Noname manuscript No.
(will be inserted by the editor)

A Detailed Study of Ray Tracing Performance: Render Time and Energy Cost

Supplemental Materials

Elena Vasiou · Konstantin Shkurko · Ian Mallett · Erik Brunvand · Cem Yuksel

Published 2018

This document provides all additional data gathered during the analysis of time and energy classification costs for ray tracing. We present each statistic across all scenes, aggregated for clarity. This is to highlight the effects of scene size.

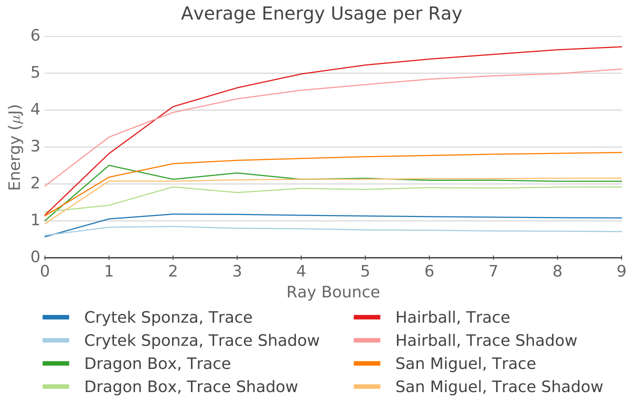


Fig. 1 Comparison of the energy contributions per ray for the Trace kernels (non- and shadow rays) for all scenes.

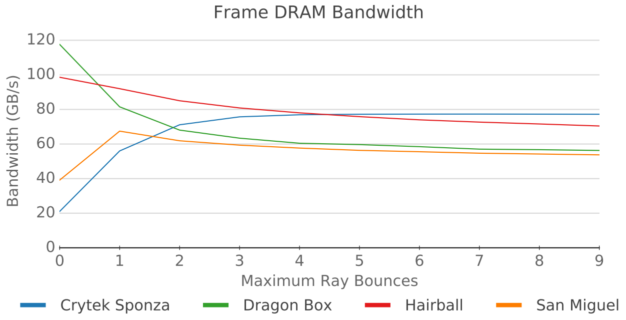


Fig. 2 DRAM bandwidth each scene used rendering to different maximum ray bounces.

University of Utah
50 Central Campus Dr, Sale Lake City, UT, 84112
E-mail: {elvasiou, kshkurko, imallett, elb}@cs.utah.edu, cem@cemyuksel.com

This is a post-peer-review, pre-copyedit version of an article published in The Visual Computer Journal

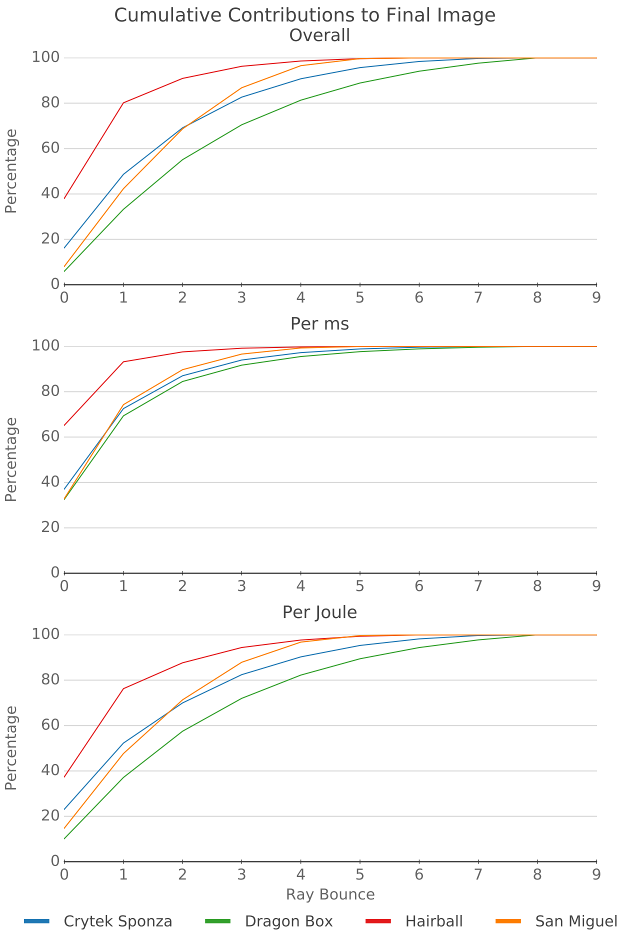


Fig. 3 Cumulative distribution of the percentage contribution to the final image using different metrics. All scenes rendered with maximum of 9 ray bounces.

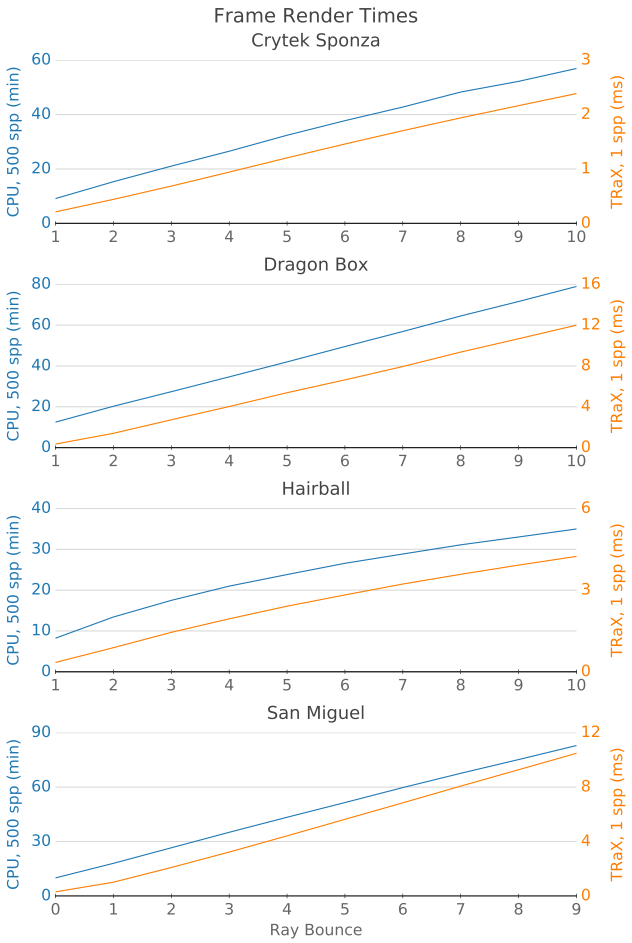
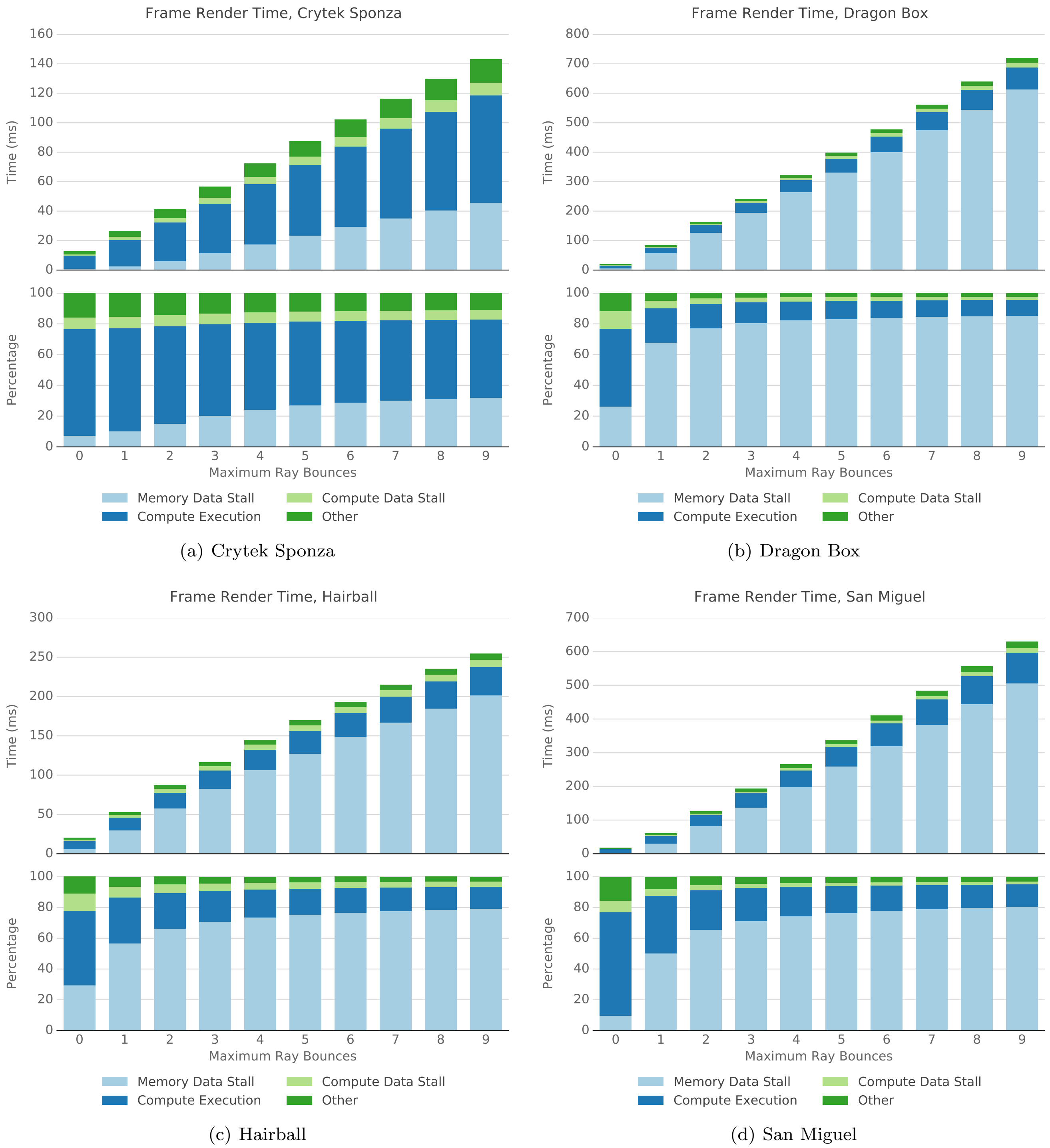


Fig. 4 Frame render times up to maximum of 9 bounces. The CPU implementation uses 500 samples per pixel (spp), while TRaX uses 1.



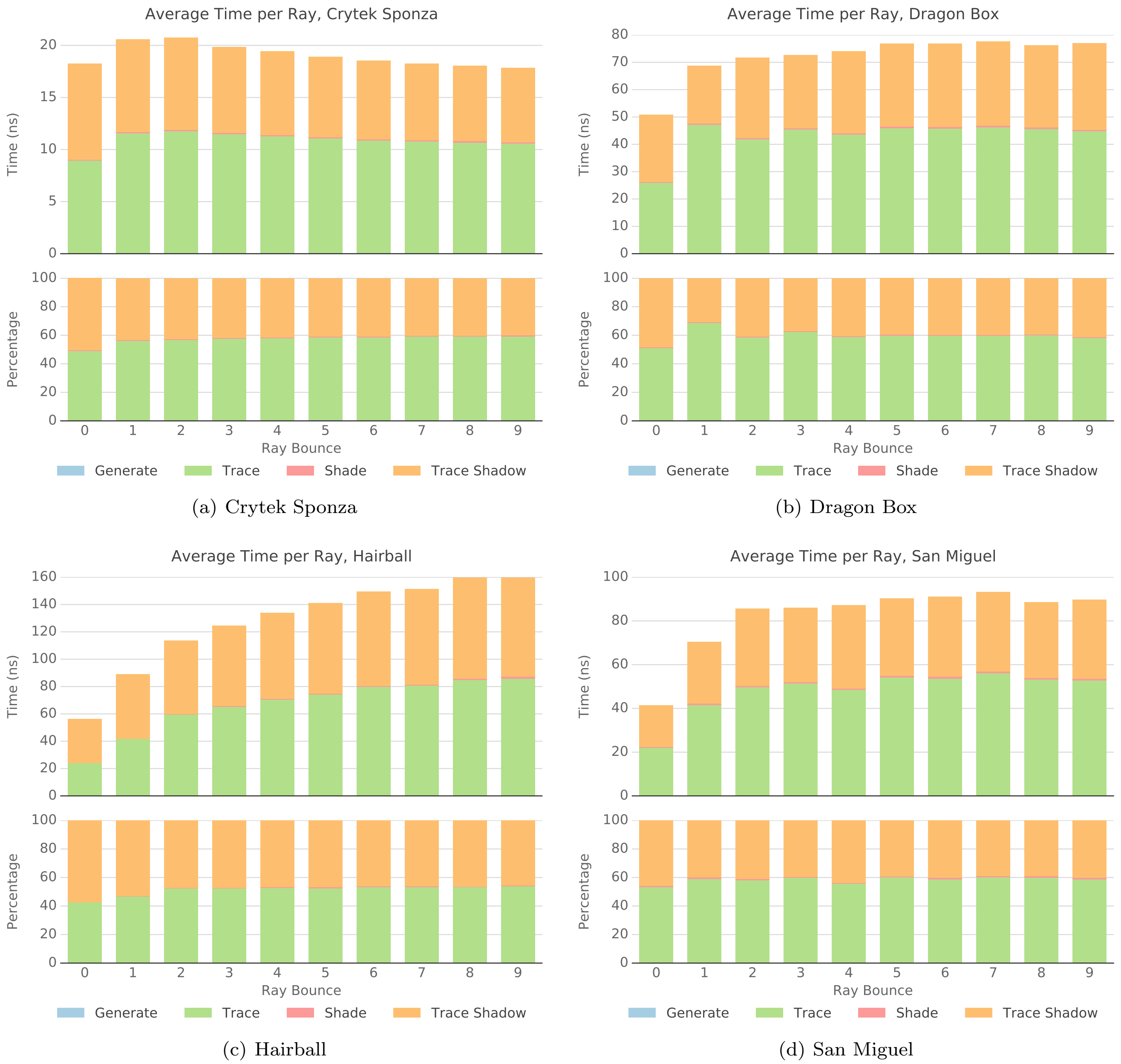
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 5 Distribution of time spent between memory and compute for a single frame of each scene rendered with different maximum ray bounces. Render times are dominated by memory stalls for the larger scenes, taking up approximately up to 80% for larger bounces. Crytek Sponza can mostly fit into cache and as such predominantly spends time on compute execution even for bounces larger than 1. For the other scenes, compute comprises a significant amount only for primary rays.



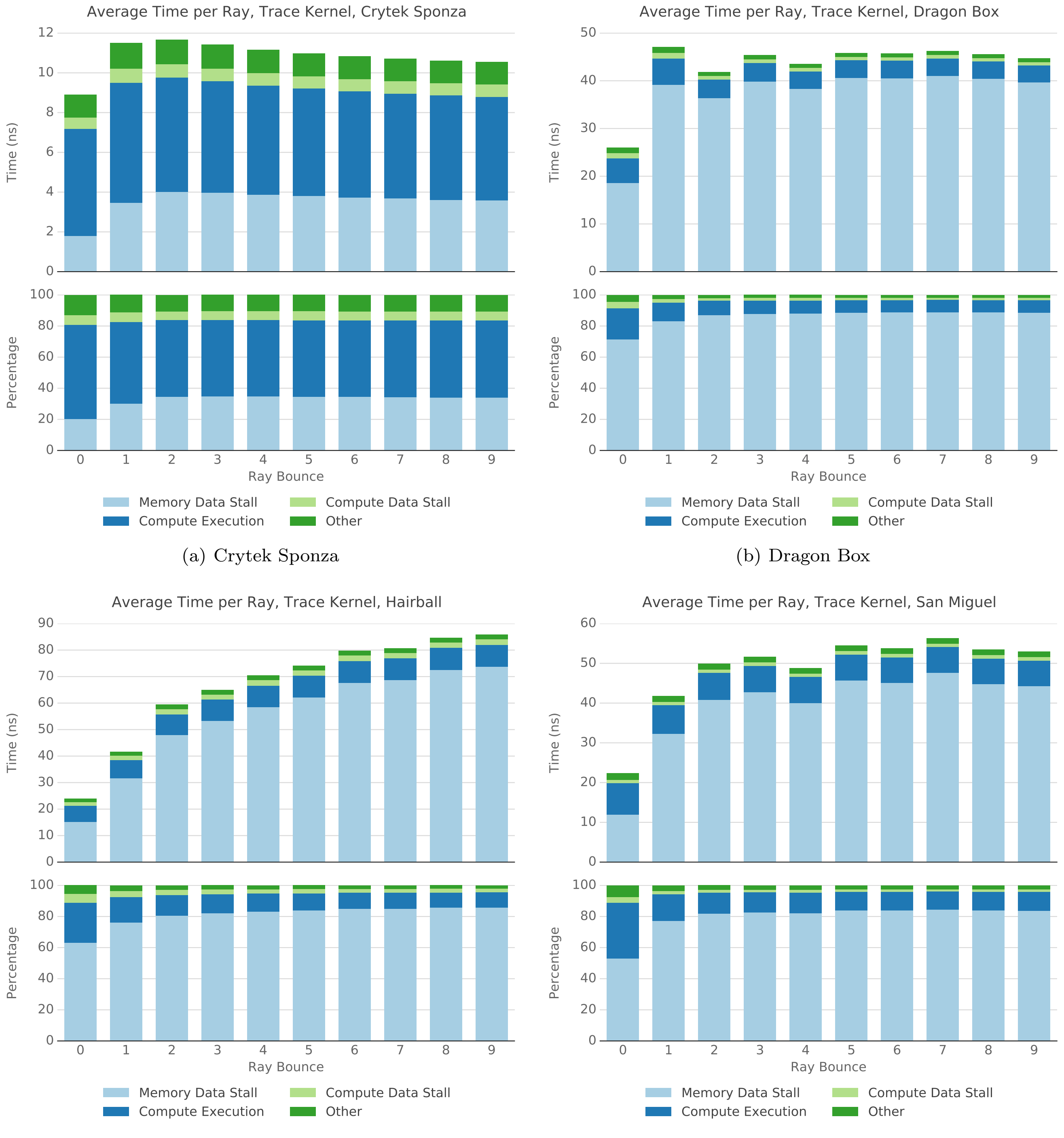
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 6 Classification of time per kernel normalized by the number of rays. Each scene rendered with maximum of 9 ray bounces. Note: contributions from the Generate and Shade kernels are negligible compared to others. Shadow Trace is somewhat less expensive than Non-Shadow Trace due to algorithm optimizations.



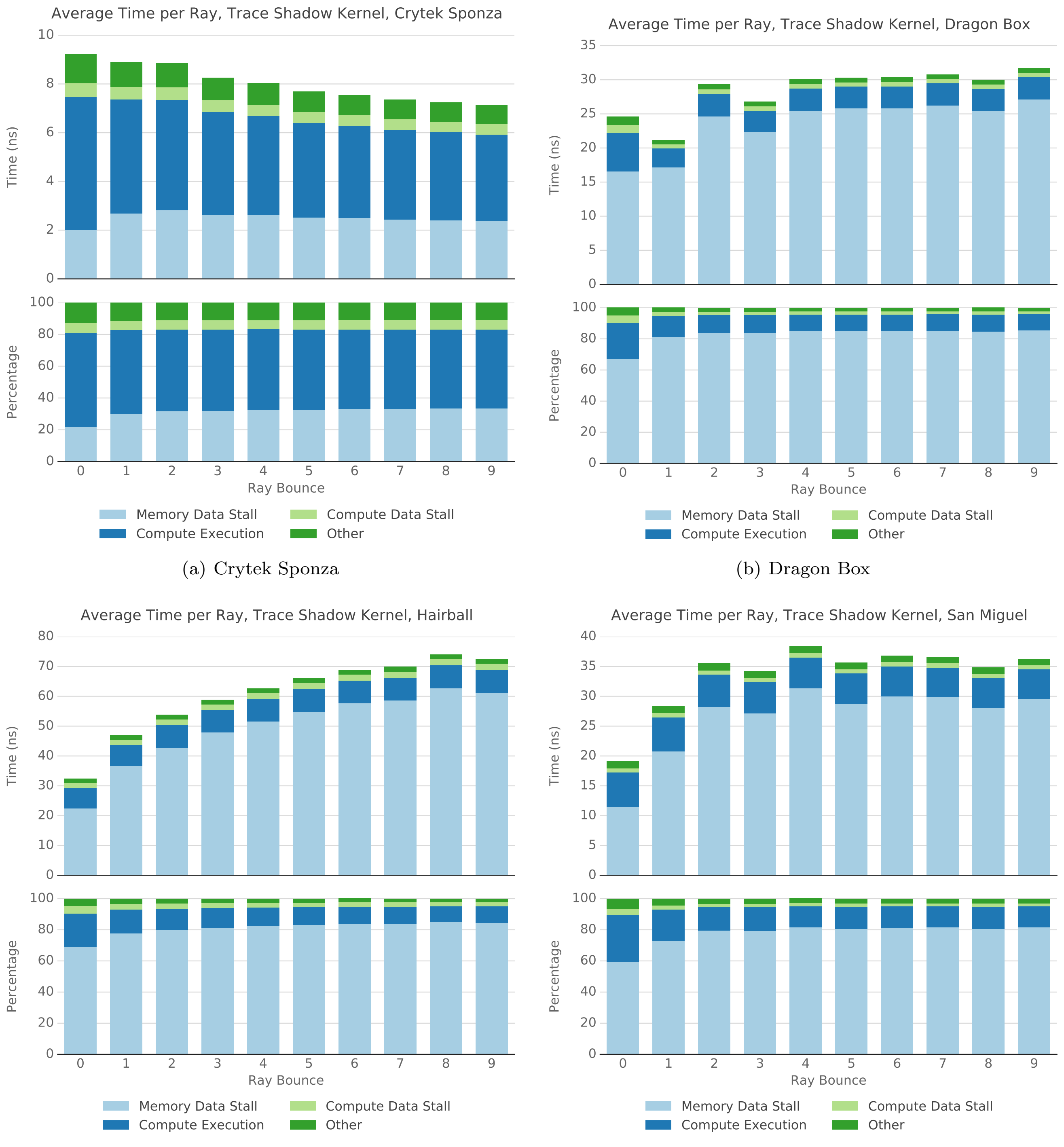
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 7 Classification of time per ray spent between memory and compute for the Trace kernel. Each scene rendered with maximum of 9 ray bounces. Similar to energy costs, the memory system takes up the majority of time spent for the Trace kernel. Note that memory stalls dominate all costs, except for in Crytek Sponza. Though even for that scene, stalls are a sizable portion of the total time spent.



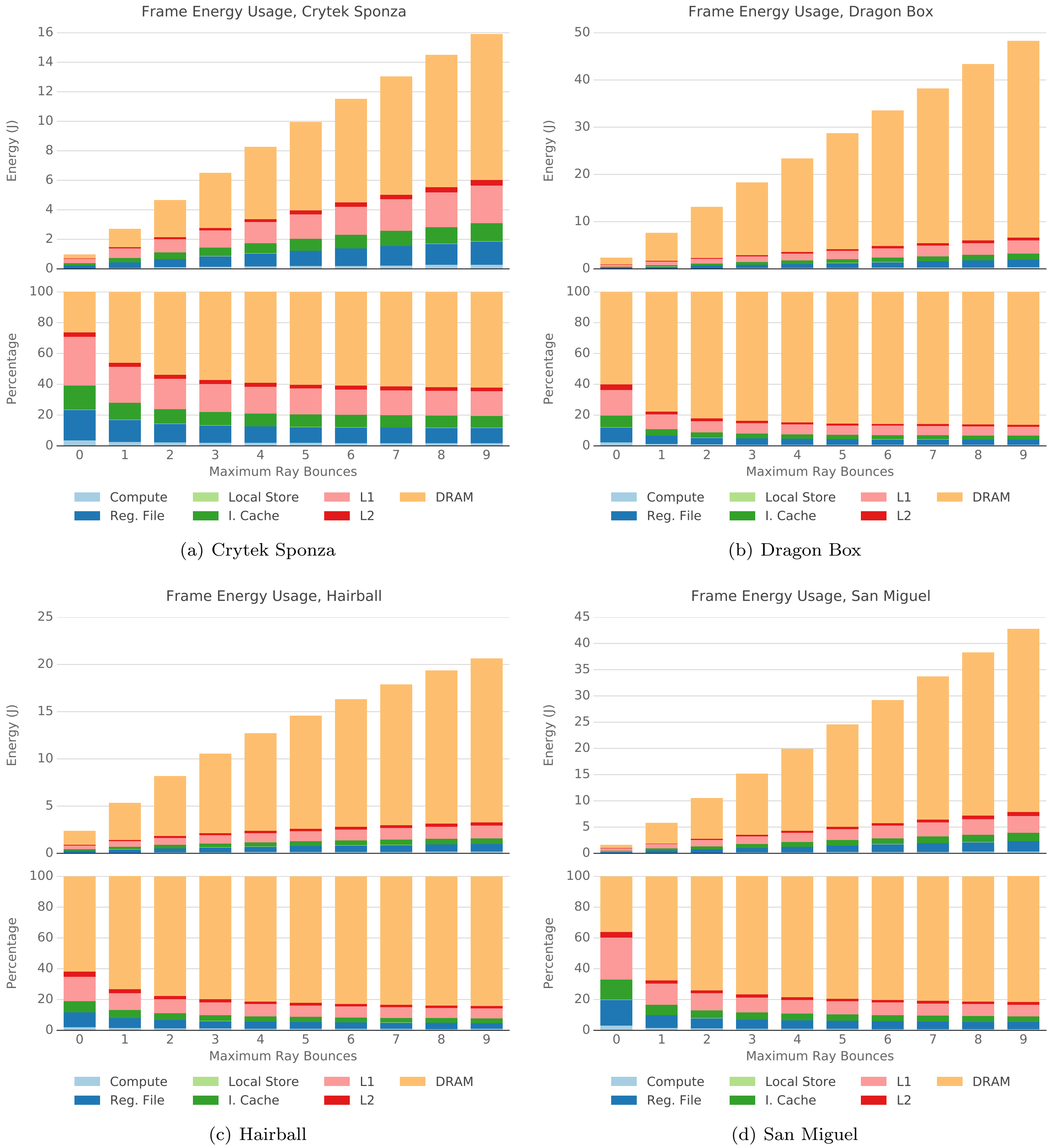
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 8 Classification of time per ray spent between memory and compute for the Trace Shadow kernel. Each scene rendered with maximum of 9 ray bounces. Similar behavior as to the one observed with the (non-shadow) Trace kernel. Memory contributions level off pretty quickly after the first bounce due to caches being thrashed and already working at capacity.



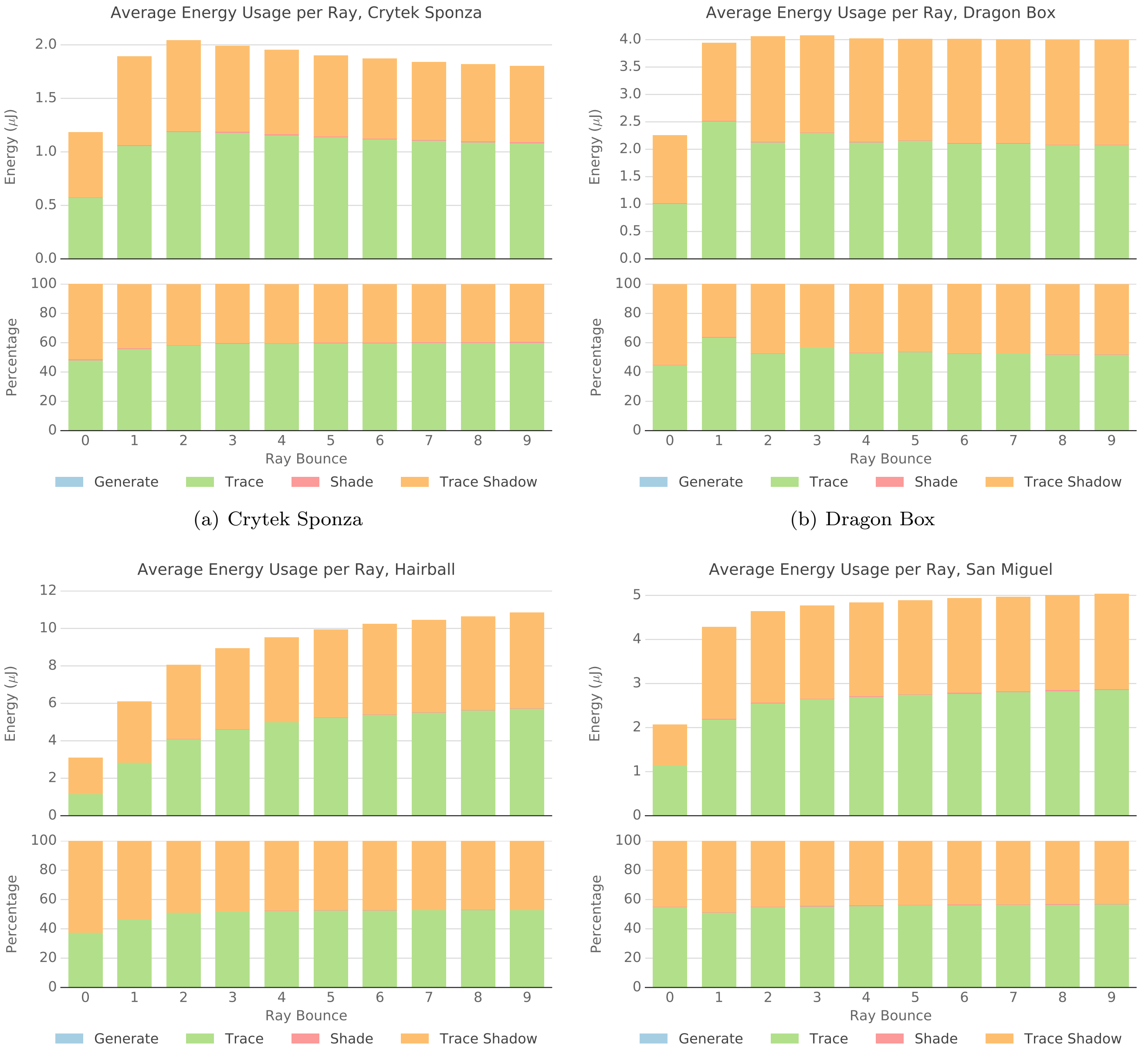
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 9 Classification of energy contributions by source for a single frame of each scene rendered with different maximum ray bounces. The majority of energy contribution originates from the memory system consistently throughout the four tested scenes.



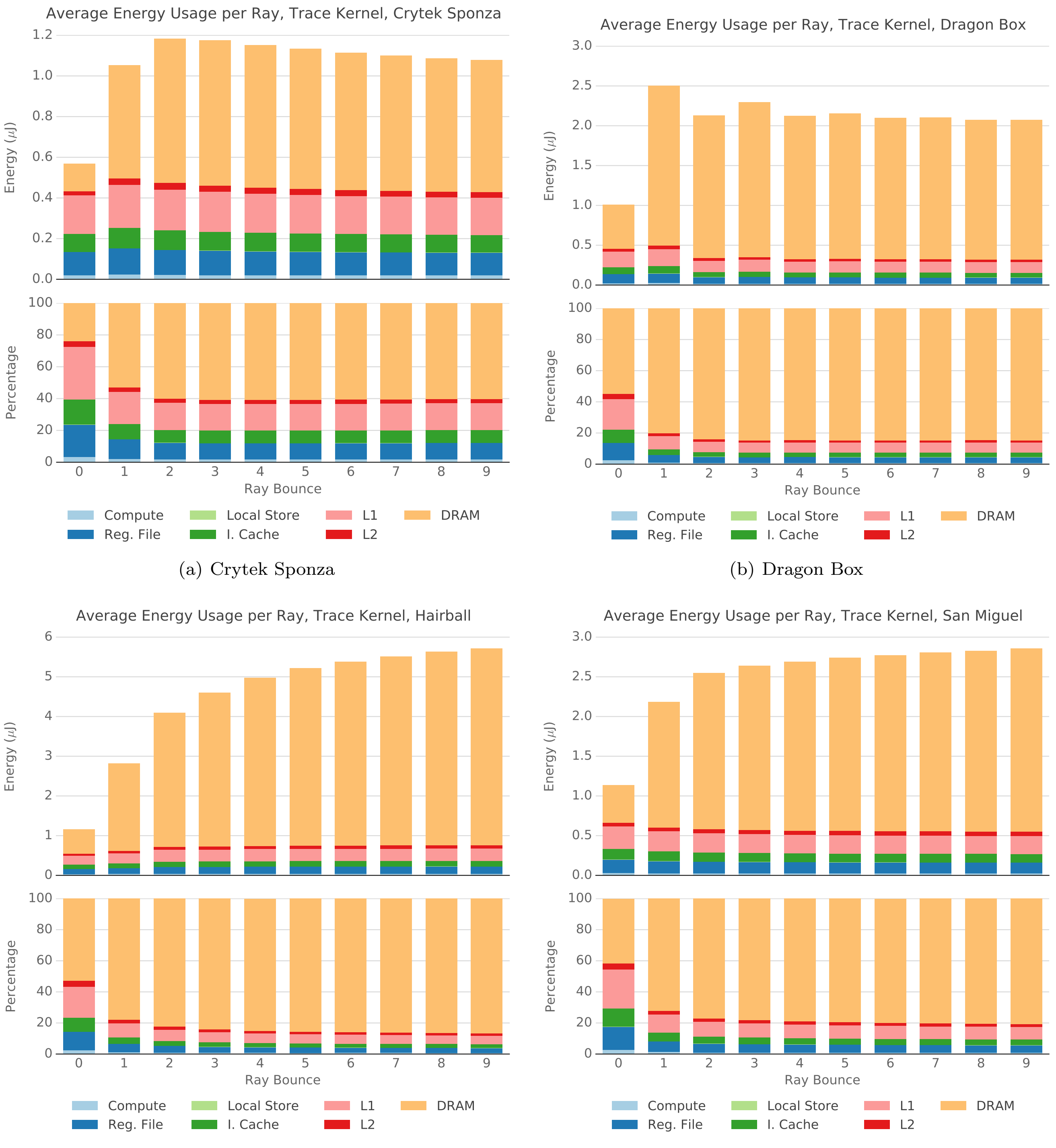
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 10 Energy classification per kernel normalized by the number of rays. Each scene rendered with maximum of 9 ray bounces. Note: contributions from the Generate and Shade kernels are negligible compared to others. Shadow Trace is somewhat less expensive than Non-Shadow Trace due to algorithm optimizations.



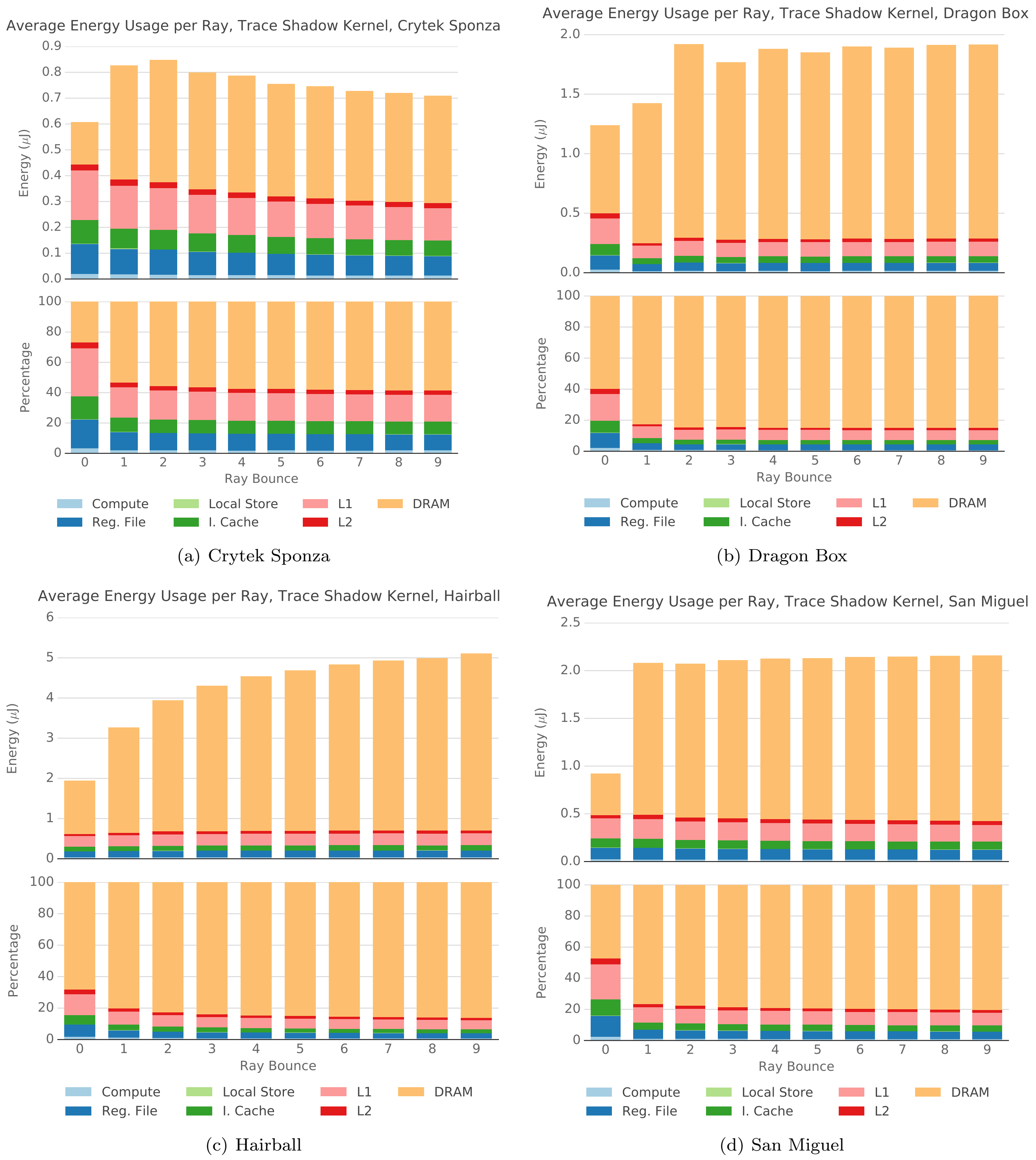
(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 11 Classification of energy contributions per ray by source for the Trace kernel. Each scene rendered with maximum of 9 ray bounces. Notice: despite hairball being smaller in triangle count than San Miguel, due to its depth complexity it has a very significant memory energy usage. Though the Dragon Box scene is relatively simple, the Dragon model can ”trap” rays thus leading to extensive use of memory for triangle intersections.



(a) Crytek Sponza

(b) Dragon Box

(c) Hairball

(d) San Miguel

Fig. 12 Classification of energy contributions per ray by source for the Trace Shadow kernel. Each scene rendered with maximum of 9 ray bounces. Shadow Trace behavior is similar to non-Shadow Trace. Again, DRAM is taking up the majority of energy contribution. Memory usage in Hairball scene is most significant of all due to depth complexity.