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, π-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
- Many case-studies and 3rd 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
  - 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
```

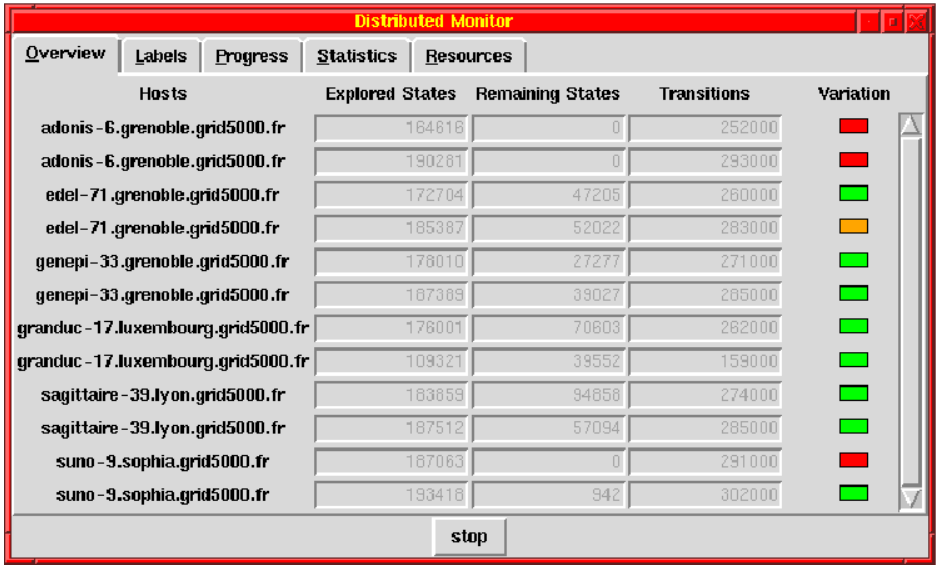
Distributed LTS

- 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



EXPERIMENTS

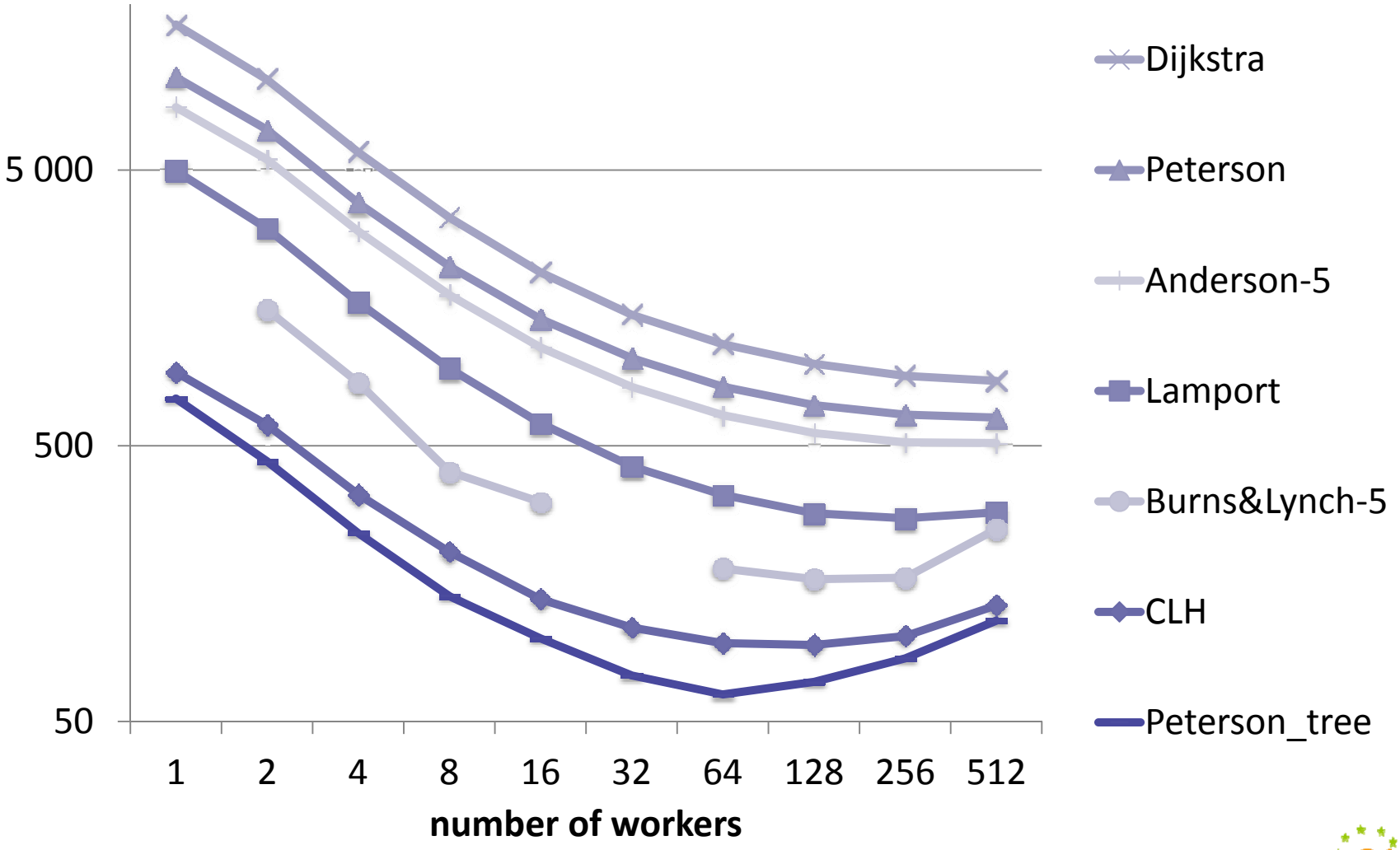
Experimental Platform: Grid'5000

- 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)
- Examples:
  Mutual exclusion protocols [Mateescu-Serwe-12]

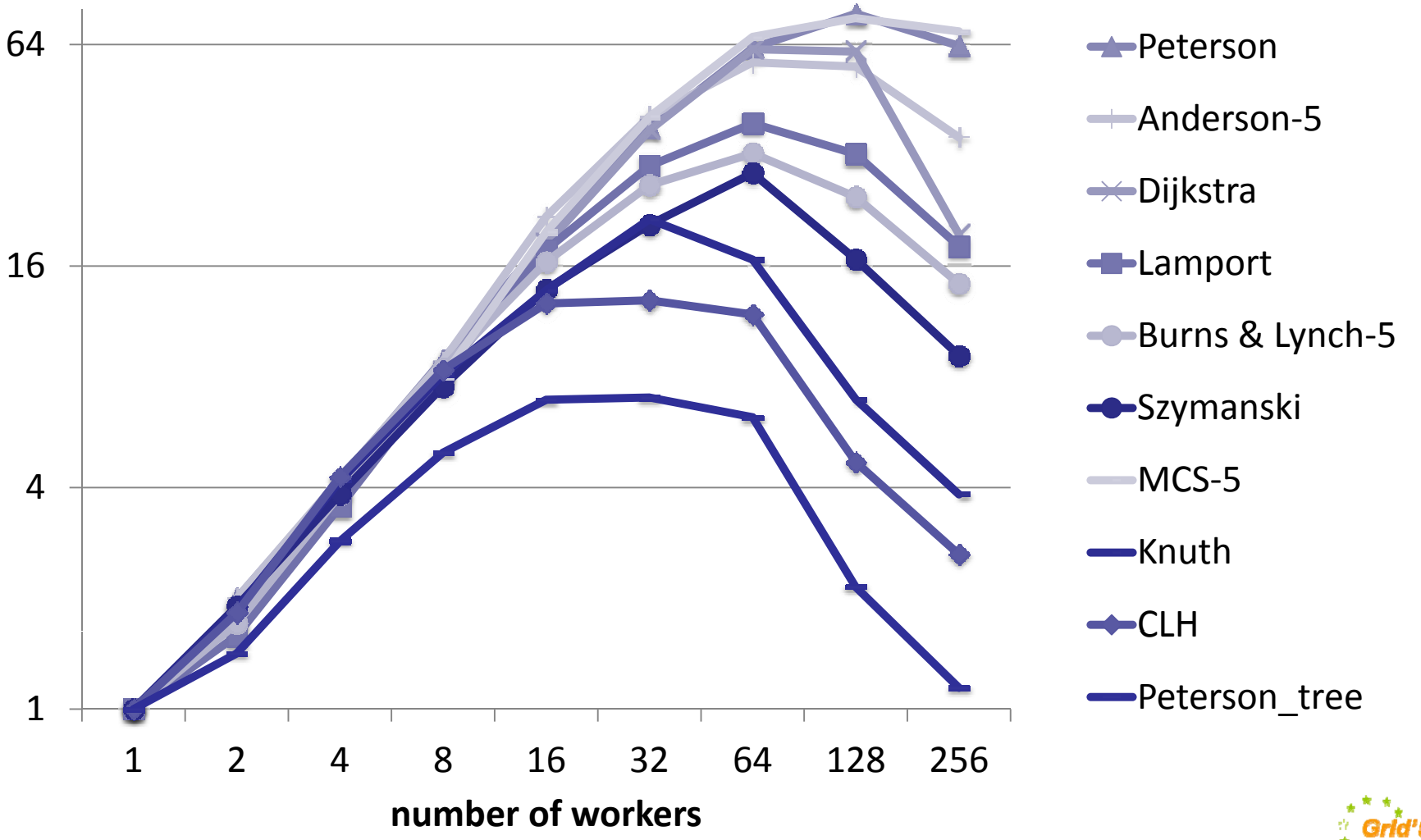
Grid'5000 http://www.grid5000.fr

STATE SPACE GENERATION

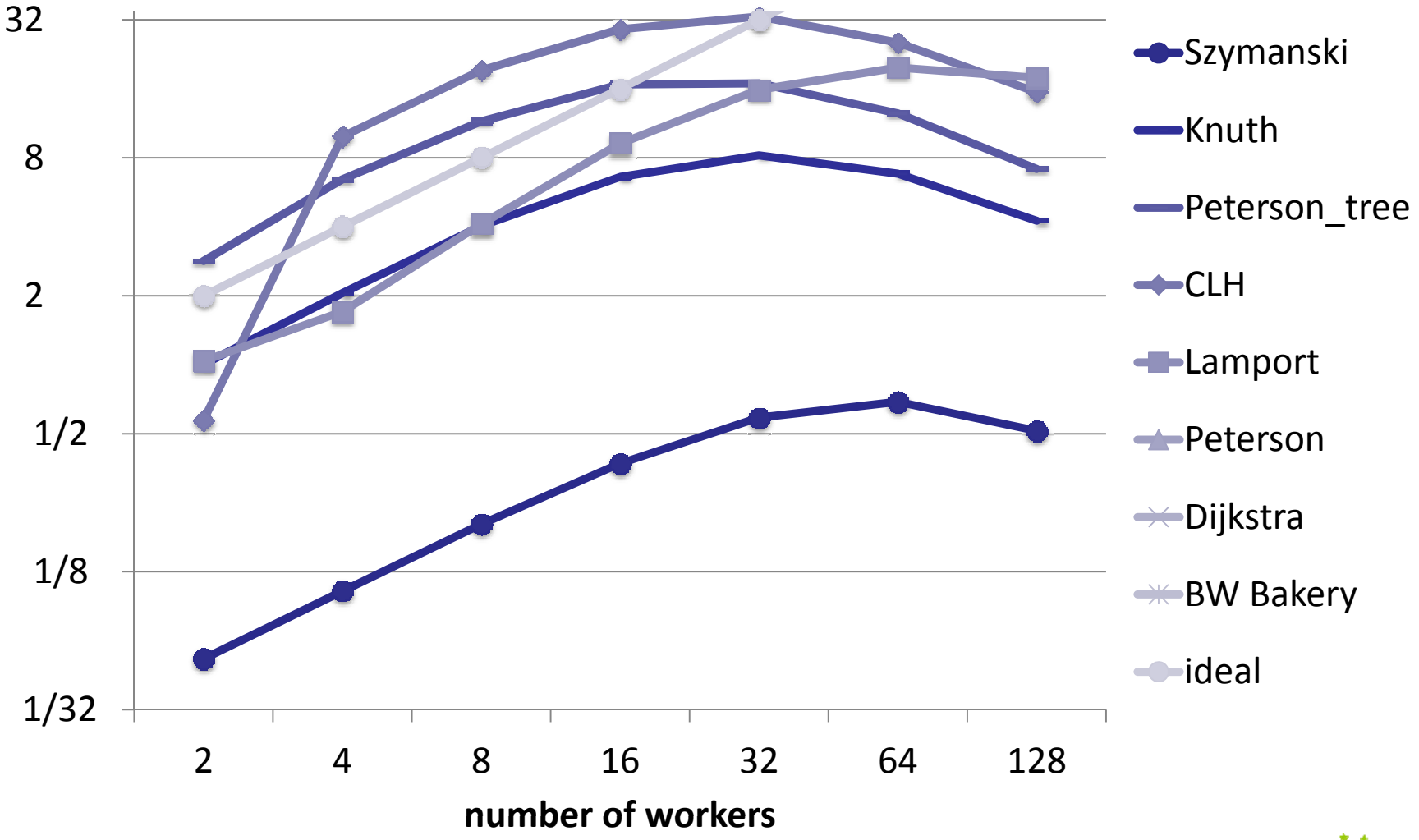
Generation: Peak Memory (MB)



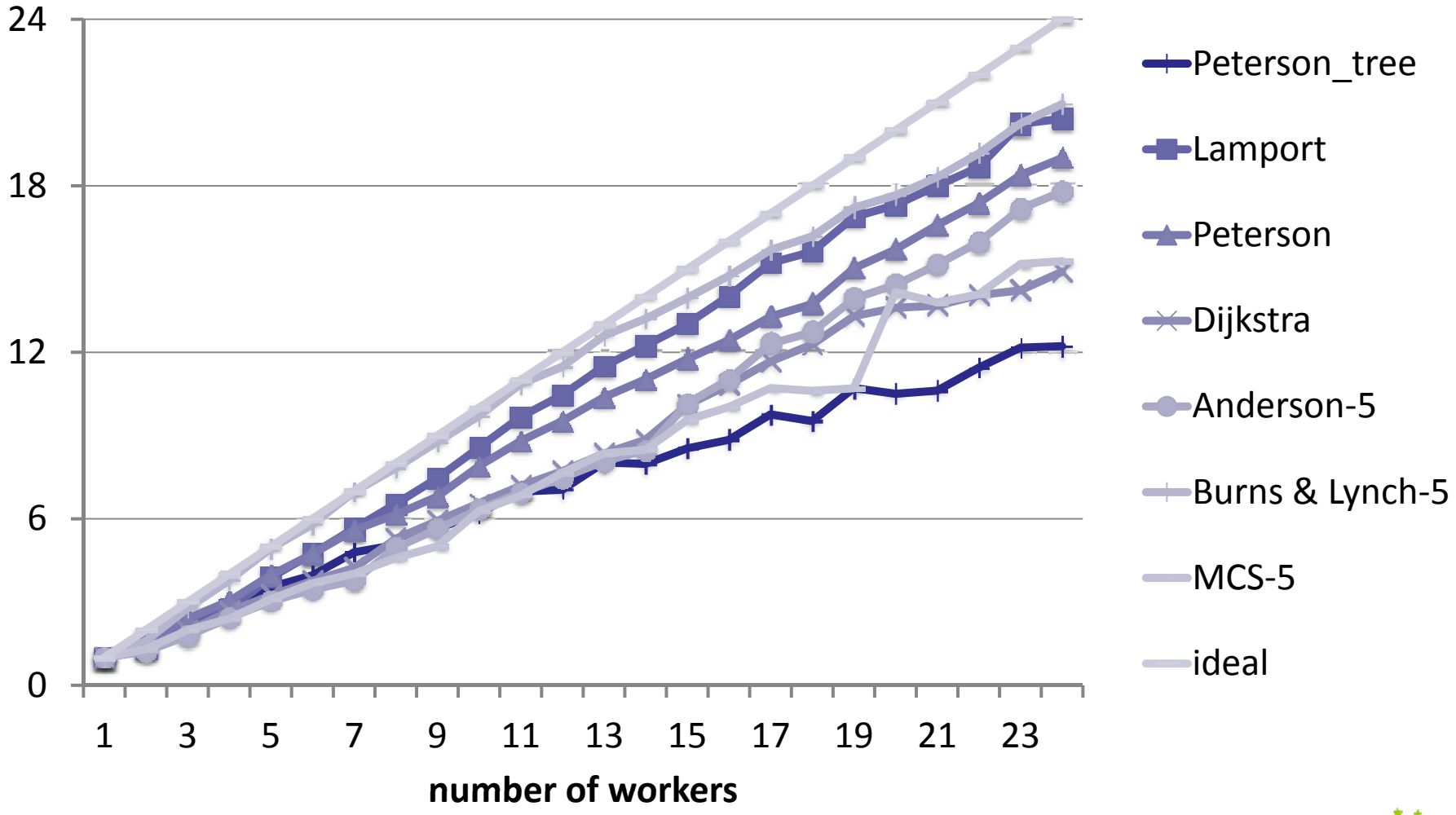
Generation: Speedup Single-Cluster



Generation: Speedup Grid

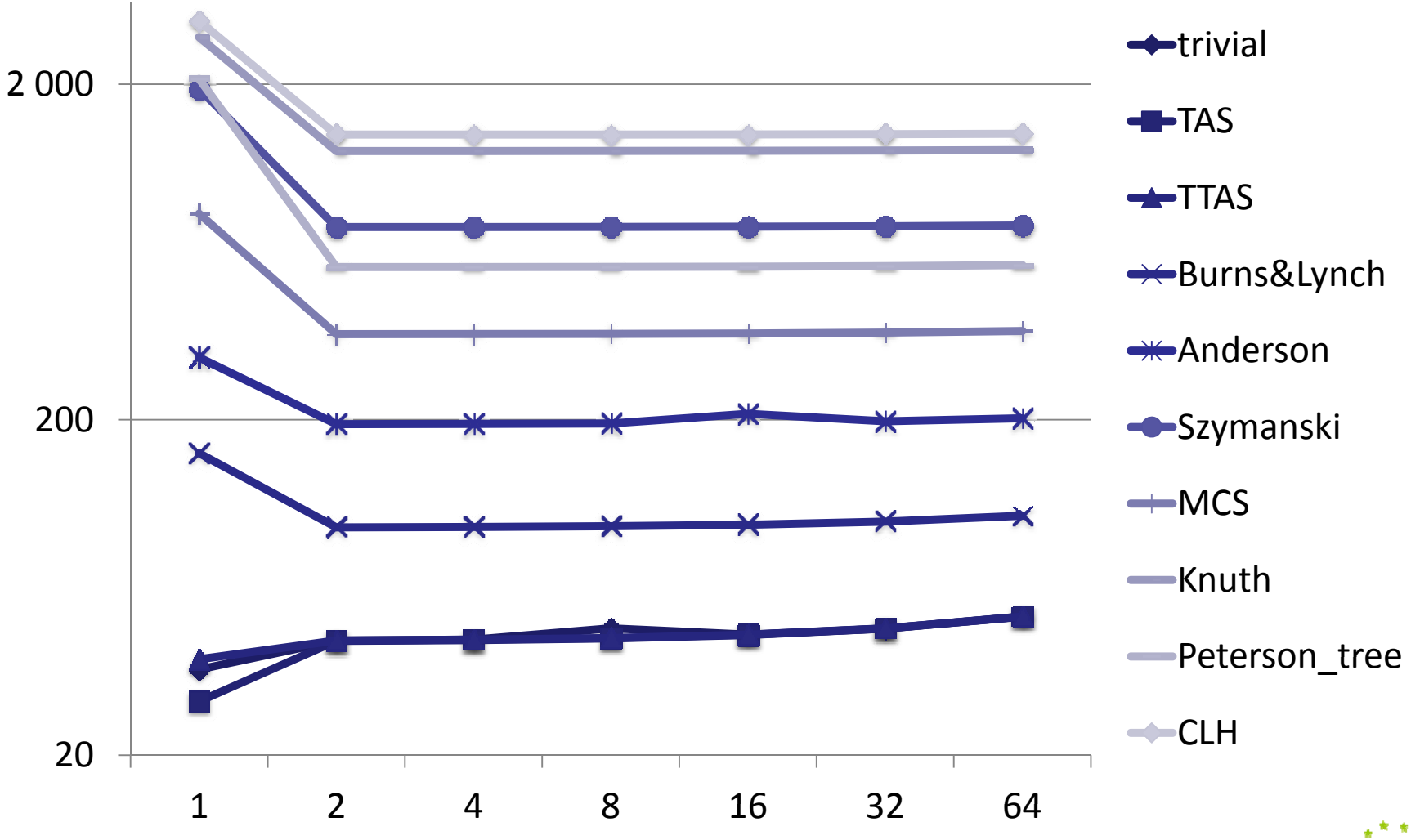


Generation: Speedup Multicore



T-CONFLUENCE REDUCTION

τ-conf. Reduction: Peak Memory (MB)

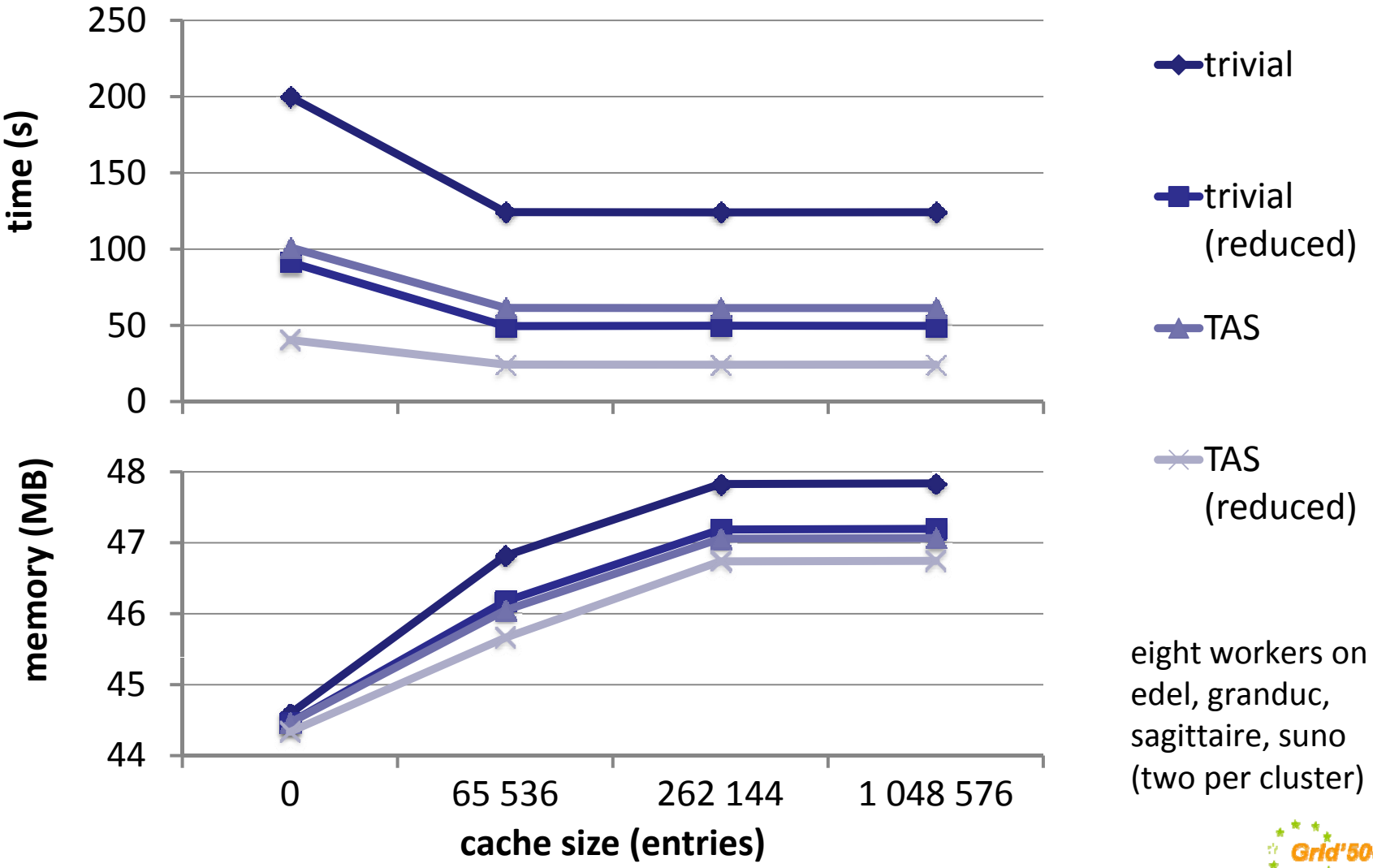


τ-conf. Reduction Comparison

example	LNT.OPEN		BCG_OPEN		PBG_OPEN	
	time	memory	time	memory	time	memory
TAS	4	29	3	33	5	44
trivial	5	36	3	33	5	44
TTAS	4	39	3	34	14	44
Burns&Lynch	43	159	24	91	541	96
Anderson	62	307	51	202	1222	194
MCS	215	822	157	390	3069	360
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	*oom*	*oom*	9227	6930	59932	6015

CACHING OF TRANSITIONS

Caching Successor Lists



eight workers on edel, granduc, sagittaire, suno (two per cluster)

Related Work

- DiVinE
  - Distributed LTL model-checking
  - Fine-grained analysis and tuning of the communication
- Spin
  - Partial-order reductions
- Preach
  - Reachability analysis of Murϕ
  - 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:
  - larger graphs / more workers
  - sequence distributed τ-confluence reduction
  - (distributed) model-checking
  - static load balancing techniques