Jay Facultad
Nathan Sahelijo

Laboratory Notes for Homework 7

Benchmarks used:

Big *– sandmark.umz*
Medium *– partial advent.umz*
Small *– midmark.um*

Table 1. Report for Improvements Through Profiling

Benchmark	*Run 1 Time (s)*	*Run 2 Time (s)*	*Run 3 Time (s)*	*Avg Time (s)*	*Instructions*	*Rel to Start*	*Rel to Prev*	*Improvement*	*Bottleneck*	*Phase / Stage*
Big	60.46	56.06	56.04	**57.52**		1.000	1.000			
Medium	20.51	20.23	19.54	**20.09**		1.000	1.000	No improvement (starting point)	Bitpack_getu	1
Small	2.61	2.22	2.22	**2.35**	2.39E+10	1.000	1.000			
Big	48.61	48.83	48.46	**48.63**		0.846	0.846			
Medium	17.1	17.08	17.23	**17.14**		0.853	0.853	Compiled with optimization flag -O1 and linked against -lcii-O1	Bitpack_getu	2
Small	1.94	1.93	1.95	**1.94**	2.22E+10	0.826	0.826			
Big	48.32	48.79	48.46	**48.52**		0.844	0.998			
Medium	17.11	17.26	17.63	**17.33**		0.863	1.011	Compiled with optimization flag -O2 and linked against -lcii-O2	Bitpack_getu	3
Small	1.93	1.95	1.95	**1.94**	2.22E+10	0.827	1.002			

Jay Facultad
Nathan Sahelijo

Laboratory Notes for Homework 7

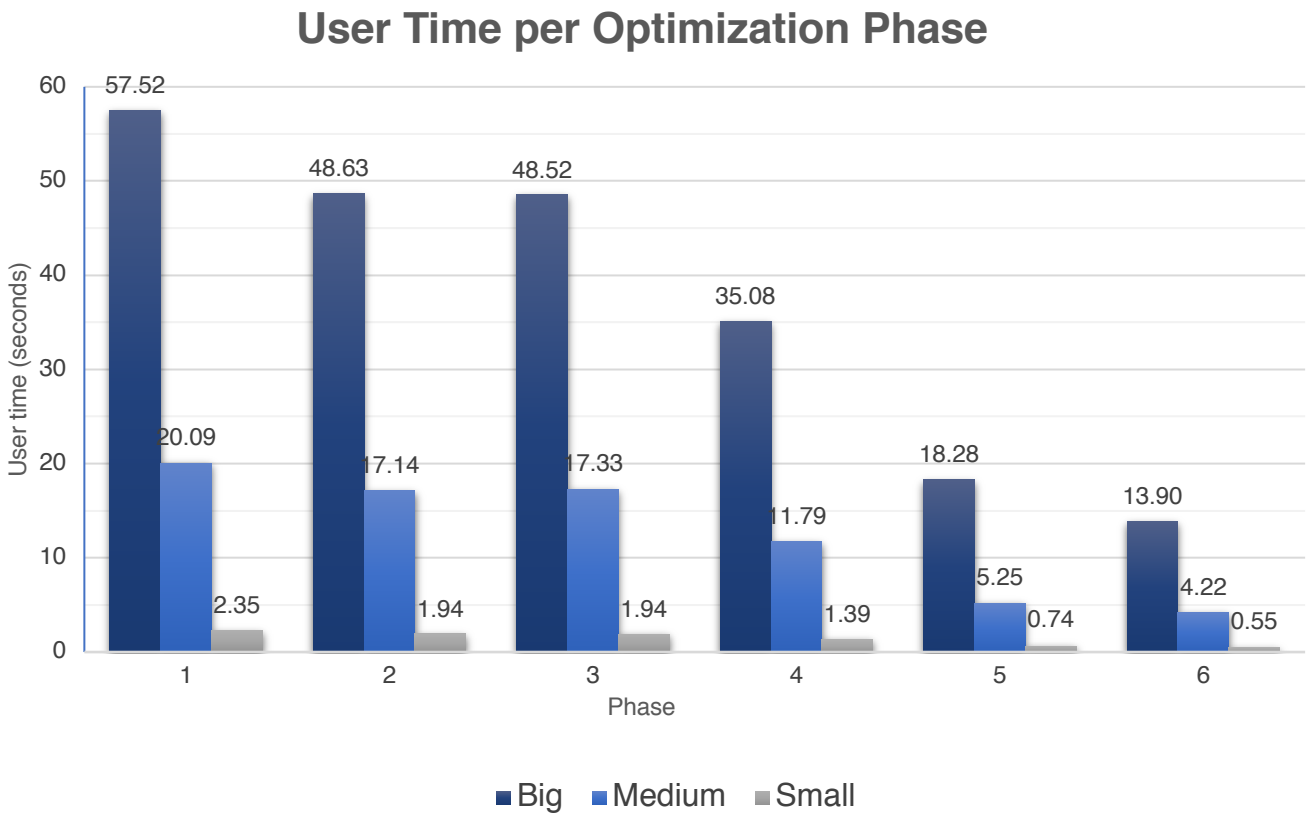
Table 1. Report for Improvements Through Profiling

Benchmark	*Run 1 Time (s)*	*Run 2 Time (s)*	*Run 3 Time (s)*	*Avg Time (s)*	*Instructions*	*Rel to Start*	*Rel to Prev*	*Improvement*	*Bottleneck*	*Phase / Stage*
Big	35.16	35.06	35.02	**35.08**		0.610	0.723	Eliminated function calls to bitpack_getu. Created smaller, inline version of bitpack_getu using local variables for mask and shift values. This removed assertions found in bitpack, and allows the use of static mask and shift values.	Seq_get, UArray_at	4
Medium	11.79	11.76	11.82	**11.79**		0.587	0.680			
Small	1.39	1.38	1.39	**1.39**	1.04E+10	0.590	0.714			
Big	18.29	18.39	18.15	**18.28**		0.318	0.521	Changed the program_counter to point to the next index directly. This was achieved by incrementing the program_counter by +1, rather than extracting the array from the sequence for each subsequent word.	segmented_load, segmented_store	5
Medium	5.24	5.26	5.26	**5.25**		0.261	0.446			
Small	0.72	0.76	0.73	**0.74**	6.94E+09	0.313	0.531			
Big	13.89	13.9	13.9	**13.90**		0.242	0.760	Removed temporary variables for Seq_get and Uarray_at pairs, removed function calls by integrating all instruction functions into one primary function.	Seq_get, UArray_at	6
Medium	4.32	4.16	4.17	**4.22**		0.210	0.803			
Small	0.55	0.55	0.55	**0.55**	6.12E+09	0.234	0.747			

See next page for visual representations.

Jay Facultad
Nathan Sahelijo

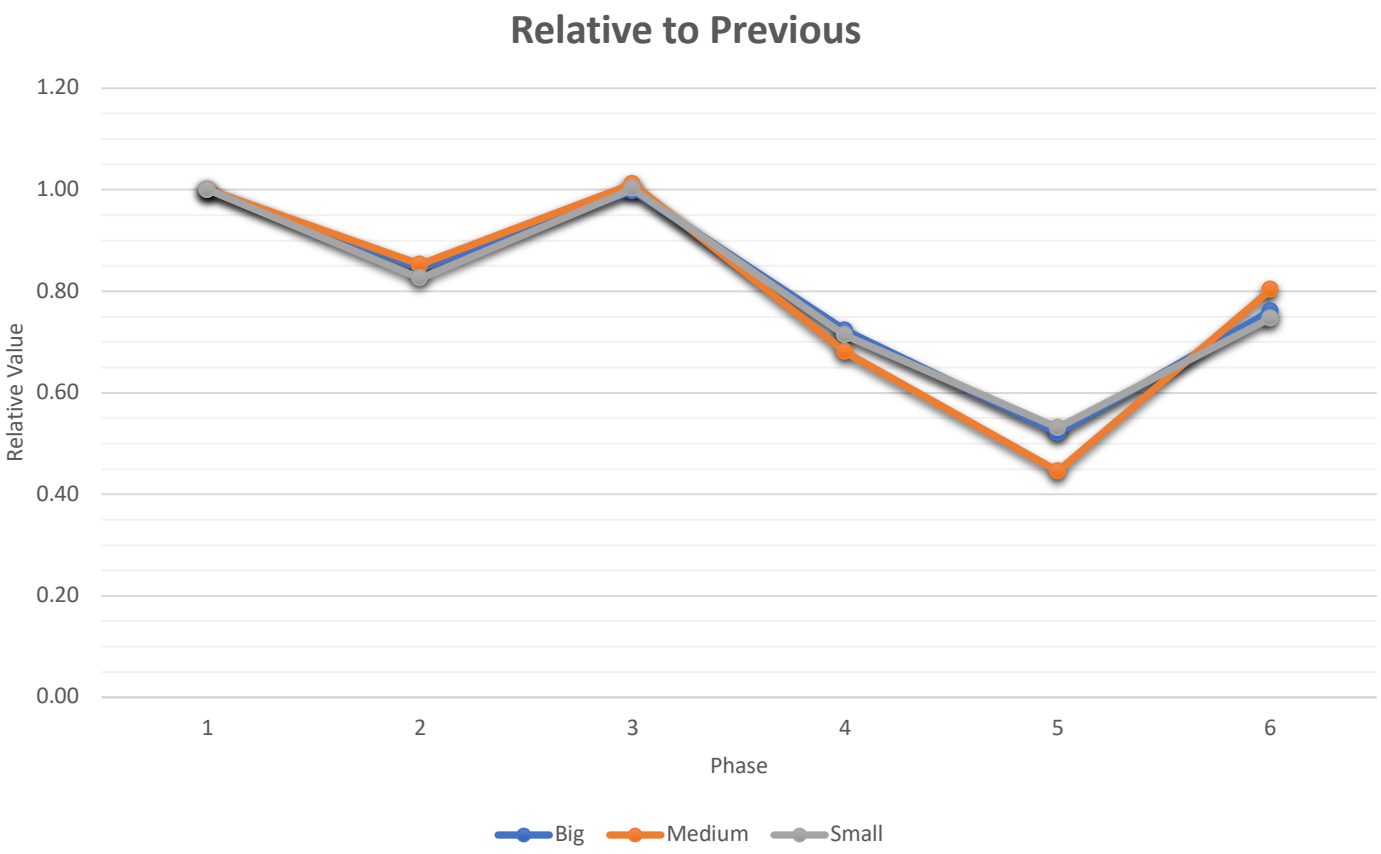
Laboratory Notes for Homework 7



Phase	*Improvements*
1	No improvement (starting point)
2	Compiled with optimization flag -O1 and linked against -lcii-O1
3	Compiled with optimization flag -O2 and linked against -lcii-O2
4	Eliminated function call to bitpack_getu. Created smaller verstion of bitpack_getus using local variables for mask and shift values. This removes assertions found in bitpack, and allows the use of static mask and shift values.
5	Changed the program_counter to point to the next index directly. This was achieved by incrementing the program_counter by +1, rather than extracting the array from the sequence at each subsequent word.
6	Removed temporary variables for Seq_get and Uarray_at pairs, removed function calls by integrating all instruction functions into one primary function

Jay Facultad
Nathan Sahelijo

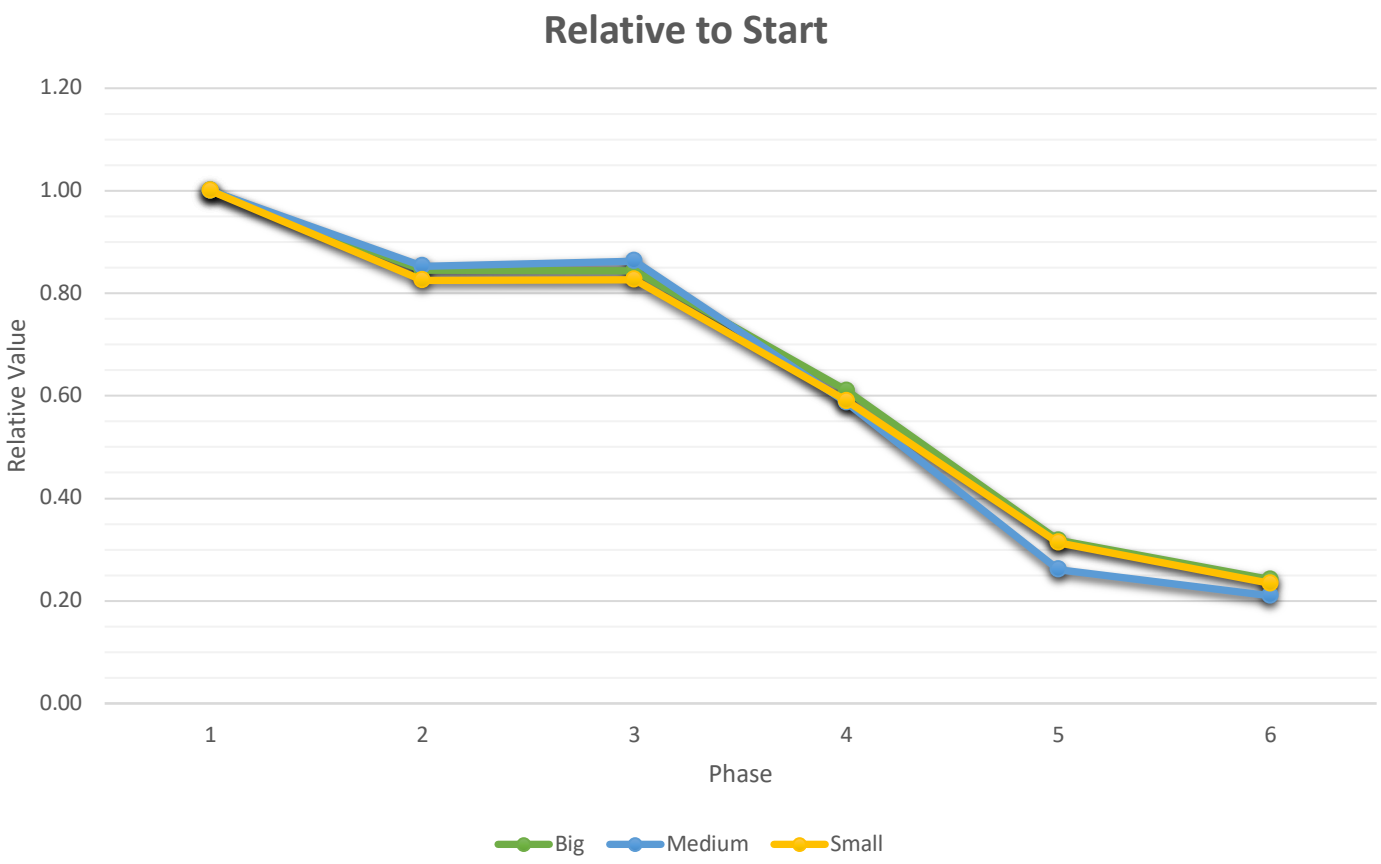
Laboratory Notes for Homework 7



Phase	Improvements
1	No improvement (starting point)
2	Compiled with optimization flag -O1 and linked against -lcii-O1
3	Compiled with optimization flag -O2 and linked against -lcii-O2
4	Eliminated function call to bitpack_getu. Created smaller verstion of bitpack_getus using local variables for mask and shift values. This removes assertions found in bitpack, and allows the use of static mask and shift values.
5	Changed the program_counter to point to the next index directly. This was achieved by incrementing the program_counter by +1, rather than extracting the array from the sequence at each subsequent word.
6	Removed temporary variables for Seq_get and Uarray_at pairs, removed function calls by integrating all instruction functions into one primary function

Jay Facultad
Nathan Sahelijo

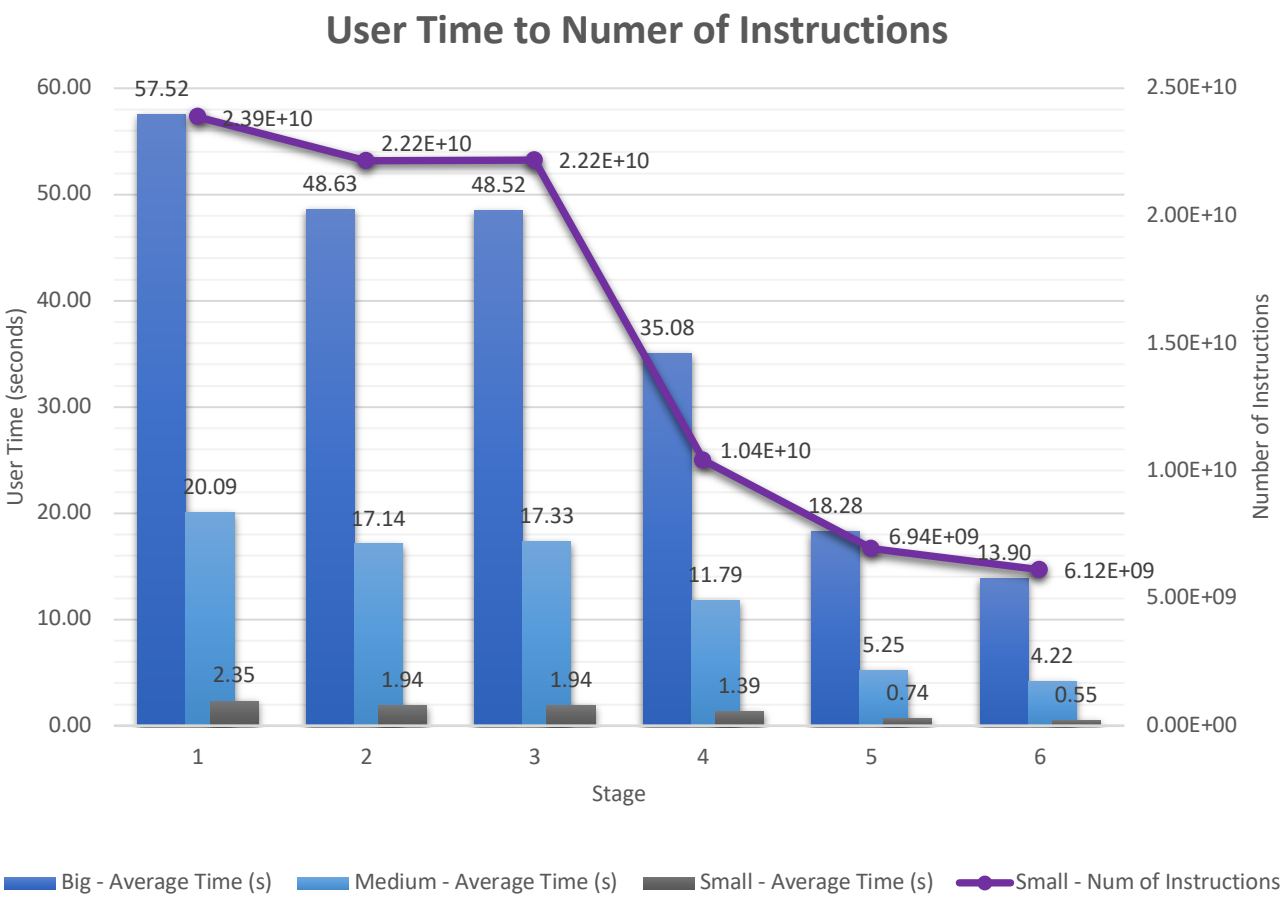
Laboratory Notes for Homework 7



Phase	Improvements
1	No improvement (starting point)
2	Compiled with optimization flag -O1 and linked against -lcii-O1
3	Compiled with optimization flag -O2 and linked against -lcii-O2
4	Eliminated function call to bitpack_getu. Created smaller verstion of bitpack_getus using local variables for mask and shift values. This removes assertions found in bitpack, and allows the use of static mask and shift values.
5	Changed the program_counter to point to the next index directly. This was achieved by incrementing the program_counter by +1, rather than extracting the array from the sequence at each subsequent word.
6	Removed temporary variables for Seq_get and Uarray_at pairs, removed function calls by integrating all instruction functions into one primary function

Jay Facultad
Nathan Sahelijo

Laboratory Notes for Homework 7



Phase	*Improvements*
1	No improvement (starting point)
2	Compiled with optimization flag -O1 and linked against -lcii-O1
3	Compiled with optimization flag -O2 and linked against -lcii-O2
4	Eliminated function call to bitpack_getu. Created smaller verstion of bitpack_getus using local variables for mask and shift values. This removes assertions found in bitpack, and allows the use of static mask and shift values.
5	Changed the program_counter to point to the next index directly. This was achieved by incrementing the program_counter by +1, rather than extracting the array from the sequence at each subsequent word.
6	Removed temporary variables for Seq_get and Uarray_at pairs, removed function calls by integrating all instruction functions into one primary function