

Performance Analysis: Bug 689153

Dave Eberly

dave.eberly@artifex.com

February 17, 2010

This report is about the performance problem processing the file 9322.pdf that is attached to the bug report. I process the file using the command line

```
gswin32c -r72 -q -sDEVICE=ppmraw -o nul: -Z: -dLastPage=1 9322.pdf
```

The source code was compiled using Microsoft Visual Studio 2005 and profiled on an Intel PC using Intel's Parallel Amplifier for timing. Intel's VTune was also used, but it provided no significantly different information. The elapsed time for the run is approximately 202 seconds. A screen capture of the top results from Parallel Amplifier is shown in Figure 1.

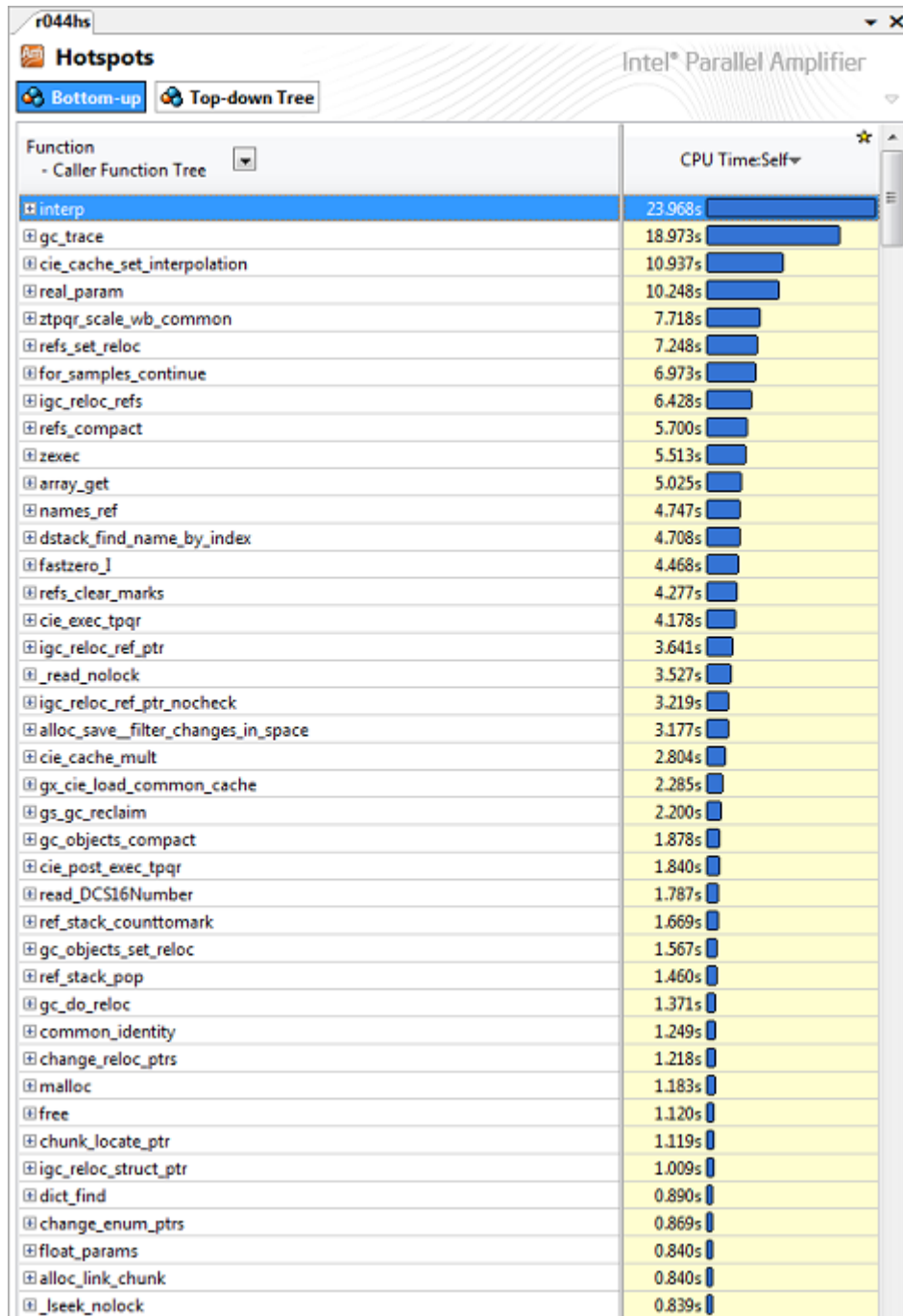


Figure 1. The summary of hotspots for 9322.pdf.

The majority of time is spent in the Ghostscript interpreter and in the garbage collection code. The function `interp` is called 625 times, but a lot of time is spent inside the function. The line of code using the most time within `interp` is `switch (r.type_xe(i_ref_packed))`. An experiment showed that this line is executed

324,886 times. The `gc_trace` function is called 334,486 times.

The third bottleneck is the function `cie_cache_set_interpolation`, as mentioned in the bug report. The function `cie_cache_set_interpolation` is called 1,493,313 times during the program execution. The `i`-loop is executed 512 times for each function call. The computational time is distributed equally for the three calls to `check_interpolation_required`. Thus, `check_interpolation_required` is called 2,293,728,768 times, which appears to be excessive.

A portion of the top-down hierarchy is shown in Figure 2.

Call Stack	CPU Time:Total	CPU Time:Self
Total	100.0%	0s
RtlInitializeExceptionChain	100.0%	0s
RtlInitializeExceptionChain	100.0%	0s
BaseThreadInitThunk	100.0%	0s
mainCRTStartup	100.0%	0s
_tmainCRTStartup	100.0%	0s
main	100.0%	0s
gsapi_init_with_args	100.0%	0s
gs_main_init_with_args	100.0%	0s
runarg	100.0%	0s
run_string	100.0%	0s
gs_main_run_string	100.0%	0s
gs_main_run_string_end	100.0%	0s
gs_interpret	100.0%	0s
gs_call_interp	100.0%	0s
interp	99.9%	24.976s
interp_reclaim	33.9%	0s
image_file_continue	12.4%	0.050s

Figure 2. Top-down hierarchy.

The call stack starting at `interp_reclaim` leads to the `gc_trace` function call. The call stack starting at `image_file_continue` leads to the `cie_cache_set_interpolation` call. For reference, Figures 3 and 4 have the call stacks to those function calls with the timing results.

[-] interp	99.9%	24.976s
[-] [-] interp_reclaim	33.9%	0s
[-] [-] [-] ireclaim	33.9%	0s
[-] [-] [-] [-] gs_vmreclaim	33.9%	0s
[-] [-] [-] [-] [-] context_reclaim	33.9%	0s
[-] [-] [-] [-] [-] [-] gs_gc_reclaim	33.9%	1.848s
[-] [-] [-] [-] [-] [-] [-] gc_trace	10.6%	19.417s
[-] [-] [-] [-] [-] [-] [-] [-] gc_do_reloc	8.9%	1.261s
[-] [-] [-] [-] [-] [-] [-] [-] [-] gc_objects_set_reloc	4.4%	1.290s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_objects_compact	3.8%	1.542s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] refs_clear_marks	2.3%	4.717s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] alloc_save_filter_changes	1.7%	0s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_strings_set_marks	0.4%	0.030s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_trace_finish	0.2%	0.080s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_trace_chunk	0.1%	0.180s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] names_trace_finish	0.1%	0.150s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] change_clear_marks	0.1%	0.181s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_free_empty_chunks	0.1%	0.070s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_clear_reloc	0.1%	0s
[-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] [-] gc_strings_compact	0.0%	0.090s
[-] gc_unmark_names	0.0%	0s
[-] gc_strings_set_reloc	0.0%	0.050s
[-] gs_heap_free_object	0.0%	0s
[-] i_status	0.0%	0.010s
[-] [Unknown frame(s)]	0.0%	0s
[-] context_state_load	0.0%	0s
[-] dstack_gc_cleanup	0.0%	0.010s
[-] context_state_store	0.0%	0s
[-] [Unknown frame(s)]	0.0%	0s

Figure 3. Call stack to gc_trace.

image_file_continue	12.4%	0.050s
↳ gs_image_next_planes	8.8%	0.070s
↳ gx_image1_plane_data	8.7%	0.070s
↳ image_render_color	8.7%	0.440s
↳ gx_default_remap_color	8.3%	0.040s
↳ gx_concretize_CIEICC	8.1%	0.130s
↳ gx_cie_check_rendering_inline	7.5%	0.020s
↳ gs_cie_jc_complete	7.5%	0.010s
↳ cie_joint_caches_complete	7.5%	0.050s
↳ cie_cache_mult	6.6%	2.670s
↳ cie_cache_set_interpolation	5.2%	10.581s
↳ cache_set_linear	0.4%	0.030s
↳ cie_cache_restrict	0.4%	0.809s
↳ cie_cache3_set_interpolation	0.0%	0.060s
↳ cie_mult3	0.0%	0.020s
↳ cie_cs_common_abc	0.0%	0.010s
↳ gx_cie_remap_finish	0.4%	0.030s
↳ icmLuMatrixFwd_lookup	0.2%	0.020s
↳ gx_restrict_CIEICC	0.0%	0.010s
↳ gx_cie_check_rendering	0.0%	0.010s
↳ gx_remap_concrete_DRGB	0.1%	0.020s
↳ gx_concrete_space_CIE	0.0%	0.010s
↳ gx_concrete_space_CIEICC	0.0%	0.010s
↳ gx_dc_pure_fill_rectangle	0.1%	0.030s
↳ gx_init_CIEICC	0.0%	0s
↳ gx_device_color_equal	0.0%	0.020s
↳ gx_get_cmap_procs	0.0%	0.010s
↳ gx_dc_no_equal	0.0%	0.010s
↳ gx_dc_pure_equal	0.0%	0.010s
↳ sample_unpack_8	0.0%	0.060s
↳ gx_image1_flush	0.0%	0.020s
↳ gx_image_plane_data_rows	0.0%	0.010s
↳ gs_image_cleanup_and_free_enum	2.3%	0.010s
↳ s_process_read_buf	1.2%	0.030s

Figure 3. Call stack to cie_cache_set_interpolation.