Object Versioning for Flow-Sensitive Pointer Analysis – Errata

Mohamad Barbar*†, Yulei Sui*, Shiping Chen†
*University of Technology Sydney, Australia
†CSIRO's Data61, Australia

MAY 27 2021

This is a follow-up to "Object Versioning for Flow-Sensitive Pointer Analysis" by Mohamad Barbar, Yulei Sui, and Shiping Chen published at CGO 2021 pointing out noticed errors in experimental data and typos in inference rules.

I. EXPERIMENTAL DATA

Table III of the paper contains incorrect timing data in the **Versioning** column (and consequently, the associated **Total time** and **Time diff.** columns). The **Versioning** column is the sum of four statistics emitted by SVF but our statistics collection script was only collecting two of those statistics. Our script attempted to extract the other two statistics but they had their names changed in SVF and the script was not updated to reflect that. Thus, this column should contain larger values.

We have fixed this error and run experiments again, and the correct data is presented in Table I. The result is that the geometric mean speed up is reduced to 3.34× from the previously reported 5.31×. This table should be noted when reading textual reference to the timing data.

VSFS is maintained by the SVF project upstream[1]. The original VSFS implementation was in some parts naive and has seen recent improvement. For example, running the same benchmarking script with the latest development version of SVF, bash takes less than 400 seconds to analyse. Recently, new options have been introduced to SVF to improve performance, and running the analysis with those enabled would further improve time. There are still plans to improve the analysis, particularly during the versioning phase.

[1]See https://github.com/SVF-tools/SVF/wiki/VSFS for status updates.

II. ARTIFACT

The artifact has been appropriately updated and is available at https://doi.org/10.6084/m9.figshare.13269662.v2.

III. RULES

This section lists typos in the rules in Fig. 10 of the paper. These rules are not incorrect, but do not use correct symbols.

A. $[\phi]^F$ and $[CAST]^F$

The p function in the conclusions should be the pt function.

B. $[LOAD]^F$, $[STORE]^F$, and $[SU/WU]^F$

The $c\_$ and $y\_$ functions should be $\mathcal{C}\_$ and $\mathcal{Y}\_$ functions, respectively. The values c and y assigned from these functions should be κ_c and κ_y, respectively.

TABLE I

TIME, IN SECONDS, AND MEMORY USAGE (MAXIMUM RESIDENT SET SIZE), IN GIGABYTES, OF ANDERSEN'S ANALYSIS, SFS (MAIN PHASE TIME ONLY), AND VSFS. TIME STATISTICS ARE SPLIT INTO 3 FOR VSFS: TIME TO VERSION OBJECTS, TIME TO PERFORM THE ANALYSIS USING VERSIONS, AND THE SUM OF THOSE TWO TIMES WHICH IS THEN USED FOR COMPARISON. THE LAST TWO COLUMNS SHOW HOW MANY TIMES FASTER (OR SLOWER) OUR APPROACH IS COMPARED TO SFS AND THE REDUCTION (OR INCREASE) IN MEMORY USAGE OF OUR APPROACH COMPARED TO SFS. OOM MEANS A BENCHMARK WAS UNABLE TO COMPLETE BECAUSE IT EXHAUSTED MEMORY RESOURCES (100 GB).

Benchmark	Andersen's		SFS		VSFS				Time diff.	Mem. diff.
	Time	Mem.	Time	Mem.	Versioning	Main phase	Total time	Mem.		
du	1.16	0.19	21.58	2.24	1.78	3.91	5.70	1.52	3.79×	1.47×
ninja	1.37	0.22	66.64	2.38	1.50	4.25	5.75	1.48	11.59×	1.60×
bake	1.12	0.19	65.33	3.13	6.12	1.41	7.52	1.71	8.68×	1.83×
dpkg	0.63	0.17	3.44	1.43	2.51	1.57	4.08	1.43	0.84×	1.00×
nano	2.92	0.41	76.09	5.64	10.52	16.59	27.11	2.18	2.81×	2.58×
i3	1.22	0.28	3.40	1.57	1.91	0.81	2.72	1.53	1.25×	1.02×
psql	1.05	0.29	9.16	1.84	1.58	0.91	2.50	1.57	3.67×	1.17×
janet	3.16	0.45	115.86	7.12	8.88	8.54	17.43	2.37	6.65×	3.01×
astyle	22.22	1.20	12758.68	84.77	28.32	1239.71	1268.03	19.54	10.06×	4.34×
tmux	24.26	1.24	507.29	12.93	95.58	165.62	261.20	7.84	1.94×	1.65×
mruby	8.06	0.66	17.15	2.78	12.21	3.39	15.60	2.65	1.10×	1.05×
mutt	14.53	1.05	1006.28	26.84	63.39	391.33	454.72	8.10	2.21×	3.31×
bash	21.75	1.56	2247.20	76.64	185.83	1433.76	1619.59	15.63	1.39×	4.90×
lynx	63.61	1.84	OOM	OOM	352.92	12016.72	12369.64	22.03	–	≥5.45×
hyriseConsole	19.22	2.83	770.34	18.62	24.47	40.92	65.39	6.97	11.78×	2.67×
Average									3.34×	≥2.11×