

lab 5 performance chart
Nicholas Wachter, Polina Volnuhina

matmul -PI	-00	-01	-02	-03
Avg CPI	5.953	16.987	17.059	17.086
Instructions	41,525,317,221	11,379,032,028	11,371,142,868	11,364,894,358
Branch misses	27,949,169	27,659,228	28,934,587	28,912,964
Cache misses	1,086,726,311	1,079,740,536	1,081,992,274	1,080,950,300
Cache refs	22,614,938,323	3,230,370,997	3,228,364,531	3,227,204,037
Runtime	207.95463(sec)	162.068036(sec)	164.132003(sec)	163.439578(sec)

matmul -ARM	-00	-01	-02	-03
Avg CPI	1.707	2.337	0.943	0.952
Instructions	141,504,263,952	20,059,326,230	17,909,779,366	17,905,208,967
Branch misses	22,773,657	19,543,794	22,411,400	19,230,040
Cache misses	33,971,572	33,735,970	33,691,784	33,693,062
Cache refs	56,870,086,528	5,258,105,705	5,257,057,791	5,256,534,068
Runtime	120.900322(sec)	23.531558(sec)	8.466954(sec)	8.639256(sec)

W/ best optimization	Original best	Loop Unroll	Column Major	LoopUnroll+Col Major
Runtime	27.869406(sec)	30.734604(sec)	7.954153(sec)	8.658616(sec)
Cache misses	1,078,260,801	1,116,411,842	33,642,081	33,706,178
Cache References	3,105,900,664	5,256,534,058	3,105,900,666	5,256,534,058

Results explanation: Both with the raspberry pi and arm processor, each optimization level tends to get better in terms of performance as optimization goes up. The optimizations that I see from updating performance was especially large for switching from row major to column major. So while we see more consistent improvement from compiler optimizations there is potential for huge improvements for code optimization. It is also important to note that overall, when cache misses and caches references go down, runtime will almost always be quicker.