Demonstrating the Portfolio Analysis in CPAchecker

Thomas Bunk

LMU Munich, Germany

October 01, 2021

Outline

- Motivation
- The CPAchecker Portfolio Analysis
- Evaluation
- Outlook

Motivation

- Emergence of large cloud service providers in recent years that provide large amounts of computing capacities

Motivation

- Emergence of large cloud service providers in recent years that provide large amounts of computing capacities
- These make it feasible to run evaluations based on wall-time measurement

Motivation

- Emergence of large cloud service providers in recent years that provide large amounts of computing capacities
- These make it feasible to run evaluations based on wall-time measurement
- This talk presents a *portfolio solver* implemented in CPAchecker that exploits the availability of such large computation environments.

Outline

- Motivation
- **The CPAchecker Portfolio Analysis**
- Evaluation
- Outlook

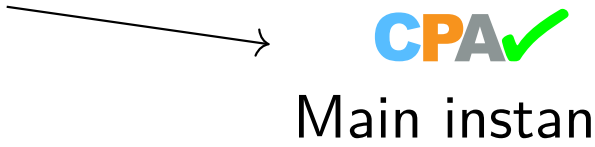
The Portfolio Analysis: An Overview (1/3)

Run CPAchecker using the `portfolio`-configuration:

- ```
  scripts/cpa.sh -portfolio <path/to/program> \
                 -spec <path/to/specification>
  ```
  ```

The Portfolio Analysis: An Overview (2/3)

- The portfolio algorithm runs the sub-analyses in parallel, possibly on large distributed machines.

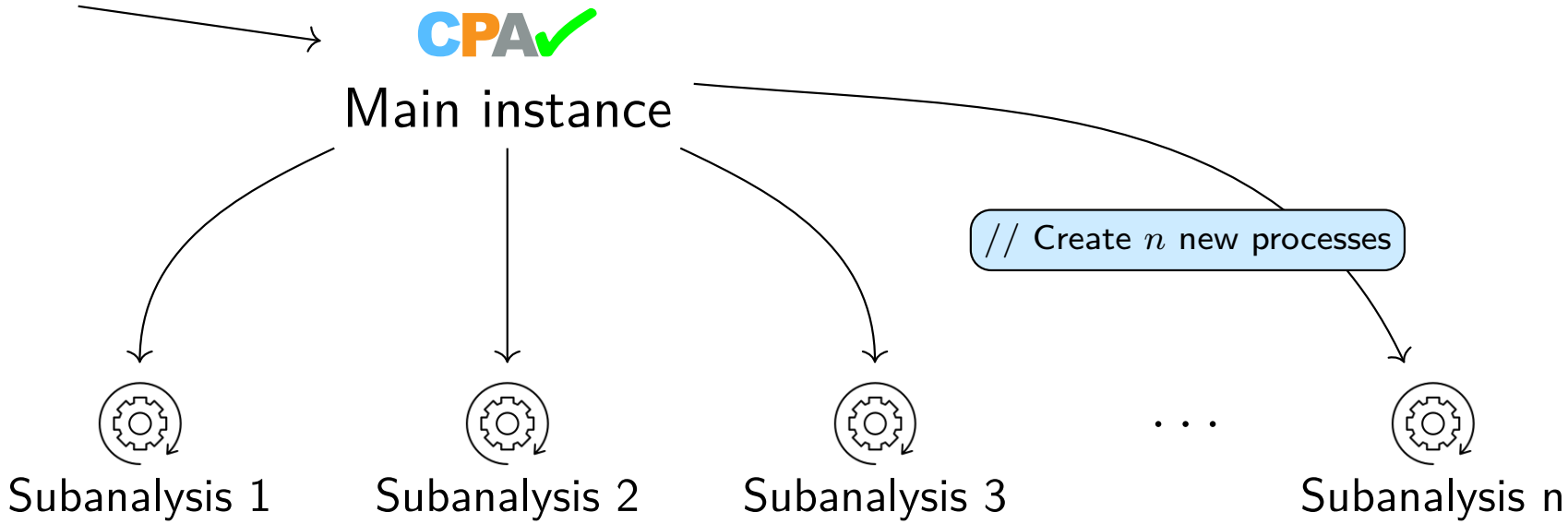


Main instance

Icons taken from Wikimedia Commons

The Portfolio Analysis: An Overview (2/3)

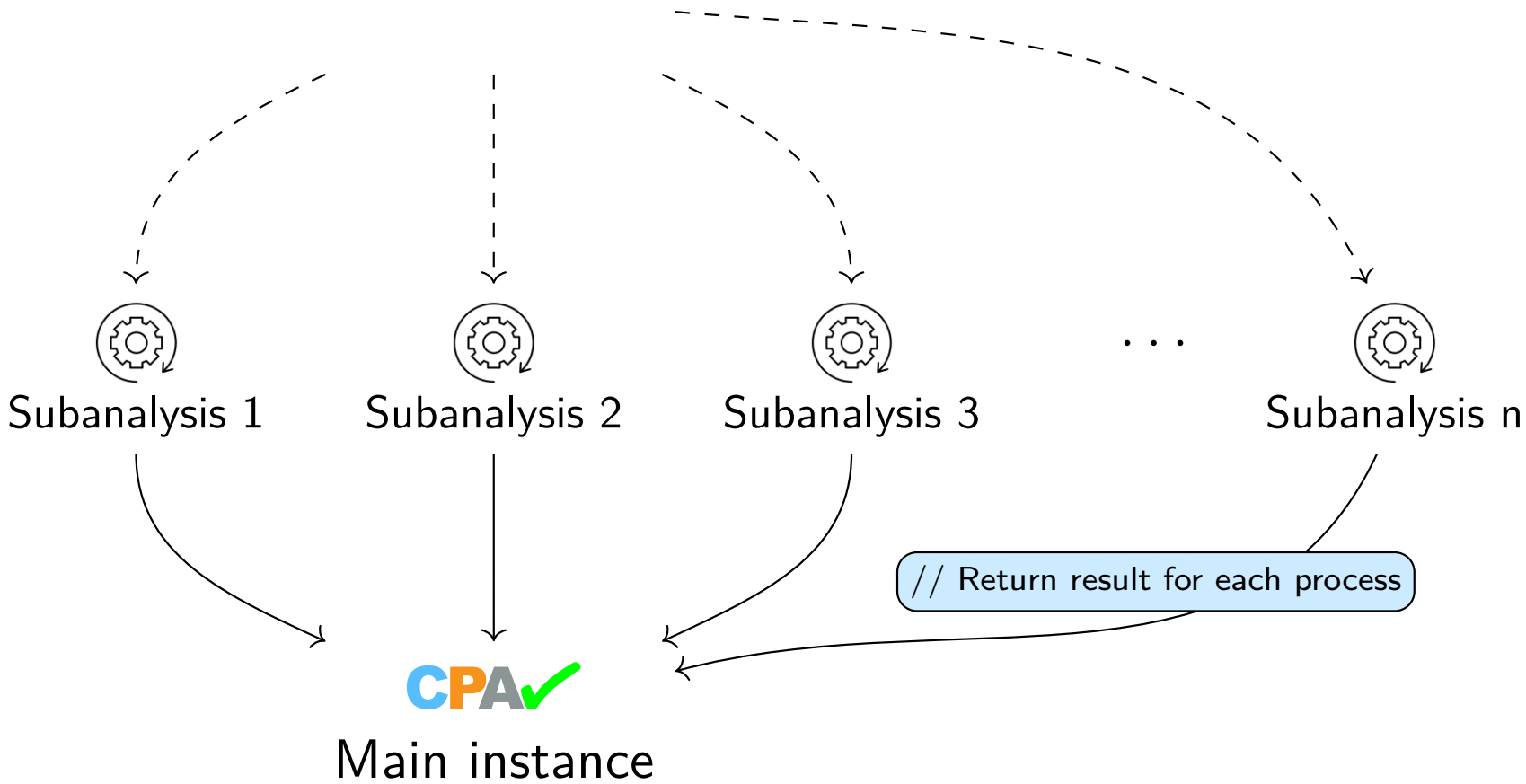
- The portfolio algorithm runs the sub-analyses in parallel, possibly on large distributed machines.



Icons taken from Wikimedia Commons

The Portfolio Analysis: An Overview (3/3)

- Based on the results of the sub-analyses, the main instance concludes with a final result



Icons taken from Wikimedia Commons

Requirements

- The analysis starts as regular CPAchecker-instance

Requirements

- The analysis starts as regular CPAchecker-instance
- That runs the sub-analyses in parallel

Requirements

- The analysis starts as regular CPAchecker-instance
- That runs the sub-analyses in parallel
- . . . not only on the same machine, but also on large distributed systems!

Requirements

- The analysis starts as regular CPAchecker-instance
- That runs the sub-analyses in parallel
- . . . not only on the same machine, but also on large distributed systems!

Approach:

The *Message Passing Interface* (MPI) standard

The MPI Standard

- Standardized interface for passing messages between processes
- Supports both point-to-point and collective communication (e.g., multicasts or broadcasts)
- Supports blocking and non-blocking communication

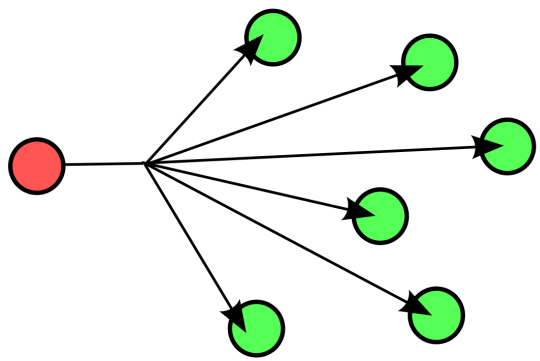


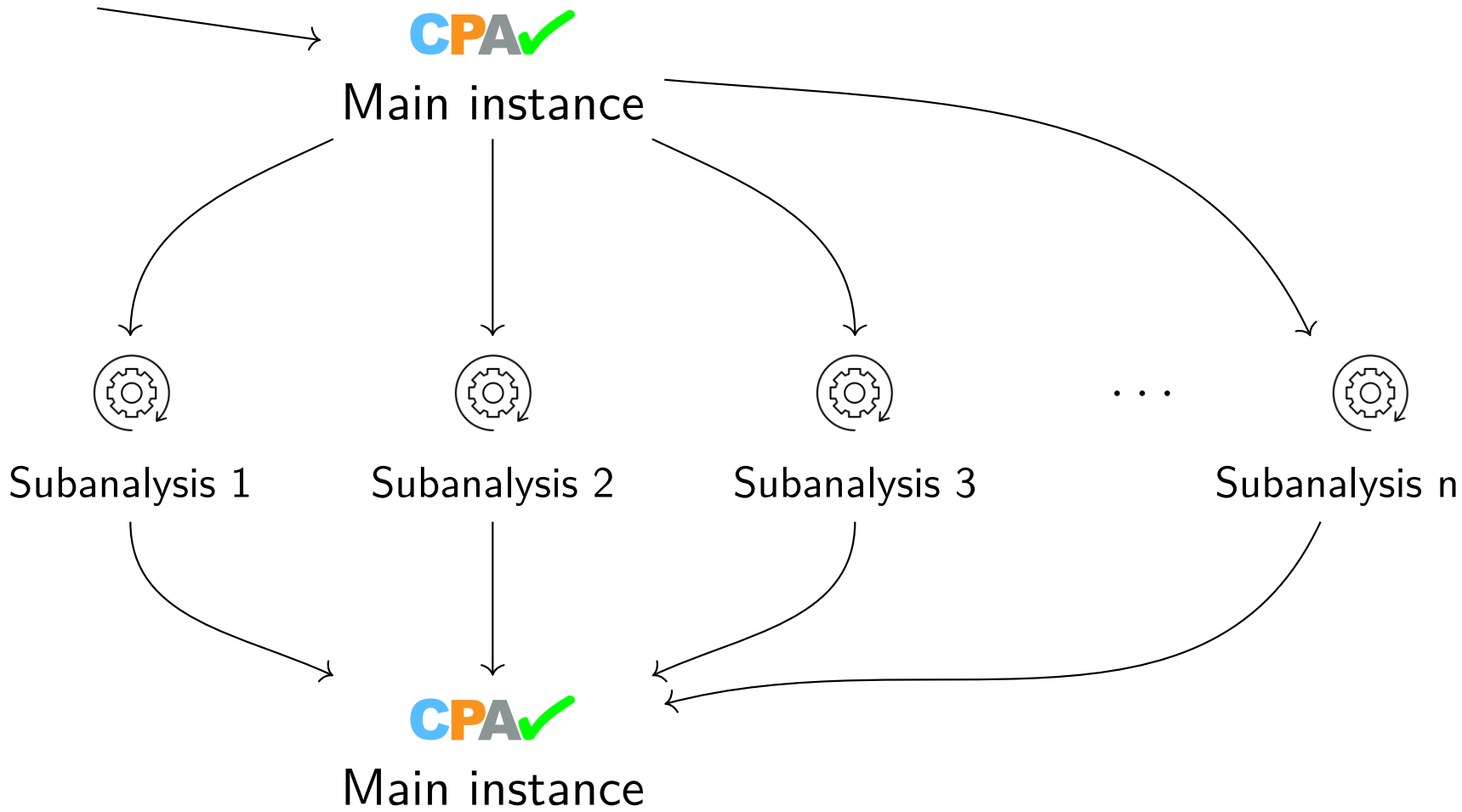
Figure: Example for broadcast communication
(Image from Wikimedia commons)

Scheduling MPI tasks

- MPI defines a standard for launching MPI programs: ***mpiexec***
- Invokes processes on a specified list of hosts
- Highly configurable – allows mapping of processes
  - e.g., to nodes, sockets, cores, L1cache, L2cache, numa, board, and more
- Example command line:

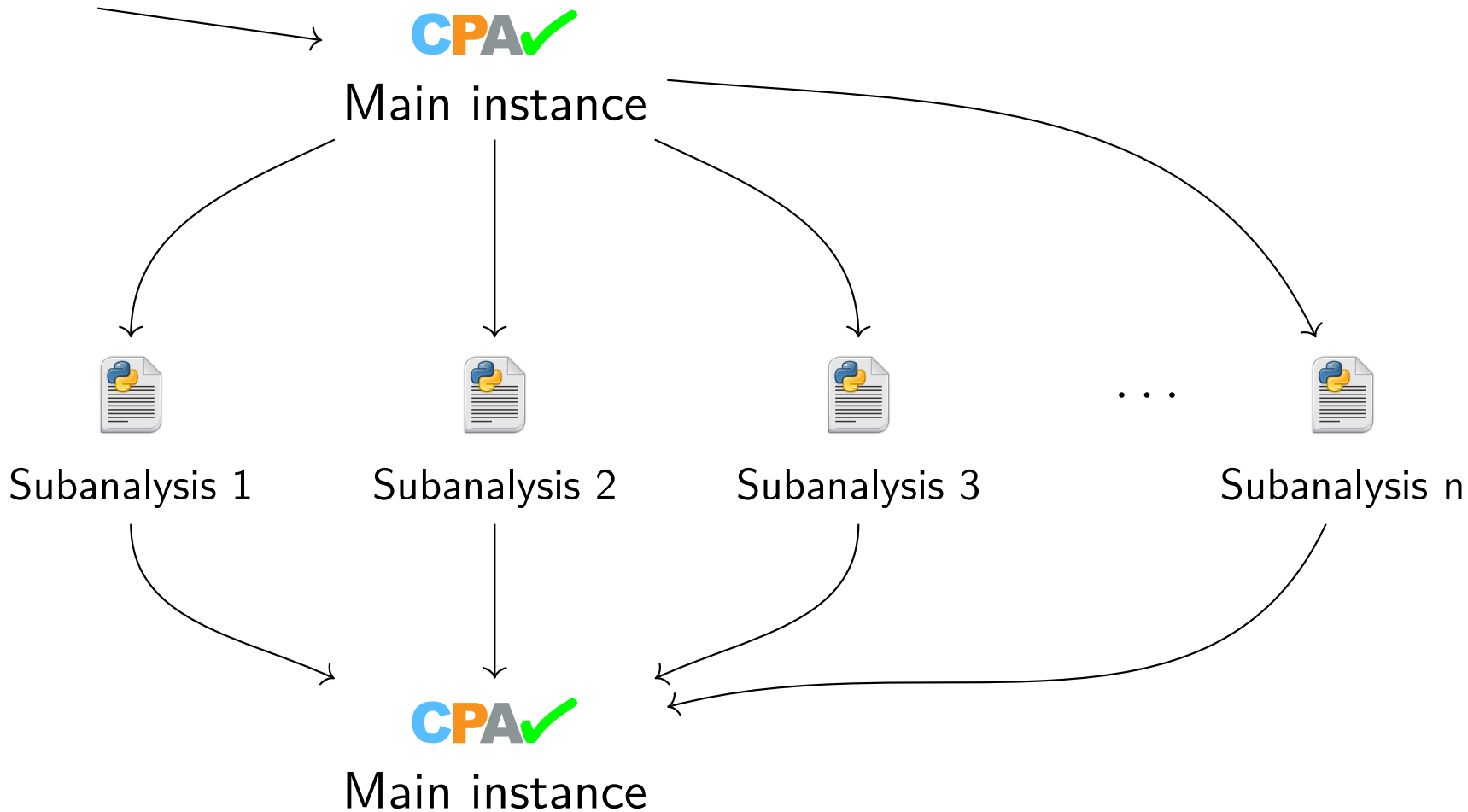
```
> mpiexec -hostfile <hostfile> -np 4 -map-by core <executable>
```

MPI in the Portfolio Analysis



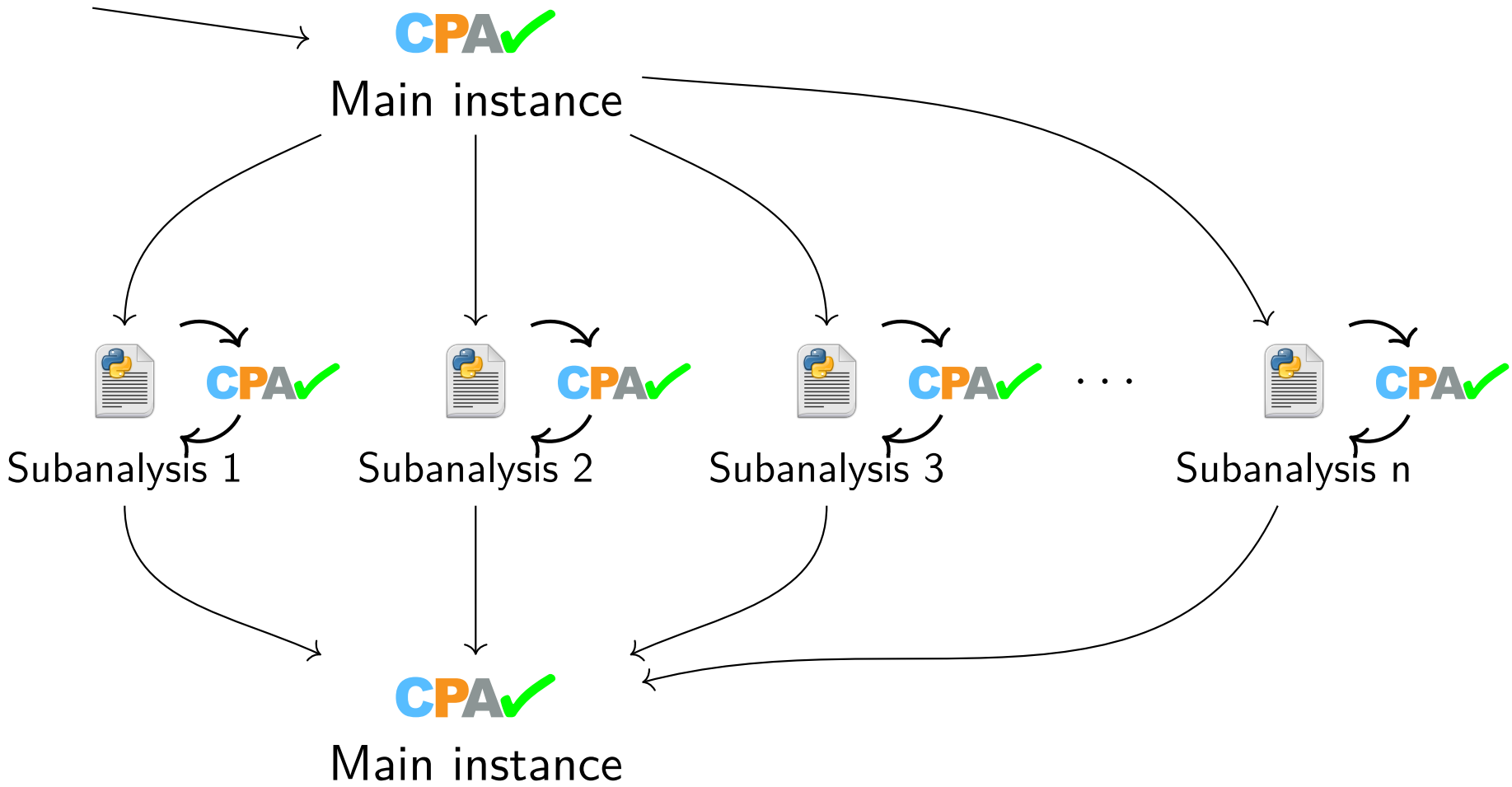
Icons taken from iconarchive.com

MPI in the Portfolio Analysis



Icons taken from iconarchive.com

MPI in the Portfolio Analysis



Icons taken from iconarchive.com

Choosen Sub-analyses in Portfolio Analysis

- Current selected analyses (specified in `config/portfolio.properties`):
  1. *ValueAnalysis* with *CEGAR*
  2. *ValueAnalysis*
  3. *Predicate-Analysis*
  4. *k*-Induction
  5. BMC
  6. BAM
  7. PDR
- Analyses perform an additional CEX-check if concluding with a `FALSE` result

Live Demo

Outline

- Motivation
- The CPAchecker Portfolio Analysis
- **Evaluation**
- Outlook

Evaluation (1/3)

See results in wall-time demo track from SV-COMP'20:

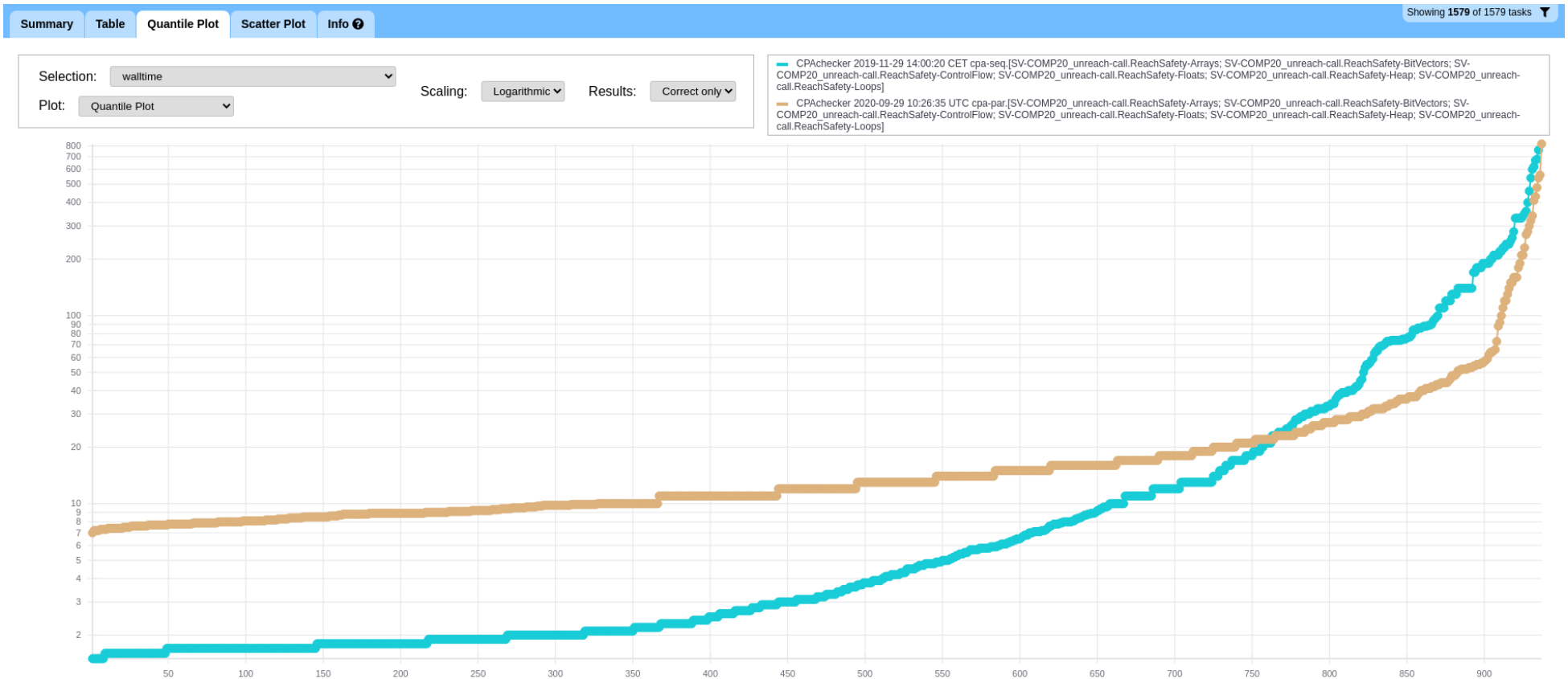
- https://sv-comp.sosy-lab.org/2020/results-demo/results-verified

Evaluation (2/3)

	CPA-Seq			CPA-Par		
	status	raw score	walltime (s)	status	raw score	walltime (s)
total	1 567	**1 492**	**440 000**	1 579	**1 556**	**510 000**
correct results	935	1 588	25 000	937	1 604	22 000
correct true	653	1 306	18 000	667	1 334	15 000
correct false	282	282	7 100	270	270	7 100
incorrect results	6	−96	400	3	−48	55
incorrect true	0	-	-	0	-	-
incorrect false	6	−96	400	3	−48	55
score (SV-COMP)	-	1 746	-	-	1 754	-

Categories: ReachSafety – Arrays, BitVectors, ControlFlow, Floats, Heap, Loops

Evaluation (3/3)



- sv-comp.sosy-lab.org/2020/results-demo/results-verified/META_-ReachSafety.table.html#/quantile?hidden=2,3,4,5&selection=walltime

Outline

- Motivation
- The CPAchecker Portfolio Analysis
- Evaluation
- **Outlook**

Outlook

- Run analysis on the whole benchmark set
- Minimize ramp-up time
- Tweak sub-analyses more towards features of program (loops, floats, recursion, etc.)

Thank you for your attention!