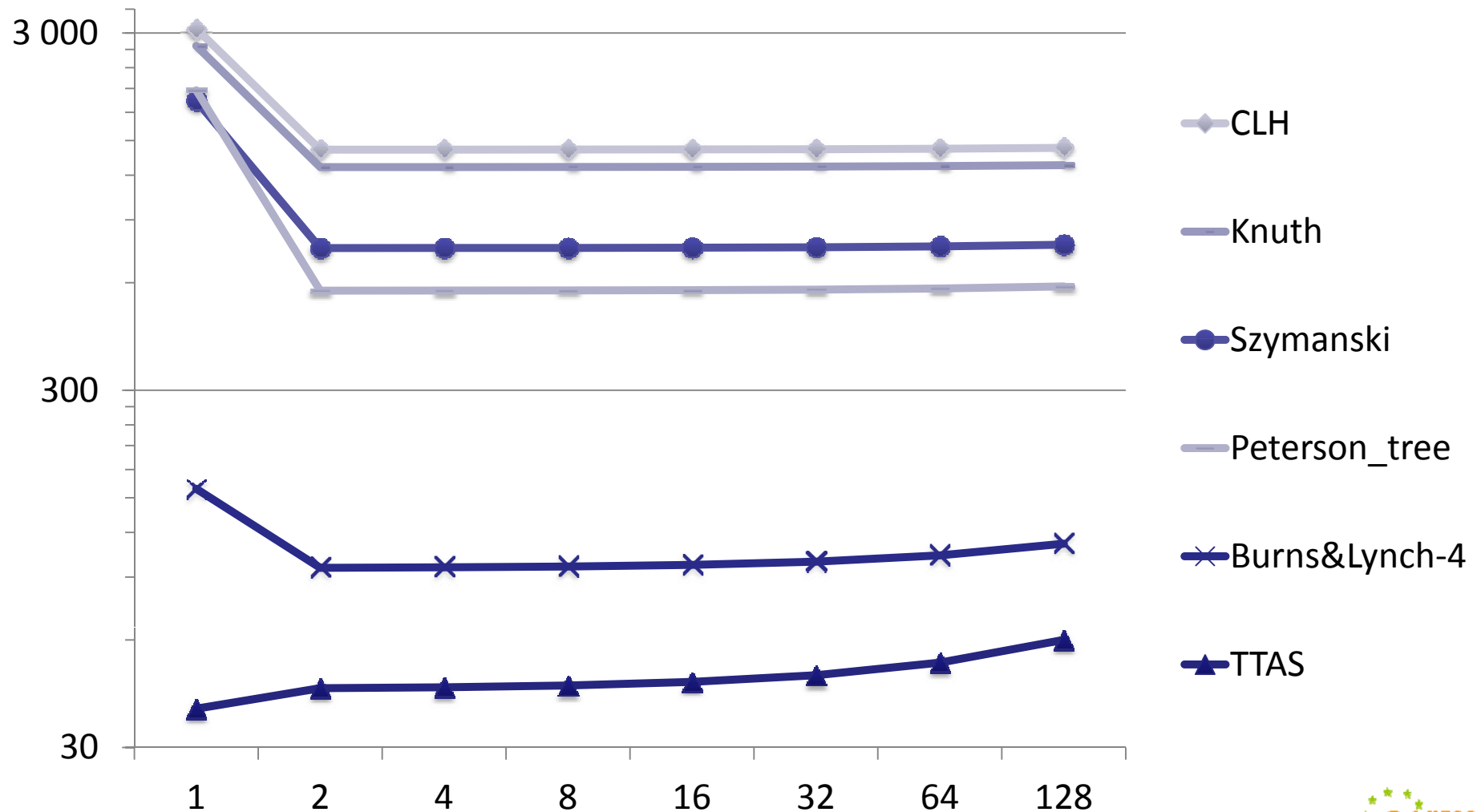
# Generation: Total Memory (GB)



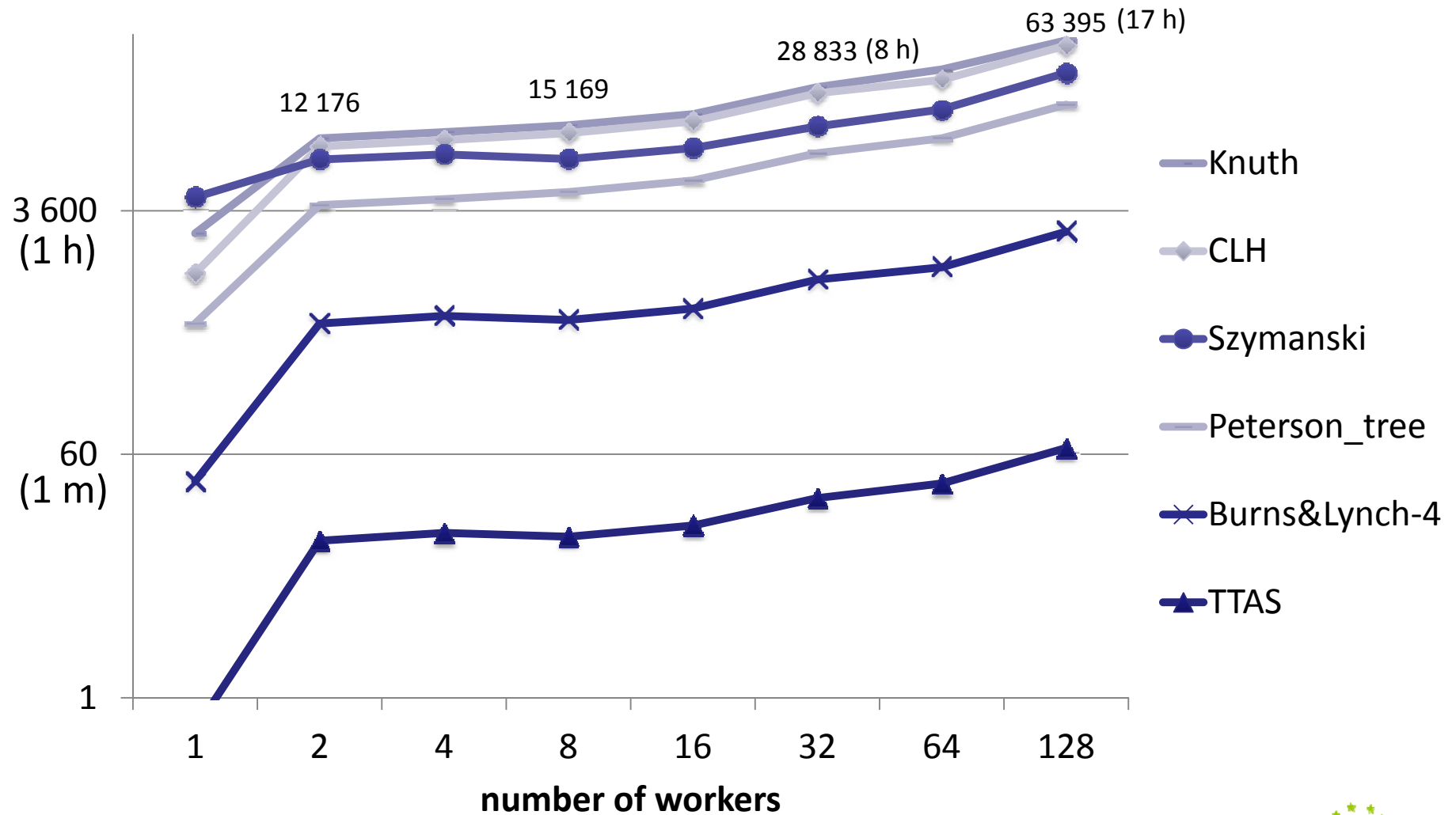
---

# T-CONFLUENCE REDUCTION USING REDUCTOR

# $\tau$ -conf. Reduction: Peak Memory (MB)



# $\tau$ -conf. Reduction: Execution time (s)



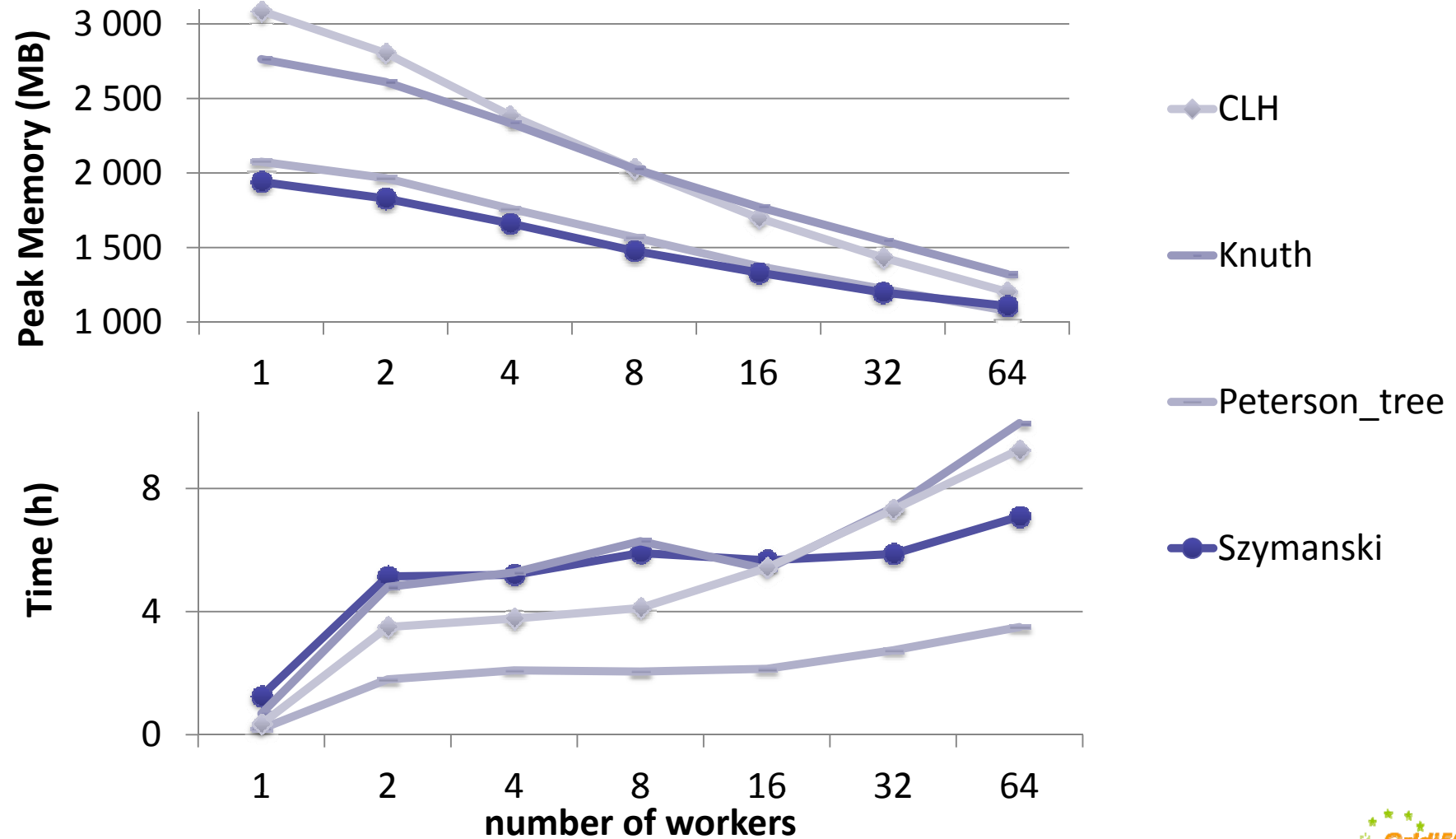
# Alternative $\tau$ -conf. Reduction

example	LNT.OPEN		BCG_OPEN		PBG_OPEN	
	time	memory	time	memory	time	memory
TTAS	4	39	3	34	14	44
Burns&Lynch-4	43	159	24	91	541	96
Peterson_tree	549	2076	365	626	3954	570
Szymanski	4530	1940	1290	842	8537	750
Knuth	2470	2764	1642	1428	12174	1264
CLH	1267	3087	1060	1585	10615	1415
Lamport	<i>oom</i>	<i>oom</i>	9227	6930	59932	6015

- LNT.OPEN: LNT.OPEN with REDUCTOR
- BCG\_OPEN: distributed generation, merge, BCG\_OPEN with REDUCTOR
- PBG\_OPEN: distributed generation, PBG\_OPEN with REDUCTOR



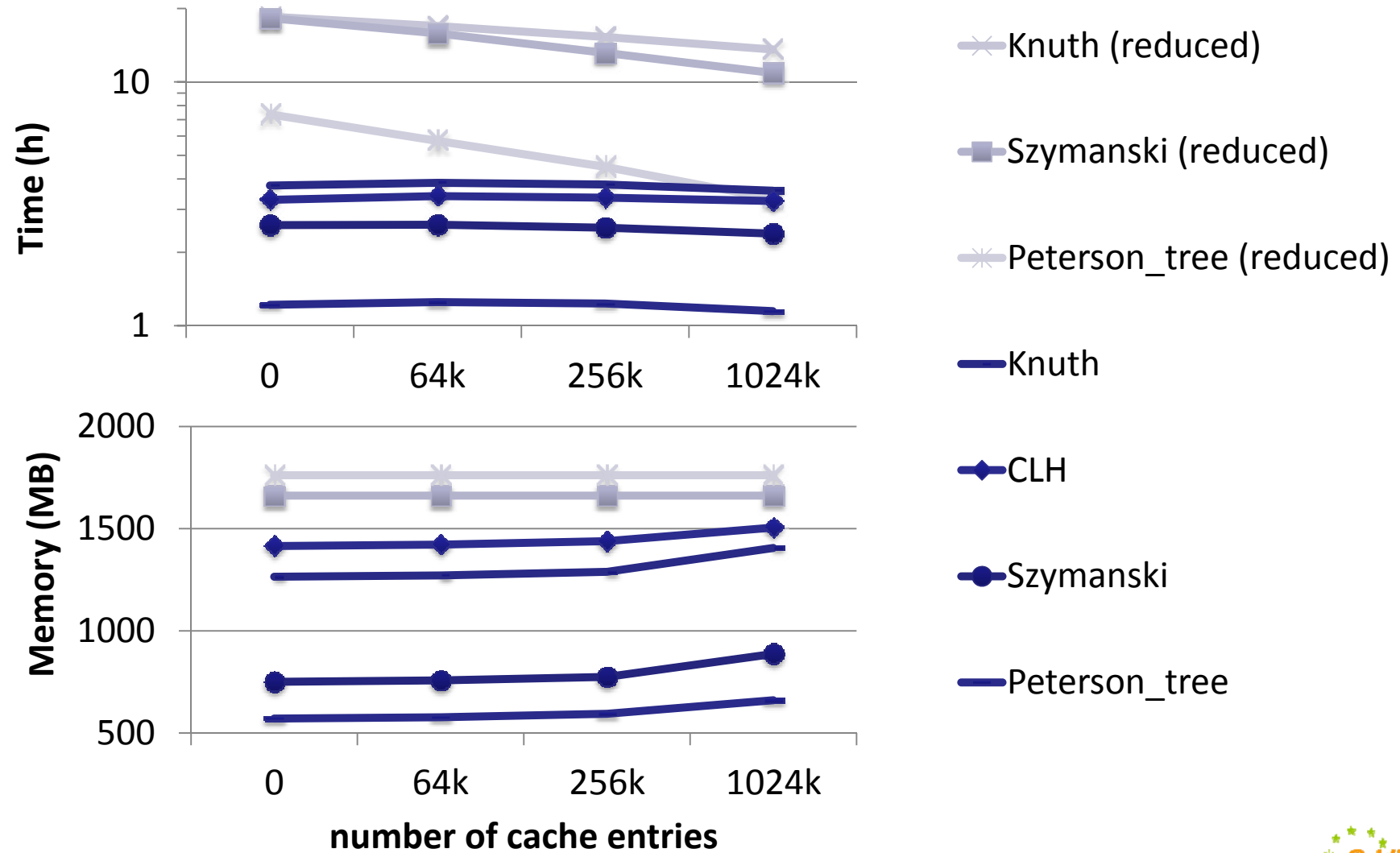
# Double $\tau$ -confluence Reduction



---

# CACHING OF TRANSITIONS

# Caching Successor Lists (Cluster)

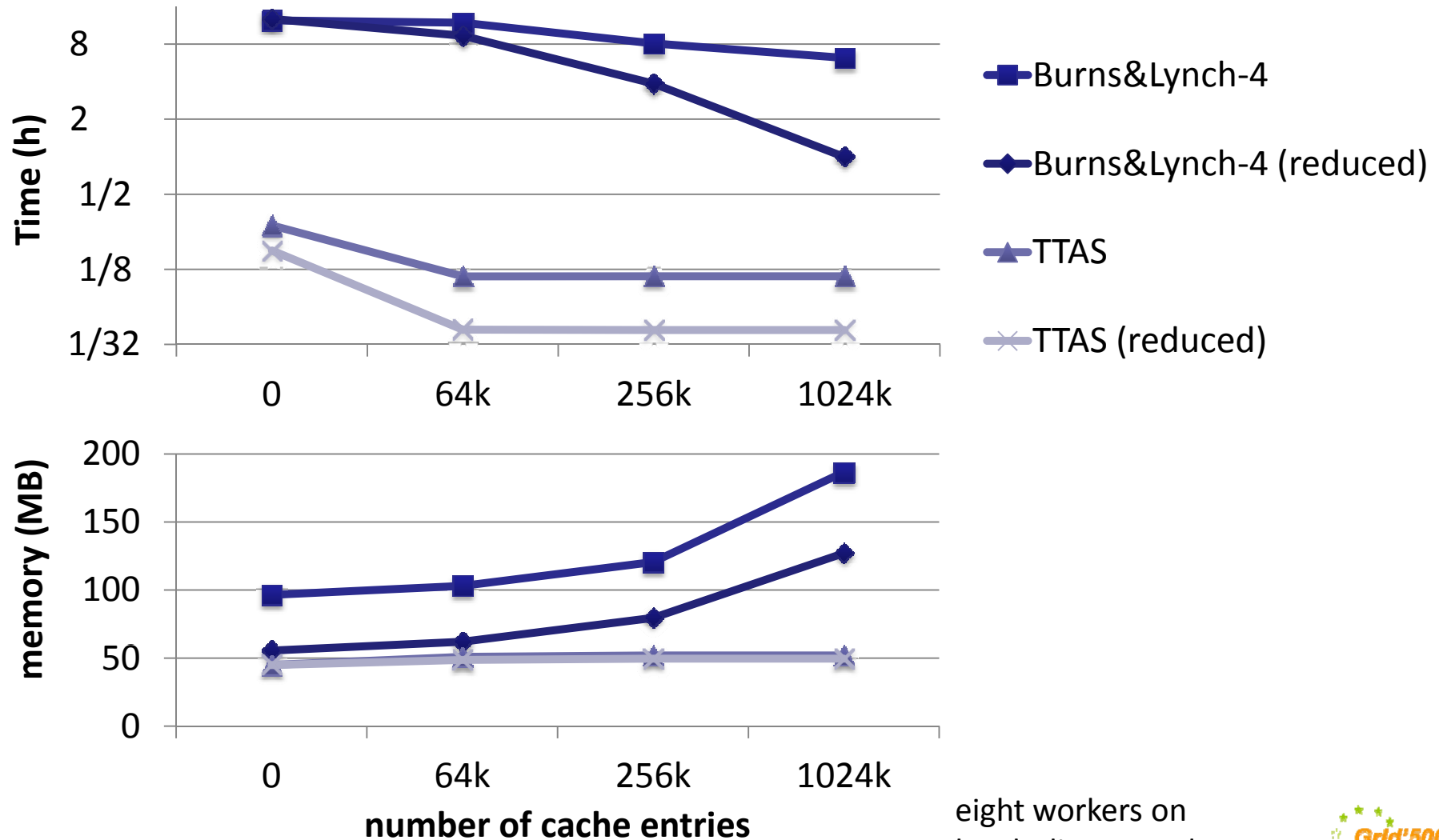


four workers on granduc





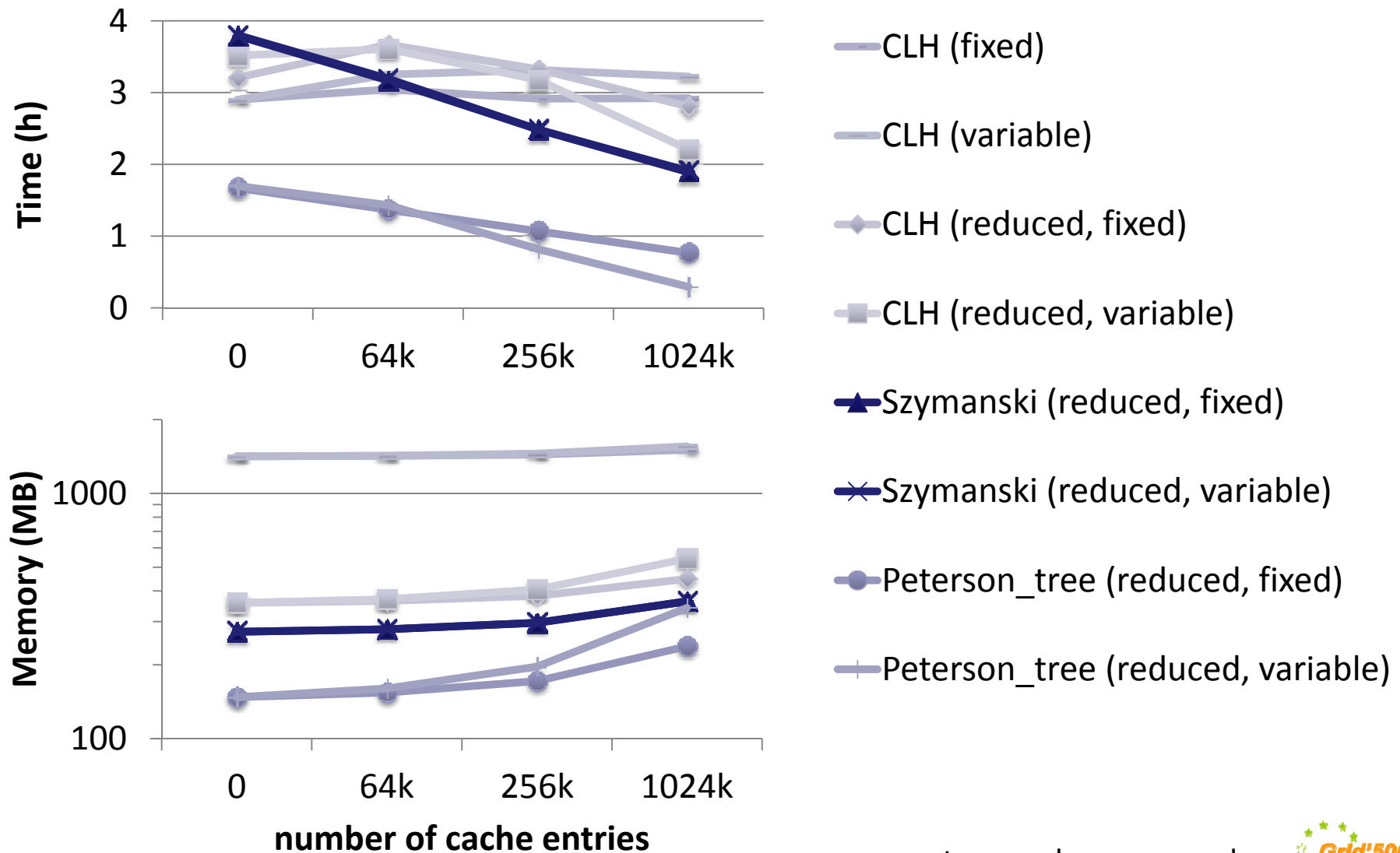
# Caching Successor Lists (Grid)



eight workers on  
borderline, granduc, suno



# Comparison of Cache Implementations



two workers on granduc



# Related Work

- DiVinE
  - Distributed LTL model-checking
  - Fine-grained analysis and tuning of the communication
- Spin
  - Partial-order reductions
- Preach
  - Reachability analysis of Mur $\phi$
  - Communication library implemented in Erlang
- ProActive distributed components
  - Experiments on PACAGrid [[Madelaine-et-al-11](#)]

# Conclusion & Future Work

- PBG format for distributed state spaces
- Experiments on Grid'5000: up to 512 workers
- Caching successors to improve performance
- Further experiments:
  - improve caesar\_network\_1 (parallel setup)
  - static load balancing techniques
  - sequence distributed  $\tau$ -confluence reduction
  - (distributed) model-checking