Modern Architecture and Improved UI for Tables of BenchExec

Bachelor Thesis of Laura Bschor

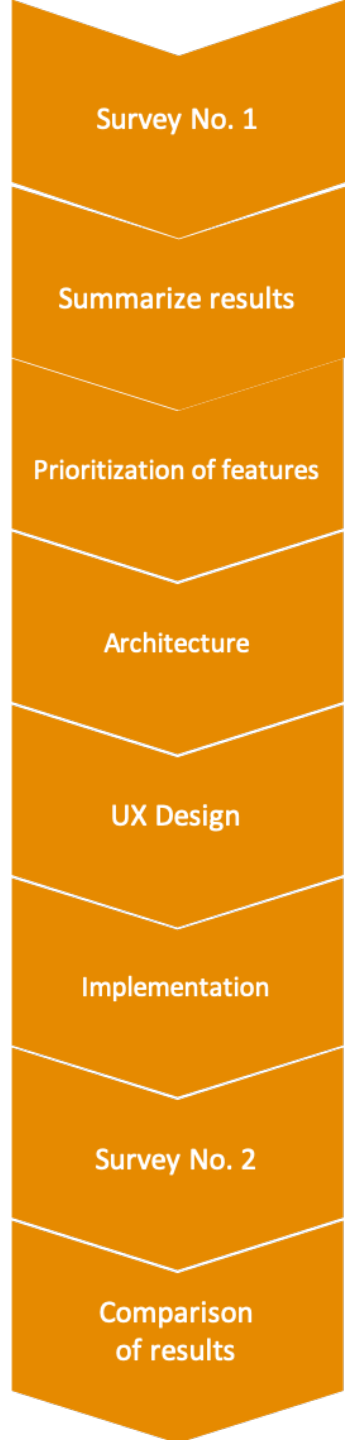
in Media Computer Science

Table Generator

- One of the major features of BenchExec
- Interactive visualisation of BenchExec's results
- Basic implementation: First commits in 2015 (implemented earlier)
- Adaptions through user requests

Goal:

State-of-the-art, holistic and intuitively usable application



Select Columns | Filter Rows | Quantile Plot | Scatter Plot | Shrink Header | Generated with BenchExec

Tool	CPAchecker trunk:32218
Limits	timelimit: 90 s, memlimit: 3000 MB, CPU core limit: 2
Host	[duernbach; ecknach; egau; fensterbach; frommbach; gaissa; geltnach; gloett; guenz; haselgraben; hasslach; huehnerbach; ilm; ilz; inn; kirnach; kronach; leubas; loisach; mangfall; mehnach; naab; ostrach; oybach; partnach; peitnach; pfettrach; saale]
OS	Linux 4.15.0-66-generic
System	CPU: Intel Core i5-4590 @ 3.30 GHz, cores: 4, frequency: 3000 MHz, Turbo Boost: enabled; RAM: 33511 MB
Date of execution	2019-11-05 09:39:59 CET
Run set	integration-induction
Options	-noout -heap 2000M -kInduction

test/programs/		status	cputime (s)	walltime (s)	memory (MB)	host
simple/bitvectors/pointer_extension2_false-unreach-label.i	unreach-label	false	3.82	2.08	143	guenz
simple/bitvectors/pointer_extension3_false-unreach-label.i	unreach-label	false	3.93	2.17	145	ilz
simple/bitvectors/pointer_extension_false-unreach-label.i	unreach-label	false	3.92	2.13	147	ecknach
simple/bitvectors/struct_false-unreach-label.i	unreach-label	false	4.47	2.44	169	peitnach
simple/bitvectors/struct_pointer_simple_false-unreach-label.i	unreach-label	false	4.47	2.43	160	ostrach
simple/bitvectors/struct_simple.2.again_false-unreach-label.i	unreach-label	false	4.50	2.48	165	geltnach
simple/bitvectors/struct_simple.2_false-unreach-label.i	unreach-label	false	4.39	2.36	172	leubas
simple/bitvectors/struct_simple_false-unreach-label.i	unreach-label	false	4.50	2.43	170	guenz
simple/bitvectors/pointer_extension_true-unreach-label.i	unreach-label	**false**	3.91	2.12	147	geltnach
simple/bitvectors/struct_pointer_cast.indirect_true-unreach-label.i	unreach-label	true	4.32	2.36	157	ecknach
simple/bitvectors/struct_pointer_cast_simplified.indirect_true-unreach-label.i	unreach-label	true	4.06	2.21	158	kirnach
simple/bitvectors/struct_pointer_cast_true-unreach-label.i	unreach-label	true	3.93	2.15	150	naab
simple/bitvectors/struct_pointer_simple_assignbeforeset_simplified_true-unreach-label.i	unreach-label	true	4.12	2.26	150	ecknach
simple/bitvectors/struct_pointer_simple_assignbeforeset_true-unreach-label.i	unreach-label	true	4.13	2.27	150	mehnach
simple/bitvectors/struct_pointer_simple_change_simplified_true-unreach-label.i	unreach-label	true	4.09	2.23	148	ostrach
simple/bitvectors/struct_pointer_simple_change_true-unreach-label.i	unreach-label	true	4.12	2.28	157	leubas
simple/bitvectors/struct_pointer_simple_pointerchange_simplified_true-unreach-label.i	unreach-label	true	4.27	2.33	166	leubas
simple/bitvectors/struct_pointer_simple_pointerchange_true-unreach-label.i	unreach-label	true	4.25	2.35	156	egau
simple/bitvectors/struct_pointer_simple_reverse_simplified_true-unreach-label.i	unreach-label	true	4.03	2.23	166	haselgraben
simple/bitvectors/struct_pointer_simple_reverse_true-unreach-label.i	unreach-label	true	3.76	2.08	148	loisach
simple/bitvectors/struct_pointer_simple_simplified_true-unreach-label.i	unreach-label	true	4.31	2.55	163	loisach
simple/bitvectors/struct_pointer_simple_true-unreach-label.i	unreach-label	true	4.30	2.35	165	gloett
simple/bitvectors/struct_ptrCast_reverse_true-unreach-label.i	unreach-label	true	4.27	2.31	160	oybach
simple/bitvectors/struct_ptrCast_reverse_typesafe_true-unreach-label.i	unreach-label	true	4.08	2.21	163	gloett
simple/bitvectors/struct_ptrCast_true-unreach-label.i	unreach-label	true	4.39	2.38	168	loisach
simple/bitvectors/struct_simple_true-unreach-label.i	unreach-label	true	4.28	2.32	154	fensterbach
simple/bitvectors/struct_true-unreach-label.i	unreach-label	true	4.47	2.48	157	egau
simple/bitvectors/structproblem_casts_true-unreach-label.i	unreach-label	true	4.00	2.17	150	naab
simple/bitvectors/structproblem_simple_true-unreach-label.i	unreach-label	true	3.90	2.15	142	duernbach
simple/pointer_aliasing/deferred_allocations_comparison_false-unreach-label.i	unreach-label	false	6.10	3.24	198	peitnach

Why improving?

```
1
2
3  function addButtons(target, callback, showQuantileButton) {
4      target = $('<div>', {id: 'buttons'}).appendTo(target);
5      if (showQuantileButton) {
6          $('<button>', {id:'button-quantile'}).appendTo(target)
7              .click(function() { graphData.isQuantile = !graphData.isQuantile; callback();})
8              .text(graphData.isQuantile ? 'Switch to Direct Plot' : 'Switch to Quantile Plot'
9          $('<br>').appendTo(target);
10     }
11
12     $('<button>', {id:'button-logScale'}).appendTo(target)
13         .click(function() { graphData.isLogScale = !graphData.isLogScale; callback();})
14         .text(graphData.isLogScale ?  'Switch to Linear Scale' : 'Switch to Logarithmic Scal
15     $('<br>').appendTo(target);
16
17     $('<button>', {id:'button-showCorrectOnly'}).appendTo(target)
18         .click(function() { graphData.showCorrectOnly = !graphData.showCorrectOnly; callback
19         .text(graphData.showCorrectOnly ?  'Switch to All Results' : 'Switch to Correct Resu
20     $('<br>').appendTo(target);
21 }
22
23 function isValidStatisticsColumn(columnIndex, allowStatus) {
24   var firstRow = $('#dataTable').find('> tbody > tr:visible')[0];
25   var firstColumnEntry = firstRow.cells[dataColumnsOffset + columnIndex];
26   var columnClasses = firstColumnEntry.classList;
27
28   if (!$(columnTitleCells[columnIndex]).is(':visible')) {
29     return false;
```

Technical status
Theoretical background
Related work
Survey results

Requirements

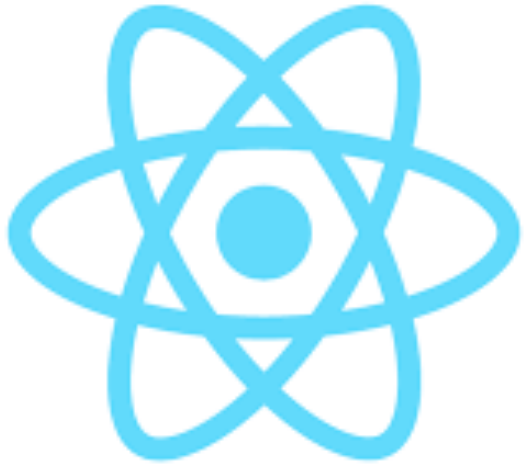
- Improving maintainability
- New structure of features
- Improving effectiveness for users
- Improving intuitiveness for users
- Reducing loading and waiting time
- Providing new features like sorting or pagination

> “ […] *Provide a newly implemented application based on a modern, lightweight, structuring, maintainable and fast state-of-the-art JavaScript framework*

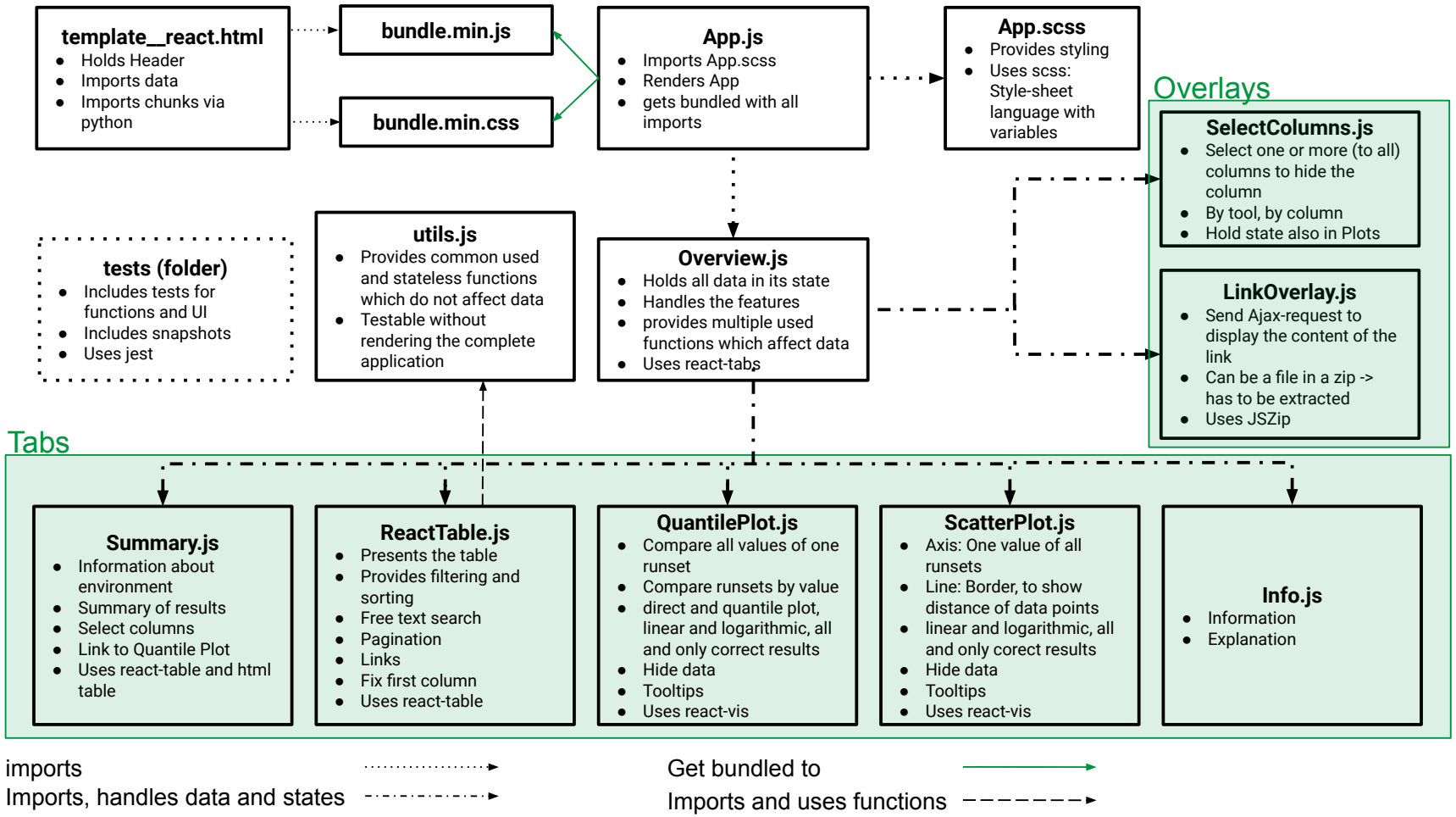
Software Architecture

- React as chosen framework

- Component-based javaScript library
- Keep track of states
- Render only the asked components at runtime
- Handle and interact with data
- Popular, well documented and is continuously ir
  (by Facebook)



Structure of Features and Deployment Process



Adapting the data structure

Old Version

```
<tbody>
{{for line in body}}
<tr>{{if line.has_sourcefile}}<td title="Clic
  {{for id, show in zip(line.id[1:], relevant
    {{if show}}

//...
//...

var run_sets = {{run_sets|json}};
var columns = {{columnTitles|json}};
```

New Version

```
const data = {
  head: {{head|json}},
  tools: {{tools|json}},
  rows: {{rows|json}},
  stats: {{stats|json}},
  props: {{relevant_id_columns|json}}
};
```

Improving the UI (User Interface)

Summary | Table (764) | Quantile Plot | Scatter Plot | Info

Environment

Tool	CPAchecker trunk:32218
Limits	timelimit: 90 s, memlimit: 3000 MB, CPU core limit: 2
Host	[duernbach; ecknach; egau; fensterbach; frommbach; gaissa; geltnach; gloett; guenz; haselgraben; hasslach; huehnerbach; ilm; ilz; inn; kirnach; kronach; leubas; loisach; mangfall; mehnach; naab; ostrach; oybach; partnach; peitnach; pfettrach; saale]
OS	Linux 4.15.0-66-generic
System	CPU: Intel Core i5-4590 @ 3.30 GHz, cores: 4, frequency: 3000 MHz, Turbo Boost: enabled; RAM: 33511 MB
Date of execution	2019-11-05 09:39:59 CET
Run set	integration-induction

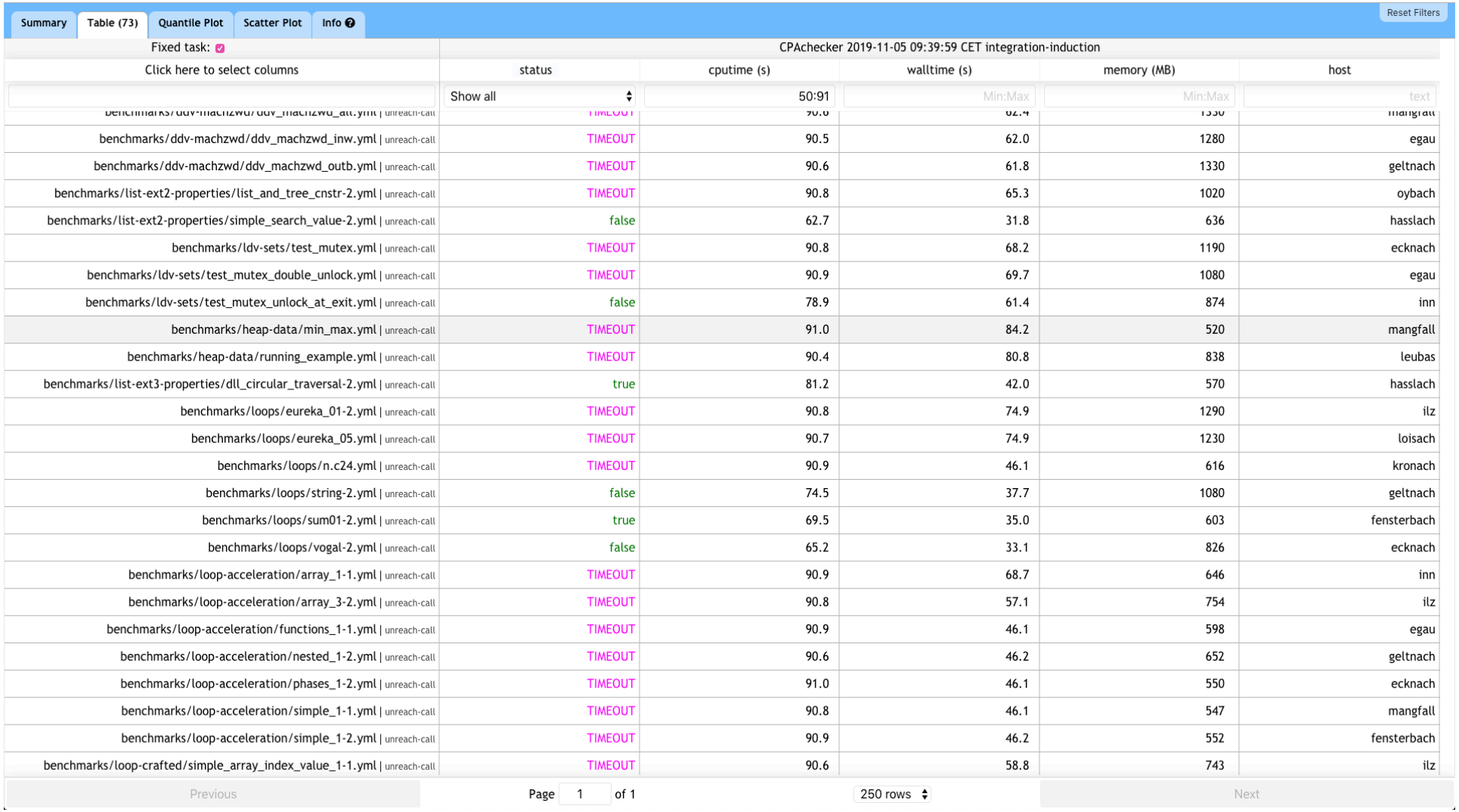
Options	`-noout`
	`-heap 2000M`
	`-kInduction`

Summary

Fixed row title: ☑	CPAchecker 2019-11-05 09:39:59 CET integration-induction				
Click here to select columns	status	cputime (s)	walltime (s)	memory (MB)	host
total	764	26700	15100	341000	-
correct results	507	5760	3070	144000	-
correct true	324	3600	1920	90100	-
correct false	183	2150	1160	53500	-
incorrect results	14	188	97.3	4430	-
incorrect true	4	12.7	7.06	526	-
incorrect false	10	176	90.3	3900	-
score (764 tasks, max score: 212)	543	-	-	-	-

Generated by BenchExec 2.3-dev

Improving the UI (User Interface)



Summary | Table (73) | Quantile Plot | Scatter Plot | Info — Reset Filters

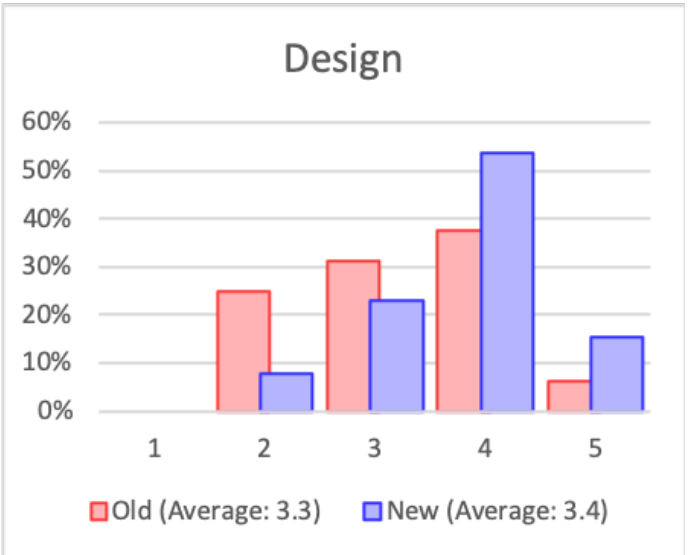
Fixed task: ☑ — CPAchecker 2019-11-05 09:39:59 CET integration-induction

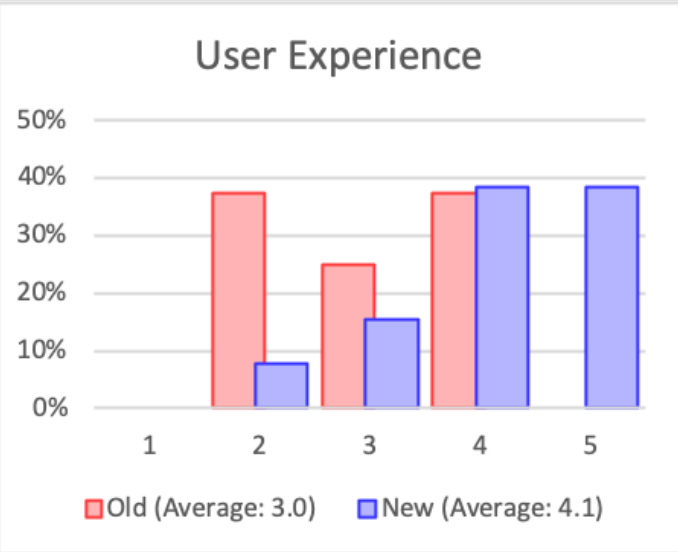
Click here to select columns

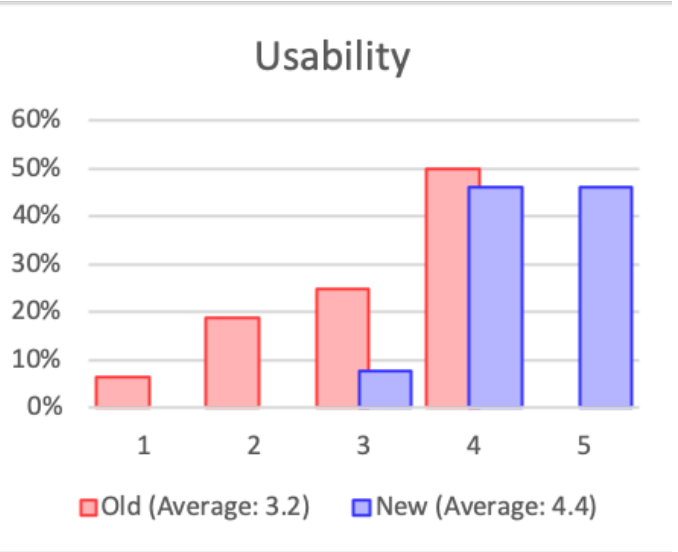
	status	cputime (s)	walltime (s)	memory (MB)	host
	Show all	50:91	Min:Max	Min:Max	text
benchmarks/ddv-machzwd/ddv_machzwd_inw.yml | unreach-call	TIMEOUT	90.5	62.0	1280	egau
benchmarks/ddv-machzwd/ddv_machzwd_outb.yml | unreach-call	TIMEOUT	90.6	61.8	1330	geltnach
benchmarks/list-ext2-properties/list_and_tree_cnstr-2.yml | unreach-call	TIMEOUT	90.8	65.3	1020	oybach
benchmarks/list-ext2-properties/simple_search_value-2.yml | unreach-call	false	62.7	31.8	636	hasslach
benchmarks/ldv-sets/test_mutex.yml | unreach-call	TIMEOUT	90.8	68.2	1190	ecknach
benchmarks/ldv-sets/test_mutex_double_unlock.yml | unreach-call	TIMEOUT	90.9	69.7	1080	egau
benchmarks/ldv-sets/test_mutex_unlock_at_exit.yml | unreach-call	false	78.9	61.4	874	inn
benchmarks/heap-data/min_max.yml | unreach-call	TIMEOUT	91.0	84.2	520	mangfall
benchmarks/heap-data/running_example.yml | unreach-call	TIMEOUT	90.4	80.8	838	leubas
benchmarks/list-ext3-properties/dll_circular_traversal-2.yml | unreach-call	true	81.2	42.0	570	hasslach
benchmarks/loops/eureka_01-2.yml | unreach-call	TIMEOUT	90.8	74.9	1290	ilz
benchmarks/loops/eureka_05.yml | unreach-call	TIMEOUT	90.7	74.9	1230	loisach
benchmarks/loops/n.c24.yml | unreach-call	TIMEOUT	90.9	46.1	616	kronach
benchmarks/loops/string-2.yml | unreach-call	false	74.5	37.7	1080	geltnach
benchmarks/loops/sum01-2.yml | unreach-call	true	69.5	35.0	603	fensterbach
benchmarks/loops/vogal-2.yml | unreach-call	false	65.2	33.1	826	ecknach
benchmarks/loop-acceleration/array_1-1.yml | unreach-call	TIMEOUT	90.9	68.7	646	inn
benchmarks/loop-acceleration/array_3-2.yml | unreach-call	TIMEOUT	90.8	57.1	754	ilz
benchmarks/loop-acceleration/functions_1-1.yml | unreach-call	TIMEOUT	90.9	46.1	598	egau
benchmarks/loop-acceleration/nested_1-2.yml | unreach-call	TIMEOUT	90.6	46.2	652	geltnach
benchmarks/loop-acceleration/phases_1-2.yml | unreach-call	TIMEOUT	91.0	46.1	550	ecknach
benchmarks/loop-acceleration/simple_1-1.yml | unreach-call	TIMEOUT	90.8	46.1	547	mangfall
benchmarks/loop-acceleration/simple_1-2.yml | unreach-call	TIMEOUT	90.9	46.2	552	fensterbach
benchmarks/loop-crafted/simple_array_index_value_1-1.yml | unreach-call	TIMEOUT	90.6	58.8	743	ilz

Previous — Page 1 of 1 — 250 rows — Next

Results of Survey No. 2

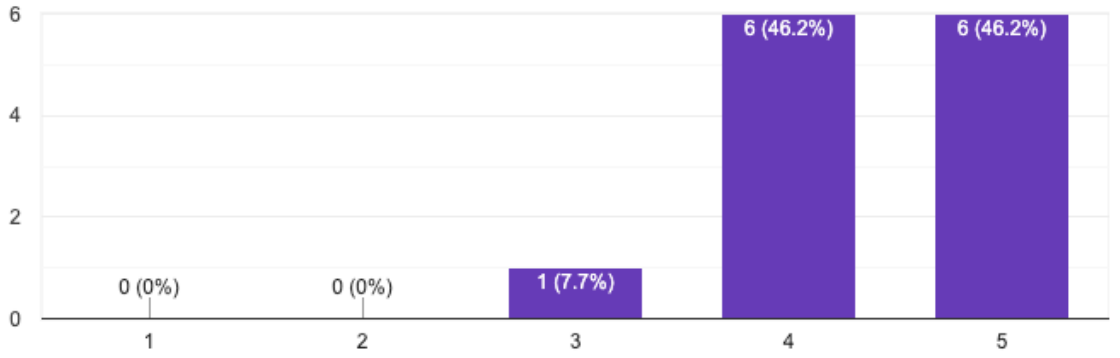




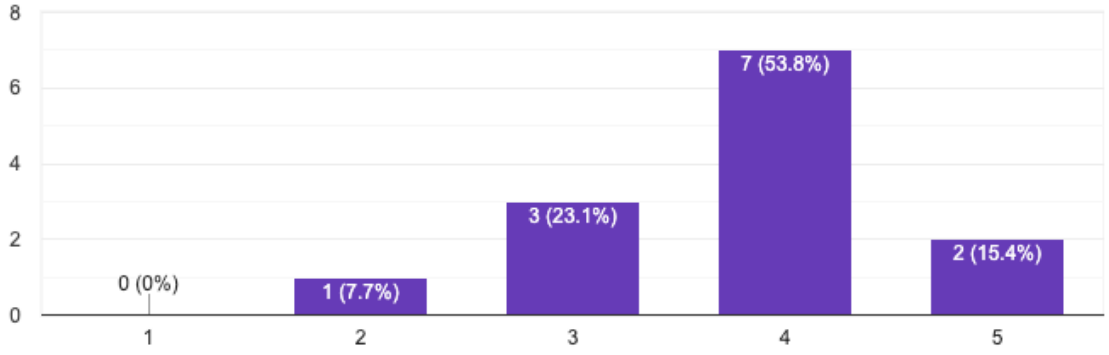


Results of Survey No. Two: Comparison of Time/Velocity

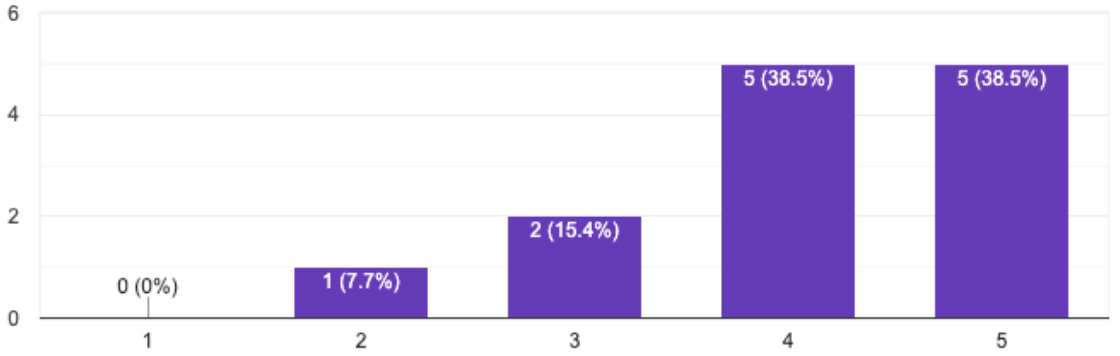
Time to find information



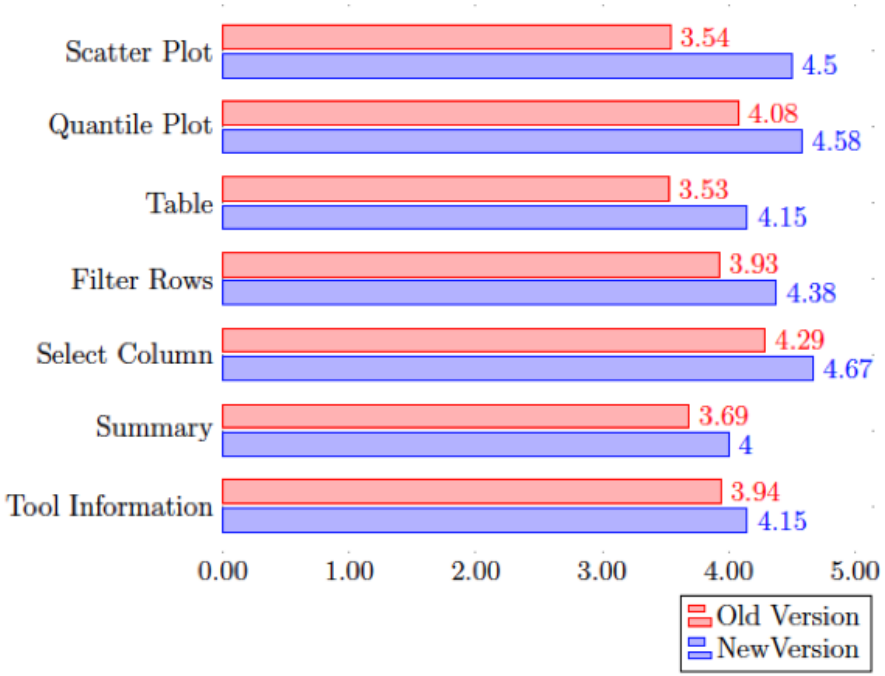
Time for loading



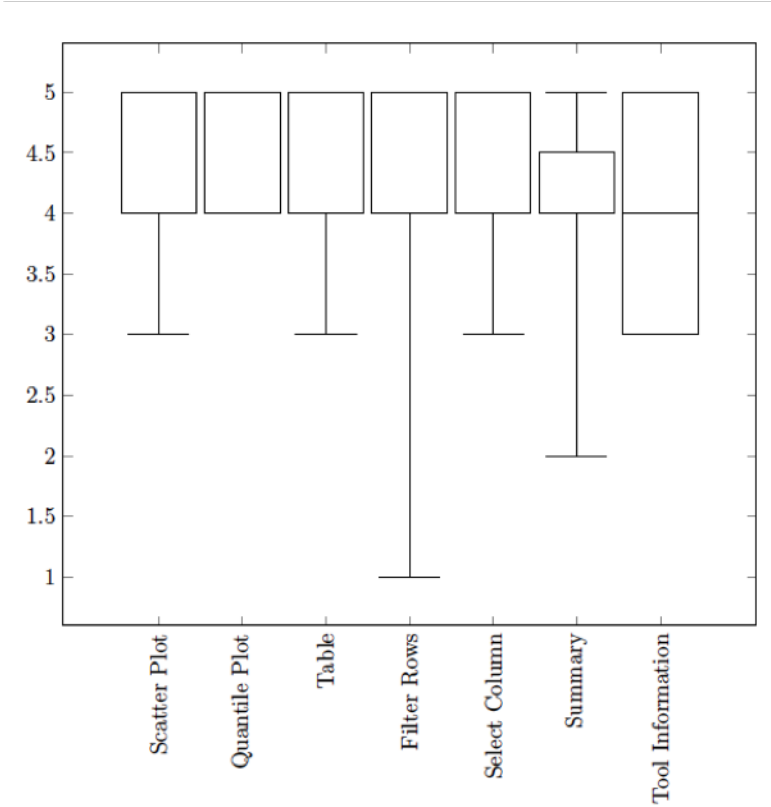
Velocity of interacting



Results of Survey No. 2



Features of both surveys



Features of the new version

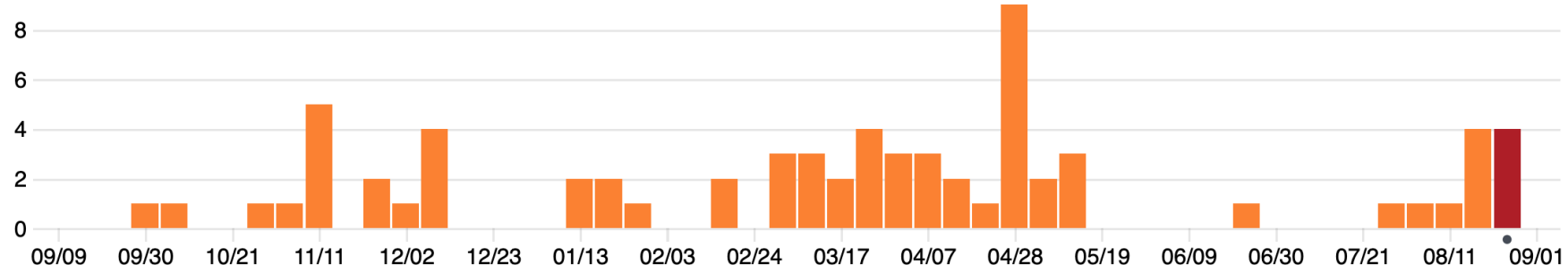
Fulfillment of Requirements

- Improving maintainability
- New structure of features
- Improving effectiveness for users
- Improving intuitiveness for users
- Reducing loading and waiting time
- Providing new features like sorting or pagination

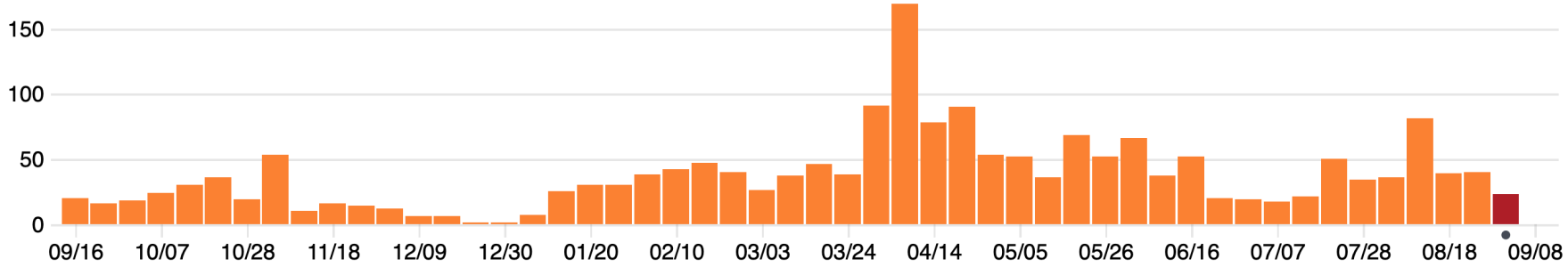
[…] Provide a newly implemented application based on a modern, lightweight, structuring, maintainable and fast state-of-the-art JavaScript framework

Additional Slides

Why React?



Number of Weekly Commits for jQuery in the Past Year –
Screenshot taken on September 8th 2019



Number of Weekly Commits for React in the Past Year –
Screenshot taken on September 9th, 2019

Why React?

Duration for…	Angular-v7.1.4	Rect-v16.6.0	vue v2.6.2
Interacting with the table	1.09	1.10	1.42
Startup metrics	2.27	1.20	1.00
Memory allocation	1.61	1.18	1.01
summary	1.66	1.16	1.14

Comparison of the Average Slowdown Statistics (non-keyed) of JavaScript Frameworks for One Benchmark Example

Popularity of	Vue	React
Contributors	278	1,304
GitHub-Stars	145,663	134,099
Used by	974,844	2,311,749
Posted jobs	1,356	4142

Comparison of the Popularity of React and Vue

Survey Statistics

Survey No. One

- two weeks
- before implementation
- 16 participants
- google forms

- most important output:
  - faster
  - first look: summary

Survey No. Two

- eleven days
- after implementation
- 13 participants
- google forms

- most important output:
  - faster now
  - higher usability