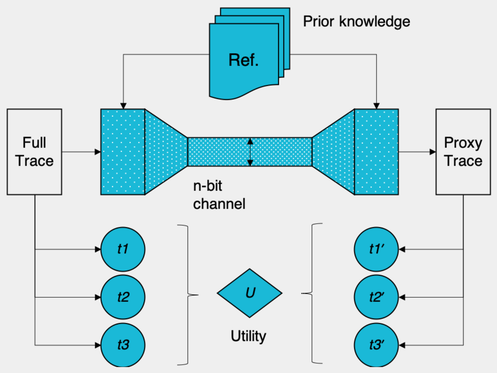
SAFER PROGRAM BEHAVIOR SHARING THROUGH

TRACE WRINGING

How do you share program trace information without giving away *ALL* of your private data? Trace wringing is a technique to minimize the number of bits shared while maximizing the utility of the proxy trace. We measure the utility in terms of whether or not certain tests (e.g. similarity of cache miss rates) are passed by the proxy trace. **We discover that in the leakage-utility tradeoff space, utility improves as more information is leaked.**



MEMORY ACCESS TRACES ARE COLLECTED FROM SPEC2006 BENCHMARKS

MODULO MEMORY HEATMAPS ARE GENERATED TO VISUALIZE MEMORY ACCESS BEHAVIOR

WITHIN A MODULO MEMORY HEATMAP OF A PROGRAM, WE OBSERVE THE OCCURRENCE OF ”PHASES”

WITHIN THE MODULO MEMORY HEATMAPS OF PROGRAM PHASES, WE OBSERVE “STRIDING BEHAVIORS” OF MEMORY ACCESSES. WE COLLECT THIS “LINE” INFORMATION THROUGH HOUGH TRANSFORMS. STRIDING BEHAVIORS ARE ASSIGNED DIFFERENT “WEIGHTS”

PACKETS WITH INFORMATION ABOUT PHASES, LINES, AND WEIGHTS ARE CREATED. THE SIZE OF THE PACKETS SHARED IS THE AMOUNT OF INFORMATION LEAKED.

A TRACE IS REGENERATED INTO A PROXY TRACE FROM INFORMATION IN THE PACKETS. WE THEN MEASURE THE UTILITY OF THE PROXY TRACES.

MODULO-MEMORY ACCESS HEATMAP

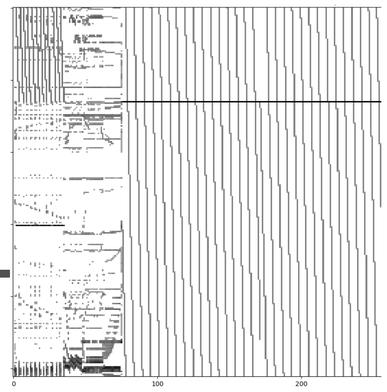


Fig. 1. The modulo-memory access heatmap for *gcc*. These heatmaps expose patterns that exist within program executions and give us a visual sense of memory access activity.

PHASE DETECTION

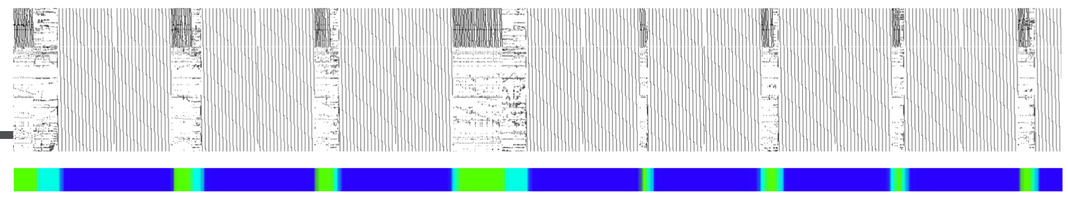


Fig. 2. The result of running the phase detector on the memory address trace for *gcc*. Each of the 3 colors labels the trace above it with a unique phase identifier.

LINE DETECTION

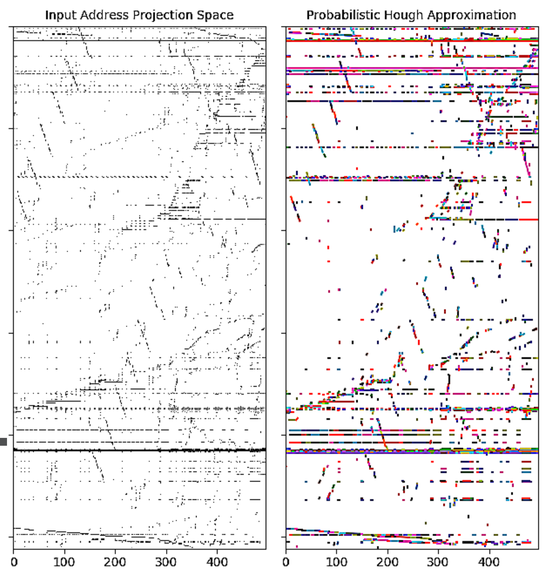


Fig. 3. Producing probabilistic Hough lines on top of the heatmap for *gcc*. The colors are used to represent distinct lines produced by the decomposition.

BIT-ERROR POINTS

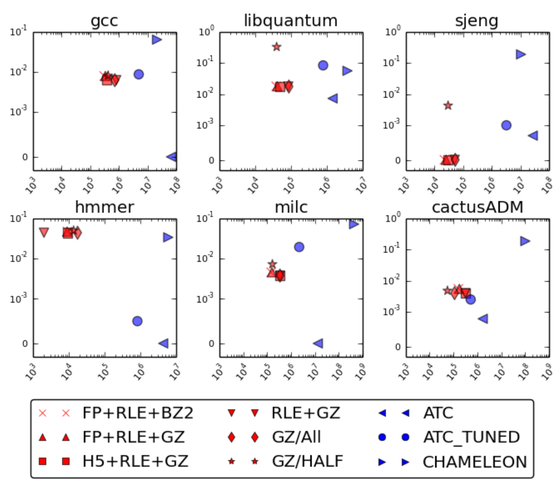


Fig. 4. We mark the bit-error points for different encodings. A packet contains information about Hough lines and labels.

PROXY TRACE

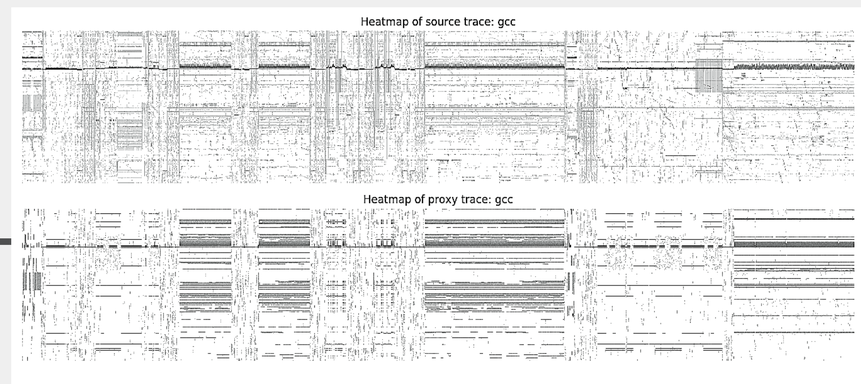


Fig. 5. Heatmaps for the original gcc memory access trace and the trace-wrung proxy generated from the shared information packets.

BEST CACHE MISS RATES OBSERVED

Benchmark	Bit Budget	Cache Configs. 8k,dm	8k,4w	16k,dm	16k,4w	32k,dm	32k,4w	Time Wringing	Time Decompression
gcc	Orig.	6.88%	3.91%	4.86%	2.79%	3.36%	2.11%	138.55s	123.37s
	Full	6.10%	3.98%	3.60%	1.27%	1.93%	0.48%		
	100k	4.82%	2.94%	2.81%	0.72%	1.40%	0.25%		
	10k	-	-	-	-	-	-		
sjeng	Orig.	12.3%	5.01%	6.45%	2.19%	4.24%	0.64%	94.42s	128.08s
	Full	12.85%	10.16%	8.22%	3.74%	4.26%	0.64%		
	100k	12.85%	10.16%	8.22%	3.74%	4.26%	0.64%		
	10k	11.89%	7.78%	1.13%	4.39%	0.25%	2.25%		
cactusADM	Orig.	8.29%	7.03%	5.44%	5.29%	2.09%	1.54%	209.94s	918.04s
	Full	9.35%	4.98%	5.21%	0.85%	2.08%	0.29%		
	100k	3.73%	0.49%	2.02%	0.14%	0.55%	0.12%		
	10k	-	-	-	-	-	-		
milc	Orig.	7.99%	7.09%	7.68%	7.03%	7.35%	6.94%	336.41s	31.36s
	Full	7.73%	7.19%	7.11%	6.66%	5.93%	5.69%		
	100k	7.51%	7.25%	6.75%	6.44%	5.46%	5.44%		
	10k	-	-	-	-	-	-		
hmmer	Orig.	27.8%	2.54%	26.8%	1.20%	17.0%	0.78%	151.79s	287.95s
	Full	23.6%	7.21%	20.53%	5.05%	10.31%	4.32%		
	100k	23.6%	7.21%	20.53%	5.05%	10.31%	4.32%		
	10k	23.6%	7.21%	20.53%	5.05%	10.31%	4.32%		
libquantum	Orig.	16.3%	16.2%	16.2%	16.2%	16.2%	16.2%	57.73s	21.89s
	Full	17.31%	17.27%	14.99%	14.90%	12.10%	11.90%		
	100k	17.31%	17.27%	14.99%	14.90%	12.10%	11.90%		
	10k	74.46%	74.44%	69.33%	69.31%	59.31%	59.32%		

Table 1. We report ground truth miss rates from the *original trace*, best miss rate using *all Hough lines*, with *100k bits*, and with merely *10k bits*.

UC SANTA BARBARA

Deeksha Dangwal*, Weilong Cui, Joseph McMahan, Timothy Sherwood

UC Santa Barbara

***deeksha@cs.ucsb.edu**