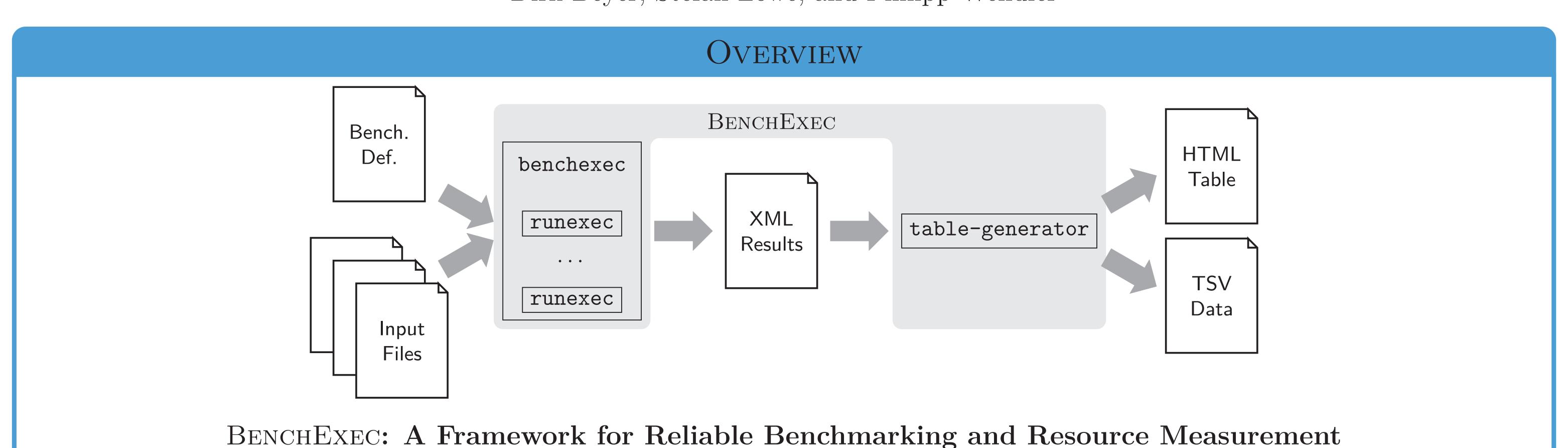


BenchExec

Reliable Benchmarking: Requirements and Solutions

LUDWIG-**MAXIMILIANS-**UNIVERSITÄT MÜNCHEN

Dirk Beyer, Stefan Löwe, and Philipp Wendler



Benchmarking Requirements

- 1. Measure and Limit Resources Accurately
- 2. Terminate Processes Reliably
- 3. Assign Cores Deliberately
- 4. Respect Nonuniform Memory Access
- 5. Avoid Swapping
- 6. Isolate Individual Runs

SCOPE

- Linux systems
- CPU-bound tool (negligible I/O)
- No use of other resources such as GPUs
- No networking / distributed execution
- No user interaction
- No malicious intent
- \Rightarrow Great for solvers, verifiers, etc.!

USE CASES

- Low-level command for isolated, limited, and measured execution of a tool
- Integration in other benchmarking frameworks via command line and Python API (used by StarExec)
- Benchmarking with large number of runs
- Competition execution (used e.g. by SV-COMP since 2016)
- Regression testing

TECHNIQUES AND FEATURES

Benchmarking containers implemented with Linux features such as

- Control groups (cgroups) for resource limitation and measurements (compatible with cgroups v1 and v2)
- Namespaces for isolation
- Overlay filesystem (overlayfs) for intercepting file writes

(same techniques as used by Docker, etc.)

- Parallel execution of tools
- Automatic calculation of distribution of cores and memory regions
- Knows about NUMA and hyper threading
- Configurable file-system layout in container (hide directories, allow write access, etc.)

EXECUTION runexec runexec Run Run Process Process Resource Limitation / Resource Limitation / Measurement Measurement Memory

TABLE-GENERATOR

- Combine results from several executions
- Define table layout
- Select and filter results
- Compute statistics
- Export raw data as TSV
- Generate interactive tables as stand-alone HTML files
- Quantile and scatter plots
- Live analysis of data



Interactive online example

Summary Tal	ole	Quantile Plot	Scatter Plot	Info 🚱					Showing 46	of 46 tasks
Fixed task:			CBMC 2015-12-11 12:11:02 CET cbmc				CPAchecker 2015-12-11 10:59:27 CET predicateA			
Click here to select columns			status	cputime (s)	walltime (s)	memUsage (MB)	status	cputime (s)	walltime (s)	memUsag (MB)
		text	Show all 🗸				Show all v			
ntdrivers/cdaudio_false-unreach-call.i			TIMEOUT	60.1	60.1	987	TIMEOUT	60.7	38.8	3070
ntdrivers/diskperf_false-unreach-call.i			false(reach)	1.24	1.26	104	false(reach)	15.8	9.00	256
ntdrivers/floppy_false-unreach-call.i.c			false(reach)	3.10	3.11	140	false(reach)	22.0	11.9	439
ntdrivers/kbfiltr_false-unreach-call.i.ci			TIMEOUT	60.1	60.1	573	false(reach)	10.8	6.08	219
ntdrivers/parport_f	alse-	unreach-call.i.	false(reach)	13.0	13.3	676	ERROR (recursion)	10.2	5.94	249
ntdrivers/cdaudio_	true-	unreach-call.i.	TIMEOUT	60.1	60.4	993	true	17.0	9.18	295
ntdrivers/diskperf_	true-	unreach-call.i.	TIMEOUT	60.2	60.1	3500	true	15.7	8.89	273
ntdrivers/floppy2_t	rue-u	ınreach-call.i.c	TIMEOUT	60.1	60.0	989	TIMEOUT	60.9	39.7	3010
ntdrivers/floppy_tr	ue-ur	reach-call.i.cil	TIMEOUT	60.0	60.4	391	true	35.6	20.2	1260
ntdrivers/parport_t	rue-u	ınreach-call.i.c	TIMEOUT	61.1	61.7	2180	ERROR (recursion)	9.85	6.13	251
ssh/s3_cInt.blast.0	1_fa	se-unreach-ca	false(reach)	11.6	12.0	411	false(reach)	15.2	8.35	268
ssh/s3_clnt.blast.0)2_fa	se-unreach-ca	false(reach)	3.79	4.15	182	false(reach)	17.1	9.61	270
ssh/s3_cInt.blast.0	3_fa	se-unreach-ca	false(reach)	3.78	3.82	182	false(reach)	16.7	9.32	269
ssh/s3_clnt.blast.0)4_fal	se-unreach-ca	false(reach)	3.77	3.79	180	false(reach)	15.8	8.98	269
ssh/s3_srvr.blast.01_false-unreach-ca			false(reach)	4.91	4.94	242	false(reach)	10.5	6.20	197
ssh/s3_srvr.blast.02_false-unreach-ca			false(reach)	4.87	4.90	229	false(reach)	10.6	5.98	253
ssh/s3_srvr.blast.0)3_fa	lse-unreach-ca	false(reach)	4.89	4.90	229	false(reach)	9.79	5.56	249
aab/a2 am.m.blaat (11 fo	laaaab a	foloo/woodh)	1 01	4.00	220	foloo(#ooolo)	10.0	0.04	05/

PAPER

- STTT 2017
- Open Access
- DOI 10.1007/ s10009-017-0469-y
- Important aspects for benchmarking, hardware influence, how to present results, ...

Thanks to all Contributors!

Aditya Arora, Laura Bschor, Thomas Bunk, Montgomery Carter, Saransh Chopra, Andreas Donig, Karlheinz Friedberger, Peter Häring, Florian Heck, Hugo van Kemenade, George Karpenkov, Mike Kazantsev, Michael Lachner, Thomas Lemberger, Sebastian Ott, Stephan Lukasczyk, Alexander von Rhein, Alexander Schremmer, Dennis Simon, Andreas Stahlbauer, Thomas Stieglmaier, Martin Yankov, Ilja Zakharov, and more (100 in total)!

TOOL BENCHEXEC

- License Apache 2.0
- No root access required for benchmarking
- Available on PyPI and github.com/ sosy-lab/benchexec



